Available online at www.sciencedirect.com
SCIENCE DIRECT®
Image and Vision Computing 22 (2004) 899–907
www.elsevier.com/locate/imavis

Neural-edge-based vehicle detection and traffic parameter extraction

D.M. Ha*, J.-M. Lee, Y.-D. Kim

Dept. of Electronics Engineering, Ajou University, And 5, Wonchon-Dong, Paldal-Ku, Suwon 442749, South Korea

Received 28 January 2003; received in revised form 12 December 2003; accepted 18 May 2004

Abstract

Vehicle detection is a fundamental component of image-based traffic monitoring system. In this paper, we propose a neural-edge-based vehicle detection method to improve the accuracy of vehicle detection and classification. In this method, the feature information is extracted by the seed-filling-based method and is presented to the input of neural network for vehicle detection and classification. The neural-edge-based vehicle detection method is effective and the correct rate of vehicle detection is higher than 96%, independent of environmental conditions. Also, traffic parameters, such as vehicle count, vehicle class, and vehicle speed, are extracted via vehicle tracking method

Keywords: Vehicle detection; Vehicle classification; Background extraction; Neural network; Traffic monitoring

1. Introduction

In recent years, laying new pavement or adding more lanes is becoming less and less feasible, thus that is no longer efficient solution for serious traffic congestion problem due to consistent increment of vehicles. One of the realistic solutions is to use the existing infrastructure more efficiently. Road traffic monitoring and control is the essential component of this solution. To monitor road traffic, it is necessary to extract traffic parameters that describe the characteristics of vehicles and their movement on the road. Vehicle counts, vehicle speed, vehicle path, flow rates, vehicle density, vehicle dimension, vehicle class and vehicle identity via the number plate are all example of useful traffic parameters. Various kinds of traffic control systems, for example, law enforcement, automatic tolls, congestion and incident detection, and increasing road capacity via automatic routing and variable speed limit, can be implemented with these traffic parameters.

Magnetic loop detector is most common method of traffic parameter extraction for traffic monitoring and control. Magnetic loops are very inexpensive and provide traffic parameters such as average speed, vehicle flow, and vehicle density. But, they have some problems as follows. First, they are very inflexible and modifications or additions require digging grooves in the road, thus producing traffic disturbances. Second, they cannot be used for more sophisticated tasks such as queue length measurement, tracking, etc. [8]. In recent years, an image-based traffic monitoring system is a remarkable alternative for magnetic loop detectors.

Current interest in the image-based traffic monitoring system is due to its ability to solve magnetic loop detector's problems described above. Also, image-based traffic monitoring systems offer a number of advantages. In addition to vehicle counts, a much larger set of traffic parameters, such as queue length, vehicle speed, vehicle class, vehicle path, etc. can be extracted. Besides, image-based traffic monitoring systems are much less disruptive to install than magnetic loops, thus they don't produce serious traffic disturbances [5]. In spite of these advantages, the vehicle detection error due to variation of ambient lighting condition, shadows and different shape or size of vehicles makes serious difficulties to extract traffic parameters [3,4].

In this paper, we propose a neural-edge-based vehicle detection method for improvement of vehicle detection and vehicle classification accuracy. This paper starts by describing an overview of related work in Section 2 and the object region extraction method based on background subtraction is presented in Section 3. In Sections 4 and 5, the neural-edge-based vehicle detection method and the traffic parameter extraction via vehicle tracking is described. Experimental result and conclusion are given in Sections 6 and 7.

* Tel.: +82-3-1219-2372; fax: +82-3-1212-8531.
E-mail address: chaboom@ajou.ac.kr

0262-8856/$ - see front matter
doi:10.1016/j.imavis.2004.05.006

2. Related work

Vehicle detection is a fundamental component of image-based traffic monitoring system, and has been implemented by various different approaches. For vehicle detection, the commonly used approaches are gray-level comparison, inter-frame subtraction, background subtraction, edge detection-based method, etc. In England, TULIP (Traffic analysis Using Image Processing) performed the vehicle detection operation with gray-level comparison [11]. There are some systems that achieved the vehicle detection through background subtraction, they are WADS (Wide-Area Detection System) in USA [7], TRIP (Traffic Research using Image Processing) group in England [2], CATS (Computer-Aided Traffic Sensor) in Belgium [7], and IDSC (Image Data Scanner and Controller) in Japan [6]. Several researchers attempted to detect vehicle by using the morphological edge detector and its modification, that is, SMED (Separable Morphological Edge Detector) [3,4]. Gray-level comparison utilizes statistical variation of gray-level features for road surface and vehicles, but is sensitive to environmental change. Moreover, it is almost impossible to determine the range of gray-levels of vehicles due to widely varying vehicle colors. Inter-frame subtraction takes difference between two successive frames so as to remove stationary part and get moving part within the image [13]. It is robust to environmental change, but unable to detect stationary vehicle. Moreover, the results of inter-frame subtraction are also influenced by speed of vehicles. Too low or too high speed may cause errors in vehicle detection. In the background subtraction method, vehicles are detected by taking difference between current input image and background image [9]. Background updating is needed due to the change of ambient lighting, shadow, weather, etc. Problems with background subtraction include accumulation of update error and sensitivity to ambient lighting conditions and shadows.

