Accurate vehicle detection and counting algorithm for traffic data collection

Conference Paper · October 2015

DOI: 10.1109/ICCVE.2015.60

Accurate Vehicle detection and counting algorithm for traffic data collection

Ye Tian[1], Yangli Wang[1]*, Song Rui[1,2], Huansheng Song[3]

[1]State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, 710071 P.R. China.

[2]Key Laboratory of Infrared System Detection and Imaging Technology, Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai, 200083, P.R. China.

[3] Department of Electronic and Information Engineering, Chang'an University, Xi'an, P.R. China

*Corresponding authors: yliwang@mail.xidian.edu.cn

***Abstract*—Number of vehicles on road is very important traffic data and is essential for transportation safety and management. In this paper, an approach for vehicle detection is presented. In this approach, virtual line based sensors which are just straight detection lines are first set on road lanes. Then two features, namely gradient and range feature, are proposed for vehicle detection. This is carried out by extracting and analyzing the two features on detection lines. Meanwhile, the solution for vehicle occlusion have also been proposed. Our proposed method has an outstanding advantage that it performs excellent in traffic jams as well as under various conditions, such as sunny, cloudy, and rainy days, or night time, or even tunnels with complex illumination. The high accuracy rate of our method is verified with the experiment results.**

Keywords—Vehicle detection, Intelligent Transportation system (ITS), line based sensor, gradient feature, range feature

I. INTRODUCTION

With the accelerating pace of urbanization, a growing number of vehicles are now running on roads, causing severe problems such as traffic jam and speeding. So traffic data is very important for traffic flow management and analysis. And vehicle number is one of the key parameter in traffic flow data. For example, via calculating the number of vehicles passed in unit time, we can get the degree of road congestion. As such, it is of great significance to deploy an intelligent transportation system (ITS) for monitoring the number of vehicles. As an integral part of ITS, vehicle detection [1] has thus gained huge research and technology interests, and a lot of methods have been proposed. These methods achieve this by detection and tracking of vehicles using the appearance (e.g. background based method) or the motion of vehicles [2], [3].

Recently, lots of vehicle detection algorithm based on computer vision has been developed. In the literature, methods for vehicle detection can be divided into the following two categories: appearance based and motion based ones, and the appearance based methods can also be divided into two categories, background subtraction based and feature extraction based methods. A color image-based background subtraction method with shadow removal algorithm is developed in [4]. And the background extraction method has been used to detect and count cars at entrances and exits of multi-story car parks in [5]. However, methods based on background subtraction are usually not very satisfactory in complex and fast-change scenes. So, most of the current methods are not purely based on background learning and subtraction. For improving vehicle detection accuracy rate, many feature based methods, especially feature points based ones e.g. [6], [7], [8], [9], [10], and [11], have been developed for different application scenarios with different feature point extraction algorithms. In motion based approaches, vehicles are detected based on some motion features. For example in [12], vehicles are detected based on different motion vectors. In [13], a time-spatial images are proposed for vehicle detection and classification. A similar approach is also used in [14] for traffic flow estimation and vehicle-type classification. For tracking vehicles, the most widely-used method is the Kalman filtering [15]. Kalman filtering is well suitable for predicting and tracking location of a vehicle when its path and speed does not change rapidly. However, none of these methods can effectively work under all complex conditions, such as rainy days, night time, and crowded scenarios with traffic jams and complex illumination. Thus, it is necessary to develop an algorithm for vehicle detection and counting, which can adapt to and handle various conditions.

In this paper, an approach with great simplicity and efficiency is proposed, which just analyzes two simple features, i.e. gradient feature and range feature. And as we expected, it works well in various weathers, different time and roads. Meanwhile, it gives outstanding results under all conditions, which is demonstrated in experiment results.

II. VEHICLE DETECTION AND COUNTING

A. Observations

In conventional vehicle counting methods, vehicles are detected in a whole video frame with features to differentiate appearance of road and vehicles. They work well for scenes with light traffic and good weathers. But for extremely crowded scenes and scenes with complex illumination, detecting vehicles from a whole video frame becomes very difficult because of little gap between vehicles, shadow and occlusions. They will connect vehicles together. To circumvent such difficulties, in this paper we just use one detection line on each lane to minimize effects of shadow, occlusion and little gap between vehicles, as shown in Fig. 1. The detection line should be set as close as possible to the bottom of the frame, where the gap between vehicle is larger, As the yellow detection lines drawn in Fig. 1.

