Journal of Marine Science and Technology

Volume 21 | Issue 6 Article 2

VEHICLE COUNTING WITHOUT BACKGROUND MODELING

Cheng-Chang Lien
Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan, R.O.C., cclien@chu.edu.tw

Cheng-Ta Hsieh
Department of Computer Science and Information Engineering, National Central University, Taiwan, R.O.C

Ming-Hsiu Tsai
Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan, R.O.C.

Recommended Citation
Lien, Cheng-Chang; Hsieh, Cheng-Ta; and Tsai, Ming-Hsiu (2013) "VEHICLE COUNTING WITHOUT BACKGROUND MODELING," *Journal of Marine Science and Technology*: Vol. 21: Iss. 6, Article 2.
DOI: 10.6119/JMST-012-0829-2
Available at: https://jmstt.ntou.edu.tw/journal/vol21/iss6/2

Journal of Marine Science and Technology, Vol. 21, No. 6, pp. 631-638 (2013)
DOI: 10.6119/JMST-012-0829-2

VEHICLE COUNTING WITHOUT BACKGROUND MODELING

Cheng-Chang Lien[1], Cheng-Ta Hsieh[2], and Ming-Hsiu Tsai[1]

Key words: vehicle detection/tracking, dual foregrounds fusion, texture-based target tracking, traffic flow.

ABSTRACT

In general, the vision-based methods for vehicle detection/tracking may face the problems of illumination variation, shadows, or swaying trees. In this study, we propose a novel vehicle detection method without background modeling to overcome the aforementioned problems. First, a modified block-based frame differential method is established to quickly detect the moving vehicles without the influences of rapid illumination variations. Second, the precise vehicles' regions are extracted with the dual foregrounds fusion method. Third, a texture-based object segmentation method is proposed to segment each vehicle from the merged foreground image blob and remove the shadows. Fourth, based on the concept of motion entropy a false foreground filtering method is developed to remove the false object regions caused by the swaying trees or moving clouds. Finally, the texture-based target tracking method is proposed to track each detected target and then apply the virtual-loop detector to compute the traffic flow. Experimental results show that our proposed system can work with the computing rate above 20 fps and the average accuracy of vehicle counting can approach 86%.

I. INTRODUCTION

Recently, many vision-based researches addressed on the vehicle or human detections [4, 8, 9, 13, 20] were proposed. In general, the moving objects could be detected by three kinds of methods: motion-based [4], background modeling [9, 13, 20], and temporal difference [8] approaches. In the motion-based approaches, the optical flow method [4] utilizes the motion flow segmentation to separate the background and foreground regions. By applying the optical flow method [4], the moving objects can be extracted even in the presence of camera motion. However, the high computation complexity makes the real-time implementation difficult.

For the background modeling methods, the construction and updating of background models [9, 13, 20] is time-consuming. For example, in [9, 13], the Gaussian Mixture Model (GMM) is frequently adopted to model the intensity variation for each pixel within a time interval and then high computing cost is required to calculate the GMM parameters. Furthermore, the foreground detection with background modeling method is extremely sensitive to the rapid illumination variation or the dynamic background changing. In [20], the Kalman filter is used to update the background model with less computational complexity. However, this method can't solve the problem of serious scene change which can make the system unable to update the background model accurately.

Instead of establishing the background model, the temporal difference methods subtract two successive frames and detect the scene change introduced by the moving objects. The advantage of this method is less susceptible to the scene change, i.e., it has capability to detect the moving objects in dynamic environments. For example, in [8], the temporal difference method is resilient to the dynamic illumination variation, but the regions of the moving objects can't be extracted completely when the objects move slowly.

Here, we propose a novel vehicle detection method without background modeling to overcome the aforementioned problems. First, a modified block-based frame differential method is established to quickly detect the moving targets without the influences of rapid illumination changes. Second, the precise targets' regions are extracted by the dual foregrounds fusion method. Third, a texture-based object segmentation method is proposed to segment each vehicle from the merged foreground image blob and remove the shadows. Fourth, a false foreground filtering method is developed based on the concept of motion entropy to remove the false object regions caused by the swaying trees or moving clouds. Finally, the texture-based target tracking method is proposed to track each detected target and then apply the virtual-loop detector to compute the traffic flow.

The information of vehicle counting may be obtained from several kinds of vision-based sensor [1, 3, 11, 15, 16]. However, these methods are difficult to judge whether the vehicle

Paper submitted 11/29/11; revised 06/04/12 ; accepted 08/29/12. Author for correspondence: Cheng-Chang Lien (e-mail: cclien@chu.edu.tw).
[1] Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan, R.O.C.
[2] Department of Computer Science and Information Engineering, National Central University, Taiwan, R.O.C.