Edge-based vehicle detection is more effective than background subtraction techniques, as the edge information still remains significant despite the variation of ambient lighting. Various surfaces and different parts and colors of a vehicle create significant edges. Even the vehicles, which have the same color as the surface of the road, reflect more light and can be detected in this way [4]. However, edge detection-based method will fail in detecting vehicle, whose edges are unclear, especially when a vehicle with dark color is within a shadow.

3. Object region extraction

The first step in vehicle detection is to extract an object region from the current input image. In this paper, we use a background subtraction method for object region extraction. In this method, we introduce a selective background updating scheme to adapt background to the change of ambient lighting and weather conditions. Also, the initial background is extracted by an automatic background extraction. Our object region extraction method consists of three steps: background subtraction, background updating, background extraction.

3.1. Background subtraction

Background subtraction takes difference between background image and current input image. Let $\{B^t_{i,j}\}$ and $\{C^t_{i,j}\}$ be the current estimated background image and the input image, respectively, where image width is $1 \le i \le K$, image height is $1 \le j \le L$. For each pixel of the input image, calculate the difference from the estimated background image

$$D^t_{i,j} = |C^t_{i,j} - B^t_{i,j}|, \tag{1}$$

and the corresponding binary difference image can be obtained by

$$DB_{i,j} = \begin{cases} 1, & \text{if } D^t_{i,j} \ge T \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where T is a threshold and it is determined via a dynamic threshold selection scheme based on MEC [12]. $DB_{i,j} = 1$ indicates that the pixel $C^t_{i,j}$ belongs to objects, otherwise it is a pixel of background.

3.2. Background updating

In background subtraction method, the result of object region extraction strongly depends on the quality of background image. Thus, it is necessary to update the background according to the change of ambient lighting, shadow, weather, etc. However, the current input image also contains foreground objects. Therefore, before we do the update we need to classify the pixels as foreground and background and then use only the background pixels in the current input image to modify the current background. Otherwise, the background image would be polluted with the foreground objects. Hence, we introduce the selective background updating method that uses binary difference image to distinguish the foreground pixels from the background pixels

$$B^{t+1}_{i,j} = \begin{cases} kB^t_{i,j} + (1-k)C^t_{i,j}, & \text{if } DB_{i,j} = 0 \\ B^t_{i,j}, & \text{otherwise} \end{cases}, \tag{3}$$

$0 \le k \le 1$.

where the value of k determines the rate of updating. We want the updating rate to be high enough so that changes in illumination are captured quickly, but low enough so that momentary changes do not persist for an unduly long amount of time. The value of k has been empirically determined to be 0.9. We have found that this gives the best tradeoff in terms of updating rate and insensitivity to momentary changes.

3.3. Background extraction

In image sequences of road traffic, it might be impossible to acquire a background image. Hence, it is desirable that the initial background image is automatically extracted from a sequence of road traffic images. Therefore, we propose an automatic background extraction method and this method is described bellow. And also, the detailed procedure of background extraction is represented with the intermediate example images in Fig. 1. The image used in this paper is