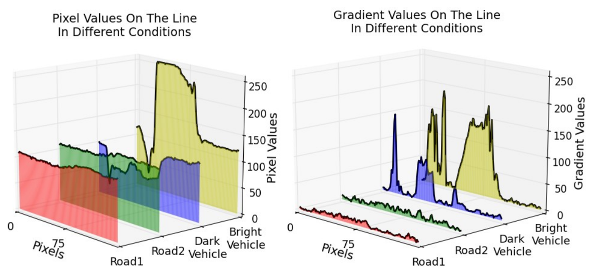
Fig. 1. Using detection line in crowded scenes.

In general, according to their colors, two types of vehicles can be found in a grayscale video sequence, i.e. vehicle with bright color and dark color, which leads to two different modes of pixel values on the detection line for a possible vehicle passing. It inspires us to detect vehicles via analyzing the pixel values on the detection line, as shown in Fig. 2(a). We pick four frames and draw the pixel values on the detection line in Fig. 2(a). Two of them are road's pixel values, and others represent the pixel values of a bright vehicle and a dark one, respectively.

From the pixel values on the detection line under different situations, four conclusions as follows can be drawn:

- Pixel values of road change little.
- Pixel values of vehicles and road are significantly different.
- The envelope of a dark vehicle's pixel values is approximately concave.
- The envelope of a bright vehicle's pixel values is approximately convex.

Based on the bar graphs in Fig. 2(a) and the above conclusions, passing vehicles can be detected using the detection line via examining modes of the pixel values. However, the pixel value modes may change with weather, illumination, and time. Meanwhile, vehicles with different colors may also fall in entirely different pixel value modes, as the blue and yellow bar graphs shown in Fig. 2(a). To handle these problems, we are thus inspired to analyze the pixel values using gradient. For this purpose, a 3x3 horizontal and vertical Sobel operator is respectively applied for generating the gradient images, as shown in Fig. 3. By the way, a canny operator [16] is also suitable for our approach. And Fig. 2(b) gives the gradient values on the detection line of the same frames in Fig. 2(a).



(a) (b)

Fig. 2. (a) Pixel values of Road, dark vehicle, and bright vehicle. (b) Gradients values of the same frames in (a)

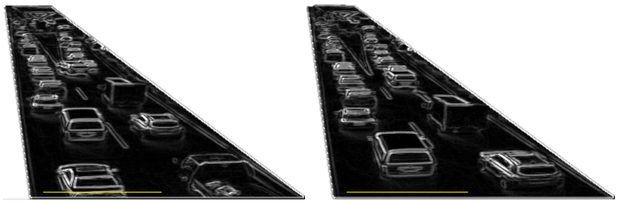


Fig. 3. Gradient map.

However, under some situations, there are false detections caused by noises, camera shake, or vehicles with the color close to road, and etc. As a result, we also propose a range feature and use it together with the gradient feature for reducing the false alarm rate. The proposed gradient feature describes the different change rate modes of pixel values of road and vehicles. Analogously, the range feature reflects the range of pixel values of them, which is defined as the difference between the maximum and the minimum pixel value on the detection line in each video frame.

Fig. 4 demonstrates the principle of the range feature. Fig. 4(a) and (b) shows respectively the first and last frame of a short clip from a test video. There are two vehicles passed the detection line during the period, as is labeled in red and blue boxes in the figure. Fig. 4(c) records the variation of the range feature in the process, whose X-axis represents frames. It should be noted that there are two crests in this bar graph representing the two vehicles in the red and the blue boxes in Fig. 4(b), respectively. Range values along the detection line look quite small when there is no vehicle passing. On the other hand, the two crests are consist of range values of vehicles, which is much larger than road's. So, the proposed range feature is helpful for decreasing false alarms in vehicle detection and counting.

