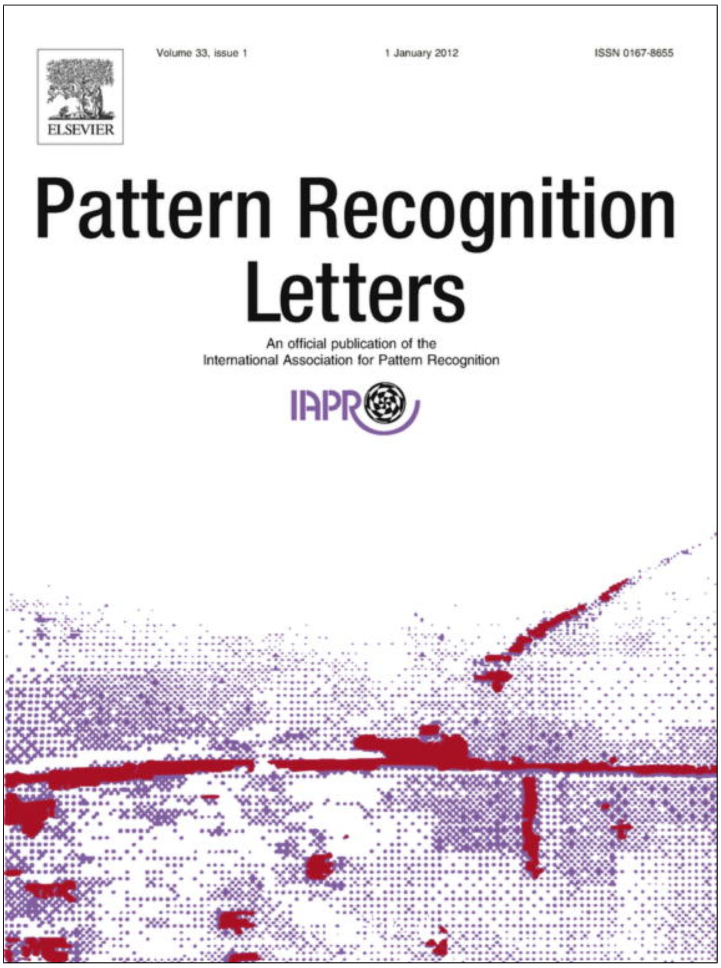


Volume 33, issue 1 1 January 2012 ISSN 0167-8655

ELSEVIER

Pattern Recognition Letters

An official publication of the
International Association for Pattern Recognition

IAPR

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Pattern Recognition Letters 33 (2012) 1134–1147

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Background modeling by subspace learning on spatio-temporal patches

Youdong Zhao [a,b,*], Haifeng Gong [b,c,d], Yunde Jia [a], Song-Chun Zhu [b,c]

[a] Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, PR China
[b] Lotus Hill Research Institute, EZhou 436000, PR China
[c] Department of Statistics, UCLA, Los Angeles, CA 90095, Unites States
[d] Google Inc., Mountain View, CA 94043, United States

ARTICLE INFO

Article history:
Received 26 May 2010
Available online 1 February 2012
Communicated by A. Shokoufandeh

Keywords:
Visual surveillance
Background modeling
Spatio-temporal patch
Subspace learning

ABSTRACT

This paper presents a novel background model for video surveillance—Spatio-Temporal Patch based Background Modeling (STPBM). We use spatio-temporal patches, called *bricks*, to characterize both the appearance and motion information. Our method is based on the observation that all the background bricks at a given location under all possible lighting conditions lie in a low dimensional background subspace, while bricks with moving foreground are widely distributed outside. An efficient online subspace learning method is presented to capture the subspace, which is able to model the illumination changes more robustly than traditional pixel-wise or block-wise methods. Experimental results demonstrate that the proposed method is insensitive to drastic illumination changes yet capable of detecting dim foreground objects under low contrast. Moreover, it outperforms the state-of-the-art in various challenging scenes with illumination changes.

1. Introduction

Background modeling is a key component in a video surveillance system with static cameras. In the past decade, various background modeling algorithms (Piccardi, 2004; Elhabian et al., 2008; Bouwmans, 2009) have been proposed and achieved good performance on well-illuminated scenes. However, the challenging scenes with lighting and illumination changes still remain unsolved, such as (1) sudden sunlight changes in daytime, (2) light turned on or off, and (3) car lighting in nighttime outdoor scenes. Especially, in a nighttime outdoor scene, the faint lighting, low signal-noise-ratio (SNR), low contrast, and drastic illumination changes are combined to form a really difficult scenario for background modeling.

