Future-data driven modeling of complex backgrounds using mixture of Gaussians

Xin Liu *, Chun Qi

School of Electronics and Information Engineering, Xi’an Jiaotong University, Xi’an, China

1. Introduction

The segmentation of moving foreground objects from video stream is the fundamental step in many computer vision applications, such as intelligent visual surveillance [1,2], human–machine interaction [3,4], and content based video coding [5]. Background subtraction is generally regarded as an effective method for extracting foreground objects [6–8]. The performance of background subtraction mainly depends on the algorithm used for modeling background. However, the background is a complex environment usually includes distracting motions which make the task more challenging, such as waving tree branches and rippling water. An adaptive background modeling algorithm should also detect shadows cast by the moving objects, and handle various changes due to the situation where new objects are introduced to the background or old ones removed from it. Furthermore, an ideal background modeling algorithm should be able to tolerate sudden background variations like the changing weather conditions or the turn on/off lights, without losing sensitivity to detect real foreground objects.

Many background modeling algorithms have been proposed (surveys [8,9]). A very popular approach is to model each pixel in a video frame with mixture of Gaussians, instead of using the exact Expectation Maximum (EM) algorithm by Friedman and Russell et al. [10], an online K-means approximation proposed by Stauffer and Grimson [11], which has become the standard formulation for the MoG approach in this field. In their approach, an online learning was used to train background model. For every image pixel, regardless of its intensity being changed or not, the learning rate of updating is controlled by a global, static parameter α that ranges between 0 and 1. Because only one new sample is observed, the equality of these two sets is tested by the hypothesis testing method. Next, a two-layer LBP-based method is proposed for foreground classification. Finally, the global and static learning rates are replaced by the adaptive learning rates for image pixels with distinct properties for each frame. By means of the proposed learning strategy, a novel background modeling for detecting foreground objects from complex environments is established. We compare our procedure against the state-of-the-art alternatives, the experimental results show that the performance of learning speed and accuracy obtained by proposed learning rate control strategy is better than existing MoG approaches.

Article info

Article history:
Received 23 August 2012
Received in revised form
12 February 2013
Accepted 31 March 2013
Communicated by M. Wang
Available online 8 May 2013

Keywords:
Background modeling
Background subtraction
Gaussian mixture models
Learning-rate
Foreground detection

Abstract

Mixture of Gaussians (MoG) is well-known for effectively in sustaining background variations, which has been widely adopted for background subtraction. However, in complex backgrounds, MoG often traps in keeping balance between model convergence speed and its stability. The main difficulty is the selection of learning rates. In this paper, an effective learning strategy is proposed to provide better regularization of background adaptation for MoG. First, the video-data is splitted into the future-data and history-data, then a set of background distributions (MoG) is computed for each case. To distinguish between dynamic and static background, the equality of these two sets is tested by the hypothesis testing method. Next, a two-layer LBP-based method is proposed for foreground classification. Finally, the global and static learning rates are replaced by the adaptive learning rates for image pixels with distinct properties for each frame. By means of the proposed learning strategy, a novel background modeling for detecting foreground objects from complex environments is established. We compare our procedure against the state-of-the-art alternatives, the experimental results show that the performance of learning speed and accuracy obtained by proposed learning rate control strategy is better than existing MoG approaches.

© 2013 Elsevier B.V. All rights reserved.
rate. However, the background features of initial phase were limited after all, this two-phase method would not improve adaption for modelling at later phase. In [15], Harville discussed some trade-off encountered by MoG and introduced a framework for guiding MoG evolution with feedback from high-level modules. In [12], not only the learning parameters but also the number of models of the mixture was adapted for each pixel. Tian et al. [16] detected the static foreground regions that were wrongly modeled as the background and proposed a weight exchange scheme to avoid a fragment problem in MoG. Tuzel et al. [17] proposed to estimate the probability distribution of mean and covariance of each Gaussian using recursive Bayesian learning. In [18], Lee et al. proposed an effective learning algorithm that improved convergence rate of background model estimation without obvious side-effects on system stability. Yang et al. [19] proposed a new rate control method to detect over-quick lighting change and adjust learning rates. However, a quick variety of lighting occurred in very short time. Furthermore, Lin et al. adopted frame difference to detect whether the background will change in future. By future-data is proposed, and a hypothesis testing method is used for guiding MoG evolution with feedback from high-level modules.

In [12], not only the learning parameters but also the number of models of the mixture was adapted for each pixel. In [15], Harville discussed some trade-off encountered by MoG and introduced a framework for guiding MoG evolution with feedback from high-level modules. In [12], not only the learning parameters but also the number of models of the mixture was adapted for each pixel. Tian et al. [16] detected the static foreground regions that were wrongly modeled as the background and proposed a weight exchange scheme to avoid a fragment problem in MoG. Tuzel et al. [17] proposed to estimate the probability distribution of mean and covariance of each Gaussian using recursive Bayesian learning. In [18], Lee et al. proposed an effective learning algorithm that improved convergence rate of background model estimation without obvious side-effects on system stability. Yang et al. [19] proposed a new rate control method to detect over-quick lighting change and adjust learning rates. However, a quick variety of lighting occurred in very short time. Furthermore, Lin et al. adopted frame difference to detect whether the background will change in future. By future-data is proposed, and a hypothesis testing method is used for guiding MoG evolution with feedback from high-level modules.

