An Improved Surendra Algorithm for Moving Object Detection

Fang Dai
School of Science
Xi'an University of Technology
Xi'an Shaanxi 710054, China
daifang@xaut.edu.cn

Dan Yang
School of Science
Xi'an University of Technology
Xi'an Shaanxi 710054, China
danyang1346@163.com

Tong Dang
School of Science
Xi'an University of Technology
Xi'an Shaanxi 710054, China
dangdang_tong@163.com

***Abstract*—A vehicle detection algorithm developed by Surendra (called Surendra algorithm) is composed of three parts: segmentation, adaptive background updating and background extraction. Surendra algorithm is sensitive to dynamic environment and is easily influenced by noise and illumination when detecting moving objects. For this reason, we present an improved Surendra algorithm (called Surendra_αInst algorithm), in which frame-difference method is applied to calculate the motion mask by replacing the method of using the Boolean AND operator between two binary images which is derived by two adjacent frames subtracted the background and thresholded to a binary image, respectively. Then the motion mask is employed to calculate the instantaneous background. At the same time, according to the change rate of background pixels, the background update coefficient is calculated to obtain a stable background image. Experiments results on five different types of image sequences showing that our Surendra_αInst algorithm, compared with Surendra algorithm, Surendra_AvgInit algorithm and Surendra_α algorithm, has higher DR and lower FAR, and the moving object detected is more integrated.**

Keywords—Moving object detection; Surendra algorithm; background update coefficient; motion mask

I. INTRODUCTION

With the development of computer vision, real time detection of moving objects in video is the basis of object tracking and behavior analysis, which has been widely used in intelligent surveillance system, intelligent traffic control system and military field, etc.

The popular moving object detection methods are among optical flow, frame difference method and background subtraction, etc. Optical flow method [1] detects the moving region in the image sequence by using the vector feature of the moving objects. It works well in the case of background motion. However, the computation is complex and time-consuming.

The motion characteristics of the image sequence are analyzed by the absolute value of difference between two adjacent frames to detect moving objects in frame difference method [2]. The algorithm is simple, and has good robustness, but the results are easily affected by the noise, and it is easy to produce "void". In contrast, the background subtraction method [3] is the most common method of moving object detection, in which the moving region is detected by the difference between the current frame and background. It is simple and easy to implement, besides, it can extract the object more completely. But it is very sensitive to changes in external environment. Therefore, the key point is to establish a real-time updating background model to reduce the impact of dynamic scene changes on motion segmentation.

The popular algorithms to model and update the background are Multi-frame average method [4] and Gaussian mixture modeling [5]. Multi-frame average method is simple in operation, but the background accuracy is low, which is not conducive to real-time updates. The background image of the Gauss-mixture model has higher accuracy, but the computation is time-consuming and the background image update speed is slow. Surendra algorithm [6] also can be used to model and update the background. But the algorithm initials the background with image containing moving objects, which may prone to ghost phenomenon. And the segmentation threshold is determined according to the histogram distribution of corresponding difference image, which cannot adapt to the change of scene. Meanwhile, in the process of background update, the background is updated with the current frame, and the background update coefficient empirically determined. This is easy to introduce foreground information into the current background, which may lead to noise in the detection results. And the algorithm has lower accuracy in complex environment.

For the ghost problem, Songlin W et al. [7] proposed an improved algorithm based on the gradient update rate algorithm to improve the problem that the background update rate is too large in the initial part of the video. However, this method needs to set some parameters, which has a certain limitations.

For that segmentation threshold of Surendra algorithm is based on the histogram distribution of difference image, and it is sensitive to noise and illumination, J Qinghua et al. [8] propose iterative method adaptively calculate thresholds of segmentation and background extraction. However, this method is time-consuming. Mingyang Y [9] combines the two frame difference method with the Surendra algorithm, and use adaptive circle segmentation algorithm to calculate thresholds, but it is sensitive to illumination and occlusion. D Feng et al. [10] use the multi-frame average method to obtain the initial background and selects OTSU method to calculate the threshold adaptively. Meanwhile they applied it to perimeter intrusion detection. When the forbidden area is set, if someone intrudes it, alarm will be triggered. However, it is sensitive to illumination and noise, and the detection accuracy is low in complex environment.

For the drawback of background update coefficient α is taken empirically, Weiyu C et al. [11] set α according to the result of frame-difference, when needs to speed up the background update, α takes a larger value, conversely, α value is smaller. But the calculation method is not specific. The calculation formula is given by Teng Long [12], so the α calculated by this formula can better reflect the changes in the background (called Surendra_α algorithm). However, the α values calculated by this method are too large, and the values of the α in some locations is beyond [0, 1], which leads to more foreground information added to current background when updating background.

For the disadvantage of the Surendra algorithm has great dependence on background in calculating the motion mask in the background update, and the calculation value of α is too large in [12], we propose an improved Surendra algorithm (called Surendra_αInst algorithm), which mainly includes two aspects: 1) an newly calculation formula of α, which is different from the formula in [12]. 2) In the process of adaptive background update, the motion mask is calculated by frame difference method, instead of the Boolean AND operator between two binary image.

