This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

An Adaptive Background Modeling Method for Foreground Segmentation

Zuofeng Zhong, Bob Zhang, *Member, IEEE*, Guangming Lu, Yong Zhao, and Yong Xu, *Senior Member, IEEE*

***Abstract*—Background modeling has played an important role in detecting the foreground for video analysis. In this paper, we presented a novel background modeling method for foreground segmentation. The innovations of the proposed method lie in the joint usage of the pixel-based adaptive segmentation method and the background updating strategy, which is performed in both pixel and object levels. Current pixel-based adaptive segmentation method only updates the background at the pixel level and does not take into account the physical changes of the object, which may result in a series of problems in foreground detection, e.g., a static or low-speed object is updated too fast or merely a partial foreground region is properly detected. To avoid these deficiencies, we used a counter to place the foreground pixels into two categories (illumination and object). The proposed method extracted a correct foreground object by controlling the updating time of the pixels belonging to an object or an illumination region respectively. Extensive experiments showed that our method is more competitive than the state-of-the-art foreground detection methods, particularly in the intermittent object motion scenario. Moreover, we also analyzed the efficiency of our method in different situations to show that the proposed method is available for real-time applications.**

***Index Terms*—Foreground segmentation, background modeling, adaptive background updating.**

I. Introduction

FOREGROUND detection is a critical step for many video processing applications, such as object tracking [1], [2], visual surveillance [3], [4], and human-machine interface [5]. It is always applied as preprocessing for high-level video analyses including pedestrian detection [6], [7], person counting [8], abandoned object detection [9], and traffic surveillance [10]–[13].

The basic idea of foreground detection is to obtain a binary map that classifies the pixels of video frame into foreground and background pixels. In other words, it provides a binary classification of the pixels. The background subtraction is no doubt the first choice to achieve this goal. It extracts the background from the current frame and regards the subtraction result as foreground. Therefore, the background model is crucial for the foreground detection. For a constrained environment, simple background model might be effective. However, this model is hard to be extended for complex cases, because simple background model is not workable under dynamic background or illumination changes.

Background modeling [14], [15] is a process of representing the background under illumination and object changes. A good background model should accurately detect the object shape, and simultaneously remove the shadow as well as the ghost. Moreover, a good background model should be flexible under different illumination conditions, such as a light switched on/off and sunrise/sunset. It should also be robust to different scenes including indoor and outdoor scenes. Besides, it is of great importance of the background model to accurately extract the moving objects which have similar color as the background and the motionless objects. The task of background modeling inevitably faces to an initialization problem, namely the first several frames normally contain the moving objects, which decreases the effectiveness of background modeling and leads to false detection. For surveillance applications, the background subtraction method is required to run in real-time.

Toyama [16] suggested that the background modeling unnecessarily tries to extract the semantic information of the foreground objects, because it has post-process steps. Therefore, most of the background modeling methods operate separately on each pixel. In this way, the shape of a foreground object can be obtained and kept for a short time. But the detection results should not only be spatially accurate, but also be temporarily stable, which means that some foreground regions should remain in the scene for a sufficiently long time, and some other should be quickly absorbed into the background. Current background modeling methods cannot perform very well in the above two aspects. The conventional solution is to keep the balance between the updating speed and the completeness of the shape. A good background modeling method should process the frames at both pixel level and blob level. Moreover, it is necessary for the background modeling method to maintain stable shape of a foreground object and adapt to the illumination

Manuscript received July 22, 2015; revised January 18, 2016; accepted July 30, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61370163, Grant 61300032, and Grant 61332011, and in part by the Shenzhen Municipal Science and Technology Innovation Council under Grant JCYJ20140904154645958. The Associate Editor for this paper was Q. Ji.

Z. Zhong and G. Lu are with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: zfzhong2010@gmail.com; luguangm@hit.edu.cn).

B. Zhang is with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: bobzhang@umac.edu.mo).

Y. Zhao is with the Mobile Video Networking Technology Research Center, Shenzhen Graduate School, Peking University, Shenzhen 518055, China (e-mail: zhaoyong@szpku.edu.cn).

Y. Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, China (e-mail: yongxu@ymail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2597441

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

and object changes. So far as we know, only a few works, e.g., W4 [3] and sample consensus background modeling method (SACON) [17], focus on the background modeling in both pixel and blob levels. Although these methods can obtain more complete object shape when the object is motionless or moves in low-speed, the extracted blob does not always contain all the pixels of the object, which leads to some parts of the object exist too long or disappear too fast.

The pixel-based adaptive segmentation (PBAS) [18] method detected the foreground objects by separately using adaptive thresholds for each pixel. The method adapted very well to different illumination changes. But the procedure which distinguishes the illumination and object changes is deficient. Thus, motionless or low-speed objects may be quickly absorbed in the background, or the regions of detected objects may have "holes." This can slow down the background updating speed to get a more complete shape of the detected object. However, it may result in another problem, namely the noise or incorrect detected regions cannot be rapidly removed. In this paper, we present a new background modeling method based on the framework of the PBAS method. We propose an adaptive background updating method that works at both the pixel level and object level. The proposed method can simultaneously tackle the background changes due to illumination and object changes. We set a counter to control the updating time of the neighbor-pixels of the current background pixel. It can retain the complete shape of the foreground objects after the objects appear in the scene. We designed another method that can clear incorrect foreground pixels which are caused on the background initialization stage. The proposed method has excellent performance in motionless or low-speed motion object scenarios. We evaluated the proposed method on the Change Detection Challenge Dataset and several traffic video of the i-Lids dataset. The experimental results showed our method can achieve promising performance, in comparison with most state-of-the-art methods.

The remainder of this paper is organized as follows: We introduce related foreground segmentation methods and the details of the pixel-based adaptive segmentation method in Section II. In Section III, we give a detailed explanation and analysis of the proposed method. Section IV shows the experimental results compared with other foreground detection methods. We conclude the paper in Section V.

II. Related Works

A. Overview of the Background Modeling Methods

Over the past decades, lots of algorithms were proposed to tackle the problem of foreground segmentation. Several excellent surveys [16], [19]–[21] introduced the field of foreground segmentation. Piccardi [16] stated that a good background modeling method should adapt to sudden light changes, high frequency foreground objects, and rapid motion changes. So a sophisticated background model is an appropriate choice, because a simple background model always assumes that the background is fixed. The foreground object is obtained simply by the difference between the current frame and the background. The W4 [3] model is a simple background modeling method. It models each background pixel by the maximum and minimum intensity values, and the maximum intensity difference between consecutive frames of the training stage. Although it works well in a constrained indoor environment, it fails to detect a foreground object when the background changes.

To construct a complex background model, Pfinder [5] used a simple Gaussian distribution to model the pixels at fixed locations over a time window. This model can adapt to gradual or slight background change, but is not workable if the background has a multi-modal distribution. Therefore, to overcome the disadvantage of the single-modal distribution model, several multi-modal distribution models were proposed. Wallflower [22] used a linear Wiener filter to train and predict background models. The model is effective in a periodically changing environment. When the background dramatically changes, the method may fail to predict the background changes. The intelligent methods are also used for the background modeling. In [40], Maddalena *et al.* explore a self-organizing neural network for background model learning.

The most famous multi-modal background modeling method is the Gaussian Mixture Model (GMM) [1], [2]. The distribution of the pixels is represented by a mixture of weighted Gaussian distributions. The background model can update the parameter of Gaussian mixtures via an iterative scheme. It can obtain good results when the background consists of non-stationary objects, such as leaves or flags. The GMM model can satisfy many practical situations. This statistic-based method for background subtraction still attracts many researchers [23]–[25]. However, when the background includes too many modes, a small number of Gaussians models are not sufficient to model all background modes. Moreover, the GMM also needs to choose an appropriate learning rate to obtain good performance.

In literatures [26], [27], the observed background values of each pixel over time are constructed as a codebook. The code words comprise the pixel variation. However, it is still vulnerable under complex environment. An improved codebook method [28] which uses the temporal and spatial information of the pixels was proposed to enhance the practicability. The codebook method can capture the background variation over a long time period, but cannot process a persistent changing object. Guo *et al.* [29] explores a multilayer codebook model in background subtraction method. The method can detect the moving object rapidly and remove most of dynamic background.