In this section, we will present the proposed background modeling in four parts. First, the work of MoG and the convergence of the learning process are analyzed. We highlight the importance of learning rate control for MoG and elaborate its relationship with pixel classification. Second, a fast and efficient convergence method in background modeling based on MoG for future-data is proposed, and a hypothesis testing method is used to test the equality of history Gaussian models and future ones, and to decide whether the background will change in future. By this decision, a classification for pixel types is proposed, the whole background pixels are divided into dynamic and static types. Third, a two-layer texture method based on LBP for foreground classification is proposed, the true foreground is distinguished from the false foreground. Fourth, a learning strategy that controls learning rate for MoG is detailed. Under this control, different learning rates can be applied to different pixel types which labeled already. The sudden illumination changes are detected by testing if the background likelihood ratio is smaller than a threshold. If a change is detected, the whole background is updated with a larger rate to respond to the once-off changes.

2. Future-data driven background modeling

In this section, we will present the proposed background modeling in four parts. First, the work of MoG and the convergence of the learning process are analyzed. We highlight the importance of learning rate control for MoG and elaborate its relationship with pixel classification. Second, a fast and efficient convergence method in background modeling based on MoG for future-data is proposed, and a hypothesis testing method is used to test the equality of history Gaussian models and future ones, and to decide whether the background will change in future. By this decision, a classification for pixel types is proposed, the whole background pixels are divided into dynamic and static types. Third, a two-layer texture method based on LBP for foreground classification is proposed, the true foreground is distinguished from the false foreground. Fourth, a learning strategy that controls learning rate for MoG is detailed. Under this control, different learning rates can be applied to different pixel types which labeled already. The sudden illumination changes are detected by testing if the background likelihood ratio is smaller than a threshold. If a change is detected, the whole background is updated with a larger rate to respond to the once-off changes.

2.1. Mixture of Gaussians approach

For a pixel \(p\), at position \((x, y)\), at time \(t\), what is known about its history \(\{X_1, \ldots, X_{t-1}\} = \{(x, y, t) : 1s \leq t \leq (t-1)\}\) is modeled by a mixture of \(K\) Gaussian distributions, where \(k(\cdot)\) is the pixel's intensity value.

The probability of observing the current pixel value is

\[
p(X_t) = \sum_{i=1}^{K} \omega_i p(X_t, \mu_{t,i}, \sigma^2_{t,i})
\]

where \(p(\cdot)\) symbolizes a Gaussian probability density function, and \(\mu\) is the mean, \(\sigma^2\) is the covariance, are the Gaussian parameters of the \(i\)th distribution, where \(\omega_{t,i}\) is the respective weight

\[
\eta(X, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(X-\mu)^2}{2\sigma^2}\right)
\]

The \(K\) Gaussians are ordered based on \(\omega_{t,i}/\sigma_i\), the first \(B\) Gaussians are used as models of the background, estimated as

\[
B_t = \arg \min_b \left( \sum_{i=1}^{b} \omega_{t,i} > T \right)
\]

\[
\sum_{i=1}^{K} \omega_{t,i} = 1
\]

where \(T\) is a measure of the minimum portion of the data that should be accounted for by the background. The pixel value \(X_t\) lying within 2.5 standard deviations of a distribution is defined as a match. If a match is found with one of the first \(B\) Gaussians, the pixel is classified as background. If none of the first \(B\) distributions be matched, the pixel is classified as foreground, namely, foreground detection is performed by

\[
\forall i \in \{1, \ldots, B\} : |X_t - \mu_{t,i}| > 2.5\sigma_{t,i}
\]

To maintain and update mixture Gaussian models, the weight of the each model is to be updated by the equation

\[
a_{t+1,i} = (1-\alpha) a_{t,i} + \alpha p(X_t | \mu_{t,i})
\]

If \(i\) is the matched Gaussian model \(p(X_t | \mu_{t,i}) = 1\), otherwise it equals 0. The first Gaussian model (noted as \(ma\)) that matches the current pixel value will be updated by

\[
\mu_{t+1,ma} = (1-\beta) \mu_{t,ma} + \beta X_t
\]

\[
\sigma^2_{t+1,ma} = (1-\beta) \sigma^2_{t,ma} + \beta (X_t - \mu_{t,ma})^2
\]

\[
\beta = \alpha \eta(X_t | \mu_{t,ma}, \sigma^2_{t,ma})
\]

If none of the \(K\) distributions match that pixel value, the least probable model is replaced by a distribution with the current value as its mean \(\mu\), an initially high variance \(\sigma^2\), and a low weight parameter \(\alpha\).