Spatial neighborhood information and temporal one are two fundamental elements to understand appearance structure and dynamic motion respectively and *complementary* to each other. For example, in a low contrast environment, the motion of foreground provides most of the visual information; whereas when there are illumination changes, the appearance of foreground contributes the main visual information. However, the traditional **pixel-wise** methods (Wren et al., 1997; Stauffer et al., 2000; Elgammal et al., 2002) model the background as a set of independent pixel processes without considering neighborhood information, the **block-wise** methods use only the spatial correlations between pixels (Seki et al., 2003; Heikkila and Pietikainen, 2006; Lin et al., 2009) or employ the spatial and temporal information separately (Monnet et al., 2003; Wang et al., 2007), and the **motion based** methods (Wixson, 2000) exploit the temporal neighborhood information alone. While it is difficult for these methods to deal with the above scenarios individually, background modeling will benefit from utilizing the spatial and temporal information jointly. Moreover, according to the illumination literature (Belhumeur and Kriegman, 1998; Basri and Jacobs, 2003; Garg et al., 2009), the illumination variations of a static object (e.g., a background patch in a surveillance scene) could be represented by a low-dimensional subspace under the assumption of Lambertian surface.

Motivated by these observations, we propose to build background models on *spatio-temporal patches* (called “bricks”), which characterize both the appearance and motion information in the spatial and temporal neighborhood of a pixel (e.g., 6 × 6 × 4 pixels as shown in Fig. 1). In the proposed method, a brick is the atomic processing unit, which differs from the traditional pixel-wise or block-wise methods. Similar to image patches (or blocks) (Belhumeur and Kriegman, 1998), we observe that under all possible lighting conditions the background bricks extracted from a given location lie in a *low-dimensional* subspace, i.e., *background subspace* or *background model*. Then, we present an efficient online subspace learning method to capture the background model and adapt it to the recent variations in a real scene. The low computation complexity of this

[*] Corresponding author at: Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, PR China. Tel./fax: +86 10 6891 4849.
E-mail addresses: zhaoyoudong@gmail.com (Y. Zhao), haifeng.gong@gmail.com (H. Gong), jiayunde@bit.edu.cn (Y. Jia), sczhu@stat.ucla.edu (S.-C. Zhu).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2012.01.012

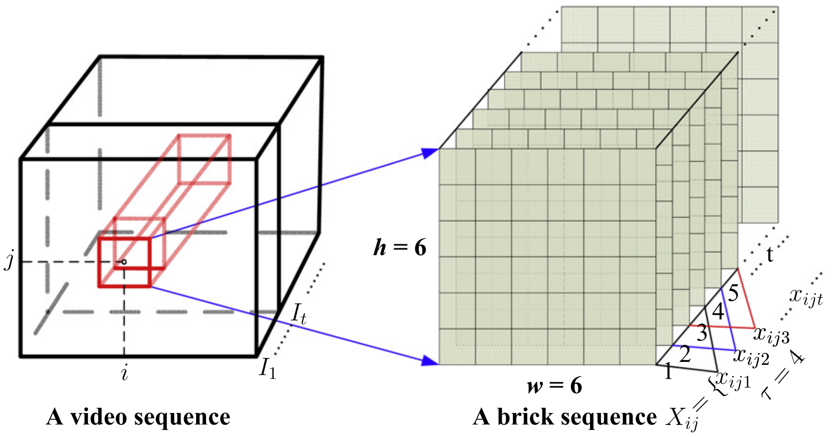


Fig. 1. A brick x_{ijt} is a small spatio-temporal patch with $h \times w \times \tau$ pixels around the point (i,j,t).

method makes it suitable for real-time applications. Here, we capture the background model of bricks at each background location independently.

Once the background model is learnt and adapted, we could perform "foreground detection" or "background subtraction" by thresholding the residual errors of incoming bricks on it. The residual errors of background bricks are usually distinct from those of foreground bricks and an adaptive threshold is proposed for reliable detection. Extensive experiments demonstrate the robustness of the proposed method in overcoming various illumination changes and low contrast in real-world video surveillance. The proposed brick based method advances the state-of-the-art in three aspects:

- *It is robust to both sudden and gradual illumination changes and achieves superior performance to the state-of-the-art.*
- *It is sensitive to dim moving objects under low contrast.*
- *It is simple yet effective for almost all the real challenging cases including indoor and outdoor, and daytime and nighttime scenes.*

1.1. Related work

While most of the **pixel-wise methods** (Wren et al., 1997; Stauffer et al., 2000; Elgammal et al., 2002; Kaewtrakulpong and Bowden, 2001; Lee, 2005) could handle gradual illumination changes by adapting their models, they often have difficulty dealing with sudden changes and are vulnerable to noises. Some methods explicitly alleviate the illumination effects by an extra illumination estimation (Messelodi et al., 2005) or a color model (Kim et al., 2005; Patwardhan et al., 2008). Recently, instead of modeling the intensities of pixels, Pilet et al. (2008) model the ratio of intensities between a stored background image and an input image by Gaussian Mixture Models to deal with sudden illumination changes. Their method successes in coping with sudden illumination changes, such as light switch in indoor scenes.