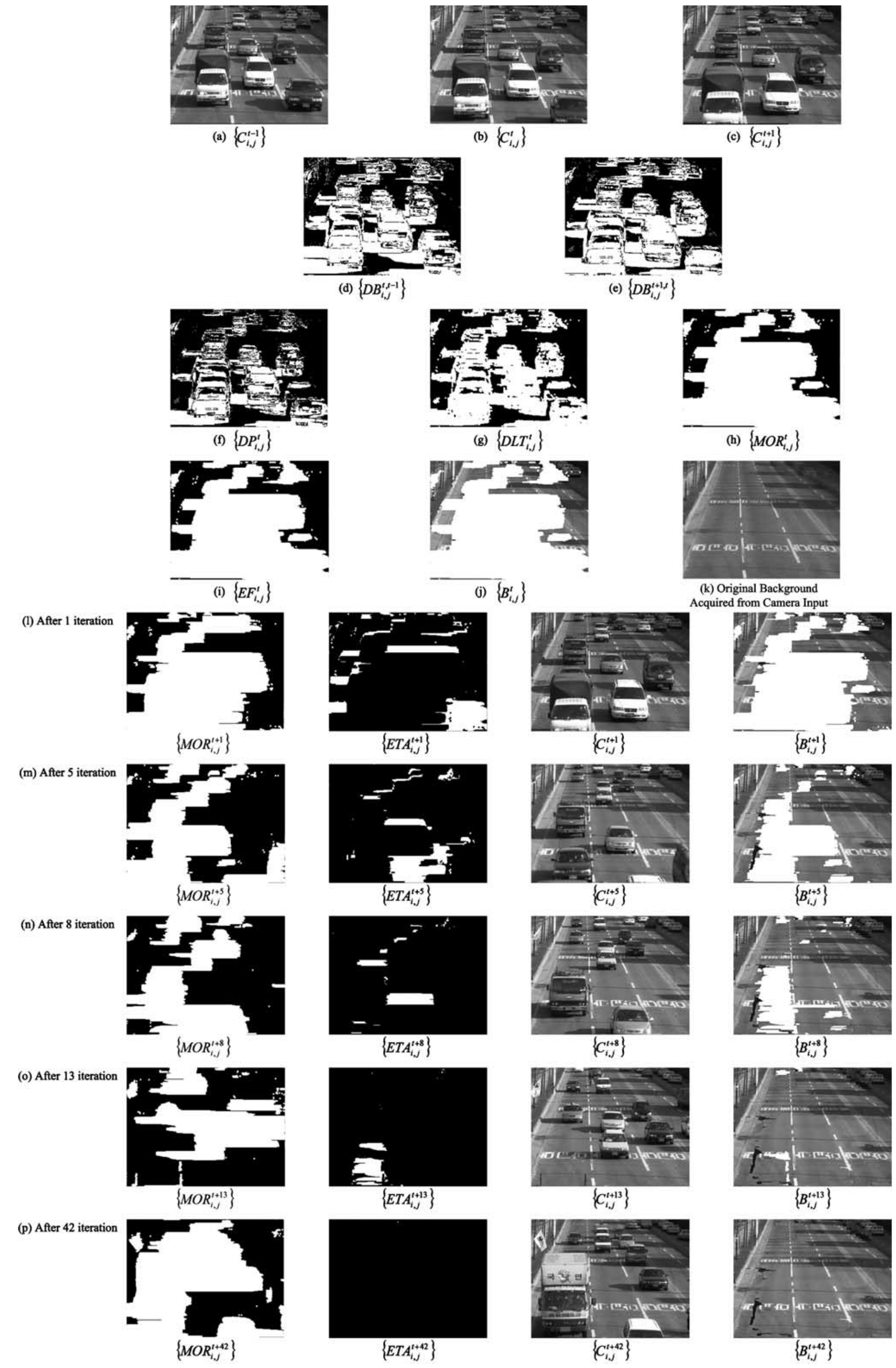


Fig. 1. The sequential procedure of background extraction.

the 8-bit grayscale image.

(1) Calculate MOR (Moving Object Region) for current input image. To extract MOR, we first calculate inter-frame subtraction of two pairs of images (frame t and frame $t-1$), and (frame t and frame $t+1$). Then, calculate DP (Difference Product) by the bitwise logical AND operation of two inter-frame subtraction images. Finally, we apply binary dilation and shape filtering (based on seed-filling) to DP image, to widen and clarify the MOR. The procedure described above, can be given the following expressions

$$DB_{i,j}^{t,t-1} = \begin{cases} 255, & \text{if } D_{i,j}^{t,t-1} \geq T \\ 0, & \text{otherwise} \end{cases}, \quad DB_{i,j}^{t+1,t} = \begin{cases} 255, & \text{if } D_{i,j}^{t+1,t} \geq T \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where

$$D_{i,j}^{t,t-1} = |C_{i,j}^{t} - C_{i,j}^{t-1}| \text{ and } D_{i,j}^{t+1,t} = |C_{i,j}^{t+1} - C_{i,j}^{t}|$$

$$DP_{i,j}^{t} = DB_{i,j}^{t-1,t} \ \& \ DB_{i,j}^{t+1,t} \tag{4.2}$$

where '&' is the bitwise logical AND operator

$$\mathrm{MOR}_{i,j}^{t} = \text{Shape filtered image of } \mathrm{DLT}_{i,j}^{t} \tag{4.3}$$

where $\mathrm{DLT}_{i,j}^{t} =$ Binary dilation of $DP_{i,j}^{t}$

(2) By using the MOR information, estimate initial background region from current input image and store this region information at the EF (Extraction Flag). This procedure can be described as following equations

$$B_{i,j}^{t} = MOR_{i,j}^{t} | C_{i,j}^{t} \tag{5.1}$$

where the symbol '|' is the bitwise logical OR operator

$$\mathrm{EF}_{i,j}^{t} = \mathrm{MOR}_{i,j}^{t} \tag{5.2}$$

(3) Calculate MOR for current input image and calculate background extraction target area (ETA)

$$\mathrm{ETA}_{i,j}^{t} = \mathrm{EF}_{i,j}^{t-1} \ \& \ \overline{\mathrm{MOR}_{i,j}^{t}} \tag{6}$$

where $\overline{\mathrm{MOR}_{i,j}^{t}} = 1'$s complement of $\mathrm{MOR}_{i,j}^{t}$

(4) Extract background pixels in the current input image and update EF

$$B_{i,j}^{t} = B_{i,j}^{t-1} \ \& \ (C_{i,j}^{t} | \overline{\mathrm{ETA}_{i,j}^{t}}) \tag{7.1}$$

$$\mathrm{EF}_{i,j}^{t} = \mathrm{EF}_{i,j}^{t-1} \oplus \mathrm{ETA}_{i,j}^{t} \tag{7.2}$$

where the symbol '$\oplus$' is the bitwise logical XOR (Exclusive OR) operator.