The \(\alpha\) defines the rate of updating that controls how fast the model converges to a new one. In the conventional MoG, for every pixel, a very small constant is commonly used for background adaptation. Unfortunately, this setting leads to slow convergence when background needs to adapt to a new cluster. For example, if a new background object comes into a scene, suppose at present the weights sum of the first \(B\) models equals unity. It will take \(N_t\) frames until the genuine object can be considered as a background. From (3) and (5) we can obtain that if the new object model's weight becomes larger than 1–\(T\), namely, until the weights sum of the old \(B\) models becomes smaller than \(T\), the genuine object can be considered to be part of the background. From (6) we can conclude

\[
(1-\alpha)^{N_t} < T
\]

So, the object should be static for at least \(N_t = \log T/\log(1-\alpha)\) frames. As noted in [25], using \(\alpha = 0.0025\) and \(T=0.8\) in MoG, if we assume that the new object will presented in every frame ideally, it would take 89 frames for the component to be included as part of the background. If the frame-rate being 20 fps, the system will take at least 4.5 s to respond to background changes. The situation...
can be worse in busy environments where a clean background is rare.

By setting $\alpha$ with a large value would improve the convergence speed. However, the system will become easier to be perturbed by noise and foreground objects. Setting global learning rate leaves no space for adjusting the updating speeds. Some authors [15,16,24] proposed learning scheme by high-level feedback, but most of them focused on foreground pixel classification and relevant learning rate control. However, the background can be described as consisting of static and dynamic pixels. The static pixels belong to the stationary objects, such as walls and furniture, and the dynamic pixels are associated with non-stationary objects, such as waving trees, fountain spurt. So, larger learning rates should be given to the dynamic pixels for background updating while to give smaller learning rates for static pixels for background stability. We assign individual learning rates for each pixel respect to space and time, and adapt them over time, so, higher flexibility in controlling background adaptation can be handled. In summation, we highlight the importance of learning rate control based on different pixel properties.

In the conventional MoG and its extensions or improvements, to the best of our knowledge, all of the literature just take use of a single coming new sample to make background updating strategy. Only one new sample, match or not, background or foreground, and the dynamic pixels are associated with non-stationary objects, such as waving trees, fountain spurt. So, larger learning rates should be given to the dynamic pixels for background updating while to give smaller learning rates for static pixels for background stability. We assign individual learning rates for each pixel respect to space and time, and adapt them over time, so, higher flexibility in controlling background adaptation can be handled. In summation, we highlight the importance of learning rate control based on different pixel properties.

In the conventional MoG and its extensions or improvements, to the best of our knowledge, all of the literature just take use of a single coming new sample to make background updating strategy. Only one new sample, match or not, background or foreground, makes the task of background learning cannot be controlled according to more delicate pixel classification. Different to the previous works, more new observations called ‘future-data’ are applied to make decision for learning at the expenses of delaying a few frames. For a pixel, the existing MoG data means the ‘history’ while the deliberate delayed frames indicate the ‘future’, how to make learning strategy by using the history-data and future-data is our main goal. Motivated by this, a novel learning strategy for MoG to improve the convergence rate without compromising model stability is proposed.

### 2.2. Future-data MoG and background classification by hypothesis testing

In this paper, we delay $N$ frames to obtain more new observations as shown in Fig. 1. Now, the existing history background distributions of pre-frames are the ‘history-data’, and of the post-frames may be regarded as ‘future-data’. So a new sample set obtained by post-frame could be regarded as a random experiment and a hypothesis testing method can be used to determine whether the pixel’s background would change in the ‘future’. If the number of $N$ frames to be considered for building the ‘future-data’ is large, the method will result in several seconds of delay which may restrict the method to be used for time-critical purposes such as unusual event detection (fire, smoke, and gunfire, etc.). However, it is difficult to establish the background model well using a smaller number of samples.

In this paper, we set $N=20$. Now, no more than 1 s later than real time, it could be accepted in many video applications. Since there are no enough samples in short sequences, we firstly use the same assumption as the literatures [26,27], which is that neighboring pixels share a similar temporal distribution. This justifies the fact that we populate the new sample set with values found in the spatial neighborhood of each pixel. More precisely, we obtain samples in the $M$-connected neighborhood of each pixel. So, the size of the new sample set as future-data is $(M+1) \times N$. From our experiments, selecting samples in the four-connected neighborhood of each pixel have proved to be satisfactory for our method. Secondly, on account of the conventional MoG’s convergence speed is too slow to model a small sample set, especially for dynamic background when new functions estimate parameters. As more samples were included in its model, the equation is similar to (4). To improve convergence speed for the changes in the background. Inspired by [18], the learning rate in (9) for matched Gaussian model $ma$

$$
\beta' = \frac{1}{\frac{1}{(M+1)N} + \frac{1}{(M+1)N}}
$$

From the above learning rate update equation, at the initial stage of a Gaussian model when only a few samples have been observed, $\beta'=1/c_{n,k}$, the parameter updates closely approximate the expected value computed by sufficient statistics. As more samples were included in its model, the equation is similar to the typical recursive learning. Our new method considers each Gaussian separately, result in quicker convergence in Gaussian parameter learning for ‘future-data’ background.

