Real-time image processing approach to measure traffic queue parameters

M. Fathy
M.Y. Siyal

Indexing terms: Image processing, Road traffic, Queue parameters, Real-time measurement

Abstract: The real-time measurement of various traffic parameters including queue parameters is required in many traffic situations such as accident and congestion monitoring and adjusting the timings of the traffic lights. In case of the queue detection, at least two algorithms have been proposed by previous researchers. Those algorithms are used for queue detection and are unable to measure queue parameters. The authors propose a method based on applying the combination of noise insensitive and simple algorithms on a number of sub-profiles (a one-pixel-wide key-region) along the road. The proposed queue detection algorithm consists of motion detection and vehicle detection operations, both based on extracting edges of the scene, to reduce the effects of variation of lighting conditions. To reduce the computation time, the motion detection operation continuously operates on all the sub-profiles, but the vehicle detection is only applied to the tail of the queue. The proposed algorithms have been implemented on an 80386-based microcomputer system and the whole system works in real-time.

1 Introduction

In recent years the application of image processing techniques in automatic traffic monitoring and control has been investigated by several researchers [1–4]. The major problem concerning any practical image processing application to road traffic is the fact that the real-world images are to be processed in real-time. In a real-world road-traffic scene, the variation of lighting conditions, different shape or size of vehicles, scene geometry, occlusion and uncontrollable motions make serious difficulties to measure different parameters. Real-time image processing is not easily achievable, particularly when the algorithms are complex, unless special hardware is used.

Current efforts of image processing applied to traffic can be divided into quantitative and qualitative analysis. Most of the reported projects were concentrated on quantitative analysis to measure simple parameters such as vehicle counts etc. The qualitative analysis, which is about the description of the traffic scene, is still at its early stage. This paper concentrates on measuring queue parameters such as its length, the period of occurrence and slope of the occurrence of each queue period. The method proposed here can be used both in the quantitative and qualitative analysis.

The application of image processing in queue detection have been investigated by at least two researchers [6, 7]. Rourke and Bell [7] have developed a FFT-based traffic monitoring and queue detection method, but this method has not been applied for quantitative analysis. The FFT-based approach, while reducing data by processing a window along the road, is still time consuming and is unable to measure the length of the queue [7]. The full frame approach proposed by Hoose [6] is also unable to measure the length of the queue and the other queue parameters. This method is mostly suitable for describing the traffic status.

The approach described in this paper is a spatial domain technique and has been implemented in real-time using a low-cost system. The aim of the algorithm is to measure the queue parameters more accurately rather than just to detect it. The queue parameters can give more valuable information than queue status to traffic engineers and traffic controllers in many traffic situations. The proposed algorithm consists of two operations, one involving motion detection and the other vehicle detection. These operations are applied to a profile consisting of small profiles with variable sizes (sub-profiles) to detect the size of the queue. To compensate for any effects of transfer of physical view to the image, the size of the sub-profile varies according to the distance from the camera and camera geometry. The motion detection is based on applying a differencing technique on the profiles of the images along the road, and the vehicle detection is based on applying edge detection on these profiles.

2 Queue detection algorithms

Image processing algorithms are generally divided into spatial domain and frequency domain techniques. In the case of traffic, for an empty road when there is no vehicle, no shadows and no road markings, the histogram of the road contains no or very few high grey values and shifts towards the origin. When the road contains objects, the histogram shifts towards a higher grey value. This effect has been used by several researchers to detect objects. Alternatively, the amplitude of FFT for an empty road contains a high component at origin (DC), and only few low components at high frequency. These high frequency components are due to the image noise and electrical noise. When the road contains objects, the high frequency components have higher amplitudes and when the road
contains more objects, the high frequency components have even higher amplitudes. These high frequency components are due to the various colours and the shapes of the vehicles in the view. So, the frequency spectrum of the surface of the road can be used for vehicle detection and for measuring other traffic parameters. This effect can also be used for queue and congestion detection and was introduced by Rourke and Bell [7] in traffic application.