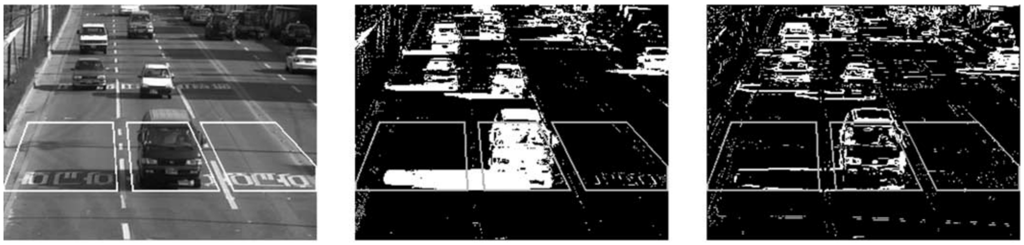
(5) Repeat steps 3 and 4, until the percentage of extracted background pixels in EF reaches to a pre-defined value (in this paper, 99 (%) is used).

4. Neural-edge-based vehicle detection

The object region extraction is the basic step of vehicle detection as previously described. An example of object region extraction result that is used in this paper is given in Fig. 2. In the figure, there is a big shadow of the first vehicle projected on the left lane and the shadow is so big that it may be mistaken as a vehicle. The active shadow due to moving vehicle produces non-vehicle region, and it ultimately makes vehicle detection error. Therefore, we propose a neural-edge-based vehicle detection method to remove vehicle detection error resulted from non-vehicle region. And also, the vehicle class can be recognized with the neural-edge-based vehicle detection method. The neural-edge-based vehicle detection method consists of three steps: edge detection, feature information extraction, and vehicle detection.

4.1. Edge detection

The background subtraction-based object region extraction method suffers from big problems caused by non-vehicle region due to active shadow, as described above. On the contrary, the edge pixels of the shadow are significantly less than that of the vehicle, as shown in Fig. 2(c). Hence, the edge information of extracted object region can be used to distinguish non-vehicle region from vehicle region and should be useful for non-vehicle region and vehicle detection error rejection. But, an edge image of the current input frame contains not only the edge of the desired object but also that of the background details.



(a) Input image **(b) Object region extraction** **(c) Edge image**

Fig. 2. An example of object region extraction and edge detection result.

Thus, we introduce a moving edge detection method to extract only the edge of the desired object and describe the method below.

Calculate edge images of current input image and background image, respectively. Sobel operator

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

is used for edge detection, which is represented as

$$d_{i,j} = \max\left\{\frac{1}{4}\sum_{y=j-1}^{j+1} w_y |I_{i-1,y} - I_{i+1,y}|, \frac{1}{4}\sum_{x=i-1}^{i+1} w_x |I_{x,j+1} - I_{x,j-1}|\right\}, \quad (8)$$

where I is current input image $\{C_{i,j}^t\}$ or background image $\{B_{i,j}^t\}$. The corresponding binary edge image is obtained by

$$\mathrm{Edge}I_{i,j} = \begin{cases} 1, \text{ if } d_{i,j} \geq T_\mathrm{E} \\ 0, \text{ otherwise} \end{cases}, \quad T_E = \alpha + \log_{10} d_{i,j}, \quad (9)$$

where α is a constant as the lowest limit of the threshold. Then obtain the moving edge image $\{E_{i,j}\}$ by subtracting edge image of $\{B_{i,j}^t\}$ from edge image of $\{C_{i,j}^t\}$.

4.2. Feature information extraction

When the classification of vehicle is performed via human visual system, the typical standard is the feature information of vehicle, such as shape and size. Likewise, we can divide the extracted object region into the vehicle and the non-vehicle region by using the feature information of it. In this paper, we used a binary seed-filling [1] based method to extract the feature information of the connected object region obtained by object region extraction. The total number of white pixels that consists of the connected object region and the number of edge pixels in that region are calculated by this method. In the method, the object region is modeled as a rectangle and its center coordinate, average length and width are also calculated.

4.3. Vehicle detection

To detect and classify vehicle, we present the data obtained by the feature information extraction method (that is, the number of white pixels and number of edge pixels in the object region, the average width and average length of that region) to the input of neural network. In this paper, we used three same feed forward neural networks, one network per each vehicle class, to classify vehicles into three categories. The configuration of each network is given below.

