

Background Modeling by Stability of Adaptive Features in Complex Scenes

Dan Yang, Chenqiu Zhao, Xiaohong Zhang[✉], and Sheng Huang, *Student Member, IEEE*

Abstract—The single-feature-based background model often fails in complex scenes, since a pixel is better described by several features, which highlight different characteristics of it. Therefore, the multi-feature-based background model has drawn much attention recently. In this paper, we propose a novel multi-feature-based background model, named stability of adaptive feature (SoAF) model, which utilizes the stabilities of different features in a pixel to adaptively weigh the contributions of these features for foreground detection. We do this mainly due to the fact that the features of pixels in the background are often more stable. In SoAF, a pixel is described by several features and each of these features is depicted by a unimodal model that offers an initial label of the target pixel. Then, we measure the stability of each feature by its histogram statistics over a time sequence and use them as weights to assemble the aforementioned unimodal models to yield the final label. The experiments on some standard benchmarks, which contain the complex scenes, demonstrate that the proposed approach achieves promising performance in comparison with some state-of-the-art approaches.

Index Terms—Multi-feature, background subtraction, motion detection, self-adaptive, histogram.

I. INTRODUCTION

BACKGROUND subtraction has a wide range of applications, including video monitoring [1], optical motion capturing [2] and multimedia applications [3]. With the recent advance in background models [1], background subtraction has become more practical and attracted more attention in the computer vision community. There is no gold-standard

Manuscript received August 29, 2015; revised April 21, 2017; accepted October 24, 2017. Date of publication November 1, 2017; date of current version December 5, 2017. This work was supported in part by the National Natural Science Key Foundation under Grant 91118005, in part by the National Natural Science Foundation of China under Grant 61173131, in part by the Natural Science Foundation of Chongqing under Grant CSTS2010BB2061, in part by the Fundamental Research Funds for the Central Universities under Grant CDJZR12098801 and Grant CDJZR11095501, and in part by the Chongqing Graduate Student Research Innovation Project under Grant CYB14034. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hamid R. Sheikh. (*Corresponding author: Xiaohong Zhang*)

D. Yang, C. Zhao, and S. Huang are with the School of Software Engineering, Chongqing University, Chongqing 401331, China (e-mail: zhaochenqiu@gmail.com).

X. Zhang is with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China, also with the School of Software Engineering, Chongqing University, Chongqing 401331, China, and also with the State Key Laboratory of Coal Mine Disaster Dynamics and Control, Chongqing University, Chongqing 400044, China (e-mail: xhongz@cqu.edu.cn).

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/TIP.2017.2768828

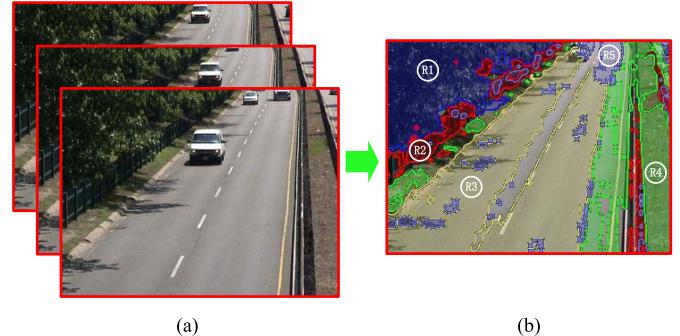


Fig. 1. The illustration of complex scenes are recognized as the mixture of R1, R2, R3, R4 and R5 which characterized by green color, gray, texture, red color and blue color as dominant features, respectively. Different dominant features are utilized for modelling the background in each zone.

definition about what is background [4]. Traditionally, a pixel is labeled as background when the variation of features' value is under a threshold. In this work, we call the degree of the variation as the stability of features. The main challenges in background subtraction comes from dynamic and complex scenes, such as waving tree, spouting fountains and rippling water. A number of methods have been proposed to address these challenges, e.g., [1], [5], which also results in the presentations of large numbers of features for background modeling. However, since the single feature only reflects single characteristics of a pixel while the characteristics of a pixel differ over a time sequence, the single-feature based algorithms are often hard to apply to complex scenes. In order to better exploit the different characteristics of pixels, this paper develops a multi-feature method, which measures the stability of features as the priorities of different features for better segmenting the foreground in complex scenes.

Commonly, a pixel is better described by several features like color, texture and so on. Among the extensive visual features, there should exist at least one dominant feature which depicts the characteristics of the pixels best and this dominant feature should be more stable than the others. Here, we define the most stable feature and the other less stable features as the dominant feature and secondary features respectively. For example, as Fig. 1(a) shows, there is a complex scene consisting of waving trees, a dark fence, a highway and soil. Clearly, the corresponding dominant features of the previous objects are different. More specifically, the dominant feature of the waving tree is the green color while the ones of the soil and the highway are the red color and texture feature respectively like Fig. 1(b) shows. The dominant features are the

most stable feature which are the strong cues of background, since the pixels of the background always remain invariant or change slowly during the changes of external conditions (e.g. illumination change or noise). In this paper, we term the *Stability* as the degree of feature value change over a time sequence.

It is well known that the features of pixels in the background are more stable. Although many researchers have already realized this fact, only very limited attention has been paid on feature stability of these pixels. In order to sufficiently exploit the feature stability of these pixels, we propose the Stability of Adaptive Features (SoAF) background model in this paper. SoAF is an assembled multi-feature background model which is composed by several single-feature background models. Each of these single-feature models also has three components and each component is a unimodal model [6]. The SoAF model is deemed as the weighted combination of the single-feature models. In the assembly step, the stability of each feature is adaptively measured via the histogram statistic. In addition, this learned stability is weighted to the corresponding single-feature model, since a strong stability of a feature is a strong cue being in the background. We apply our work to some standard change detection datasets. The experiments show that SoAF can achieve promising performance.

In summary, there are three desirable properties that have been provided by our approach and proved by comprehensive experiments.

- 1) *Self-Adaptability*: SoAF can adaptively select different dominant features over the pixel and time sequence domains, for example, gray-scale and texture features are selected as dominant features in zones R2 and R1 of Fig. 1 respectively (the details will be explained in Section V-A).
- 2) *Complementarity*: Dominant feature plays a vital role in SoAF, but we also do not neglect the contributions of the secondary features. Actually, we empirically find that the secondary features that are good complementaries of the dominant feature (this part will be explained in Section V-B).
- 3) *Robustness*: In the experiments, SoAF always adaptively selects the dominant feature and the experimental results show that SoAF works very well in complex scenes including waving tree, spouting fountains, rippling water and so on (this part will be explained in Section V-C).

The remainder of this paper is organized as follows: Section II introduces the related works of our proposed approach; we introduce the feature extraction process in Section III; In Section IV, the processes of foreground detection and background updates are presented; The experiments and the experimental results' analysis are demonstrated in Section V. Finally, this paper is briefly summarized in Section VI.

II. RELATED WORK

Recently, many impressive change detection algorithms have been presented. Consequently, a large number of features were utilized for change detection. In this paper, we discuss these algorithms in two sections. The algorithms based on

single features are reviewed in Section II-A, and algorithms based multi-feature is discussed in Section II-B, which our method belongs to.

A. Algorithms Based on Single-Features

We divided the single-feature algorithms into three categories by the features they used. The algorithms based on spectral features (color features), spatial features (edge feature, texture features) and temporal features (motion features).

The first category (e. g., [7]–[14]) employs gray or color to model the background. Among those algorithms, the single Gaussian model [7] was the most popular technique, which models each pixel with a Gaussian distribution. Then, Stauffer and Grimson [8] presented the mixture of Gaussians (MoG) which approximated each background pixel by a K-Gaussians mixture model. Plenty of methods improved upon MoG, and survey [15] gave several examples. Beside the MoG, Codebook and ViBe are also two representative approaches proposed by Toyama *et al.* [16] and Barnich and Van Droogenbroeck [11], [12] respectively. In addition, there also several methods that employed the color information for foreground detection, such as those enhanced from MoG [15], [17], the approaches analyzed the distribution of color features by color histogram statistic [13], [18], [19], and the algorithms utilized the color invariants to detect moving objects [14].

The second category (e.g., [6], [20]–[23]) utilized the texture features for modeling the background. The texture-based approach [6] presented by Heikkilä *et al.* [20] is a classic method which utilized the texture for foreground detection, and several algorithms are inspired by it (e.g., [21]–[23]). For example, Heikkilä *et al.* [21] extracted the texture by SILTP, and utilized kernel density estimation for approximating the multimodal of textures. Yeh *et al.* [22] utilized the multi-level textures for foreground detection, which extended to a hierarchical coarse-to-fine approach to model the background.

The third category is temporal features, which were associated with inter-frame changes at the pixel (e.g., [16], [24]–[26]). Toyama *et al.* [16] proposed the wallflower algorithm and utilized inter-frame changes to maintain an algorithm for solving this canonical problem. Li *et al.* [25] used Bayes decision rule for classifying the background and foreground based on inter-frame color co-occurrence statistics. Wixson [26] detected salient motion through the intermediate-stage vision integration of optical flow. Also, Lin *et al.* [24] extracted a sequence of regular video bricks, which were utilized to cope with disturbances from moving objects and scene changes.

B. Algorithms Based on Multi-Feature

Compared with algorithms based on single feature, the multi-features based algorithms benefits from the merit of each feature (e.g., [2], [3], [27]–[39]). And we presented the features and how these features are used by the previous algorithms in the Table I.

