An Accurate Vehicle Counting Approach Based on Block Background Modeling and Updating

Junfang Song
School of Information Engineering
Chang'an University
Tibet Nationalities Institute
Xi'an, Shaanxi, China

Huansheng Song
School of Information Engineering
Chang'an University
Xi'an, Shaanxi, China

Weixing Wang
School of Information Engineering
Chang'an University
Royal Institute of Technology
Xi'an, Shaanxi, China

Abstract—**Vehicle flow volume on the motorway in the urban area is successfully used to realize traffic decision-making and guidance. In this research, we introduced some critical components into a vehicle flow detection system. In order to enhance the accuracy and instantaneity of the vehicle counting in quick-change traffic conditions, an adaptive modeling and updating method of background was proposed. In the vehicle flow detection system, first a number of simplified and comprehensive pretreatments were used as preceding conditions, especially in detection region setting and calibration. Next, a novel background modeling and updating strategy was adopted in background subtraction. Then a location patching method was applied for vehicle recognition and counting. Experimental results show that the new approach can detect and count vehicle timely and accurately in a variety of traffic situations, meanwhile the accuracy rate of the counting is over 97%, which demonstrates this novel approach is robust and powerful enough.**

Keywords-vehicle flow detection; background modeling, background updating; background subtraction; location patching

I. INTRODUCTION

In recent years, video surveillance is an important technical in Intelligent Transportation Systems (ITS), it can provide rich information parameters of traffic for human understanding, such as vehicle speed，vehicle flow, average lane share, vehicle queue length and average vehicle spacing, which can provide a good guide for transportation planning and traffic control. Vehicle flow is the most important parameter, hence, research on the method of vehicle flow counting is essential to improve the accuracy rate and real-time performance.

Many different methods of vehicle flow counting have been proposed now, but the commonly used are inductive loop, sonar, microwave detectors and video-based [1]. Recently, with the development of computer vision technology, video-based method has attracted a lot of attention on account of its easy installation, operation and maintenance, also many related work have been done. Luis et al. [2] presented an adaptive segmentation strategy and a two-step tracking approach to detect and count vehicles, the two-step tracking approach combined the linear 2D Kalman filter [3] and 3D volume estimation using Markov Chain Monte Carlo (MCMC) process[4]. The method can adapt to the sudden illumination and weather variations very well. Nieto et al [5] introduced a vehicle tracking system based on 3D model estimation. It completed two prime functions including vehicle flow counting and vehicle classification by the process of background segmentation, blob extraction and 3D tracking. The proposed system can well adapt to a variety of traffic conditions. Roya et al.[6] first used Kalman filter[3][18] and background differencing techniques to complete multiple vehicles tracking, then utilized morphological operations for vehicles contour extraction and their recognition, finally, multiple vehicles classification were realized perfectly in complex mixed traffic.

Through the above methods, we notice that most of the traffic analysis was accomplished as following steps, moving target detection and feature tracking. One difficulty in this way is feature selection before tracking, because on the one hand we do not know which feature is suitable for distinguishing vehicles clearly from various moving targets, on the other hand, some features sometimes are easily vanished in a tracking process, so many scholars have made great efforts in this regard. Huang-Deng-Yuan et al. [7] designed a feature-based vehicle flow analysis and measurement system, which was constituted with moving object segmentation, background updating, feature extraction, vehicle tracking and classification. In terms of feature extraction, they took much effort and finally used aspect ratio as the feature to recognize vehicle. Although the statistical accuracy rate based on this feature reached on an average of cars and bikes of 96.9%, but some vehicles are prone to be false alarm. Hsieh et al. [8] proposed a vehicle classification method via vehicle size and linearity features, in view of this method also with the help of lane-dividing lines, vehicles were divided accurately into different categories like cars, buses, and trucks. Yet it is sleepy while vehicles move across the lane. Liu et al.[9]came up with a multi-scale Harris corner detection algorithm and a multi-resolution optical flow tracking algorithm based on wavelet to decompose the object displacement and match the object feature in all levels of the wavelet pyramid. This method can extract the corner feature of the moving vehicles and match the feature points accurately with a high real-time performance, no matter under what kind of scene. The only problem is that corner detection algorithm is relatively complex and costs large memory. In the mentioned articles, some chose corner point as a characteristic, the others selected size or aspect ratio as the feature, no matter which they all have their shortcomings in target tracking. Thinking of this, we try to complete the vehicle flow counting using location matching method combined with virtual line

