EURASIP Journal on Image
and Video Processing

RESEARCH Open Access

Segmentation of moving objects in image sequence based on perceptual similarity of local texture and photometric features

K. L. Chan

Abstract

The segmentation of moving objects in image sequence can be formulated as a background subtraction problem—the separation of objects from the background in each image frame. The background scene is learned and modeled. A pixelwise process is employed to classify each pixel as an object or background based on its similarity with the background model. The segmentation is challenging due to the occurrence of dynamic elements such as illumination change and background motions. We propose a framework for object segmentation with a novel feature for background representation and new mechanisms for updating the background model. A ternary pattern is employed to characterize the local texture. The pattern and photometric features are used for background modeling. The classification of pixel is performed based on the perceptual similarity between the current pixel and the background model. The segmented object is refined by taking into account the spatial consistency of the image feature. For the background model update, we propose two mechanisms that are able to adapt to abrupt background change and also merge new background elements into the model. We compare our framework with various background subtraction algorithms on video datasets.

Keywords: Moving object segmentation, Background subtraction, Local ternary pattern, Video surveillance, Dynamic background

1 Introduction

Moving objects such as humans or vehicles are often the focus of image sequence analysis. The segmentation of moving objects is the first key problem which can be formulated as a background subtraction. In that sense, the background scene is modeled. Object pixels are segmented when they are found to be different from the background model. A common background subtraction framework contains background modeling, joint background/foreground classification, and background model updating. Survey on background subtraction techniques can be found in [1]. Sobral and Vacavant [2] presented a recent review and evaluation of 29 background subtraction methods. Background subtraction techniques can be categorized based on the ways to model the background scene. For instance, the background scene can be characterized in terms of statistical parameters. Bouwmans [3] presented a survey on statistical background modeling. Alternatively, the background model is represented as a bag of visual words—intensities or colors sampled over a short image sequence. Texture-based methods model the background scene by analyzing the relationship between the neighboring pixels.

1.1 Parametric methods

In [4], pixelwise background colors are modeled by a single Gaussian distribution. Stauffer and Grimson [5] proposed a modeling of background colors using a mixture of Gaussian (MoG) distributions that can tackle repetitive background motions and illumination changes. The background model is initialized using *K*-means which approximate an EM algorithm. Pixel value that does not match any of the most probable background distributions is regarded as foreground. Parameters of the MoG model are updated after background/foreground classification. Since its introduction, MoG has gained widespread popularity and inspired many improvements. For instance, in

Correspondence: itklchan@cityu.edu.hk
Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong

contrast with a fixed number of Gaussians in the original MoG model, Zivkovic [6] proposed an algorithm for selecting the number of Gaussian distributions using the Dirichlet prior. A comprehensive survey on the improvements of the MoG model can be found in [7].

1.2 Non-parametric methods

This category of background subtraction methods does not assume the pdf of the scene follows a known parametric form. Elgammal et al. [8] proposed an algorithm for estimating the pdf directly from the previous pixels using kernel estimator. Kim et al. [9] proposed to represent the background by codebooks which contain quantized background colors. Barnich and Van Droogenbroeck [10] proposed a sample-based background subtraction algorithm called ViBe. The background model is initialized by randomly sampling the pixels on the first image frame. The pixel of the new image frame is classified as background when there is a sufficient number of background samples similar to the new pixel. It defines a fixed sphere centered at the current pixel value and searches for similar background samples. If the sphere is too large, a foreground pixel may be wrongly labeled as background. If the sphere is too small, a background pixel may not find matched samples. Hofmann et al. [11] proposed a similar non-parametric sample-based background subtraction method with nine tunable parameters. Van Droogenbroeck and Paquot [12] introduced some modifications for improving ViBe. Haines and Xiang [13] presented a non-parametric background modeling method based on Dirichlet process Gaussian mixture models. Gao et al. [14] and Liu et al. [15] regarded the observed video frames as a matrix, which can be decomposed into a low-rank matrix of background and a structured sparse matrix of the foreground. Monnet et al. [16] proposed an online auto-regressive model to predict the new image frame. Change is detected as a difference between the prediction and actual observation. Mahadevan and Vasconcelos [17] proposed to model the background by dynamic texture (DT) features obtained in the center and surrounding windows. Background subtraction is performed by calculating the mutual information between DT features and classes (background, foreground). Sheikh and Shah [18] presented a non-parametric density estimation method to model the background. In [19], dynamic background is modeled by a local dependency histogram which characterizes the spatial dependencies between a pixel and its neighbors.

1.3 Texture-based methods

Recent research showed that modeling background by local patterns can achieve higher accuracy. Heikkilä and Pietikäinen [20] proposed to model the background of a pixel by local binary pattern (LBP) histograms estimated around that pixel. Their method can segment moving objects correctly in indoor image sequence even though the colors of the foreground and the static background are similar. It also can segment moving objects in outdoor image sequence with dynamic elements because the pattern exploits information over a larger area than a single pixel. Liao et al. [21] proposed the scale invariant local ternary pattern (SILTP) which can tackle illumination variations. It can perform poorly in flat regions and produces holes in the segmented objects. St-Charles et al. [22] proposed a pixelwise background modeling method using local binary similarity pattern (LBSP) estimated in the spatio-temporal domain. Ma and Sang [23] proposed the multi-channel SILTP (MC-SILTP), which is an improvement of SILTP, with a pattern computed from RGB color channels. As shown in our experimentation in Section 6, it still results in many misclassifications. Background modeling by analyzing the neighboring pixels using neural networks and Markov random fields is also proposed. Spatiotemporal background modeling has gained more attention recently, e.g., [24, 25], because it also contains motion information—the variation of the local pattern over time.

1.4 Background model update and object refinement

Object segmentation can be improved via background model updating or foreground model. Many background subtraction methods like [5] update parameters of matched background model with a fixed learning factor. In [11], the foreground decision threshold and model update rate can be adaptively adjusted along the video sequence. In [10], a random policy is employed for updating the background model at the pixel location and its neighbor. Van Droogenbroeck and Paquot [12] inhibited the update of neighboring background model across the background-foreground border. In [22], a pixel-level feedback scheme can adjust some parameters to respond to the background changes. That makes the modeling method capable of tackling both static and dynamic background. Kim et al. [26] proposed a PID tracking control system for the foreground segmentation refinement. In [27], MoGs are used to model the color distribution of swimmer pixels. Sheikh and Shah [18] also used a non-parametric density estimation method to model the foreground.

In [28], we proposed a novel perception-based local ternary pattern for background modeling. In this paper, our first contribution is to extend the perception-based local ternary pattern and its features. Instead of summing the pattern codes to form one feature as in [28], in this paper, each pattern is represented by the cardinalities of individual pattern codes, and the dimension of the pattern feature is increased to nine. Our perception-based local ternary pattern makes full use of color information which is better than other texture-based methods like [21] using only gray values. With thorough explanation and real examples, we demonstrate that the texture pattern is more informative, and its features can be used effectively to characterize various dynamic circumstances in the scene. In our second contribution, we propose a new background model update and object refinement processes. Many background subtraction methods have very simple background model updating. For instance, [10] updates the background model at the pixel location and its neighbor randomly. The sensitivity to background change may be slow. Our background model update can tackle abrupt background change and also merge new elements (e.g., non-moving objects) into the background model. Many background subtraction methods, e.g., [22], devise complex feedback scheme for background model update. However, they lack any object refinement process. In our method, the segmented object can be refined by taking into account the spatial consistency of the image feature. This step can result in a more accurate segmented shape of the object.

2 Method

Many background subtraction methods assume that the scene is stationary or changes slowly. Background subtraction becomes harder under various complex circumstances—camouflage, illumination changes, background motions, intermittent object motion, shadows, camera jitter, etc. It is a challenging task to achieve accurate background/foreground segregation in videos containing those dynamic scenes and over a long duration. The features for modeling the background scene, which are also used in the background-foreground classification, are very important. A pattern, with multiple pixels, can characterize the local texture more effectively than an individual pixel. Also, the refinement process is needed for improving the segmented object. To detect a moving object in a long image sequence, background model updating is necessary.