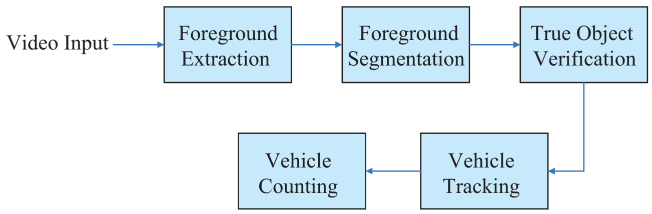


Fig. 1. The block diagram of the traffic flow analysis without background modeling.

appears in the virtual detector region or not under different lighting conditions. Furthermore, the lack of tracking information will make the vehicle counting inaccurate. Here, a two-line vehicle crossing scheme is applied to count the moving vehicles. The system block diagram is shown in Fig. 1. Our system consists of the following modules: vehicle extraction (foreground detection with dual foregrounds, foreground segmentation, and true object verification), vehicle tracking (Kalman filter tracking), and vehicle counting.

II. VEHICLE DETECTION WITHOUT BACKGROUND MODELING

In general, vehicle detection can't be accurate on the light variation or cluster scenes. In this section, we propose a novel vehicle detection method without the background modeling to segment the vehicles on the light variation or cluster scenes. Furthermore, the motion-based false object filtering method is proposed to remove the false vehicle detections.

1. Adaptive Block-Based Foreground Detection

Traditional pixel-based frame difference method can introduce many fragmented foreground regions because the spatial relationships among neighboring pixels are not considered. Hence, if we detect the moving objects using the frame difference method, both the local and global properties should be considered. Thus, we proposed the block-based foreground detection method to extract more complete foregrounds. First, each frame is divided into the non-overlapped blocks. The block size can be adjusted according to the object size. The foreground detection algorithm is described as the following steps.

1. The RGB color space is transformed into the YC_rC_b color space and the moving objects in the Y, C_r, and C_b channels are detected separately.
2. In each image block, if the number of detected foreground pixels exceeds a specified threshold, then we categorize this block into the foreground block.
3. By fusing the foreground regions detected from the Y, C_r, and C_b channels with the rule-based method, we can obtain a more complete object region.

The reason why we adopt the rule-based method is that Y, C_r, and C_b channels have different property in foreground

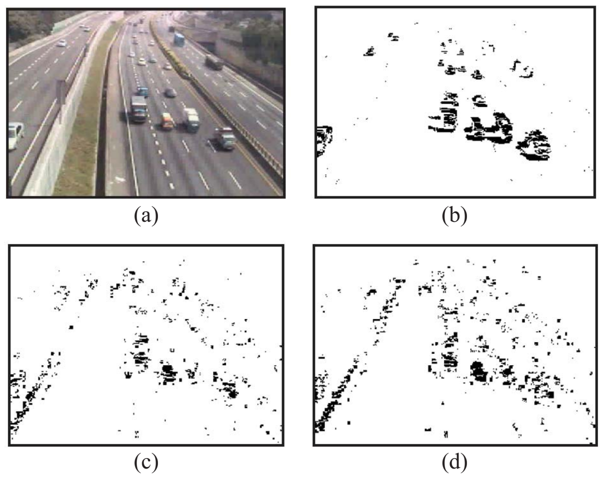


Fig. 2. The original image is shown in (a) and the detected foregrounds for Y , C_r and C_b channels are shown in (b), (c), and (d) respectively.

detection. In Fig. 2(b), the objects are detected with grayscale variation in the Y channel; while in Figs. 2(c) and 2(d), the objects are detected with the color variations in the C_r and C_b channels. The relationship between grayscale and color information are complementary.

If the region changes in the Y channel but doesn't change in the C_r and C_b channels, then it may be the object shadow. On the contrary, the region changes in the C_r and C_b channels but doesn't change in the Y channel, and then this region may be a noise region. Hence, in our system, the object detection with both the grayscale and color channels is established. We adopted the rule-based method to determine whether the pixel/block belongs to foreground or not, which is described as:

1. If the difference value in the Y channel exceed 1.5 times threshold of the Y channel ($1.5T_Y$) for a image block, we classify this block into foreground directly.
2. Otherwise, the object is detected in the Y, C_r, and C_b channels with the rule: $(Y > T_Y)\&\&((C_r > T_{Cr})||(C_b > T_{Cb}))$. If the pixel/block changes in the grayscale and color channels simultaneously, then we classify this pixel/block as the foreground.

Through the careful observation, we found that the block-based method can extract the more complete foreground than the pixel-based method. When the block size becomes larger, the foreground becomes more complete and the computing time is also faster. However, the condition of wrong connection between different vehicles occurs more frequently. Fig. 3 shows the comparison between the block-based and pixel-based method. It is obvious that the moving vehicles extracted with the block-based shown in Figs. 3(c)-(e) are more complete than the pixel-based method shown in Fig. 3(b). In our system, we select the block size as 2×2 to fit the size of