Recently, the subspace methods such as Robust Principle Component Analysis (RPCA) methods have made great progress on moving object detection [30]. RPCA explores the assumption that the low-rank background pixels and the sparse foreground objects can decompose to the foreground objects from the video frame matrix [31]–[33]. It is widely studied in literatures [35], [36]. Zhou *et al.* proposed a detected contiguous outliers in the low-rank representation (DECOLOR) method [34] for object detection and background learning by a single optimization process. In [37], the authors proposed a three-term low-rank matrix decomposition (background, object, and turbulence) method to detect the moving objects with the purpose of tolerating the turbulence. Wen *et al.* [38] proposed a unified framework to integrate the statistical features and subspace method for background subtraction. They believed that the performance of moving object subtraction can be improved by considering the advantages from both types of methods. With the same idea, the Independent Component Analysis is applied to foreground segmentation [39]. It assumes that the foreground and background of an image are independent components, and it can train a de-mixing matrix to separate the foreground and background. This method can rapidly adapt to sudden illumination changes.

As a non-parametric method, the sample consensus (SACON) background modeling method [17] employs color and motion information to obtain the foreground objects. It constructs the background model by sampling a history of the N observed images using the first-in first-out strategy. The background model of the SACON method can adapt to complex scenarios, such as inserted background objects, slow motion objects, and lighting changes.

Instead of the background model updating rule of the SACON method, the universal background subtraction algorithm (ViBe) [41] updates the background by a random scheme. It is regarded as a non-parametric model. Moreover, ViBe updates the background pixels by diffusing the current pixel into neighboring pixels via a different random rule. The adaptability of ViBe is powerful for most scenarios.

B. The Pixel-Based Adaptive Segmentation Method

ViBe initializes the background model using only the first frame and the threshold for foreground segmentation is fixed. This limits the adaptability of ViBe. PBAS was proposed to improve ViBe. PBAS incorporates the ideas of several foreground detection methods and control system theory, and is a non-parametric background modeling method. Following the basic idea of ViBe, PBAS also uses the history of N frames to construct a background model. For the background pixels and its neighboring ones, they will be updated with a random scheme. Unlike ViBe, PBAS initializes the background model using the first N frames, and classifies the foreground pixel using the dynamic threshold which is estimated for each pixel. Moreover, the adjustable learning rate lying in PBAS can control the speed of background updating. The diagram of PBAS is presented in Fig. 1.

From Fig. 1, it can be seen that the algorithm has two important parameters: the segmentation decision threshold $R(x_i)$ and background learning rate $T(x_i)$. We define the background model $B(x_i)$ at pixel x_i as $B(x_i) = \{B_1(x_i), \ldots, B_k(x_i), \ldots, B_N(x_i)\}$ which presents an array of N observed values at pixel x_i. Pixel x_i is classified as the foreground pixel according to

$$F(x_i) = \begin{cases} 1 & \#\{\text{dist}(I(x_i), B_k(x_i)) < R(x_i)\} < \#_{\min} \\ 0 & \text{else} \end{cases} \quad (1)$$

where $F(x_i) = 1$ means that pixel x_i is a foreground pixel, and $F(x_i) = 0$ means that x_i is a background pixel. $I(x_i)$ is the pixel value of pixel x_i. The distance threshold

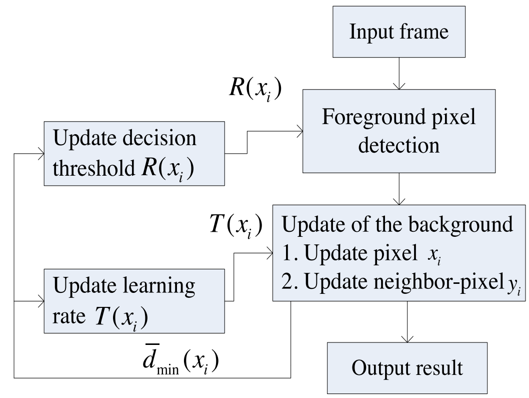


Fig. 1. Diagram of the PBAS method.

$R(x_i)$ can be dynamically changed at each pixel over time. $\#\{\text{dist}(I(x_i), B(x_i)) < R(x_i)\}$ is defined as the numbers of the pixels located at x_i when the distance between pixel value $I(x_i)$ and background value $B_k(x_i)$ is less than $R(x_i)$, and threshold $\#_{\min}$ is predefined and fixed. Since the dynamic changes of the background at each frame, $R(x_i)$ needs to automatically adjust as follows:

$$R(x_i) = \begin{cases} R(x_i) \cdot (1 - R_{\text{inc/dec}}), & \text{if } R(x_i) > \bar{d}_{\min}(x_i) \cdot R_{\text{scale}} \\ R(x_i) \cdot (1 + R_{\text{inc/dec}}), & \text{else} \end{cases} \quad (2)$$

where $R_{\text{inc/dec}}$ and R_{scale} are fixed parameters. $\bar{d}_{\min}(x_i)$ is defined as $\bar{d}_{\min}(x_i) = 1/N \sum_k \min(I(x_i), B_k(x_i))$, and is an average of N minimal distances between pixel value $I(x_i)$ and background pixel value $B_k(x_i)$ at pixel x_i . So the change of $R(x_i)$ is determined by $\bar{d}_{\min}(x_i)$.

The other parameter is the background learning rate $T(x_i)$ which controls the speed of the background absorption. A large $T(x_i)$ means that a foreground object will be merged into the background quickly. The method defines the updating rule of the learning rate $T(x_i)$ as follows:

$$T(x_i) = \begin{cases} T(x_i) + \frac{T_{\text{inc}}}{d_{\min}(x_i)}, & \text{if } F(x_i) = 1 \\ T(x_i) - \frac{T_{\text{dec}}}{d_{\min}(x_i)}, & \text{if } F(x_i) = 0 \end{cases} \quad (3)$$

where T_{inc} and T_{dec} are fixed parameters. They are independently set to increase or decrease $T(x_i)$. Furthermore, the method defines an upper bound T_{upper} and lower bound T_{lower} to prevent $T(x_i)$ from exceeding the normal range. When $T(x_i)$ is larger than T_{upper} or smaller than T_{lower}, the PBAS makes $T(x_i) = T_{\text{upper}}$ or $T(x_i) = T_{\text{lower}}$ respectively. In fact, the method does not directly employ the learning rate $T(x_i)$, but randomly updates the background pixels with probability $p = 1/T(x_i)$. The lower the $T(x_i)$ is, the higher the p will be, which also means that the pixel will be updated with higher probability.

III. THE PROPOSED METHOD

A. Motivation

According to previous discussion, PBAS determines the foreground objects pixel-by-pixel, and updates the background

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

Fig. 2. Example of the effect of the values of $T(x_i)$. (a) Frame of the video. (b) Ground truth. (c) Result of $T(x_i) = 1$. (d) Result of $T(x_i) = 100$.

at each pixel. It does not take into account the spatial and temporal relationship of the foreground pixels belonging to different objects. In other words, the pixel-based updating method cannot adapt to physical changes of foreground objects. The variation of the learning rate $T(x_i)$ is another factor that affects the completeness of the shape of a detected object. The detected regions (including a lighting change or object region in the same frame) are all affected when we adjust the learning rate. When the learning rate is high, the method can obtain a high quality motion detection result under poor illumination condition. But static objects or low-speed objects are usually quickly absorbed in the background. Moreover, PBAS updates the background by diffusing current pixels to neighboring pixels until the foreground object is completely absorbed, so the diffusion effect may aggravate the foreground object absorption. The reason is that the high learning rate result in the background "eats-up" a small object or some parts of a big object. In order to maintain the completeness of the foreground of a motionless or low-speed motion object, we can assign a small value to the learning rate. But slow background updating results in the fact that incorrect foreground detection or noise cannot be quickly removed.

From the above analysis, the background updating procedure of the PBAS method works only at the pixel-level. It lacks the flexibility for different categories of foreground pixels, and cannot select the appropriate updating strategy for foreground pixels belonging to different objects. Fig. 2 depicts an example of different learning rates. Fig. 2(a) is the source frame, Fig. 2(b) is the ground truth corresponding to Fig. 2(a), (c), and (d) are the detection results when $T(x_i)$ is 1 and 100 respectively. It can be seen that the box near the man is completely absorbed by the background when the learning rate $T(x_i) = 100$. However, for a low learning rate, the foregrounds of the sitting man and box remain exist. In addition, there are "holes" in the foreground regions of the box and man. Obviously, Fig. 2(c) is closer to the ground truth, but the effect of background absorption is still obvious for some parts of the foreground objects. Therefore, the method should keep the balance between the updating speed and the completeness of shape.

