A Real Time Window Based Machine Vision System for Counting Number of Vehicles on a Street Lane

Mitra Tabaei Befrouei, Mansour Jamzad
tabaei@ce.sharif.edu, Jamzad@sharif.edu
Department of Computer Engineering
Sharif University of Technology, Tehran, Iran

Abstract

In this paper we propose a real time image processing technique for counting the number of vehicles passing along a street lane. In this method, real time speed is achieved by processing, at each frame, only the pixels inside a defined small window across a lane. For detecting vehicles, several processes are done including, enhancement, background differencing and mask based edge detection. After these processes a binary image showing the vehicle edges is created inside the small window. Extracting the number of vehicles is achieved by analyzing the number of edge pixels detected inside the small window for a specific time interval. In addition, we have used a real time image enhancement method to overcome the problem of variation in ambient light levels during different hours of day. Our method was tested in different traffic situations and ambient light levels. The results of vehicle counting were compared with that done by a human expert. Satisfactory results with accuracies over %94.8 were obtained.

Key Words

Machine Vision, Traffic Control, Vehicle Count, Window Based Method.

1. Introduction

Urban traffic control and monitoring has received special attention due to the increase of vehicles congestion and environmental pollution in large cities. Traffic information is needed for timing of traffic lights, traffic future status anticipation and planning, decreasing air pollution and fuel consumption, increasing security, and finally general public satisfaction. Various traffic parameters such as flow rate, average speed, congestion, travel time, queue length, etc. can be measured by machine vision based traffic systems. Although there are several techniques used for gathering traffic information such as magnetic inductive loops, radar, infrared and ultrasonic [1], but video image processing techniques has shown to be more efficient and flexible for computing a broad range of traffic parameters. However, there are some difficulties with machine vision methods due to the diverse circumstances that are caused by rain, fog, shades, reflection, changes of illumination and varied vehicle shapes [2].

In this paper, we present a window-based image processing method for counting the number of vehicles passing along a street lane. By processing only the pixels inside a small window in each frame, we could achieve real time processing speed (1/30 seconds/frame). For detecting vehicles in the small window, enhancement, background differencing and edge detection is performed. Then by applying a threshold a binary image showing the edge points and background was produced. Then at each frame, the number of edge pixels associated with vehicles, are stored in an array. After a time interval, these arrays are processed and the diagram generated from number of edge pixels in each frame is analyzed to extract number of vehicles.

Similar window based methods for counting the number of vehicles based on applying a constant threshold to the number of vehicle edges are given in [3,4,8]. In addition a preliminary version of this work is given in [9]. However, in this paper we have applied a real time enhancement technique based on power law transform [6] and used a new approach for edge detection [5]. The overall performance of our new approach was better than our previous work presented in [9].

Our method is implemented in visual C++ using a Matrox Meteor II frame grabber on a Pentium II 600 MHz PC and was tested by using videotapes recorded under different ambient illumination levels and traffic conditions.

2. Vehicle Detection

For detecting vehicles, we define a small window across a traffic lane such that the width of this window covers almost the full width of that lane. In each frame only the pixels inside this window are processed in order to achieve real time results.

2.1 Image Enhancement

For increasing contrast and making the edges of the vehicles sharper, the pixels of the window are enhanced with the Power-law transformation [6], that is defined as follows:

$$s = cr^{g} \qquad (1)$$

In equation (1), r is input gray level, s is output gray level, c is a constant and g is a positive constant. If g is selected to have values above '1', the contrast of the output image increases and when g is below '1', the transformed image becomes brighter. Fig.1, shows the result of enhancement achieved by Power-Law transformation with $g = 2.5$ on a sample image.

Figure 1. (a), Sample image before applying enhancement

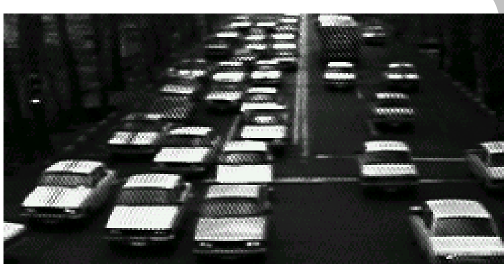

Figure 1. (b), Result of enhancement on image 1-(a) using equation (1).More real edges are extracted in this image due to it s higher contrast.

2.2 Edge Detection

For each traffic scene before beginning the processes, a background image is constructed using a voting algorithm [7]. Due to the variation in ambient light, the background image is updated concurrently by a separate procedure.

Background differencing is done inside the window for eliminating constant shades or other stationary objects.

