

A Method for Tracking Vehicles Under Occlusion Problem

Cuihong Xue¹, Yang Yu¹(✉), Luzhen Lian¹, Yude Xiong¹,
and Yang Li²

¹ School of Control Science and Engineering, Hebei University of Technology,
Tianjin 300130, China

xuecuihong@scse.hebut.edu.cn,

{yuyang, yuming}@hebut.edu.cn,

{lianluzhen_tjnu, zmgxyd2009}@163.com

² China United Network Communications Corporation Limited,
Tianjin 300100, China

liyangle@tj.chinaunicom.com

Abstract. A method of overcoming occlusion in vehicle tracking system is presented here. Firstly, the features of moving vehicles are extracted by the vehicle detection method which combines background subtraction and bidirectional difference multiplication algorithm. Then, the affection of vehicle cast shadow is reduced by the tail lights detection. Finally, a two-level framework is proposed to handle the vehicle occlusion which are NP level (No or Partial level) and SF level (Serious or Full level). On the NP level, the vehicles are tracked by mean shift algorithm. On the SF level, occlusion masks are adaptively created, and the occluded vehicles are tracked in both the original images and the occlusion masks by utilizing the occlusion reasoning model. The proposed NP level and SF level are sequentially implemented in this system. The experimental results show that this method can effectively deal with tracking vehicles under ambient occlusion.

Keywords: Occlusion masks · Vehicle detection · Intelligent transportation system (ITS) · Occlusion reasoning model

1 Introduction

The increasing number of vehicles makes the intelligent transportation system (ITS) more and more significant. For the surveillance concern, various techniques have been proposed to alleviate the pressure of transportation systems. The vision-based system has absorbed many researchers owing to its lower price and much direct observation information. ITS extracts useful and accurate traffic parameters for traffic surveillance, such as vehicle flow, vehicle velocity, vehicle count, vehicle violations, congestion level and lane changes, et al. Such dynamic transportation information can

This work was supported by the National Nature Science Foundation of China under Grant No. 60302018 and Tianjin Sci-tech Planning Projects (14RCGFGX00846)

© Springer International Publishing Switzerland 2015

Y.-J. Zhang (Ed.): ICIG 2015, Part I, LNCS 9217, pp. 136–150, 2015.

DOI: 10.1007/978-3-319-21978-3_13

be disseminated by road users, which can reduce environmental pollution and traffic congestion as a result, and enhance road safety [1]. Multiple-vehicle detection and tracking is a challenging research topic for developing the traffic surveillance system [2]. It has to cope with complicated but realistic conditions on the road, such as uncontrolled illumination, cast a shadow and visual occlusion [3]. Visual occlusion is the biggest obstacle among all the problems of outdoor tracking.

The current method for handling the vehicle occlusion can be divided into three categories, feature-based tracking [4–8], 3D-model-based tracking [9] and reasoning-based tracking [10]. Based on normalized area of intersection, Fang [5] proposed to track the occluded vehicles by matching local corner features. Gao [11] uses a Mexican hat wavelet to change the mean-shift tracking kernel and embedded a discrete Kalman filter to achieve satisfactory tracking. In [9], a deformable 3D Model that could accurately estimate the position of the occluded vehicle was described. However, this method has a high computational complexity. Anton et al. [10] presented a principled model for occlusion reasoning in complex scenarios with frequent inter-object occlusions. The above approaches do not work well when the vehicle is seriously or fully occluded by other vehicles.

In order to cope with the vehicle occlusion in the video sequence captured by a stationary camera, we propose a two-level framework. The basic workflow of the proposed method is illustrated in Fig. 1, in which two modules is consisted: (1) vehicle detection unit; (2) vehicle tracking unit. In vehicle detection module, features are extracted for fast vehicle localization by combining the improved frame difference algorithm and background subtraction algorithm, and with the aid of the distance between the pair of tail lights, the vehicle cast shadow is reduced. In vehicle tracking module, the two-level framework is proposed to handle vehicle occlusion: (1) NP level (No or Partial level), tracking vehicles by mean-shift algorithm; (2) SF level (Serious or Full level), handling occlusions by occlusion reasoning model. Occlusion masks are adaptively created, and the detected vehicles are tracked in both the created occlusion masks and original images. For each detected vehicle, NP and SF levels are implemented according to the degree of the occlusion (see Fig. 2). With this framework, most partial occlusions are effectively managed on the NP level, whereas serious or full occlusions are successfully handled on the SF level.

The rest of this paper is given as follow: in Sect. 2 the moving vehicle detection method is presented. NP level occlusion handling is described in Sect. 3, and SF level occlusion handling is introduced in Sect. 4. Section 5 shows the experimental results and discussion. The conclusion is given in Sect. 6.