- Number of network layers: 3
- Number of input neurons: 48
- Number of hidden layer neurons: 8
- Number of output layer neurons: 1

The network detects vehicle and classifies it into three categories, that is, small, medium, and big. In the feed forward network, the input vector for the network is incident on the input layer and distributed to subsequent hidden layers and finally to the output layer via weighted connections. Each neuron in the network operates by taking the sum of its weighted inputs and passing the result through a non-linear activation function. This is shown mathematically as:

$$\mathrm{out}_i = f(\mathrm{net}_j) = f\left(\sum_j W_{ij}\mathrm{out}_j + \theta_i\right) \quad (10)$$

Here out_i is the output of the ith neuron in the layer under consideration; out_j is the output of the jth neuron in the preceding layer. There are several conventionally used choices for the non-linear function, f. In this paper, the sigmoid function is used as given below equation:

$$f(\mathrm{net}_i) = \frac{1}{1 + \mathrm{e}^{(-\mathrm{net}_i/Q_0)}} \quad (11)$$

The term Q_0 in above equation is referred to as the temperature of the neuron. The higher the temperature the more gently the sigmoid changes [10].

The network is trained with back propagation [10] and 3000 samples that collected under various weather and lighting condition are used as a training set. The training involves three stages:

(1) The feed forward of input training vectors.
(2) The back propagation of the associative error.
(3) The adjustments of the weights.

The flowchart of the training algorithm is shown in Fig. 3.

5. Traffic parameter extraction via vehicle tracking

In this paper, traffic parameters, such as vehicle count, vehicle speed, and vehicle class, are extracted via vehicle detection and tracking method. The vehicle object and its class are recognized via the neural-edge-based vehicle detection method that proposed in Section 4. The individual vehicle is tracked within a tracking area on each lane. The vehicle tracking area has a trapezoidal shape and an entrance zone and an exit zone are located at the top and bottom of the area as shown in Fig. 2. When a vehicle is detected in the entrance zone, it is added to the linked-list that includes the information about it, such as its position, class, and frame count. The information and linked-list are updated as the vehicle object moving within the tracking area. When a vehicle is detected in the exit zone, the vehicle count on each lane is calculated and the vehicle class is

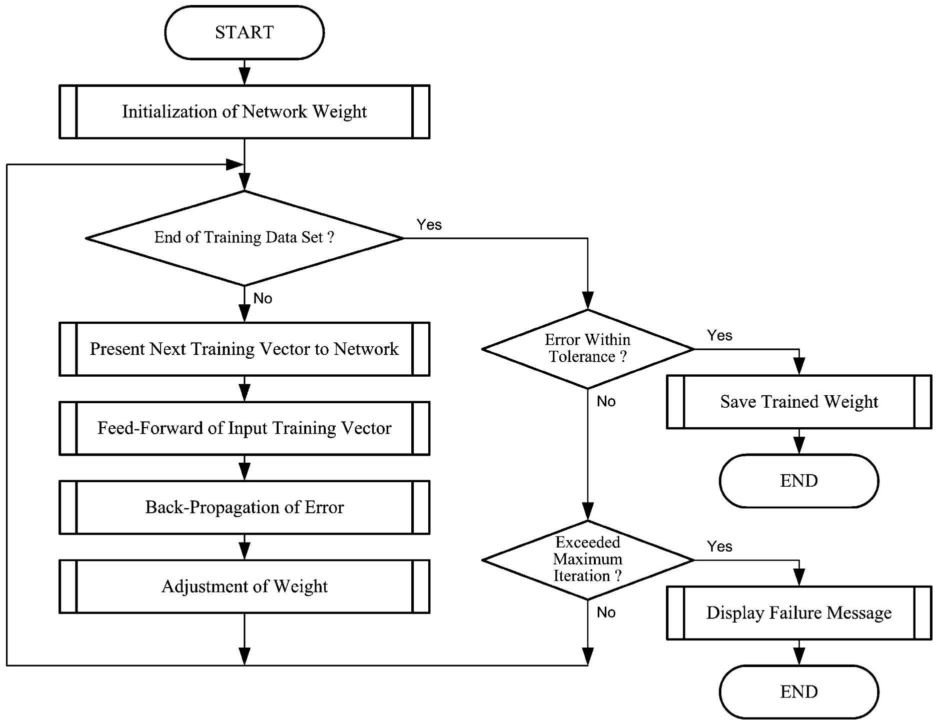


Fig. 3. Back propagation flow chart.