Block-wise methods use spatial correlations between pixels to improve robustness to noises and illumination changes. Seki et al (2003) exploit the cooccurrence of adjacent blocks for background subtraction. Heikkila and Pietikainen (2006) present a texture-based method (TBMOD), which employs the Local Binary Pattern (LBP) operator and can tolerate considerable illumination variations. In (Yao and Odobez, 2007), a multi-layer method is proposed, which combines the LBP feature and a color feature. In (Grabner and Bischof, 2006; Lin et al., 2009), classification based methods are proposed using image blocks. Edge (or gradient-based) features are used to model the background for its robustness to illumination changes in (Yang and Levine, 1992). A fusion of color and edge information is used in (Jabri et al., 2000). Noriega and Bernier (2006) combine local kernel histograms and contour-based features for background subtraction.

Motion based methods exploit temporal neighborhood information for foreground detection. Wixson (2000) defines a salient motion that tends to move in a consistent direction over time and detects the salient motion by integrating frame-to-frame optical flow. The information of successive frames enhances the saliency of moving objects despite of the similarity of appearances to the background.

A subspace learning based method for background modeling is first introduced by Oliver et al. (2000). This method establishes a global subspace over the whole frames, i.e., the eigen-background model, which can handle the global illumination changes to a certain degree. But they cannot deal with local illumination changes, and fail to distinguish slow moving foreground objects. While some improvements (Li, 2004; Skočaj et al., 2007; Skočaj and Leonardis, 2008) are made to deal with the slow moving objects, the local illumination change still remains unsolved. Some other methods (Monnet et al., 2003; Wang et al., 2007) operate on image blocks and exploit an additional prediction model (e.g., on-line auto-regression model) to predict future frames to capture the dynamic changes in temporal domain. While their methods capture the spatial and temporal information by two separate models respectively and can deal with the local illumination changes, they still miss detections in low contrast cases like other block-wise methods, which could be addressed by our brick-based method that models the spatio-temporal variations jointly in the brick space.

Some researchers also use the **spatio-temporal information** for background modeling. Pless (2005) builds background models based on the responses of spatiotemporal derivative filters at each pixel. Wang et al. (2006) integrate spatial and temporal dependencies for foreground segmentation and shadow removal via a dynamic probabilistic framework based on the conditional random field.

Stereo information is also employed to construct the background model invariant to illumination changes (Ivanov et al., 2000). Their method requires an off-line construction of disparity fields mapping the primary background images via using two or more cameras and suffers from both missing and false detections due to certain geometric considerations. Lim et al. (2005) introduce an improvement to Ivanov's method (Ivanov et al., 2000) to alleviate the false detections and some other issues.

1.2. Paper organization

The rest of the paper is arranged as follows: Section 2 presents two conjectures about the distribution of a video brick sequence, which is the motivation of the novel proposed algorithm.

Section 3 presents our background model based on video bricks and online subspace learning and introduces an adaptive thresholding scheme for foreground object detection. In Section 4, the parameter selections of the proposed method are discussed in details and the comparison results are described. The discussion and conclusions are made in Section 5.

2. Empirical distributions of video bricks

A video brick x_{ijt}, as shown in Fig. 1, consists of a small number of image patches extracted at the same location in successive video frames. All bricks in a brick sequence X_{ij} distribute in the brick space according to their variations of appearance and motion, which are often caused by illumination changes, foreground occluding, or noises, etc. There are mainly three types of bricks: (i) normal background bricks, (ii) bricks with illumination changes and (iii) bricks with foreground occlusion (Fig. 4). The first two are considered as *background bricks*. In order to build background model on video bricks, we need to study the properties of the space distribution of video bricks in the surveillance scenario.

The set of image patches (or blocks) of an object, under all possible lighting conditions, usually lies in a low-dimensional subspace in the image space (Belhumeur and Kriegman, 1998; Basri and Jacobs, 2003; Garg et al., 2009). As the temporal extension of image patches, here, we have the following two conjectures on the space distribution of video bricks in visual surveillance.

Conjecture 1. *The set of bricks at a given static background location under all possible lighting conditions lies in a very low-dimensional (2–5) manifold (or background subspace) embedded in the high-dimensional brick space.*

The static background means that except illumination variation there is not any other real physical motion, e.g., trees or water waving in a wind.

