Weather-adapted Vehicle Detection for Forward Collision Warning System

En-Fong Chou and Din-Chang Tseng

***Abstract* – A preceding vehicle detection method for forward collision warning system is proposed. The proposed method utilizes the horizontal and vertical edges instead of underneath shadow to detect preceding vehicles without being influenced by variant weather conditions. We first extract horizontal edges and then check the vertical edges above horizontal edges to confirm the location of a horizontal edge just beside an object and generate a candidate vehicle with vertical borders while horizontal edge is located at the bottom of the object. If the bottom of the object can't be found by a horizontal edge, we find the vertical borders and bottom of the object by searching symmetric vertical edge pair. Then, we estimate the width of the object to select candidate vehicles. At last, we use *SVM* to verify the candidate vehicles. In experiments, the proposed method was evaluated on variant weather conditions such as sunny day, cloudy day, and rainy day. The detection rate of preceding detection is over 90% in sunny day and cloudy day and is over 80% in rainy day.**

***Index Terms* - Advanced safety vehicle, forward collision warning, vehicle detection, computer vision**

I. Introduction

Vehicle safety becomes more important due to the popularization of vehicles in recent years. Every minute, on average, at least one person dies in traffic accident in the world. To reduce the risk of causing fatal traffic accidents, many companies integrate research and hardware equipments to develop a certain mechanism for adding more safety to vehicle. Many related researches about vehicle safety including lane detection, vehicle detection, blinding spot detection and traffic signs detection. All those researches have trying to use camera or sensor to obtain the information content in the perception of the surrounding scenario, of which we can analysis by computer vision to warn the driver.

Robust vehicle detection with camera is a big challenge. Many studies have been proposed to solve the vision-based vehicle detection. Sun *et al.* [1] divided various vehicle detection methods into three categories generally. The three categories are knowledge-based, stereo-based and motion-based. Knowledge-based vehicle detection [2-4] integrates many features of vehicle such as shadow, horizontal edges, vertical edges, symmetry, corner, color and texture to locate the vehicle in image. We will introduce some of them as follows

Betke *et al*. [3] presented two methods to recognize the passing cars and distant cars. In the first method, they used difference of brightness between current frame and earlier frame to detect the vehicle and recognized by tracking. In the second method, they combined horizontal and vertical edge maps to detect the vehicle. First, they searched prominent edges in the thresholded edge map to find the potential region. Then, they find the horizontal map projection value along column for each row and the vertical map projection value along row for each column to determined threshold values T_v and T_h. They searched from both left and right to centre until the projection value lies above T_v in both cases and, similarly, from both top and bottom to centre until the projection value lied above T_h in both methods to generate detected box. Then they computed the aspect ratio to check if it is closed to 1. At last, they used template to verify the detected box and output the result.

Jin *et al*. [4] proposed a method preceding vehicle detection based on multi-characteristics fusion. At first, they use the bottom shadow of vehicle to establish the areas of interesting (*AOI*) for preceding vehicles. Then, they extract three characteristics: texture, edge and symmetry of vertical edges in *AOI*. The preceding vehicles are indentified by fusion of these three characteristics. At last, they find the location of vehicle by vertical and horizontal edge projection in the *AOI*.

Wang and Lien [5] proposed a novel statistical approach to detect vehicles based on three significant sub-regions of the vehicle. The system includes a training process and a testing process. At first, they normalize the geometrical positions of vehicle image which is extracted from training database to create canonical vehicle image. Then, they normalize the light conditions of resulting canonical vehicle images and extract three sub-regions: roof, left, and right taillight (or headlight). To reduce the computation, they apply *ICA* and *PCA* to transform the sub-region images to classify vehicle and non-vehicle classes.

In order to enhance the correctness of the vehicle detection, many researchers used difference methods to verify the target such as entropy of the target region, symmetry property, and template matching. Classifier is popular in pattern recognition and many researches used it to make verification robust. There are many classifiers such as neural network, the *AdaBoost* algorithm and the support vector machine (*SVM*). The classifier is selected based on the requirement of system.