By above method, the distribution of new observations of each pixel is characterized by a mixture of Gaussians. We also choose first $B_0$ distributions with (3) are used as models of the background of the future. Then, for a pixel, the ordered first $B_{t-1}$ and $B_{t}$ distributions are the background models of history and future. If $B_{t-1} \neq B_{t}$, it means that the number of the background model has changed. So, the background should be changed in the future. By equaled, we are doing hypothesis testing for the history and future models one-to-one which determine whether the background would change in the future. In our papers, the two-sample $t$-test [28] is used to test the equality of these two population means, defined as

$$
H_0 : \mu_{t-1,j} = \mu_{t,j} \quad H_1 : \mu_{t-1,j} \neq \mu_{t,j}
$$

![Fig. 1. Illustration for how to get “future-data”. (a) Original. (b) Delay $N$ frames.](image-url)
Illustration of Block-LBP. First row: Block-LBP encodings. In each sub-region, average gray-values of image intensity is computed. These average gray-values are then thresholded by that of the center block. Block-LBP is then obtained. Second row: encodings with noises. The circled pixels are changed with noises, and no encodings are affected by those changes.

Local binary pattern (LBP) [30] feature is invariant to local illumination variations such as cast shadow because LBP is obtained by comparing local pixels values. The original LBP operator labels the pixels of an image by thresholding the $3 \times 3$-neighborhood of each pixel with the center value and considering the result as a binary string. It is a powerful mean of texture descriptors. The encoding is shown in the first column of Fig. 2. However, the LBP operator is not robust to local image noises when neighboring pixels are similar, as illustrated in the second column of Fig. 2 for an example.

In this work, we propose an extension to the basic LBP, called Block LBP, to overcome the limitations of LBP. In Block LBP, the comparison operator between single pixels in LBP is simply replaced with comparison between average gray-values of sub-regions, the process be illustrated in the first row of Fig. 3 for an example. Using average values over the regions, the large scale filters reduce noise, and make the representation more robust than LBP, as illustrated in the second row of Fig. 3 for an example. Note that the scalar values of averages over blocks can be computed very efficiently [31] from the summed-area table [32]. For this reason, Block-LBP feature extraction can also be very fast, it only incurs a little more cost than the original LBP. A comparison experiment is shown in Fig. 4, where the selected background pixel is similar to its neighborhood, and the statistics of the pixel processes (300 frames) with LBP and Block-LBP descriptors are displayed. It can be seen that the LBP is more variable than Block-LBP, and the latter is almost invariant among all the 300 frames counted.

In Fig. 5, two features were compared for the background region with shadows. As can be seen from the histograms, for background with and without shadows, the Block-LBP operator performs perfectly, almost not influenced by shadows as only a few patterns being different between the two marked image regions, while LBP histogram shows larger difference.
We adopt a base framework described in [30] for false foreground detection, using not only original LBP but also Block-LBP descriptor. In each frame, for foreground pixels detected by MoG, the true foreground (defined $F_{t,f} = 1$) is distinguished from the false foreground ($F_{t,f} = 0$) by above two-layer LBP foreground detection. Rather than digging into the details of framework in [30], we place the focus on the learning rate control in the following discussions.

2.4. Learning rate control

In this paper, we adopt different learning rate settings for four pixel types of real foreground, false foreground, hot background and cold background respectively. The four types can be easily discriminated. Thus, a classification for pixel $p(x_0,y_0)$ at time $t'$ can be defined as

$$C_t = \begin{cases} 
0 & \text{if } F_{t,b} = 0 \text{ and } (F_{t,f} \neq 0 \text{ or } 1) \text{ (cold background)} \\
1 & \text{if } F_{t,b} = 1 \text{ and } (F_{t,f} = 0 \text{ or } 1) \text{ (hot background)} \\
2 & \text{if } F_{t,f} = 0 \text{ (false foreground)} \\
3 & \text{if } F_{t,f} = 1 \text{ (real foreground)}
\end{cases}$$

For real foreground, we set $\alpha_r = 0$ to suppress the learning of real foreground objects into the background. For the application of abandoned or missing object detection, the real stationary foreground object will not be merged into the background quickly. Regarding the case of false foreground, we favor faster learning of false foreground into background. So the learning rate used in (7) and (8) is defined as

$$\beta = \alpha_l q(X_{t} - \mu_{t-1,b}, \sigma^2_{t-1,b})$$

and

$$b = \min_j |X_{t} - \mu_{t-1,j}|$$

where $b$ is the best matched model, and $q(\cdot)$ is used to estimate the similarity between the false foreground intensity value and the Gaussian model. The corresponding learning rate is then set to the similarity measure multiplied by a large value of learning rate $\alpha_f$ for a fast false foreground learning.

For a pixel of hot background, if set a very small learning rate like the conventional MoG approaches, it will result in a very sensitive system to dynamic background. In contrast, by setting $\alpha_h$ to a large value, which results in a quick updating for Gaussian models, such as waving trees and flickering surface of water, system will be more capable of tolerating background variations. For a pixel of cold background, a small $\alpha_c$ is preferred for background model stability.