2 Moving Vehicle Detection

2.1 Moving Vehicle Detection

At present, frame difference [12] and background subtraction [13] are conventional approaches for detecting moving vehicle in the vision-based system. Frame difference can keep the contour of the moving vehicle very well. However, it is failed when the speed of inter-frame is less than one pixel or more than the length of the moving

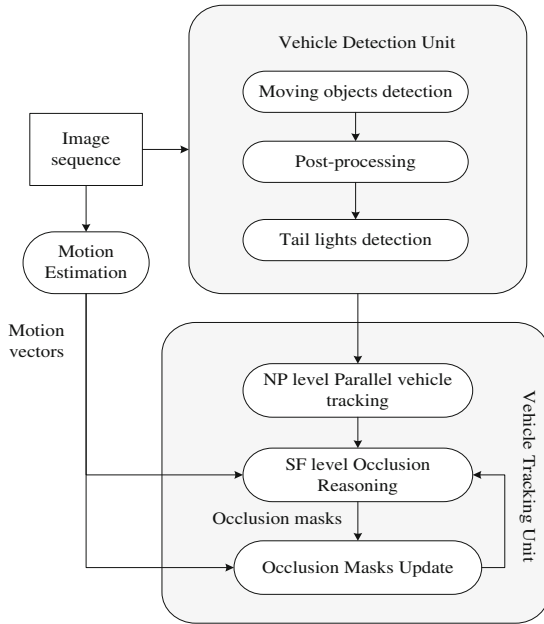


Fig. 1. Overview of the proposed framework.

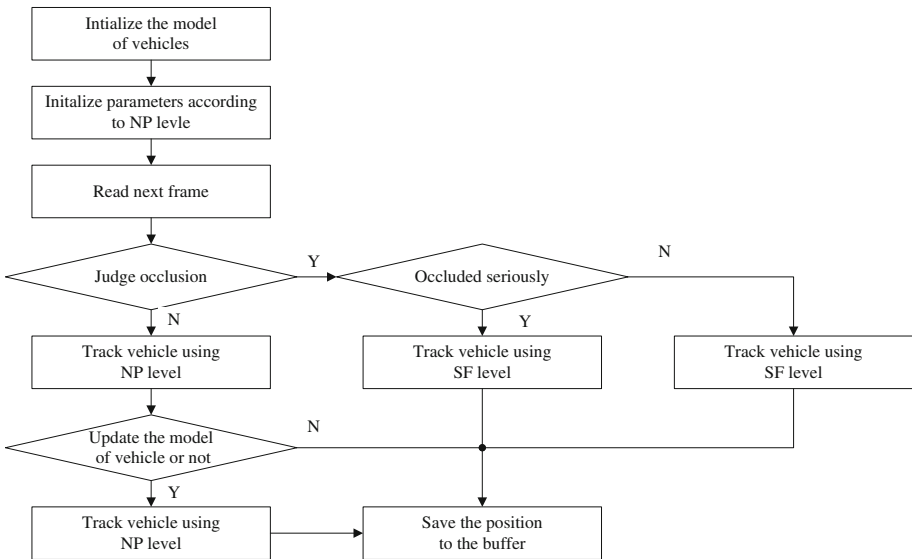


Fig. 2. Diagram of the two-level framework.

vehicle. So, a bidirectional difference multiplication algorithm is used here which can meet the needs of slow or fast speed. The proposed method can overcome the shortage

of frame difference as the method adopts the discontinuous frame difference and bidirectional frame difference. The bidirectional difference multiplication algorithm detects targets based on relative motion and can detect multiple vehicles at the same time. In order to overcome the disadvantage when the speed of inter-frame is more than the length of the vehicle, here firstly, the bidirectional frame differences are calculated among the current k th frame, the $(k - 2)$ th frame and the $(k + 2)$ th frame images; furthermore, the difference results are multiplied in order to strengthen the motion region. The frame difference image $D_{fd}(x,y)$ is given by Eqs. (1) and (2):

$$\begin{cases} D_{k+2,k}(x,y) = |f_{k+2} - f_k| \\ D_{k,k-2}(x,y) = |f_k - f_{k-2}| \end{cases} \quad (1)$$

$$D_{fd}(x,y) = D_{k+2,k}(x,y) \times D_{k,k-2}(x,y) \quad (2)$$

In which $D_{k+2,k}(x,y)$ is the difference between the $(k + 2)$ th frame and the k th frame. And $D_{k,k-2}(x,y)$ is the difference between the $(k-2)$ th frame and the k th frame. The basic diagram is shown in Fig. 3.