SVM can easily discriminate the object and non-object by training in advance, many researches utilized *SVM* for classification. Sun *et al.* [6, 7] extracted different features such as *Haar* wavelet and *Gabor* filter to feed the *SVM* and

Manuscript received Feb.26, 2011; revised Mar.22, 2011. This work is partially funded by the National Science Council, Taiwan under the grant of the research project *NSC* 97-2221-*E*-008 -073 -*MY*3.

En-Fong Chou is with the Institute of Computer Science and Information Engineering, National Central University, Chungli 32001, Taiwan (e-mail: kobe31200@gmail.com).

Din-Chang Tseng is with the Institute of Computer Science and Information Engineering, National Central University, Chungli 32001, Taiwan (corresponding author Tel: -886-3-4227151; e-mail: tsengdc@ip.csie.ncu.edu.tw).

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

verify the vehicle. With *SVM,* they have high accuracy in vehicle detection. Instead of using detecting whole vehicle to classify, many researches propose a component-based method to detect significant parts of vehicle. Leung [8] proposed a method to detect the significant parts such as wheel, roof and back bumper in entire image by using *AdaBoost* classifier. Then, they combine the detected parts of vehicle and classify them by using *SVM*.

Vehicle detection is easily influenced by weather conditions such as rainy day, foggy day, night, and shadow. The information of surrounding scenario in these weather conditions is quite different with normal day. The proposed scheme integrates vehicle features to detect the preceding vehicle under variant weather conditions. The proposed scheme detects preceding vehicle based on the information of lane mark. The scheme contains two major parts: vehicle detection and vehicle verification to detect the candidate vehicle and verify it is vehicle. We briefly describe these two major parts of scheme as follows:

(*i*) Vehicle detection: To detect preceding vehicle robustly under variant weather conditions, the features have to adapt to the weather. The proposed procedure uses horizontal and vertical edge to detect vehicles. At first, the proposed scheme detects lane to obtain the information of lane mark which helps us to generate a searching region and delete the redundant horizontal edges. We also propose a method to determine a searching region when result of lane detection is wrong. To reduce the complexity of problem, the proposed procedure defines a dynamic threshold to obtain bi-level horizontal and vertical edges. The proposed scheme chooses a horizontal edge and finds the vertical edge around two ends of it to generate a candidate vehicle. After that, the proposed verification procedure is used for confirming the candidate vehicle is a vehicle.

(*ii*) Vehicle verification: The proposed verification uses Support Vector Machine (*SVM*) to verify candidate vehicles. The *SVM* model is trained by many vehicle and non-vehicle images which size are of 32×32.

The remaining sections are organized as follows. Section 2 presents the proposed approaches on the vehicle detection. Section 3 presents the methods of vehicle verification. Section 4 reports the experiments and the results. Section 5 is the conclusions.

II. Vehicle Detection

Cameras have the advantages of cheap and versatile functions over radar devices to detect vehicles; however, the vision detection is easily influenced by variant weather conditions. We here use multiple edges which are the most robust features to detect vehicles. We use gradient operators to extract horizontal and vertical edges. The mutuality of the horizontal and vertical edges is used to detect vehicle.

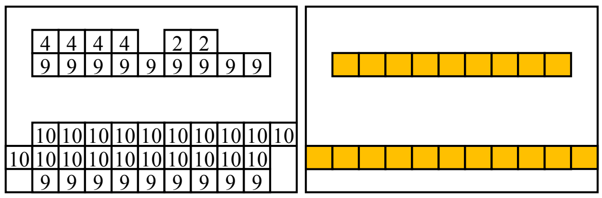
A. Generation of bi-level gradient images

To simplify the vehicle detection scheme, the proposed procedure utilizes bi-level edges which are acquired by generating bi-level gradient images to detect vehicles. To generate bi-level gradient images, the thresholding procedure is necessary. The optimal threshold should be determined depend on different weather conditions. Since the scheme only utilize a video camera to acquire information, it is difficult to differentiae from different weather conditions. Therefore, the dynamic thresholding procedure is proposed.

