International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 8, No. 2 (2015), pp. 201-212
http://dx.doi.org/10.14257/ijsip.2015.8.2.20

On-board Robust Vehicle Detection Using Knowledge-based Features and Motion Trajectory

Wenhui Li, Peixun Liu, Ying Wang, Hongyin Ni, Chao Wen and Jiahao Fan

1. College of Computer Science and Technology, Jilin University
2. State Key Laboratory of Automotive Simulation and Control, Jilin University
wangying_jlu@163.com

Abstract

This paper presents a robust and efficient method for vehicle detection in dynamic traffic environments. First, two adaptive vehicle hypothesis generation methods based on shadow and vehicle wave are presented, and then we assemble these two features into vehicle hypothesis. A hypothesis verification algorithm based on vehicle motion trajectory is proposed, the on-line hypothesis verification algorithm based on vehicle motion trajectory can not only reduce the false positive alarm caused by interferences, but also handle the problem that the classifiers generated in the off-line training phase is closely related to the diversity of positive and negative samples. Quantitative analysis on both public vehicle image datasets and real-time video presents a result of 85.58% detection rate with 4.13% false positive rate. And our algorithm could run as fast as 40ms/frame on PC platform.

Keywords: *Vehicle detection, Vehicle wave, Motion trajectory, Knowledge-based features fusion*

1. Introduction

As an important part of the Intelligent Transportation Systems, Intelligent vehicles are developed to solve not only safety but also energy-saving problems that have drawn intensive attention. Owing to the developments of vision sensors, vision-based vehicle detection technologies have become the focus in intelligent vehicle research field. Forward Collision Warning System (FCWS), Blind Spot Detection Systems (BSDS) are essential parts of intelligent vehicles. They all demand on the performance of vehicle detection methods. The majority of methods reported in the literatures follow three main phases: 1) Hypothesis Generation (HG). 2) Hypothesis Verification (HV) [1] and 3) Vehicle Tracking (VT) whose results ensure the vehicles in the image are continuously detected.

To measure the relative distances and speeds between the self-vehicle and target-vehicles precisely, various hypothesis generation approaches have been proposed in the literatures. They can be classified into one of the following three categories [1]: knowledge-based, stereo-based, and motion-based methods. Knowledge-based methods employ a priori knowledge to generate a vehicle region of interest in an image. Plenty of representative approaches use information about shadows underneath vehicles [2, 14], symmetry of a vehicle in the horizontal and vertical directions [2], vertical and horizontal edges of different views of a vehicle [2-4, 10, 13, 14, 17], corners on surface of a vehicle [2, 4], texture, and vehicle rear-lamp [16, 18]. Stereo-based methods are also employed for vehicle detection by using the stereo information, such as Inverse Perspective Mapping (IPM) [6]. In addition, motion- based methods exploit spatial features are used to distinguish between vehicles and background. For example, vehicles in images are detected by using relative motion information which is obtained by the calculation of optical flow. HV is performed to verify the correctness of the results detected in HG phase.

ISSN: 2005-4254 IJSIP

HV Methods can be classified into two categories: template-based and appearance-based [1]. Appearance- based approaches are more widely used in HV phase, the brief steps of these methods are: first capture the variability of vehicle appearance and then learn the characteristics of the vehicle class from a set of training images. Learning-based methods yield a decent performance in the recent literatures, such as Haar+Adaboost [7], HoG+SVM [8], PCA-ICA+GMM [9], minimum Mahalanobis distance classifier [2, 10], HOG+Adaboost [5] and Active-learning framework [11]. VT is processed to make sure the vehicles in video sequences are continuously detected. Kalman fliter, Meanshift, on-line boosting, Particle Filtering [6, 8], parse optical flow computation [2] and RANSAC-based outlier rejection [2] are frequently used in VT phase.