The rest of this paper is organized as follows. In Section 2, the Surendra algorithm is introduced and our Surendra_αInst algorithm is described in Section 3, which is the main part of this paper. Experimental results and conclusions are presented in Section 4 and Section 5, respectively.

II. Surendra Algorithm

Surendra algorithm calculates instantaneous background according to the motion mask which is derived by Boolean AND operator between two adjacent frames subtracted the background and thresholded to a binary image. At those pixels that correspond to foreground objects (where the mask is 1), latest background is sampled to update the background. Otherwise the current image is sampled to calculate the background. Its specific process is as follows [6].

1) Build the initial background image B_0;

2) The following steps are performed on ith frame I_i, i=1, 2, ... ,N, N is the number of video frames.

Step 1: Generate the motion mask by background subtraction. Compute the binary object masks OM_{i-1} and OM_i from update interval i-1 and i, respectively. Motion mask MM_i is obtained by (3).

$$OM_{i-1}(x,y)=\begin{cases}1, & I_{i-1}(x,y)-B_{i-1}(x,y)\geq T_1\\ 0, & I_{i-1}(x,y)-B_{i-1}(x,y)<T_1\end{cases} \quad (1)$$

$$OM_{i}(x,y)=\begin{cases}1, & I_{i}(x,y)-B_{i-1}(x,y)\geq T_2\\ 0, & I_{i}(x,y)-B_{i-1}(x,y)<T_2\end{cases} \quad (2)$$

$$MM_i(x,y)=\sim OM_{i-1}\ \&\ OM_i \quad (3)$$

Step 2: Calculate the instantaneous background according to the motion mask with the following expression.

$$IB_i(x,y)=\begin{cases}I_i(x,y), & MM_i(x,y)=0\\ B_{i-1}(x,y), & MM_i(x,y)=1\end{cases} \quad (4)$$

Step 3: Adaptive updates the background. The current background is set to be the weighted average of the instantaneous background and previous background.

$$B_i(x,y)=\alpha IB_i(x,y)+(1-\alpha)B_{i-1}(x,y) \quad (5)$$

Where, $\alpha\in[0,1]$ is a background update coefficient, which is suggested to take the value of 0.1.

Step 4: Calculate the foreground image by background subtraction. Moving object detection is to extract moving objects from the current frame image. After obtaining the background image B_i , the current frame I_i and foreground are calculated by background subtraction, then the difference image can be obtained by the threshold segmentation:

$$DB_i(x,y)=\begin{cases}1, & |I_i(x,y)-B_i(x,y)|\geq T_3\\ 0, & |I_i(x,y)-B_i(x,y)|<T_3\end{cases} \quad (6)$$

Where, the threshold T_i (i=1,2,3) in (1), (2) and (6) is given by the corresponding pixel value of searching toward increasing pixel intensity for a location on the histogram of their difference image that is 10% lower than the peak value, which starts from the pixel value corresponding to the peak of the histogram.

III. Surendra_αInst Algorithm

A. Background Initialization

Surendra algorithm adopts an image which was taken during the day and containing the object as the initial background to detect the moving objects in the video taken at dusk. It is easy to produce ghost, and the initial background is not convenient to obtain. The first frame of the video or the multi-frame average [10] is considered as the initial background in our algorithm. Fig. 1 shows the detection results of two background initialization methods in partial frame of "AVG-TownCentre" video.

In Fig. 1, (a) is the different frame of the original sequence, (b) shows the detection results of Surendra algorithm which take the first frame of the video as initial background, (c) shows the detection results of Surendra algorithm which initial background with multi-frame average (25 frames). Comparing the detection results in Fig. 1, we can see that when take the first frame of the video as initial background, there obviously appear false detection (such as the green mark position in Fig. 1), and with more noise. However, when the background is initialized by the multi-frame average method, the detection result is less noisy and more accurate. To a certain extent, it effectively solves the problem of ghost. Therefore, the multi-frame average method [10] is applied to initialize the background in this paper.

This work was supported by Shaanxi science and technology research projects (No. 2015GY004)