Each feature has its own disadvantages. For example, the gray features are sensitive to illumination change while

TABLE I
THE FEATURES AND THE FUSION OPERATOR ARE USED IN MULTI-FEATURES BASED ALGORITHMS

Background models	Features	How multi-features are used in these background models ?
Zaharescu et al. [28]	Intensity, Color and Texture	Extended the components of codeword by multiple features.
Noh et al. [40]	Color and Texture	Extended the components of codeword and verified foreground by sub-image region.
Klare et al. [41]	Intensity, Color and Texture	Used multiple features alternatively in MoG.
Chen et al. [29]	Intensity, Color	Filtered the result of MoG which based on intensity by color features.
Zhang et al. [35]	Color and Texture	Fused the measures based on multi-features by Fuzzy Integral.
Baf et al. [42]	Color and Texture	Fused the measures based on multi-features by Choquet Integral.
Chiranjeevi et al. [27]	Intensity and Texture	Fused the measures based on multi-features by Sugeno and Choquet Integral.
Yao et al. [39]	Color and Texture	Controlled the contribution of multiple features by global weights.
Chen et al. [44]	Color and Texture	Fused multiple features by shared model which is represented by samples of pixel.
Xue et al. [30]	Intensity, Color	Integrated multiple features and modeling with MoG.
Han et al. [3]	Intensity, Color and Haar-like	Combined multiple features into a vector which as the input of SVM.
Li et al. [38]	Intensity, Texture	Adaptively Weighted the measures based on multiple features
Ji et al. [31]	Intensity, Color and Texture	Adaptively Weighted the measures based on multiple features with different contribution.
SoAF	Intensity, Color and Texture	Adaptively weighted the features according to its stability.

texture features cannot handle the objects' edge well [34]. Zaharescu *et al.* [28] extended the components of codeword by multi-features to cope with local illumination change. Furthermore, Noh and Jeon [40] also analyzed the statistics of color and texture features by codebook scheme. And the foreground blob is verified by the Hausdorff distance between the shape of blob boundary and the edge-map of the corresponding sub-image region. In contrast, Klare and Sarkar [41] proposed a meta-learning algorithm that incorporates multiple instantiations of the MoG, and used different features alternatively to against the varying illuminations. Similarly, Chen *et al.* [29] utilized color features as a filter to capture the candidate from the MoG result based gray features. And a statistical frequency counting is used to distinguish true flames from those candidates.

Unlike the extension or combination of models based on single feature, Zhang and Xu [35] utilized fuzzy integral to fuse the texture and color features for background subtraction. And Baf *et al.* [42] proposed a fuzzy approach using the Choquet integral to avoid the uncertainty in the classification. Hence, Chiranjeevi and Sengupta [27] proposed a model level fuzzy aggregation measure to ensure the robustness of algorithms. In particular, the Sugeno and Choquet integrals are incorporated to compute fuzzy similarities from the ordered similarity function values for each model.

Besides the integration of similarity measures for different features, Yao and Odobezi [39] used a global constant-valued weight to indicate the contribution of integrating advantages of multi-features. And Xue *et al.* [30] proposed a double judge method to remove the moving objects' shadow. The shadow is determined by the angle of color and the brightness between the shadow and the background. Moreover, Han and Davis [3] combined the intensity, color and Harr-like features into a features vector, and the SVM is utilized to label if a pixel is foreground or background. In addition, Chen *et al.* [43] fused multiple features by shared model which is represented by a sequence of samples based on sample consensus. And the shared models are updated by random selecting a pixel matched the model with an adaptive update rate.

Multiple features can compensate the defect of single one which is shown in [38] work Li and Leung [38] weighted result of two difference measures, and then presented an integration by minimizing an energy function with additional constraint

of smoothness. Motivated by the idea of Li and Leung's work [38], Ji and Wang [31] focused on using different features to compensate with each other for alleviating their respective defects and setting the confidence of model and weights of model. Also, Ji and Wang's work [31] is closely related to our proposed approach. The main difference between our work and it is that Ji and Wang [31] designed specific similarity models for each feature while the SoAF uses the uniform similarity model which is actually a framework whose feature collection can consist of any kind of feature rather than the grayscale, color and texture. Similarly, there is a same difference which also exists between Chiranjeevi and Sengupta's work [27] and SoAF.

In addition, in Ji and Wang's work [31], because of way the multi-features are constructed, every feature contributes to the background and foreground classification, meaning that unimportant features affect the outcome. Hence, it may be not suitable for foreground detection for a particular pixel. In contrast, in SoAF, the components of the background model are independent, so the situation does not exist that a feature with high stability is not suitable for foreground detection.

Self-adaptability is an important property of multi-feature models, since there are large numbers of parameters that need to be tuned. Fortunately, most of the multi-feature models have such properties and adaptively learn these parameters (e.g., [2], [32], [33], [37], [44]). Similar to these works, SoAF also uses the self-adaptability which makes it more practical.

III. FEATURE EXTRACTION

In this section, the introduction of the feature extraction process is presented. It should be noted that the feature vector of SoAF can consist of any kind of feature. Here, we choose gray, color and texture as the features, since they are three of the most commonly adopted features in background modeling.

The gray feature is actually the gray-scale of pixel. In order to improve the robustness of SoAF, we extract the normalized RGB as the color feature. However, both gray and color features are sensitive to the shadows of moving objects. To tackle this issue, we enhance the SILTP texture descriptor and incorporate it into our feature vector. The SILTP texture descriptor [21] is an extension of Local Binary Pattern (LBP) [45] which enjoys more robustness to illumination change. However, SILTP does not inherit

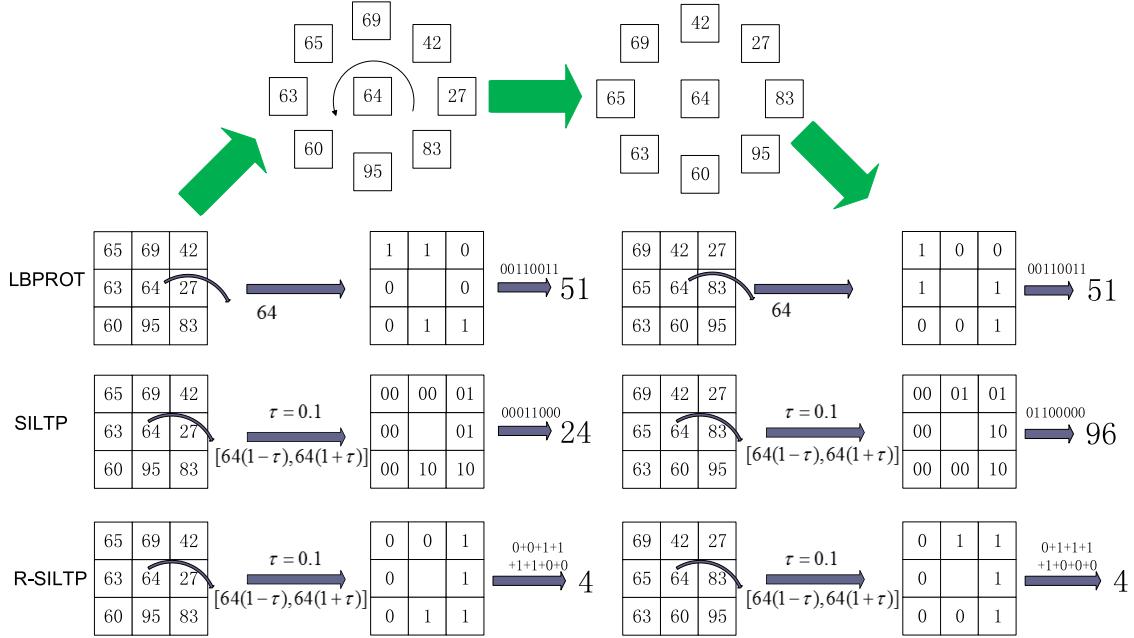


Fig. 2. The encoding processes of LBPROT, SILTP and R-SILTP in a pixel block. The left part is the encoding process of the original image block and the right part is the one of its rotational versions.

the rotational invariance of LBP. Considering the trade off between the rotational invariance of textures and the computational complexity of the feature extraction process, we propose an improved SILTP texture descriptor named Rotated Scale Invariant Local Ternary Pattern (R-SILTP). The feature extraction process of R-SILTP is shown as follows:

$$f_{texture} = R\text{-SILTP}_{N,R,K}^{\tau}(x_c, y_c) = K \times \sum_{k=0}^{N-1} s_{\tau}(g_c, g_k),$$

where g_c corresponds to the gray-scale of the center pixel (x_c, y_c) of the local neighborhood and g_k are that of its N neighboring pixels equally spaced on a circle of radius R . τ is a scale factor indicating the comparison range. K is used to expand the value so we can see the results of R-SILTP by the naked eye, s_{τ} is a piecewise function defined as:

$$s_{\tau}(g_c, g_k) = \begin{cases} 1, & \text{if } g_k > (1 + \tau)g_c \\ 1, & \text{if } g_k < (1 - \tau)g_c \\ 0, & \text{otherwise.} \end{cases}$$

We enhance SILTP from three aspects: First, R-SILTP requires less memory for storage since R-SILTP only uses one bit to encode the differences between the center pixel and its neighborhood in a binary way. For example, in an 8 bit memory space, the SILTP can only encode 4 pixels while R-SILTP encodes 8 pixels. Second, the computational complexity of R-SILTP is lower, since we can just simply count the number of 1 instead of binary encoding. Third, the rotational invariance of LBP is partially regained in R-SILTP.

We show the encoding processes of SILTP, R-SILTP and the LBPROT [46] in Fig. 2. In the first row of Fig. 2, since LBPROT is rotational invariant, it captures the same result for one texture block and the rotational version of this block.

In contrast, the encoding result of SILTP does not have such a property as shown in the second tow of Fig. 2. However, in the bottom row of Fig. 2, R-SILTP has the same behaviour as LBPROT. Clearly, R-SILTP inherits the rotational invariance of LBPROT. Moreover, R-SILTP also inherits the robustness of SILTP [21], since it is an improved version of SILTP.