B. Description of the Proposed Method

We updated the background models by introducing a selective updating strategy. The background model can be updated at both pixel level and object level. Our updating strategy enables the background to adapt to the changes of object and illumination. The proposed method can rapidly remove the influence of lighting changes, and retain the shape of the foreground object.

Aiming at distinguishing the change of illumination from the change of the object, we constructed a counter (similar to [17]), COM, which counts the times that each pixel is continuously identified as a foreground pixel. For pixel m in the t-th frame, we increased the value of $\mathrm{COM}_t(m)$ by 1 when this pixel is classified as the foreground pixel. Once the pixel is classified as a background pixel, $\mathrm{COM}_t(m)$ is set to zero. The procedure is presented as:

$$\begin{cases} \mathrm{COM}_t(m) = \mathrm{COM}_{t-1}(m) + 1 & \text{if } F_t(m) = 1 \\ \mathrm{COM}_t(m) = 0 & \text{otherwise.} \end{cases} \tag{4}$$

In other words, the value of $\mathrm{COM}_t(m)$ shows the number of frames in which pixel m is continuously marked as the foreground pixel. It implies that pixel m belongs to an object if $\mathrm{COM}_t(m)$ is very large. The maximum of $\mathrm{COM}(m)$ at pixel m is always small when this pixel is in a region with a strong change of lighting, because changes of illumination often cause sudden appearance and disappearance of lighting and shadow. However, for a pixel of an object, particularly a motionless or low-speed motion object, the value of $\mathrm{COM}(m)$ is always sufficiently large. By using an appropriate threshold, we can distinguish the change of a lighting pixel from the change of an object pixel. The designed method starts to update the neighboring pixels of pixel m, when the value of $\mathrm{COM}(m)$ is larger than threshold T_b. The proposed updating process is similar to the neighboring pixels updating process of PBAS, and it used randomly selected neighboring pixels of pixel m to replace the randomly selected background sample pixels of corresponding location [18]. The purpose of this method is to weaken the diffusion effect when the background updates the foreground objects for obtaining the almost complete shape of a foreground object. For the region of illumination changes, however, the maximum of $\mathrm{COM}(m)$ does not always exceed threshold T_b. So the background updating diffusion effect can rapidly remove the region of lighting changes. From our experience, the variance of threshold T_b cannot obviously affect the result. So we can fix it as an appropriate value.

This updating model works well in most cases. However, when the initial frames contain a foreground object, the model cannot adaptively update an incorrect background caused by the initial frames. Fig. 3 shows such an instance. In the video "baseline_highway" of the Change Detection Challenge dataset, a car is emerging in the scene in the beginning of the video. Fig. 3(a) shows a beginning frame which is used to initialize the background model. Fig. 3(b) and (c) present a source image and detection result. It can be seen that the "first car" is still in the result image. This is because the initial background object region is again detected as a foreground object, while in fact, no true object appears in this region at that time. So it can be regarded as a "static object" in the scene. Whether or not an object passes that background object region, the "static object" will be kept in the scene. Even through the values of counter COM of some pixels from that background object region exceed threshold T_b, the diffusion effect of the background updating is not obvious for those pixels. The object background region cannot be updated by a new background. This leads to incorrect detection results for the whole sequence.

(a) (b) (c)

Fig. 3. Example of incorrect foreground object caused by initialization. (a) Beginning frame of the video. (b) Source frame. (c) Detection result.

In order to overcome the above disadvantage, we proposed another background updating strategy. We used a random strategy to regard pixel m whose $\mathrm{COM}_t(m)$ exceeds threshold T_f as a background pixel. The updating process replaces pixel m with a randomly selected background sample pixel, whose strategy is similar to [18]. This means that if a pixel is marked as a foreground pixel for a long time, it may become a new background pixel. This method can remove the incorrect background region which is caused by an initial foreground object, because the "static object" caused by an incorrect background region can be easily updated into the background. The method uses new background pixels to gradually replace the pixels from the incorrect background region. These two updating strategies seem to be contradictory, but in fact they are mutually promoted. The purpose of the previous strategy which updates the neighboring pixel is to weaken the diffusion effect of background updating for obtaining the stable representation of the objects, and the latter one which updates current pixels allows the newly obtained background pixels to be rectify the incorrect background region. Both these updating strategies are object-level strategies. They are integrated with the pixel-level strategy of the PBAS method to generate a hybrid updating method for acquiring better foreground detection accuracy.

Threshold T_f should be larger than T_b. In fact, T_f which controls the time that starts to update the background pixels of an object should be longer than T_b which controls the time that begins to weaken the diffusion effect of background updating for an object. If T_f is less than T_b, our method changes the foreground pixels of an object to the background pixels before the method starts to weaken the diffusion affect of background updating. So the effect of retaining the shape of the object is invalid, and T_b is meaningless. As a result, we should set a larger value of T_f to obtain an ideal result. If T_f is too small, but larger than or equals to T_b, the result of our method is almost the same as that of the PBAS method. The proposed method is summarized in Algorithm 1.

Algorithm 1: An Adaptive Background Updating Algorithm

Input: A frame.
Output: A binary image.
Initialization: First N frames are used to construct the background model. Counter COM is set to 0
Procedure:
1. Pixel m is classified as a foreground pixel or background pixel;
2. If pixel m is classified as a background pixel
   a) replace randomly selected background sample pixel $B_i(m)$ with pixel m, i is a random number;
   b) if $\mathrm{COM}_t(m) > T_b$, randomly select the neighboring pixel p of pixel m and update this pixel into a randomly selected background sample pixel $B_i(p)$ of pixel p, i is a random number;
   c) counter $\mathrm{COM}_t(m)$ is set to 0;
3. If pixel m is classified as a foreground pixel
   a) 1 is added to counter $\mathrm{COM}_t(m)$;
   b) if $\mathrm{COM}_t(m) > T_f$, replace randomly selected background sample pixel $B_i(m)$ with pixel m, i is a random number;

C. A Probabilistic Interpretation for the Proposed Method

From the perspective of probability, we give another interpretation of our background updating strategy. Because this strategy independently operates pixels, we can split the problem of the background pixel updating into a sub-problem of each background pixel updating. To illustrate the reasonability of the proposed method, we present the probability that the updated pixel belongs to either category (illumination or object) for the background pixel updating sub-problem. Because the PBAS method and our method update the background pixel by the same random scheme, we can assume as follows: a pixel is updated with probability $P(A)$, and a neighboring pixel of this pixel is updated with probability $P(B|A)$.

Based on the proposed pixel classification method, we classify the pixels as two categories: ω_1, the pixel belongs to illumination pixels and ω_2, the pixel belongs to object pixels. x represents the event that the pixel is updated. By applying Bayes' rule, the posterior probability $P(\omega_i|x)$ that the pixel which is updated belongs to ω_1 or ω_2 can be written as

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)},\ i = 1, 2 \tag{5}$$

where $P(x|\omega_i)$ is likelihood function which means the updating probability of the pixel belonging to ω_i. Here, we can approximate $P(x|\omega_i)$ with $P(B|A)$. $P(\omega_i)$ is the prior probability that means the pixel belongs to $\omega_i, i = 1, 2$.

The posterior probabilities $P_1(\omega_i|x)$ and $P_2(\omega_i|x)$ of the PBAS method and our method can be rewritten as

$$\begin{aligned} P_k(\omega_i|x) =& \frac{P_k(x|\omega_i)P_k(\omega_i)}{P_k(x)} \\ =& \frac{P(B|A)P_k(\omega_i)}{P_k(x)},\ i = 1, 2; k = 1, 2. \end{aligned} \tag{6}$$

Our method places the pixel into two categories. The classification method leads to a pixel has a higher probability being an illumination pixel than being an object pixel. So we define the prior probabilities as $P_2(\omega_1)$ and $P_2(\omega_2)$ of ω_1 and ω_2 as $P_2(\omega_1) > P_2(\omega_2)$. The posterior probabilities $P_2(\omega_1|x)$ and $P_2(\omega_2|x)$ of ω_1 and ω_2 can be written as

$$P_2(\omega_1|x) = \frac{P(B|A)P_2(\omega_1)}{P_2(x)} > P_2(\omega_2|x) = \frac{P(B|A)P_2(\omega_2)}{P_2(x)}. \tag{7}$$

From the posterior probabilities, we can find that an updated pixel is more likely to belong to the category of illumination pixels rather than object pixels. This means that our method accelerates the updating speed of illumination pixels, and the updating speed of object pixels becomes slower. The updating diffusion effect for object pixel is weakened. So it can keep a stable representation of the object.