Detecting vehicles by searching the whole image by using Appearance-based approaches are excessively computational which cannot achieve the real-time requirment for driver reaction. The efficient hypothesis generation approaches could achieve high detection rate, but generate plenty of false positives. Most of appearance-based hypothesis verification methods need offline learning to generate the classifiers whose performance are closely related to the diversity of vehicle samples. Therefore, to resolve the above problems, we first assemble two features based on shadow and vehicle wave into vehicle hypothesis. Then an on-line hypothesis verification algorithm based on vehicle motion trajectory is used to remove the false positives whose positions in images are discrete within a certain time sequence.

The rest of the paper is organized as follows: Section 2 discusses the fusion of two knowledge-based HV methods which based on shadow and horizontal-edge pixels histogram. Section 3 presents the proposed method based on vehicle motion trajectory. Results for the proposed method are shown in Section 4. Finally, Section 5 draws conclusions.

2. Detection by Knowledge-based Features

In this section, we first introduce a shadow-based vehicle detection method to detect the vehicle ROIs in the image. Then vehicle detection method based on vehicle waves which generate from the Horizontal edge pixels histogram (HEPH) are used to get rid of much interference in the vehicle ROIs. Finally, these two knowlege-based methods are fused together to form the final results of HG phase.

2.1. Detection based on Shadow

The basic principle of shadow-based vehicle detection method is: the regions underneath vehicles are distinctly darker than the other regions on an asphalt paved road, and the grayscale value of pixels in shadow regions are much lower than that of any other pixels in the same image. Grayscale histogram (GH) can reflect the whole image grayscale value distribution well. The grayscale values of vehicle shadow pixels belong to the lower parts of GH. So we can detect the shadow regions underneath vehicles by segmenting GH with an adaptive threshold *threshold_BW*.

$$I\_shadow(x,y)=\begin{cases}0, & I\_grayscale(x,y)<threshold\_BW\\ 255, & otherwise\end{cases} \tag{1}$$

$$threshold\_BW,\quad if\quad p_{k-1}<th\_BW\ \&\&\ p_k\geq th\_BW \tag{2}$$

$$p_k=\sum_{j=0}^{k}\frac{n_j}{N},k\in[0,L-1] \tag{3}$$

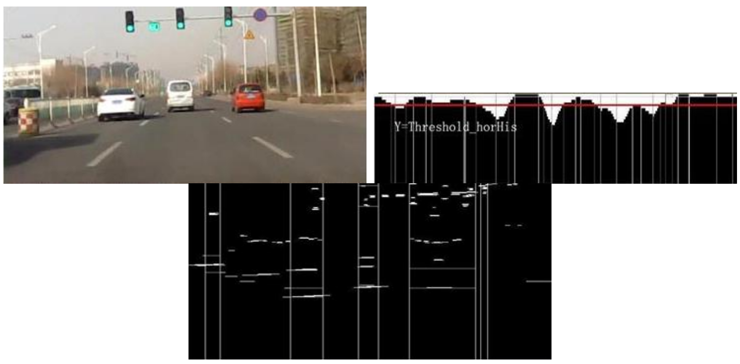
In Equation (1), *I_shadow* is the shadow image, and *I_grayscale* is the grayscale image of origin image. Equation (2) is used to get the threshold *threshold_BW*. p_k is the percentage which is acquired by dividing the number of pixels whose grayscale values belong to the 0-level to the k-level in GH by N. N presents the number of pixels in *I_grayscale*. n_j is the bin-value of the j^{th} level in GH, L is the number of levels in GH. If p_{k-1} is lower than *th_BW* and p_k is more than *th_BW*. *threshold_BW* is set to the grayscale value in the k^{th} level of GH. Figure 1 shows the process of vehicle detection method based on shadows. Figure 1 (b) is the detected shadow image of Figure 1 (a) when threshold *th_BW* is set to 5%, and Figure 1(c) is erosion result of Figure 1(b). To obtain the shadow-lines underneath the vehicles at the bottom of image, each pair of shadow-lines in Figure 1(c) whose distances are within 5 pixels are combined from the top to bottom. Black lines in Figure 1(d) are the final results of shadow-based method.