Moreover, we also follow the experimental way of the justification experiments in [47] for comprehensively comparing R-SILTP and SILTP. Both the R-SILTP and SILTP are considered as the input of the MoG, which is a popular background subtraction method, with the same parameter. Fig. 3 shows the Recall(Re), Precision(Pr) and F-Measure(Fm) plots on the highway, pedestrians and PETS2006 video sequences respectively. Compared with SILTP, R-SILTP is robust to the rotation of pixel block but loses direction information. However, the robustness improvement of R-SILTP over SILTP is validated by the observations from the experiments that the R-SILTP is higher than SILTP in the Pr plot. And the Fm plot also shows the superiority of R-SILTP over SILTP.

IV. MODELING THE BACKGROUND WITH MULTI-FEATURE

In this section, we introduce the methodology of SoAF and its flow chart is shown in Fig. 4. We present the feature capturing and the background modeling processing in Section IV-A and Section IV-B respectively. Then, we introduce the foreground segmentation in Section IV-C. Finally, the “double threshold” process for foreground segmentation is discussed in Section IV-C.

A. Feature Vector

As mentioned above, any kind of feature can be applied using the SoAF framework, since its feature vector is the combination of them, as shown in part 1 of Fig. 4. In this

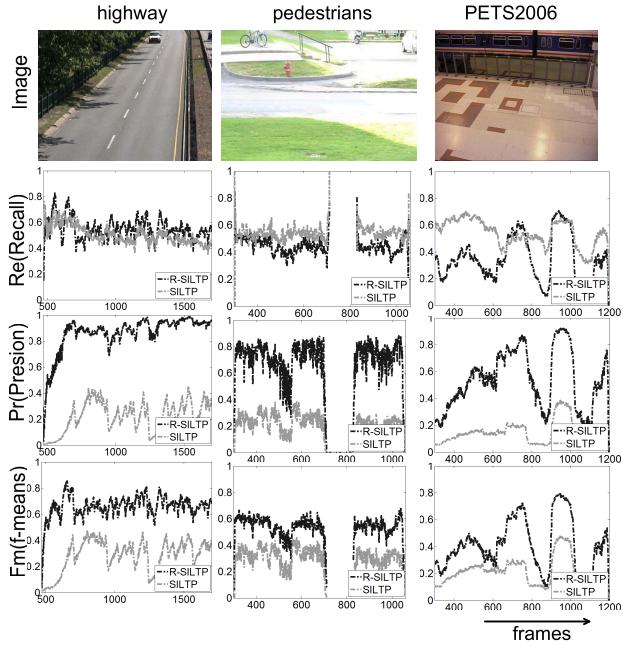


Fig. 3. An experimental example to compare the encodings of SILTP and R-SILTP in highway, pedestrians and PETS2006 video sequences. Both SILTP and R-SILTP are input into MoG and the second to fifth rows show the Re, Pr and Fm score respectively.

work, we choose gray, color and texture, which are the three most commonly used features in change detection, to construct the feature vector. It is defined as follows:

$$F = \{f_{gray}, f_{red}, f_{green}, f_{blue}, f_{texture}\}. \quad (1)$$

In the feature vector, its first, second, third, fourth and fifth entries are the gray-scale, red, green, blue and texture features respectively.

We define the dominant feature as the entry of the feature vector with the highest stability. We measure the degree of stability via the intensity of the peak of the distribution of the corresponding feature. Fig. 5 demonstrates the variation and the concentration over 1000 frames on two pixels P_1 with location (50,120) and P_2 with location (189, 157) in outdoor scenes. Feature variation and concentration from two pixels are shown in Fig. 5(a) and Fig. 5(b), the intensity-time plot demonstrates the variation of the feature intensities, while the histogram of these features in time sequences shows the stability of each feature. In particular, to simplify the problem, we only use one unimodal model to analyze the stability of each feature. In this condition, the dominant features are decided by the highest peak of features' histogram.

From the intensity-time plot shown in Fig. 5(a), all plots have strong variances during the whole video sequence. However, benefiting from the clear texture and the bright green color of the fence, the stability of the texture and the green color are still good in this pixel and yield good results for SoAF. In this condition, we see that SoAF often adaptively finds the most stable entry in some extreme scenes like dark regions.

According to the observations of Fig. 5(b), the distribution of features has some strong variations around frame 250 and

frame 800 in each subfigure, which reveals strong cues of a moving object. However, in the frames without moving objects, there still exists tiny variations in the distribution of gray features. In contrast, the distribution of textures remains almost invariant. In this case, we find the stability of the texture feature is higher than other features and better for foreground detection. Actually, the stability of textures is the highest one which is selected as the dominant feature of this pixel by SoAF. Moreover, Fig. 5(b) demonstrates the complementarity contributions of secondary features in SoAF. As we see, the stability of the color red is also good and suitable for change detection, which is considered a good complement to the dominant features.

B. Background Modeling

The motivation of SoAF is letting the feature with higher stability contribute more for foreground detection. Each background of SoAF consists of five components which are described by the same statistical single-peak distribution with different feature inputs. Also the peak of the histogram of each feature over the time sequence is used as the weight of its corresponding component. In this way, the component of the background described by the dominant feature have the highest weight to classify a pixel as foreground or background, and others with lower weights are also complemented to enhance performance. In addition, inspired by a popular technique [6] for dealing with dynamic backgrounds, there are more than one background model in SoAF for handling the case that the histogram in the time sequence of a particular feature is multimodal like the one of the texture shown in Fig. 5(a).

We explain the background model procedure for one pixel, but the procedure is identical for each pixel. The SoAF background model for a particular pixel is represented by a group of background model $\{c_1, c_2, \dots, c_L\}$ which actually is an adaptive multimodal distribution, where L is selected by the user (usually its value is 3-5). Each of the adaptive multimodal distributions include five components which are the same single-peak probability distribution with different input features. Each distribution is described by a mean of feature values f , a bin width s and a weight h where they are respectively defined as the center of single-peak distribution, the range of the distribution, and the peak value of the distribution. In other words, the background model in SoAF is represented by the feature vector F (defined in Eq. 1), a bin width vector S , and a weight vector H . S and H can be mathematically denoted as follows,

$$S = \{s_{gray}, s_{red}, s_{green}, s_{blue}, s_{texture}\}, \quad (2)$$

$$H = \{h_{gray}, h_{red}, h_{green}, h_{blue}, h_{texture}\}, \quad (3)$$

where s_{gray} is the bin width of the gray features and is used to judge if the current pixel is matched with the gray component of the corresponding background model. Similarly, the s_{red} , s_{green} , s_{blue} and $s_{texture}$ are matched in the same way. The h_{gray} , h_{red} , h_{green} , h_{blue} and $h_{texture}$ are peaks of the distribution which are used as weights for evaluating the contribution of each component in foreground detection.

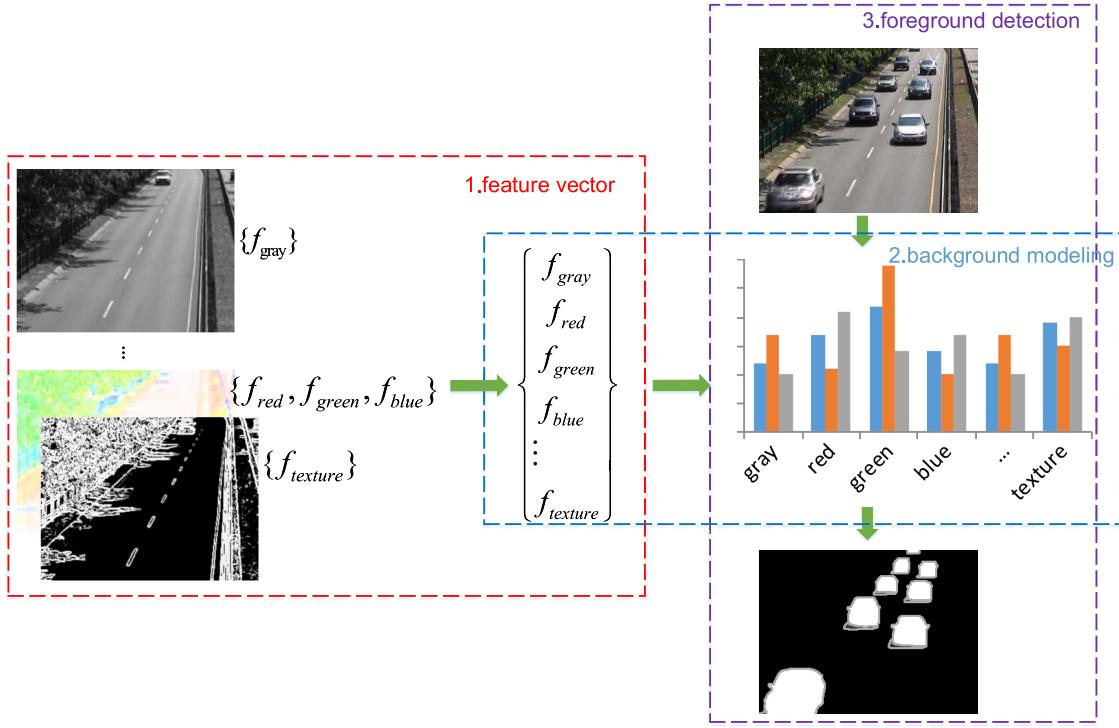


Fig. 4. The flow chart of SoAF, there are three steps of SoAF. 1. Features vector, we extract several features and fuse them into a feature vector. 2. Background modeling, the feature vector is analyzed by a histogram and the peaks of the histogram are used as the weight of each entry in the feature vector. 3. Foreground detection, the contribution of different features are different.

In this way, our background model is represented by:

$$c_i = \{F_i, S_i, H_i\} = \{f_{i,j}, s_{i,j}, h_{i,j}\}$$

$$i \in \{1, \dots, L\}, \quad j \in \{\text{gray, red, green, blue, texture}\},$$

where c_i represents the i -th background, L is the number of the background model.