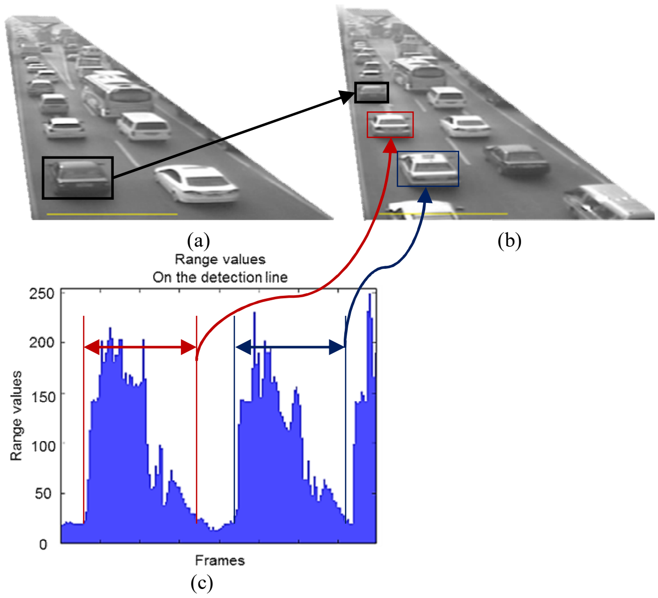


Fig. 4. The range values of a short sequence. (a) The start frame of the sequence. (b) The end frame. (c) Bar graph of the range values in each frame.

For further processing of the gradient feature, binarization of gradients is applied for reducing smaller gradient values. In the next sub-section, the integration of gradient feature and range feature in vehicle detection and counting is proposed. And the algorithm is given as a pseudo-code.

B. Algorithm

The vehicle detection and counting algorithm is given in pseudo-code 1, and for illustrating our approach more clearly, the flow chart of this pseudo-code is shown in Fig. 5, which is more visual.

pseudo-code 1 vehicle detection and counting algorithm

Input :

G: Array of the binarized gradient values on detection line in each frame.

R: Range value on detection line in each frame.

Delay: The threshold used for improving accuracy. In this algorithm, it is set to 5.

T: The threshold used for judging whether a range value represents a vehicle or not. It is set to 70.

Output:

Count: Vehicle counter

```
State ← "Road"
D ← Delay
Count ← 0
while read frames
   do if State == "Road" && sum(G) != 0 then
        D ← D – 1
        if D == 0 then
          State ← "Vehicle"
          D ← Delay
   else if State == "Vehicle" && R <= T && sum(G) == 0 then
        D ← D – 1
          if D == 0 then
             State ← "Road"
             Count ← Count + 1    //vehicle detected
             D ← Delay
   else
        D ← Delay
return Count
```

The constant ***Delay*** in this Algorithm is used for improving accuracy of vehicle detection. It is used to guarantee that only if the sum of gradient (binarized) values on the detection line is zero in ***Delay*** (e.g. 5) continuous frames, can it be confirmed that there is a vehicle passing the detection line. Similarly, only if the passing vehicle has not been detected in ***Delay*** continuous frames, can we consider that a vehicle has left the detection line.

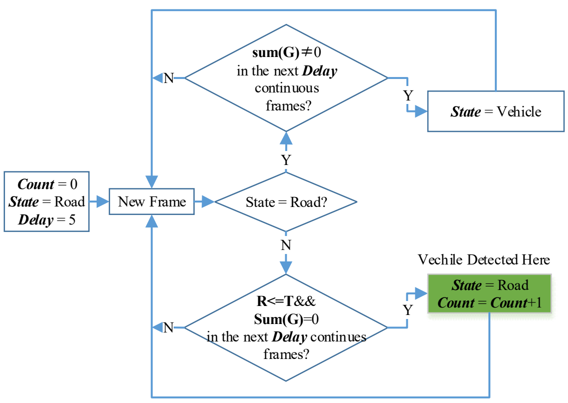


Fig. 5. Flow chart of our proposed vehicle detection and counting algorithm.

III. Occlusion Handling

Occlusion is a common problem in vision based vehicle detection algorithms. It connects two adjacent vehicles, which directly cause miss detections in many existing algorithms. In [12], a multilevel framework including intraframe, interframe, and tracking is developed for detecting and handling vehicle occlusion, which is effective, but not suitable for our vehicle detection method. So, we propose a new approach based on our vehicle detection algorithm in this section.

As a matter of fact, we barely find occlusions between small vehicles at the bottom of each frame in practical observations, for the reason that gaps between vehicles are maximum there. But occlusion caused by large vehicles like trucks, buses, and vans happens frequently, as shown in Fig. 6. So we can handle occlusions according to the change of vehicles' shape. However, in a video sequence with a top-down perspective, we can hardly get the height of an object, which leads us to handle occlusions through analyzing the width of each vehicle.