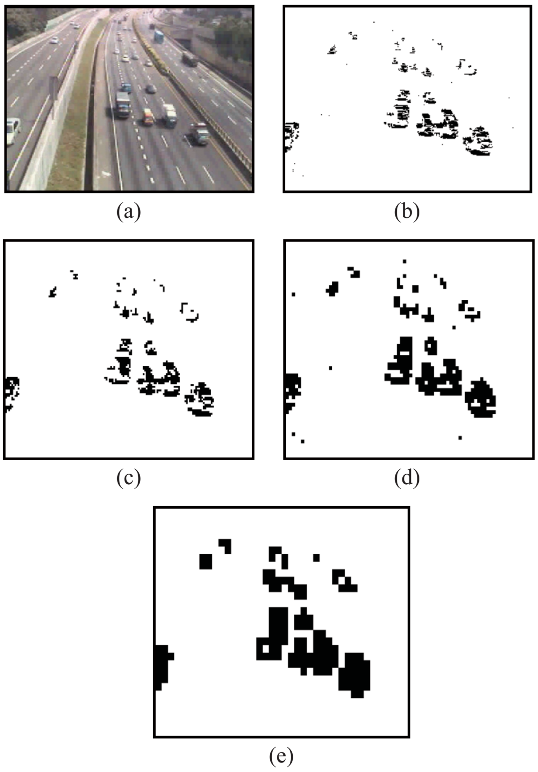


Fig. 3. Block-based foreground detections. (a) Original image, (b) Pixel-based method, (c) Block-based method with block size 2 × 2, (d) Block-based method with block size 4 × 4, and (e) Block-based method with block size 8 × 8.

moving vehicles. So far, the block-based foregrounds belong to the short-term foregrounds that will be combined with the long-term foreground to generate a more precise and complete foreground, which will be described in the next section.

2. Precise Object Region Extraction with Dual Foregrounds Fusion

The traditional frame difference methods often generate strong response in the boundary of moving object, but lower response occurs within the region of a moving object shown in Fig. 4(b). When the object becomes larger, the incomplete object detection will become more serious. To tackle this problem we apply both the characteristic of short-term and long-term foregrounds to make the foreground more complete. The short-term foreground can define precise object boundary and the long-term foreground can extract a more complete object region shown in Fig. 4(c) with motion history information. The long-term foreground is constructed by accumulating successive short-term foregrounds. By projecting the short-term foreground onto the long-term foreground, searching the precise object boundary based on the short-term foreground shown in Fig. 4(f), and preserving all information about the long-term foregrounds between the boundaries of short-term foreground, we can extract the precise region of a moving vehicle shown in Figs. 4(g) and 4(h).

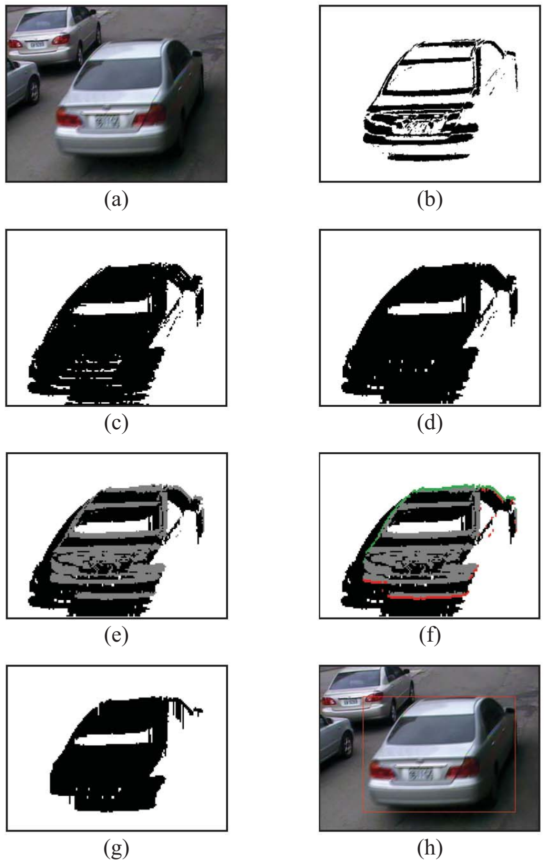


Fig. 4. Procedures for precise object region extraction with dual foregrounds fusion. (a) Image of moving vehicles, (b) Extraction of short-term foreground, (c) Extraction of long-term foreground, (d) Morphologic smoothing on image (c), (e) Integration of short-term (gray region) and long-term (black region) foregrounds, (f) Fusion with dual foregrounds (Red: low bound; green: upper bound), (g) Vehicle extraction by fusing the short-term and long-t erm foregrounds, and (h) The bounding box for the extracted vehicle in (g).

3. Foreground Segmentation

In the outdoor environment, the serious shadow problem can introduce the inaccurate and false foreground detection shown as the object #4 in Fig. 5(b) and improper merging with neighboring objects shown as the object #2 in Fig. 5(b). Based on the careful observation, the texture is not obvious in the shadow region illustrated as object #0 in Fig. 5(b). The texture analysis is then used to eliminate the object shadows. Here, the texture analysis is performed by analyzing the gradient content obtained from the Sobel and Canny operations [2]. By projecting the edge information along horizontal and vertical axes, we can find the proper segmentation position using the Ostu's segmentation method [12]. Fig. 5(c) shows the edge projection histogram along the vertical direction for the object #2 in Fig. 5(b), in which the blue and red bins denote the increase and decrease regions respectively, the black bins indicate the positions of peak points, and the green bin indicate the segmentation position. Fig. 5(d) shows the segmentation