Our method, as shown in Fig. 1, contains three sequential parts (background model initialization, moving object segmentation, object refinement) and a feedback scheme (background model update). The background model is generated by analyzing the initial frames of the image sequence. The background model contains local texture and photometric features per pixel. The principle of perception-based local ternary pattern and the estimation of texture features are explained in the following section. The computation of the photometric feature and the generation of the initial background model are presented in Section 4. The background model is then used to segment moving objects in each frame of the image sequence. The object segmentation framework is described in Section 5. The moving objects just

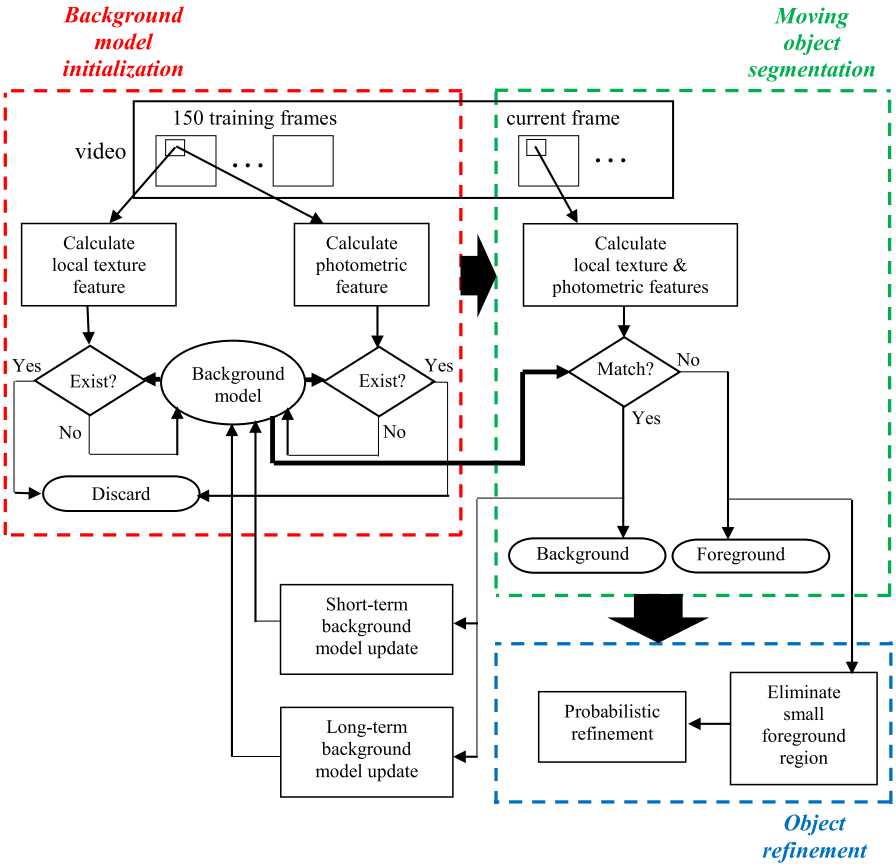
Background model initialization
Moving object segmentation
video
150 training frames
current frame
...
Calculate local texture feature
Calculate photometric feature
Calculate local texture & photometric features
Yes
Exist?
No
Background model
Exist?
Yes
No
Match?
No
Yes
Discard
Background
Foreground
Short-term background model update
Long-term background model update
Probabilistic refinement
Eliminate small foreground region
Object refinement

Fig. 1 Overview of our moving object segmentation method

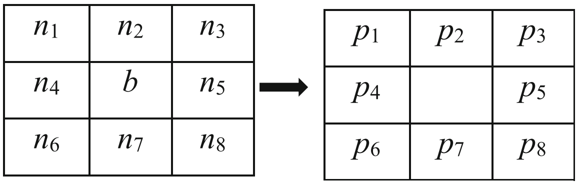

Fig. 2 A block of pixels with center pixel *b* and neighbors n_1 to n_8 is transformed into a ternary pattern

segmented can be refined by the probabilistic scheme. Finally, the background model is updated based on the segmentation result. Object refinement and background model update are explained in Section 6. We test and compare our method with various background subtraction algorithms using common video datasets. The numeric and visual results are demonstrated and discussed in Section 7. Finally, we draw a conclusion in Section 8.

3 Perception-based local ternary pattern

We propose a novel perception-based local ternary pattern (P-LTP) which can be used effectively to characterize various dynamic circumstances in the scene. Figure 2 shows a block of 3 × 3 pixels. Each pixel of the block, n_1 to n_8 (except the center pixel), is compared with the confidence interval (CI) of the center pixel *b*. CI(*b*) is defined by (CI_l, CI_u) where CI_l and CI_u are the lower bound and upper bound of CI, respectively. The pattern value *p* is set according to the following equation:

$$p_k = \begin{cases} 0, & \mathrm{CI}_l \le n_k \le \mathrm{CI}_u \\ 1, & n_k > \mathrm{CI}_u \\ -1, & n_k < \mathrm{CI}_l \end{cases}, 1 \le k \le 8 \qquad (1)$$

The confidence interval CI(*b*) can be defined as ($b - d_1$, $b + d_2$). According to Weber's law [29], d_1 and d_2 depend on the perceptual characteristics of *b*. That is, they should be small for darker color and large for brighter color. Haque and Murshed [30] derived the linear relationship $d_1 = d_2 = c \times b$, where *c* is a constant. The background model is represented by the confidence interval centered at the mean value or the intensity of a recently observed background pixel. A new pixel is identified as a background if its intensity falls within the confidence interval. We adopt the human visual perception characteristics in transforming pixel colors into a local ternary pattern. In P-LTP, CI(*b*) is defined as ($b - c_1 b$, $b + c_2 b$). Using peak signal-to-noise ratio (PSNR) measure, *b* and $b - c_1 b$ are just perceptually different from each other if:

$$20\ \log_{10}\frac{I_{\max}}{b - c_1 b} - 20\ \log_{10}\frac{I_{\max}}{b} = T_p \qquad (2)$$

where $I_{\max}$ is the maximum intensity, and T_p is the perceptual threshold. Similarly, *b* and $b + c_2 b$ are just perceptually different from each other if:

$$20\ \log_{10}\frac{I_{\max}}{b} - 20\ \log_{10}\frac{I_{\max}}{b + c_2 b} = T_p \qquad (3)$$

To determine c_1 and c_2, the equations are simplified.

$$c_1 = \frac{10^{\frac{T_p}{20}} - 1}{10^{\frac{T_p}{20}}} \qquad (4)$$

$$c_2 = 10^{\frac{T_p}{20}} - 1 \qquad (5)$$

Assume that T_p is 1.0 dB, $c_1 = 0.1087$, and $c_2 = 0.1220$.

Figure 3a illustrates the formation of a conventional local binary pattern from a block of 3 × 3 pixels. The