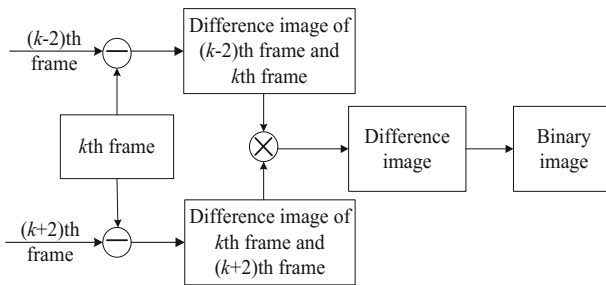


Fig. 3. Diagram of the bidirectional multiplication algorithm.

The primary idea of background subtraction is to subtract the current frame image from the background model pixel by pixel first, and then to sort the obtained pixels as foreground or background by a threshold δ . Compared with complicated background modeling techniques, the multi-frame average method has the high computation speed and low memory requirement. Therefore, we utilize a multi-frame average method to update the background model. This method is formulated as follows,

$$D_{bd} = \begin{cases} 1 & \text{foreground, if } |I-B| > \delta \\ 0 & \text{background, otherwise} \end{cases} \quad (3)$$

where I is the current frame image, B is the background image.

The background model needs to be updated momentarily due to the constant changes of the real traffic environment. Here the multi-frame average method is used to update the background model by Eq. (4).

$$B_{k+1} = (1 - \alpha)B_k + \alpha I_k \quad (4)$$

Where α ($0 < \alpha < 1$) determines the speed of the background updating. B_k and B_{k+1} mean the background model at the time k and $k + 1$ independently.

The more precise foreground image D is obtained by using Eq. (5), adding D_{fd} and D_{bd} pixel by pixel, in which background subtraction can keep the integrity of the moving vehicle and the bidirectional difference multiplication algorithm can obtain the full contour of the moving vehicle information.

$$D = D_{fd} \cup D_{bd} \quad (5)$$

2.2 Post-processing

The binarized foreground image D may have holes and some noise (see Fig. 4(b)), which influence results of target tracking. A flood-fill operation is utilized on D to eliminate the holes in the motional target (see Fig. 4(c)). Moreover, the morphological closing and opening operators are utilized to fuse the narrow breaks and remove the noise (see Fig. 4(d)).

2.3 Tail Lights Detection

In order to reduce the affection of vehicle cast shadow, the distance between the pair of tail lights is utilized to determine the precise region of the moving vehicle, and the

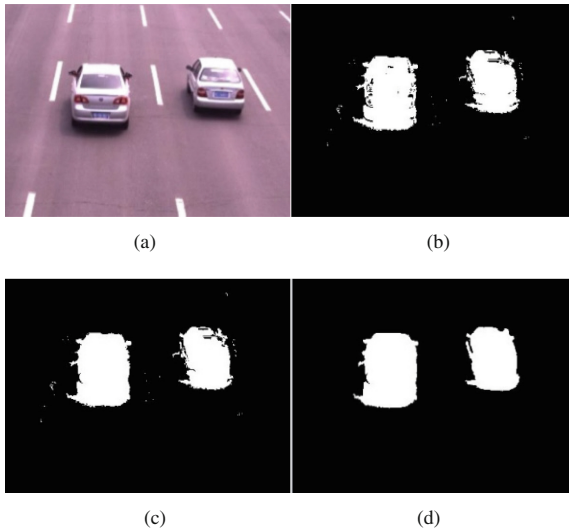


Fig. 4. Moving vehicle detection. (a) Original image. (b) Detection result of Section 2.1. (c) The result after the flood-fill operation. (d) The final binarized image by using morphological closing and opening operators.

method was first proposed by Qing et al. [14]. The tail lights are detected in the obtained partial image. Then, the following constraint is used to determine the light pairs.

$$\begin{cases} |h_{c_1} - h_{c_2}| \leq d \\ w_{\min} \leq w_{c_1 c_2} \leq w_{\max} \end{cases} \quad (6)$$

Where c_1 and c_2 are the barycenter of the tail lights, h_{c_1} and h_{c_2} represent the height of c_1 and c_2 respectively; d is the image resolution. $w_{c_1 c_2}$ is the width between the pair of tail lights for a moving vehicle.

3 NP Level Occlusion Handling

In this paper, the mean-shift algorithm is utilized to track detected vehicles in a video sequence. Usually, mean-shift algorithm adopts color histogram of the detected target as feature [15]. In order to find more similar candidate area in the neighborhood of the moving target, we define Bhattacharyya coefficient [16] as the similarity function in this paper.