For every possibly detected vehicle, we will record its width from the first time it reaches the detection line till it finally leaves it. These width values are used for occlusion detection, as shown in Fig. 7. Fig. 7(a), (b), (e) and (f) show respectively four sample frames from two short clips of a video sequence. Fig. 7 (c) and (d) represent the change of width values when occlusion happens. As the orange and red boxes show, the width value jumps when a vehicle is closely followed by a large one, which can be a reliable feature for occlusion detection. Our occlusion detection algorithm is shown in Fig. 8, and the variable ***count*** is a reference of it in pseudo-code 1.

IV. Experimental Results

A. Testing Conditions

To evaluate the performance of our proposed approach, we have carried out extensive experiments in different weathers and conditions. The tests are carried out on a Windows 8 platform with an Intel Core i5 CPU (Central Processing Unit), and 4GB RAM (Random Access Memory). The image processing library we used is OpenCV. All of the video sequences we used are from real world scenes including city roads, high ways and tunnels. The frame rate of all video sequences is at 25 fps (frames per second). There are two frame sizes, specifically 720x288 and 640*360, with the latter corresponding to the sequence from night time. Our method is tested in five kinds of circumstances: 1) sunny day; 2) cloudy day; 3) rainy day; 4) tunnel; and 5) night time.

Fig. 6. Observed occlusions

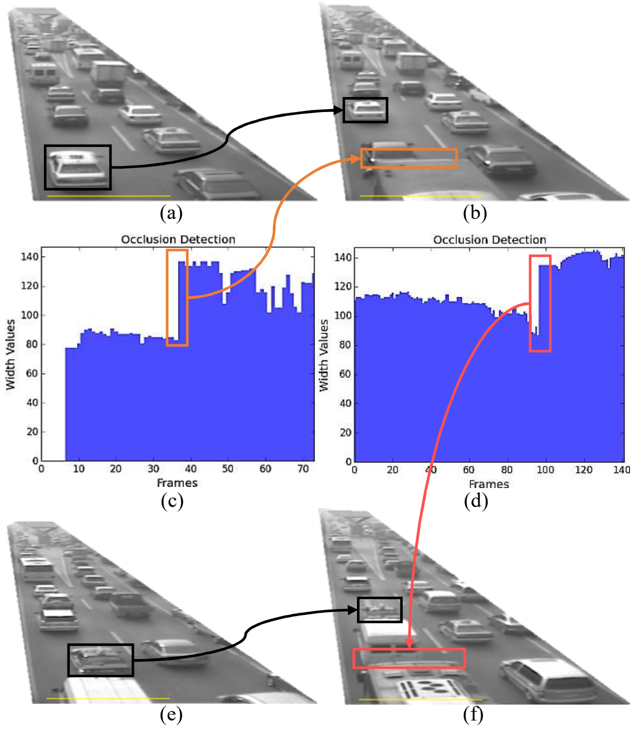


Fig. 7 The change of width value when occlusion happens. (a) The first frame of a video sequences. (b) The last frame of a video sequences, related to (a). (c) Bar graph of width values in the frames between (a) and (b). (d) Width values in the frames between (e) and (f). (e) The first frame of a video sequences. (f) The last frame of a video sequences, related to (e).

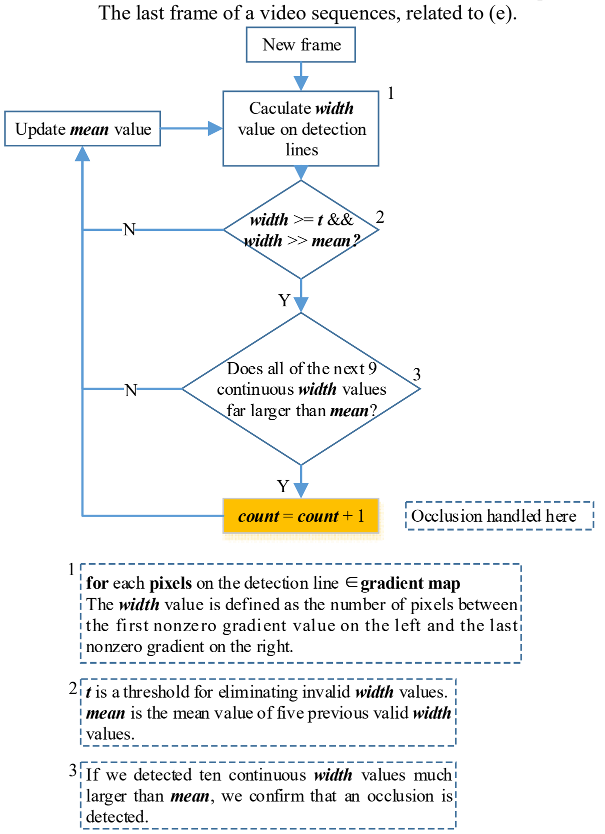


Fig. 8. Occlusion handling algorithm.