At this stage, we used edge-detecting operators to detect all edges inside the window. It is clear that the number of edge points detected, highly depends on the type of edge detector operator used. We have tested most well known operators such as Sobel, Canny and Laplacian of Gaussian [6]. The results were compared with that of a morphological operator [3] and a gradient operator [5] that determines the edge strength in four directions of horizontal, vertical, northeast diagonal, and northwest diagonal. For simplicity, in the rest of the paper this gradient operator is called "*GR operator*". We compared the results from the morphological operator and the GR operator with each other and concluded that the *GR operator* could detect more true edges and has a better performance especially when applied on the backside of vehicles.

The morphological operator is applied on the gray scale image. It is defined by the following equation [3].

$$G = \text{Dilate (F) - Erode (F)} \qquad (2)$$

Where, F and G are input and output images. In this equation the structuring element for dilation and erosion is a 3î 3 square with zero elements. Using this structuring element, the dilate operator substitute the gray scale of a pixel with the maximum gray scale in its 3î 3 neighborhood. In arbitrary, the Erosion operator changes gray scale of a pixel with the minimum gray scale of its 3î 3 neighborhood. The above-mentioned GR operator uses four 3î 3 masks as shown in Fig.2. [5].

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

2	1	0
1	0	-1
0	-1	-2

0	1	2
-1	0	1
-2	-1	0

Figure 2. Masks for *GR operator*, determining the strength of Horizontal, Vertical, Northeast diagonal and Northwest diagonal edges.

For each pixel, four edge strengths are calculated as a weighted sum of the pixel values in its 3î 3 neighborhood with the weight coefficients given in Fig.2.

The local maximum edge strength of the pixel is defined as the maximum of the four edge strengths.

Because of the weights used in above masks, this operator can detect edges in four directions: horizontal, vertical, northeast diagonal and northwest diagonal. Figure 3 shows some sample cases in which the GR operator detected much more edges compared with that of morphological operator. Note that the edge detection algorithm is applied only on a small rectangular sub-window selected near the image center. This sub-window is seen as a black rectangle in upper left image in Fig. 3.

For dark color vehicles or the long vehicles such as buses or trucks, gradient operator detects more true edges and therefore the under count errors due to dark vehicles and over count errors due to long vehicles are decreased.

Figure 3. Left images show the edges detected with morphological operator. Right images show the result with the GR operator. Note that the edges are detected only inside the small sub-window that is visible at image center.

2.3 Constructing a binary image for window

After detecting edges, we apply a threshold for creating a binary image. This threshold is automatically determined by analyzing the histograms of some frames. As shown in figures 3 ~ 5, the edge pixels are visualized with white and non-edge pixels with black. Fig.4 shows an example of the above mentioned histogram. It represents the gray level count of pixels inside the small window.
For each histogram, two local maximums apart at some minimum distance are chosen. It is assumed that one of these maximums is for edge pixels and the other one for non-edge pixels. A minimum between these two maximums is selected as the threshold for the associated frame. By averaging the thresholds obtained from a few frames, a main threshold is determined according which the binary image is constructed.

3. Counting the Number of Vehicles

Since our system is a real time one, after starting the system, it determines the number of edges inside the window in each frame and accumulates the results in an array. After a time interval, this array is analyzed for counting the number of vehicles. The analysis method is based on a pattern that shows the behavior of the number of edges produced during the passage of a vehicle through the window. This pattern has the following general characteristics.

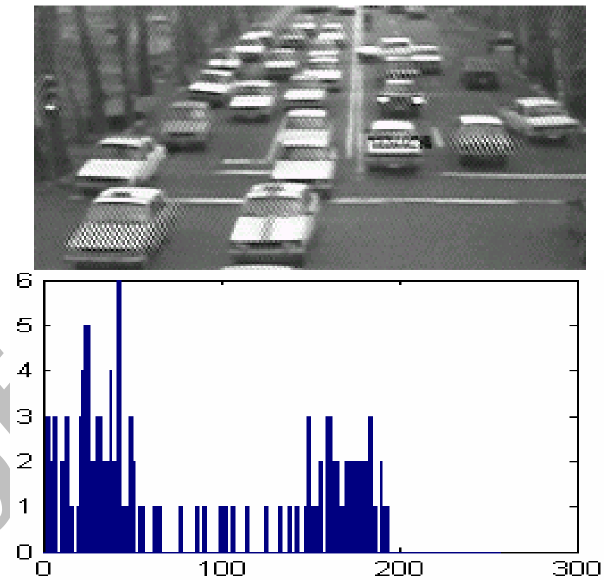

Figure 4. Above, a sample frame. Below, the associated histogram for gray scale image inside the window. The threshold selected for this histogram is 95.