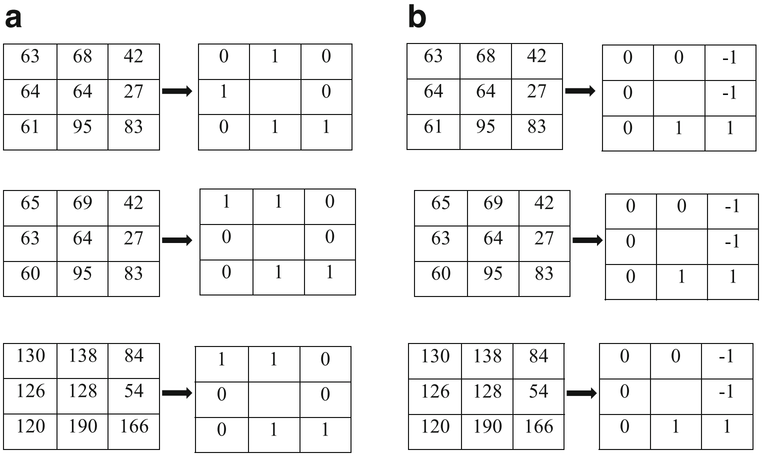

Fig. 3 Formation of local pattern **a** LBP. **b** P-LTP

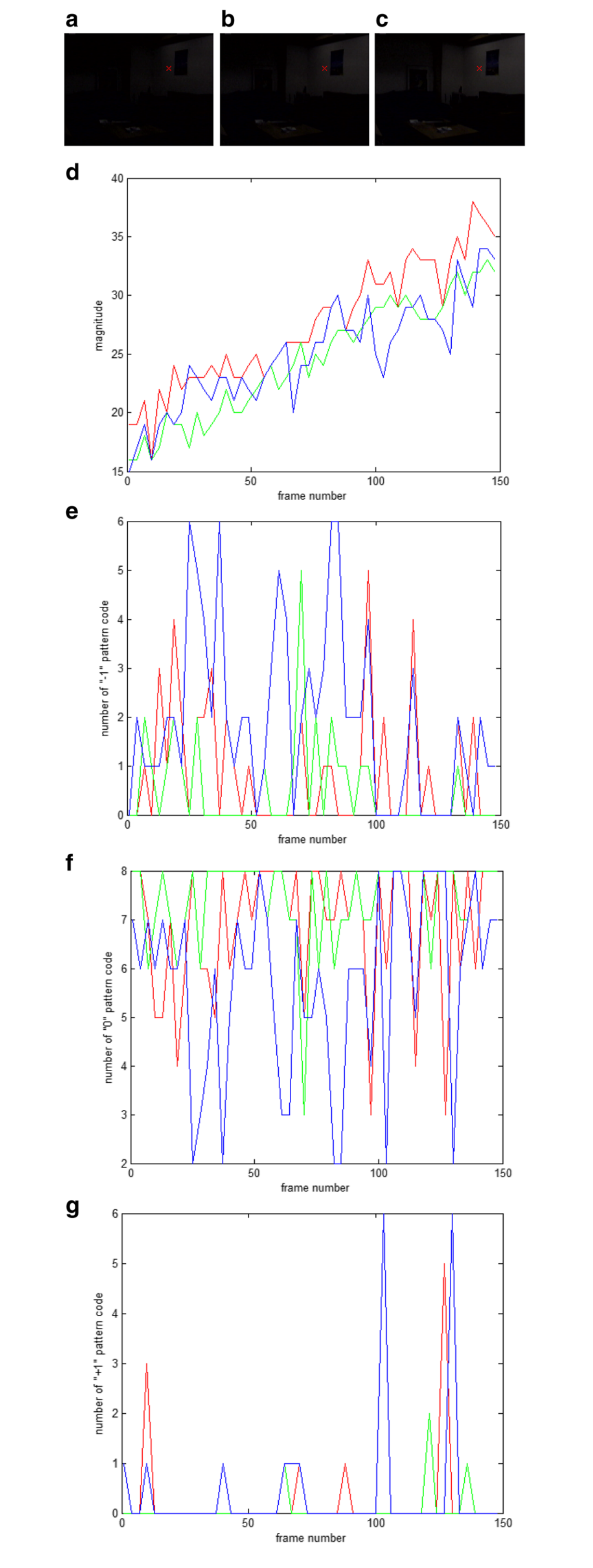

Fig. 4 Frames of an image sequence and plots of pixel values and P-LTP features for three color components (in red, green, and blue lines). **a** Frame 0. **b** Frame 74. **c** Frame 149. **d** Magnitudes of *R*, *G*, and *B* versus frame number. **e** Cardinality of "− 1" pattern code. **f** Cardinality of "0" pattern code. **g** Cardinality of "+ 1" pattern code

first row shows the formation of LBP for a noise-free image. The second row indicates that LBP is not robust to additive random noise in the image. The third row also shows that LBP cannot keep its invariance against random noise plus scale transform of intensity. Figure 3b illustrates the formation of P-LTP using Eq. (1) and the estimated c_1 and c_2 just mentioned under the same circumstances as in Fig. 3a. The confidence intervals for the patterns in the first and second rows are 57 and 72. The confidence intervals for the pattern in the third row are 114 and 144. It can be seen that P-LTP is always the same which means it is robust against random noise and scale transform.

For each color component, pixel values are transformed into ternary pattern codes as mentioned previously. The pattern is concisely represented by the cardinalities of individual pattern codes.

$$P_{c,t} = |p_k = t|, 1 \le k \le 8, t = \{-1, 0, 1\}, c = \{R, G, B\} \tag{6}$$

Finally, a nine-dimension P-LTP feature vector is formed. The advantage of the P-LTP feature over the original pixel value can be seen in Fig. 4. Figure 4a–c shows some image frames, and the selected point is indicated by the "x" mark. The illumination of the scene is gradually increasing. Figure 4d shows the plots in red, green, and blue for the *R*, *G*, and *B* pixel values, respectively, of the selected point. It is clear that magnitudes of the color components are increasing along the image sequence due to lighting up of the scene. Figure 4e–g shows the plots of cardinalities of pattern codes "− 1," "0," and "+ 1," respectively, along with the image sequence. Each figure shows the plots in red, green, and blue for the cardinalities of a pattern code estimated from the *R*, *G*, and *B* pixel values, respectively. In contrary to the magnitudes of color components, the nine P-LTP features are quite stable along the image sequence due to their invariance to intensity change. The selected point is on a smooth texture region. Cardinality of "0" pattern code is higher than that of "− 1" and "+ 1" pattern codes. Liao et al. [21] proposed a scale invariant local ternary pattern (SILTP) to represent the gray levels of the pixels. However, as shown in Fig. 5a, SILTP feature is not rotation invariant. For instance, if the image is rotated 90° clockwise, the feature value becomes 97. The P-LTP feature, as shown in Fig. 5b, is rotation invariant. The range of P-LTP feature for one color component is 9!, which is much larger than that of SILTP feature with 256-bin histogram representation.

4 Background model initialization

Many computer vision systems demand accurate segmentation of moving objects in the video. It is common that the method involves a learning process, and the background model is generated from the video. The background model must be versatile since simple or complex scenes may be encountered. The features representing the background scene are very important. In our framework, the background is modeled using textural and photometric features.

We observed various challenges in real scenes. Dynamic background elements such as tree and water produce many false-positive errors. Camera jitter also produces false-positive errors. It is because the background model does not contain sufficient and representative samples. Same as [21], we use 150 image frames in the background model initialization. At each pixel, the P-LTP features are computed with a fixed block size of 3 × 3 pixels. Only distinctive P-LTP features are entered into the background model. If a similar P-LTP feature already exists in the background model, it will be discarded. Let B be the background model and b is the number of P-LTP features in the background model. Let N be the number of initialization image frames and n is the frame index. The algorithm for P-LTP feature selection is given below.

```
Algorithm for P-LTP feature selection
$B \leftarrow \varnothing$, $B_{temp} \leftarrow \varnothing$, $b = 0$
For each pixel
    For $n = 1$ to $N$
        Compute $P^n_{c,t}$
        If $p\_dist(P^n_{c,t}, P^i_{c,t}) > \varepsilon_p$, $1 \le i \le b$, $c = \{R, G, B\}$
            $b = b + 1$
            $o_b = 1$
            $B_{temp}(b) \leftarrow P^n_{c,t}$
        Else
            $k = \arg\min_i p\_dist(P^n_{c,t}, P^i_{c,t})$
            $o_k = o_k + 1$
    For $i = 1$ to $b$
        If $o_i > \varepsilon_o$
            $B \leftarrow B_{temp}(i)$
```

The similarity between the current P-LTP feature and the existing P-LTP feature in the background model is calculated by:

$$p\_\mathrm{dist}\left(P^n_{c,t}, P^i_{c,t}\right) = \sum_{t=\{-1,0,1\}} \left|P^n_{c,t} - P^i_{c,t}\right| \tag{7}$$

If the cardinality of a pattern code differs by one, there is also another discrepancy of one in the cardinality of

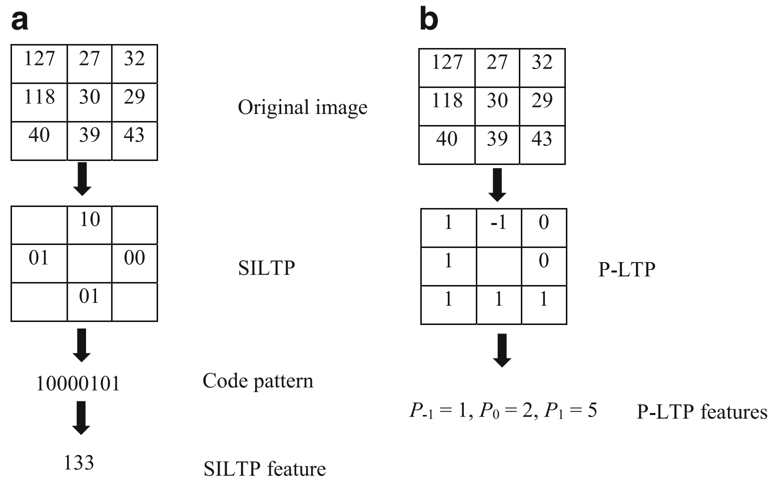

Fig. 5 Pattern feature estimation: (a) SILTP and (b) P-LTP

another pattern code. We assume that if two sets of P-LTP features are similar, the discrepancy of pattern code cardinality should not occur more than twice. Therefore, we fixed ε_p as 4. Also, we count the number of occurrences o_i for each distinctive P-LTP feature. In case the initialization sequence contains a moving object, its P-LTP feature should not be included in the background model. Assume that the moving object P-LTP feature occurs less often than the background P-LTP feature. We fixed the filter threshold ε_o as 10% of the length of the initialization sequence.

Besides P-LTP features, we also enter photometric features into the background model. We observed that for homogeneous textures, the P-LTP feature is not sufficient to differentiate the background and object even though they are of different color. Figure 6 shows two homogeneous textures, their colors and P-LTP features. The first row shows some image frames, and the selected point is indicated by the "x" mark. Figure 6d shows the color values of the selected point. Colors of the human are different from the background colors. The P-LTP features (Fig. 6e–g) are more or less the same for the two textures. The photometric features used for background modeling are the color components $C = \{R, G, B\}$ and intensity I which is calculated by:

$$I = \sqrt{R^2 + G^2 + B^2} \tag{8}$$

Only distinctive photometric features are entered into the background model. The algorithm for photometric feature selection is given below.