As shown in Figs. 1 and 2, we have implemented the FFT algorithm on an 80386-based microcomputer and processing speeds of 0.2 frames/second have been achieved for a profile of 512 pixels length. The processing speed achieved by FFT algorithm is sufficient for real-time queue detection, as the status of a queue rarely changes within a few seconds. However, setting a proper threshold value for this method of queue detection is not straightforward.

3 Spatial domain queue detection technique

As we needed to detect the queue and measures its parameters, we decided to concentrate on a method which is less sensitive to noise and easily implementable in real-time. To detect and measure queue parameters, two different algorithms have been used. The first algorithm is a motion detection and the second is a vehicle detection operation. As the microcomputer systems operate sequentially, a motion detection operation is firstly applied and then if the algorithm detects no motion, a vehicle detection operation is used to decide whether there is a queue or not. The reason for applying motion detection first is that in this case vehicle detection mostly gives the positive result, while in reality there may not be any queue at all. So by applying this scheme, the computation time is further reduced.

4 Motion detection operators

A simple method for motion detection is based on differencing two consecutive frames and applying noise removal operators. In this method the histogram of the key region parts of the frames are analysed by comparing with a threshold value to detect the motion. To reduce the amount of data and to eliminate the effects of minor motions of the camera, the key region has to be at least a 3-pixel-wide profile of the image along the road. In this method, a median filtering operation is firstly applied to the key region (profile) if each frame and a one-pixel-wide profile is extracted. Then the difference of two profiles is compared to detect for motion. This process is shown in Fig. 3.

![Fig. 1 FFT of a profile of a traffic scene: no object in the scene](image1)

![Fig. 2 FFT of a profile of a traffic scene: objects in the scene](image2)

![Fig. 3 Algorithm of motion detection](image3)

![Fig. 4 Profile of traffic scene: first frame of moving vehicles](image4)
than the case when there is no motion. Therefore, the motion can be detected by selecting a threshold value and in this approach the selection of the threshold value is an easy process.

Fig. 5  Profile of traffic scene: second frame of moving vehicles

Fig. 6  Profile of traffic scene: difference between first and second frames of moving vehicles

The size of the profile for queue detection is an important parameter as there might be motion in some part of it, while there may not be motion in the other parts. So, the profile along the road is divided into a number of smaller profiles (sub-profiles) with variable sizes and the motion detection algorithm operates continuously from the front sub-profile up to the other sub-profiles which detect queues and does not operate on the next sub-profile. The size of the sub-profiles are reduced by the distance from the front of the camera, to compensate the effect of the transfer of the three-dimensional view of the camera to a two-dimensional image. This transformation causes the equal physical distances of the profiles to be transferred to the unequal distances according to the camera parameters, such as height, field of view, angle of the optical axis and the number of the lines in the image.
By knowing the coordinates of 6 reference points of the real-world condition and the coordinates of their corresponding images, the camera parameters \((a_{11}, a_{12}, \ldots, a_{34})\) are calculated. However, in our case we have assumed a flat plane traffic scene \((Z_0 = 0)\), hence the operations are simplified as follows:

\[
\begin{bmatrix}
X_0, Y_0, 1, 0, 0, 0, -Y_1X_0, -Y_1Y_0 \\
0, 0, X_0, Y_0, 1, -Z_1X_0, -Z_1Y_0
\end{bmatrix} = \begin{bmatrix}
a_{11} \\
a_{12} \\
a_{13} \\
a_{21} \\
a_{22} \\
a_{23} \\
a_{31} \\
a_{32} \\
a_{33}
\end{bmatrix}
\]

(1)

In this case, by knowing the coordinates of four reference points, the camera parameters are estimated. The method implemented here is based on using the above equation and by knowing the real-world length of some object and measuring the image length of these objects.