Let us denote a particular pixel in the new video frame, which is described by $P = \{p_{gray}, p_{red}, p_{green}, p_{blue}, p_{texture}\}$. The P will match with the current L background models for labeling the pixel. The process of the match between each component of the background model and each entry of the vector P is completely independent, which means the gray entry of current pixel p_{gray} is only matched with the gray component of the background model. We only interpret the match process of the gray component, since the match processes of other components are done the same way. If the absolute value of the distance between p_{gray} and $f_{i,gray}$ is below the bin width of the gray component $s_{i,gray}$, we consider this pixel matched with the gray component of the i -th background model. Also, $f_j(i)$ is used to judge if the current pixel P is matched with j -th component of the i -th background model. This process is defined as follows:

$$f_j(i) = \begin{cases} 1 & \text{if } |f_{i,j} - p_j| < s_{i,j} \\ 0 & \text{if } |f_{i,j} - p_j| \geq s_{i,j} \end{cases} \quad (4)$$

, for example, $f_{gray}(2) = 1$ means that the p_{gray} is matched with the gray component of the second background model, then the component of the second background model is

updated as follows:

$$f_{i,j} = a_b \times f_{i,j} + (1 - a_b) \times p_j,$$

$$s_{i,j} = a_m \times s_{i,j} + (1 - a_m) \times (|f_{i,j} - h_j| + K_j),$$

$$w_{i,j} = w_{i,j} + 1, \quad i = 2, \quad j = \text{gray},$$

where a_b and a_m are user-settable learning rates for updating $f_{i,j}$ and $s_{i,j}$. K_j is utilized to make sure that the bin width is nonzero. Furthermore, if there are no background matches with h_{gray} in the gray component then the gray component of the background model with the lowest weight is replaced by p_{gray} , and the width and weight of this component is reset.

The multimodality of the texture feature shown in Fig. 5(b) is represented by the texture components of several background models in SoAF. In addition, each peak of the modality in the texture histogram is used as the weight of the texture component for different models. Moreover, the bin of the histogram is represented by the bin width which is used to judge if the current pixel matches the background. Hence, the SoAF background model is deemed as a model which is described by the histograms of multiple features shown in Fig. 6.

C. Foreground Detection

In the early work [31], it was investigated that a pixel can be described by several features including gray, color, and texture. If more information is effectively utilized, foreground objects are better detected. However, not all features are suitable for foreground detection with respect to a particular pixel, there are always some features more suitable for foreground

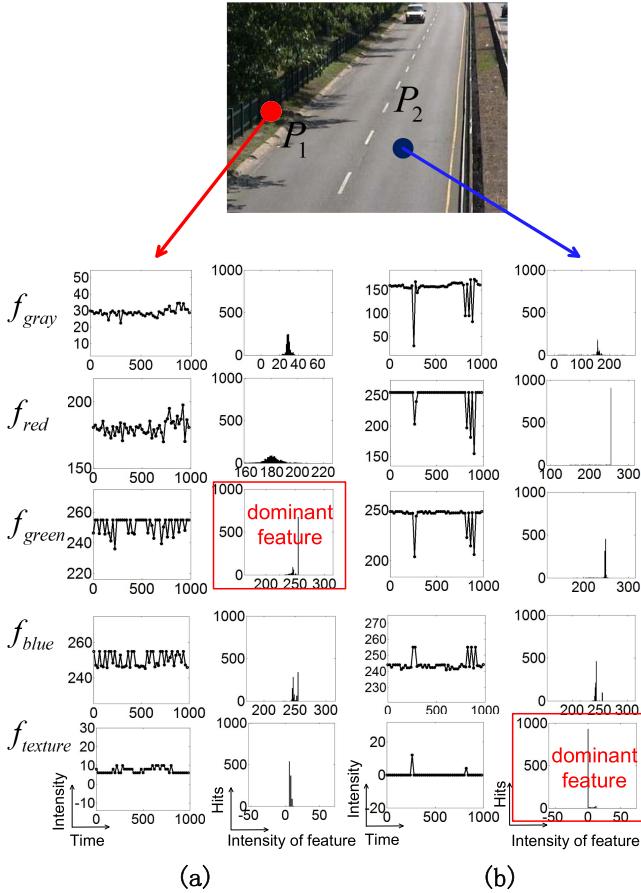


Fig. 5. The illustration shows the comparison of variation and concentration between dominant features and secondary features. We sample the feature vector in pixel \$P_1\$ and \$P_2\$ over 1000 frames on a highway video sequence. The intensity variation and the histogram in the time sequence of each entry in the feature vector on two pixels are shown in (a) and (b), respectively. The color red is the dominant feature (shown in (a) on pixel \$P_1\$ (red point) which is a pixel on a dark fence, and the texture features are the dominant feature (shown in (b)) on pixel \$P_2\$ (blue point) which is a pixel on the highway.

detection while others are not. For example, texture features are often more suitable in scenes under variations of illumination in comparison with gray-scales. In SoAF, the suitable features will contribute more for pixel classification.

The proximity between the current pixel and the background is given by the ratio of \$\theta_m\$ to \$\theta_s\$. \$\theta_m\$ is the sum of weights of the components in all models which are matched with the current pixel,

$$\theta_m = \sum_{i=1}^{i=L} \sum_{j \in G} f_j(i) w_{i,j}. \quad (5)$$

Similarly, \$\theta_s\$ is the sum of weights of components in all models and defined as follows:

$$\theta_s = \sum_{i=1}^{i=L} \sum_{j \in G} w_{i,j}, \quad (6)$$

where \$G = \{gray, red, green, blue, texture\}\$, is the feature collection composed by the gray, red, green, blue, and texture features for representing the different components of background model, and \$f_j(i)\$ is the match function which was already shown in Eq. 4. Finally, the proximity is calculated as

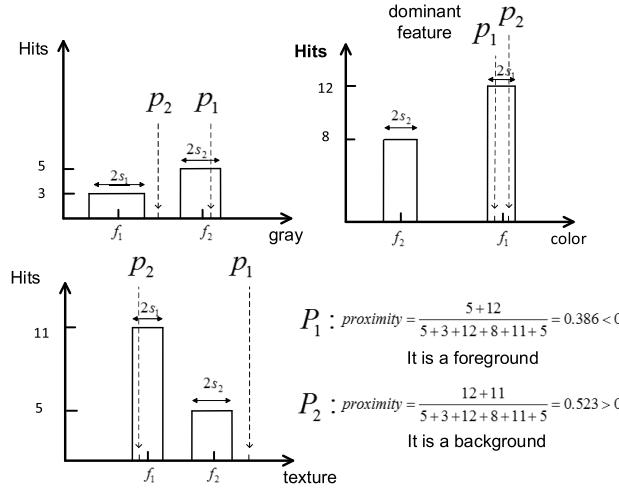


Fig. 6. The process of SoAF for foreground detection. These three histograms represented the gray, color, and texture components of the background. The two boxes of each histogram represent two different background models. \$P_1\$ and \$P_2\$ are two pixels waiting for classification as foreground and background. Effected by secondary features, although both \$P_1\$ and \$P_2\$ are matched with a background model using color features, which are the dominant features in this pixel position, the classified result of the two pixels are completely different which shows the necessity of using secondary features.

follows:

$$\text{proximity} = \frac{\theta_m}{\theta_s}.$$

Then the computed proximity can be compared with the threshold value to classify this pixel as background or foreground.

In SoAF, besides the dominant features utilized for foreground detection, the secondary features are also complemented to enhance the performance. Fig. 6 shows a typical example for demonstrating the contributions of the secondary features. In this example, we assume the number of background models \$L = 2\$. In addition, there are three components, namely gray, color, and texture, for each background model. In this way, SoAF is represented by three statistic histograms shown in Fig. 6. There are two boxes in each histogram figure which represents components of two models \$c_1\$ and \$c_2\$. From Fig. 6, we see the peak of the histogram of the color feature is the highest. Therefore the color feature is considered as the dominant features with respect to this pixel. The pixels \$P_1\$ and \$P_2\$ belong to the foreground and background respectively. In the color histogram, both \$P_1\$ and \$P_2\$ are inside the single-peak distribution of \$f_1\$, which means both \$P_1\$ and \$P_2\$ are matched with the background \$c_1\$ in the color component. However, in the gray histogram, only \$P_1\$ matches with background \$c_2\$ while in the texture histogram, only \$P_2\$ matches with background \$c_1\$. Considering the effects of the secondary features, we find that the proximities of \$P_1\$ and \$P_2\$ are completely different and these two pixels are correctly labelled using SoAF (see the last subfigure in Fig. 6). In conclusion, from the example shown in Fig. 6, we know that the secondary features are still necessary in SoAF.

D. Double Threshold to Segment Foreground and Background

In this section, a “double threshold” method for segmenting foreground and background is introduced. The double

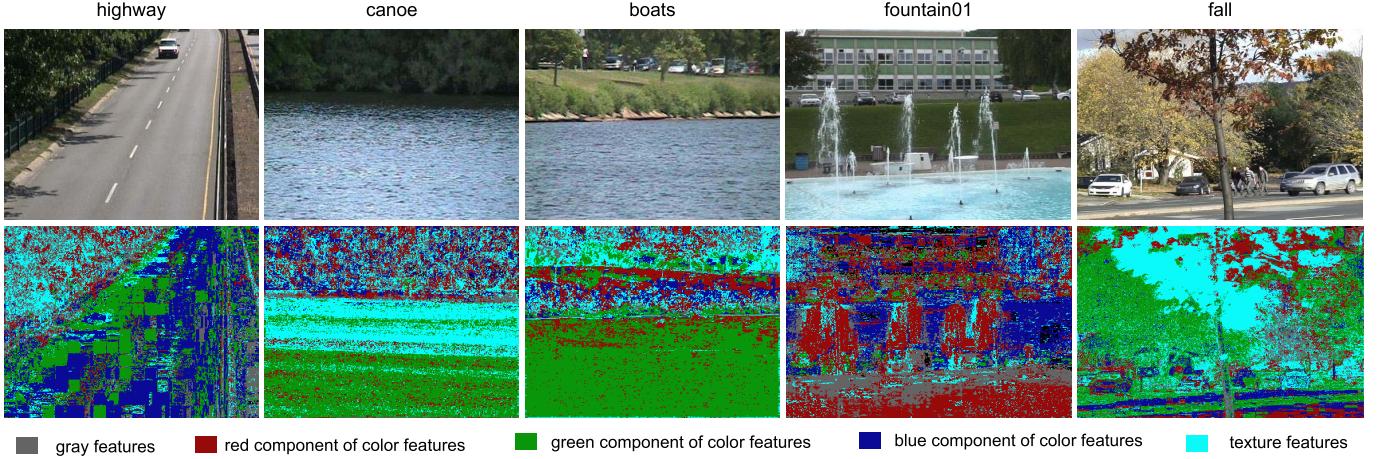


Fig. 7. The demonstration of different dominant features on each pixel in five video sequences highway, canoe, boats, fountain01, and fall.

threshold is often used in edge detection to solve the breakpoint problem [48]. There are two threshold values in “double threshold”, a high threshold value T_h and a low threshold value T_l . The pixel is segmented as background if the proximity of the pixel is greater than T_h . When the proximity of a pixel is higher than T_l but lower than T_h , we find if there are background pixels around this pixel within the range of radius R_T . The pixel is classified as background if there exists background pixels around it, otherwise the pixel is classified as foreground. The “double threshold” method has a good ability for dealing with break points and shadows of moving objects. This process improves the robustness of our model. Experiments in Section V-D validate the superiority of the double threshold over the single threshold.