Because PBAS processes two categories of pixels in the same way, we can define the prior probabilities $P_1(\omega_1)$ and $P_1(\omega_2)$ of ω_1 and ω_2 as the same (=0.5). We also give the relationship of the posterior probability between PBAS and the method for two categories of pixels [42]. For illumination pixel ω_1, we obtain

$$P_1(\omega_1|x) = \frac{P(B|A)P_1(\omega_1)}{P_1(x)} < P_2(\omega_1|x) = \frac{P(B|A)P_2(\omega_1)}{P_2(x)}. \tag{8}$$

For object pixel ω_2, we have

$$P_1(\omega_2|x) = \frac{P(B|A)P_1(\omega_2)}{P_1(x)} > P_2(\omega_2|x) = \frac{P(B|A)P_2(\omega_2)}{P_2(x)}. \tag{9}$$

It can be seen that the probability of a pixel being an illumination pixel for the proposed method is larger than that for PBAS when this pixel is updated. Simultaneously the probability of an updated pixel being an object pixel for the proposed method is smaller than that for PBAS. This also means that the proposed method can update an illumination pixel faster and retain more complete shape of the object than PBAS.

D. The Relationship With Other Background Updating Methods

All the proposed method, PBAS, and ViBe use nonparametric background pixel updating procedure. They all update the background pixel using random scheme, and simultaneously randomly update the neighboring pixel of the current background pixel. The pixel updating strategies do not need the parameter controlling.

However, the proposed method is different from PBAS and ViBe. As presented earlier, the proposed updating strategy integrates the pixel-level and object-level updating rules. It can select different updating rules for various objects by a classification scheme. However, PBAS and ViBe just update the background pixel-by-pixel. The proposed method contains double updating rules: one rule controls the updating time to remain the completeness of object and removes the illumination changes; another rule deals with the incorrect background region which is caused in background initialization. In other words, we simultaneously employ the updating strategy to deal with the foreground and background. It means that the proposed method can rectify the incorrect detected pixel quickly. However, PBAS and ViBe both exploit the updating rule in the background. Finally, the counting rule of the foreground pixels of the proposed method allows the user to achieve different detection results by adjusting the updating time for different scenes. Moreover, the solo friendly parameter T_f is easier to understand and use.

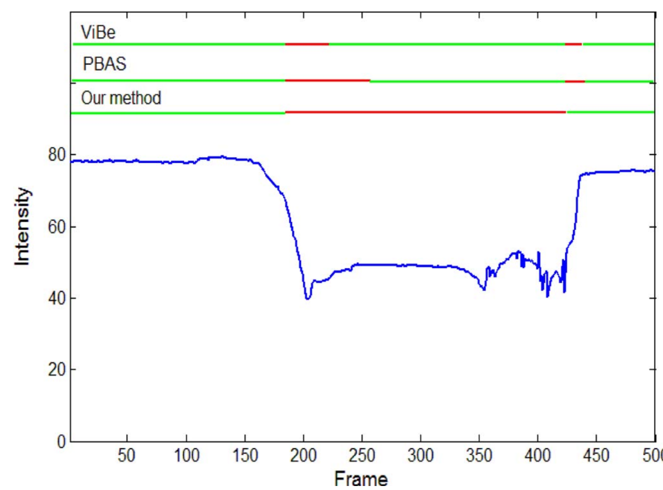


Fig. 4. Comparison analysis of different updating rules.

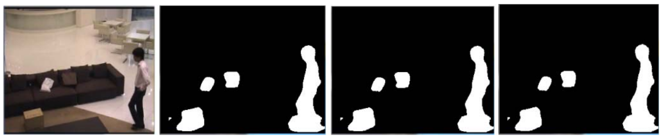
Foreground detection analysis: To analysis the performance of three detection methods, a detection profile of the average pixels of a region from a video is presented in Fig. 4. It shows the average intensities for each frame (blue curve) and corresponding detection results of different methods. The foreground and background detection results are represented with red and green lines respectively. In the Figure, a static object is observed from frame 180 to 440. The proposed method correctly detects the static object until it is removed. However, PBAS and ViBe both fail to detect the static object because they both absorb the object into the background quickly. When the static object is removed, they both fail again. The reason is that the removed object existing in the background is treated as a new foreground.

IV. Experimental Results

In this section, we showed the performance of our method. We first analyzed the influence of parameters, and then present the compared experimental results on two datasets. Finally, we gave the average running time of our method on image sequences of different sizes.

The datasets we used to evaluate our method are outdoor traffic videos from the i-Lids dataset [45] and the Change Detection Challenge 2014 Dataset [44]. We chose four traffic sequences from the i-Lids dataset including PV-Easy, PV-Medium, PV-Hard, and PV-Night as a traffic video dataset. The first three sequences are different traffic videos with complex environment during the day, and the last one is at night. The Change Detection Challenge 2014 dataset has 53 videos of eleven categories including scenarios of indoor and outdoor views with various weathers, night videos, static objects, small objects, shadows, camera jitter, and dynamic backgrounds. Human-annotated benchmarks are available for all videos.

The metric to evaluate the foreground detection methods is to assess the output of the method with a series of the ground-truth segmentation maps. In order to measure the performance of the methods against every output image, we used the following terms: true positive (TP), false positive (FP), true negative



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

Fig. 5. Example of different values of T_b. (a) Source frame. (b) Result of $T_b = 10$. (c) Result of $T_b = 20$. (d) Result of $T_b = 50$.

(TN), and false negative (FN). True positive is the number of correctly detected foreground pixels. False positive is the number of the background pixels that are incorrectly marked as foreground pixels. True negative is the number of correctly marked as background pixels. False negative is the number of foreground pixels incorrectly marked as background pixels [44]. The metrics that we used to quantify the segmentation performance are as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$F - \text{measure} = 2\frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{12}$$

We also used the Percentage of Correct Classification (PCC) to standardize evaluation of detection performance containing both foreground and background pixels [41]. It is calculated as follows:

$$\text{PCC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{13}$$

The foreground detection methods should maximize PCC, because PCC presents the percentage of the correctly classified pixels containing both foreground and background pixels. So when PCC is higher, the performance of the method is better.

The ROC (Receiver Operating Characteristic) and the AUC (Area Under Curve) [47] are also used to evaluate the detection method. The ROC curve is the curve whose x and y axes are the false positive rate (FPR) and the true positive rate (TPR) respectively. The AUC score is the area under the ROC curve.

A. The Determination of the Parameters

In addition to the parameters of PBAS, the proposed method has two parameters, T_b which controls the updating time of the neighbor-pixels and T_f which controls the updating time of the pixel that is marked as a foreground pixel for a long time. To study the influence of each parameter individually, all parameters of PBAS were set as default parameters for all experiments. From our observation, the variation of T_b cannot obviously affect the results. In order words, the stable shape of the foreground object can be kept in the scene for different values. So we fixed the T_b value. There is an example of the effect of different T_b in Fig. 5. Fig. 5(b)–(d) show the detected results where T_b is 10, 20, and 50 respectively. It was observed that the outputs of different values of T_b were almost the same.

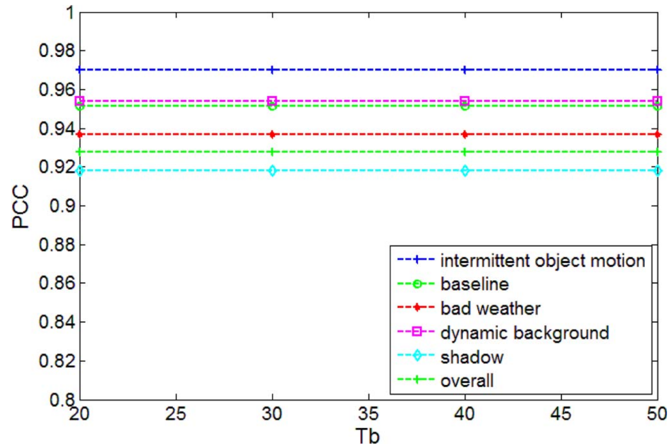


Fig. 6. Variation of T_b and PCC with different scenes.