Conjecture 2. *Due to the diversity of foreground, the bricks with foreground occlusion are widely scattered in the high-dimensional brick space and can be well separated from the background subspace.*

Two experiments are conducted to validate the above two conjectures. We select a patch with 15×15 pixels on the road surface in a night outdoor scene and use successive 7 patches to form a brick. A set of 20,000 bricks containing various changes is captured. To facilitate the analysis, the brick set is manually divided into three subsets:

- ■ Subset-A 14,968 normal background bricks (i.e., excluding the random noises, there are no any other factors, e.g., lighting changes, acting on the brick).
  - □ Subset-A1 14,168 bricks selected from Subset A randomly.
  - □ Subset-A2 800 bricks selected from Subset A randomly.
- ■ Subset-B 3449 background bricks with only lighting changes.
  - □ Subset-B1 2649 bricks selected from Subset B randomly.
  - □ Subset-B2 800 bricks selected from Subset B randomly.
- ■ Subset-C 1583 bricks with foreground occlusion.

The bricks in subsets A and B are considered as "background bricks", while the ones in subset C are "foreground bricks". For training and testing, subsets A and B are further randomly divided into two subsets respectively. Then we analyze the space distribution of bricks in each subset off-line by principal component analysis (PCA) (Oliver et al., 2000).

Fig. 2 plots the curves of eigenvalues learned from two subsets. Subsets A2 and B are used for learning in Fig. 2(a). In Fig. 2(b), the training samples are all the bricks in subset C. By preserving 95% information, we can see that background bricks lie in a low-dimensional (2–4) subspace, whereas the foreground bricks distribute in a high-dimensional (>20) space. For instance, in Fig. 3 the distribution of the background bricks with only lighting changes (subset B) on the first two principal components of the background subspace is shown. The first component describes the brightness of the bricks and the second interprets the directions of car lighting on the bricks. There are several underlying curves connected by points, each of which stands for a procedure that the brightness of bricks changes with a car passing the scene. The curves above the zero point (e.g., the curve marked by red square) correspond to different cars passing in the same direction, while the curves below the zero point (e.g., the curve marked by blue circle) correspond to cars passing in an opposite direction. From Fig. 3, we can find that the drastic car lighting changes on a road surface could be characterized meaningfully by a 2 dimensional subspace.

To demonstrate the discriminative power of the background model, two background subspaces are learnt on subsets A1 and B1 by keeping 95% information respectively. The residual errors of test bricks on them are shown in Fig. 4(a) and (b), where the horizontal coordinate is the index of test bricks (the first 1583 bricks are from subset C, the middle 800 are from subset A2 and the last 800 come from subset B2). The residual errors of the foreground bricks are distinguished from those of the background bricks. Fig. 4(c) and (d) give the corresponding ROC curves (red solid)[1] of classification of foreground/background. Note that both subspaces separate the foreground bricks from the background ones. These results demonstrate that normal background bricks and bricks with lighting changes do lie in the same subspace, from which the foreground bricks could be separated well. Moreover, it is worth noting that the discriminative power of the background model learnt on bricks with lighting changes (B1) is higher than that learnt on normal background bricks (A1). For comparison, we also perform the same experiments in the block level. That is, the basic processing unit is an image block (or patch) instead of the video brick. The corresponding ROC curves (blue dashed) are illustrated in Fig. 4(c) and (d). It is apparent that the background subspaces learnt on video bricks are more discriminative than the ones learnt on image blocks.

3. Background modeling on video bricks

Based on the conjectures on space distribution of video bricks, an online subspace learning method is employed to capture the background subspaces and to adapt them on-the-fly.

3.1. Notation

As a new frame I_t arrives, we extract a patch with $h \times w$ (e.g., 6×6) pixels around each pixel (e.g., at point (i,j)). It is combined with the previous $\tau - 1$ patches (e.g., $\tau = 4$) to form a video brick x_{ijt} with $h \times w \times \tau$ pixels (as shown in Fig. 1). Then at a location (i,j), we obtain a brick sequence $X_{ij} = \{x_{ij1}, x_{ij2}, \ldots, x_{ijt}, \ldots\}$. Each brick x_{ijt} is reshaped into a D-dimensional ($D = h \times w \times \tau$) column vector.

The background bricks, including normal background bricks and bricks with only illumination changes lie in a low-dimensional background subspace S_{ij} in the $\mathbb{R}^D$ brick space, which can be learnt and maintained by an online subspace learning method (Weng et al., 2003; Ross et al., 2008). The foreground detection is implemented by thresholding the distance $L(x_{ijt}, S_{ij})$ between the incoming brick x_{ijt} and the background model S_{ij}.

For simplicity of notation, the formulation of the proposed algorithm is based on one brick sequence extracted from a fixed local

[1] For interpretation of color in Figs. 3 and 4, the reader is referred to the web version of this article.