Fig. 9 shows the sample frames from each of the test scenes. Fig. 9(a) shows a scene in a sunny day when vehicles are moving with heavy and sharp shadow effects. Fig. 9(b) shows a scene in a cloudy day with very crowded traffic load. Fig. 9(c) shows a scene in a rainy day. There is obvious reflection of vehicles on road, which is caused by the rain drops in this situation. In Fig 9(d), it is a tunnel with very complicated illumination and fast moving vehicles. Fig 9(e) shows a nighttime scene with complex and weak illumination and dark background.

B. *Test results*

Table I presents the overall detection results of our approach under the aforementioned test conditions. In the table, the first column lists the test conditions, while each row shows respectively total number of vehicles and corresponding detection results. This table mainly gives some significant items in vehicle detection such as actual and detected vehicle number, correct detection number, false and miss detection number and other statistics. The ground truth data is counted manually. The detection rate is calculated with the following equation:

$$R_{real} = \frac{N_{cd}}{N_{real}} \tag{1}$$

Where R_{real}, N_{cd} and N_{real} are defined as the detection rate, correct detection number and actual vehicle number, respectively. It can be seen that our approach works reliably in different situations and weathers with very low error rate. The impact of different weather on our approach is quite limited. In sunny day, it is the heavy shadow effects that lead to the mild degradation of accuracy of vehicle detection. Meanwhile, in a tunnel with such complex illumination as the one shown in Fig. 9 (d), our method also has satisfactory performance.

C. *Comparisons with Other Approaches*

To verify the performance of our approach, we also compare our proposed method with five state-of-the-art ones. Specifically, Wu et al [17] method which is the only one that works under different weather conditions as with our approach; Mithun et al method [13] and Bouvié et al method [7] for daylight time; Two methods proposed in [15] and [18] for night time. The methods in [15] and [18] for night time will not work in day light time, while the one in [13] will not work in night time.

For comparison purpose, we implemented the methods in [7], [15], [18], [13], and [17]. The parameters involved in these methods are fixed to the ones as specified in the corresponding papers.

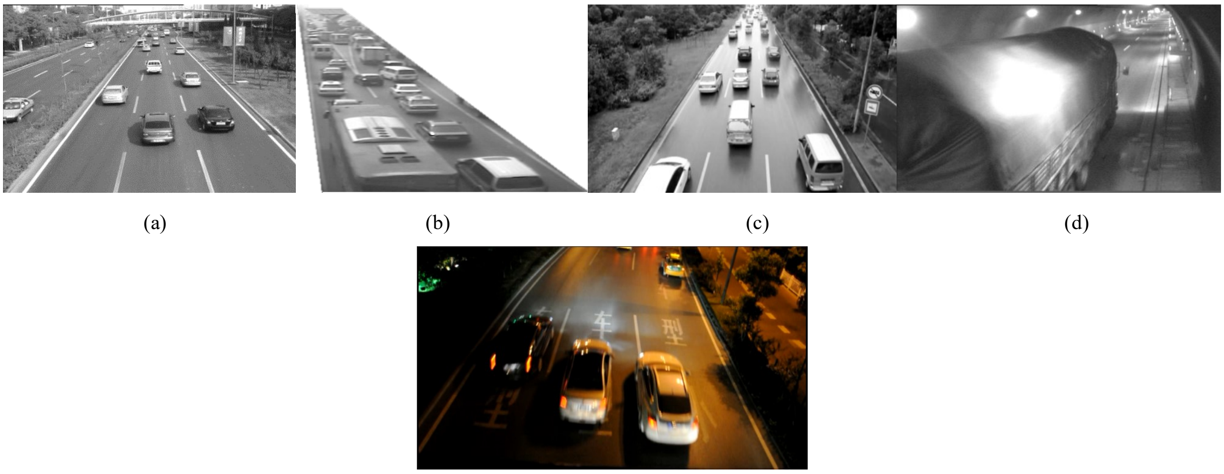
The comparison results are listed in Table II. In the table, the first column lists test conditions, and other columns list the vehicle counting accuracy rate of respective method. In the last row of the table, average accuracy rates of the methods are listed. The average is weighted by the total number of vehicles in a scene. Some approaches which are not applicable for certain scenarios are marked as N/A in this table.