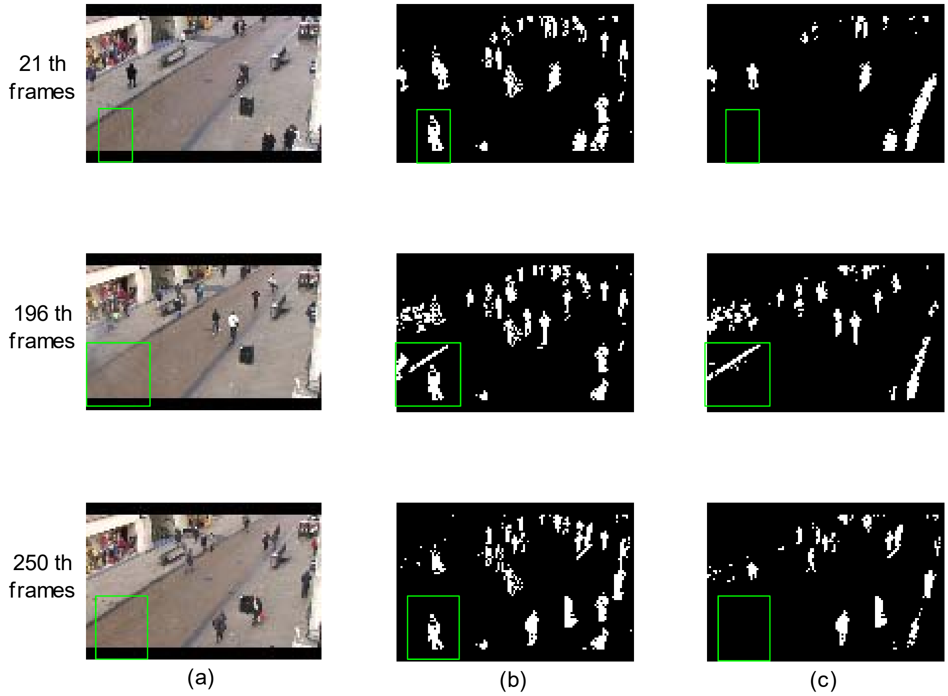

Fig. 1. Moving object detection results with different initial background.

B. Background Update Coefficient

The background update coefficient is vital in the process of updating the background, which should reflect the changes of background. Based on this, we present a new method for calculating the coefficient of background updating. That is, whether the current background pixel location needs to be updated or not depends on the current background change rate at this location.

When the current background is updated with (5), for i=1, 2, ..., N (N is the number of video frames), if the background change rate $\rho_i(x,y) = |I_i(x,y) - B_{i-1}(x,y)|^{-1}$ is large, and then α takes a larger value at location (x, y). At this point, a large amount of current frame information is required. Therefore, we construct (7) to characterize the relationship between ρ_i and α_i in this paper.

$$\alpha_i(x,y) = \frac{1}{1+e^{\beta_i(x,y)}},\ \beta_i(x,y) = \rho_i(x,y)^{-2} \quad (7)$$

Where, $\rho_i(x, y) = 0$ indicates that the gray values of the background pixels remain unchanged. The change of α_i reflects the change of the background pixels. When α_i becomes smaller and the background update rate slows down accordingly, and when the α_i increases, the real-time update of the background pixels is also fast, which meets the real-time requirements.

Because the Surendra algorithm and its improved algorithm, such as Surendra_AvgInit algorithm [10] and Surendra_α algorithm [12], are all in the case of stationary camera, so we need to reduce the foreground information to add in the background update. That is, it is necessary to reduce the value of the coefficient from [0.5, 1] to [0, 0.5]. We select 68th frames of “viptraffic” video in MATLAB database to compare our Surendra_αInst algorithm with Surendra_α algorithm in the calculation of α_i at different positions in same frame. At the same time, the relationship between the background change rate and the position of the image is given, as shown in Fig. 2 and 3.

Fig. 2 shows the relationship between the background change rate and the pixels position (only shown one row of the image), and Fig. 3 shows the relationship that the calculation of α_i by Surendra_α algorithm and Surendra_αInst algorithm change with the position of the image, respectively.

Fig. 2 shows that most of the background change rates are in [0.2, 0.5]. When the background is updated, most of the background update coefficients are close to each other, only a small number of them take a large value.

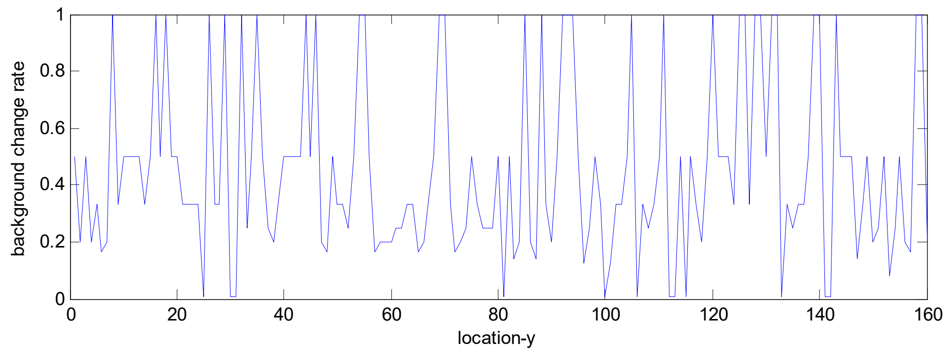

Fig. 2. Relationship between the background change rate and the pixels position.