V. EXPERIMENTS

In this section, we have conducted several experiments to analyze SoAF in details.

All the video sequences which contain 11 categories in changedetection.net (CDN) benchmark [49] are used for validation. We choose the CDN benchmark due to the following reasons: it represents a realistic, large-scale video data consisting of 53 video sequences which is claimed to be the largest dataset for change detection; this dataset provides the ground truth for each frame in video sequences; this dataset also provides the detection results of many representative background modeling algorithms which help us to compare SoAF directly with other algorithms. And all the metrics for evaluation is following in CDN benchmark [49].

A. The Self-Adaptability of SoAF

In this section, the self-adaptability of SoAF is demonstrated. The self-adaptability of SoAF means that dominant features over pixel and time sequence domains are adaptively selected by SoAF. To show this phenomenon, we mark different pixels with different dominant features using five colors in four video sequences (highway, canoe, fountain01, and fall) [49]. And the parameters of SoAF are shown in the first row of Table IV.

Fig. 7 shows all experimental results. In the highway video sequence, the color features are selected as dominant features in most part of highway road. Moreover, since we utilize several unimodal model to analyze the features’ stability, the dynamic texture is chosen as dominant features in rippling water and waving tree which is shown in canoe and fall video sequences respectively. In contrast, in the fountain01 video sequence, the gray features are considered as dominant features in the shiny water surface. From the observations of the experiments, the dominant features of different pixels are different which reflects the self-adaptability of SoAF.

B. The Complementarity of SoAF

In this section, we experimentally analyse the complementarity of SoAF. In the SoAF background model, the secondary features are utilized to enhance the foreground model and overcome the defects lead by dominant features. In order to show the importance of secondary features, we implement SoAF without using the secondary features, then compare SoAF and SoAF without secondary features in the PETS2006 and canoe video sequences. The comparison results are shown in Fig. 8 and Fig. 9. We plot the Re and Sp of each frame in the PETS2006 and canoe video sequences in Fig. 8 and Fig. 9. The reason we choose these two metrics is because Re is considered as a measure for completeness for the foreground and Sp is considered as a measure for completeness for background, which are the most intuitive metrics for quantitative evaluation.

From the Re and Sp plots shown in Fig. 8, the Re of SoAF is higher than SoAF without secondary features in both PETS2006 and canoe video sequences, which means the foreground segmented by SoAF is more complete than SoAF without secondary features. Meanwhile, both the Sp of SoAF and SoAF without secondary features are close to 1, which means the background segmented by SoAF and SoAF without secondary features has roughly equal performance. This experiment demonstrates the impact of secondary features to the performance of SoAF. Clearly, the secondary features enhance the foreground segmented by dominant features.

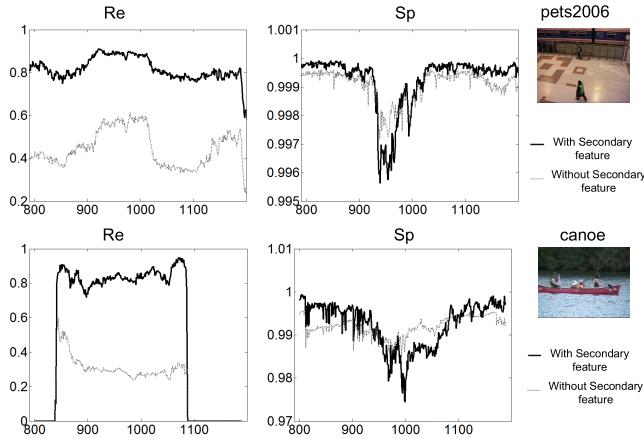


Fig. 8. The quantitative comparison between SoAF and SoAF without secondary features in the PETS2006 and canoe video sequences based on Re and Sp.

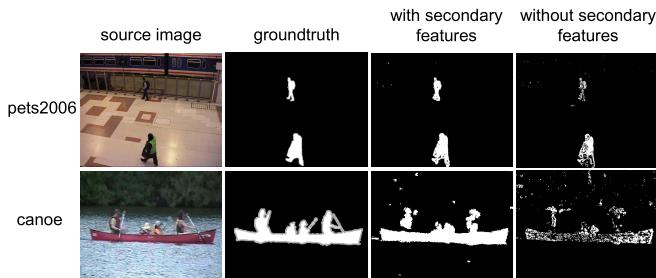


Fig. 9. The qualitative comparison of SoAF and SoAF without secondary features in the 700-th frame of the PETS2006 sequence and 946-th frame of the canoe video sequence respectively.

Fig. 9 show the foregrounds segmented by SoAF and SoAF without secondary features in the same frames. In the PETS2006 video sequence, there is a pedestrian with a soft shadow. Due to the stability of the dominant features, both SoAF and SoAF without secondary remove the shadow of moving objects in the foreground. However, it is obvious that the foreground of SoAF is better than SoAF without secondary features, since there are cavities in the foreground segmented by SoAF without secondary features. In addition, this situation is also shown in the canoe video sequence. From this observation, we conclude that the secondary features are an indispensable part of SoAF, since it enhances the foreground and overcomes defects lead by dominant features.

C. The Comprehensive Evaluation of SoAF

In this section, the comprehensive evaluation of SoAF is shown. We compare SoAF with nine state-of-the-art algorithms based on single feature and multi-features. The algorithms based on single feature includes MST [50], RMoG [51], ViBe+ [52], ViBe [12], MoG [8], improved MoG (IMoG) [53] and RECTGAUSS [23]. And the multi-features based algorithms includes MultiLayer in [39] and MultiCue in [40]. The results of algorithms based single features are directly referring from the CDN benchmark [49]. Moreover, since the lack of results of multi-features based algorithms in

CDN benchmark [49], the implementation of MultiCue and MultiLayer are captured from BSGLibrary [54] with defaults parameters. In SoAF, we adopt two groups of parameters in all experiments which are recorded in Table IV. The group of parameters in the first row of Table IV is used for the result of SoAF₁ while the one in the second row is used for SoAF₂.

Table II reports the performances of SoAF and other compared algorithms on the video sequences belonging to Baseline, Dynamic Background, Shadow and Thermal scenes under the metrics of Re, Pr and Fm. And the evaluation of SoAF on other categories video sequences are shown in Table III. Due to the length of paper, Fig. 10 only shows the foreground segmented by SoAF₂ and other algorithms in a specific frame of video sequences belonging to Baseline, Dynamic Background and Shadow scenes. In particular, since the IMoG and ViBe+ are the enhanced versions of MoG and ViBe respectively, we only present the foregrounds learned by IMoG and ViBe+.

In the Baseline scenes, there are four video sequences and SoAF achieves significant improvement about Fm score in PETS2006 and office video sequences. The PETS2006 is a video of a subway station. In this video, the shadow of the pedestrians and reflected light on floor are the challenges. The Re score of the proposed approach is better than the one of the others in this video sequence, which result from the dominant features chosen by SoAF. Benefited from the secondary features, SoAF also achieves a good Pr score but slightly worse than MultiLayer [39] which is also based on multi-features. Moreover, since proposed approach adaptively weighting the features according their stability, SoAF achieved the highest Fm score in PETS2006 video sequence. And the same situation also shows in office video sequence. However, SoAF does not get good Fm scores in highway and pedestrians video sequences, since the moving objects jamming the stability of features.

In the Dynamic Background scenes, SoAF works well in the boats and fountain02 video sequences. Compared with other multi-features based algorithms, SoAF utilizes several unimodal model to handle the dynamic features and works much better in the Dynamic Background scenes. However, since only R-SILTP considers the neighbour information in our feature vector, SoAF cannot work very well in the fall video sequence compared with algorithms which mainly focus on the exploitation of the neighbour information of pixels like RMoG [51] and ViBe+ [52]. In addition, the SoAF also achieves the state-of-the-art in bungalows, busStation, cubicle and peopleInShade video sequences. The dominant features with the highest stability are suitable for handling the moving objects' shadow. And the other multi-features based algorithms also work well but the proposed approach still shows its advantage in performance.