(a) (b) (c) (d)

Figure 1. Vehicle Detection based on Shadow

2.2. Detection Based on Vehicle Wave

Different views of a vehicle, especially rear/frontal views, contain many regular horizontal and vertical structures. The horizontal and vertical edges are typical characteristics of these structures. Recently, edge-based methods [2, 4, 10] are widely used in the vehicle hypothesis generation phase due to the high efficiency of features in performance, simpleness in implementation. In this paper, we propose a vehicle detection algorithm based on HEPH which is generated by summing the horizontal edge pixels in each column.



(a) (b) (c)

Figure 2. Vehicle Detection based on Vehicle Wave

Figure 2 illustrates the process of vehicle detection method based on vehicle wave. Our vehicle detection method based on vehicle wave can be divided into following three steps:

Set the Detection ROI. The top region in the image captured from vehicle-mounted camera is always sky. To reduce the computational load, we first set 2/3 bottom region of image as detection ROI *Detection_region*, Figure 2 (a) is the *Detection_region* of image.

Calculate the Vehicle Waves. Horizontal edges in the *Detection_region* are detected by applying the horizontal Sobel filter. The HEPH of *Detection_region* is computed by summing the horizontal edge pixels in each column of *Detection_region*. Then, vehicle waves are obtained by processing a Median filtering on it. Figure 2(b) illustrates the vehicle waves of vehicles.

Calculate the Coordinates of Vehicles in the Image. The x-coordinates of the left-top and right-bottom points of vehicles in the image are obtained by segmenting the vehicle waves with a threshold line. The threshold line Y=Threshold_horHis is set adaptively according to the width of *Detection_region* by Equation (4).

$$Threshold\_horHis = \sum_{i=0}^{Width\_roi} \frac{Pn_i}{Width\_roi} \tag{4}$$

Width_roi is the width of *Detection_region*, Pn_i is the number of horizontal edge pixels in each column. The white vertical lines in Figure 2(c) are left and right borders of vehicles. According to these borders, we get several white rectangles named *Detection_region_s*, which are shown in Figure 2(c). There are a large number of horizontal lines in each *Detection_region_s*. The distance between two adjacent horizontal lines belonging to the same vehicles are short. Due to the sizes of vehicles captured from different distances are various, we set an adaptive threshold according to the width of *Detection_ region _s*. Therefore, two adjacent lines belong to the same vehicle area could be judged by the following rules. The y-coordinates of the left-top and right-bottom points of vehicles are obtained by scanning horizontal lines from top to bottom：

Rule1. If $Horizontal\_line_i$ is the first line appeared in *Detection_region_s*, $Horizontal\_line_i$ is the top border of the first object in *Detection_region_s*.

Rule2. If $Horizontal\_line_i$ is the last line appeared in *Detection_region_s*, $Horizontal\_line_i$ is the bottom border of the last object in *Detection_region_s*.

Rule3. If the distance between $Horizontal\_line_i$ and its adjacent line $Horizontal\_line_{i+1}$ is smaller than *Threshold_dis*, the two horizontal lines are belonging to the same object.

Rule4. If the distance between $Horizontal\_line_i$ and its adjacent line $Horizontal\_line_{i+1}$ is larger than *Threshold_dis*. $Horizontal\_line_{i+1}$ is the bottom line of Jth object and $Horizontal\_line_{i+1}$ is the top line of J+1th object.

$$Threshold\_dis = \frac{W\_detection\_region\_s}{th\_M} \tag{5}$$

Plenty of top and bottom horizontal-lines are obtained after scanning the horizontal lines from top to bottom in each *Detection_region_s*. According to the scales of vehicles and the symmetry of a vehicle in the horizontal direction, pairs of top and bottom horizontal lines are regarded as the top and bottom borders of vehicles. Therefore, we acquire the coordinates of the left-top and right-bottom points of vehicles in the image.