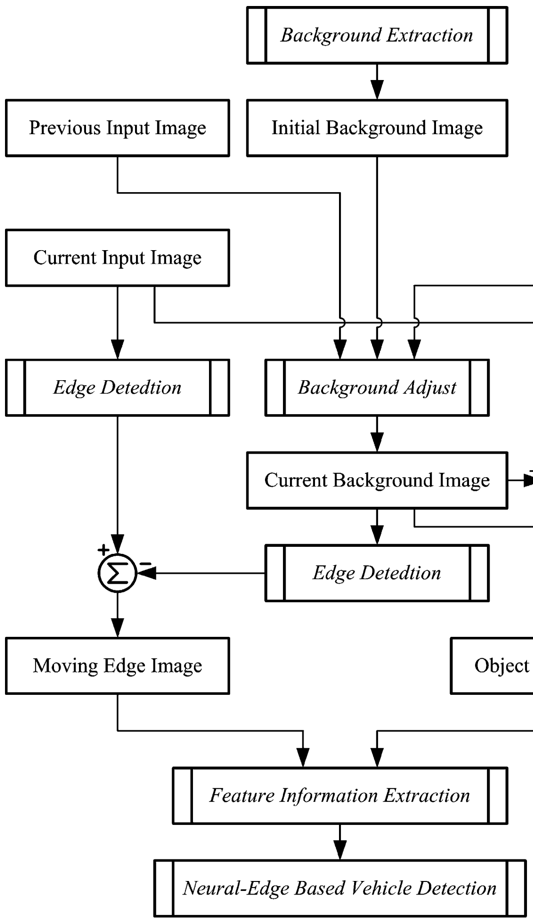


Fig. 4. Block-diagram for neural-edge-based vehicle detection method.

Table 1
Results of vehicle detection

Methods	Lanes								
	Left lane			Middle lane			Right lane		
	Missed detection rate (%)	False detection rate (%)	Correct rate (%)	Missed detection rate (%)	False detection rate (%)	Correct rate (%)	Missed detection rate (%)	False detection rate (%)	Correct rate (%)
Background subtraction	3.64	2.03	94.33	5.26	1.59	93.15	3.26	1.12	95.62
Edge detection	2.42	0.89	96.69	3.16	1.21	95.63	3.47	1.42	95.11
Neural-edge	0.40	0.13	99.47	1.02	0.33	98.65	0.64	0.60	98.76

determined by using the information that accumulated in the linked-list. And the vehicle speed is computed by the equation as given below

$$\text{Speed (km/s)} = \frac{D \times F}{F_t} \tag{12}$$

where D is the physical distance between the entrance zone and exit zone of the tracking area, F and F_t are frame rate and number of frames taken for the vehicle to pass the tracking area, respectively. Usually high frame rate results in more accurate speed estimation.

6. Experimental result

6.1. Experimental environment

The block diagram for the neural-edge-based vehicle detection and traffic parameter extraction method proposed in this paper is presented in Fig. 4. The analog image output from the video camera that mounted over the road is converted to digital data by the frame grabber embedded in a personal computer system and the digitized traffic image is used for algorithm processing. In this paper, the computer system and software configuration given below is used to implement and evaluate the performance of the proposed neural-edge-based vehicle detection and traffic parameter extraction method. The time requires to process one frame (width: 320, height: 240, color: 8-bit grayscale) is from 47 to 62 ms, according to the processing speed of seed-filling operation.

- Main Computer System
  - CPU: Pentium-III 700 MHz, Main Memory: SDRAM 64 MB
- Digital Image Acquisition Device: DT3155 PCI Frame Grabber
- Operating System: MS Windows 2000 Professional Edition
- S/W Development Tool: MS Visual C++ 6.0

6.2. Experimental result

In this paper, we proposed the neural-edged-based vehicle detection method and traffic parameter extraction method via vehicle tracking. To evaluate the performance of those methods, we use more than 60 h test video that collected on various road and under various weather and lighting condition.

6.2.1. Vehicle detection result

The vehicle detection results of the proposed neural-edge-based method and other methods (that is, background subtraction and edge detection method) are given in Table 1. In the table, 'missed detection' means that there was a vehicle present in the tracking area, but the algorithm failed to detect it, while 'false detection' corresponds to the opposite condition. In background subtraction method, missed detection mainly comes from vehicles whose luminance is similar to the road surface and false detection is mainly caused by active shadows resulted from vehicles passing through the neighbor lane. On the other hand, in edge detection method, missed detection comes from vehicles whose edge is not clear enough and false detection

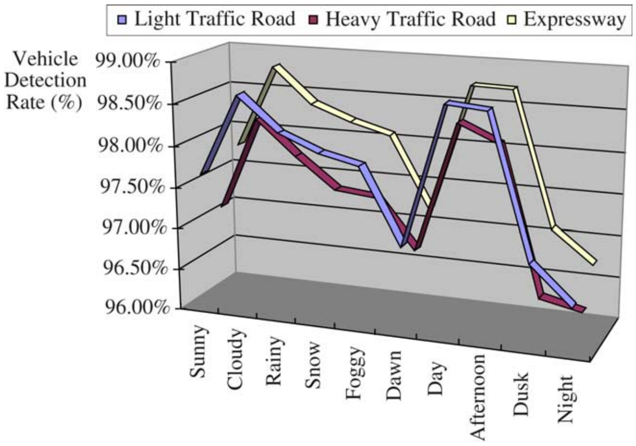


Fig. 5. The result of vehicle detection under various environmental conditions.

is mainly caused by the passive shadows resulted from roadside buildings and trees and the land marks on road. Generally, they make enough edge pixels to be detected as a vehicle.