The above equation is used to reduce the sizes of the sub-profiles in such a way that each sub-profile represents an equal physical distance. In this manner a threshold value can be selected for all sub-profiles, for the queue detection purpose. The number of sub-profiles along the roadside depends on the resolution and the accuracy required. However, the size of the profiles should not be too small so that the effect of the noise could not be eliminated. Our experiments show that the length of sub-profile should be about the length of the vehicle, in order to assure that the operation of both vehicle and motion detection algorithms work accurately.

5 Vehicle detection algorithms

Following the application of the motion detection operation, a vehicle detection operation is applied on the profile of the unprocessed image. Many algorithms have been developed by various researchers for vehicle detection. To implement the algorithm in real-time, two strategies are often applied: key region processing and simple algorithms. Most of the vehicle detection algorithms developed so far are based on a background differencing technique. However, this method is sensitive to the variations of ambient lighting and it is not suitable for real-world applications.

The method used here is based on applying edge detector operators to a profile of the image. Edges are less sensitive to the variation of ambient lighting and have been used for detecting objects in full frame applications. The method used here is based on applying an edge detector, consisting of separable median filtering and morphological operators, SMED (separable morphological edge detector) [5, 8] to the key regions of the image. The SMED edge detection has shown to have a low computational cost and is less sensitive to noise, compared with many other edge detectors. In this vehicle detection approach, the SMED is applied to each sub-profile of the image and the histogram of each sub-profile is processed by selecting dynamic left-limit value and a threshold value to detect vehicles.

The left-limit selection program selects a grey value from the histogram of the window, where there are approximately zero edge points above this grey value. When the window contains an object, the left-limit of the histogram shifts towards the maximum grey value, otherwise it shifts towards the origin. The process of left-limit value selection is repeated for a large number of frames (100 frames), and the minimum of the left-limit of these frames are selected as the left-limit for the next frames (100 frames). The histograms of a profile of a traffic scene following the application of the SMED operator for different traffic conditions is shown in Figs. 10–12.

![Histogram of a window containing no object](image1.png)

Fig. 10 Histogram of a window containing no object
Total pixels: 588

![Histogram of a window containing a large part of an object](image2.png)

Fig. 11 Histogram of a window containing a large part of an object
Total pixels: 588

For threshold selection, the number of edge points greater than the left-limit grey value of each window is extracted for a large number of frames (200 frames) to get enough parameters below and above a proper threshold value. These numbers are used to create a histogram (Fig. 13), where its horizontal and vertical axes correspond to the number of edge points greater than left-limit and the frequency of repetition of these numbers for a period of operation of the algorithm (200 frames). This histogram is smoothed using a median filter and we expect to get two
peaks in the resultant diagram (Fig. 14), one peak related to the frames passing a vehicle and the other related to the frames without vehicles for that window. However, as it is seen in Fig. 14, there are other numbers of edge points (30–40) between the two peaks 20 and 60, which are related to the vehicles which partially pass the window. We use a statistical approach based on selecting a point on the horizontal axis of Fig. 14, where the sum of the entropy of the points above and below this point is maximum. This point is selected as the threshold value for the next period.

This method of left-limit and threshold selection algorithm compensates for variations in lighting conditions and other background changes. This algorithm imposes no significant computation overhead as uses the information extracted by the vehicle detection operation.

6 Results and discussions

The main queue parameters we were interested in identifying were the length of the queue, the period of occurrence and the slope of the occurrence of the queue behind the traffic lights. To measure these parameters on a desired road, the program works in such a way that after each 10 seconds, the presence of the queue and its length is reported. To implement the algorithm in real-time, it was decided that the vehicle detection operation should only be used in a sub-profile where we expect the queue will be extended (tail of the queue). This procedure is shown in Fig. 15.