3.1 Target Model and Candidate Model

The first step is to initialize the position of the target region in the first frame. Target model can be described as the probability density distribution of the color feature value in the target region [17]. The probability distribution function (PDF) of the target model q_u and the candidate model p_u are calculated as follow:

$$q_u(x) = C \sum_{i=1}^n k \left(\left\| \frac{x_i - x_0}{h} \right\|^2 \right) \delta[b(x_i) - u] \quad (7)$$

Where x_0 is the barycenter of the target region, $k(\|x\|^2)$ represents kernel function; h indicates the bandwidth of the kernel function, and b is the color histogram index function of pixels.

$$p_u(y) = C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{x_i - y}{h} \right\|^2 \right) \delta[b(x_i) - u] \quad (8)$$

where C_h is the normalization coefficient.

3.2 Bhattacharyya Coefficient and Target Position

Here Bhattacharyya coefficient is defined as the similar function, which can be expressed as

$$\rho(y) = \rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y)q_u} \quad (9)$$

The similarity function is Taylor expansion in the point of $p_u(y_0)$, and the formula is given as follows:

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0)q_u} + \frac{C_h}{2} \sum_{i=1}^n \omega_i k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \quad (10)$$

where

$$\omega_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}} \quad (11)$$

The best candidate model has the largest value of $\rho(y)$ and can be searched by mean shift iterations. First, the barycenter of the target region y_0 in the current frame is set to as the barycenter of the target region in the previous frame; that is $y_0 = x_0$. Then, searching the optimal matching position y_1 around the barycenter of y_0 by the Bhattacharyya coefficient and the updated barycenter of the target region y_1 is calculated by (12). At last, stopping the iterative convergence when $\|y_1 - y_0\| < \varepsilon$, and the barycenter of the target will be replaced by y_1 .

$$y_1 = \frac{\sum_{i=1}^n \omega_i (x_i - y_0) g \left(\left\| \frac{y_0 - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n \omega_i g \left(\left\| \frac{y_0 - x_i}{h} \right\|^2 \right)} \quad (12)$$

So the moving target barycenter is adjusted gradually from the initial position to the real position.

4 SF LEVEL OCCLUSION HANDLING

On the NP level, partial occlusion can be easily solved, whereas, for serious or full occlusions, this level appears to be ineffective. An occlusion reasoning model which combines with constructing occlusion masks to estimate the information of the motional vehicle is proposed on the SF level.

On the SF level, occlusion reasoning model is used to track moving objects based on barycenter and vector here. Vehicle tracking is described as follows. Let VC_n^i be the motion vector of the i th detected a vehicle in the n th frame. The barycenter position of VC_n^i is denoted as $(C_{n,x}^i, C_{n,y}^i)$. The average motion vector of VC_n^i is denoted as

$(V_{n,x}^i, V_{n,y}^i)$. Therefore, we can estimate the barycenter position of VC_n^i as $(\tilde{C}_{n,x}^i, \tilde{C}_{n,y}^i)$ in the $(n + 1)$ th frame by Eq. (13):

$$\begin{cases} \tilde{C}_{n,x}^i = C_{n,x}^i + V_{n,x}^i \\ \tilde{C}_{n,y}^i = C_{n,y}^i + V_{n,y}^i \end{cases} \quad (13)$$

Here VC_n^i and VC_{n+1}^j are the same vehicle if VC_n^i and VC_{n+1}^j meet the constraining conditions as follows:

$$\begin{cases} |A_n^i - A_{n+1}^j| < Threshold_A \\ \sqrt{(\tilde{C}_{n,x}^i - C_{n+1,x}^j)^2 + (\tilde{C}_{n,y}^i - C_{n+1,y}^j)^2} < Threshold_C \end{cases} \quad (14)$$

where A_n^i and A_{n+1}^j are areas of detected vehicles in the i th and $(i + 1)$ th frames.

Occlusion masks represent the estimated images that make up of the moving vehicle regions occluded by other vehicles or no-vehicles. The detected vehicles are tracked in both adaptively obtained occlusion masks and captured images.