Algorithm for photometric feature selection

```
$B \leftarrow \varnothing, B_{temp} \leftarrow \varnothing, b = 0$
For each pixel
    For $n = 1$ to $N$
      Compute $I$
      If $c\_dist(C^n, C^i) > \varepsilon_c, 1 \le i \le b$
          $b = b + 1$
          $o_b = 1$
          $B_{temp}(b) \leftarrow C^n, I$
      Else
          $k = \arg\min_i c\_dist(C^n, C^i)$
          $o_k = o_k + 1$
  For $i = 1$ to $b$
      If $o_i > \varepsilon_o$
          $B \leftarrow B_{temp}(i)$
```

We adopted [9] to measure the similarity between the current pixel color and the existing photometric feature in the background model:

$$c\_\mathrm{dist}\left(C^n, C^i\right) = \sqrt{\|C^n\|^2 - \frac{\left(C^n \cdot C^i\right)^2}{\left\|C^i\right\|^2}} \tag{9}$$

We fixed ε_c as 10.

5 Moving object segmentation

Moving objects in the image sequence are segmented by comparing each pixel of the image frame with the background model. It is a background/foreground segregation process. If features of the pixel match with the background model, it is classified as background. Otherwise, it is a foreground (object) pixel. The algorithm for moving object segmentation is shown below.

Algorithm for Moving Object Segmentation

```
$F \leftarrow \varnothing, f = 0, B_{temp} \leftarrow \varnothing$
For each pixel p
    Compute $P_{c,t}, I, CI(I)$
    If $segment\_cond = 1$
        If $p\_dist(P_{c,t}, P^i_{c,t}) < \varepsilon_p \wedge c\_dist(C, C^i) < \varepsilon_c, \wedge CI_l(I) \le I^i \le CI_u(I), 1 \le i \le b$
            $p \leftarrow background$
        Else
            $p \leftarrow foreground$
    Else
        If $p\_dist(P_{c,t}, P^i_{c,t}) < \varepsilon_p$
            $p \leftarrow background$
        Else
            $p \leftarrow foreground$
    If $p = background$
        $B_{temp} \leftarrow P_{c,t}, C, I$
    Else
        If $p\_dist(P_{c,t}, P^i_{c,t}) > \varepsilon_p, 1 \le i \le f, c = \{R, G, B\}$
            $f = f + 1$
            $o_f = 1$
            $F(f) \leftarrow P_{c,t}$
        Else
            $k = \arg\min_i p\_dist(P_{c,t}, P^i_{c,t})$
            $o_k = o_k + 1$
        If $c\_dist(C, C^i) > \varepsilon_c, 1 \le i \le f$
            $f = f + 1$
            $o_b = 1$
            $F(f) \leftarrow C, I$
        Else
            $k = \arg\min_i c\_dist(C, C^i)$
            $o_k = o_k + 1$
```

Under the default condition ($segment\_cond = 1$), all features (P-LTP, color, and intensity) of the pixel are used. The segmentation can reduce false-negative errors due to the homogeneous textures within the object region. However, when the condition (e.g., illumination) of

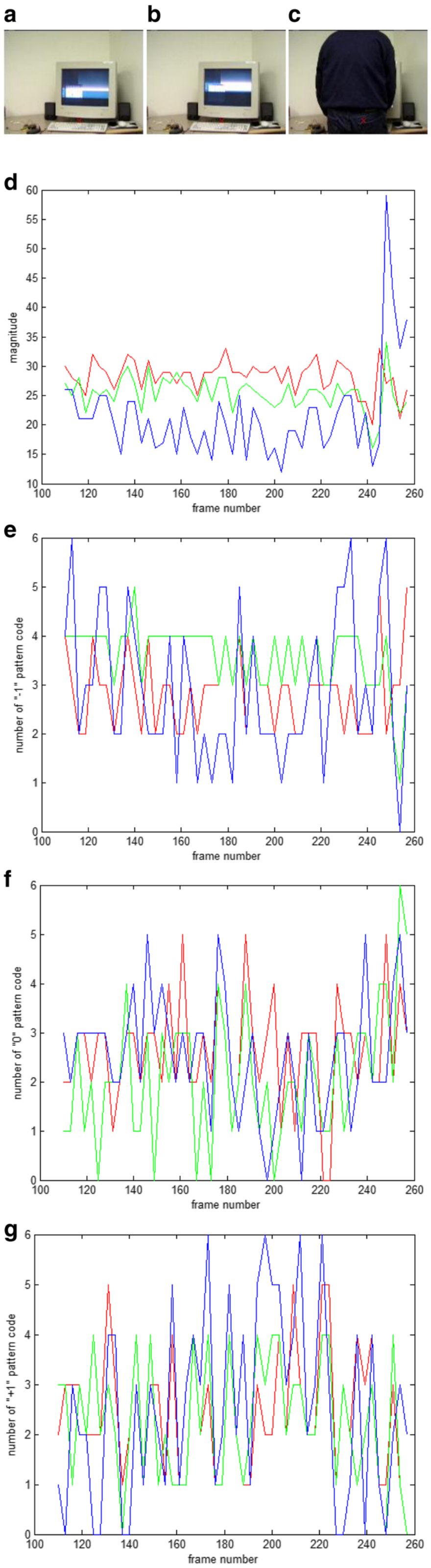

Fig. 6 Two homogeneous textures and the corresponding colors and P-LTP features (in red, green, and blue lines). **a** Frame 110. **b** Frame 184. **c** Frame 259. **d** Magnitudes of *R*, *G*, and *B* versus frame number. **e** Cardinality of "– 1" pattern code. **f** Cardinality of "0" pattern code. **g** Cardinality of "+ 1" pattern code

the scene changes drastically, photometric features are not invariant and will not be used in the classification. Control of *segment_cond* will be explained in the next section. The similarity of the intensity is measured with respect to the confidence interval of the current pixel's intensity. The lower and upper bounds are computed based on the constants c_1 and c_2 as explained in Section 3. If the current pixel is classified as background, its features are saved temporarily. If it is classified as foreground, we save and count the number of occurrences for each distinctive feature. This foreground model F will be analyzed in the background model update process as explained in the following section.

6 Object refinement and background model update

The background/foreground segregation result may contain false-positive and false-negative errors. For instance, isolated scene pixels may have features deviate from the background model due to illumination change or background motion. As they are not connected to form a region, they can be discarded without affecting the detection of real moving objects. Therefore, foreground regions less than 15 pixels are eliminated. The remaining foreground regions may have holes. The object's silhouette may be distorted. These false-negative errors are usually caused by the similarity of the object's feature to the background model. We analyze the spatial consistency of the image feature and refine the object probabilistically. Let x be a foreground (FG) pixel. Its neighboring background (BG) pixels y are defined by:

$$y \mid \text{dist}(x,y) < D, x = FG, y = BG \tag{10}$$

where dist() is the city block distance and D is fixed as 1. y is changed to FG when they have image features more similar to neighboring FG pixels than neighboring BG pixels. If *segment_cond* is 1, the image feature to be analyzed is P-LTP.

$$y_i = \text{FG} \ \text{if} \quad \log\frac{P(y_i = \text{FG})}{P(y_i = \text{BG})} > T_f \tag{11}$$

$$P(y_i = \mathrm{FG}) = \exp\left(-\sum_j \left|P^{y_j}_{c,t} - P^{y_i}_{c,t}\right|\right), \mathrm{dist}\left(y_i, y_j\right) < D, y_j = \mathrm{FG} \tag{12}$$

$$P(y_i = \mathrm{BG}) = \exp\left(-\sum_j \left|\mu\left(P^{y_j}_{c,t}\right) - P^{y_i}_{c,t}\right|\right), \mathrm{dist}\left(y_i, y_j\right) < D, y_j = \mathrm{FG} \tag{13}$$

where $\mu()$ is the mean of the P-LTP features in the background model. If *segment_cond* is 2, the consistency of the color component is checked.

$$P(y_i = \mathrm{FG}) = \exp\left(-\sum_j \left|C^{y_j} - C^{y_i}\right|\right), \mathrm{dist}\left(y_i, y_j\right) < D, y_j = \mathrm{FG} \tag{14}$$

$$P(y_i = \mathrm{BG}) = \exp\left(-\sum_j \left|\mu\left(C^{y_j}\right) - C^{y_i}\right|\right), \mathrm{dist}\left(y_i, y_j\right) < D, y_j = \mathrm{FG} \tag{15}$$

T_f is fixed as 2. Figure 7 shows the original image frame and the segmented human with and without probabilistic refinement. Many holes inside the human region are filled up. The silhouette is also improved.

