A neural-vision based approach to measure traffic queue parameters in real-time

M.Y. Siyal *, M. Fathy

School of Electrical and Electronics Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

Received 3 April 1998; received in revised form 30 March 1999

Abstract

The real-time measurement of queue parameters is required in many traffic situations such as accident and congestion monitoring and adjusting the timings of the traffic lights. Previous methods proposed by researchers for queue detection are based on traditional image processing algorithms. The method proposed here is based on applying the combination of edge detection and neural network algorithms. The edge detection technique is used to detect vehicles and estimate the motion, while neural network is used to measure the queue parameters. The neural network is trained for various road traffic conditions and is able to provide better results than the traditional image processing algorithms. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Image processing; Edge detection; Queue detection; Neural networks

1. Introduction

The application of image processing for queue detection has been investigated by at least two researchers. Rouke and Bell (1991) developed an FFT-based queue detection method, but this method is unable to measure the length of the queue. The full frame-based approach proposed by Hoose (1991) is also unable to measure the length of the queue. The approach described in this paper is based on edge detection and neural network techniques and has been implemented in real-time using low-cost system. The aim of the algorithm is to measure the queue parameters 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 queue detection algorithm consists of a motion detection and vehicle detection operation. These operations are applied to a profile consisting of small profiles with variable sizes (sub-profiles) to detect the size of the queue. 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 first step for the queue detection operation is to detect vehicles. A common and simple vehicle detection technique is “Background differencing technique”. This technique is based on pixel-by-pixel comparison of a background image of a traffic scene (without any moving vehicles) and the
current frame of the scene and has been used by various researchers (Dickinson and Waterfall, 1984; Dickinson and Wan, 1989; Inigo, 1997). In practice, the effectiveness of this method depends on the accuracy of the background updating techniques and selection of a suitable threshold value. Various researchers (Ali and Dagles, 1992; Fathy and Siyal, 1995) have proposed different algorithms for background updating techniques. However, they do not work well under various lighting and weather conditions and therefore, are not suitable for real-world applications, such as traffic analysis (Dickinson and Wan, 1989; Hoose, 1991).

An alternative vehicle detection technique used in image processing is based on edge detection technique. Edge-based vehicle detection is generally more effective than background differencing and has been used by few researchers in traffic applications (Hoose, 1991; Fathy and Siyal, 1995). In fact, the edge information of the objects remains still significant despite the variation of the ambient lighting. The analysis of traffic images has proved that various surfaces and different parts and colours of a vehicle create significant edges. Even the cars, which have the same colour as the surface of the road, reflect more light and can be detected.

There are two types of edge-based detection techniques: conventional gradient-based edge detectors and morphological edge detectors. The conventional gradient-based edge detection operations have found wide acceptance in image processing applications. However, morphological edge detectors have shown better performance than conventional edge detectors while having a lower computational cost (Fathy and Siyal, 1997).

The second step for the queue detection operation is to detect and measure the queue parameters. So far, research efforts in this area have been concentrated on investigating traditional image processing approaches, which have not provided satisfactory results (Rouke and Bell, 1991; Hoose, 1991; Ali and Dagles, 1992; Fathy and Siyal, 1997, 1995). In this paper a novel neural network approach is introduced which can be trained for different kinds of roads and can detect and measure queue parameters. We have conducted extensive tests and experiments and have compared results with the traditional image processing methods. The results show that the neural network approach provides better results.

2. Queue detection algorithm

As we were interested to detect the queue and measure its parameters, we decided to concentrate on a method which is less sensitive to noise and can be easily implemented in real-time on a Pentium-based microcomputer system. To detect and measure queue parameters, two different algorithms have been used. The first algorithm is a motion detection algorithm 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 operation first is that the traffic scene we analyse for queue detection are expected to contain vehicles and in this case vehicle detection operation mostly will give positive result, while in reality there may not be any queue at all. So by applying this scheme, the computation time is further reduced.

2.1. The motion detection algorithm

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 it with a threshold value to detect the motion. To reduce the amount of data and to eliminate the effects of minor motion of the camera, the key region has to be at least a 3-pixels wide profile of the image along the road. In this method, a median filtering operation is firstly applied to the key region (profile) of each frame and then, the difference of the two profiles is compared to detect motion. When there is motion, the difference of the profiles is larger than the case when there is no motion. Therefore, the motion can be easily detected. This process is shown in Fig. 1. The results of the operations on the profiles of a road containing motion and stopped conditions are
shown in Fig. 2. The review of Fig. 2 indicates that when there is motion, the difference of the profiles is larger than the case when there is no motion.