2.3. Knowledge-based Features Fusion

The intensity of the shadows underneath vehicles depends on the illumination of images, which closely relates to the weather condition. The shadow of the tree or road boundary is similar to the vehicle. In addition, the location and length of shadow underneath the vehicle in the image vary with the direction of the sunlight. As it is illustrated in Figure 3 (a), the detected shadow is longer than the bottom line of the vehicle, and the trees beside the road are also detected as vehicles. Buildings and trees on both sides of road are detected by using method based on horizontal edges. In this paper a feature fusion method based on two features is introduced, the processing flow is presented in Figure 3. The white rectangles in Figure 3(a) are the detection results of shadow-based method. Figure 3(b) is the horizontal edge image of vehicle area in Figure 3(a) by applying the horizontal Sobel filter. Figure 3(c) is the HEPH acquired by summing the horizontal edge pixels in each column of Figure 3(b). Figure 3(d) is the vehicle wave generated from HEPH, and Figure 3(e) is the final result detected using rules described in Section 2.2.

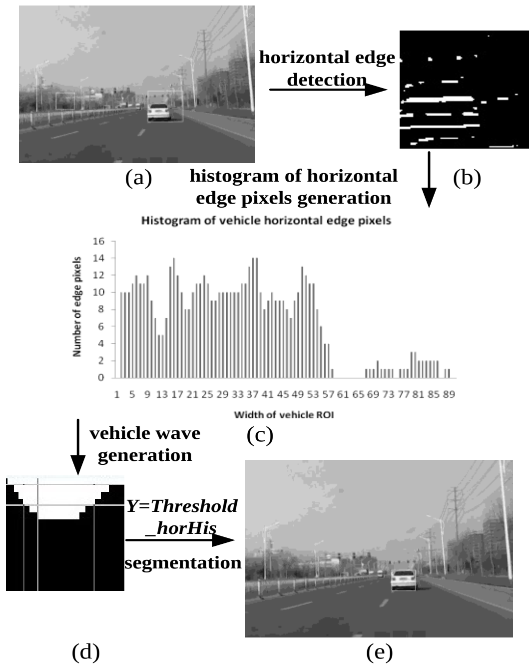


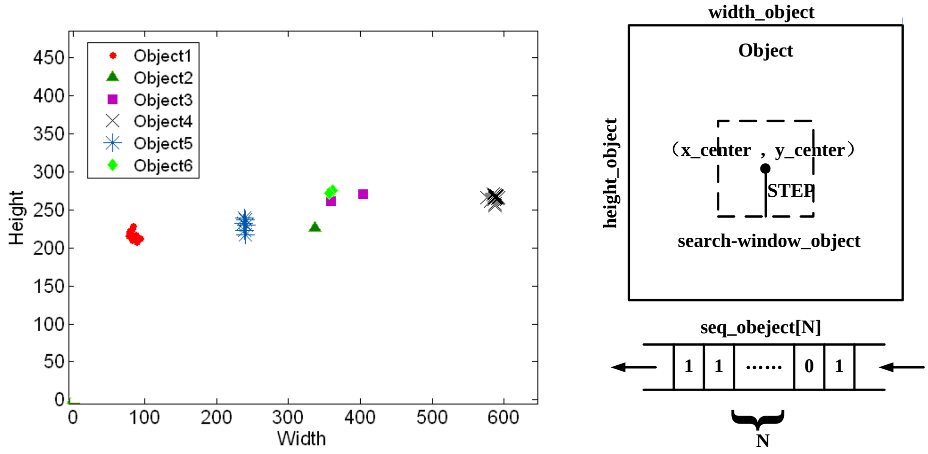
Figure 3. Vehicle Detection based on Feature Fusion

3. Verification by Motion Trajectory

The method introduced in Section 2.3 can detect vehicle well, but there are many false positives when detect the vehicles in more complicated scenes. To reduce the false positives, we propose a video-based vehicle verification algorithm based on motion trajectory. The main basis of this algorithm is relative velocities between the host vehicle and the target vehicles are different from the ones between the host vehicle and the false positives. And the vehicle detection rate is higher than the false alarm rate of false positives in continuous N frames. Therefore, the motion trajectories of vehilces are continious and the motion trajectories of false alarms are discreted. As shown in Figure 4(a), there are M objects detected by our multi-feature fusion method in continuous N frames. To fully use the information of these objects in every frame, each detected object is represented in form of Object. In Figure 4(a), Object1, Object4 and Object5 are vehicles in the video.