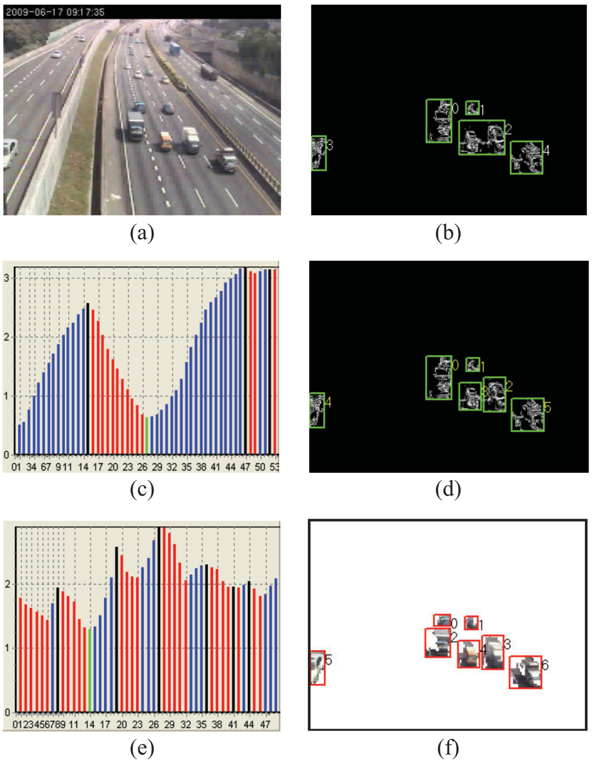


Fig. 5. The procedures for the vehicle segmentation. (a) Highway video, (b) Canny edge detection and object labeling, (c) Edge projection along the vertical direction for object #2 in (b), (d) Object labeling after vertical segmentation, (e) Edge projection along the horizontal direction for object #0 in (b), and (f) Final results of the vehicle segmentation.

result for object #2 in Fig. 5(b). Fig. 5(e) shows the edge projection histogram along the horizontal direction for object #0 in Fig. 5(b). Once the merged vehicle region is partitioned, each precise vehicle region shown in Fig. 5(f) is labeled via the labeling algorithm [5].

In general, the distribution of texture density of a moving object is higher than the distribution in its shadow region. Hence, we can utilize the texture densities to separate the regions of moving object and its shadow. By removing the boundary vertical regions with small texture density (green bins) shown in Fig. 6(b), the shadow region can be separated from the object region shown as the green regions in Fig. 6(a).

4. True Object Verification

For a real moving object, the direction distribution of motion vectors should be consistent, i.e., the motion entropy in the object region will be low. Based on this observation, the value of motion entropy for each detected foreground region can be used to distinguish whether the detected object is a true object or not. First, we apply the three-step block matching method [10] to find the motion vectors for each foreground region and compute the orientation histogram of the motion vectors. Second, we compute the motion entropy for each detected foreground with Eq. (1).

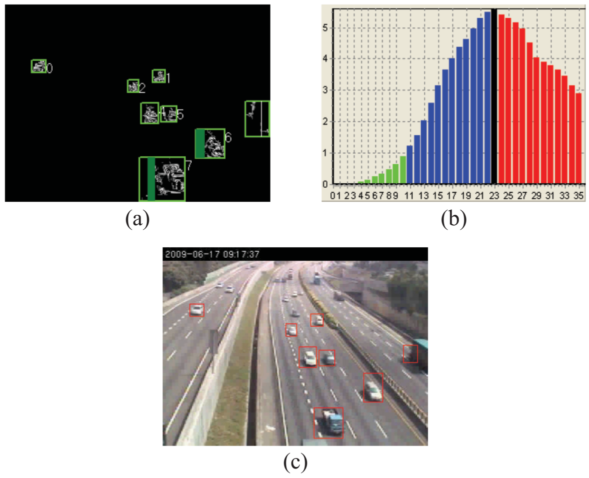


Fig. 6. Shadow Removal with the property of texture density. (a) The texture of foreground, (b) The texture density distribution for the region of object #6, and (c) The result of shadow removal.

$$E_m = -\sum_{i=1}^{K} p_m^i \log_2 p_m^i \tag{1}$$

where m is object index, i is the index of i-th bin in the orientation histogram, K is the number of total bins in the orientation histogram, and $p^i{}_m$ is the probability of the i-th bin in the orientation histogram. The motion entropies in these false detected regions are very large and then these false detected regions can be removed with the motion entropy filtering process.

III. VEHICLE TRACKING

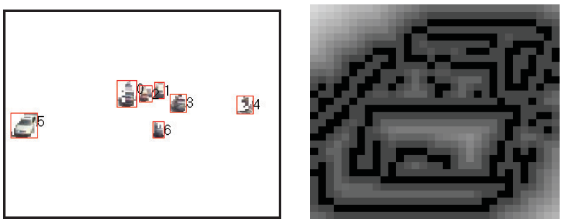
In this study, the texture feature that is not easily influenced by the illumination variation is utilized as the measurement in the vehicle tracking algorithm.