Fundamental Research Special Project for the Central Universities (2014G3242005）

which can avoid the problem caused by feature tracking, the method has been put into practice in some literatures. Sun et al. [10] presented a statistical learning method for moving vehicle detection. In paper they detect the moving vehicle through the diversification of pixel intensities and local texture in designed virtual area. In this method, using the feature gained from many training samples to decide a pixel is background or target, it is not robust in all weather. Wang et al. [11] put forward a method to detect vehicles in detection area using template tag chain, which had 3 uniform distribution areas, respectively representing the front cover, car windshield and roof. The method could not detect vehicles accurately while the vehicles moved across one road to another. Hence, in our new method we select 2D boundaries to recognize vehicles in whole detection zone and then count them.

Moving target detection is the other difficulty, which is not only an essential part of vehicle flow counting, but also a foundation for vehicle analysis such as object tracking, classification and recognition. The algorithms used commonly in moving target detection are optical flow, background subtraction and frame difference [12-14]. In the traffic scene, background is stationary because of fixed traffic-camera site and angle, so background subtraction algorithm is used widespread in detecting motion vehicles due to its convenience, but owing to the illumination and weather influence, the background need to be updated timely. Background updating involves a crucial aspect as modeling. In general, we use the these models: Gaussian model [15-17], Kalman filtering [3][18], Counting model[19] etc, they are all based on pixels. In practice, it will sometimes appear a lot of isolated noise points using the above models caused by variance environment, which would result in bad background subtraction effect. So we propose a method using binary digital sequence block-based to build model, it can not only adapt to the complex background environment, but also is quick in building background model.

This paper is organized as follows: Section 2 describes the system structure. We introduce system pretreatments in Section 3 and moving target detection in Section 4 while vehicle counting is discussed in Section 5. Finally, experimental results and conclusion are given in Section 6 and Section 7.

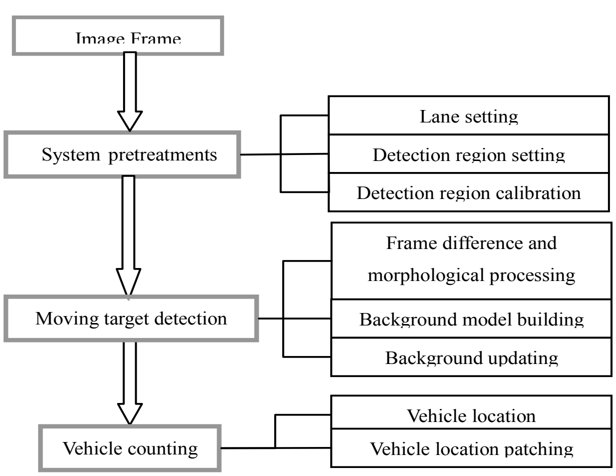


Fig. 1 System overview of the proposed approach

II. SYSTEM STRUCTURE

Fig. 1 is the system structure of the proposed approach. There are several modules in the flowchart. First, the system is set to provide premise ready for vehicle detection and traffic counting. Second, moving vehicles are detected from video sequences. Next, vehicle location is identified. Finally, vehicles are calculated based on location patching result.

III. SYSTEM PRETREATMENTS

In this system, pretreatments mainly contain lane setting, detection region setting and calibration. In the whole process, lane lines are indicated and read primarily in the first frame, then a detection region is set up and standardized on basis of it.

A. Lane Setting

On one hand, lane lines are the significant symbols in road which can separate the moving vehicles into several areas; On the other hand, lane lines are the crucial boundaries with them the detection region can be set，so lane setting is very important. In road scene, the lane lines are marked with white markings, hence, lane setting is a process to access the points on each white marking, and connect them into a straight line, which represents the lane line in image processing, meanwhile write down the pixel coordinates of all points. The results of lane setting for the scene are shown in Fig. 2.

Fig. 2 Lane letting results

B. Detection Region Setting

In order to reduce system load and improve the real-time performance, we extract a part in the scene as the detection region, where the image feature is researched. Usually we draw two horizontal lines to form a detection region combining with the boundary lane lines, in such a method, because the road is not in the same direction with the detection region, it will result in the vehicle location patching and counting difficult. In order to solve the problem, we set the two parallel lines vertical to actual road direction, which are shown in Fig. 3, they are denoted in green. In this system, they are automatically formed by clicking the location on image denoted with black points (3,4,5), the point 3 and 4 are equidistant of the white markings in each lane, point 5 is free. Distance between the two parallel lines depends on a special need in accordance with the different scenarios. After that, the detection region ACDB is finally formed.