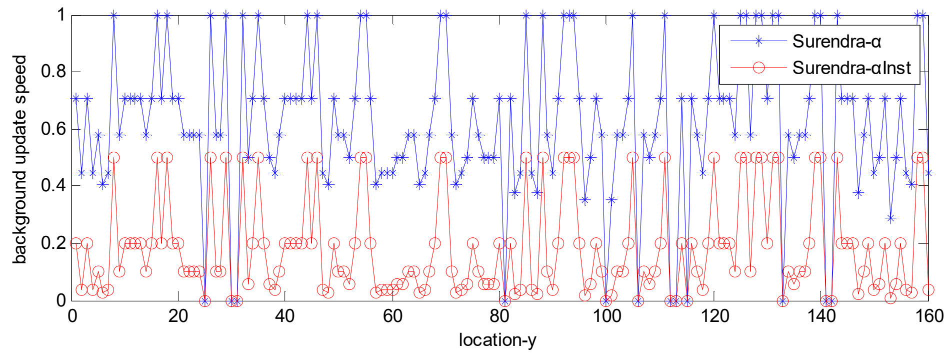

Fig. 3. α_i calculated by Surendra_α and Surendra_αInst algorithm.

In Fig. 3, most values of Surendra_α algorithm are in [0.4, 0.7], and there are many values of 1. It means that amount of current frame information is required to update the background, and lead to introduce the noise into the detected moving region. On the contrary, most of the values of α_i is calculated by (7) are in [0, 0.2], indicating that the current background is highly reliable and does not need to be updated. For the location where the background changes, the calculated values of α_i are around 0.5, it adds appropriate amount of current frame information to the background update, which makes the updating background more accurate.

C. Motion Mask

The motion mask determines which image should be sampled during the background update. At those location where the motion mask is 0 (corresponding to the background pixels), the current frame is sampled. At those location where the motion mask is 1 (corresponding to the foreground pixels), the previous background is sampled. The accuracy of the motion mask directly affects the detection results of moving objects.

Surendra algorithm takes (3) to calculate the motion mask, which is easily affected by external conditions, such as the change of illumination and weather. It makes the extracted motion mask noisy, which affects the detection results. However, the frame difference method is not sensitive to the change of illumination and scene, it can adapt to a variety of dynamic environment with a good stability. Therefore, we employ the frame difference method to calculate the motion mask. The specific process is as follows:

1) Calculate the difference image between current frame I_i and previous frame I_{i-1} to obtain the motion mask MM_i:

$$MM_i(x,y)=\begin{cases}1, & |I_i(x,y)-I_{i-1}(x,y)|\geq T_4\\ 0, & |I_i(x,y)-I_{i-1}(x,y)|<T_4\end{cases} \tag{8}$$

2) Get instantaneous background according to motion mask:

$$IB_i(x,y)=\begin{cases}I_i(x,y), & MM_i(x,y)=0\\ B_{i-1}(x,y), & MM_i(x,y)=1\end{cases} \tag{9}$$

3) Update the background with the weighted of the instantaneous background and previous frame background:

$$B_i(x,y)=\alpha_i(x,y)IB_i(x,y)+(1\text{-}\alpha_i(x,y))B_{i-1}(x,y) \tag{10}$$

Where, $\alpha_i(x, y)$ is the weight coefficient of ith frame at location (x, y).

4) Obtain moving area by background subtraction. After obtaining the background image B_i, the current frame I_i and foreground are calculated by the background subtraction, and then the moving objects can be obtained by the threshold segmentation.

$$DB_i(x,y)=\begin{cases}1, & |I_i(x,y)-B_i(x,y)|\geq T_5\\ 0, & |I_i(x,y)-B_i(x,y)|<T_5\end{cases} \tag{11}$$

Where, T_4 and T_5 is calculated by OTSU threshold method.

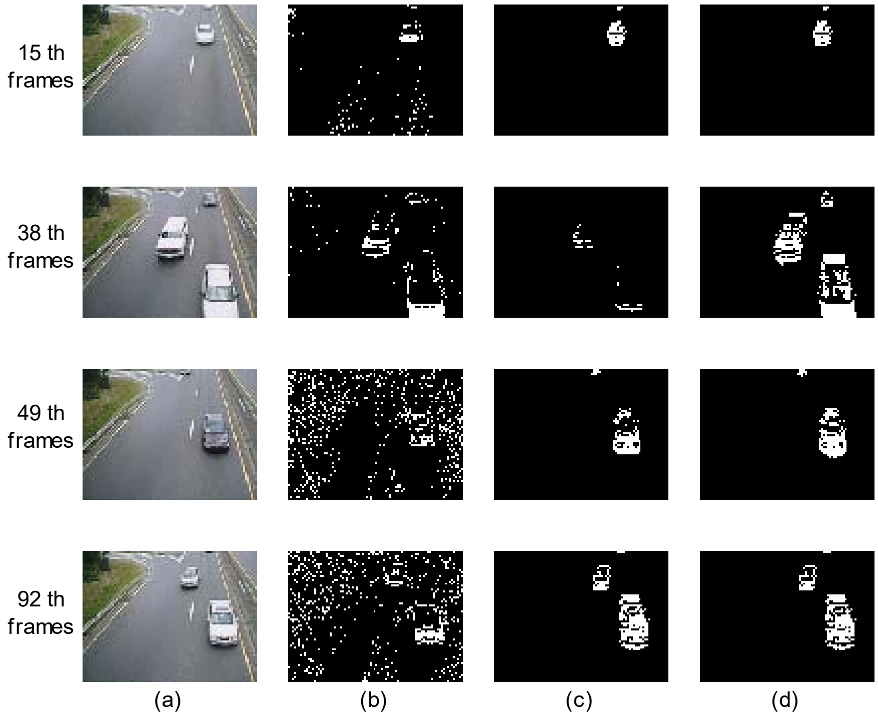

Fig. 4. Motion mask calculated by Surendra algorithm and our algorithm.