3.1. Definition of Object

A data structure is defined for these objects: Object = *{x_center, y_center, width_Object, height_Object, search-window_Object, seq_Object*$_j$*}*. Figure 4(b) illustrates the definition of Object. *x_center, y_center* are x , y-coordinates of center point of object, *width_Object, height_Object* are width and height, *search-window_Object* is a square which center is object center , and the width of the square is 2×STEP , *seq_Object*$_j$,j=1, 2…N, is used to record the times that the center of object is in *search-window_Object.*



(a) (b)

Figure 4. Distributions of Each Object in Continuous N Frames and Definition of Object

3.2. HV Algorithm based on Motion Trajectory

HV Algorithm Based on Motion Trajectory

Initialization: Vehicles are detected by HG method in current frame, and then record the information of vehicles by using data structure $Object_j$.

for continuous N frames **do**

for all vehicles $Object_j$ **do**

for all vehicles $Object_i$ in previous frame **do**

if x_center_i-STEP$\leq x_center_j \leq x_center_i$+STEP&&$y\_center_i$-STEP$\leq y_center_j \leq y_center_i$+STEP

then

$seq\_Object_{in} = 1$

$flag_j = 1$

else

$seq\_Object_{in} = 0$

end for

if $flag_j = 0$ then

create a new $Object_{i+1}$

end if

end for

for all seq_Object_i **do**

if $seq\_Object_{in} = 1$

count_1++

else

count_0++

end for

---

if count_1 ≥ *th_V*

$Object_i$ is a vehicle

end if

if count_0=N

$Object_i$ is a false positive

end if

end for

5. Experiments

5.1. Experimental Datasets and Performance Metrics

Two main data sets were used to quantify the performance of vehicle detection. The first data set is consisted of 126 distinct static vehicle test images which are from the publicly available Caltech 1999 data set [19]. The second data set is the video data set, in order to evaluate the performance of our proposed system. We applied the system on a testing set consisting of 6 different video sequences, detailed in Table 1. The length of each video sequence is different. More than 50 different vehicles are captured in these videos. The videos were captured by CMOS cameras with 640× 480 resolutions under different seasons, weather conditions and urban roads. Scenario1, 2, 3 are captured from a CCD camera which is mounted behind the driving mirror, and Scenario4, 5, 6 are captured from another CCD camera which is mounted under the rearview mirror. Programs run on a PC with Intel Core i3 540 3.07GHz CPU equipped. As Ref. [10, 11], we use the following indicators to measure the algorithm performance: detection rate (DR), false alarm rate (FAR), localization, robustness, efficiency. In this paper, the performance of a detection module is quantified by the metrics which are widely used [9-11]: detection rate, false alarm rate and average processing time per frame.

Table1. Details of Test Videos

Video	Frames	Weather	Application sceenarios
Scenario1.avi	100	Sunny	Urban road
Scenario2.avi	300	Cloudy	Urban road
Scenario3.avi	300	Cloudy	Urban road
Scenario4.avi	200	Sunny	Urban road
Scenario5.avi	300	Snowy	Urban road
Scenario6.avi	100	Rainy	Urban road

In our research the detection rate is the proportion of non-occluded and half-occluded vehicles in the camera's view that are detected, and the distances between cameras and targets are within 50m in the real scenarios. This quantity measures recall and localization, DR is defined as Equation (6). The false alarm rate is the percentage of detections that were not real vehicles among detections. We define the FAR by dividing the number of false positives by the total number of detections. FAR is also a measure of precision and localization, it is defined as Equation (7). The processing time is closely related to the number of vehicles in each frame. In this paper we use a metric named average processing time per frame (AVT) to measure the efficiency of algorithms processing on each video. The AVT is defined as Equation (8).

$$Detection\ Rate(DR) = \frac{Number\ of\ detected\ vehicles}{Total\ number\ of\ vehicles} \quad (6)$$