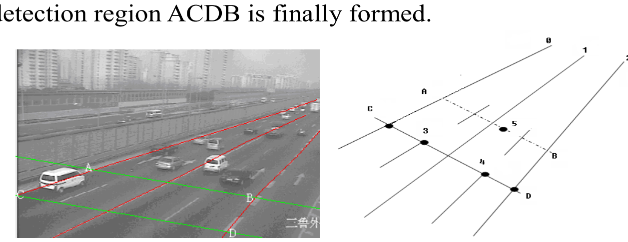


Fig. 3 Detection region setting result and schematic

C. Detection Region Calibration

Calibration is successive correspondences of the pixel points on mutual parallel lines in one direction from the detection region to a fixed rectangle region, the whole process is based on the unchangeable parallel performance of affine transformation [20][21][22]and linear interpolation, which can greatly improve the system efficiency. Fig. 4 shows the principle in detail, the mutual parallel lines are denoted with black solid lines, and the pixel points that need to be linear interpolation or not are separately denoted with green and red. Fig. 5 shows the calibrated result, using which the moving target detection in part 3 and vehicle counting in part 4 are implemented.

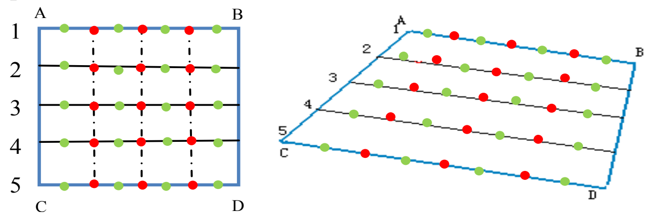


Fig. 4 Schematic of detection region calibration

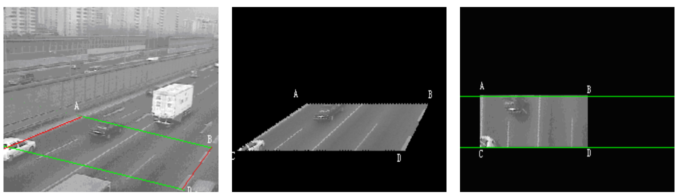


Fig. 5 Result of detection region calibration

IV. MOVING TARGET DETECTION

The goal of moving target detection is to find out the motion part and do threshold segmentation from a series of mixed scene. This paper mainly uses background subtraction algorithm to detect moving target, but when the initial background is not fully updated or updating fails, we replace it with frame differences algorithm. So we first split the moving targets through frame differences, and extract their binary results; next give a related digital sequence for each frame of binary video sequences, which tells us directly which part need to be updated during this time; then update the background image using an adaptive strategy and obtain a new background subtraction result.

A Frame Differences and Morphological Processing

Sometimes the objects (vehicles) are moving slowly in traffic, in order to avoid the fracture and broken target detection results while using frame differences algorithm, here the interval of frames we choose two. As shown in Fig. 6(a), 6(b), the result is a binary segmentation for frame difference image, in the image results, white pixel indicates that the point is on target motion area, while the black pixel indicates that the point is on background area. After analysis of these results, we notice that because of the adjacent pixel similarity in a moving object, some small empty holes are still appeared in binary result, which will bring out deviation in digital sequence counting. Considering the relationship of adjacent pixels, we do expansion for the binary segmentation result, Fig. 6 (c) 、(d) are the frame difference results with expansion

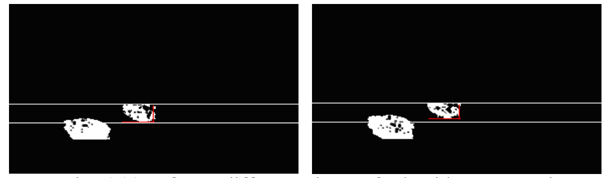


Fig. 6 (a) A frame difference image for k with no expansion.

Fig. 6 (b) A frame difference image for k+2 with no expansion.