The proposed occlusion reasoning model is described in Fig. 5. In the n th frame, the motion vectors and the predicted motion vectors are $\{VC_n^1, VC_n^2, VC_n^3, \dots, VC_n^i\}$ and $\{\tilde{VC}_n^1, \tilde{VC}_n^2, \tilde{VC}_n^3, \dots, \tilde{VC}_n^i\}$ individually. First, the predicted vehicle \tilde{VC}_n^i is matched with all of the vehicles in the frame $(n + 1)$ th with the Eq. (14). If \tilde{VC}_n^i matches none of the vehicles in the frame $(n + 1)$ th, it will be checked in a preset exiting region which is decided by the barycenter position of VC_n^i , vehicle area A_n^i and mean of motion vector. If the unmatched vehicle in the exiting region, it means that the vehicle VC_n^i has moved out of the scene in the frame $(n + 1)$ th. If not, it is assumed that the vehicle VC_n^i moves into the occlusion in the frame $(n + 1)$ th. Therefore, the occlusion mask is created and added the unmatched \tilde{VC}_n^i . Occlusion mask is created for each occluded vehicle; that is to say, there is only one vehicle in each occlusion mask. Then, the position of vehicle \tilde{VC}_n^i in occlusion mask is updated according to the motion vector frame by frame. Moreover, the vehicles in the occlusion mask are matched with all of the vehicles in the next frame. If the vehicle in the occlusion mask matches with a vehicle in the next frame, it is assumed that the vehicle in the occlusion mask has moved out of the occlusion mask. At last, the matched vehicle in the occlusion mask is deleted with the occlusion mask.

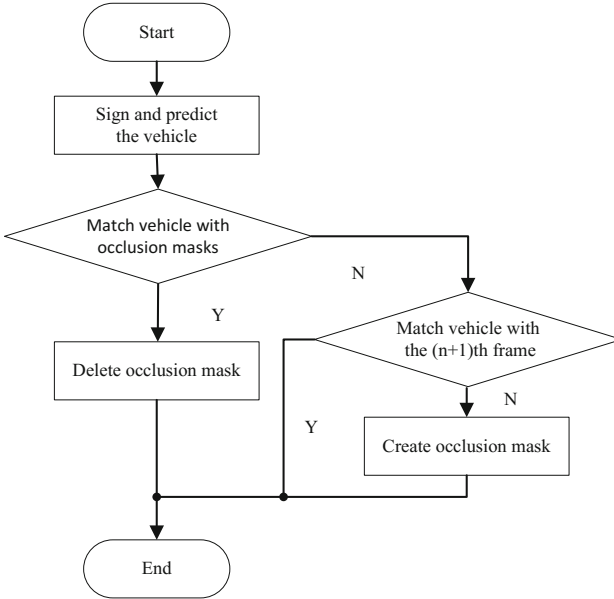


Fig. 5. The workflow of the occlusion reasoning model.

5 Simulation Results

The proposed method is evaluated on traffic sequences. The vehicle is partially occluded, seriously occluded or fully occluded, and finally, the vehicle appears in the scene again when a vehicle is overtaken by another vehicle. The image sequences used in experiments are captured by a stationary camera and live traffic cameras, which are available on a Web page at <http://vedio.dot.ca.gov/>.

Experimental results are shown in Fig. 6. The results show that the proposed method has better performance compared to other vehicle detection algorithms, background subtraction [18], adaptive background subtraction [19], the frame difference [20] and Bidirectional multiplication method [21].

In Fig. 7, the vehicles are accurately tracked under the condition of without occlusion. A typical occlusion is used to demonstrate the efficient of the proposed framework. In Fig. 8, a vehicle is overtaken by another vehicle and is occluded. In Fig. 9, the vehicles are tracked by the mean-shift algorithm, which is not efficient when the vehicle is seriously occluded.

The proposed framework is quantitatively evaluated on real-world monocular traffic sequences. The traffic sequences including un-occluded and occluded vehicles are used in our experiments, and the results are shown in Table 1. For partial occlusion, the handling rate of the proposed framework is 85.3 %. The inefficient conditions mainly occur in the same color vehicles. For the full occlusion, the handling rate is 84.6 %, and tracking errors mainly appear in the situation of seriously or fully occlusion for a long time. The average processing times of the methods proposed by Faro [22], Zhang [23],