1. Kalman Filter

In general, the detected vehicles can be tracked with the methods of Kalman filter or particle filters. With the efficiency consideration, we apply the Kalman filter [17] to track each detected vehicle on the roadway. Each detected vehicle is tracked with the constant-velocity motion model. The state and measurement equations of Kalman filter are defined in Eq. (2).

$$\begin{aligned} x_k &= Ax_{k-1} + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \tag{2}$$

In Eq. (2), x_{k-1} denotes the state at frame k-1, z_k denotes the measurement at frame k, The random variables w_k and v_k represent the process and measurement noise. They are assumed to be independent, white, and normal distributed and defined as Eq. (3).



(a) (b)

Fig. 7. The extractions of the texture features with the distance transform for each tracked vehicle. (a) Tracked vehicles, and (b) Texture features for a tracked vehicle.

$$p(w) \sim N(0,Q)$$
$$p(v) \sim N(0,R) \quad (3)$$

where, Q denotes the process noise covariance matrix and R denotes the measurement noise covariance matrix. In order to determine whether an observed vehicle belongs to a new incoming vehicle or a previously existed vehicle, we propose an efficient matching algorithm to calculate the matching distance. For the texture matching, the texture feature can be generated from the canny edge [2] and the distance transform [6]. With the canny edge detection method we can retrieve the rough contours of the detected moving vehicles that is hard to be influenced by the illumination variation and shadow, which is shown in Fig. 7. Then, the distance transform is applied to each vehicle's contour image to retrieve the texture content for the purpose of vehicle matching.

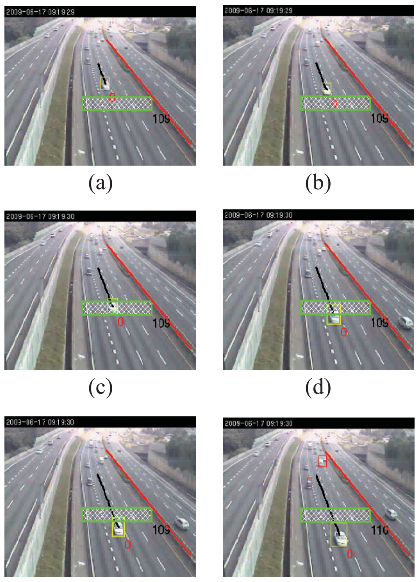
Distance transform matching is a technique for finding the matching distance between two different texture images by minimizing shifted template matching distance. The optimal matching position can be acquired with the formula in Eq. (4).

$$(m^*, n^*) = \arg\min_{m,n} \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \left| Tem(i-m, j-n) - Ref(i,j) \right|^2 \quad (4)$$

where *Tem* is the template image and *Ref* is the reference image. The size of matching region is $M \times N$. The optimal matching distance $d(m^*, n^*)$ can be found at the position (m^*, n^*). If the value of optimal matching distance $d(m^*, n^*)$ is less than the specified threshold, we can initialize a new Kalman filter to track or update the Kalman filter with the observed vehicle position.

2. Vehicle Counting

Here, we set a virtual detecting region on the entire road to monitor whether the vehicles reach the detecting region or not. When a tracked vehicle touches this region, we start to monitor its trajectory. If the vehicle's trajectory satisfies the following two conditions, then the vehicle is counted. First, the vehicle is detected in the virtual detecting region. Second, the length of the trajectory of a tracked vehicle must be larger than a specified length threshold. Some examples of vehicle counting are shown in Fig. 8. In Fig. 8, the red line is used to separate the different directions of traffic flow and the green region is the virtual detection region for vehicle counting. The black text represents the current number of the passed vehicles.



(a) (b) (c) (d) (e) (f)

Fig. 8. The procedures of vehicle counting. The tracked vehicle is shown from (a) to (b). The images from (c) to (e) show that the tracked vehicle is driving through the virtual detection region. The image in (f) shows that the tracked vehicle passes the virtual detecting region.

IV. EXPERIMENTAL RESULTS

In this study, the video sequence of PetsD2TeC2 (resolution: 384×288) [7] and the traffic videos(resolution: 348×240) captured from the website of Taiwan Area National Freeway Bureau (http://www.nfreeway.gov.tw) are used as the test videos. All the experiments are performed on the platform of Windows XP and Intel CPU E6550 2.33GHz.