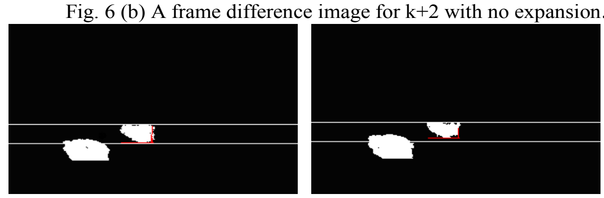


Fig. 6 (c) A frame difference image for k with expansion

Fig. 6 (d) A frame difference image for k with expansion

B Background Modeling

Using the binary segmentation result of part A, we build a digital sequence corresponding to it, and then extract the background feature and build the background model, Fig.7 shows the process in details. First, we divide the detection region into many same blocks as the broken lines fenced area in the figure, each block is composed of *3x2* pixels and denote with (r_i, col_j), each pixel point is denoted by 1 or 0, while it is white pixel, its value is 1, otherwise, its value is 0. Then for each block, using the following algorithm to complete filtering and finally decide it is a motion target block or background block，the algorithm is as follows :

- $f_0 = \frac{num_0}{num_0 + num_1}$, $f_1 = \frac{num_1}{num_0 + num_1}$ num_0 represent the number of *"0"* pixel point , num_1 represent the number of *"1"* pixel point. If $(f_0 \geq f_1)$, all the pixels value in the block are assigned to *"0"*, and formed a *0-block*. If $(f_0 \geq f_1)$, all the pixels value in the block are assigned to *"1"*, and formed a *1-block*. For example, in the block (*r2, col7*), f_0 =4/6, f_1 =2/6, so the pixels value in this block are assigned *to "0"*, the block is considered as a *0-block*.
- Attain the number of *1-block* in each column col_j and give the counter B_j , j is the column number, $j = 1,2,....,12,..N$. If ($B_j > T$), the column col_j is marked by *"1"* in R. If ($B_j < T$), the column col_j is marked by *"0"* in R. T is a threshold value, here it is 2.

R is a digital sequence formed finally on the horizontal direction as shown in Fig.7, then analyze the sequence R, *0-col* represents there is no motion or a little motion. Count the number of continuous *0-col* and remark it with N_0 , if N_0 reaches a certain value, we consider that the image marked with continuous *0-col* is new background and later update it into the background model using a suitable formula.

C Background Updating

To avoid the edge effects while replacing the background model with new background，people usually used weight rules to attain it, but we update the background according to the following model：

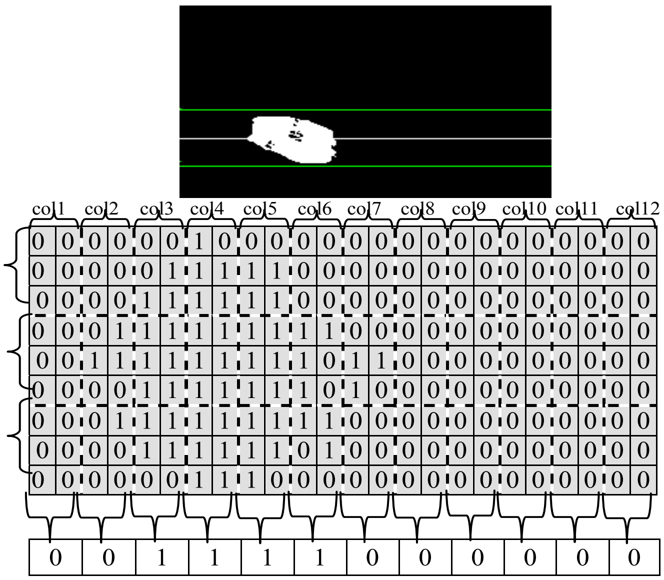


Fig.7 Details of forming digital sequence between two green lines

$$B(x,y)=\begin{cases}B_0(x,y)\times(1-\alpha)+F(x,y)\times\alpha, & N_0>T_B \\ B_0(x,y), & others\end{cases} \quad x\in[L,R] \quad (1)$$

When $N_0 > T_B$, record the starting block L and termination block R, and update the grayscale sequence $F(x,y)$ between them into background $B_0(x,y)$ on the basis of ratio α, then form a new one $B(x,y)$. Others, if there is no 0-col or $N_0 < T_B$, background $B_0(x,y)$ keep constant. Weight α is a free variable, it directly affect the speed of background updating, α greater, the speed of background updating slower, contrary, the sooner. Also the threshold T_B selection is very important, it can not be a random value，but a set one according to the lane and vehicle width, in here we choose 1/3 of the lane width.