It starts from a minimum when a vehicle enters into the window, reaches to a maximum during the passage, and finally when the vehicle is about to exit the window it becomes minimum again. Fig. 5 shows an example of edges produced during the passage of a vehicle through the window in 14 consecutive frames.
The number of vehicles can be determined by analyzing the increasing and decreasing pattern of number of edges during a vehicle passage through the window. In order to count a vehicle, first we find local minimums, between each two consecutive minimums, if there exists a considerable maximum, it is counted as one vehicle.
This process is visualized in Fig. 6. According to the above pattern, 7 vehicles were detected during the 200 frames examined. These vehicles are visualized as small filled circles in Fig.6. In this figure, the peaks and valleys are smooth. That is because we have applied a median smoothing filter on it in order to avoid possible miscalculation due to false local maximums and minimums.

4. Experimental Results

The accuracy of our system was measured by comparing the result of number of vehicles counted with that done a human expert. In light traffic conditions our method accuracy was %100. In addition, we have tested our method in medium and heavy traffic conditions.
In the following the results for two ambient illumination levels are presented. Vehicle counting was done in the third lane of a street crossing.

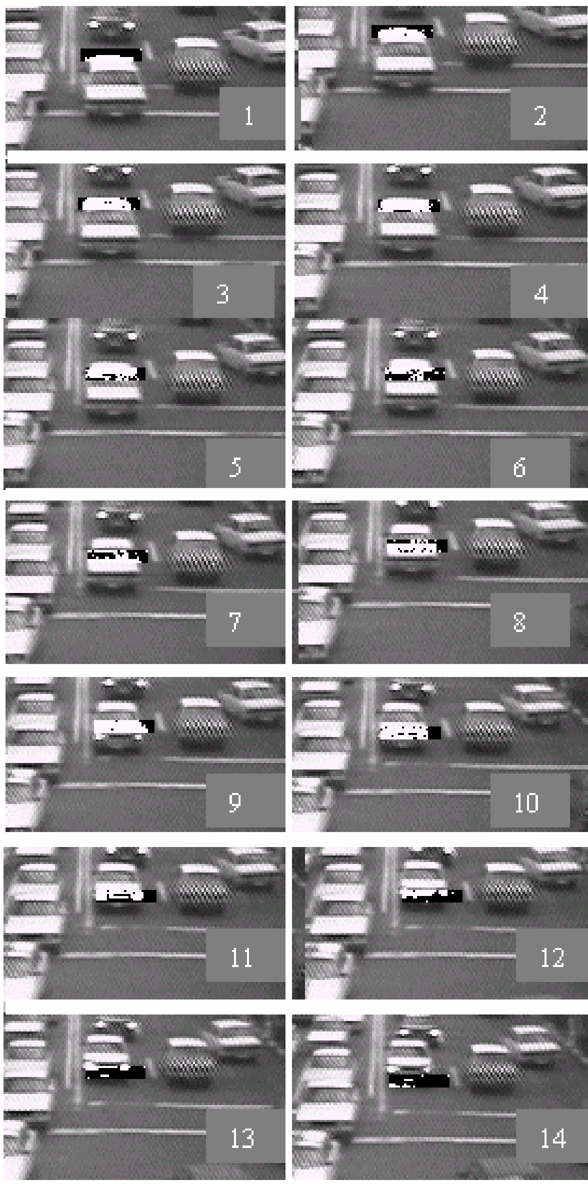

Figure 5. Edges produced during the passage of a vehicle are shown in 14 consecutive frames. The edge detection operation is done only inside the small rectangular sub-window at around the center of image.

Fig. 7 shows the result for two illumination levels, clear and dark sky where the traffic situation was changing between medium and heavy during the test. The accuracy of vehicle counting by our method was measured as, %96.7 and %94.8 for clear and dark sky, respectively.

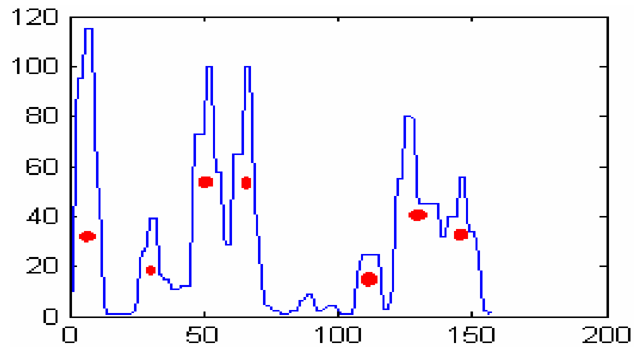

Figure 6, Diagram of the number of vehicle edges during 200 frames. Small filled circles show the time when a vehicle is detected.