1. The Detection of Moving Vehicles

Here we use the PetsD2TeC2 video sequence to evaluate the performance of foreground detection for our method, traditional frame difference method, and the method of background subtracting with Gaussian Mixture Model method. The block size is chosen as 2×2 to fit the size of moving object in our system. The lower bound of object size is set to be 10 blocks for the object detection. The foreground detections in Figs. 9(b) and 9(f) show the scene adaptability for the frame difference method. However, the problem of fragmented region occurs. The foreground detection in Fig. 9(c) shows the object completeness for the background modeling method [7]. However, in Fig. 9(g), a noisy image is generated by the inefficient background updating process. The foreground detections in Figs. 9(d) and 9(h) show that

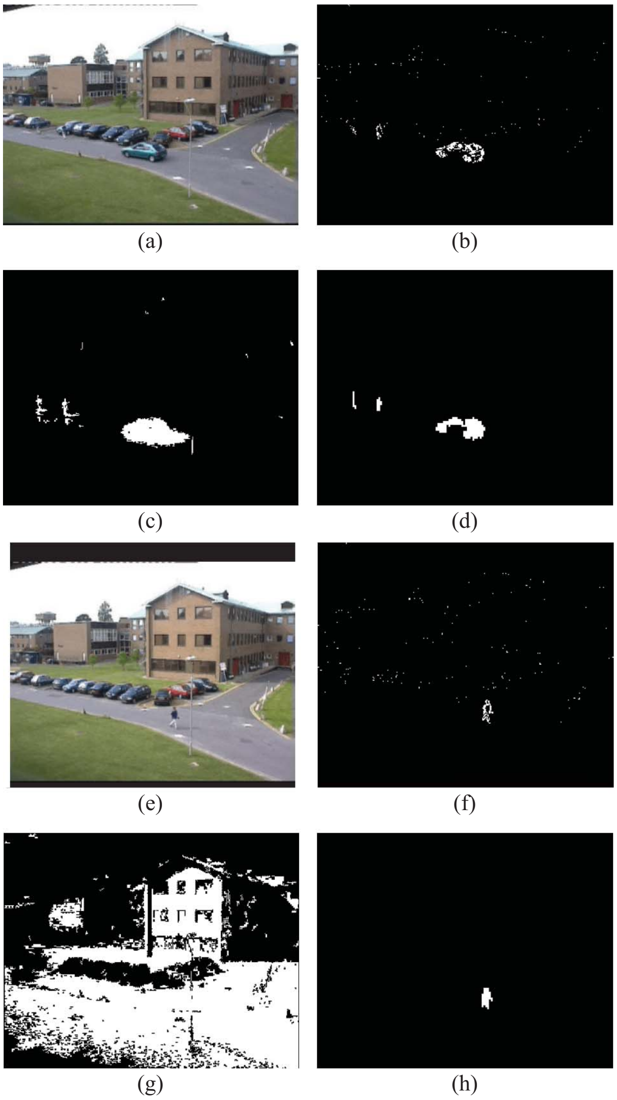


Fig. 9. Two examples of foreground detections at frames #1023 and #2517 are illustrated in PetsD2TeC2 videos. The original frames are shown on images (a) and (e). The foreground detection using the conventional frame difference method is shown in images (b) and (f). The foreground detections using the GMM background modeling method are shown in images (c) and (g). The foreground detections using our proposed method are shown in images (d) and (h).

the good scene adaptability and detection completeness for our proposed method. It is obvious that our method outperforms the other typical methods in terms of scene adaptability and detection completeness. In Fig. 10, we illustrate two additional simulation results for the vehicle detection on the freeway. Figs. 10(a) and 10(c) show the video sequences of detected vehicles on the freeway. Figs. 10(b) and 10(d) show the bounding boxes on the detected vehicles.

2. Vehicle Tracking and Counting

In Fig. 11, we show that each vehicle can be tracked robustly under a normal condition on the freeway. In Fig. 12,