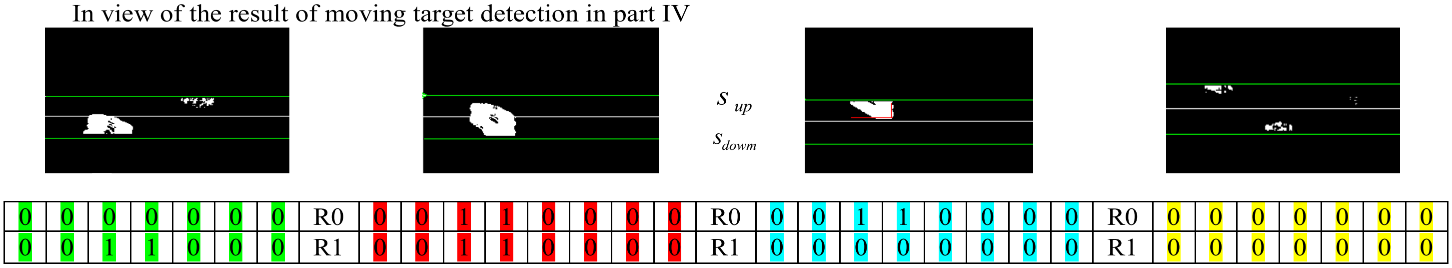
V. VEHICLE COUNTING

In view of the result of moving target detection in part IV and analysis of the whole process from a moving vehicle enters the detection region to leaves away from it, we consider to select 2D projection boundaries (rear and right boundary features of a vehicle) for determining a vehicle location and use an appropriate matching criterion to carry on clustering for objective locations, finally we can work out the vehicle numbers.

A Vehicle Pattern Identification

An important innovation of this new method in vehicle counting is that we select 2D projection boundaries for determining a vehicle pattern. In binary segmentation result, each Target Region (*TR*) had been represented with white color, which can be approximately regarded as a rectangular block. According to the projection principle, the down, right boundary of the rectangle is respective correspondence of the tail, right side of a vehicle. If we can get the two important parameters in every binary segmentation result, the vehicles can be extracted accurately.

In this part we first divide the detection region into two parts（s_{up} and s_{dowm}）, which are separated by a horizontal white line in Fig.8. When a vehicle moves from s_{dowm} to s_{up} , Its state can be divided into four categories, all of them can be indicated by a combination of binary code 0 and 1, which respectively represents no-motion part and motion part in s_{up} or s_{dowm}. As shown in Fig.8,（a）is a state that the vehicle is about to enter s_{up} , it has motion part in s_{dowm} but has not in s_{up}, we represent it with (0,1); (b) is a state that partial vehicle has entered s_{up}, it has motion part both in s_{dowm} and s_{up}, we represent it with (1,1);（c）is a state that the full vehicle has entered s_{up} , it has motion part in s_{up} but has not in s_{dowm}, we represent it with (1,0); (d) is a state that the vehicle leaves away s_{up} but no vehicle enters s_{dowm}, it has no-motion part both in s_{dowm} and s_{up} , we represent it with (0,0). After such a classification, we observe that only when a vehicle is in the (c) state, the two important boundaries can be obtained fully, so we can take it as a vehicle location.



（a）（0，1） (b)（1，1） (c)（1，0） (d) （0，0）

Fig.8. Digital sequences of each vehicle moving state

The next we transform each state above into two digital sequences, which facilitate the location analysis and further identification. The similar work is done for both s_{up} and s_{dowm} of every frame in getting the digital sequence according to the steps of background model building in part IV. After that, there forms two sequences called *R0* and *R1* as shown in Fig.8. From the two sequences, we can analyze the vehicle location in a digital way easily and then determine a final vehicle location with the following conditions:

$$L_{TR}\geq T_C \quad \&\& \quad L_{SR}\geq T_D \qquad (2)$$

In the digital sequence, while 1-col appears in R0 and 0-col appears in R1 in a same column, we call it *Col-TR*. L_{TR} is the number of continuous *Col-TR* in the digital sequence, next to whose right, if there is *0-col* in *R0*, counting the number of continuous 0-col and remarked with L_{SR}. When

$L_{TR} > T_C$, and meanwhile $L_{SR} > T_D$, we obtain the vertical vector projection and mark them with short red lines as shown in Fig 9(a),(b),(c), this moment a vehicle location is formed. The threshold value T_C is set in the light of the width of vehicle, we often choose 2/3 of minimum width car (here, T_C =9). T_D is the other threshold value (here, T_D =2).