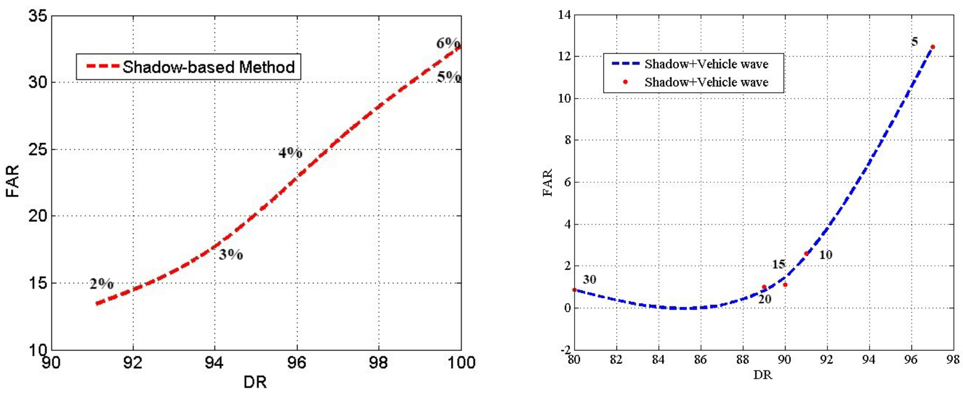
$$False\ Alarm\ Rate(FAR) = \frac{Number\ of\ false\ positives}{Number\ of\ detections} \quad (7)$$

$$AVT = \frac{Sum\ of\ processing\ time\ for\ each\ frame}{Number\ of\ frame} \quad (8)$$

5.2. Main Parameter Settings

We first evaluate the proposed methods in Section 2 on publicly available Caltech 1999 and images in our test videos. Figure 5 (a) illustrates the ROC curve of vehicle detection method based on shadow. The ROC curve is obtained by setting various values to threshold *th_BW* in Eq. (2). To detect more vehicles and less false positive, according to Figure 5(a), we set *th_BW*=5% in further experiments of this paper.

As it is shown in Figure 5 (b), ROC curve presented in section 2.3 is obtained by setting various values to the threshold *th_M* in Equation (5). The threshold *th_BW* in Equation (2) is 5%, *th_M* is set to 20 in further experiments, Figure 6 shows the detection results of two knowledge-based features fusion (shadow and vehicle wave) on publicly available Caltech 1999, the green lines underneath the vehicles are results of shadow-based method and the red boxes are results of two features fusion method presented in of Section 2.3.



(a) (b)

Figure 5. ROC of Shadow-based Method and Shadow+vehicle Wave Method

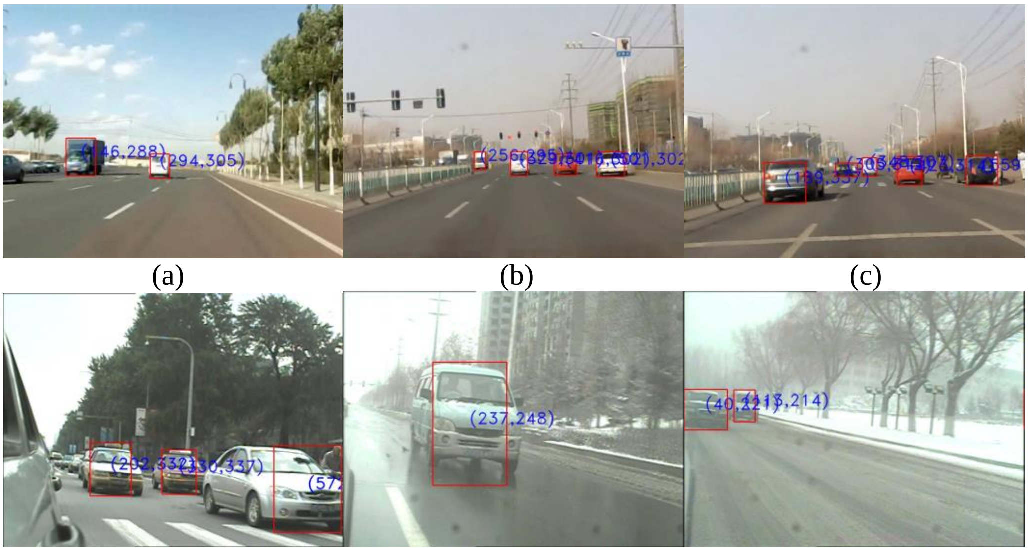
Figure 6. Detection Results of Two Knowledge-based Features Fusion on Publicly Available Caltech 1999