The procedure analyzes the distribution of edge responses and tries to retain edge responses which belong to vehicles. Since vehicles appear on the road generally, the procedure only analyzes the distribution of edge responses in *ROI*. In general, edge responses which belong to vehicles are higher in *ROI*. With the condition, the procedure reserves the edge points which have higher edge response. By observation, the positive and negative horizontal edge points whose response is in top 20% are reserved. The vertical edge points whose response is in top 10% are reserved, too. With thresholding procedure, the bi-level gradient images are generated.

B. Extraction of significant horizontal edges

Since bi-level gradient images have no geometric meanings, the proposed procedure analyzes them and extracts geometric information. By searching in whole image from bottom to up, left to right, the edges whose length is longer than a pre-defined threshold T_e are reserved. The purpose of defining threshold T_e is to delete the edge which is small and could be the noise. After deleting petty edges, the reserved edges are used for generating significant edges. If reserved edges are adjacent in *y*-order direction, the proposed procedure generates a significant edge whose ends is the most left and right ends of these reserved edges. Then, *y*-position of the significant edge is same as the *y*-position of the longest edge in these reserved edges. Otherwise, the significant edge is same as the reserved edge. The example of extracting significant horizontal edge is shown in Fig. 1, where a gird means a pixel and the number in grid is the length of edge.



(a) (b)

Fig. 1. The example of extraction significant edge when T_e = 9. (a) The bi-level edge points. (b) The result of extracting significant edges.

C. Deletion of short horizontal edges

Not all significant edges are used for detect vehicles. Only the edges whose length almost equal to the width of vehicle are acquired; however, the width of vehicles is various. We here define 1.5 meter to be the shortest width of a vehicle. Any horizontal edges whose length is shorter than the shortest width will be deleted. The actual length of a horizontal edge can be estimated from edge length in the image by the method proposed by Tseng *et al.* [9, 10] or simply comparing the edge length with the same-location known lane width as an example shown in Fig. 2.

D. Generation of candidate vehicles

To generate candidate vehicles, the proposed method is divided into two parts: (*i*) checking vertical edge patency

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and searching the left and right border where the negative horizontal edge is at the bottom of vehicle and (*ii*) finding symmetrical vertical edge pair around the horizontal edge.

Fig. 2. The estimation of horizontal edge length.

a. Checking vertical edge patency above horizontal edge

The negative horizontal edge at the bottom of vehicle has similar characteristic with the underneath shadow which is the widely used feature of vehicle. However, the underneath shadow can not be detected under rainy or foggy day that will cause vehicle detection failed. In contrast with the underneath shadow, the negative horizontal edge can be detected under various weather conditions. Since the proposed vehicle detection is weather-adapted, it utilizes negative horizontal edge to substitute the shadow.

When the horizontal edge is at the bottom of vehicle, it satisfies two constraints: it is on the road and the strength of vertical edge above horizontal edge is much larger than below. With the first constraint, the negative horizontal edges which are inside the lane mark are selected. Then, with the second constraint, the proposed procedure defines a searching region on each ends of edge based on the width of negative horizontal edge and finds the largest accumulation of vertical edge below and above the horizontal edge as shown in Fig. 3.

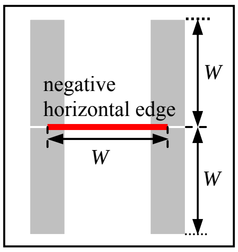


Fig. 3. The diagram of searching region of finding the largest vertical edge accumulation.

The procedure calculates the accumulation of vertical edge per column in searching region and searches the largest accumulation. Let S_{LU}, S_{RU}, S_{LD} and S_{RD} be the largest accumulation in searching region W above and below the left and right end of edge, and W be the width of negative horizontal edge. If these values satisfy a criterion which defined as

$$(S_{LU} \geq 0.5W) \cap (S_{RU} \geq 0.5W) \cap (S_{LD} \leq 0.2W) \cap (S_{RD} \leq 0.2W), \quad (1)$$

the negative horizontal edge is at the bottom of the vehicle. Moreover, the left and right vertical borders of the vehicle are determined as the column where S_{LU} and S_{RU} happened. Since the distance between two vertical borders and the width of vehicle are the same, it must larger than 1.5 meters. Therefore, the procedure estimates the actual distance W_{actual} between two vertical borders to generate a candidate when distance longer than 1.5 meters.