B Vehicle Location Patching

Vehicle matching means to search the same vehicle from two or more frame images in the scene at different times, in this process，relevant measure is critical to decide the accuracy of matching. The relevant measure is to find the objective location as registration result by selecting an optimal matching criterion among the objective range [23]. Under normal condition, the vehicle always goes left upwards or right upwards, and its speed is limited, meanwhile we notice that the horizontal position of the same vehicle between the present frame and the former one changes little. So we can match the vehicle and analyze the matching result on the basis of follow-up rules, afterwards calculate the vehicle numbers.

Matching criterion:

a）center of objective location should be at top of matching location in portrait.

b）objective location makes the function value smallest.

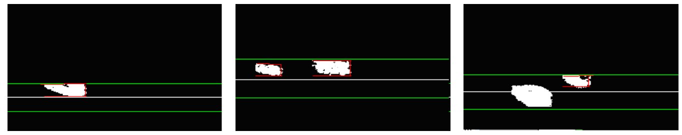
Centre distance between the matching location and objective location is often selected as a proper registration rule to measure the degree of matching. The distance measure methods[24][25] consist of Minimal absolute distance, Minimum mean square error, Minimal mean absolute distance, Mahalanobis distance, Hausdorff distance, and so on. Here, the minimal mean absolute distance is selected to work out minimal absolute distance between the matching location and all of the objective locations.

VI. EXPERIMENTAL RESULTS

The testing is divided into two phases. In first phase, we used large number of date to debug and improve its ability to detect and count the vehicle. This phase allow us to improve the method for validating the accuracy of detection result under various conditions, such as multi-lane vehicle moving, vehicle crossing lane to move and vehicle mutual covered situation. All date are tested on Windows 7 platform with a Pentium 4 3.2-GHz central processing unit (CPU) and a 2-GB random access memory (RAM).The proposed system is implemented with Visual C++ on raw video format, and the frame frequency is 25f/s, for each frame, it has 720×288 pixels, 256 gray levels.

In second phase, the on-line system was tested on several roads in cities of Fuzhou, Kunming, Shanghai and Chongqing in China to see whether the system could detect and count vehicles accurately in multiple scenarios, which contains highways and urban roads. We had completed the surveillance testing with a network transmission from the remote cameras, they were connected as the example shown in Fig.10. The modules of background building, background updating, vehicle detection, location patching and counting algorithms are placed on the video processors, while the detection result and relevant data are saved on the host PC. The video processor is composed of 20DSPs, each of which processes the data collected by one camera. Meanwhile, the department

three-dimensional boundaries of this part by the horizontal and



（a） (b) （c）

Fig .9 Vehicle location identification results

of traffic management could monitor real-time traffic flow through the Ethernet.

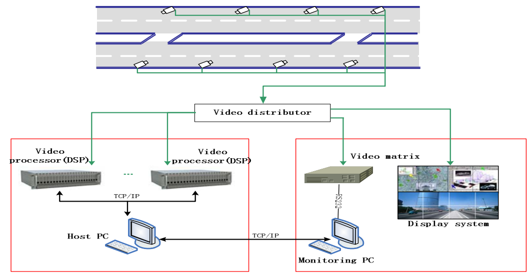


Fig.10 The on-line system structure

At the first testing phase, we tested a set of sequences with various traffic conditions for 25 minutes. Three detection results are demonstrated about typical traffic conditions in Fig.11, also the stage results of background updating are shown clearly. Fig.11 (a) is the detection result on multi-lane vehicles moving, Fig.11 (b) is another special situation on the vehicle moving across lane, both of them are accurate. Then we find sometimes it is inaccurate when vehicles are mutual covered just because adhesions are occurred as shown in Fig.11(c). The results of first phase show that the new method could detect and count vehicles timely and accurately in addition to the special situation when vehicles were mutual covered seriously.