With the above notations, the learning rate $\alpha, \beta$ at time $t'$ for pixel $p(x_0,y_0)$ can now be specified by

$$\alpha, \beta = \begin{cases} 
\alpha_c, \alpha_f q(X_{t}, \mu_{t-1,ma}, \sigma^2_{t-1,ma}) & \text{if } C_t = 0 \\
\alpha_h, \alpha_h q(X_{t}, \mu_{t-1,ma}, \sigma^2_{t-1,ma}) & \text{if } C_t = 1 \\
\alpha_f, \alpha_f q(X_{t}, \mu_{t-1,b}, \sigma^2_{t-1,b}) & \text{if } C_t = 2 \\
\alpha_c, 0 & \text{if } C_t = 3
\end{cases}$$

Background modeling often encounters challenges from once-off [34] background changes, like sudden changes in illumination and other scene parameters alter the appearance of the background. It is hard to be caught and adapted in a reasonable time by the conventional MoG with small learning rate. Thanks to future-data, if

$$\frac{N_c + N_h}{H \cdot W} < T_o$$

where $N_c$ and $N_h$ are the numbers of cold and hot background pixels, and $H, W$ denote the size of the frame, $T_o$ is a given threshold. Thus, the once-off background changes can be obtained if the background likelihood ratio less than $T_o$. In this case, we set whole pixels in the frame with a global, large value learning rate $\alpha_o$ to respond to the once-off change.

Fig. 4. Comparison of LBP and Block-LBP features for two background pixels on real video. (a) Shows a frame from PETS 2006 dataset [33], with two marked pixels. (b) and (c) are the histograms of two pixels from frames over time, with LBP and Block-LBP descriptors respectively (300 frames counted).

Fig. 5. Comparison of LBP and Block-LBP features with shadow. (a) and (b) are two frames from the “Shopping Mall” dataset [34], with two 10 $\times$ 10 regions drawn. Regions contain the same background with and without shadows. (c) LBP histogram of two regions. (d) Block-LBP histograms of two regions.
3. Experiment

3.1. Effects of separate steps and learning rate tuning

In order to better understand the performance of the proposed algorithm, we analyzed the effects of separate steps one by one. First, to evaluate the effectiveness of background classification and its learning rate control, we just discriminated the pixels into three pixel types: hot background, cold background, and foreground (FG). The first sequence we use here is the “Campus” [34], with two sources of dynamic motion: plant waving because of the wind, and their shadows on the ground surface. The “Campus” is a commonly used foreground segmentation test sequence, especially for evaluating the effectiveness of dynamic background modeling techniques. In Fig. 6, comparisons among original MoG [11] and the proposed method with the same number of Gaussian models $K = 5$ are presented. For MoG algorithm, we experimented with different learning rates with $\alpha$ between 0.001 and 0.1. For the proposed algorithm, three pixel types with various different learning parameters tuning were compared, with $\alpha_h$ of hot background between 0.01 and 0.1, $\alpha_c$ of cold background between 0.001 and 0.01, and $\alpha_{FG}$ of foreground between 0.001 and 0.01. At the beginning of the sequence, the scene is empty from frames 1050th to 1200th. As shown in Fig. 6(c), false positives (FP), which is the number of background pixels incorrectly classified as foreground, caused by waving trees were observed. By imposing background classification and its learning rate control, the proposed method handled this dynamic background immediately (after frame 1060th almost no false positives). We observed that the problem of false positives cannot be alleviated obviously by giving a higher learning rate ($\alpha = 0.05$) for original MoG [11], while the proposed method with two kinds of pixels learning rate setting ($\alpha_h = 0.025$, others = 0.005) can yield superior performance than the original MoG. On the other hand, hot background pixels with higher learning rate ($\alpha_h = 0.06$) did not perform better than a smaller setting ($\alpha_h = 0.05$ or 0.04). This is not surprising, because the distributions become easier perturbed by new matched samples, and the system becomes very sensitive to noise and retains too short history to be available in practice. As shown in Fig. 6(e) and (f), true positives (the total number of correctly

![Fig. 6. True and false positives comparisons of sequence “Campus”. (a) Frames with empty scene. (b) Pixel classification result of three types (hot background in red, cold background in black, and foreground in white). (c) False positives of frames 1050–1200th by different learning rate settings (MoG: $\alpha$, proposed: $\alpha_h$, $\alpha_c$, $\alpha_{FG}$). (d) Frames with moving object. (e) True positives of 12 frames with moving object. (f) False positives of 12 frames with moving object. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)](image)

![Table 1](image)

Table 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\alpha_r$</th>
<th>$\alpha_f$</th>
<th>$\alpha_h$</th>
<th>$\alpha_c$</th>
<th>$\alpha_o$</th>
<th>$T_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.002</td>
<td>0.01</td>
<td>0.04</td>
<td>0.004</td>
<td>0.04</td>
<td>0.2</td>
</tr>
</tbody>
</table>

![Fig. 7. True and false positives comparisons of sequence “Shopping Mall”. (a) Original frame. (b) Pixel classification result of three types (false foreground in blue, real foreground in white, and background in black). (c) True positives of 12 frames (MoG: $\alpha$, proposed: $\alpha_r$, $\alpha_f$, $\alpha_{BG}$). (d) False positives of 12 frames. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)](image)
detected foreground pixels) and false positives are obtained for the frames which contain moving objects (person, car). The proposed method’s detection accuracy both in terms of two criteria better than original MoG [11]. It is noticeable that the proposed method with the learning rate setting \((\alpha_f = 0.04, \alpha_r = 0.004, \alpha_{BG} = 0.002)\) can provide much better performance than others (qualitative results are illustrated in Fig. 9), regardless of the scene with moving object or not. In addition to the ‘Campus’ sequence, other test sets are used to demonstrate that our proposed approach can handle complex background variations, and reported in Section 3.2.