However, the detection may be wrong when the underneath shadow is too long as shown in Fig. 4. To solve this problem, we propose a method to re-search the new border base on the actual width of candidate. In general, the width of vehicles is less than 3.0 meters. The procedure finds the candidate whose width is longer than 3.0 meters and re-searches the vertical border of it.

Fig. 4. The example of long underneath shadow.

Since the negative horizontal edge is detected and located on the shadow, there is at least one vertical border will fall on the end of horizontal edge. With the restriction, the procedure can be divided into several steps:

i. Selecting detected left vertical border and search the new right vertical border in the region as shown in Fig. 5.
ii. Calculating the accumulation S_U, S_D of vertical edge above and below the horizontal edge points in region.
iii. Finding the max S_U in all S_U which satisfies the criterion as $(S_U \geqq 0.6W_{new}) \cap (S_D \leqq 0.2W_{new})$ and $W_{new}=W/W_{actual}$.
iv. The new right vertical border is found at the column where the max S_U happened.

Then, we find a new left vertical border by the similar way. Through the above steps, the candidate is generated based on the detected vertical border and new vertical border.

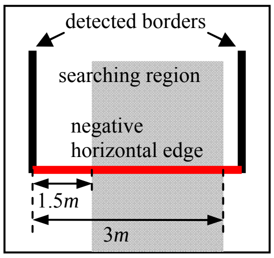


Fig. 5. The diagram of searching region of finding a new vertical border.

b. Finding the symmetric vertical edge pair

The key step of preceding vehicle detection is estimating the preceding vehicle distance to warn the driver. The incorrect preceding vehicle distance will increase the risk of forward collision. Because estimating the preceding vehicle distance is based on the bottom of vehicle, it must be found accurately. The bottom of vehicle can be found by two conditions, the shadow and the symmetric left and right borders of vehicle. As mentioned in before, the procedure substitutes negative horizontal edge for shadow. In this section, since the bottom can't be found by horizontal edge, we propose a method which utilizes the symmetric vertical edge pair to confirm the bottom, left and right borders of vehicle. The method includes two rules: (*i*) the horizontal edge is not located on the most right or left sides of image and (*ii*) others.

The first rule is happened in most cases that vehicles have

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

two vertical borders in image. Since the two vertical borders of vehicle are isometric and symmetric, the proposed procedure finds left and right borders by symmetric vertical edges. To find the vertical edge pair around the horizontal edge, the procedure can be divided into several steps:

i. Defining a searching region around each ends of horizontal edge based on the width of horizontal edge.
ii. Searching a continuous bi-level vertical edge in each region as shown in Fig. 6, where the arrows show the searching directions.
iii. Reserving the pixel of bi-level vertical edge which appears on same y-position in each region as shown in Fig. 7.
iv. Determining the left border, right border, and bottom of vehicle.
v. Estimating the actual distance between two borders.
vi. The actual distance between two borders is longer than 1.5 meters.

The second rule is happened when the vehicle located on the most left or right sides in image that the vehicle only has one vertical border. Since the vehicles only have one vertical border, we can't find the symmetric vertical edge as mentioned in first rule. To solve this problem, the proposed procedure determines a searching region as mentioned in first rule but only on one side of horizontal edge and finds a continuous vertical edge in the region to be the vertical border of vehicle. The bottom of vehicle is determined as the bottom of border.

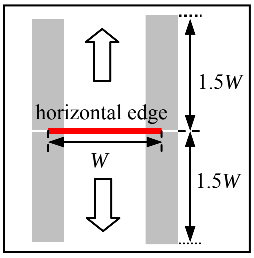
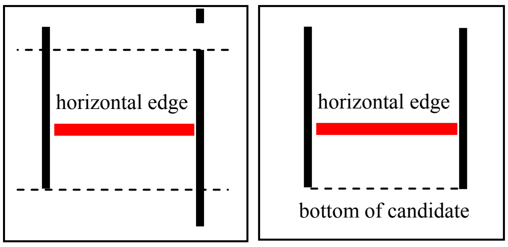


Fig. 6. The diagram of searching region of finding symmetry vertical pair.