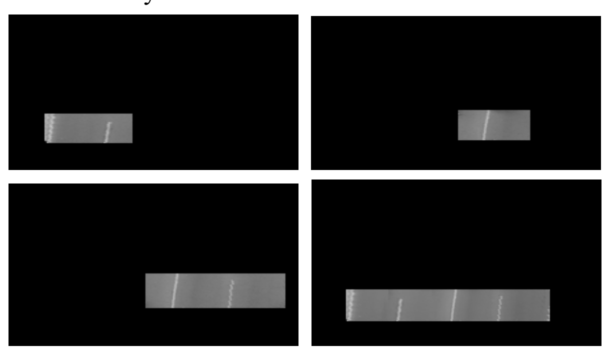
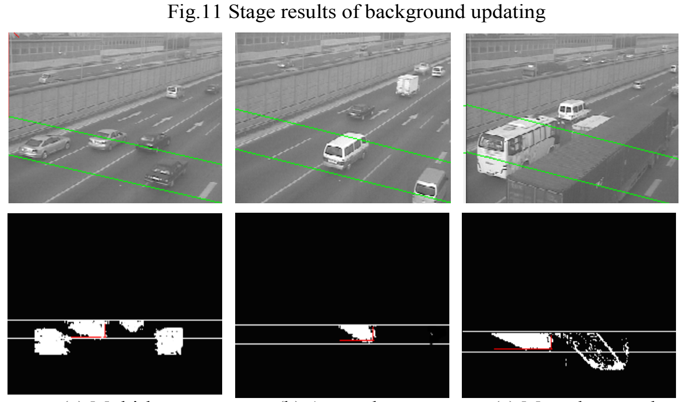


Fig.11 Stage results of background updating



(a) Multi-lane (b) Across lane (c) Mutual covered

Fig.12 Detection results of different traffic situations

The second phase of testing we focused on highway and urban roads of four big cities with large traffic, and having a variety of traffic situations. During the approximate three-week test period, we tested 52 hours data with different conditions and scenarios. Table 1 shows the numerical results of vehicle counting for four scenarios. We compared the number of automatically detected result to human visual counted for the same video sequence. Based on the results in Table 1, we note that the average detection rate can reach over 97% using the new method.

TABLE I VEHICLE TEST RESULTS COMPARISION

Test Scenario	The Proposed Method	Manual Method	Deviation	Accuracy Rate
Video1	4021	3931	90	97.76%
Vodeo2	3220	3165	55	98.24%
Vodeo3	3157	3107	50	98.43%
Vodeo4	4078	3972	106	97.46%

VII. CONCLUSION

A real-time and intelligent vehicle flow detection method has been presented in this paper, especially the system pretreatments before vehicle detection, which have reduced the system load and simplify the whole algorithm. Meanwhile the novel background modeling and updating method used have eliminated the influence of illumination changes on the background environment and ensured the detection accuracy of the 2D projection boundary. Finally, the location patching strategy have provided convenience for the vehicle identification and counting. In off-line and on-line system testing phases, the suggested method not only has high detection rate but also can count vehicles timely. The method has been applied in the traffic surveillance system on some highways and urban roads of big cities during the day in China. In next step, we still need to improve the algorithm for resolving vehicle flow detection in night, and further study the vehicle flow detection method in traffic jam and chaotic traffic scenes.

ACKNOWLEDGEMENT

The new methods in this paper are supported by“863” Research Subject (2012AA112312), the Science Project of Traffic Science and Technology (201231849A70) and Fundamental Research Special Project for the Central Universities (2014G3242005）

REFERENCES

[1] JunFang S, ANing B, Ru X, “A reliable counting vehicles method in traffic flow monitoring.” Image and signal Processing, 4th International Congress on IEEE, 2011, 1: 522-524.

[2] Unzueta Luis, Marcos Nieto, Andoni Cortés, Javier Barandiaran, Oihana Otaegui, and Pedro Sánchez, “Adaptive Multicue Background Subtraction for Robust Vehicle Counting and Classification,” Intelligent Transportation Systems, IEEE Transactions on 13, no. 2, 2012: 527-540.

[3] McCall, Joel C, Mohan M. Trivedi, “Video-based lane estimation and tracking for driver assistance, survey, system, and evaluation.”Intelligent Transportation Systems, IEEE Transactions on 7, no. 1, 2006: 20-37.

[4] N. K. Kanhere, S. J. Pundlik, and S. T. Birch field, “Vehicle segmentation and tracking from a low-angle off-axis camera,” in Proc. IEEE CVPR, 2005, pp. 1152–1157.