We select part of frames of "viptraffic" video in MATLAB database to test the calculation of motion mask (see Fig. 4).

In Fig. 4 (a) shows different frame of the original sequence, (b) motion mask of Surendra algorithm, (c) motion mask of our Surendra_αInst algorithm with Surendra's thresholding method, and (d) motion mask of our Surendra_αInst algorithm with OTSU threshold method. When the motion mask in Fig. 4(b) and (c) are compared, the motion mask calculated by our algorithm is more completely and less contaminated with noise, which verifies the effectiveness of frame difference method to calculate the motion mask.

In order to compare the threshold segmentation method of Surendra algorithm and our algorithm, we contrast the motion mask of Fig. 4(c) with Fig. 4(d). Comparison shows that the motion mask of our algorithm which takes OTSU method to calculate the segmentation threshold is more reliable, for that its object is more completely and close to the real scene with no miss or incomplete object. So we take the OTSU threshold method to compute T_4 and T_5 in (11).

IV. Experiment Results

In order to verify the effectiveness of our algorithm, the experimental results of our Surendra_αInst algorithm and Surendra algorithm [6], Surendra_α algorithm [12] (give the formula to calculate α) and Surendra_AvgInit algorithm [10] (initial the background with multi-frame average) are compared and analyzed. The three outdoor videos are selected as the test sequence, which are the "AVG-TownCentre" video in Multiple Obeject Tracking Benchmark Test Set, the video "viptraffic" in MATLAB database and the video "house" in [13], respectively. The experimental results are shown in Fig. 5 to 7.

In Fig. 5 to 7, (a) shows different frame of the original sequence, (b) detection results of Surendra algorithm, (c) detection results of Surendra_α algorithm, (d) detection results of Surendra_AvgInit algorithm, and (e) detection results of Surendra_αInst algorithm.

From Fig. 5, we can see that the Surendra_AvgInit algorithm is sensitive to the environment and noise, the detection results contain so many noise that almost impossible to identify the object. Surendra algorithm, Surendra_α algorithm and Surendra_αInst algorithm all can extract the whole object area completely, but Surendra algorithm and Surendra_α algorithm obviously appear false objects (such as the mark position at 76th and 196th frame). Comparison shows that, Surendra_αInst algorithm can detect the complete objects area with not false and missed objects, which provides reliable guarantee for the following object tracking and identification.

Fig. 6 shows that Surendra algorithm is sensitive to dynamic environment and there are missed objects in 1012th and 1030th frame. Surendra_α algorithm is sensitive to illumination. Although the object can be detected in some frames, but it contains more noise, especially in 1012th, 1030th frames where objects cannot be detected due to the noise. Surendra_AvgInit algorithm can detect the object more completely and the noise is less, but some frame cannot detect the object for the impact of illumination change (such as the mark position in Fig. 6). In contrast, our algorithm can adapt to the change of illumination, and the detection result contains less noise. Moreover, the moving objects detected by our algorithm are closer to the real situation.

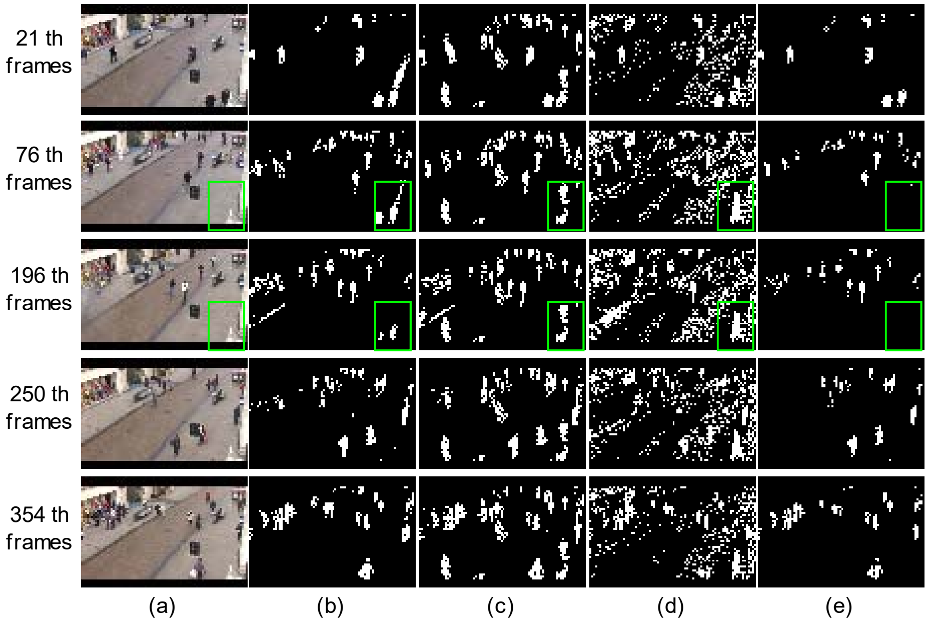

Fig. 5. Moving objects detection results of partial frames in "AVG-TownCentre" video.