(a) (b)
Fig. 7. The example of finding the symmetric vertical edge pair. (a) The original bi-level vertical edges. (b) The vertical edge pair.

III. Vehicle Verification

The candidate vehicle which is generated by vehicle detection has to be verified that it is a vehicle. The proposed method use symmetry to filter overlapping candidates. Since *SVM* is primarily two-class classifier, the proposed vehicle verification uses it to verify the candidate vehicles. To make results more stable, we utilize a statistical approach to achieve that. Finally, the preceding vehicle distance is estimated.

A. Deletion of overlapping candidate vehicles

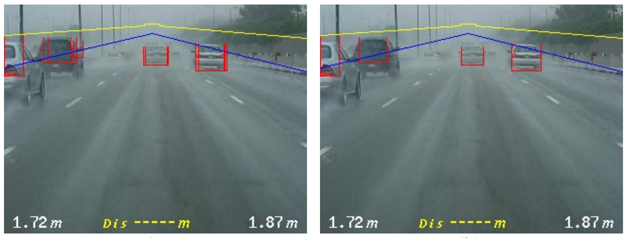
Since the vehicle detection generated candidate vehicles by every horizontal edges, there are many candidates which are overlapping each other belong to the same vehicle. Nevertheless, a vehicle requires a candidate only. With the constraint, deletion of overlapping candidate vehicles is proposed. Because vehicles are high symmetric, the proposed method deletes redundant candidates by estimating symmetry of candidates. The formula of estimating symmetry is defined as

$$Symmetry = \frac{\sum_{h=1}^{H}\sum_{w=1}^{W/2} eng(w,h)}{H}, \quad (2)$$

where H and W are the height and width of the candidate, respectively. The *eng* is defined as

$$eng(w,h) = \begin{cases} \frac{1}{2}[E(\frac{w}{2}-w,h)+E(\frac{w}{2}+w,h)], & \text{if } |E(\frac{w}{2}-w,h)-E(\frac{w}{2}+w,h)| < th \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where $E(x, y)$ is the vertical edge response of (x, y) and T is a pre-defined threshold. The most symmetrical candidate in those overlapping candidates is reserved as shown in Fig. 8.



(a) (b)
Fig. 8. The result of deleting overlapping candidates. (a) The detected candidates. (b) The reserved candidates after deleting overlapping candidates.

B. Support vector machine

The concept of *SVM* which is proposed by Vapnik [11] is to find a separating hyper-plane between two classes and maximizing the distance of either class from hyper-plane. The accuracy of *SVM* is affected by many factors such like parameter, input data and kernel type. In fact, the optimal parameter and kernel type are acquired by testing. Therefore, the input data which is the only factor can be decided in advance must be the significant features that can distinguish the vehicle and non-vehicle clearly.

Many features are used for *SVM* features such like gray level image, edge response, *Gabor* feature and *Haar* wavelet coefficients. By comparing experiment results with different *SVM* features, we choose *Haar* wavelet coefficients to feed the *SVM* which is proposed by Sun *et al.* [6, 7]. First, the candidates are scaled to 32×32. Then, a 5 level *Haar* wavelet decomposition is applied on them and yields 1024 coefficients. Since the *HH* sub-band of the first level encode mostly noise, the proposed method only keeps 768 coefficients.

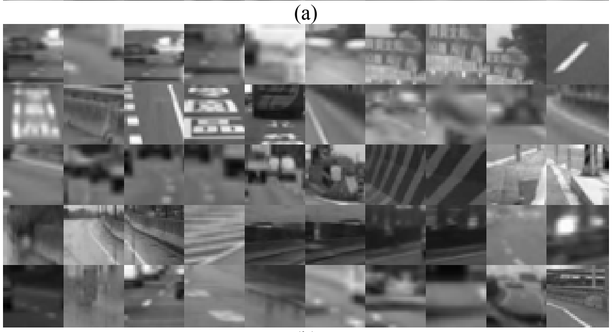
The *SVM* must be trained by positive and negative data in advance. To verify vehicles correctly under different weather conditions, the images of vehicle which are under different weather conditions are selected to be the positive data. In addition, the images of non-vehicle (*e.g.*, road sign and road surface) are selected to be the negative data. The *SVM* is trained by the vehicle (positive) and non-vehicle (negative) images which are captured from the videos as shown in Fig. 9.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Experiments