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-profiles up to the sub-profiles, which detect the queue, and does not operate on the next sub-profiles (Fig. 3). 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 operations of both algorithms (the vehicle detection and motion detection) work accurately.

2.2. The vehicle detection algorithm

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 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, called SMED (Separable Morphological Edge Detector) (Fathy and Siyal, 1995) to the key regions of the image (Fig. 4). The SMED edge detection has shown to have a low computational cost and is less sensitive to noise compared with many other edge detectors. Following the application of edge detection operation, the number of pixels having greater value than the threshold is used to recognise a vehicle. The threshold value is automatically selected by analysing the histogram.

After applying the edge detection operation, a status vector with values 0 (zero) or 1 (one) is obtained for each window. A zero (dark point) value determines no vehicles and a one (white point) shows a vehicle on that frame. By analysing the status vector various queue parameters are calculated. The main queue parameters we were interested to extract are 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 every 10 s, the presence of the queue and its length is reported graphically. To implement the algorithm in real-time, the vehicle detection operation is only used in a sub-profile where we expect the queue will be extended (tail of the queue). This procedure is shown in Fig. 5.

3. Measuring queue parameters by neural network

To detect and measure queue parameters, we first detect vehicles by using the vehicle detection operation described above. We use neural networks to analyse the status vector and measure the queue parameters.
Fig. 2. The results of the operations on the profiles of a road containing motion and stationary conditions.
A neural network with 10 inputs, 10 outputs and a hidden layer was used for pattern recognition and measuring the queue parameters. In this method, the pattern vector of 10 consecutive frames is applied to a back propagation neural network. Each input of neural network is the percentage of white points for that frame and each output represents the length of queue for 10 frames. The training of neural network is done for each road separately. In our approach, 200 samples are used for training. The neural network trainer is shown in Fig. 6. As it can be seen, the user can set the parameters for the training.

The training involves three stages:
1. The feed forward of input training pattern.
2. The back propagation of the associative error.
3. The adjustments of the weights.

The flowchart of the training algorithms is shown in Fig. 7.
Following the training of the network, the computed weights \( W_{ij} \) are extracted and are used to compute the outputs as follows:

\[
S_j = \sum_i W_{ij} * X_i, \quad i = 0, \ldots, \text{input}, \quad j = 1, \ldots, \text{hidden},
\]

where input and hidden are the numbers of inputs and hidden layers in our system. These values are constant and equal to 10. The output of each hidden layer is computed using signed function as follows:

\[
h_j = (1 + e^{-S_j})^{-1}, \quad j = 1, \ldots, \text{hidden},
\]

where \( S_j \) is obtained from Eq. (1). The main outputs of neural network is computed as follows:

\[
S_j = \sum_i W_{ij} * h_i, \quad i = 0, \ldots, \text{hidden}, \quad j = 1, \ldots, \text{output}.
\]
Fig. 7. Neutral network training algorithm.
When $h_j$ is computed from (2) and output is constant value equal to 6. The output of network is given as

$$O_j = (1 + e^{-x})^{-1}, \quad j = 1, \ldots, \text{output}.$$  \hspace{1cm} (4)

4. Results

The algorithm has been implemented in a user-friendly package in MS Windows 95/98 environment. The result of queue parameters by traditional image processing method along with a manual measurement of the queue is shown in Fig. 8, while Fig. 9 represents the results by neural network approach. As it can be seen, the neural network approach provides better results. The average error rate for traditional image processing is 18.9% (Table 1), while the error rate for neural network approach is only 1.7% (Table 2). The detailed data for both approaches is given in Tables 1 and 2 for a period of 24 h.

The time required to execute the queue detection on our system (PentiumII-400-based microcomputer) varies between 0.08 and 0.12 s for the time when there is no queue and when the queue is fully

---

![Fig. 8. The result of queue parameters (traditional approach).](image1)

![Fig. 9. The result of queue parameters (neural network approach).](image2)
present in the scene. Considering the other time required for decision making, and reporting, the average processing speed of about 6 frames per second is achieved. This speed is found to be sufficient for on-line reporting of queue parameters.

5. Conclusions

An algorithm for measuring queue parameters has been introduced in this paper. The algorithm uses a new technique by applying a combination of edge detection and neural network operations. 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. Neural network is more accurate than the heuristic approach as considering all states of window pattern is difficult in real-world traffic scene because the vehicles move with different speeds and directions. In this case, an intelligent approach such as neural network, which is trained for many traffic scenes and situations, can compute the parameters more accurately.

References