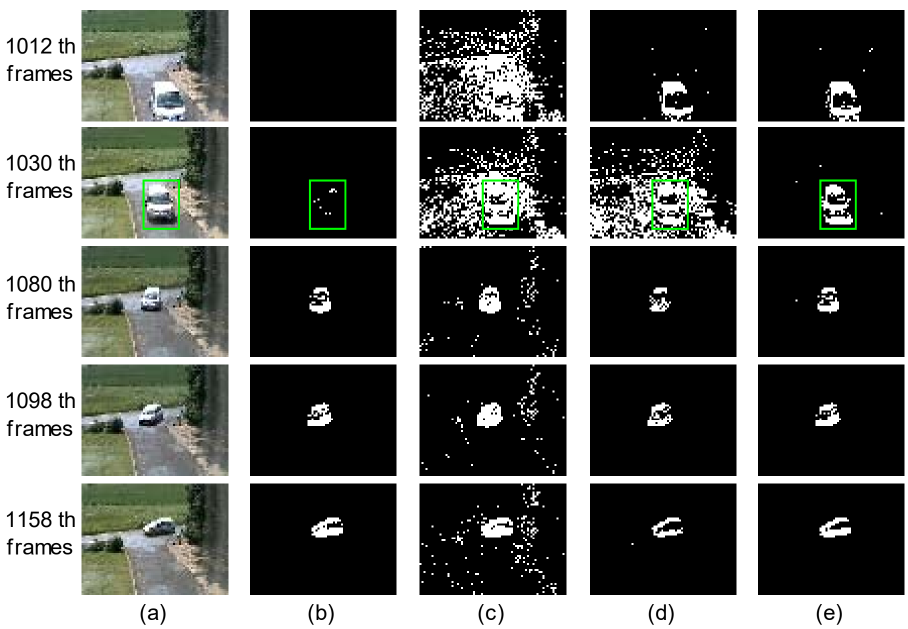

Fig. 6. Moving objects detection results of partial frames in "house" video.

In Fig. 7, the detection result by Surendra algorithm appears obviously as missed objects in 15th and 68th frame. There are also some objects partially detected. Surendra_α algorithm, Surendra_AvgInit algorithm and Surendra_αInst algorithm all can extract the whole object area completely, but the detection results by Surendra_AvgInit algorithm contain noise. Surendra_α algorithm and Surendra_AvgInit algorithm have good robustness to environmental changes, but it all has false objects appearance (such as the small object in 49th frame). In contrast, although the object detection by Surendra_αInst algorithm is not complete when the color of the object and background is close, but there will be no more false and missed detection.

The metric of detection results most widely used is detection rate (DR) and false alarm rate (FAR), which are calculated by (12).

$$DR = \frac{TP}{TP + FN} \quad FAR = \frac{FP}{TP + FP} \tag{12}$$

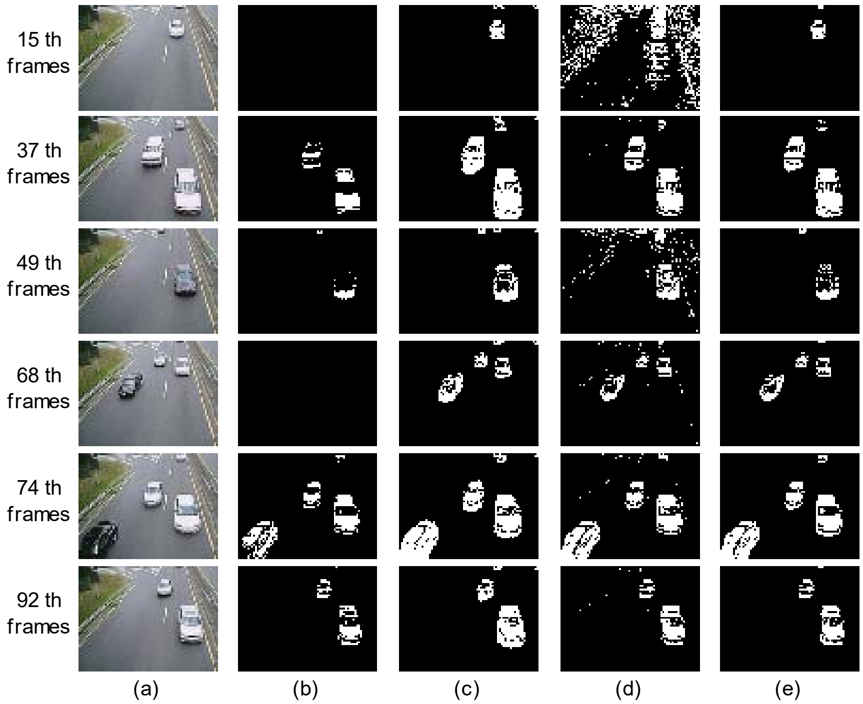

Fig. 7. Moving objects detection results of partial frames in "viptraffic" video.