[5] Nieto Marcos, Luis Unzueta, Javier Barandiaran, et al, “Vehicle tracking and classification in challenging scenarios via slice sampling.” EURASIP Journal on Advances in Signal Processing, no. 1, 2011: 1-17.

[6] Roya and Mansour Jamzad, “Real time classification and tracking of multiple vehicles in highways.” Location Recognition Letters 26, no. 10, 2005: 1597-1607.

[7] Huangdeng Y, Chaoho C, Yufeng L, et al, “Feature-based vehicle flow analysis and measurement for a real-time traffic surveillance system.” Journal of Information Hiding and Multimedia Signal Processing 3, no. 3, 2012: 279-294.

[8] Hsieh J W, Yu S H, Chen Y S, et al. “Automatic traffic surveillance system for vehicle tracking and classification.” Intelligent Transportation Systems, IEEE Transactions on, 2006, 7(2): 175-187.

[9] Liu M, Wu C, Zhang Y, “Multi-resolution optical flow tracking algorithm based on multi-scale Harris corner points feature.” Control and Decision Conference on IEEE, 2008: 5287-5291.

[10] Mingxia Sun, Kunfeng Wang, Ming Tang, “Video vehicle detection through multiple background-based features and statistical learning.” Intelligent Transportation Systems, 14th International Conference on IEEE, 2011:1337-1342.

[11] Huibin W, Fan M, Qinwu L, et al. “New Method of Traffic Flow Video Detection Based on Combining of Virtual Coil and Templates Chain.” Microcomputer Applications, 2010, 11: 005.

[12] Zhenxue C, Guoyou W, Chenyun L, “Vehicle flow detection statistic algorithm based on optical flow.” Signal Processing and Information Technology, 5th International Symposium on IEEE, 2005: 264-268.

[13] A. Ajmal, I. M. Hussain, “A simple and novel method for vehicle detection and classification.” Applied Sciences and Technology, 9th International Bhurban Conference on IEEE, 2012:58-63.

[14] Lunhui Xu, Wenping Bu, “Traffic flow detection method based on fusion of frames differencing and background differencing.” In Mechanic Automation and Control Engineering (MACE), Second International Conference on IEEE, 2011, pp. 1847-1850.

[15] Wang G, Xiao D, “Background updating technique in complex traffic scene based on sensor fusion.”Journal of Transportation Systems Engineering and Information Technology, 2010, 10(4): 27-32.

[16] Bouttefroy P L M, Bouzerdoum A, Phung S L, et al. “On the analysis of background subtraction techniques using Gaussian mixture models.” Acoustics Speech and Signal Processing (ICASSP), International Conference on. IEEE, 2010: 4042-4045.

[17] Xiaoping W, Lijie Z, “Improved method of moving objects detection based on single-Gaussian background model”, Computer Engineering and Applications, July 2009, vol.45, n.21, pp 118-120.

[18] Scott J, Pusateri M A, Cornish D, “Kalman filter based video background estimation,” Applied Imagery Location Recognition Workshop. IEEE, 2009: 1-7.

[19] Xiaofei L, Zhonghui M, “An Algorithm of Background Extraction Based on Counting of Histogram Combining with Multi-Frame Average.” Journal of Nanjing University of Posts and Telecommunications, December 2008, vol 28:74-77.

[20] Tsang P W M, Yuen T Y F, Situ W C, “Enhanced affine invariant matching of broken boundaries based on particle swarm optimization and the dynamic migrant principle.” Applied Soft Computing, 2010, 10(2): 432-438.

[21] Gao X, Zhang C, Huang Y, et al, “A robust high-capacity affine-transformation-invariant scheme for watermarking 3D geometric models.” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 2012, 8(2S): 34.

[22] Peng L I, “Method for Vehicle Speed Detection Based on Camera Calibration.” Electronic Science and Technology, 2011, 7-24.

[23] Shumway R H, Stoffer D S, “Time series analysis and its applications.” Springer Science, Business Media, 2010.

[24] Liwei W, Yan Z, Jufu F, “On the Euclidean distance of images.” Location Analysis and Machine Intelligence, IEEE Transactions on 27, 8, 2005: 1334-1339.

[25] Aiazzi Bruno, Luciano Alparone, Stefano Baronti, “Quality issues for compression of hyper spectral imagery through spectrally adaptive DPCM.” Satellite Data Compression, Springer New York, 2011:115-147.