We devise two mechanisms to adapt the background model to complications such as a sudden global change in the scene and appearance of new background features. They are important to ensure accurate object segmentation over a long image sequence. In the short-term background model update, an abrupt and extensive scene change is identified by analyzing the current segmented result S_t and a previous segmented result S_{t-k}, where t is the current time instant and k is fixed as 15. Normally, the background model will allow all stored features (P-LTP, color, and intensity) being used in the moving object segmentation (*segment_cond* = 1). It will be changed (*segment_cond* = 2) such that only P-LTP features are used when the following two conditions are met:

$$\sum \mathrm{XOR}(S_t, S_{t-k}) > T_S \tag{16}$$

$$\frac{|\mu(I_t) - \mu(I_{t-k})|}{\mu(I_t)} > T_I \tag{17}$$

where T_S is fixed as 0.3 times the image frame size, $\mu(I)$ is the mean intensity, and T_I is fixed as 0.2. Figure 8 shows the original image frames and the segmented results before and after the sudden scene change. Initially, all the background pixels are identified correctly. When some lights are turned off, many pixels are wrongly classified as foreground. With short-time background updating, many pixels are correctly identified as background immediately.

In the long-term background model update, we accumulate the outcomes of background/foreground classification over a period of time. We choose the same duration for background initialization which is 150 frames. During this period, if the current pixel is classified as background, its features are considered as reliable background features and will be stored temporarily. If it is classified as foreground, there are two possibilities. It may be a true object pixel or a new background pixel. Each distinctive foreground pixel will have its features saved into the foreground model F. We also count the number of occurrences for each distinctive foreground feature. By the end of the period, if the number of occurrences is large enough (we fixed the threshold as 0.8 times the update period), those foreground features, together with all the temporary background features, are saved as the updated background model. Figure 9 shows the original image frames and the segmented results

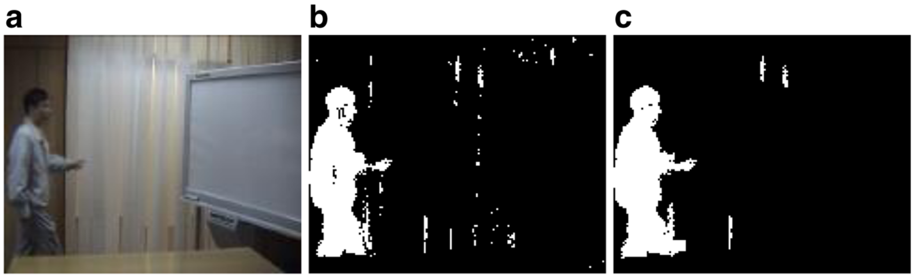

Fig. 7 Probabilistic object refinement. **a** Original image frame. **b** Segmented human without refinement. **c** Segmented human with probabilistic refinement

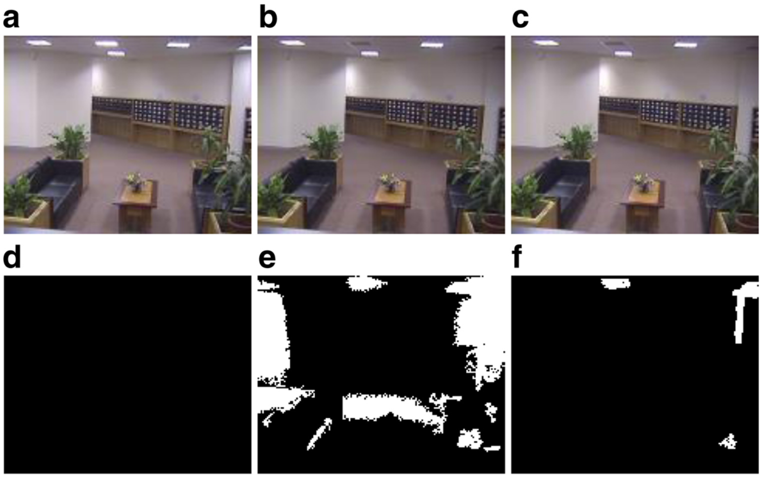

Fig. 8 Short-term background model update. **a**–**c** Original image frames 124, 126, and 128. **d**–**f** Segmented results

before and after the long-term background update. The illumination of the scene is increasing and causes more false-positive errors. At time instant 450, the long-term background update is carried out. All the background pixels are correctly identified.

7 Results and discussion

We evaluated the performance quantitatively in terms of Recall, Precision, and F-Measure (F1). Recall gives the ratio of detected true-positive pixels to a total number of foreground pixels present in the ground truth which is the sum of true-positive and false-negative pixels. Precision gives the ratio of detected true-positive pixels to a total number of foreground pixels detected by the method which is the sum of true-positive and false-positive pixels. F1 is the weighted harmonic mean of Precision and Recall. It can be used to rank different methods. The higher the value of F1, the better is the accuracy.

We compared our method with other background subtraction algorithms on three publicly available datasets. The Wallflower dataset [31] contains six videos. Each video comes with one manually labeled ground truth. They have the same image frame size of 160 × 120 pixels. The Star dataset [32] contains nine videos. Each video comes with 20 manually labeled frames as ground truths. The videos have a different image frame size, from 160 × 120 pixels to 320 × 256 pixels. Finally, we tested our method and compared with other algorithms on ChangeDetection.net dataset [33]. Each video comes with large amount of labeled ground truths. They have larger image frame size, from 320 × 240 pixels to 720 × 576 pixels.

As for our method, the first 150 image frames of the video are used for background model initialization which

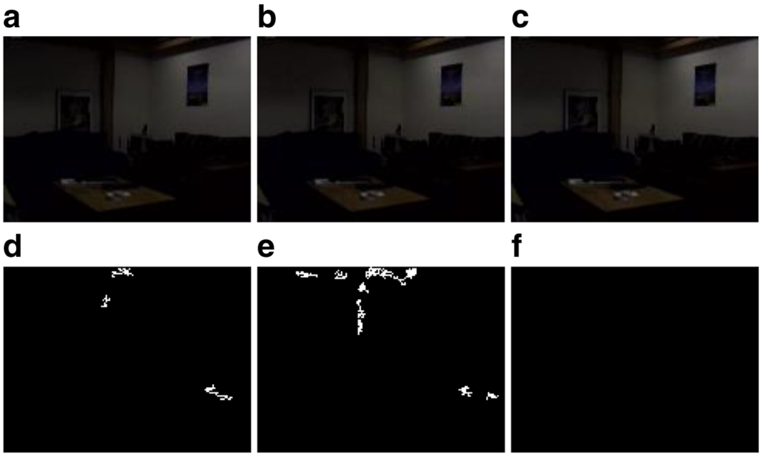

Fig. 9 Long-term background model update. **a**–**c** Original image frames 370, 410, and 450. **d**–**f** Segmented results

Table 1 F1 results on the Wallflower dataset

Sequence	Our method	ViBe	SILTP	MC-SILTP
Bootstrap	*0.867*	0.478	0.766	0.740
Camouflage	*0.975*	0.931	0.927	0.896
ForegroundAperture	0.693	0.644	*0.849*	0.665
LightSwitch	0.700	0.159	0.730	*0.745*
TimeOfDay	*0.832*	0.394	0.175	0.181
WavingTrees	*0.952*	0.933	0.712	0.946
Average	*0.836*	0.590	0.693	0.695
Variance	*0.015*	0.095	0.071	0.075

The best results are shown in italics

is the same as the best setting in [21]. Other parameters are: $T_p = 1.0$, $\varepsilon_p = 4$, $\varepsilon_o = 15$, $\varepsilon_c = 10$, $T_f = 2$, $T_S = 0.3$ × image frame size, and $T_I = 0.2$. The perceptual threshold of 1.0 dB corresponds to just visually observable difference between neighboring pixels. If a pattern is a background, we assume that any one pattern code cardinality should not differ from the existing background pattern by more than 2. Therefore, the total difference between the pattern features should not be more than 4. Through experimentation, we observed that moving object pattern occurs less than 10% of the length of the initialization sequence. The photometric feature similarity threshold is set the same as [9]. False-negative error is rectified when its features are more likely to be foreground than background. Through experimentation, we found that the threshold for the log-likelihood ratio can be set as 2 properly. To detect a global change in the