(a) (b) (c)

Fig. 7. Example of different values of T_f in wet snow scene. (a) Source frame. (b) Result of $T_f = 50$. (c) Result of $T_f = 150$.

(a) (b) (c)

Fig. 8. Example of different values of T_f in a traffic crossing scene. (a) Source frame. (b) Result of $T_f = 50$. (c) Result of $T_f = 150$.

Fig. 6 shows the values of the PCCs while T_b values varied in different scenarios. It can be seen that the PCCs did not vary when the T_b value increases in each scene. In other words, the different PCCs cannot obviously influence the detected results. Because of this, we empirically fixed an appropriate value that was equal to 20 as the T_b value.

However, the T_f value can affect the detected results. We should choose different optimal values for different scenarios. For scenes in which the background rapidly changes, such as the bad weather and camera jitter, we should select a lower value. But for scenes in which the background is relatively stable, especially intermittent object motion scenario, the optimal T_f value is large. Figs. 7 and 8 show two instances of the influences of different T_f values. Fig. 7 presents a wet snow scene, and Fig. 8 shows a traffic crossing scene. In the wet snow scene, Fig. 7(b) and (c) present the results of T_f as 50 and 150 respectively. It is obvious that a lower value is a better choice, because the incorrect foreground pixels caused by the snow should be rapidly updated into the background. In the traffic crossing scene, the appropriate value should be set larger.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

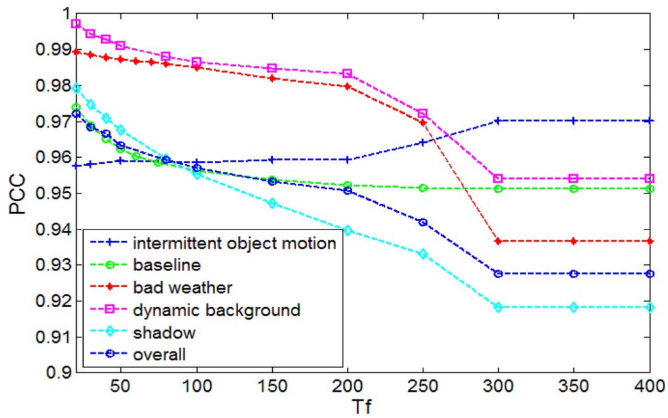


Fig. 9. Variation of T_f and PCC with different scenes.

It was confirmed by the results presented in Fig. 8(c). The high value of 150 can obtain a more stable shape of the stopping car than the low value which here is 50.

We also present the relationship between the PCC and the T_f value. From Fig. 9, it can be seen that variation of the T_f value and PCC is different in various categories of scenarios. For most scenes, the curves of PCC gradually decrease. A large T_f value cannot control the background to adapt to rapid changes of the environment. In intermittent object motion scenes, however, a better value of PCC can be obtained by increasing the value of T_f.

When the T_f value was larger than 300, the detection results almost did not vary. When the T_f value was lower than 30, the results of our method were almost the same as that of PBAS. So, a reasonable range of T_f is from 30 to 300.

B. *Results of the Traffic Video Dataset and Change Detection Challenge 2014 Dataset*

We compared our method to six state-of-the-art foreground segmentation methods, the Gaussian mixture model (GMM) [1], the sample consensus background model (SACON) [17], ViBe [41], the pixel-based adaptive segmenter (PBAS) [18], the background model re-initialization (BMRI) method [43], and DECOLOR [34]. GMM is a pixel-based parametric method and SACON is a pixel and blob-based parametric method. ViBe and PBAS are pixel-based nonparametric methods, and they are the two top state-of-the-art foreground detection methods reported [44]. BMRI is a luminance change detection method. We integrate it with the ViBe method in our experiments. DECOLOR is RPCA-based method.

For GMM, we used the implementation available in OpenCV Library [46]. We adjusted the parameters of it by the suggestion in OpenCV. The programs of ViBe, PBAS, and DECOLOR were provided by the authors of ViBe, PBAS, and DECOLOR respectively. We used the best parameters of two methods suggested by the authors. Because the code of the SACON method was not available, we coded the program ourselves, and selected the optimal parameters following the advices in [17]. To obtain better comparable results, we made some post-processes to the output of the methods. In this paper, we used 3×3 median filtering and the morphological close operation as the post-processes for all methods.

First, we show the experimental results of our method and six foreground detection methods on the traffic video dataset in Fig. 10. We selected two typical frames from each video to represent each video. The first, second and third two rows are PV-easy, PV-Medium, and PV-Hard videos respectively, and last two rows are night videos. Fig. 10(a) shows the original frame of the video, and Fig. 10(b) is the result of our method. Fig. 10(c)–(h) are the results of PBAS, ViBe, GMM, SACON, BMRI-ViBe, and DECOLOR respectively. Visually our method obtained satisfactory results for the videos of different difficulties, including night video. The other six foreground detection methods all missed some minor pedestrians and vehicles, and some incorrect detection objects existed. Even SACON failed to detect foreground objects in night video, because of the strong illumination. This means that our method is suitable for traffic scenes.

We present another comparative experiment on the Change Detection Challenge dataset. In this experiment, we extensively tested the proposed method under various conditions. The scenarios used to evaluate our method contained bad weather, camera jitter, dynamic background, intermittent motion objects, low frame, night, PTZ, shadows, and thermal images. The thermal video was captured by a far-infrared camera. There were several videos for each scenario. We used the same six foreground detection methods used in the previous experiment to compare with our method. The setting of parameters and post-processes were the same as the previous experiment.

Fig. 11 shows the foreground segmentation results of an intermittent object motion video. We selected six frames from the video to show the advantage of our method. Fig. 11(a) and (b) are original frames and ground truth respectively of the frames. Fig. 11(d)–(i) are the results of state-of-the-art foreground detection methods, and Fig. 11(c) shows the result of our method. It can be seen that our method retained the stable shape of the three bags until they are removed. However, all other foreground segmentation methods absorbed parts or whole bags into the background in a short time. Fig. 12 shows results in a traffic crossroad video. We chose four frames from the video. The proposed method could still obtain correct and fuller foreground objects, such as the stopping or low-speed cars. GMM and BMRI-ViBe have incorrect detection object because of the background initialization. Visually, the results of our method looked better than other methods, and were closer to the ground truth.

Table I presents four evaluation metrics of our method on the Change Detection Challenge 2014 dataset. Our method performed well for most scenes, including *baseline, camera jitter, intermittent object motion, night, shade, thermal, and turbulence*. The proposed background updating method could adapt to rapid background changes caused by camera displacement, sudden illumination changes, or a large number of objects in motion. It simultaneously adapted to slow background changes and static objects.