In the Thermal and Camera Jitter scenes, SoAF works well in the diningRoom and lakeSide video sequences and achieves the highest Fm score in Camera Jitters category. As we mentioned before, SoAF can capture stable features in extreme scenes which included the thermal image. Although there is no color features in the thermal image, the gray features and texture features still effect in this kind of scene. SoAF adaptively

TABLE II

THE PERFORMANCE COMPARISON OF SOAF AND NINE STATE-OF-THE-ART ALGORITHMS ON THE VIDEO SEQUENCES BELONG TO BASELINE, DYNAMIC BACKGROUND, SHADOWS AND THERMAL SCENES UNDER THE METRICS OF RE, PR AND FM (FROM LEFT TO RIGHT IN EACH CELL)

Videos	Baseline				Dynamic Background		
	highway	office	pedestrians	PETS2006	boats	canoe	fall
MST	0.83 0.92 0.87	0.70 0.94 0.80	0.95 0.96 0.95	0.78 0.73 0.75	0.51 0.45 0.48	0.91 0.86 0.89	0.85 0.27 0.41
RMoG	0.79 0.95 0.87	0.43 0.92 0.59	0.91 0.96 0.94	0.70 0.81 0.75	0.82 0.85 0.83	0.90 0.97 0.94	0.72 0.64 0.67
MoG	0.92 0.93 0.92	0.49 0.75 0.59	0.99 0.92 0.95	0.88 0.79 0.83	0.76 0.70 0.73	0.87 0.90 0.88	0.88 0.29 0.44
IMoG	0.89 0.92 0.90	0.51 0.93 0.66	0.98 0.94 0.96	0.85 0.81 0.83	0.70 0.80 0.75	0.85 0.92 0.89	0.86 0.28 0.42
ViBe	0.85 0.92 0.88	0.79 0.97 0.87	0.94 0.96 0.95	0.71 0.86 0.77	0.54 0.32 0.40	0.86 0.85 0.86	0.79 0.27 0.40
ViBe+	0.93 0.93 0.93	0.70 0.92 0.80	0.95 0.96 0.96	0.73 0.89 0.80	0.43 0.69 0.53	0.92 0.97 0.94	0.89 0.71 0.79
RECTGAUSS	0.84 0.92 0.88	0.48 0.94 0.64	0.94 0.91 0.92	0.41 0.90 0.56	0.49 0.93 0.64	0.65 0.89 0.75	0.83 0.14 0.24
MultiLayer	0.85 0.99 0.91	0.20 0.91 0.33	0.95 0.96 0.95	0.73 0.95 0.82	0.24 0.99 0.39	0.42 0.96 0.58	0.69 0.54 0.61
MultiCues	0.90 0.81 0.85	0.73 0.78 0.76	0.94 0.63 0.76	0.85 0.60 0.70	0.38 0.79 0.52	0.08 0.56 0.14	0.91 0.07 0.14
SoAF ₁	0.93 0.80 0.86	0.99 0.54 0.70	0.97 0.73 0.84	0.99 0.56 0.72	0.98 0.14 0.24	0.97 0.41 0.58	0.96 0.14 0.24
SoAF ₂	0.69 0.98 0.81	0.92 0.89 0.90	0.86 0.93 0.89	0.94 0.90 0.92	0.83 0.83 0.83	0.94 0.86 0.90	0.76 0.33 0.46
Videos	Dynamic Background			Shadow			
	fountain01	fountain02	overpass	backdoor	bungalows	busStation	CopyMachine
MST	0.49 0.08 0.14	0.85 0.79 0.82	0.82 0.85 0.84	0.82 0.90 0.85	0.85 0.72 0.78	0.60 0.86 0.70	0.82 0.87 0.85
RMoG	0.55 0.12 0.20	0.91 0.83 0.87	0.84 0.97 0.90	0.85 0.72 0.78	0.73 0.89 0.81	0.67 0.93 0.78	0.42 0.79 0.55
MoG	0.80 0.04 0.08	0.87 0.75 0.80	0.83 0.92 0.87	0.85 0.51 0.64	0.89 0.72 0.80	0.73 0.88 0.80	0.54 0.79 0.64
IMoG	0.75 0.04 0.08	0.84 0.75 0.79	0.81 0.94 0.87	0.82 0.51 0.63	0.87 0.72 0.79	0.71 0.88 0.79	0.54 0.84 0.66
ViBe	0.58 0.05 0.10	0.80 0.86 0.83	0.76 0.85 0.81	0.80 0.90 0.85	0.81 0.70 0.75	0.68 0.91 0.78	0.75 0.89 0.81
ViBe+	0.62 0.19 0.30	0.87 0.88 0.87	0.84 0.93 0.88	0.84 0.87 0.86	0.88 0.72 0.79	0.72 0.88 0.79	0.68 0.81 0.74
RECTGAUSS	0.00 0.01 0.00	0.03 0.99 0.05	0.87 0.92 0.89	0.88 0.98 0.93	0.84 0.72 0.78	0.69 0.90 0.78	0.44 0.89 0.59
MultiLayer	0.87 0.21 0.34	0.87 0.89 0.88	0.46 0.86 0.60	0.91 0.96 0.94	0.90 0.98 0.94	0.48 0.94 0.63	0.38 0.93 0.54
MultiCues	0.74 0.02 0.04	0.92 0.72 0.81	0.70 0.77 0.73	0.95 0.83 0.89	0.45 0.79 0.57	0.90 0.76 0.82	0.63 0.69 0.66
SoAF ₁	0.92 0.04 0.07	0.98 0.23 0.37	0.99 0.28 0.43	0.97 0.74 0.84	0.99 0.91 0.95	0.97 0.70 0.81	0.88 0.60 0.71
SoAF ₂	0.64 0.11 0.19	0.88 0.85 0.87	0.85 0.78 0.81	0.79 0.95 0.87	0.89 0.97 0.93	0.77 0.93 0.84	0.64 0.84 0.73
Videos	Shadow		Thermal				
	cubicle	peopleInShade	corridor	diningRoom	lakeSide	library	park
MST	0.69 0.72 0.70	0.92 0.82 0.86	0.56 0.64 0.60	0.35 0.74 0.48	0.07 0.93 0.13	0.66 0.93 0.77	0.41 0.97 0.58
RMoG	0.58 0.71 0.64	0.75 0.80 0.77	0.58 0.92 0.71	0.36 0.99 0.53	0.15 0.96 0.26	0.20 0.99 0.33	0.43 0.83 0.57
MoG	0.81 0.55 0.65	0.94 0.84 0.89	0.83 0.81 0.82	0.70 0.93 0.80	0.40 0.93 0.56	0.28 0.85 0.42	0.64 0.81 0.71
IMoG	0.79 0.55 0.65	0.93 0.84 0.88	0.83 0.84 0.84	0.69 0.92 0.79	0.36 0.92 0.52	0.29 0.82 0.42	0.59 0.85 0.70
ViBe	0.73 0.76 0.74	0.94 0.84 0.89	0.79 0.92 0.85	0.59 0.96 0.73	0.22 0.97 0.36	0.62 0.92 0.74	0.50 0.92 0.65
ViBe+	0.76 0.86 0.80	0.98 0.84 0.91	0.77 0.94 0.85	0.58 0.97 0.73	0.21 0.97 0.34	0.69 0.95 0.80	0.45 0.91 0.61
RECTGAUSS	0.66 0.62 0.64	0.80 0.60 0.69	0.36 1.00 0.53	0.45 0.98 0.62	0.02 1.00 0.04	0.18 0.97 0.31	0.21 0.86 0.34
MultiLayer	0.61 0.95 0.74	0.75 0.94 0.84	0.19 0.89 0.31	0.22 0.97 0.36	0.05 0.90 0.10	0.05 0.90 0.09	0.56 0.96 0.71
MultiCues	0.69 0.76 0.72	0.94 0.88 0.91	0.61 0.87 0.71	0.41 0.91 0.56	0.17 0.76 0.28	0.25 0.91 0.39	0.75 0.78 0.76
SoAF ₁	0.99 0.28 0.43	0.99 0.68 0.81	0.90 0.63 0.74	0.90 0.73 0.81	0.67 0.48 0.56	0.24 0.79 0.37	0.91 0.61 0.73
SoAF ₂	0.88 0.73 0.80	0.92 0.94 0.93	0.00 0.00 0.00	0.06 0.98 0.11	0.00 0.00 0.00	0.00 0.00 0.00	0.00 0.00 0.00

TABLE III

THE PERFORMANCE COMPARISON OF SOAF AND SIX STATE-OF-THE-ART ALGORITHMS ON BAD WEATHER CAMERA JITTER, INTERMITTENT OBJECT MOTION, LOW FRAMERATE, NIGHT VIDEOS, PTZ AND TURBULENCE SCENES UNDER THE METRICS OF RE, PR AND FM (FROM LEFT TO RIGHT IN EACH CELL)

	Bad Weather	Camera Jitter	Intermittent Object Motion	Low Framerate	Night Videos	PTZ	Turbulence
MST	0.60 0.77 0.64	0.72 0.40 0.51	0.57 0.60 0.45	0.61 0.29 0.34	0.58 0.43 0.42	0.80 0.02 0.04	0.68 0.49 0.53
RMoG	0.56 0.90 0.68	0.67 0.76 0.70	0.45 0.80 0.54	0.58 0.59 0.53	0.55 0.43 0.43	0.64 0.22 0.25	0.58 0.57 0.46
MoG	0.72 0.77 0.74	0.73 0.51 0.60	0.51 0.67 0.52	0.58 0.69 0.54	0.53 0.41 0.41	0.65 0.12 0.15	0.79 0.43 0.47
IMoG	0.69 0.81 0.74	0.69 0.49 0.57	0.55 0.65 0.53	0.53 0.67 0.51	0.48 0.42 0.40	0.61 0.07 0.10	0.78 0.35 0.42
MultiLayer	0.29 0.89 0.42	0.62 0.80 0.69	0.21 0.91 0.32	0.77 0.55 0.59	0.50 0.35 0.39	0.67 0.30 0.34	0.63 0.68 0.65
MultiCue	0.61 0.81 0.67	0.52 0.65 0.55	0.36 0.44 0.37	0.69 0.66 0.66	0.52 0.47 0.46	0.68 0.03 0.06	0.76 0.59 0.61
SoAF ₁	0.57 0.58 0.55	0.93 0.49 0.63	0.82 0.30 0.41	0.92 0.39 0.47	0.91 0.19 0.31	0.87 0.02 0.04	0.83 0.13 0.18
SoAF ₂	0.20 0.94 0.31	0.74 0.77 0.75	0.56 0.47 0.42	0.66 0.52 0.49	0.49 0.30 0.30	0.56 0.09 0.12	0.15 0.54 0.21

chooses the grayscale and texture features according to their stabilities in this scene, and the same phoneme has also shown in the result of MoG. Moreover, in the Camera Jitter, since the stable features enjoy the higher weights for segmenting the foreground, SoAF shows the sufficient robustness to against the camera jitter. Moreover, the multi-model we utilized for analyzing the features' stability also make a good contribution to the foreground segmentation of SoAF in this scene.