Among them, TP and FN are the number of correctly detected and not detected moving object, respectively. FP represents number of nonmoving objects detected by error. Table 1 shows the average DR and average FAR of the four algorithms (Surendra algorithm, Surendra_α algorithm, Surendra_AvgInit algorithm and Surendra_αInst algorithm) in the experiments of 100 different scenes.

TABLE I. THE AVERAGE DR AND AVERAGE FAR OF FOUR ALGORITHMS

Name	***Surendra***	***Surendra_α***	***Surendra_AvgInit***	***Surendra_αInst***
DR	70.46%	85.69%	74.34%	91.57%
FAR	25.27%	34.53%	5.51%	1.24%

The above experimental results show that four algorithms can effectively detect moving objects in the term of DR. The DR of Surendra_αInst algorithm is the best, which is high to 91.57%. However, in the case of FAR, four algorithms are obviously different. The FAR of Surendra algorithm and Surendra_α algorithm is as high as 25.27% and 34.53%, respectively. Surendra_AvgInit algorithm and Surendra_αInst algorithm has been greatly improved. Compared with the other three algorithms, our algorithm can adaptively extract and update the background, and can effectively detect moving objects, with the features of high DR and low FAR.

For that our Surendra_αInst algorithm works well in the image sequence with stationary cameras, we did same experiments for moving cameras. The detection results of Surendra_αInst algorithm and of other algorithms are shown in Fig. 8 and 9.

The "car_1" is a video recorded by a non-stationary camera. Fig. 8 shows that Surendra_AvgInit algorithm (shown in Fig. 8(d)) is vulnerable to environmental interference, resulting in excessive noise in extracted the moving objects, and almost impossible to distinguish the objects. Surendra algorithm (shown in Fig. 8(b)) and Surendra_α algorithm (shown in Fig. 8(c)) have better robustness to environmental changes, but detection results in 6th, 87th and 230th frame containing more noise. In contrast, Surendra_αInst algorithm (shown in Fig. 8(e)) can detect the completely moving object, and the extracted object is less affected by noise. In the 6th frame, Surendra_αInst algorithm can extract the vehicle, and Surendra algorithm and Surendra_α algorithm can hardly detect moving objects. However, in the 850th frame, our algorithm and other improved algorithms all appear false detection.

The difference between "car_2" and "car_1" video is that the "car_2" video is shot under the condition of strong light, and the surrounding environment changes greatly, this kind of video is very challenging in object detection. Fig. 9 shows that the four algorithms are almost impossible to detect the moving object due to the impact of light. The detection results of Surendra_AvgInit algorithm (shown in Fig. 9(d)) are worst. In the 900th and 3650th frames, the four methods cannot detect the object, but the Surendra_αInst algorithm (shown in Fig. 9 (e)) is less affected by the noise. In 230th and 450th frames, Surendra algorithm (shown in Fig. 9(b)), Surendra_α algorithm (shown in Fig. 9(c)) and Surendra_αInst algorithm, although affected by the environment and the detection results contain more noise, but it also can extract the objects. The results show that our Surendra_αInst algorithm has the least noise, and the detection results are the best.

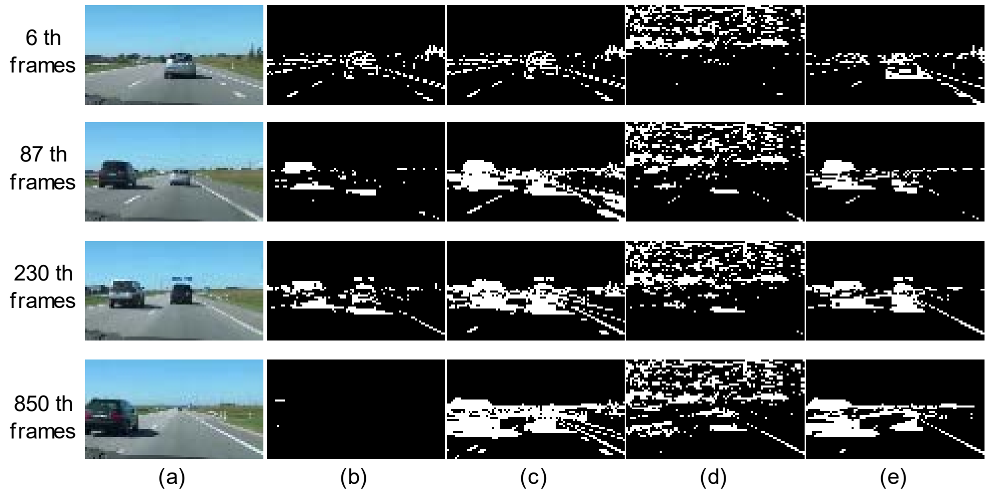

Fig. 8. Moving objects detection results of partial frames in “car_1” video.