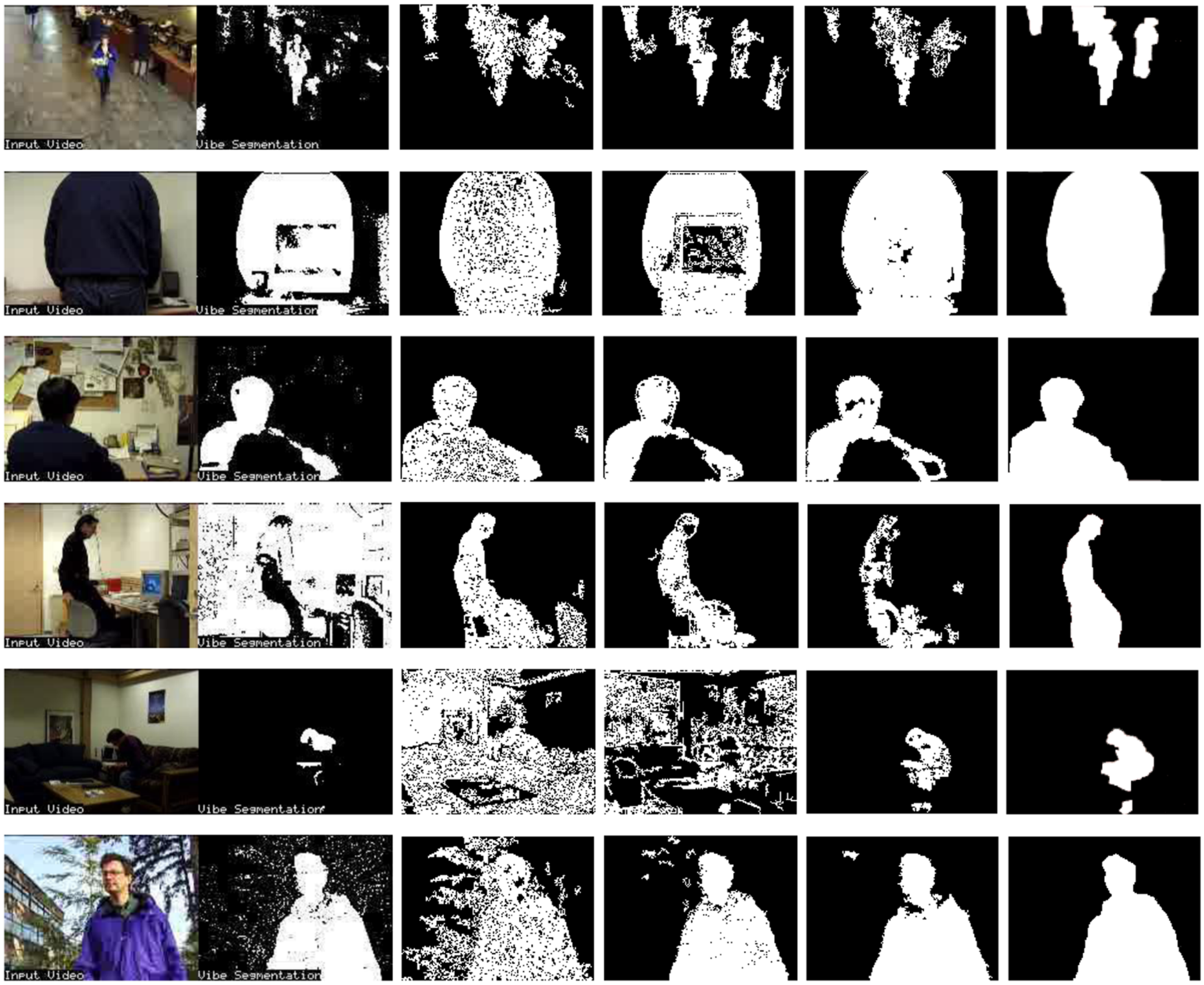

Fig. 10 Background subtraction results on the Wallflower dataset—original image frames (first column), results obtained by ViBe (second column), results obtained by SILTP (third column), results obtained by MC-SILTP (fourth column), results obtained by our method (fifth column), and ground truths (last column)

scene and appearance of new background features, we found that the conditions of 30% change in the segmentation result and fractional change of mean intensity of 0.2 are suitable for all testing image sequences.

7.1 Wallflower dataset

We selected non-parametric method ViBe [10], texture-based method SILTP [21], and MC-SILTP [23] for comparison. Based on sample consensus, ViBe can achieve very good results with very few tunable parameters. It was showed that ViBe performs better than many state-of-the-art parametric and non-parametric algorithms such as [6, 9]. ViBe uses RGB color model and a fixed spherical distance of 30 in matching new pixel with background samples. It keeps 20 background samples, and the new pixel is identified as background with two matches. SILTP employs scale invariant local patterns. It was showed [21] that it performs better than other LBP-based methods [20]. MC-SILTP is one latest texture-based method and can perform even better with pattern obtained by cross-computation on RGB color channels. We implemented SILTP with the same set of parameters as reported in [21]. The only parameter value which was not mentioned is the number of training frames. Through experimentation, we find that the number of training frames is best fixed as 150. Similarly, we implemented MC-SILTP with the same setting as reported in [23]. The Wallflower dataset contains videos exhibiting gradual illumination change (TimeOfDay), sudden illumination change (LightSwitch), similar background and object color (Camouflage), moving background elements (Waving Trees), etc.

Table 1 shows the F1 results of our method, ViBe, SILTP, and MC-SILTP. The best result in a given row is highlighted. No method can achieve the highest F1 on all videos. Our method can achieve the highest F1 on four videos. Overall, texture-based methods perform better than ViBe. MC-SILTP performs slightly better than SILTP. Our method achieves the highest average F1 which is 14% higher than the second best method MC-SILTP. Also, our method can achieve consistently high F1 as indicated by the lowest variance.

Figure 10 shows the visual results. In "Bootstrap," humans already exist in the initialization image sequence. ViBe produces more false-negative errors. Our method and SILTP relatively have lesser false-negative errors. Our method also has lesser false-positive errors than MC-SILTP. In "Camouflage," the difficulty is that the monitor and the clothing have a similar color. Therefore, ViBe, SILTP, and MC-SILTP produce many false-negative errors. With probabilistic refinement, our method can drastically reduce false-negative error. In "ForegroundAperture," the human remains stationary and stooped over the desk for some time. Features of the human are included in the background model. When the human rises, all methods produce false-negative errors. In "LightSwitch," ViBe cannot adapt to the sudden change of light. Other methods can quickly respond. In "TimeOfDay," the room is very dark in the beginning. The light is turned on gradually, and a human enters the room. SILTP and MC-SILTP cannot adapt to the change and result in a large amount of false-positive errors. ViBe performs better, but the segmented human is small. Benefit by the P-LTP feature, our method can segment a larger human with a minimal false-positive error. In "WavingTrees," ViBe and SILTP produce many false-positive errors in the trees behind the human. MC-SILTP still produces a moderate amount of false-positive error. As the P-LTP feature is rotational invariant, our method is quite effective in identifying the waving trees as background. In summary, our method can achieve a consistent and accurate performance under various kinds of complication in the background scene.

Table 2 F1 results on the Star dataset

Sequence	Our method	ViBe	SILTP	MC-SILTP	MoG	LBP-B	LBP-P
AirportHall	*0.728*	0.496	0.681	0.659	0.579	0.477	0.503
Bootstrap	*0.790*	0.514	0.754	0.649	0.541	0.528	0.520
Curtain	0.885	0.775	*0.912*	0.707	0.505	0.661	0.714
Escalator	0.569	0.445	*0.639*	0.439	0.366	0.591	0.539
Fountain	0.809	0.425	*0.835*	0.504	0.779	0.705	0.753
ShoppingMall	*0.813*	0.522	0.796	0.513	0.670	0.547	0.629
Lobby	0.765	0.029	*0.788*	0.690	0.684	0.503	0.523
Trees	*0.727*	0.345	0.425	0.222	0.554	0.629	0.606
WaterSurface	*0.934*	0.801	0.743	0.570	0.635	0.768	0.822
Average	*0.780*	0.483	0.730	0.550	0.590	0.601	0.623
Variance	0.011	0.052	0.019	0.024	0.014	*0.010*	0.013

The best results are shown in italics

7.2 Star dataset

We selected ViBe [10], SILTP [21], MC-SILTP [23], parametric method MoG [5], texture-based methods blockwise LBP (LBP-B) [34], and pixelwise LBP (LBP-P) [20] for comparison. Table 2 shows the F1 results. The numeric results of SILTP, MoG, LBP-B, and LBP-P are from [21]. Our method can achieve the highest F1 on five videos and second best on three videos. Overall, our method achieves the highest average F1 than all comparing methods which is 5% higher than the second best method SILTP. Our method has the second lowest variance which is very near to LBP-B.