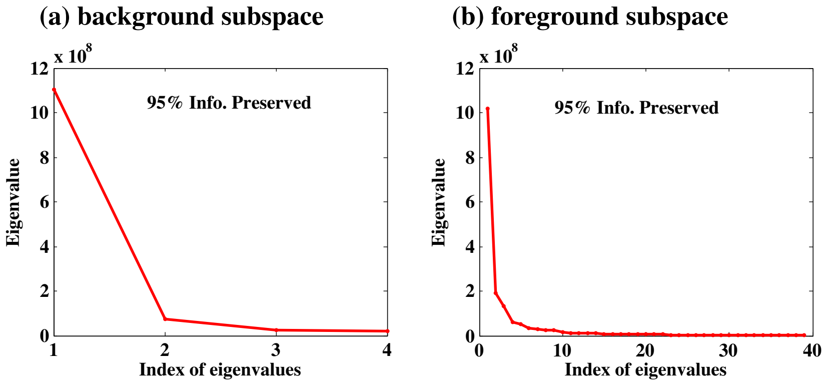


Fig. 2. Plots of eigenvalues for two brick subspaces learnt from two subsets.

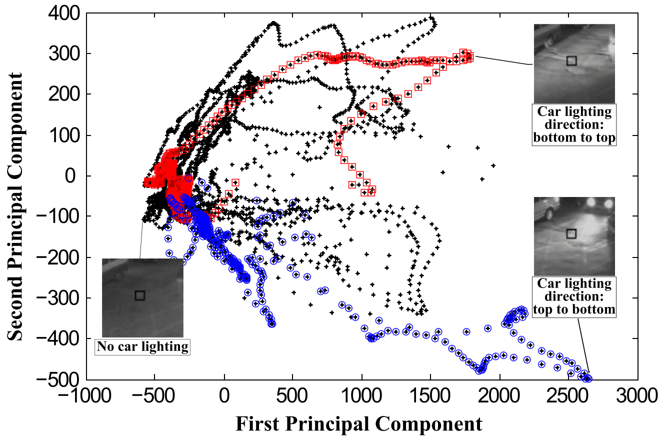


Fig. 3. The distribution of bricks with lighting changes on the first two background PCs. Two of the underlying curves corresponding to two cars passing in an opposite direction are marked by red square and blue circle respectively.

patch and the subscript of the location (i,j) is omitted. The extension to the whole frame is straightforward.

3.2. Background modeling by online subspace learning

In this paper, we adopt an online principal component analysis algorithm, which is a modified version of the Candid Covariance-free IPCA (CCIPCA) (Weng et al., 2003) algorithm, to compute the principal components of a brick sequence, i.e., the background model. The online version inherits the merits of CCIPCA in fast convergence rate and low computational complexity. The main differences are the learning rate setting and updating scheme. By our new learning rate, as discussed below, the online version develops the CCIPCA algorithm to adapt the recent variations. The new updating scheme keeps the model away from outliers (e.g., foreground bricks). These properties make it suitable for the video surveillance application.

Given a brick sequence $X = \{x_1, x_2, \ldots, x_t, \ldots\}$ with the latest estimated mean brick μ_{t-1}, the first d dominant eigenvectors of the background subspace S are estimated recursively by following two equations:

$$v_{k,t} = [1-\eta(t)]v_{k,t-1} + \eta(t)u_{k,t}\left\langle u_{k,t}, \frac{v_{k,t-1}}{\|v_{k,t-1}\|}\right\rangle, \quad (1)$$

$$u_{k+1,t} = u_{k,t} - \left\langle u_{k,t}, \frac{v_{k,t}}{\|v_{k,t}\|}\right\rangle \frac{v_{k,t}}{\|v_{k,t}\|}, \quad (2)$$

where $\langle\cdot,\cdot\rangle$ denotes inner product, $v_{k,t}$ is the kth $(1 \leqslant k \leqslant d)$ eigenvector updated by the tth brick, $u_{1,t} = x_t - \mu_{t-1}$, $u_{k+1,t}$ is the residual brick after being projected onto the first k estimated eigenvectors and $\eta(t)$ is the learning rate. In the initialization stage $(1 \leqslant t \leqslant d$, and $t = k)$, $v_{k,t} = x_t - \mu_{t-1}$ and $\mu_0 = 0$. To speed up the convergence of estimation, a batch PCA algorithm could be performed on the first few bricks (e.g., $T_{\mathrm{init}} = 50$) to initialize the recursive algorithm instead of using the first d samples directly. In addition, the mean μ_t is updated by

$$\mu_t = [1-\eta(t)]\mu_{t-1} + \eta(t)x_t. \quad (3)$$

The learning rate $\eta(t)$ is set as

$$\eta(t) = \frac{1-\alpha}{c(t)} + \alpha, \quad (4)$$

where α is a constant learning rate and $c(t)$ counts the number of matching observations for the model S. In (4), it is apparent that in the initial stage of learning a model when only a few matching samples have been observed, $\eta(t) \approx 1/c(t)$, and parameters are updated in a manner consistent with the CCIPCA algorithm (Weng et al., 2003). As more samples are introduced for the model estimation, $\eta(t)$ approaches α and behaves like a typical recursive learning (Lee, 2005).