The advantage of the proposed method was confirmed by PCC, recall, precision, and F-measure scores in Table II. It can be seen that the proposed method obtained higher PCC and recall scores. It indicates the proposed method detected

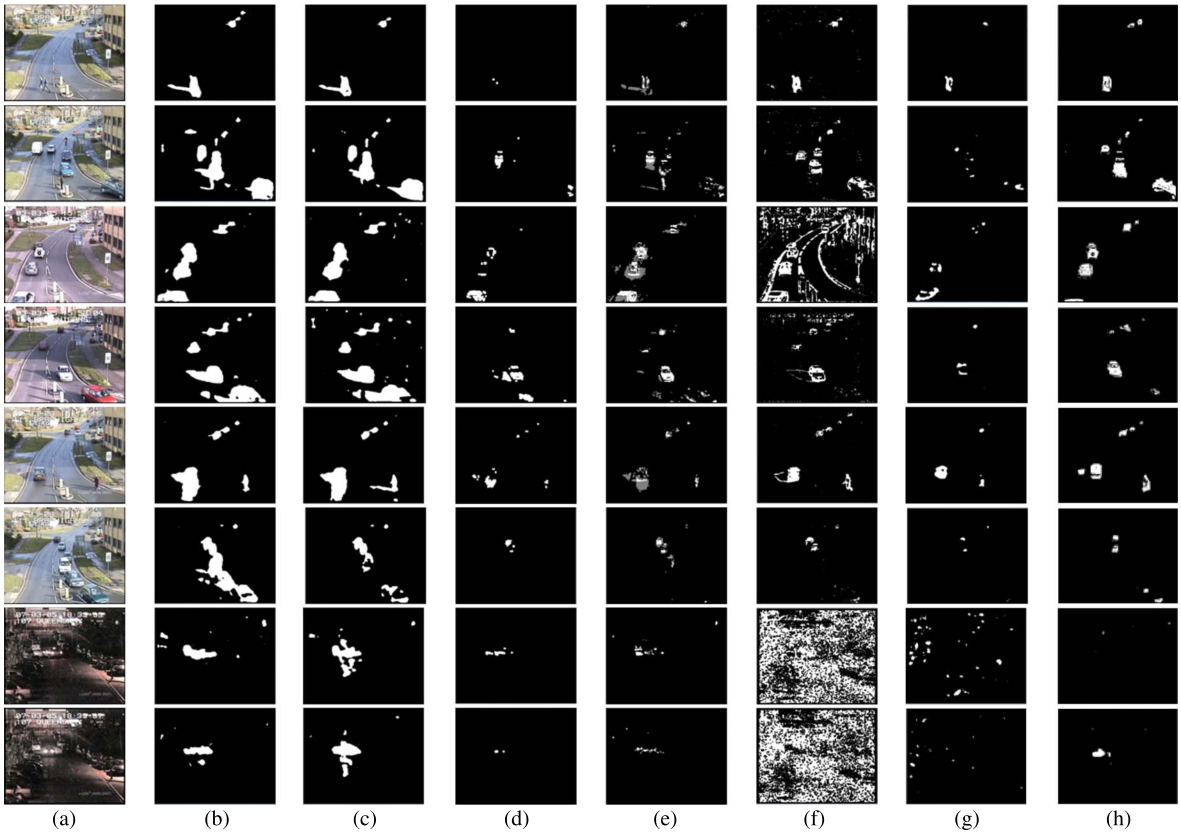


Fig. 10. Foreground detection results of traffic videos. (a) Original frame. (b) Proposed method. (c) PBAS [18]. (d) ViBe [41]. (e) GMM [1]. (f) SACON [17]. (g) BMRI-ViBe [43]. (h) DECOLOR [34].

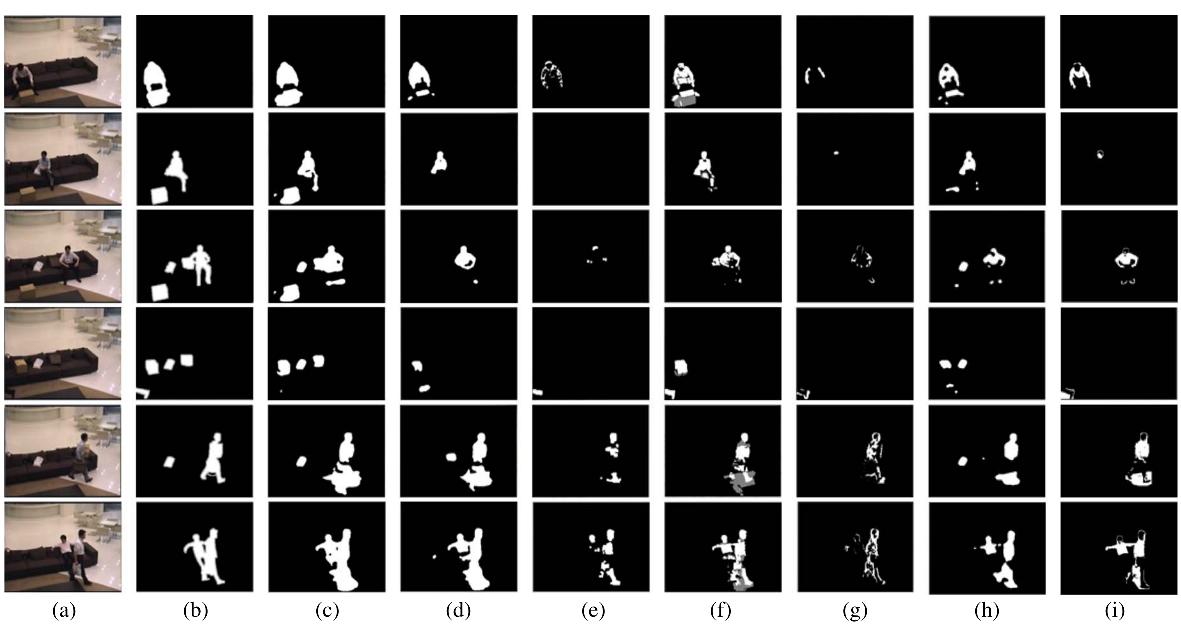
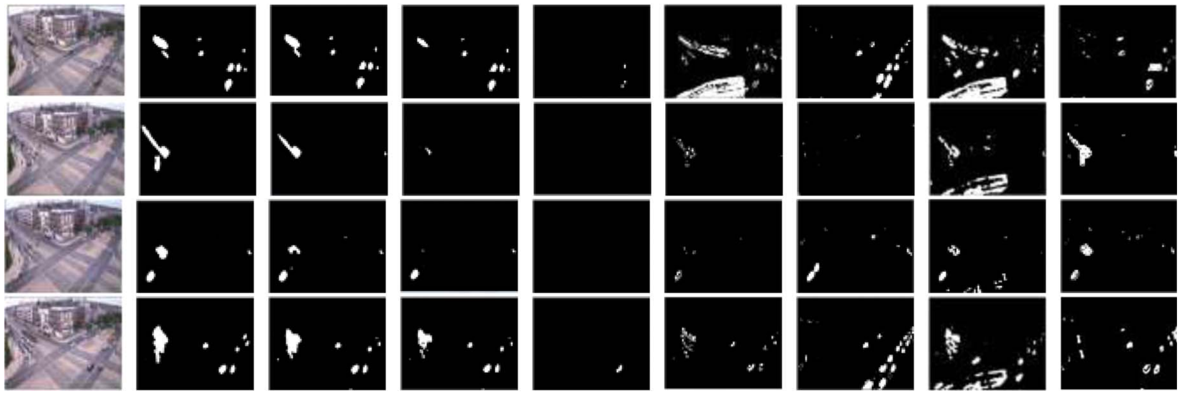


Fig. 11. Foreground detection results of an intermittent object motion video from the Change Detection Challenge 2014 dataset. (a) Original frame. (b) Ground truth. (c) Proposed method. (d) PBAS [18]. (e) ViBe [41]. (f) GMM [1]. (g) SACON [17]. (h) BMRI-ViBe [43]. (i) DECOLOR [34].

more correct foreground and background pixels, and less incorrect pixels. Our method obtained the best F-measure score compared with the two top foreground detection methods (PBAS and ViBe) and RPCA-based method (DECOLOR). The F-measure which joins the recall and precision to evaluate performance showed that our method achieved better global superiority, even when our method did not give the best precision score. For each evaluation metric, we give the compared results for five foreground detection methods in different scenarios in Figs. 13–16. PCC, recall, and F-measure shown in Figs. 13, 14, and 16 all present scores of our method that were almost higher than the others. In Fig. 15, however, the precision

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



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

Fig. 12. Foreground detection results of a crossroad video from the Change Detection Challenge 2014 dataset. (a) Original frame. (b) Ground truth. (c) Proposed method. (d) PBAS [18], (e) ViBe [41]. (f) GMM [1]. (g) SACON [17]. (h) BMRI-ViBe [43]. (i) DECOLOR [34].