Followed that, we focus on the foreground classification and its learning rate control. We consider the scenarios with false foreground, and the sequence use here is ‘Shopping Mall’ [34], which every frame contains people, no clean background training, and significant shadows of moving persons cast on the ground from different directions. To demonstrate the effectiveness of proposed learning strategy, we just discriminated the pixels into three pixel types: false foreground, real foreground, and background (BG). In Fig. 7, comparisons among original MoG [11] and the proposed method with the same number of Gaussian models \(K=5\) were presented (qualitative results are illustrated in Fig. 14). For the proposed method, three pixel types with various different learning parameters tuning were compared, with \(\alpha_f\) of false foreground between 0.001 and 0.1, \(\alpha_r\) of real foreground between 0.001 and 0.01, and \(\alpha_{BG}\) of background between 0.001 and 0.01. As shown in Fig. 7(c) and (d), the proposed method can yield superior performance than the original MoG [11] in terms of true and false positives. In particular, the LBP [30] is used to remove shadows to enhance foreground objects, as a post-processing filtering for original MoG. We chose the foreground results obtained by original MoG with small learning rate \((\alpha_f \approx 0.0025)\), as it can provide better performance than other MoG settings in this sequence. It can be seen that the proposed method also can provide much better performance than MoG with LBP filtering. The reason is that the false foreground detected by LBP is treated as background in the proposed method, and learnt into the background models by a higher learning rate \(\alpha_f\). So that the proposed method not only use texture (LBP) but also the intensity feature to filter shadows. It is noticeable that there will degrade the performance of proposed method by setting a small learning rate for false foreground (such as \(\alpha_f = 0.004, \alpha_r = 0.004, \alpha_{BG} = 0.004\)), of which the scores are similar to original MoG \((\alpha = 0.0025)\) with LBP filtering as shown in Fig. 7(c) and (d). And the proposed method with the learning rate setting \((\alpha_f = 0.01, \alpha_r = 0.002, \alpha_{BG} = 0.004)\) can provide much better performance than others.

3.2. Data and qualitative results

Experimental results for moving object detection using the proposed learning strategy for MoG have been tested on three video test sets with ground-truth data, include Wallflower\(^1\) [35] (seven video sequences), Li\(^2\) [34] (nine video sequences) and Sheikh\(^3\) [36] (two video sequences). In all experiments, with the results in Section 3.1, we set the parameters with the value as shown in Table 1.

Comparisons have been made with four state-of-the-art algorithms, which include three MoG methods: (1) the original MoG algorithm proposed by Stauffer et al. [11] (hereafter referred to as MoG); (2) Lee et al.’s method [18] (Lee); (3) Lin et al.’s method [24] (Lin); and a newly background subtraction approach: (4) Maddalena et al.’s SC-SOBS [23], an improvement of SOBS [22].

All of the MoG algorithms with the same number of Gaussian models \(K=5\) and same threshold \(T=0.8\) which adopted in [25]. For MoG algorithm, we experimented with various different learning parameters with \(\alpha\) between 0.0025 and 0.05, until the results are optimal over the entire sequence. Quantitative experimentation has been performed on different learning parameters for MoG and is reported subsequently. For SC-SOBS, we adopt the parameter settings which have been reported in their literature directly.\(^4\)

Due to the paper’s length, we describe 10 different sequences, in fact, all sequences in above three test sets are tested in this paper, while those 10 sequences represent typical situations critical for video surveillance systems, and present qualitative results obtained with the proposed method.

To provide a better understanding about the classification results, four colors, including red, black, white and blue, are employed to represent pixels of hot background, cold background, real foreground and false foreground respectively.

In Figs. 8 and 9, those two sequences with typical dynamic background, caused by motion of tree branches, waving flags and

---

\(^1\) http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

\(^2\) http://perception.i2r.a-star.edu.sg/

\(^3\) http://www.cs.cmu.edu/~yaser/

\(^4\) http://www.na.icar.cnr.it/~maddalena/
their shadows on the ground surface, which should be tolerated in foreground detection, and camouflage foreground objects (i.e., the color of the foreground object is similar to that of the covered background). In both cases, the proposed method separated the background and foreground satisfactorily. The results have shown that the proposed method can provide better performance in handling such non-stationary background than others.

In Fig. 10, this scenario contained significant motion of the curtain, as well as the background changes caused by automatic gain adjustment, the person wore clothes of bright colors, which are similar to the color of the curtain. The results have shown that the proposed method has detected the person quite well in such an environment.

In Figs. 11–13, those three scenes show the test sequence about the water, with non-stationary background of the fountain and sea waves. Fig. 12 shows results on a particularly challenging outdoor sequence, with three sources of dynamic motion: (1) the fountain, (2) the tree branches above, and (3) the shadow of the trees branches on the grass below [36]. In both cases, the proposed method can yield superior performance than the four former works in terms of the test results. Yet, we would like to point out a weakness of the proposed method. As it can be seen in Fig. 11 (g) and (h), the only drawback of proposed learning strategy is that the presence of a static object in the first frames, in subsequent frames, the object moves and uncovers the real background, will introduce an artifact be classified as foreground, commonly called a ghost [37]. The strategy of the suppress the learning of real foreground objects into the background making the ghost fade over time slowly.