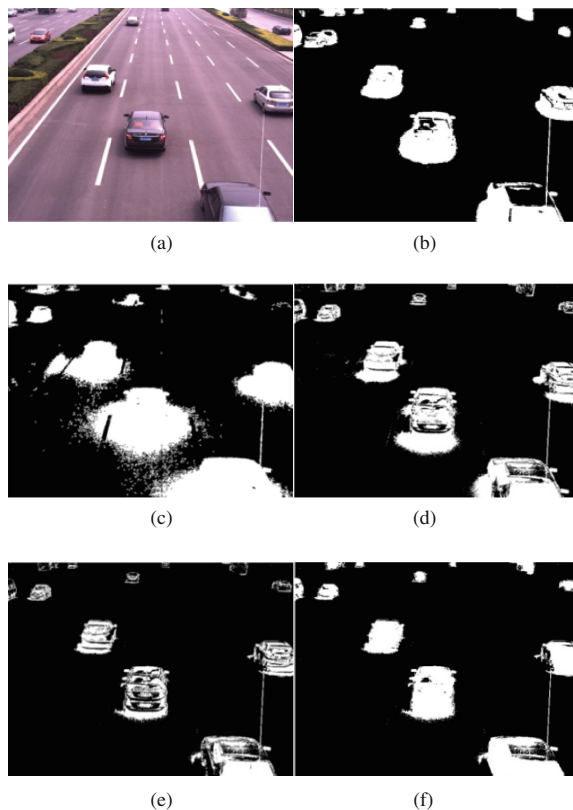


Fig. 6. Moving region detection (a) Original image. (b) Moving region detected by background subtraction. (c) Moving region detected by adaptive background subtraction (d) Moving region detected by frame difference. (e) Moving region detected by Bidirectional multiplication method. (f) Moving region detected by the proposed method.

Qing [14] and our method are shown Table 2. The image sequences are captured by live traffic cameras, which are available on a Web page at <http://vedio.dot.ca.gov/>. From the experiments, we can see the proposed method has a good balance between vehicle counting and times.

The mean processing time of the proposed method for a 3-min-long video is 236.3 s, whereas Faro's method and Zhang's method reach average processing time of about 247 s and 280.3 s. Qing's method reaches an average processing time of about 223.7 s; however, it cannot handle the serious occlusion and full occlusion.

In Fig. 10, the dashed line is the un-occluded vehicle, whereas the solid line shows the vehicle 2 which was occluded by vehicle 1, and the Asterisk line is the estimated position by the proposed method. The vehicle 2 was occluded at the 19th frame and reappeared at the 30th frame. In Fig. 8, the position of vehicle 2 is estimated by the occlusion reasoning model in 1340th to 1360th frame. The proposed method is proved to be accurate by the Fig. 9.

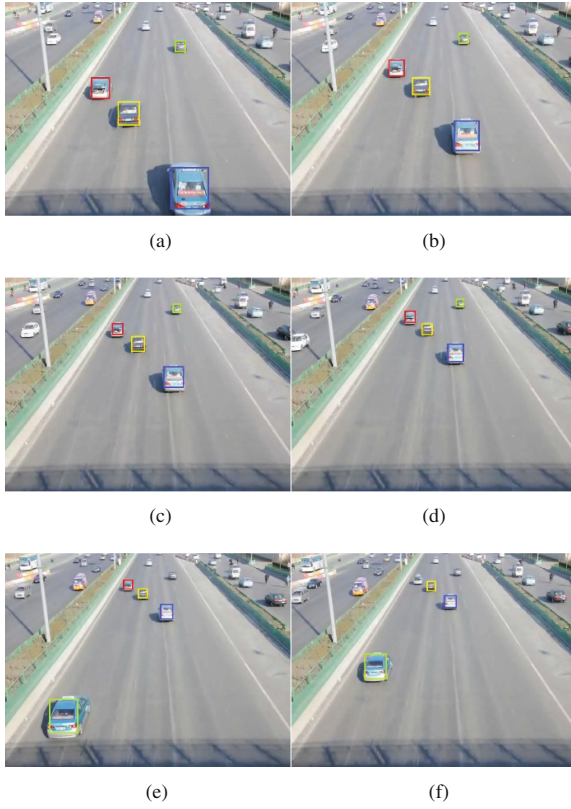


Fig. 7. Vehicles tracking by the proposed method without occlusion.

Table 1. Quantitative evaluation of the proposed framework

Occlusion	NP level	SF level	Total	Rate
Partial 184	148	9	157	85.3%
Full 39	0	33	33	84.6%