TABLE I
AVERAGE EVALUATION METRICS OF THE CHANGE DETECTION CHALLENGE 2014 DATA SET

Scenarios	PCC	Recall	Precision	F-measure
Bad Weather	0.9854	0.6043	0.3114	0.4110
Baseline	0.9603	0.9014	0.8488	0.8743
Camera Jitter	0.9378	0.7904	0.3588	0.4935
Dynamic Background	0.9913	0.6050	0.2001	0.3007
Intermittent Motion Object	0.9747	0.8141	0.8351	0.8244
Low Frame	0.8967	0.6390	0.6448	0.6418
Night Video	0.9204	0.8021	0.5021	0.6175
PTZ	0.8446	0.7224	0.2871	0.4108
Shadow	0.9677	0.9141	0.503	0.6489
Thermal	0.9849	0.8614	0.6283	0.7266
Turbulence	0.9735	0.7100	0.5584	0.6251
Overall	0.9492	0.7603	0.5161	0.6148

TABLE II
COMPARISON OF OUR METHOD WITH FOUR OTHER METHODS ON THE CHANGE DETECTION CHALLENGE 2014 DATA SET

Method	The proposed method	PBAS	ViBe	GMM	SACON	BMRI-ViBe	DECOLOR
PCC	**0.9492**	0.9301	0.9220	0.9239	0.9150	0.9253	0.9447
Recall	**0.7603**	0.6397	0.3072	0.3865	0.3728	0.5816	0.5840
Precision	0.5161	0.4559	**0.6322**	0.5549	0.3890	0.3937	0.5311
F-measure	**0.6148**	0.5323	0.4134	0.4556	0.3807	0.4695	0.5562
AUC	**0.8755**	0.8660	0.7526	0.7942	0.7409	0.8036	0.8547

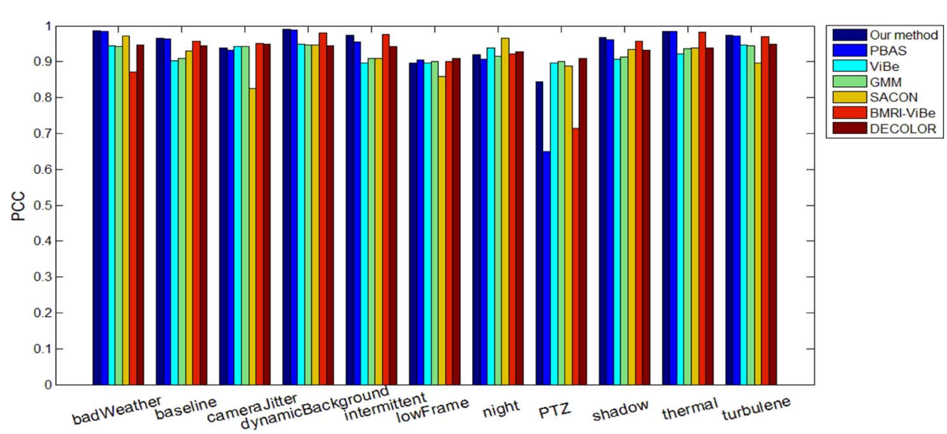


Fig. 13. PCC of different methods.

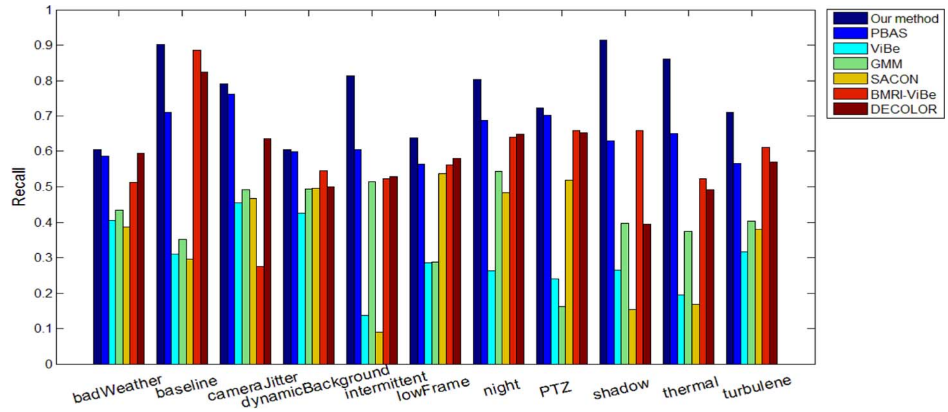


Fig. 14. Recall of different methods.

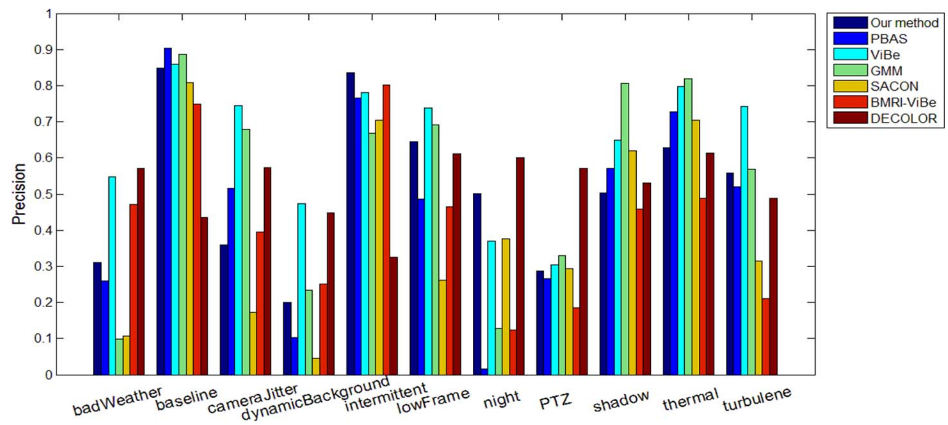


Fig. 15. Precision of different methods.

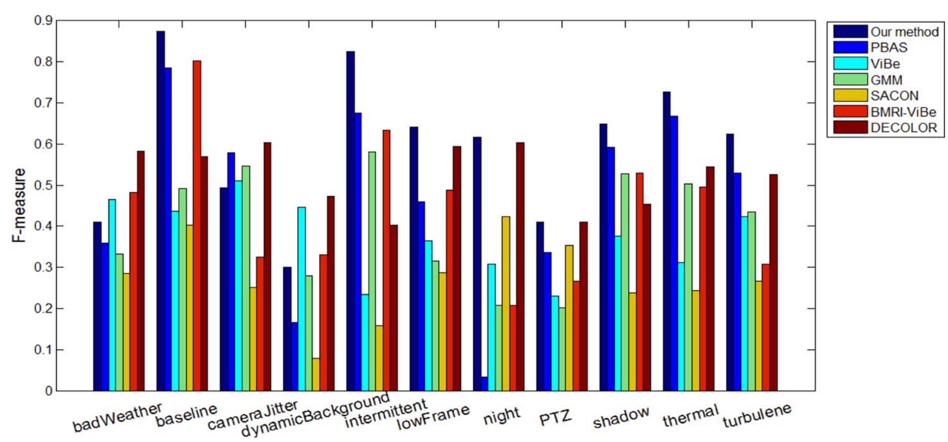


Fig. 16. F-measure of different methods.

of our method had a good performance on some scenarios, such as intermittent objects in motion. Fig. 17 is the ROC curves of all methods. It is observed that the proposed method achieves the best performance. Table II also lists the AUC scores. The proposed method obtains the highest AUC score among all detection methods. This confirms the corresponding ROC curve.

C. Comparison of Average Computing Time

We also compared the processing time of all these methods. We used three videos of different sizes: 320×240, 640×350, and 720×540 to estimate the times. All videos were 25 fps,

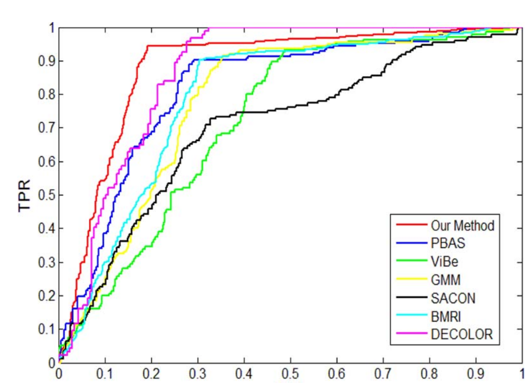


Fig. 17. ROC curve of different methods.