5.3. Results and Comparisons

We use videos introduced in Section 5.1 to test our methods. Table 2 shows the detailed results by setting N to 8, STEP to 5 and th_V to 4 according to the experimental results on different scenarios. Comparisons between the shadow-based + vehicle-wave-based and shadow-based + vehicle-wave-based + Motion trajectory are shown as Table 2. Results in Table 2 shows the HV algorithm based on motion trajectory can reduce the false alarms. As it is shown in Figure 7, the proposed method can detect both rear and front of vehicles in different scenarios. The numbers in Figure 8 are the center coordinates of vehicles in images.

Table 2. Results of Proposed Methods on Test Videos

Video	Shadow+Vehicle-Wave		Shadow+Vehicle-wave+Motion trajectory	
	DR	FAR	DR	FAR
Scenario1	100%	20%	100%	5%
Scenario2	90%	25%	85%	1.08%
Scenario3	82%	51%	72.78%	5%
Scenario4	99.71%	6.30%	98.27%	0.67%
Scenario5	90.35%	2.50%	88.42%	0%
Scenario6	72%	25%	69%	13%
Average	89.01%	21.63%	85.58%	4.13%



(a) (b) (c)

(e) (f) (g)

Figure 7. Detection Results of Real Data Sets, (a), (b), (c), (d), (e), (f) is Detection Result of Scenaio1, 2, 3, 4, 5 ,6 respectively

To verify the performance of our method, we compare our method to the vehicle detection methods in [7-9]. Table 3 shows that these three methods outperform our method in terms of accuracy, but the processing time of their methods are all above 500ms. Comparing with other methods, our algorithm could run as fast as 40ms/frame on PC platform. Considering both the accuracy and the processing time of algorithms, our method outperforms the other methods.

Table 3. Comparisons with the Other Methods

Methods	Accuracy (DR/FAR)	AVT ms/frame (Resolution)	Application scenarios
Tsai[7]	96.6% /2%	570 (640×480)	Sunny, tunnel, high way
Niknejad[8]	97%/0.26%	500 (640×480)	Sunny, urban
Wang[9]	94.2%/0.002%	500 (320×240)	Sunny,urban
Proposed	85.6%%/4.13%	40 (640×480)	Sunny, rainy, snowy, urban

6. Conclusions

In this paper, a new vehicle detection method is proposed. Robustness and efficiency are achieved by combining two adaptive HG approaches based on shadow and vehicle wave. The HG step make sure the efficiency and robustness of our method. Our on-line hypothesis verification algorithm which based on vehicle motion trajectory can not only reduce the false positive alarm caused by interferences, but also solve the problem that the classifiers generated in the off-line training phase is closely related to the diversity of positive and negative samples. Experimental results show that our method can well detect vehicles in videos. But, to generalize our algorithm, there are still several problems to be resolved, such as our method can not handle the problem that many vehicles occluded each other during the vehicle detection phase. To improve the performance of vehicle detection methods, we will address these issues and improve the multi-vehicle detection to an upper-level.

Acknowledgements

This work is supported by the grants from Jilin Planned Projects for Science Technology Development [Grant No. 20120305, No. 20140101181JC and No. 20130522119JH]. Ph.D. Programs Foundation of Ministry of Education of China [Grant No. 20130061110054].