A traffic scene with different queue conditions is shown in Figs. 16–19. The results of the operations of the algorithms on the traffic scene of Figs. 16–19 for a period of 40 minutes along with a manual measurement of the queue is shown in Fig. 20. As it can be seen from Fig. 20, the queue is slowly building up behind the traffic light (for example from time 10 to 100), and then it disappears sharply (for example at time 100). The reason for a rapid disappearance is that the algorithm has been implemented in such a way that the queue condition is not detected, when there is motion in the front sub-profile and slow motion in other sub-profiles. The algorithm can also be easily implemented in other manners required by the traffic experts (for example if motion is on two front sub-profiles, then a queue is not detected). The results of operations of the algorithms compared with manual observations of images confirm that the queues are detected and its parameters are measured accurately in real-time.
The time required to execute the queue detection with our system varies between 0.3 to 0.5 s from the time when there is no queue and when the queue is fully present in the scene. The time required for vehicle detection is less than 0.2 s, as this is the time required for the largest sub-profile which exists in front of the scene. The time required for motion detection is 0.12 s. This is the time when the queue is fully present in the scene and the algorithm is executed for the entire scene.

Considering the other times required for decision making, reporting and threshold selection, the average processing speed is about 2 frames per second. The threshold selection operation using an entropy function to find the valley point of the histogram, takes about 1 s while with a heuristic approach to get a good approximation of a valley point, it takes about 0.25 s. However, the threshold selection is only executed in variable periods greater than 6 minutes, in the times when there is no queue and the vehicle detection and the motion detection is only executed for the first sub-profile. The whole system is in operation and works in real-time.

The algorithm for measuring queue parameters is composed of vehicle detection and motion detection. As measuring the error (or the difference) of queue length measurement under various lighting and weather conditions is not easy (the percentage of the error is very low),
we decided to test the algorithm by separately testing the vehicle detection and motion detection algorithms. The results of vehicle detection algorithms under various weather and lighting conditions are shown in Fig. 21. As can be seen there are up to 5% errors, due to changing of lanes by the drivers.

![Graph showing vehicle count under different weather conditions](image)

**Fig. 21** Result of vehicle count

The motion detection algorithm is stable under various lighting and weather conditions, as the lighting conditions remain constant during the time difference between two consecutive frames.

7 Conclusions

An algorithm for measuring queue parameters such as the period of occurrence between queues, the length and the slope of occurrence, have been introduced in this paper. The algorithm uses a new technique by applying a combination of simple but effective operations and has been implemented in real-time on an 80386 based microcomputer system. In order to reduce the computation time, a motion detection operation is applied on all sub-profiles, while the vehicle detection operation is only applied when it is necessary. The vehicle detection operation uses an edge-based technique which is less sensitive to noise. The threshold selection is done dynamically to compensate the effects of variations of lighting and it does not introduce any significant computational cost. The queue measurement algorithm has been applied to various traffic scenes with different lighting conditions, using a DT2867 frame grabber and an 80386DX PC-based system. The queue detection and measuring algorithm along with measurements of other traffic parameters such as counting and speed measurement (using the proposed vehicle detection algorithm), all work in real-time on our system. The results show that this queue measurement approach can determine the length of the queue to within 95% accuracy. This error is mainly due to the objects located very far from the camera and can be reduced by adjusting the size of sub-profiles more appropriately, by analysing camera parameters more accurately (as we have used a simplified transformation function for transferring 3-dimensional images to 2-dimensional images).

The results of the operation of the algorithms on the desired traffic scene show that the algorithm can accurately measure traffic parameters with an acceptable resolution and accuracy. No attempt has been made to increase the resolution of measuring the queue parameters, as in practice, measuring the exact values of queue parameters have no usage for traffic control and measurement. However, there should not be any problem to increase the resolution up to a specific value, using the proposed method.

A practical implementation of this approach called ‘Variable Sign System’, has been operational since early 1995. This system alarms the drivers for heavy traffic, one kilometre before the intersection. This intelligent sign system even directs the drivers to alternative routes, when it detects heavy traffic on their paths.

8 References