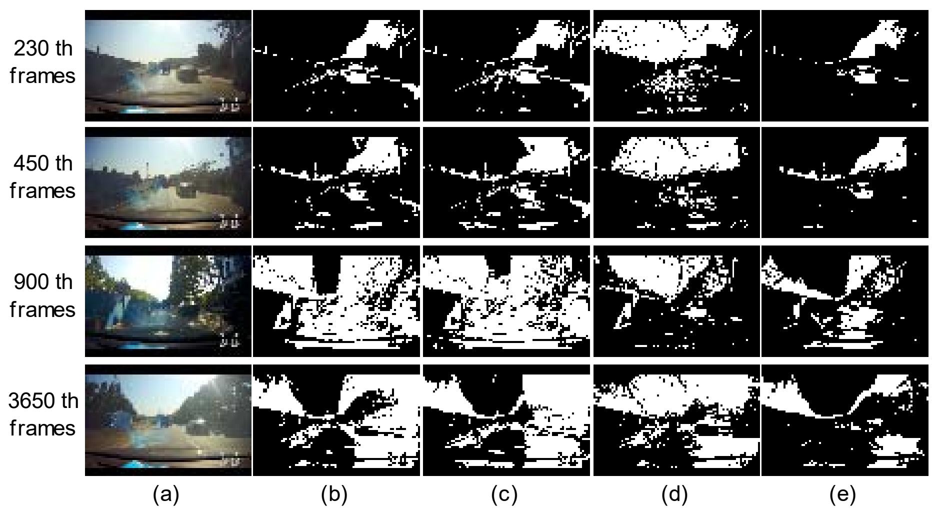

Fig. 9. Moving objects detection results of partial frames in “car_2” video.

V. Conclusions

The Surendra_αInst algorithm is proposed to improve the problem that Surendra algorithm is sensitive to the dynamic environment and is easily affected by the noise and illumination change in this paper. In the process of background updating, the calculation formula of the background update coefficient is given, and the instantaneous background is calculated according to the motion mask that is derived by frame-difference method, which can better detect the moving objects. The experimental results show that our proposed algorithm can detect the moving objects more completely, and it is less affected by noise, and the detection results are better than the other three algorithms for videos taken by stationary cameras. However, in the case of image sequence with moving camera, our algorithm and other three improved algorithms are all affected by the environment. The detection results are unsatisfactory. The next work is to explore an effective method for moving object detection with moving camera, and improve the accuracy of object detection.

Acknowledgment

This work was supported by Shaanxi science and technology research projects (No. 2015GY004).

References

[1] Barron J L,Fleet D J, and Beauchemin SS, “Performance of optical flow techniques,” International Journal of Computer Vision, IEEE, vol.12, no. 1, pp. 42-77, Jun. 1992.

[2] Meyer D, Denzler J, and Niemann H, “Model based extraction of articulated objects in image sequences for gait analysis,” International Conference on Image Processing IEEE Computer Society, IEEE, vol.3, pp.78, Oct.1997.

[3] Gang Li. “New moving target detection method based on background differencing and coterminous frames differencing,” Chinese Journal of Scientific Instrument, vol.27, no.8, pp.961-964, Aug.2006.

[4] Xin W, and Yin XC, “A moving object detection method based on background reconstruction,” Microcomputer Information, vol.24, no.10, pp.284-286, Oct. 2008.

[5] Lee D S, Hull J J, and Erol B, “A Bayesian framework for Gaussian mixture background modeling,” International Conference on Image Processing, IEEE, Vol. 2, pp.III-973-6, Sept. 2003.

[6] Surendra Gupte, et al., "Detection and Classification of Vehicles," IEEE Transactions on Intelligent Transportation Systems, IEEE, vol.3, no.1, pp.37-47, Aug. 2002.

[7] Songlin W, Haodong L, and Taifang W, "Moving Target Detection Algorithm Research Based on Background Subtraction Method," 2013 3rd International Conference on Multimedia Technology (ICMT), Atlantis Press, pp.1179-1186, Nov. 2013.

[8] Qinghua J, and Suping Y, "Object detection algorithm based on Surendra background subtraction and four-frame difference," Computer Application and Software, vol.31, no.12, pp.242-244, Dec. 2014.

[9] Mingyang Y, "A moving objects detection algorithm in video sequence," International Conference on Audio, Language and Image Processing, IEEE, pp.410-413, Jul. 2015.

[10] Feng D, Wang H, and Zhong H,Yu L, "Perimeter intrusion detection based on improved Surendra background update algorithm," International Conference on Digital Image Processing, vol.8009, No.4, pp.357-366, 2011.

[11] Weiyu C, and Liming M, "Mixed method of moving target detection for image sequence," China Academic Journal Electronic Publishing House, pp.329-334, 2009.

[12] Teng L, "Detection algorithm of improved moving objects based on Surendra," Electronic Design Engineering, Vol.21, no.14, pp.151-153, Jul. 2013.

[13] Olivier Barnich, and Marc Van Droogenbroeck, "VIBE: A powerful random technique to estimate the background in video sequences," International Conference on Acoustics, Speech and Signal Processing, IEEE, pp. 945-948, Apr. 2009.