Unfortunately, SoAF does not perform well in the PTZ camera and Turbulence scenes. This is due to the fact that

the PTZ camera is a moving camera where the positions of the captured pixels are rapidly shifted in the time sequence. In the Turbulence category, since the features we chosen are mainly used for addressing the foreground detection issue in the normal scenes, it is hard to evaluate the stabilities of different features.

In this work, all the experiments are run on the PC with a 2.4 GHz CPU and 4G RAM in the Matlab 2010b environment under Windows 8. The size of all the images are kept the source size of dataset and all images have not been normalized.

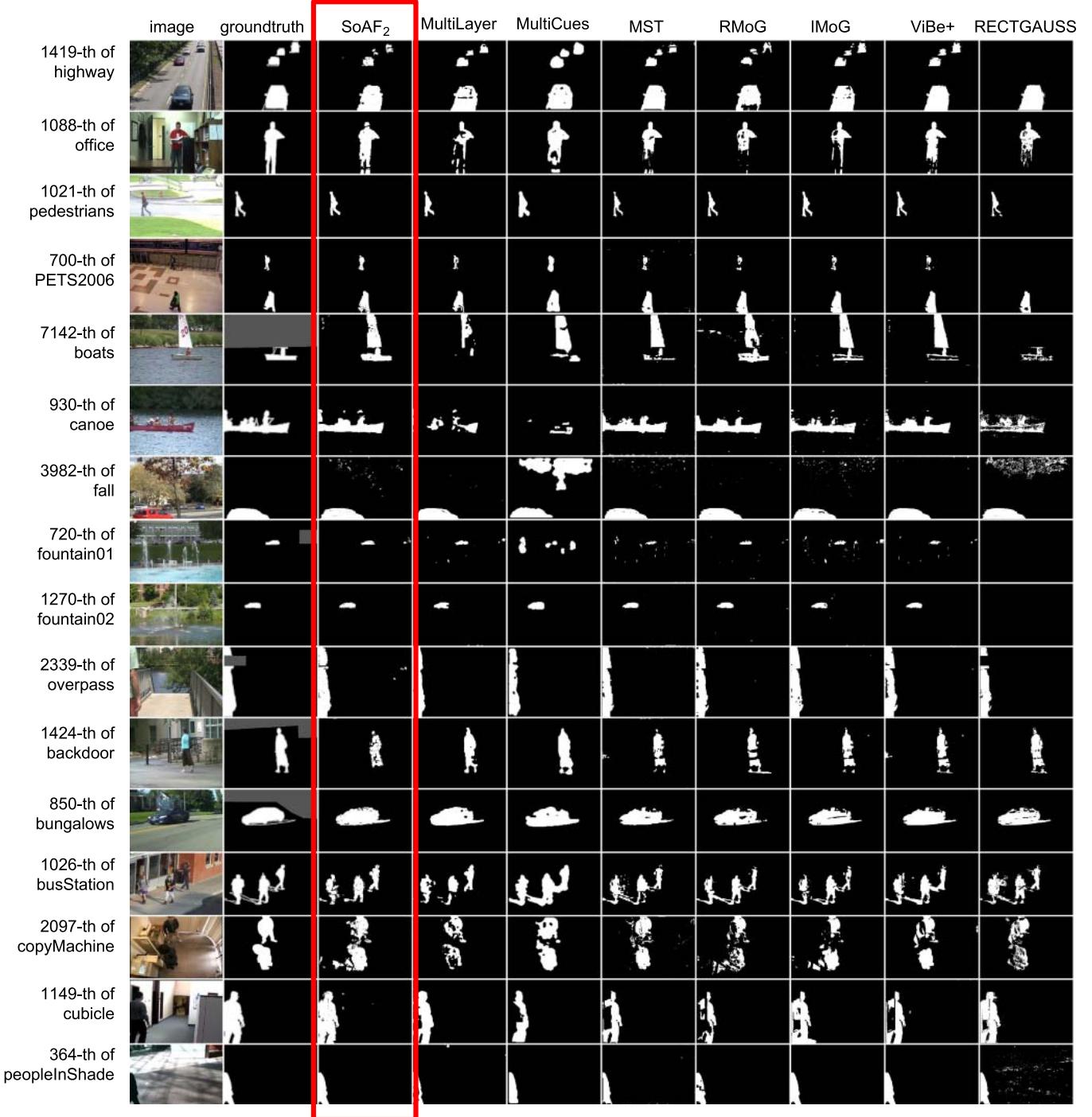


Fig. 10. The qualitative comparison of SoAF and other seven algorithms (include MultiLayer [39], MultiCue [40], MST [50], RMoG [51], ViBe+ [52], Zivkovics MoG [53] and RECTGAUSS [23].) on video sequences belong to the Baseline, Dynamic Background and Shadow scenes.

SoAF is implemented by C++, achieving a frame rate of 30fps.

D. Influence of Double Threshold

In this section, we experimentally analyze the influence of “double threshold” by ROC [36]. A ROC space was defined by FPR and TPR as x and y axes respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). The experiments are run in three frames on different video sequences and the results are shown

TABLE IV

THE PARAMETER VALUE OF SOAF₁ (FIRST ROW) AND SOAF₂ (SECOND ROW) FOR RESULTS IN TABLE II, TABLE III AND FIG. 10

L	R-SILTP ^T _{N,R,K}	a _b	a _m	T _h	T _l	R _T	K _j
4	R-SILTP ^T _{25,2,1}	0.99	0.99	0.5	0.1	8	(4,8,8,8,2)
4	R-SILTP ^T _{25,2,1}	0.99	0.99	0.7	0.3	8	(4,8,8,8,2)

in Fig. 12. The first row of Fig. 12 is the 700-th frame in PETS2006, which is used to evaluate the robustness for a moving object’s shadow, since there are people with obvious

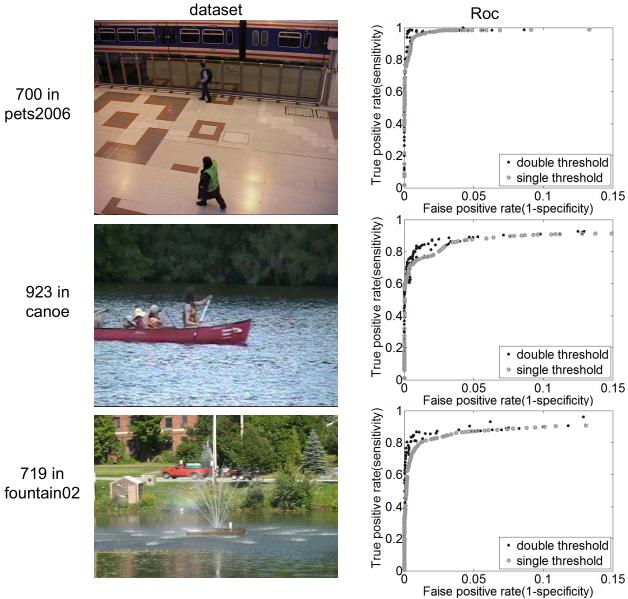


Fig. 11. The quantitative comparison between double threshold and single threshold in the ROC line, and line of double threshold is closer to the point $(0, 1)$.

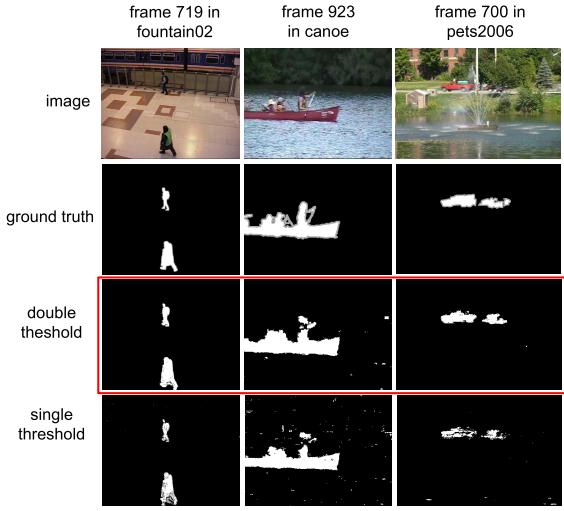


Fig. 12. The comparison of double threshold and single threshold for foreground detection. From top to bottom are the 700-th frame of pets2004, the 923-th frame of canoe, and the 719-th frame of fountain02. From left to right are source image, ground truth, foreground segmented by double threshold, foreground segmented by single threshold.

shadows in this frame. The second row of Fig. 12 is the 923-th frame in canoe, which belongs to the dynamic scenes. This image contains a canoe in the water, and the water usually produces a lot of noise. So we test which approach has better robustness for noise. The bottom row of Fig. 12 is used to test the ability of linking the break point, since the middle part of the car is blocked by the fountain, but not completely separated.

From Fig. 11, we see the ROC of double threshold is closer than the ROC of single threshold to the point $(0, 1)$ in all three images, which means the double threshold benefits more with less cost in foreground segmentation. We also show the foreground image that is segmented by double threshold and

single threshold to compare these two approaches visually. The result of comparison is shown in Fig. 12.