TABLE I. EXPERIMENT RESULTS OF THE VEHICLE DETECTION

Scenarios	Total Real Number	Total Detection Number	Correct Detection Number	False Detection Number	False Detection Rate	Miss Detection Number	Miss Detection Rate	Real Detection Rate
Sunny Day	363	363	359	2	0.55%	2	0.55%	98.90%
Cloudy Day	235	233	233	0	0%	2	0.86%	99.15%
Rainy Day	169	169	167	1	0.59%	1	0.59%	98.82%
Tunnel	25	25	25	0	0%	0	0%	100%
Night Time	117	117	115	1	0.85%	1	0.85%	98.29%
Total	909	907	899	4	0.44%	6	0.66%	98.90%

TABLE II. COMPARISONS WITH OTHER APPROACHES' DETECTION RATE

Scenarios	Wu et al. [17]	Mithun et al.[13]	O'Malley et al.[15]	Sina et al. [18]	Bouvié et al. [7]	Our Proposed Method
Sunny Day	98.72%	93.15%	N/A	N/A	94.30%	98.90%
Cloudy Day	98.57%	95.52%	N/A	N/A	N/A	99.15%
Rainy Day	94.08%	N/A	N/A	N/A	N/A	98.82%
Night Time	91.45%	N/A	98.73%	88.20%	N/A	98.29%
Tunnel	N/A	N/A	N/A	N/A	N/A	100%
Weighted Average	96.83%	94.08%	98.73%	88.20%	94.30%	98.90%



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

(e)

Fig. 9. Test Video sequences in various conditions. (a) Sunny day. (b) Cloudy day. (c) Rainy day. (d) Tunnel. (e) Night time.

It can be seen that our proposed approach is the only one which can adapt to all of the test conditions including different weather, time, and locations. At night time, we achieve a satisfactory result as well as in daylight time using the same method. It is of great value that our approach can work very well in various conditions just with the modifications of some thresholds. As shown in the last row of Table II, our method achieves the best results when considering the total detection rate, which owes to the employment of the steady and excellent feature in our approach under different conditions.

D. Test on Public Database

To further verify our proposed approach's superiority, we also test it on a publicly available database, i.e. the RTM database [19]. We use M-30 (7520 frames, 800x480 @30fps) and M-30-HD (9390 frames, 1200x720 @30fps) to test our algorithm, which are respectively recorded in a sunny day and a cloudy day, as shown in Fig. 10. Then our method is compared with the database author's on these video sequences.

Table III gives the results of our approach on these video sequences. This time we only compare with the method in [19] which is developed by the RTM database's author. From the table, we can see that our proposed method obtains very accurate detection results for the sequences tested. The error rate of our method is about 0.39% and 1.27%. While [19] gives so many false alarms with the error rate of 13.3% and 44.25%. Meanwhile, the results have demonstrated the remarkable performance of our approach under different weather conditions.

(a) (b)

Fig. 10. The RTM database. (a) M-30 (sunny day). (b) M-30-HD (cloudy day)

TABLE III. COMPARISON WITH METHOD IN [19] WITH A PUBLIC DATABASE.

Sequences	**Ground Truth**	**Our Proposed Method**	**Guerrero et al. [19]**
M-30	256	255	290
M-30-HD	235	238	339

V. CONCLUSIONS

We have presented an approach for vehicle detection and counting in this paper. For vehicle detection, this paper has developed an algorithm using gradient and range features. Meanwhile, the solution for occlusions in this method has also been proposed. Our proposed method performs excellently in traffic jams and various conditions, such as sunny, cloudy, and rainy days, or even in tunnel with complex illumination. The high accuracy rate of our method has been verified based on the experiment results. In addition, our system can be easily set up by drawing a virtual detention line on each lane and set up some thresholds. In future works, we will improve the accuracy of our method in night time, and a new vehicle tracking and speed estimate algorithm is almost done using our vehicle detection algorithm.