The proposed vehicle detection method is evaluated on different weathers conditions such as sunny day, cloudy day, rainy day, heavy rainy day, evening and toward the sun. Moreover, we also tested the algorithm with different environments which can effects the procedure on the road surface such like long underneath shadow of vehicle, ground sign and shadow on the road. The examples of different driving environments are shown in Fig. 10. The used image is of 320×240 pixels.

(a)



(b)

Fig. 9. The training data. (a) The vehicle images (positive data). (b) The non- vehicle images (negative data).

(a) (b)

(c) (d)

Fig. 10. The examples of different driving environments. (c) Rainy day. (g) Long underneath of vehicle. (h) Ground sign. (i) Shadow on road surface.

Before the experiment, we have to trained a *SVM* model which is used for vehicle verification. We selected 430 vehicle images and 380 non-vehicle images to train the *SVM* model. After generating *SVM* model, we tested our algorithm with video. Based on the algorithm we proposed, all vehicles are bounded by left, right borders and bottom line, and the results of the preceding vehicle detection under different driving environments are shown in Figs. 11 - 15.

The preceding vehicle detection rate under various weather conditions is given in Table 1, where #Frame is the number of frames in a video; #Vehicle is the total number of vehicles appeared in the same lane as the host vehicle; #Detection is the number of detected vehicles; #False negative is the number of vehicle which isn't detected; #False positive means the number of non-vehicle which is detected as a vehicle; detection rate is the ratio of detected vehicle number to the appeared vehicle number.

Fig. 11. The results of preceding vehicle detection in sunny day in an image sequence.

Fig. 12. The results of preceding vehicle detection with ground sign in cloudy day in an image sequence.

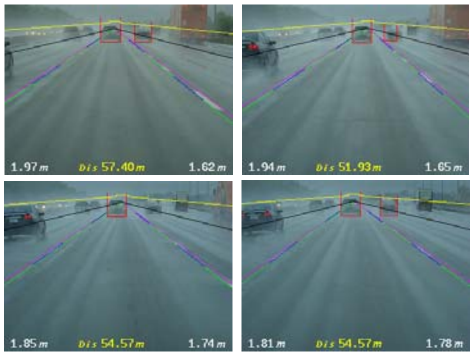


Fig. 13. The results of preceding vehicle detection in rainy day in an image sequence.

Fig. 14. The results of preceding vehicle detection in heavy rainy day in an image sequence.

The purpose of using *SVM* to verify vehicles is to reduce the number of false negative while keeping the detection rate as the detection without *SVM*. In fact, using *SVM* to

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

verify the vehicles may reduce the detection rate; especially in the situation when the borders and bottom of vehicle are not detected correctly or the vehicles are very blurred under bad weather conditions. Also, we found that using *SVM* to verify the vehicles increases the processing time too. We compare the detection rate, the number of false positive, the number of false negative, and processing time between vehicle detection with and without *SVM* as shown in Table. 2.

Fig. 15. The results of preceding vehicle detection with the shadow on road surface in an image sequence.

Table 1. The Vehicle Detection Rate under Different Weather Conditions

	# Frame	# Vehicle	# Detection	# False negative	# False positive	Detection rate
Cloudy day	4000	3745	3236	509	14	86%
Sunny day	5000	4835	4670	165	29	97%
Rainy day	3000	3000	2574	426	11	86%
Ground sign	198	158	148	10	0	94%
Shadow	2000	2000	1924	76	31	96%

Table 2. The Comparison of Vehicle Detection With/Without *SVM*

Weather	With *SVM*	Detection rate	# False negative	# False positive	Processing time (*s*)
Cloudy day	*no*	89%	412	31	0.02424
	yes	86%	509	14	0.03602
Sunny day	*no*	98%	97	116	0.02084
	yes	97%	165	29	0.03181
Rainy day	*no*	89%	325	37	0.03012
	yes	86%	426	11	0.03600
Shadow	*no*	98%	40	115	0.02162
	yes	96%	76	31	0.03389

V. Conclusions

In this study, we used a camera mounted on the moving vehicle to capture image to prevent forward collision by detecting preceding vehicles.