In the experiments, compared with background subtraction and edge detection method, the neural-edge-based method achieves better results, since it takes advantage of the two methods. In this method, we extract the feature information of object region with the seed-filling-based method that combines background subtraction and moving edge detection result and then present the feature information to the input of feed-forward neural network. By using the feature information and neural network, the accuracy of vehicle detection and vehicle classification is improved. In the neural-edge-based vehicle detection method, most of vehicle detection errors are due to the behavior of drivers where they change lanes very frequently and other factors such as occlusion.

The vehicle detection results of the neural-edge-based method that applied to the test video collected on various roads and under various weather and lighting condition are given in Fig. 5. With the proposed method, the vehicle

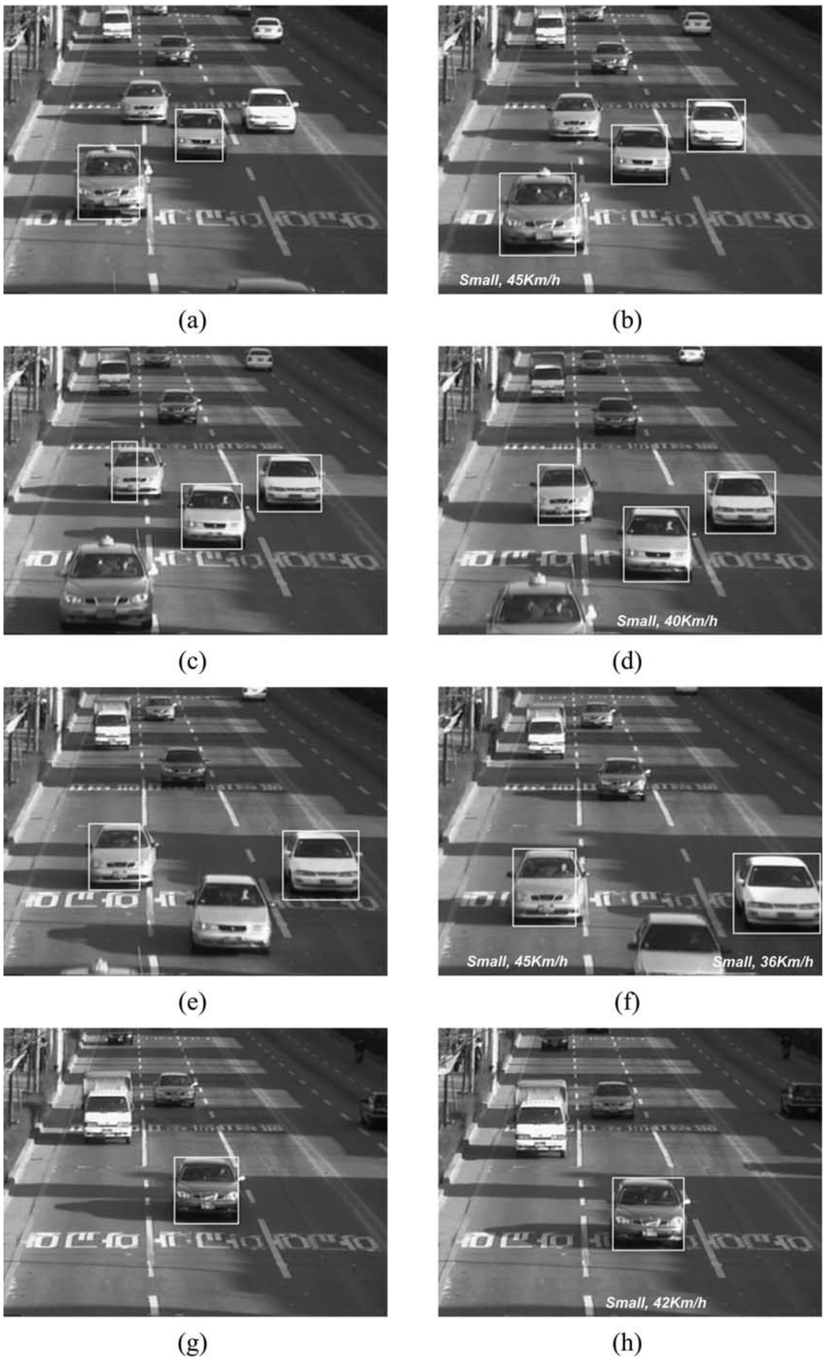


Fig. 6. An example of traffic parameter extraction.