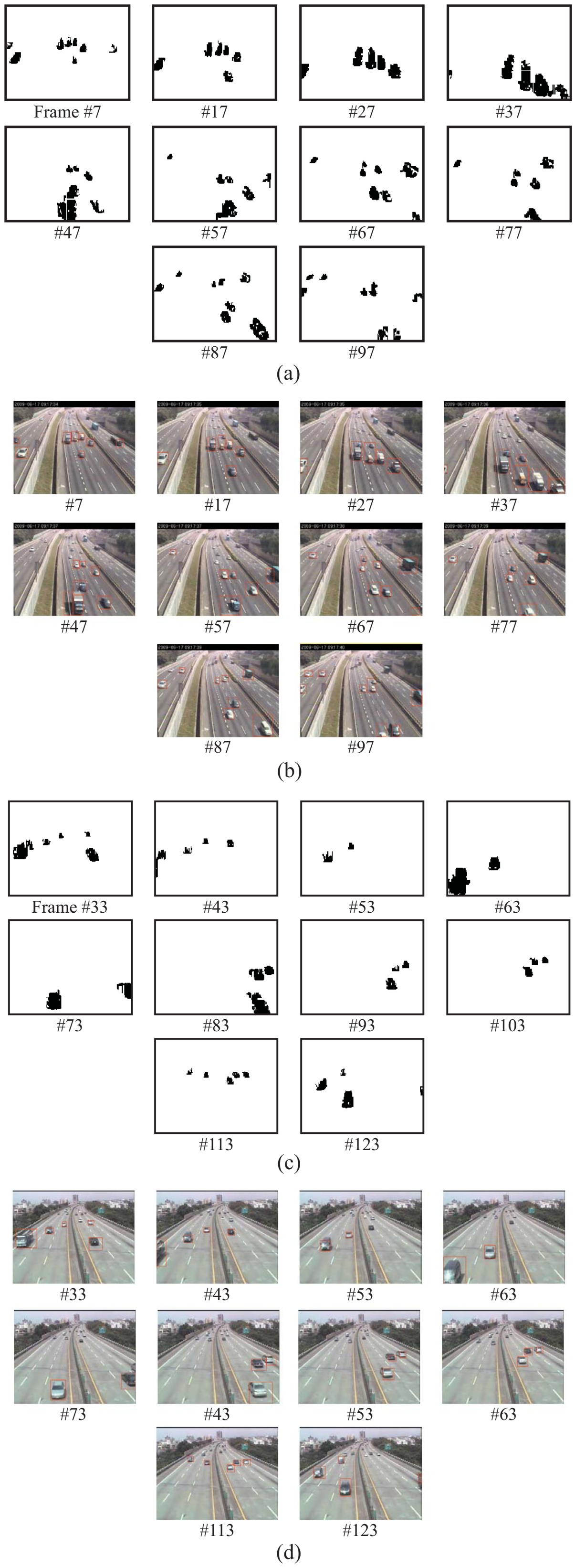
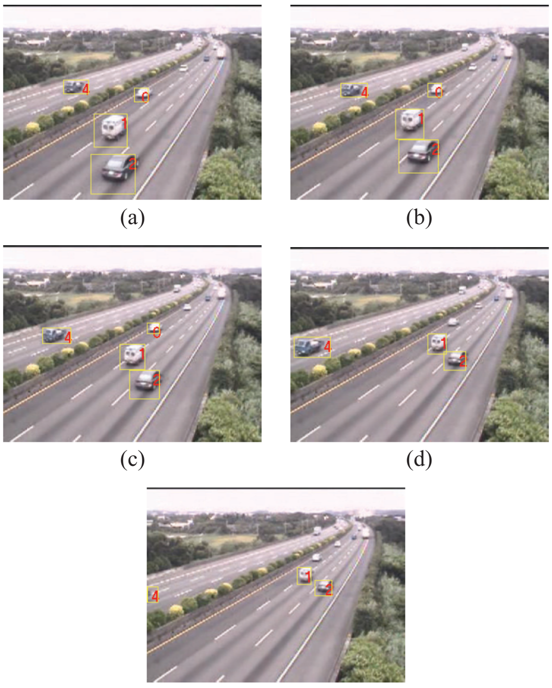


Fig. 10. The detection for the moving vehicles on the freeway. The detected vehicles in two video sequences are shown in (a) and (c). The bounding boxes on the detected vehicles for (a) and (c) are shown in (b) and (d) respectively.

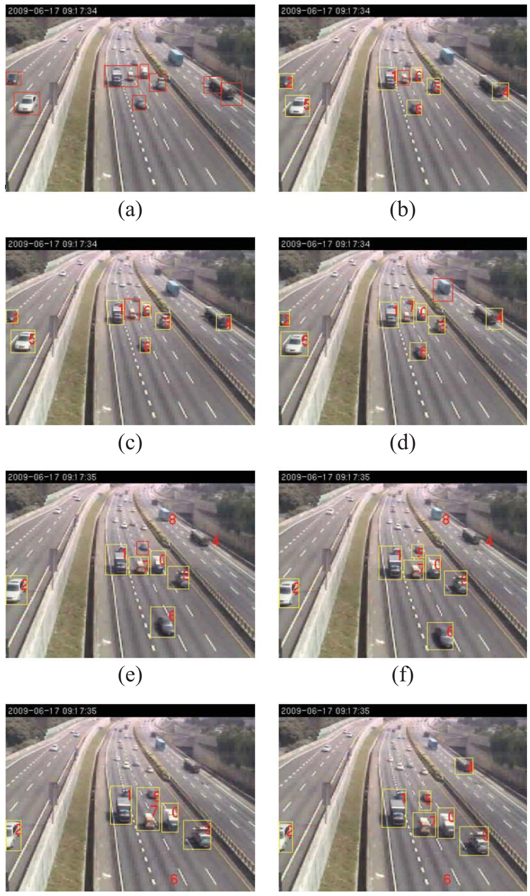


(a) (b) (c) (d) (e)

Fig. 11. Vehicle tracking on the freeway scene.

Table 1. The accuracy analysis of the vehicle counting.

Video Clips	1^{st} video clip	2^{nd} video clip	3^{rd} video clip
Actual number	85	88	76
Detected number	90	94	85
False positive	8	9	11
False negative	3	3	2
Accuracy	89%	88%	83%



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 12. Vehicle tracking on the crowd freeway scene.

we illustrate the vehicle tracking on the crowd freeway scene. In Fig. 12(a), the system starts to detect the moving vehicles denoted as the red rectangular bounding, i.e., the vehicles with red bounding boxes have not been tracked yet. Figs. 12(b) to 12(h) show the complete vehicle tracking processes including the vehicle segmentation with the texture distribution mentioned in section 2.3 and vehicle tracking with Kalman filter. The tracked vehicles are denoted as the yellow rectangular bounding boxes annotated with index numbers.