VI. ACKNOWLEDGMENT

This work was supported by NSFC Grant No. 61401337, the 111 Project (B08038), Fundamental Research Funds for the Central Universities, and Key Laboratory of Infrared System Detection and Imaging Technology, Shanghai Institute of Technical Physics, Chinese Academy of Sciences.

REFERENCES

[1] S. C. Huang, C. C. Lin, & M. S. Wang, (2012, September). “A Vision-Based Vehicle Speed Warning System.” In Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on (pp. 832-834). IEEE.

[2] S. Sivaraman, & M. M. Trivedi, (2003). “Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis.” Intelligent Transportation Systems, IEEE Transactions on 14.4, 1773-1795.

[3] R. A. Hadi, G. Sulong, & L. E. George, (2014). “Vechile Detection and Tracking Techniques: A Concise Review.” Signal & Image Processing: An International Journal (SIPIJ), 2014, 5(1).

[4] Li, Shuguang, et al. "Video-based traffic data collection system for multiple vehicle types." IET Intelligent Transport Systems 8.2 (2013): 164-174

[5] F. Maali, (2012). “Vision-based car counting for multi-story carparks.” .U.S. Patent No. 8,339,454. Washington, DC: U.S. Patent and Trademark Office.

[6] K. V. Kumar, P. Chandrakant, S. Kumar, & K. J. Kushal, (2014, January). “Vehicle Speed Detection Using Corner Detection.” In Signal and Image Processing (ICSIP), 2014 Fifth International Conference on (pp. 253-258). IEEE.

[7] C. Bouvie, J. Scharcanski, P. Barcellos, & F. Lopes Escouto, (2013, May). “Tracking and counting vehicles in traffic video sequences using particle filtering.” In Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International (pp. 812-815). IEEE.

[8] R. Zhao, & X. Wang, (2013). “Counting Vehicles from Semantic Regions.” Intelligent Transportation Systems, IEEE Transactions on, 14(2), 1016-1022.

[9] V. K. Madasu, & M. Hanmandlu, (2010, June). “Estimation of vehicle speed by motion tracking on image sequences.” In Intelligent Vehicles Symposium (IV), 2010 IEEE (pp. 185-190). IEEE.

[10] T. Moranduzzo, & F. Melgani (2014). “Automatic Car Counting Method for Unmanned Aerial Vehicle Images.” Geoscience and Remote Sensing, IEEE Transactions on, 52(3), 1635-1647.

[11] N. K. Kanhere, & S. T. Birchfield, (2008). “Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features.” Intelligent Transportation Systems, IEEE Transactions on, 9(1), 148-160.

[12] W. Zhang, Q. J. Wu, X. Yang, & X. Fang, (2008). “Multilevel framework to detect and handle vehicle occlusion.” Intelligent Transportation Systems, IEEE Transactions on, 9(1), 161-174.

[13] N. C. Mithun, N. U. Rashid, & S. M. Rahman, (2012). “Detection and classification of vehicles from video using multiple time-spatial images.” Intelligent Transportation Systems, IEEE Transactions on, 13(3), 1215-1225.

[14] Yang, Mau-Tsuen, Rang-Kai Jhang, and Jia-Sheng Hou. “Traffic flow estimation and vehicle-type classification using vision-based spatial–temporal profile analysis.” IET Computer Vision 7.5 (2013): 394-404.

[15] R. O'Malley, E. Jones, & M. Glavin, (2010). “Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions.” Intelligent Transportation Systems, IEEE Transactions on, 11(2), 453-462.

[16] J. C Canny, (1986). “A computational approach to edge detection.” Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 679-698.

[17] B. F. Wu, & J. H. Juang, (2012). “Adaptive Vehicle Detector Approach for Complex Environments.” Intelligent Transportation Systems, IEEE Transactions on, 13(2), 817-827.

[18] I. Sina, A. Wibisono, A. Nurhadiyatna, B. Hardjono, W. Jatmiko, & P. Mursanto, (2013, September). “Vehicle counting and speed measurement using headlight detection.” In Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on (pp. 149-154). IEEE.

[19] Guerrero-Gomez-Olmedo, R. and Lopez-Sastre, R.~J. and Maldonado-Bascon, S. and Fernandez-Caballero, A. (2013) “Vehicle Tracking by Simultaneous Detection and Viewpoint Estimation.” In IWINAC 2013, Part II, LNCS 7931, 306-316.