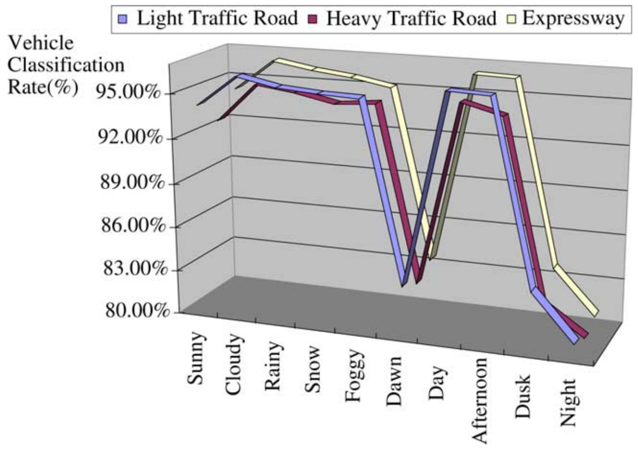


Fig. 7. The result of vehicle classification under various environmental conditions.

detection rate achieved is higher than 96%, independent of environmental conditions. The vehicle detection rate of sunny day is lower than that of others (that is, cloudy, rainy, snow, and foggy day), because the shadow effect of sunny day is more serious for vehicle detection than others. Also, the vehicle detection rate becomes lower at dawn, dusk, and night, because the edge information of vehicle reduces and the outline of vehicle is unclear due to absence of sunlight. But, there are still enough edge and outline information to detect vehicle that created by the reflection of headlight.

6.2.2. Traffic parameter extraction result

In this work, traffic parameters, such as vehicle counts, vehicle class, and vehicle speed, are extracted by tracking the vehicle detected via neural-edge-based method. These traffic parameters are essential component in traffic monitoring and control system to efficiently use existing infrastructures, as indicated previously. Fig. 6 gives an example of vehicle detection and traffic parameter extraction by means of the proposed method. The results of vehicle classification are given in Fig. 7, where the vehicle classification rate is defined as the ratio between computer and manual classification result. The experiments show that the method based on vehicle tracking works well under light traffic condition. In contrast, under heavy traffic condition, the method may mistake a vehicle as a bigger one in vehicle classification when the vehicle is occluded with other following vehicle in input images. It is clear that high location of camera will reduce such kind of mistake. However, the vehicle that changes lane within the tracking area may cause trouble because it crosses two lanes. The vehicle classification ratio becomes lower at dawn, dusk, and night, because the edge and outline of vehicle are often unclear. This problem can be solved by the artificial roadside illumination that makes the edge and outline of vehicle clear. But, this solution requires expensive installation of roadside streetlamp, thus cannot be applied easily. We will try to solve this problem in the future work.

7. Conclusion

In this paper, we propose the neural-edge-based vehicle detection and traffic parameter extraction method. In the proposed vehicle detection method, the feature information is extracted by using the seed-filling-based feature extraction method that combines background subtraction and moving edge detection result. And then, we improved the accuracy of vehicle detection and classification by using this feature information as an input of neural network. The method is effective and the correct rate of vehicle detection is higher than 96% independent of environmental conditions. Traffic parameters, such as vehicle counts, vehicle class, and vehicle speed, are extracted by tracking the vehicle detected via neural-edge-based vehicle detection method. These traffic parameters are essential component in traffic monitoring and control system to efficiently use existing infrastructures.

References

[1] P. Burger, D. Gillies, Interactive computer graphics: functional, procedural and device-level methods, Addison-Wesley, Reading, MA, 1989.
[2] K.W. Dickinson, R.C. Waterfall, Image processing applied to traffic: a general review, Traffic Eng. Contr. 1 (1984) 6–13.
[3] M. Fathy, M.Y. Siyal, Real-time image processing approach to measure traffic queue parameters, IEE Proc.—Vis. Image Signal Process. 142 (5) (1995) 297–303.
[4] M. Fathy, M.Y. Siyal, A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters, IEEE Trans. Vehicular Technol. 47 (4) (1998) 1342–1349.
[5] S. Gupte, O. Masoud, Detection and classification of vehicles, IEEE Trans. Intell. Transport. Syst. 3 (1) (2002) 37–47.
[6] N. Hashimoto, Development of an image processing traffic flow measurement system, Sumitomo Electron. Tech. Rev. 25 (1988) 133–138.
[7] N. Hoose, Computer Image Processing in Traffic Engineering, Taunton Research Studies Press, UK, 1991.
[8] R.M. Indio, Application of machine vision to traffic monitoring and control, IEEE Trans. Vehicular Technol. 38 (3) (1989) 112–122.
[9] P.G. Michalopoulos, Vehicle detection video through image processing: the autoscope system, IEEE Trans. Vehicular Technol. 40 (1) (1991) 21–29.
[10] A.S. Pandya, R.B. Macy, Pattern Recognition with Neural Networks in C++, IEEE Press, 1995, p. 73–146.
[11] A. Rourke, M.G.H. Bell, Traffic analysis using low cost image processing, in: Proceedings of the Seminar on Transportation Planning Methods, PTRC, Bath, UK, 1988.
[12] J.-C. Yen, F.-J. Chang, S. Chang, A new criterion for automatic multilevel thresholding, IEEE Trans. Image Process. 4 (3) (1995) 370–378.
[13] H. Zhang, Y. Gong, D. Patterson, A. Kankanhalli, Moving object detection, tracking and recognition, The Third International Conference on Automation, Robotics and Computer Vision, 1994, p. 1990–1994.