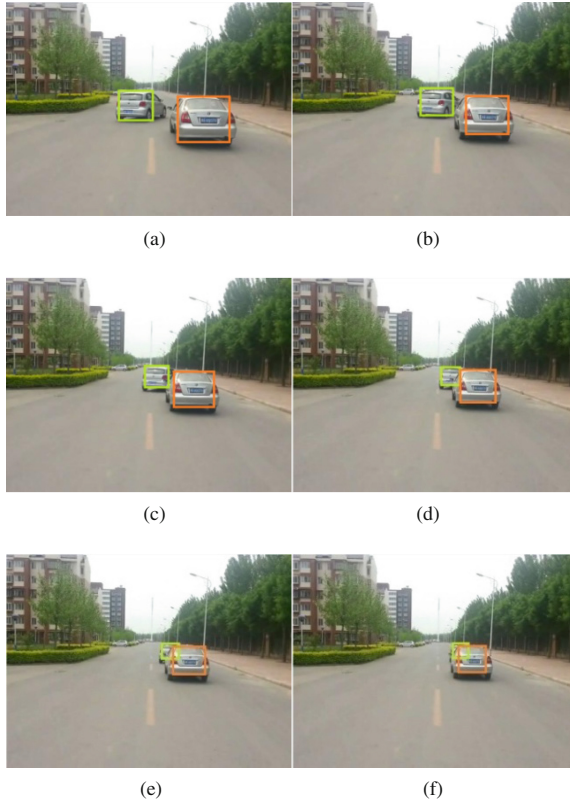


Fig. 8. The proposed method handling of the vehicle occlusion.

Table 2. Comparison the average processing times for 3-min-long videos

Algorithm	Average processing times (s)
Our method	236.3
Faro's method	247
Zhang's method	280.3
Qing's method	223.7

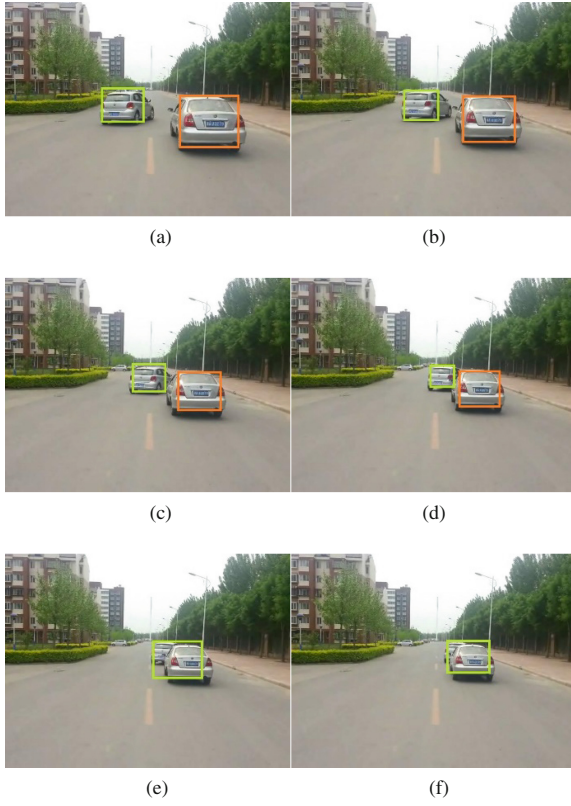


Fig. 9. Mean-shift algorithm handling of the vehicle occlusion

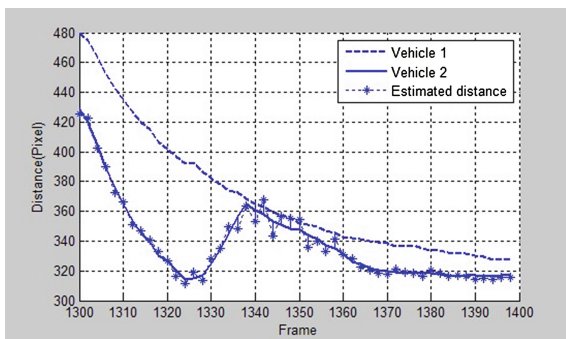


Fig. 10. Real position and estimated position during a tracking.

6 Conclusions

This paper presents a novel two-level framework to handle vehicles occlusion, which consists of NP level and SF level. On the NP level, color information is selected as the observation model, un-occluded vehicles are tracked by mean shift algorithm in parallel; partial occluded vehicles is separated into different color tracking. If vehicles have the same color, NP is failed to handle the occlusion and then resorts to the SF level. On the SF level, occlusion masks are created, and the occluded vehicles are tracked by the occlusion reasoning model.

The quantitative evaluation shows that 85.3 % of partial occlusions can be correctly tracked. Serious and full occlusions can be handled efficiently on the SF level. However, the proposed framework failed to handle the same color vehicle occlusion due to only color information as the observation model in the mean-shift algorithm. The future work is to use further features to handle the occlusion tracking when the same color vehicles occluded.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (Grant No. 60302018) and Tianjin Sci-tech Planning Projects (14RCJFJX00845). The authors would like to thank Shuo Shi, Cuihong Xue, Yan Wang and Hongjiao Yang for their help and valuable suggestions to improve the presentation of the paper.