In Fig. 14, those typical environment is shopping center. There is constant motion, and every frame contains people, no clean background training. Significant shadows of moving persons cast on the ground from different directions. The proposed method has obtained the satisfactory results in these environments, detected and removed shadows successfully.

In Fig. 15, in these situations, the motion of escalators would make the background modeling difficult. Further, the background model is hard to be established if there are frequent human flows in the scenes. Our test results have shown that the proposed method performed quite satisfactorily for such difficult scenarios.

In Figs. 16 and 17, those two scenes with a light switch on/off should be quickly updated into background models, and system should not lose the sensitivity to detect real foreground objects. In this example, the proposed method is quite successful in modeling the background, and that the persons have been almost perfectly detected. From the comparisons with four methods in this example, it is clear that this typical once-off background change can be properly handled by the proposed method but not by former MoG approaches.

Visually, the results of proposed method look better and are the closest to ground-truth references. This is confirmed by the results of quantitative evaluation.

3.3. Quantitative evaluations

To get an accurate evaluation of the proposed method, the criteria of recall and precision are employed. Recall, also known as detection rate, gives the percentage of detected true positives as compared to the total number of true positives in the ground truth

\[
\text{recall} = \frac{TP}{TP + FN}
\]

where TP is the total number of true positives, and FN is the total number of false negatives, which accounts for the number of
Fig. 10. Foreground detection results of sequence ‘Curtain’. (a) Original frame. (b) Ground truth. (c) MoG [11]. (d) Lee [18]. (e) Lin [24]. (f) SC-SOBS [23]. (g) Proposed. (h) Pixels classification result. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Fig. 11. Foreground detection results of sequence ‘Fountain’. (a) Original frame. (b) Ground truth. (c) MoG [11]. (d) Lee [18]. (e) Lin [24]. (f) SC-SOBS [23]. (g) Proposed. (h) Pixels classification result.
foreground pixels incorrectly classified as background

\[
\text{precision} = \frac{TP}{TP + FP} \tag{25}
\]

Precision, also known as positive prediction, that gives the percentage of detected true positives as compared to the total number of pixels detected by the method, is generally used in conjunction with the recall. Where \( FP \) is the total number of false positives. Generally, a method is considered good if it reaches high recall values, without sacrificing precision. So, the \( F\)-Measure \( (F_1) \) [25] metric also adopted, that is the harmonic mean of precision and recall

\[
F_1 = 2 \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \tag{26}
\]

Firstly, we have used a test sequence in Sheikh [36], with 500 test frames in total, and the manually segmented ground-truth is supplied for each frame. As reported in [36], the sequence contained an average camera motion of approximately 14.66 pixels. In the sequence, the scene is empty for the first 276 frames, after which two objects (first a person and then a car) move across the field of view. Fig. 18 first shows a qualitative illustration of the results as compared to the ground truth and four state-of-the-art algorithms. Here, several different learning rate settings were tested for the MoG approach. The corresponding quantitative comparison is shown in Fig. 19. As shown in the third row of Fig. 18, at the beginning of the walking person move across the scene, frames 312th and 332th (300th–335th frames), the MoG takes use of a small learning rate of 0.0025 performs poorly and labels a large number of background pixels as foreground (false positives) due to the dynamic background which caused by the periodicity of the camera motion. While the proposed approach has almost no false detections, which be proved by the performance of the precision in Fig. 19(b). On the other hand, the MoG uses a relatively high learning rate of 0.05 may handle dynamic motions at the beginning of the sequence. However, the system becomes easier to perturbed by noise and foreground objects. As shown in the fifth row of Fig. 18, at frame 432th (400th–470th frames), the MoG with learning rate of 0.05 labels a huge amount of foreground pixels as background (false negatives) on the inner...
areas of the moving car. It is proved that MoG with a high learning rate will cause substantial degradation in performance, as shown in Fig. 19(a). According to the overall performance shown in Fig. 19, the MoG uses a fixed learning rate is unsuited to dynamic scenes since the learning strategy of different pixels should be different. Clearly, with an adaptive learning strategy, the proposed method’s detection accuracy both in terms of three criteria higher than the other approaches.

Moreover, for the sake of comparison, video test sets Li [34], include nine video sequences are adopted. Most of the sequences contain several thousand video frames, with 20 test frames manually labeled for each sequence as the ground-truth. The scores of the five algorithms (two different learning rates for MoG) for test sets are compared in Fig. 20, clearly, the proposed method can yield superior performance than the four former works in terms of the three criteria.

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.neucom.2013.03.013.

Our method has been implemented in C++. Tests were executed on a PC with a 2.8 GHz Core i7 CPU and 4 GB of RAM. Finally, to complete our analysis, in Table 2 we report the mean number of frame-rate (FPS) by different resolution over the whole video sequences described in Section 3.2 for the proposed method and four state-of-the-art works. Table 2 shows that the proposed method is always slower than three former MoG works, but the frame-rate is sufficient to obtain real-time processing.