TABLE III
COMPARISON OF AVERAGE FRAMES PER SECOND (FPS)

The size of video	The proposed method	PBAS	ViBe	SACOM	GMM	BMRI-ViBe	DECOLOR
320×240	43.5	41.7	62.5	47.6	**62.5**	59.5	1.12
640×350	18.5	15.1	27	22.2	**58.8**	25.3	0.52
720×540	10.6	8.9	31.2	27	**47.6**	21.7	0.24

and were converted to gray images as input images. Table III shows average frames per second on our computing platform (2.3 GHz Core i5 CUP, 3GB of RAM, C implementation). From the results, we found the average computation speed of the proposed method was faster than PBAS and DECOLOR, but slower than other method for all sizes sequences. But the good detection performance of our method can compensate for the disadvantage of the running time. Moreover, the running time of the proposed method is sufficient to satisfy real-time applications. If the result of the proposed method is similar to that of PBAS, the proposed method has the advantage of running speed.

V. CONCLUSION

In this paper, we proposed a robust and effective background modeling method. The proposed method uses the advantages of the pixel-based adaptive segmentation method. PBAS only updates the background at the pixel-level. So it causes motionless or low-speed motion objects to be absorbed by the background quickly, or partial regions of the foreground objects are neglected. The proposed method adopts a updating strategy that can update the background at the pixel-level and object-level. We constructed a counter to record the times in which a pixel is continuously classified as a foreground pixel for all image pixels. We can control the updating time by using the value of the counter. This updating mechanism can work well in most scenarios. The experimental results show that our proposed method can achieve better results than other methods. Because of the lower computation time, our method can adapt to many real-time applications. In particular, our method can obtain satisfactory performance in urban traffic scenes. However, our method cannot deal with the objects whose color is similar as the background efficiently, because the gray feature cannot well distinguish the object and background. Another unsolved problem of our method is that parameter T_f varies with scenes and the optimal value of T_f is not known. We will explore these issues in future. The texture feature (such as SILTP [48]) should be helpful for improving the robustness of the background model. We also attempt to design a procedure to select an optimal value for different scenes.

ACKNOWLEDGMENT

Thanks to Dr. Edward C. Mignot, Shandong University, for his linguistic advice.

REFERENCES

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.

[2] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[4] S. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 1–14, Jan. 2011.

[5] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Feb. 2012.

[7] F. Xu, X. Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 63–71, Mar. 2005.

[8] Y. L. Hou and G. K. H. Pang, "People counting and human detection in a challenging situation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 24–33, Jan. 2011.

[9] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M. T. Sun, "Robust detection of abandoned and removed objects in complex surveillance videos," *IEEE Trans. Syst., Man, Cybern. C, Appl.*, vol. 41, no. 5, pp. 565–576, Sep. 2011.

[10] N. Buch, S. A. Velastin, and J. A. Orwell, "Review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.

[11] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[12] S. Huang and B. Chen, "Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1920–1931, Dec. 2013.

[13] B. Chen and S. Huang, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 837–847, Apr. 2014.

[14] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques—State-of-art," *Recent Patents Comput. Sci.*, vol. 1, no. 1, pp. 32–54, Jan. 2008.

[15] M. Yang, C. Huang, W. Liu, S. Lin, and K. Chuang, "Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 595–608, Apr. 2015.

[16] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Conf. Syst., Man, Cybern.*, 2004, pp. 3099–3104.

[17] H. Wang and D. Suter, "Background subtraction based on a robust consensus method," in *Proc. IEEE Conf. Pattern Recognit.*, 2006, pp. 223–226.

[18] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 38–43.

[19] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 14, pp. 4–21, May 2014.

[20] Y. Xu, J. Dong, B. Zhang, and D. Xu, "Background modeling methods in video analysis: A review and comparative evaluation," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 43–60, Jan. 2016.

[21] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vol. 11/12, pp. 32–66, May 2014.

[22] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, vol. 1, pp. 255–261.

[23] D. S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[24] Q. Zang and R. Klette, "Robust background subtraction and maintenance," in *Proc. 17th IEEE Int. Conf. Pattern Recognit.*, 2004, vol. 2, pp. 90–93.

[25] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th IEEE Int. Conf. Pattern Recognit.*, 2004, vol. 2, pp. 28–31.

[26] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, vol. 5, pp. 3061–3064.

[27] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, Jun. 2005.

[28] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU-Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, Aug. 2010.

[29] J. Guo, C. Hsia, Y. Liu, M. Shih, C. Chang, and J. Wu, "Fast background subtraction based on a multilayer codebook model for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1809–1821, Oct. 2013.

[30] T. Bouwmans and E. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understand.*, vol. 122, pp. 22–34, May 2014.

[31] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1809–1821, May 2011.

[32] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3047–3064, May 2012.

[33] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, "MahNMF: Manhattan nonnegative matrix factorization," *Mach. Learn.*, vol. 1, no. 5, pp. 11–43, 2012.

[34] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[35] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, Aug. 2015.

[36] Y. Sun, X. Tao, Y. Li, and J. Lu, "Robust 2D principal component analysis: A structured sparsity regularized approach," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2515–2526, Aug. 2015.

[37] O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 450–462, Feb. 2013.

[38] J. Wen, Y. Xu, J. Tang, Y. Zhan, Z. Lai, and X. Guo, "Joint video frame set division and low-rank decomposition for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2034–2048, Dec. 2014.

[39] D. M. Tsai and S. C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, Jan. 2009.

[40] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.

[41] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[42] C. X. Ren, D. Q. Dai, and H. Yan, "Robust classification using ℓ 2, 1-norm based regression model," *Pattern Recognit.*, vol. 45, no. 1, pp. 2708–2718, Jul. 2012.

[43] F. Cheng, B. Chen, and S. Huang, "A background model re-initialization method based on sudden luminance change detection," *Eng. Appl. Artif. Intell.*, vol. 38, pp. 138–146, Feb. 2015.

[44] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 1–8.

[45] i-LIDS Dateset for AVSS 2007. [Online]. Available: ftp://motinas.elec.qmul.ac.uk/pub/iLids

[46] "Open source computer vision library," OpenCV, Nizhny Novgorod, Russia. [Online]. Available: http://www.opencv.org

[47] J. Ke, A. Ashok, and M. Neifeld, "Block-wise motion detection using compressive imaging system," *Opt. Commun.*, vol. 284, no. 5, pp. 1170–1180, Mar. 2011.

[48] H. Han, J. Zhu, S. Liao, Z. Lei, and S. Z. Li, "Moving object detection revisited: Speed and robustness," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 6, pp. 910–921, Jun. 2015.

Zuofeng Zhong received the B.S. degree in computer science and technology from Guangxi Normal University, Guilin, China, in 2007 and the M.S. degree in computer science and technology from Guangdong University of Technology, Guangzhou, China, in 2011. He is currently working toward the Ph.D. degree in computer science and technology at Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China.

His current research interests include image processing and video analysis.

Bob Zhang (M'12) received the B.A. degree in computer science from York University, Toronto, ON, Canada, in 2006; the M.A.Sc. degree in information systems security from Concordia University, Montreal, QC, Canada, in 2007, and the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, ON, in 2011.

After graduating from the University of Waterloo, he remained with the Center for Pattern Recognition and Machine Intelligence, and later worked as a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau, Taipa, Macau. His research interests include medical biometrics, pattern recognition, and image processing.

Guangming Lu received the B.S. degree in electrical engineering, the M.S. degree in control theory and control engineering, and the Ph.D. degree in computer science and engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 1998, 2000, and 2005, respectively.

From 2005 to 2007, he was a Postdoctoral Fellow with Tsinghua University, Beijing, China. He is currently a Professor with the Biocomputing Research Center, Shenzhen Graduate School, HIT, Shenzhen, China. His current research interests include pattern recognition, image processing, and automated biometric technologies and applications.

Yong Zhao received the Ph.D. degree in automatic control and applications from Southeast University, Nanjing, China, 1991.

Then, he joined Zhejiang University, Hangzhou, China, as an Assistant Researcher. In 1997, he went to Concordia University, Montreal, QC, Canada, as a Postdoctoral Fellow. In May 2000, he was a senior Audio/Video Compression Engineer of Honeywell Corporation. Since 2004, he has been an Associate Professor with Peking University Shenzhen Graduate School, Shenzhen, China, where he is also currently the Head of the Laboratory of Mobile Video Networking Technologies. His current research interests include video compression and video analytics, with special focus on applications of these new theories and technologies to various industries.

Yong Xu (M'06–SM'15) received the B.S. and M.S. degrees in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2005.

He is currently with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His current research interests include pattern recognition, biometrics, bioinformatics, machine learning, image processing, and video analysis.