Figure 11 shows the visual results of our method, ViBe, SILTP, and MC-SILTP. Some videos contain busy human flows (AirportHall, Bootstrap, Escalator, ShoppingMall). "Curtain" has a slowly moving curtain in the background. "Fountain" and "WaterSurface" contain moving water. In "Lobby," the light is dimmed and turned on later. "Trees" has waving trees and banner in the background. In "AirportHall," ViBe and MC-SILTP produce more false-negative errors. The results of our method and SILTP are close, only that the latter method has more false-positive and negative errors. A similar comparison can be observed in "Bootstrap." In "Curtain,"

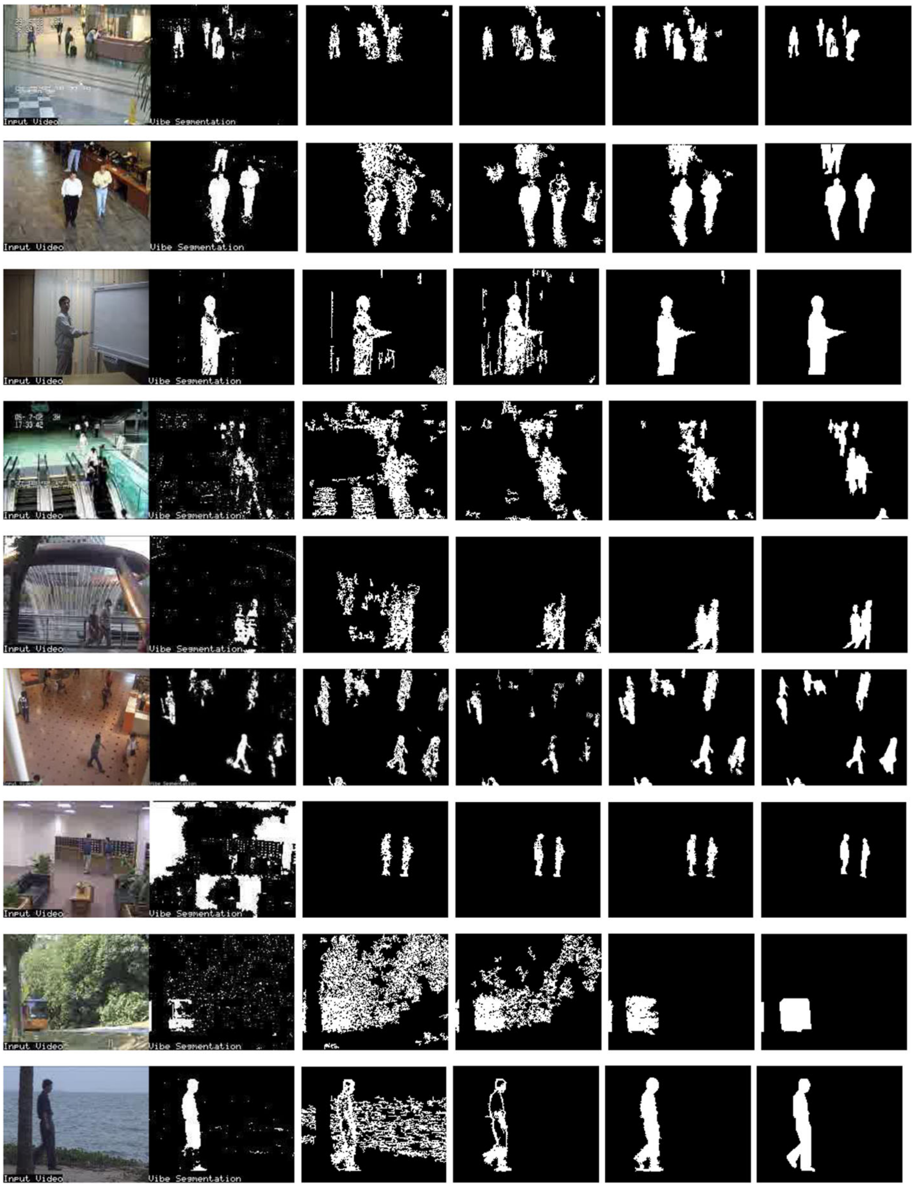

Fig. 11 Background subtraction results on the Star dataset—original image frames (first column), results obtained by ViBe (second column), results obtained by SILTP (third column), results obtained by MC-SILTP (fourth column), results obtained by our method (fifth column), and ground truths (last column)

Table 3 F1 results on the ChangeDetection.net dataset dynamic background image sequences

Sequence	Our method	ViBe	SILTP	MC-SILTP
boats	*0.566*	0.178	0.015	0.127
canoe	*0.850*	0.783	0.107	0.695
fountain01	*0.196*	0.061	0.008	0.050
fountain02	*0.632*	0.563	0.024	0.098
overpass	*0.704*	0.685	0.066	0.209
Average	*0.590*	0.454	0.044	0.236
Variance	0.059	0.101	*0.002*	0.069

The best results are shown in italics

other methods produce many false-negative errors. MC-SILTP also produces many false-positive errors. "Escalator" is a difficult video. ViBe, MC-SILTP, and our method can model the background. In "Fountain," our method takes advantage of the rotational invariant P-LTP feature and accurately identifying the fountain as background. Our method can segment all humans in "ShoppingMall" with low false-positive and negative errors. In "Lobby," ViBe cannot adapt to the change of light while the results of our method are close to SILTP and MC-SILTP. ViBe, SILTP, and MC-SILTP produce a large amount of false-positive errors in "Trees." Our method has no problem in tackling the waving trees as well as the window of the bus. Similarly, the P-LTP feature can model the moving water in "WaterSurface." In ViBe, the legs are almost missing completely, and MC-SILTP also produces many false-negative errors. Our method can segment the whole figure of the human. In this experiment, it can be seen that our method can achieve a stable performance in all videos.

7.3 ChangeDetection.net dataset

We compared our method with ViBe, SILTP, and MC-SILTP on image sequences of the dynamic background category. Due to larger frame size, background scene changes are more vigorous than the videos of previous datasets. The testing image sequences contain complex backgrounds such as swaying trees, camouflage, fountain, and water surface. Table 3 shows the F1 results. Fountain01 is a very difficult image sequence with small foreground object and large regions of changing background. All methods achieve very low F1 results. Overall, our method can achieve the highest F1 on all image sequences, and the average F1 is 14% higher than the second best method ViBe.

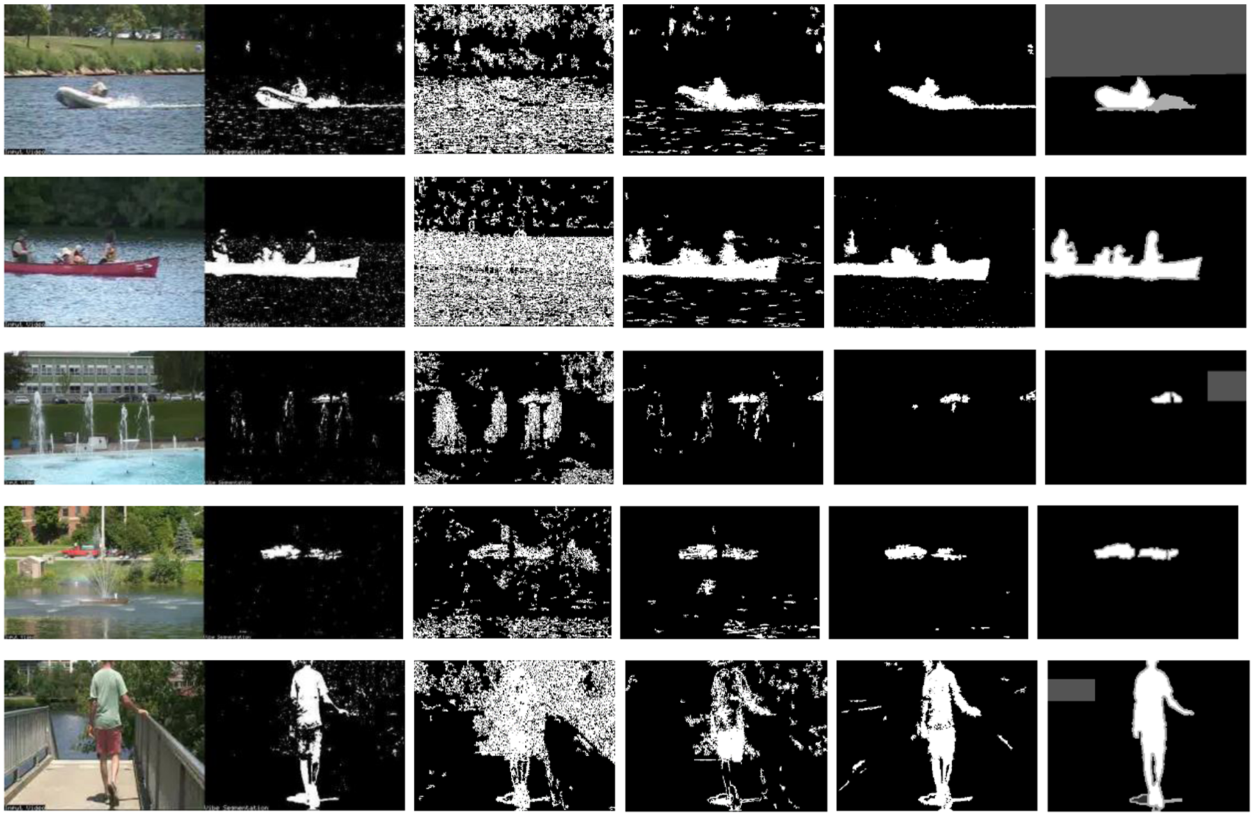

Fig. 12 Background subtraction results on the ChangeDetection.net dataset dynamic background image sequences—original image frames (first column), results obtained by ViBe (second column), results obtained by SILTP (third column), results obtained by MC-SILTP (fourth column), results obtained by our method (fifth column), and ground truths (last column)

Figure 12 shows the visual results of our method, ViBe, SILTP, and MC-SILTP. "Boats" has water surface, people, and cars in the background. "Canoe" has water surface and trees in the background. SILTP cannot model the background well. MC-SILTP performs much better than SILTP, but like ViBe, produce many false-positive errors in the water surface. Our method can effectively model the whole background and segment the boat and canoe. "Fountain01" and "fountain02" have flowing water in front and moving cars behind the fountain. Again, SILTP cannot model the fountain well. MC-SILTP still produces many false-positive errors in the fountain. ViBe results in a smaller foreground region. Our method can model the flowing water and segment the foreground in sufficient large size and good shape. "Overpass" contains a human walk along the bridge with swaying trees in the background. The legs and bridge have a similar color. SILTP cannot model the trees well. MC-SILTP and ViBe produce many holes (false-negative errors). Our method can effectively model the swaying trees and segment the human.