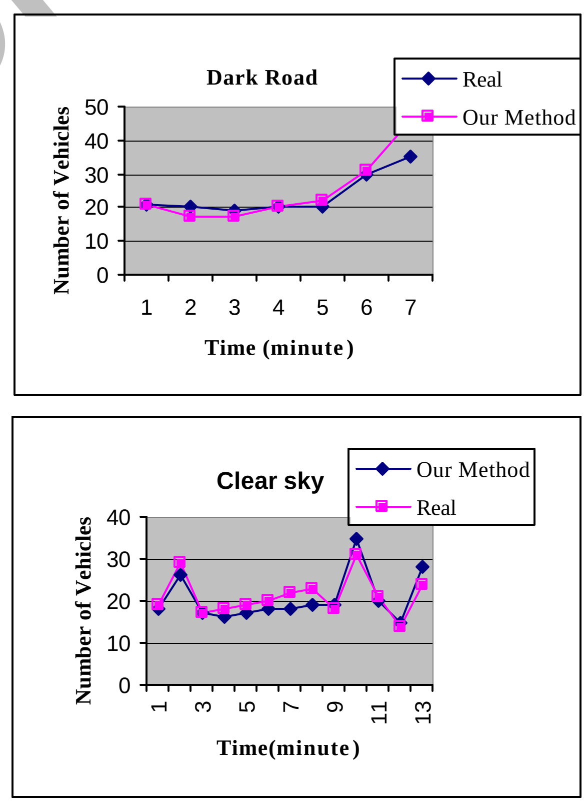

Figure 7., The accuracy of our method in counting number of vehicles compared with the results by a human expert in two different illuminations situation.

5. Conclusions

In this paper we proposed a window-based method for real time processing of video images obtained from a street to extract the number of vehicles passing through a street lane.
Similar window based methods for counting the number of vehicles based on applying a constant threshold to the number of vehicle edges are given in [3,4,8]. The main contribution of our work is its ability to perform real time pre-processing of image enhancement inside a sub-window (i.e. in order to get higher accuracy for edge detection algorithm) and using a reliable edge detector operator (i.e. the GR operator) that could detect more true edges compared to most well known edge detector operators. As another advantage, we shall mention the ability of our system in counting the number of vehicles by analyzing a diagram that represents the number of vehicle edges detected inside a sub-window during a time interval. These contributions were the main reasons why we could achieve higher accuracy in counting the number of vehicles..
Our analysis method for vehicle counting is based on the pattern of the number of edges produced during the passage of a vehicle through the sub-window. The general behavior of this pattern is as follows, starts from a minimum when a vehicle enters the window, goes to a maximum during the passage of vehicle through the window and again returns to a minimum when the vehicle exits the window.
However, we can extend our method to count vehicles in more than one lane by defining more small sub-windows in appropriate places on those lanes. In addition, we can extend our method to compute vehicles type and speed.

Acknowledgment

We would like to thank a lot the “Control Traffic Company of Tehran” for their close cooperation in providing the necessary video tapes recorded in real time situations needed for this work.

6. References

[1] Latifi M, “An Image Processing System for Measuring Traffic Parameters”, M.Sc. Thesis, Electrical Faculty, Iran University of Science and Technology, winter 1993.

[2] Darvishi A, “Traffic Quality Analysis with Image Processing”, M.Sc. Thesis, Computer Engineering Faculty, Sharif University of Technology, Summer 1995.

[3] Fathy M, Siyal M.Y, “An image detection technique based on morphological edge detection and background differencing for real time traffic analysis”, Pattern Recognition Letters 16, pp.1321-1330, 1995.

[4] Fathy M, Siyal M.Y, “A Window-Based Image Processing Technique for Quantitative and Qualitative Analysis of Road Traffic Parameters”, IEEE Transactions on Vehicular Technology, Vol. 47, NO. 4, Nov. 1998.

[5] Fan J, Aref W.G et al, "An improved automatic isotropic color edge detection technique", Pattern recognition letters, Vol. 22, pp.1419-1429, 2001.

[6] Gonzalez R.C, Woods R.E, "Digital Image Processing (2nd Edition)", Prentice Hall, 2002.

[7] Tabaei M., "Design & Implementation of a Machine Vision based System for Extracting Traffic Parameters in Highways", M.Sc. Thesis, Computer Engineering Faculty, Sharif University of Technology, Fall, 2002.

[8] Sato A, Cipriano A, “Image processing applied to real time measurement of traffic flow”, Proceedings of the 28th Southeastern Symposium on System Theory (SSST’96), IEEE, 1996.

[9] Tabaie Bafrui M, Jamzad M, “An image processing based method for measuring the Number of Vehicles”, Proceedings of 2nd Iranian conference on Machine vision and Image processing, pp. 181-188, Tehran, Iran, Feb 13-15, 2003.