4. Conclusions

Mixture of Gaussians is a widely-used background subtraction method. In MoG’s learning process, there is a trade-off in the selection of the learning rate. Typically, a very small constant is used to maintain the stability of system. While this leads to slow convergence when background needs to adapt to a new cluster. This paper proposes a novel learning strategy for the MoG. We assign individual learning rates for each pixel with respect to space and time, and update them over time. The proposed approach is tested in most of the challenging situations mentioned in [25] including:

1. Gradual illumination changes: Outdoor environment (see Figs. 8, 12 and 18).
2. Sudden illumination changes: Switch on/off light (see Figs. 16 and 17).
3. Dynamic background: Waving trees and flags (see Figs. 8 and 9), motion of the curtain (see Fig. 10), fountain and water surface waves (see Figs. 11–13), escalators (see Fig. 15).
4. Camouflage: Objects poorly differ from the appearance of background (see Fig. 10).
5. Shadows: Shadows cast by moving objects (see Figs. 14 and 18).
6. Bootstrapping: No clean background training (see Figs. 14 and 15).
7. Video noise: Camera automatic gain adjustment (see Fig. 10), camera motion (see Fig. 18).

Quantitative evaluation and comparison with the existing method have shown that an improved performance for foreground object detection in complex background has been achieved. Experimental results show that, via different roles of different learning strategy, the trade-off between robustness to background changes and sensitivity to detect foreground can be effectively regularized. In the future, we will focus on improving the computational efficiency of the proposed approach.
Fig. 15. Foreground detection results of sequence "Escalator". (a) Original frame. (b) Ground truth. (c) MoG [11]. (d) Lee [18]. (e) Lin [24]. (f) SC-SOBS [23]. (g) Proposed. (h) Pixels classification result.

Fig. 16. Foreground detection results of sequence "Lobby". (a) Original frame. (b) Ground truth. (c) MoG [11]. (d) Lee [18]. (e) Lin [24]. (f) SC-SOBS [23]. (g) Proposed. (h) Pixels classification result.
Fig. 17. Foreground detection results of sequence "Light Switch". (a) Original frame. (b) Ground truth. (c) MoG [11]. (d) Lee [18]. (e) Lin [24]. (f) SC-SOBS [23]. (g) Proposed. (h) Pixels classification result.

Fig. 18. Foreground detection results of sequence "camera motion". The top row presents the original frames: 312th, 332th 380th, 432th frames. The second row shows ground truth segmentation. The third, fourth and the fifth rows are the results obtained by MoG [11] using the learning rate $\alpha$ with 0.0025, 0.01, and 0.05 respectively. The sixth row is the results of Lee [18]. The seventh row is the results of Lin [24]. The eighth row is the results of SC-SOBS [23]. While the last row is the results of proposed.
Fig. 19. Quantitative comparison of sequence “camera motion” (a) compares the Recall, (b) compares the Precision and (c) compares the F-Measure values with four state-of-the-art MoG works against proposed method.

Fig. 20. Comparisons of Recall, Precision and F-Measure values for test sets Li [34]. (a) Curtain. (b) Campus. (c) Lobby. (d) Shopping Mall. (e) Hall. (f) Bootstrap. (g) Escalator. (h) Water surface. (i) Fountain.
Table 2
Comparisons of frame-rate (FPS) for five methods.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Methods</th>
<th>MoG</th>
<th>Lee</th>
<th>Lin</th>
<th>SC-SOBS</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>160 × 120</td>
<td>191.31</td>
<td>188.97</td>
<td>181.67</td>
<td>121.03</td>
<td>147.12</td>
<td></td>
</tr>
<tr>
<td>160 × 128</td>
<td>189.76</td>
<td>187.01</td>
<td>180.11</td>
<td>110.95</td>
<td>145.36</td>
<td></td>
</tr>
<tr>
<td>320 × 256</td>
<td>56.61</td>
<td>52.91</td>
<td>46.30</td>
<td>31.94</td>
<td>37.74</td>
<td></td>
</tr>
<tr>
<td>360 × 240</td>
<td>53.18</td>
<td>49.38</td>
<td>45.17</td>
<td>28.54</td>
<td>35.31</td>
<td></td>
</tr>
</tbody>
</table>

Acknowledgments

The work is supported by National Natural Science Foundation of China (No. 60972124), National High-tech Research and Development Program of China (863 Program) (No. 2009AA01Z321) and Research Fund for the Doctoral Program of Higher Education of China (No. 20110201110012).

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.neucom.2013.03.013.

References


Xin Liu received the B.S. degree in Computer Science from Changchun University of Science and Technology, Changchun, China, in 2003, and the M.S. degree in Computer Science from Kunming University of Science and Technology, Kunming, China, in 2007. He is currently working toward the Ph.D. degree at the School of Electronics and Information Engineering, Xian Jiaotong University, Xian, China. His current research interests mainly include video surveillance, object tracking, and behavior understanding.

Chun Qi is currently a professor and Ph.D. supervisor at School of Electronics and Information Engineering, Xian Jiaotong University. His current research interests mainly include image processing, pattern recognition and signal processing.