The learning rate setting in (4) distinguishes our new online version algorithm from the CCIPCA algorithm (Weng et al., 2003) and the method adopted by Zhao (2008). In (Weng et al., 2003), the model updated by (1) and (2) with learning rate $\eta(t) = (1 + l)/t$ (l is a positive amnesic parameter) reflects the long term cumulative distribution and could not adapt to distribution changes over time in a real surveillance task. If the learning rate is set to a fixed value α (e.g., 0.001) like Zhao (2008), the model estimated will reflect the most recent observations within roughly $L = 1/\alpha$ window with exponential decay. However, in this way, it loses the good property of the CCIPCA in convergence rate.

The incoming bricks do not always belong to the background subspace S, so it is necessary to keep these *outliers* away from disturbing the subspace model. When a new brick x_t arrives, the distance L between x_t and S is recursively computed as follows,

$$u_{k+1,t} = u_{k,t} - \left\langle u_{k,t}, \frac{v_{k,t-1}}{\|v_{k,t-1}\|}\right\rangle \frac{v_{k,t-1}}{\|v_{k,t-1}\|}, \quad (5)$$

$$L(x_t, S) = \|u_{d+1,t}\|, \quad (6)$$

where $k = 1, 2, \ldots, d$ and $u_{1,t} = x_t - \mu_{t-1}$. If L is less than a pre-defined threshold T (an adaptive threshold will be discussed in the next subsection), x_t will be marked as background and used to update S, otherwise foreground. Note that the distance L in (6) is just the residual error of x_t on S and different from the reconstructed error computed in the traditional manner:

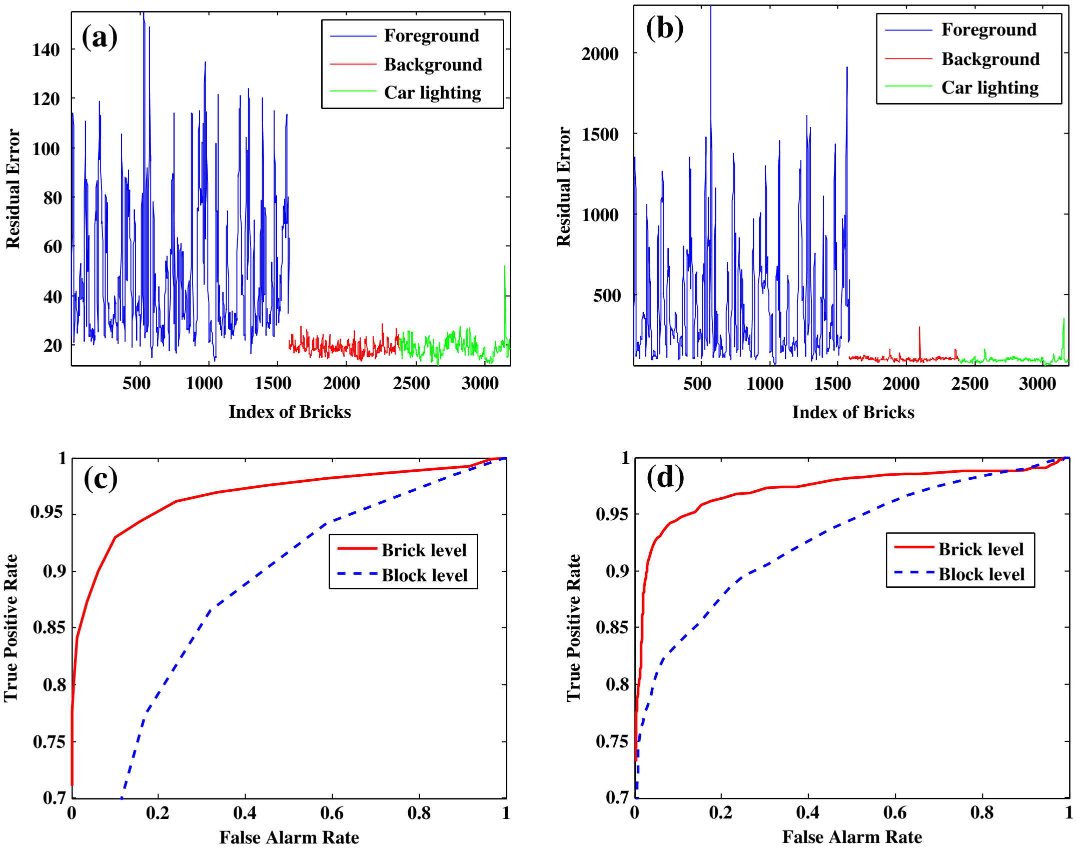


Fig. 4. Residual error curves of three types of bricks projected on two background subspaces learnt on (a) normal background bricks and (b) background bricks with lighting changes. (c) and (d) illustrate the corresponding ROC curves of classification of background/foreground in the brick level (red solid) and block level (blue dashed).