We detected the lane marks to generate *ROI* for vehicle detection and get the vanishing point to estimate the distance of the detected preceding vehicle. We also proposed an exception procedure based on the vanishing point and camera height. If the location of the vanishing point is out of the expectation range or missed, we used pre-defined vanishing point to estimate the preceding vehicle distance. To make vehicle detection adapt to different weather conditions, we used horizontal and vertical edges instead of underneath shadow of vehicles to detect vehicles. Moreover, we used an adaptive thresholding method to generate bi-level images.

We used vertical edge to check the position of the negative horizontal edge under the object. Since the negative horizontal edge is located at the bottom of object, it is simple to find the left and right vertical borders of object. However, in some times, we can't find the bottom of object by negative horizontal edges. Therefore, we proposed a method to find the vertical edge pair around the horizontal edge. The bottom of vehicle and two vertical borders are generated by the vertical edge pair.

To enhance the accuracy, we used the *SVM* to verify detected candidates. Then, we estimated the preceding vehicle distance to alarm the driver.

The techniques of computer vision and image processing have been applied in many related fields to the Intelligent Transportation System (*ITS*). This work aids the safety vehicles. However, it is hard to detect the nearby vehicles accurately just using image processing. Moreover, we must find other significant features such as taillights and headlights to detect vehicles at night. With detecting vehicles at night, our work will be more complete. The above problems are just the next challenge in our future work.

Acknowledgment

This work is partially funded by the National Science Council, Taiwan under the grant of the research project *NSC* 97-2221-*E*-008 -073 -*MY*3.

References

[1] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE, Trans. Pattern Analysis and Machine Intelligence*, vol.28, no.5, pp.694-711, May, 2006.

[2] J. M. Armingol, A. de la Escalera, C. Hilario, J. M. Collado, J. P. Carrasco, M. J. Flores, J. M. Pastor, and J. Rodríquez, "IVVI: Intelligent vehicle based on visual information," *Robotics and Autonomous Systems*, vol.55, no.12, pp.904-916, Dec. 2007.

[3] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, vol.12, no.2, pp.69-83, Sep. 2000.

[4] L.-S. Jin, B.-Y. Gu, R.-B. Wang, L. Guao, Y.-B. Zhao, and L.-H. Li, "Preceding vehicle detection based on multi-characteristics fusion," in *Proc. IEEE Int'l Conf. on Vehicular Electronics and Safety*, Beijing, China, Dec.13-15, 2006, pp.356-360.

[5] C.-C. Wang, and J.-J. Lien, "Automatic vehicle detection using local features–A statistical approach," *IEEE Trans. Intelligent Transportation Systems*, vol.9, no.1, pp.83-96, Mar. 2008.

[6] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machine," in *Proc. IEEE Int'l. Conf. on Digital Signal Processing*, Santorini, Greece, Jul.1-3, 2002, pp.1019-1022.

[7] Z. Sun, G. Bebis, and R. Miller, "Quantized wavelet features and support vector machines for on-road vehicle detection," in *Int'l. Conf. on Control, Automation, Robotics and Vision*, Singapore, Dec.2-5, 2002, pp.1641-1646.

[8] B. Leung, *Component-based Car Detection in Street Scene Images*, Technique Report, Massachusetts Institution of Technology, 2004.

[9] D.-C. Tseng, *Monocular Computer Vision Aided Road Vehicle Driving for Safety*, *U.S. Patent*, No. 6765480, Jul. 20, 2004.

[10] C.-W. Lin, H.-Y. Wang and D.-C. Tseng, "A robust lane detection and verification method for intelligent vehicles," in *Proc. of The 3rd Int'l. Conf. on Intelligent Information Technology Application*, Nachang, China, Nov.21-22, 2009, pp.521-524.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)