8 Conclusions

We propose a method for the segmentation of moving objects in the video. The background model is represented by perception-based local ternary pattern and photometric features. The local ternary pattern is robust to random noise and invariant to scale transform. The feature derived from the local pattern is also rotational invariant. Each pixel of the image frame is classified as either background or object by matching the image feature with the background model. The segmented object can be refined by taking into account the spatial consistency of the image feature. The background model is updated periodically along the image sequence for adapting to the changes of the scene. We devise two mechanisms—short-term update to respond to a sudden global change in the scene and long-term update for incorporating new background features. We compare our method with various background subtraction algorithms. Testing is performed on three common video datasets, containing many types of complication. The quantitative and visual results show the consistency and accuracy of our method over others.

Abbreviations

BG: Background; CI: Confidence interval; DT: Dynamic texture; FG: Foreground; LBP: Local binary pattern; LBP-B: Blockwise LBP; LBP-P: Pixelwise LBP; LBSP: Local binary similarity pattern; MC-SILTP: Multi-channel SILTP; MoG: Mixture of Gaussian; P-LTP: Perception-based local ternary pattern; PSNR: Peak signal-to-noise ratio; SILTP: Scale invariant local ternary pattern; ViBe: Visual background extractor

Availability of data and materials

The video datasets used are publicly available and cited in the list of references.

Authors' contributions

The research was performed solely by the author. The author read and approved the final manuscript.

Authors' information

KL Chan received the M.Sc. Degree in Electronics from the University of Wales Institute of Science and Technology, UK. He received the Ph.D. degree from the University of Wales College of Medicine, UK. He is currently an Assistant Professor of the Department of Electronic Engineering, the City University of Hong Kong. His research interests include image processing and computer vision.

Competing interests

The author declares that he has no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 26 January 2018 Accepted: 16 July 2018
Published online: 27 July 2018

References

1. SY Elhabian, KM El-Sayed, SH Ahmed, Moving object detection in spatial domain using background removal techniques – state-of-art. Recent Pat. Comput. Sci. 1, 32–54 (2008).
2. A Sobral, A Vacavant, A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput. Vis. Image Underst. 122, 4–21 (2014).
3. T Bouwmans, Recent advanced statistical background modeling for foreground detection: a systematic survey. Recent Pat. Comput. Sci. 4(3), 147–176 (2011).
4. CR Wren, A Azarbayejani, T Darrell, A Pentland, Pfinder: real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. 19(7), 780–785 (1997).
5. C Stauffer, WEL Grimson, Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 747–757 (2000).
6. Z Zivkovic, in *Proceedings of International Conference on Pattern Recognition*. Improved adaptive Gaussian mixture model for background subtraction (2004), pp. 28–31.
7. T Bouwmans, F El Baf, B Vachon, Background modeling using mixture of Gaussians for foreground detection – a survey. Recent Pat. Comput. Sci. 1, 219–237 (2008).
8. A Elgammal, R Duraiswami, D Harwood, LS Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE 90(7), 1151–1163 (2002).
9. K Kim, TH Chalidabhongse, D Harwood, LS Davis, Real-time foreground-background segmentation using codebook model. Real-Time Imaging 11, 172–185 (2005).
10. O Barnich, M Van Droogenbroeck, in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*. ViBe: a powerful random technique to estimate the background in video sequences (2009), pp. 945–948.
11. M Hofmann, P Tiefenbacher, G Rigoll, in *Proceedings of IEEE Workshop on Change Detection at IEEE Conference on Computer Vision and Pattern Recognition*. Background segmentation with feedback: the pixel-based adaptive segmenter (2012), pp. 38–43.
12. M Van Droogenbroeck, O Paquot, in *Proceedings of IEEE Workshop on Change Detection at IEEE Conference on Computer Vision and Pattern Recognition*. Background subtraction: experiments and improvements for ViBe (2012), pp. 32–37.
13. TSF Haines, T Xiang, Background subtraction with Dirichlet process mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 36(4), 670–683 (2014).
14. Z Gao, L-F Cheong, Y-X Wang, Block-sparse RPCA for salient motion detection. IEEE Trans. Pattern Anal. Mach. Intell. 36(10), 1975–1987 (2014).
15. X Liu, G Zhao, J Yao, C Qi, Background subtraction based on low-rank and structured sparse decomposition. IEEE Trans. Image Process. 24(8), 2502–2514 (2015).
16. A Monnet, A Mittal, N Paragios, V Ramesh, in *Proceedings of IEEE International Conference on Computer Vision*. Background modelling and subtraction of dynamic scenes (2003), pp. 1305–1312.

17. V Mahadevan, N Vasconcelos, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Background subtraction in highly dynamic scenes (2008), pp. 1–6.
18. Y Sheikh, M Shah, in *Proceedings of IEEE Conference On Computer Vision and Pattern Recognition*. Bayesian object detection in dynamic scenes, vol 1 (2005), pp. 74–79.
19. S Zhang, H Yao, S Liu, in *Proceedings of International Workshop on Visual Surveillance*. Dynamic background subtraction based on local dependency histogram (2008).
20. M Heikkilä, M Pietikäinen, A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Mach. Intell. 28(4), 657–662 (2006).
21. S Liao, G Zhao, V Kellokumpu, M Pietikäinen, SZ Li, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes (2010), pp. 1301–1306.
22. P-L St-Charles, G-A Bilodeau, R Bergevin, SuBSENSE: a universal change detection method with local adaptive sensitivity. IEEE Trans. Image Process. 24(1), 359–373 (2015).
23. F Ma, N Sang, in *Proceedings of Asian Conference on Computer Vision*. Background subtraction based on multi-channel SILTP (2012), pp. 73–84.
24. L Lin, Y Xu, X Liang, J Lai, Complex background subtraction by pursuing dynamic spatio-temporal models. IEEE Trans. Image Process. 23(7), 3191–3202 (2014).
25. C Spampinato, S Palazzo, I Kavasidis, A texton-based kernel density estimation approach for background modeling under extreme conditions. Comput. Vis. Image Underst. **122**, 74–83 (2014).
26. SW Kim, K Yun, KM Yi, SJ Kim, JY Choi, Detection of moving objects with a moving camera using non-panoramic background model. Mach. Vis. Appl. 24, 1015–1028 (2013).
27. H-L Eng, J Wang, AHK Siew Wah, W-Y Yau, Robust human detection within a highly dynamic aquatic environment in real time. IEEE Trans. Image Process. 15(6), 1583–1600 (2006).
28. KL Chan, in *Proceedings of the 15th IAPR International Conference on Machine Vision Applications*. Saliency/non-saliency segregation in video sequences using perception-based local ternary pattern features (2017), pp. 480–483.
29. RC Gonzalez, RE Woods, *Digital Image Processing* (Pearson/Prentice Hall, Upper Saddle River, 2010).
30. M Haque, M Murshed, Perception-inspired background subtraction. IEEE Trans. Circuits Syst. Video Technol. 23(12), 2127–2140 (2013).
31. K Toyama, J Krumm, B Brumitt, B Meyers, in *Proceedings of International Conference on Computer Vision*. Wallflower: principles and practice of background maintenance (1999), pp. 255–261.
32. L Li, W Huang, IY-H Gu, Q Tian, Statistical modelling of complex backgrounds for foreground object detection. IEEE Trans. Image Process. 13(11), 1459–1472 (2004).
33. N Goyette, P-M Jodoin, F Porikli, J Konrad, P Ishwar, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Changedetection.net: a new change detection benchmark dataset (2012), pp. 16–21.
34. M Heikkilä, M Pietikäinen, J Heikkilä, in *Proceedings of British Machine Vision Conference*. A texture-based method for detecting moving objects (2004), pp. 187–196.