$$L(x_t, S) = \left\| u_t - \sum_{k=1}^{d} \frac{v_{k,t-1}}{\| v_{k,t-1} \|} \left\langle u_t, \frac{v_{k,t-1}}{\| v_{k,t-1} \|} \right\rangle \right\|, \tag{7}$$

where $u_t = x_t - \mu_{t-1}$. Eq. (7) requires the eigenvectors $\{v_{k,t-1}\}_{k=1}^{d}$ orthogonal to each other. In this case, (5) and (6) are equivalent to (7). However, if the orthogonality does not hold strictly, just like our current case, the residual in (6) is significantly different from that in (7) (Weng et al., 2003).

3.3. Threshold setting and foreground detection

Threshold setting. Fig. 5 illustrates the results of an indoor scene with two different threshold schemes. The residual errors and its lateral view are shown in column 2, which are calculated by (5) and (6). Note that the residual errors of moving foreground objects are distinctive. It is straightforward to detect foreground objects by a fixed threshold T (e.g.,70) (in column 3). In (Zhao et al., 2008), a fixed threshold for all pixels and all frames often achieves satisfactory results of foreground detection. However, there are still many missing detections on the bodies.

In experiments, we find that besides the variations of appearance and motion the residual error of an incoming brick x_t on the background subspace is mainly influenced by two factors: the intensity of the mean brick μ_{t-1} and the intensity of x_t. In other words, the residual errors in the region with high intensity are often higher than those in the region with low intensity. And the

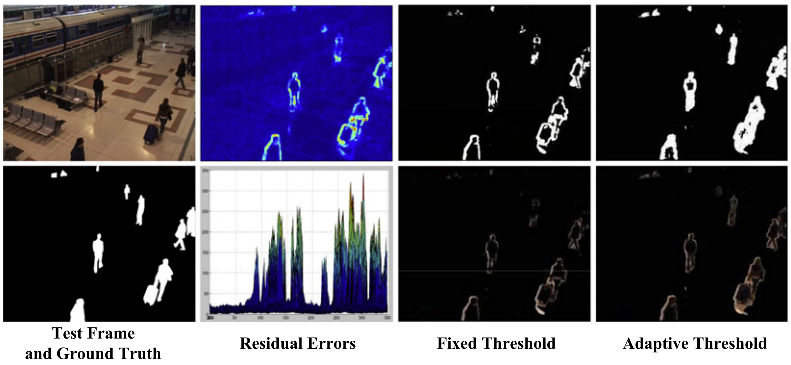


Fig. 5. Comparison of two threshold schemes: fixed and adaptive threshold. The residual error map and its lateral view are shown in 2nd col. In the lateral view of the residual errors, the vertical axis indicates the values of residual errors and the horizontal axis stands for the width (352 pixels) of the testing frame. Foreground masks (right-up) and detected foreground objects (right-down) are shown.

intensity of incoming brick has a similar influence. Based on these observations, we propose an adaptive threshold T_{adap} for each brick sequence empirically, which is proportional to the norm of μ_{t-1} and adjusted by the difference $\delta_t = \sum_i (x_t^i - \mu_{t-1}^i)$ between x_t and μ_{t-1}. Here, x_t^i is the intensity of the i-th pixel in the tth brick x_t.

$$T_{\text{adap}} = \begin{cases} \|\mu_{t-1}\|/\lambda + \delta_t/\zeta, & \text{if } \delta_t > C, \\ \|\mu_{t-1}\|/\lambda + \delta_t/\rho, & \text{if } \delta_t < -C, \\ \|\mu_{t-1}\|/\lambda, & \text{otherwise}, \end{cases} \tag{8}$$

where λ is the main threshold factor, ζ and ρ are the accessorial factors, and C is a pre-defined positive threshold for the difference. In addition, a minimum threshold $T_{\min}$ for T_{adap} is kept, i.e., $T_{\text{adap}} = T_{\min}$, if $T_{\text{adap}} \leqslant T_{\min}$, avoiding a too small threshold. Employing this adaptive threshold, the foreground objects are detected more reliably as shown in Fig. 5. In the following, all tests are performed using the adaptive threshold.