References

[1]. Z. H. Sun, G. Bebis and R. Miller, “On-Road Vehicle Detection: A Review”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, **(2006)**.
[2]. J. Arrospide, L. Salgado, M. Nieto, and F. Jaureguizar, “On-board robust vehicle detection and tracking using adaptive quality evaluation” , IEEE International Conference on Image Processing, San Diego, California, USA, **(2008)** October.
[3]. B. F. Lin, Y. M. Chang, L. C. Fu, P. Yung, and L. A. Chuang, “Integrating Appearance and Edge Features for Sedan Vehicle Detection in the Blind-Spot Area”, IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 2, **(2012)**.
[4]. J. Hwang, K. Huh and D. Lee, “Vision-based Vehicle Detection and Tracking Algorithm Design”, Optical Engineering, vol. 48, no. 12, **(2009)**.
[5]. B. Southall, M. Bansal and J. Eledath, “Real-time Vehicle Detection for Highway Driving”, IEEE Conference on Computer Vision and Pattern Recognition Miami, Florida, USA, **(2009)** June.
[6]. J. Arrospide, L. Salgado, and M. Nieto, “Multiple object tracking using an automatic variable dimension particle fliter”, IEEE International Conference on Image Processing, Hong Kong, (**2010)** September.
[7]. Y. M. Tsai, K. Y. Huan, C. C. Tsai and L. G. Chen, “An exploration of on-road vehicle detection using hierarchical scaling schemes” , IEEE International Conference on Image Processing, Hong Kong, **(2010)** September.
[8]. H. T. Niknejad, A. Takeuchi, S. Mita and D. McAllester, “On-Road Multivehicle Tracking Using Deformable Object Model and Particle Filter With Improved Likelihood Estimation”, IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 2, **(2012)**.
[9]. C. R. Wang and J. J. Lien, “Automatic Vehicle Detection Using Local Features-A Statistical Approach”, IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, **(2008)**.
[10]. D. Alonso, L. Salgado, M. Nieto, “Robust Vehicle Detection Through Multidimensional Classification For on Board Video Based Systems”, IEEE International Conference on Image Processing, San Antonio, Texas, USA, **(2007)** September.
[11]. S. Sivaraman and M. M. Trivedi, “A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking”, IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 2, **(2010)**.

[12]. A. Jazayeri, H. Cai, J. Y. Zheng and M. Tuceryan, "Vehicle Detection and Tracking in Car Video Based on Motion Model", IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, **(2011)**.
[13]. S. Tanaka, K. Yamada, T. Ito, and T. Ohkawa, "Vehicle detection based on perspective transformation using rear-view camera", International journal of vehicular technology, **(2011)**.
[14]. Y. M. Chang, S. S. Huang, L. C. Fu, P. Y. Hsiao and M. F. Lo, "Vehicle detection and tracking under various lighting conditions using a particle filter", IET Intelligent Transport Systems, vol. 6, no. 1, **(2012)**.
[15]. Y. Zhu, D. Comaniciu, M. Pellkofer, and T. Koehler, "Reliable Detection of Overtaking Vehicles Using Robust Information Fusion", IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 4, **(2006)**.
[16]. R. O. Malley, E. Jones and M. Glavin, "Rear-Lamp Vehicle Detection and Tracking in Low-Exposure Color Video for Night Conditions", IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 2, **(2010)**.
[17]. L. W. Tsai, J. W. Hsieh and K. C. Fan, "Vehicle Detection Using Normalized Color and Edge Map", IEEE Transactions on Image Processing, vol. 16, no. 3, **(2007)**.
[18]. A. Fossati, P. Schönmann and P. Fua, "Real-time vehicle tracking for driving assistance", Machine Vision and Applications, vol. 22, **(2011)**.
[19]. Caltech Cars, **(1999),** http://www.vision.caltech.edu/html-files/archive.html.

Authors

Wenhui Li, was born in 1961. He is a Professor and doctor supervisor in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His major research interests include computer vision, image processing, pattern recognition, graphics, virtual reality, CAD and geometric constraint.

Peixun Liu, was born in 1986. Since 2011, He has been working on the PhD degree in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His research interests include computer vision, image processing and pattern recognition.

Ying Wang, was born in 1983. She received her PhD degree in 2012 from College of Computer Science and Technology, Jilin University, P. R. China. She is now a lecturer in the College of Computer Science and Technology, Jilin University. Her research interests include automotive simulation and control, face recognition, fire/smoke detection, pedestrian detection and object tracking.