After the vehicle tracking, we set a detecting region on the roadway to count the moving vehicles. The detecting region across multiple lanes is used to count the vehicles. The accuracy analysis of vehicle counting is listed in Table 1. In the first and second video clips, the traffic flow is normal and then the accuracy of the vehicle counting is satisfied. In the third video clip, the accuracy of vehicle counting is lower because a few large buses introduce very serious occlusions and wrong segmentation problems. Experimental results show that our proposed system can work in real time with the computing rate above 20 fps and the accuracy of vehicle counting can approach 86%.

V. CONCLUSION

In this paper, we propose a novel vehicle detection method without background modeling in which the modified block-based frame differential method, the precise object region extraction with dual foregrounds, the foreground segmentation, and the true object verification are integrated to develop a scene adaptive vehicle detection system. The texture-based target tracking method is proposed to track each detected target and then apply the virtual-loop detector to analyze the traffic flow. The computing efficiency of the vehicle detection/tracking is about 20 fps and the accuracy of vehicle counting can approach 86%.

REFERENCES

1. Baş, E., Tekalp, A. M., and Salman, F. S., "Automatic vehicle counting from video for traffic flow analysis," *Proceeding of IEEE Intelligent Vehicles Symposium*, pp. 392-397 (2007).
2. Canny, J., "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679-698 (1986).
3. Chen, T. H., Lin, Y. F., and Chen, T. Y., "Intelligent vehicle counting

method based on blob analysis in traffic surveillance," *Proceeding of International Conference on Innovative Computing, Information and Control*, pp. 238-238 (2007).
4. Dai, X. and Khorram, S., "Performance of optical flow techniques," *Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 236-242 (1992).
5. Dellepiane, S. G., Fontana, F., and Vemazza, G. L., "Nonlinear image labeling for multivalued segmentation," *IEEE Transactions on Image Processing*, Vol. 5, No. 3, pp. 429-446 (1996).
6. Ghafoor, A., Iqbal, R. N., and Khan, S., "Image matching using distance transform," *Proceeding of Scandinavian Conference on Image Analysis*, Vol. 2749, pp. 654-660 (2003).
7. Huang, T., Qiu, J., Sakayori, T., Goto, S., and Ikenaga, T., "Motion detection based on background modeling and performance analysis for outdoor surveillance," *Proceeding of IEEE International Conference on Computer Modeling and Simulation*, pp. 38-42 (2009).
8. Jain, R., Martin, W., and Aggarwal, J., "Segmentation through the detection of changes due to motion," *Compute Graph and Image Processing*, Vol. 11, pp. 13-34 (1979).
9. Kamijo, S., Matsushita, Y., Ikeuchi, K., and Sakauchi, M., "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 2, pp. 703-708 (2000).
10. Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion compensated interframe coding for video conferencing," *Proceeding of National Telecommunication Conference*, pp. G5.3.1-G5.3.5 (1981).
11. Lei, M., Lefloch, D., Gouton, P., and Madani, K., "A video-based real-time vehicle counting system using adaptive background method," *Proceeding of IEEE International Conference on Signal Image Technology and Internet Based Systems*, pp. 523-528 (2008).
12. Liao, P. S., Chen, T. S., and Chung, P. C., "A fast algorithm for multilevel thresholding," *Journal of Information Science and Engineering*, Vol. 17, pp. 713-727 (2001).
13. Lien, C. C., Wang, J. C., and Jiang, Y. M., "Multi-mode target tracking on a crowd scene," *Proceedings of the International Conference on International Information Hiding and Multimedia Signal Processing*, Vol. 2, pp. 427-430 (2007).
14. Nummiaro, K., Meier, E. K., and Gool, L. J. V., "An adaptive color-based particle filter," *Image Vision Computing*, Vol. 21, No. 1, pp. 99-110 (2002).
15. Qin, B., Zhang, M., and Wang, S., "The research on vehicle flow detection in complex scenes," *Proceeding of IEEE International Symposium on Information Science and Engineering*, Vol. 1, pp. 154-158 (2008).
16. Wang, K. F., Li, Z. J., Yao, Q. M., Huang, W. L., and Wang, F. Y., "An automated vehicle counting system for traffic surveillance," *Proceeding of IEEE International Conference on Vehicular Electronics and Safety*, pp. 1-6 (2007).
17. Welch, G. and Bishop G., "An introduction to the Kalman filter," Report No. TR 95-041, Department of Computer Science University of North Carolina (2006).
18. Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A., "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 780-785 (1997).
19. Xu, M., Orwell, J., Lowey, L., and Thirde, D., "Architecture and algorithms for tracking football players with multiple cameras," *Proceeding of IEEE International Conference on Vision, Image and Signal Processing*, Vol. 152, No. 2, pp. 232-241 (2005).
20. Zhou, J., Gao, D., and Zhang, D., "Moving vehicle detection for automatic traffic monitoring," *IEEE Transactions On Vehicular Technology*, Vol. 56, No. 1, pp. 51-59 (2007).