References

1. Pang, C.C.C., Yung, N.H.C.: A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images. *IEEE Trans. Intell. Transp. Syst.* **8**(3), 441–459 (2007)
2. Jian, W., Xia, J., Chen, J.-M., Cui, Z.-M.: Adaptive detection of moving vehicle based on on-line clustering. *J. Comput.* **6**(10), 2045–2052 (2011)
3. Mei, X., Ling, H.B.: Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2259–2272 (2011)
4. Gopalan, R., Chellappa, R.: A learning approach towards detection and tracking of lane markings. *IEEE Trans. Intell. Transp. Syst.* **13**(3), 1088–1098 (2012)
5. Fang, W., Zhao, Y., Yuan, Y.L., Liu, K.: Real-time multiple vehicles tracking with occlusion handling. In: *Proceedings of 6th International Conference on Image and Graphics*, Hefei, Anhui, pp. 667–672 (2011)
6. Zhao, R., Wang, X.G.: Counting vehicles from semantic regions. *IEEE Trans. Intell. Transp. Syst.* **14**(2), 1016–1022 (2013)
7. Prioletti, A., Trivedi, M.M., Moeslund, T.B.: Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation. *IEEE Trans. Intell. Transp. Syst.* **14**(3), 1346–1359 (2013)
8. Feris, R.S., Zha, Y., Pankanti, S.: Large-scale vehicle detection, indexing, and search in urban surveillance videos. *IEEE Trans. Intell. Transp. Syst.* **14**(1), 28–42 (2012)
9. Manz, M., Luettel, T., Hundelshausen, F., Wuensche, H.: Monocular model-based 3D vehicle tracking for autonomous vehicles in unstructured environment. In: *Proceedings of International Conference on Robotics and Automation*, Shanghai, China, pp. 2465–2471 (2011)

10. Andriyenko, A., Roth, S., Schindler, K.: An analytical formulation of global occlusion reasoning for multi-target tracking. In: Proceedings of IEEE Conference on Computer Vision Workshops, Barcelona, pp. 1839–1846 (2011)
11. Gao, T.: Automatic Stable Scene based Moving Multitarget Detection and Tracking. *J. Comput.* **6**(12), 2647–2655 (2011)
12. Weng, M.Y., Huang, G.C., Da, X.Y.: A new interframe difference algorithm for moving target detection. In: Proceedings of 3th International Congress on Image and Signal Processing, Yantai, China, pp. 285–289 (2010)
13. Jian, W., Xia, J., Chen, J.-m., Cui, Z.-m.: Moving object classification method based on SOM and K-means. *J. Comput.* **6**(8), 2045–2052 (2011)
14. Qing, M., Hoang, W.D., Jo, K.H.: Localization and tracking of same color vehicle under occlusion problem. In: Proceedings of Mecatronics-REM, Paris, France, pp. 245–250 (2012)
15. Wang, L.F., Yan, H.P., Wu, H.Y., Pan, C.H.: Forward–backward mean-shift for visual tracking with local-background-weighted histogram. *IEEE Trans. Intell. Transp. Syst.* **14**(3), 1480–1489 (2013)
16. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, SC, pp. 13–15 (2000)
17. Tian, G., Hu, R.M., Wang, Z.Y., Zhu, L.: Object tracking algorithm based on mean-shift algorithm combining with motion vector analysis. In: Proceedings of Education Technology and Computer Science, Wuhan, Hubei, China, pp. 987–990, March 2009
18. Barnich, O., Droogenbroeck, M.V.: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.* **10**(6), 1709–1724 (2011)
19. Mandellos, N.A., Keramitsoglou, I., Kiranoudis, C.T.: A background subtraction algorithm for detecting and tracking vehicles. *Expert Syst. Appl.* **38**(3), 1619–1631 (2011)
20. Huang, J.Y., Hu, H.Z., Liu, X.J., Liu, L.J.: Research on recognition of motional vehicle based on second-difference algorithm. In: Proceedings of IEEE International Symposium on Industrial Electronics, Seoul Korea, pp. 292–296 (2009)
21. Chen, C., Zhu, F.H., Ai, Y.F.: A survey of urban traffic signal control for agent recommendation system. In: Proceedings of 5th International IEEE Conference on Intelligent Transportation System, Anchorage, AK, pp. 327–333 (2012)
22. Frao, A., Giordano, D.: Adaptive background modeling Integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *IEEE Trans. Intell. Transp. Syst.* **12**(4), 1398–1412 (2011)
23. Zhang, W., Wu, Q.M.J., Yang, X.K., Fang, X.Z.: Multilevel framework to detect and handle vehicle occlusion. *IEEE Trans. Intell. Transp. Syst.* **9**(1), 161–174 (2008)