From Fig. 12, in the first row, we see the double threshold almost wipes off moving objects shadow while single threshold does not. In the second row of Fig. 12, the background image segmented by single threshold has many noisy points, while the background segmented by “double threshold” is clear. This situation demonstrates that double threshold is more robust for noise. Next, the third row of Fig. 12 shows that double threshold has better ability for linking break point, since the two parts of car are linked in the foreground segmented by “double threshold”.

VI. CONCLUSION

In this paper, we proposed a multi-feature background model named Stability of Adaptive Features (SoAF) for background modeling in complex scenes. This approach measures the stability of features and then adaptively selects different dominant features to model the background from the pixel and time-sequence domains. The dominant features are primarily used by the model while the secondary features are used as complements to overcome the detects brought about by the dominant feature and further enhance the model. The comprehensive results from the experiments conducted on the standard dataset, which contains both of complex scenes and baseline scenes, demonstrates that SoAF achieves promising performance.

REFERENCES

- [1] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Comput. Sci. Rev.*, vols. 11–12, pp. 31–66, May 2014.
- [2] Z. Chen and T. Ellis, “A self-adaptive Gaussian mixture model,” *Comput. Vis. Image Understand.*, vol. 122, pp. 35–46, May 2014.
- [3] B. Han and L. S. Davis, “Density-based multifeature background subtraction with support vector machine,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.
- [4] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, “Image change detection algorithms: A systematic survey,” *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.
- [5] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.
- [6] M. Heikkila and M. Pietikäinen, “A texture-based method for modeling the background and detecting moving objects,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.
- [7] 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.
- [8] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 1999, p. 252.
- [9] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Background modeling and subtraction by codebook construction,” in *Proc. Int. Conf. Image Process.*, vol. 5, Oct. 2004, pp. 3061–3064.
- [10] 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.
- [11] O. Barnich and M. Van Droogenbroeck, “ViBe: A powerful random technique to estimate the background in video sequences,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2009, pp. 945–948.
- [12] 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.
- [13] D. Kit, B. Sullivan, and D. Ballard, “Novelty detection using growing neural gas for visuo-spatial memory,” in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Sep. 2011, pp. 1194–1200.

- [14] H. Zhou, Y. Chen, and R. Feng, "A novel background subtraction method based on color invariants," *Comput. Vis. Image Understand.*, vol. 117, no. 11, pp. 1589–1597, Nov. 2013.
- [15] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection—A survey," *Recent Patents Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.
- [16] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, 1999, pp. 255–261.
- [17] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 543–560.
- [18] Y. Kita, "Background modeling by combining joint intensity histogram with time-sequential data," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 991–994.
- [19] C.-M. Kuo, W.-H. Chang, S.-B. Wang, and C.-S. Liu, "An efficient histogram-based method for background modeling," in *Proc. IEEE Int. Conf. Innov. Comput. Inf. Control*, Dec. 2009, pp. 480–483.
- [20] M. Heikkilä, M. Pietikäinen, and J. Heikkilä, "A texture-based method for detecting moving objects," in *Proc. Brit. Mach. Vis. Conf.*, vol. 28, no. 4, 2004, p. 21.1.
- [21] M. Heikkilä, M. Pietikäinen, and J. Heikkilä, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1301–1306.
- [22] C.-H. Yeh, C.-Y. Lin, K. Muchtar, and L.-W. Kang, "Real-time background modeling based on a multi-level texture description," *Inf. Sci.*, vol. 269, pp. 106–127, Jun. 2014.
- [23] D. Riahi, P. L. St-Onge, and G.-A. Bilodeau, "RECTGAUSS-tex: Block-based background subtraction," Dept. Génie Inform. Génie Logiciel, École Polytechn. Montréal, Montréal, QC, Canada, Tech. Rep. EPM-RT-2012-03, 2012, pp. 1–9.
- [24] L. Lin, Y. Xu, X. Liang, and J. Lai, "Complex background subtraction by pursuing dynamic spatio-temporal models," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3191–3202, Jul. 2014.
- [25] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection in changing background based on color co-occurrence statistics," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 2002, pp. 269–274.
- [26] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 774–780, Aug. 2000.
- [27] P. Chiranjeevi and S. Sengupta, "Neighborhood supported model level fuzzy aggregation for moving object segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 645–657, Feb. 2014.
- [28] A. Zaharescu and M. Jamieson, "Multi-scale multi-feature codebook-based background subtraction," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 1753–1760.
- [29] J. Chen, Y. He, and J. Wang, "Multi-feature fusion based fast video flame detection," *Building Environ.*, vol. 45, no. 5, pp. 1113–1122, 2010.
- [30] T. Xue, Y. Wang, and Y. Qi, "Multi-feature fusion based GMM for moving object and shadow detection," in *Proc. IEEE Int. Conf. Signal Process.*, vol. 2, Oct. 2012, pp. 1119–1122.
- [31] Z. Ji and W. Wang, "Detect foreground objects via adaptive fusing model in a hybrid feature space," *Pattern Recognit.*, vol. 47, no. 9, pp. 2952–2961, 2014.
- [32] Y. Dong and G. N. DeSouza, "Adaptive learning of multi-subspace for foreground detection under illumination changes," *Comput. Vis. Image Understand.*, vol. 115, no. 1, pp. 31–49, 2011.
- [33] Y. Wang, Y. Liang, L. Zhang, and Q. Pan, "Adaptive spatiotemporal background modelling," *IET Comput. Vis.*, vol. 6, no. 5, pp. 451–458, Sep. 2012.
- [34] C.-Y. Lin, C.-C. Chang, W.-W. Chang, M.-H. Chen, and L.-W. Kang, "Real-time robust background modeling based on joint color and texture descriptions," in *Proc. Int. Conf. Genetic Evol. Comput.*, 2010, pp. 622–625.
- [35] H. Zhang and D. Xu, "Fusing color and texture features for background model," in *Proc. 3rd Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2006, pp. 887–893.
- [36] K. H. Zou, A. J. O'Malley, and L. Mauri, "Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models," *Circulation*, vol. 115, pp. 654–657, Feb. 2007.
- [37] J. M. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin, "Foreground-adaptive background subtraction," *IEEE Signal Process. Lett.*, vol. 16, no. 5, pp. 390–393, May 2009.
- [38] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 105–112, Feb. 2002.
- [39] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
- [40] S. Noh and M. Jeon, "A new framework for background subtraction using multiple cues," in *Proc. Asian Conf. Comput. Vis.*, 2013, pp. 493–506.
- [41] B. Klare and S. Sarkar, "Background subtraction in varying illuminations using an ensemble based on an enlarged feature set," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2009, pp. 66–73.
- [42] F. E. Baf, T. Bouwmans, and B. Vachon, "Foreground detection using the Choquet integral," in *Proc. 9th Int. Workshop Image Anal. Multimedia Interactive Services*, May 2008, pp. 187–190.
- [43] Y. Chen, J. Wang, J. Li, and H. Lu, "Multiple features based shared models for background subtraction," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 3946–3950.
- [44] N. Greggio, A. Bernardino, C. Laschi, P. Dario, and J. Santos-Victor, "Self-adaptive Gaussian mixture models for real-time video segmentation and background subtraction," in *Proc. Int. Conf. Intell. Syst. Des. Appl.*, 2010, pp. 983–989.
- [45] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [46] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern Recognit.*, vol. 33, no. 1, pp. 43–52, Jan. 2000.
- [47] C. Silva, T. Bouwmans, and C. Frélicot, "An extended center-symmetric local binary pattern for background modeling and subtraction in videos," in *Proc. Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2015, pp. 395–402.
- [48] Q. Chen, Q.-S. Sun, P. A. Heng, and D.-S. Xia, "A double-threshold image binarization method based on edge detector," *Pattern Recognit.*, vol. 41, no. 4, pp. 1254–1267, 2008.
- [49] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changecoloration.net: A new change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 1–8.
- [50] X. Lu, "A multiscale spatio-temporal background model for motion detection," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 3268–3271.
- [51] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of Gaussians for dynamic background modelling," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2013, pp. 63–68.
- [52] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 32–37.
- [53] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. Int. Conf. Pattern Recognit.*, vol. 2, 2004, pp. 28–31.
- [54] A. Sobral, "BGSLibrary: An OpenCV C++ background subtraction library," in *Proc. 9th Workshop Viso Comput. (WVC)*, Rio de Janeiro, Brazil, Jun. 2013, pp. 38–43.



Dan Yang received the B.Eng. degree in automation, the M.S. degree in applied mathematics, and the Ph.D. degree in machinery manufacturing and automation from Chongqing University, Chongqing. From 1997 to 1999, he held a post-doctoral position with the University of Electro-Communications, Tokyo, Japan. He is currently the Vice President of Chongqing University, where he is also a Professor with the School of Software Engineering. He has authored over 100 scientific papers and some of them are published in some authoritative journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, CVPR, and BMVC. His research interests include computer vision, image processing, pattern recognition, software engineering, and scientific computing.



Chenqiu Zhao received the B.S. and M.S. degrees in software engineering from Chongqing University, Chongqing, China, in 2014 and 2017, respectively. He is currently a Research Associate with the Institute for Media Innovation, Nanyang Technological University, Singapore, and planning to seek opportunities for his Ph.D. degree. His current research interests include computer vision, video segmentation, pattern recognition, and deep learning.



Sheng Huang (S'14) received the B.Eng. degree in software engineering and the Ph.D. degree in computer science from Chongqing University, Chongqing, in 2010 and 2015, respectively. From 2012 to 2014, he was a Visiting Student with the Department of Computer Science, Rutgers University, Piscataway, NJ, USA. He has authored several scientific papers in venues, such as CVPR, BMVC, ICME, ISBI, and ICIP. His research interests include pattern recognition, computer vision, machine learning, and biometrics.



Xiaohong Zhang received the M.S. degree in applied mathematics from Chongqing University, China, where he also received the Ph.D. degree in computer software and theory, in 2006. He is currently a Professor with the School of Software Engineering, Chongqing University. His current research interests include data mining of software engineering, topic modeling, image semantic analysis, and video analysis.