Abstract: Various researchers for a number of years have investigated traffic data collection and analysis. Many techniques have been proposed to speed up the operations and some intelligent approaches have been developed to compensate the effects of lighting, shadows and occlusions. To achieve real-time processing, it is necessary to reduce the amount of data to be processed, so the first question is to determine the suitable location for the key regions or windows. Previously, researchers have either used full frame processing approach, which requires more computing power and thus is not practical for real-time applications, or have used window based techniques, but defining of windows was done manually, which is not practical in real-world traffic applications. In this paper, an automatic approach is described to measure the size and location of windows required for this purpose. The results demonstrate that this method provides better results than the fixed window size.

Keywords: Image Processing, Transportation, Pattern Recognition, and Segmentation.

1. Introduction

The real-time measurement and analysis of various traffic parameters are increasingly required for traffic control and management. Image processing technique is now considered as an attractive and flexible method for automatic analysis and data collections in traffic engineering [1-4].

The first image processing algorithm required in traffic application is filtering and segmentation. Background differencing is the simplest technique to segment the traffic images [5]. However, this technique has problems of accurately updating the background frame and automatically selecting a suitable threshold value. Edge detection based segmentation of traffic scene has the advantage of being less sensitive to variation of ambient lighting and shadows [6]. However, combined background differencing and edge detection has the advantage of eliminating stationary vehicles, shadows and the road markings and is less sensitive to variations of lighting [2]. We use this vehicle detection technique to detect moving vehicles and measure traffic parameters such as vehicle count and the queue parameters.

To achieve, real-time processing it is necessary to reduce the amount of data to be processed, so the first question is to determine the suitable location for the key regions or windows. Previously, researchers have either used full frame processing approach [1,5,7], which requires more computing power and thus is not practical for real-time applications, or have used window based techniques [3], but defining of windows was done manually, which is not practical in real-world traffic applications. In this paper, an automatic approach is described to measure the size and location of windows required for this purpose. We have conducted extensive tests and experiments using automatic window approach and have compared results with the fixed window size methods. The results show that our approach provides better results.

2. The Vehicle Detection Process

The algorithm used for vehicle detection is based on applying low pass filtering and combined background differencing and edge detection operations on windows located across the roads. This vehicle detection technique has better performance than the background differencing operations as colours of vehicles and the ambient lighting changes in traffic scenes are less sensitive to edge detection. Following the application of vehicle detection operation, the number of pixels having greater value than the threshold is used to recognise a vehicle [6].

2.1 The size of Window

The size of the window has a strong effect on the performance of the whole system. The window size should be located vertical to the road direction for
vehicle counting. As the width of a window is increased up to the length of vehicles, the accuracy increases but the computation time increases, which slows down the whole system. On the other hand, the width of the window is dependent on the road type, as in highways (expressways, motorways etc.) the vehicles move more rapidly than other roads. On highways the width of the windows should be more, as there is a risk of missing vehicles on some frames.

We have conducted extensive experiments in order to determine the best window size for various situations. The percentage of errors of vehicle detection algorithm versus window width for a road is shown in figure 1. The graph of the computation times against window width for the same road is shown in figure 2. As it can bee seen, a compromise can be made between the accuracy needed and a processing speed required by selecting the width of the window.

In our system the area to locate windows and the number of lanes on the road and the road type is determined by the operator manually. Then the size of windows is computed and is located automatically across the road. The width of the windows is adjusted every half an hour time periods regularly as the relative speed of the vehicle changes during daytime. The width of a window is computed as follow:

\[ W = L \times V \]  

Where \( L \) is the length of each window and is defined on the location of window on image and \( V \) is a speed factor determined by the operator and is adjusted automatically by comparing relative speed of vehicles continuously.

Following computing the windows and their sizes, the vehicle detection program uses the co-ordinates of each window to detect vehicles.

3. Measurement Of Traffic Parameters

As described earlier, to detect vehicles, windows with suitable sizes are placed across each lane of the road. In this vehicle detection approach, the edge detector is applied to each window, and the number of edge points is compared with a threshold value to decide, whether the window contains the object or not. The threshold value is automatically calculated by processing the histogram of each window.

Following the application of edge detection operation to the windows a status vector is created.

To count the number of vehicles, the status vector is analysed. In this manner, for each frame, if a vehicle is detected in the window, a ‘1’ is stored at the status vector of the window; otherwise, a ‘0’ is stored at the status vector of the window. The group of 1s (ones) corresponds to a vehicle and a group of 0s (zeros) corresponds to the distance between two vehicles.

The vehicle detection operation along with a motion detection operation is used for queue length measurement. In this case the window is located along the road direction. The algorithm used to detect and measure queue parameters consists of two operations, one involving motion detection and the other vehicle detection. These operations are applied to a window to detect the size of the queue. As the microcomputer systems operate sequentially, a motion detection operation is firstly applied and then if the algorithm detects no motion, the 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 scenes we analysed for queue detection are expected to contain vehicles and in this case vehicle detection mostly gives the positive result, while in reality there may not be any queue at all.

The method for motion detection is based on differencing two consecutive frames and applying noise removal operators. The vehicle detection algorithm is only applied when the motion detection algorithm detects ‘no motion’. A graph for errors and processing speed against window width for queue operation is shown in figure 3. In this approach we use linear transformation to detect and measure queue from the front to the rear side of the image.

4. Results

The algorithms were tested under various traffic and lighting conditions using an Intel Pentium-based computer system operating at a clock speed of 200 MHz together with a frame grabber. The algorithm has been implemented in a user-friendly package in MS Windows environment. We have conducted extensive experiments for a continuous period of 6 - 8 hours.

Figure 4 shows the results of counting vehicles with fixed window size, while figure 5 shows the result when the program automatically adjusts the window size. As it can be seen, with this approach, the accuracy of vehicle detection has been improved. The result of queue parameter is shown in figure 6. As it can be seen, with this approach, the status of traffic can be established. Depending on the length and the speed of the queue, motorists can be informed about the traffic status and even could be redirected to alternative routes.

5. Conclusion

In this paper a novel image processing approach was described to analyse and measure road traffic
parameters. An automatic window size computation was introduced to enhance the performance of edge-based vehicle detection operation. Results demonstrated that this approach is more accurate than the fixed window size approach.

To further increase the accuracy in the case of vehicle not moving on their lanes, half size windows could be used. This approach could detect motorcycles or other small objects or people. If the window detects an object and its adjacent window doesn’t detect it, the object is considered to be motorcycle, people or any other small object. Also by taking the queue parameters into account, the traffic status could be determined and the traffic lights could be adjusted automatically.

References


Figure 2. Graph of window width versus processing speed for detecting vehicles

Figure 3. Graph of window width versus processing errors and speed for queue parameters

Figure 4. Results of Counting Vehicles (fixed Window Size)
Figure 5. Results of Counting Vehicles (automatic Window size)

Figure 6. Results of Queue Parameters