Algorithm 1. Spatio-Temporal Patch based Background Modeling (STPBM) (for one brick sequence)

```
**Initialization:** Perform batch-PCA on first $T_{\text{init}}$ samples.
  The first $d$ PCs obtained as the initial values of $v_{i,0}$,
  $0 < i \leqslant d$ and the mean vector as the initial value of $\mu_0$.
  $c(0) = 0$, $N_{\text{fg}} = 0$;
**While** new brick $x_t$, $(t = 1,2,\ldots)$ **do**
  Compute matching threshold $T_{\text{adap}}$ by (8)
  **If** $T_{\text{adap}} < T_{\min}$ **then** $T_{\text{adap}} = T_{\min}$
  Compute residual error $L(x_t,S)$ by (5) and (6)
  **If** $L < T_{\text{adap}}$ **then**// Background
    $c(t) = c(t-1) + 1$;
    Compute learning rate $\eta(t)$ by (4)
    Update the background model $S$ by (1)–(3);
    **If** $N_{\text{fg}} > 0$ **then** $N_{\text{fg}} = N_{\text{fg}} - 1$;
  **Else**// Foreground
    $N_{\text{fg}} = N_{\text{fg}} + 1$;
    **If** $N_{\text{fg}} > T_{\text{stay}}$ **then** // Object staying
      Update the background model $S$ by (1)–(3);
    **Endif**
  **Endif**
**Endwhile**
```

Foreground detection. As a new brick x_t arrives, the residual error L of x_t on the subspace S is recursively computed by (5) and (6). If L is less than the threshold T_{adap}, x_t will be classified as background, otherwise foreground. If there is a position in the scene persisting to be foreground for N_{fg} frames and $N_{\text{fg}} > T_{\text{stay}}$, the incoming bricks will be used to update the background model. The N_{fg} is a counter and the threshold T_{stay} controls the stay time of foreground objects in the scene, which is application-related. Note that the threshold T_{stay} is introduced to account for the background evolution. For example, a car staying for a long time is usually considered as background. The proposed algorithm is called Spatio-Temporal Patch based Background Modeling (STPBM) and summarized in Algorithm 1 for one brick sequence.

3.4. Computational complexity

Let N be the number of pixels in a testing video frame, d be the number of principal components used and $(D = h \times w \times \tau)$ be the dimension of a brick. In Algorithm 1, there are two major processing steps, namely, computing residual error $L(x_t,S)$ and updating the background model S. The computational complexity of these two steps are both $O(dD)$. Then the computational complexity of Algorithm 1 is also $O(dD)$. If extended to the whole testing video frame, the computational complexity of the proposed algorithm of background modeling is $O(dDN)$. The value of d is quite small (usually 2–5) and therefore the computational complexity can be estimated to be $O(DN)$. It is apparent that the spatiotemporal scale of video bricks and the resolution of testing videos are two major factors that affect the performance of the proposed algorithm in terms of time. In order to achieve real time performance, the space–time scale of video bricks should be kept small and the resolution of testing videos should be not too high. In our real system, we usually adopt a partial-update scheme, i.e., only updating a small part of all the models (e.g., five rows every frame) in turn with each incoming frame. This scheme is effective to further reduce the computational complexity with performance reduced slightly.

4. Experiments

In this section, we first introduce a dataset for experimental evaluation, which contains various challenging problems for background modeling. The parameter setting of our STPBM is discussed in details. Then we compare our method to the state-of-the-art (Stauffer et al., 2000; Heikkila and Pietikainen, 2006; Pilet et al., 2008). Modified AS: We also evaluate our method on a common used dataset (Toyama et al., 1999) and an artificial dataset (Brutzer et al., 2011). All the video clips used in the following experiments, the results and the corresponding parameters for all methods are supplied as Supplementary material and available at "http://www.hyphone.org/stpbm".

4.1. Data sets

We evaluate the proposed algorithm on 10 challenging scenes shown in Fig. 11 and 13, where scenes 1–3, 5–7, and 9 are from LHI Dataset (Yao et al., 2007), scenes 4[2] and 8[3] are from PETS database, and scene 10 is from Heikkila and Pietikainen (2006). Each scene contains special problems with which a surveillance system often faces in real-world conditions. The resolution of all the video sequences is resized to 352 × 288 with 25 fps. And we only utilize the intensity information of video sequences. For each scene, three or four representative frames are labeled manually as ground truth for quantitative analysis. In total, there are 29 labeled frames with resolution 352 × 288 pixels.

Scenes 1 and 2 are two nighttime urban traffic scenes and scene 2 has more noises and lower contrast. Scene 3 is a night highway scene in distance. The three nighttime outdoor scenes contain heavy illumination changes due to car lighting. Scene 4 is collected from a train station, in which there are shadows, reflections etc. Scene 5 has 691 frames and is a very busy traffic scene, where many pixels are covered by moving cars from the first frame to the end. In scene 6, there are non-connected shadows on the road due to the occlusion of trees, waving leaves, low contrast and camera gain. Scene 7 is a campus scene in distance, and has many very small pedestrians, waving trees and low contrast regions. Scene 8 is another campus scene that has sudden sunlight changes. Scene 9 is a port scene that contains water ripple and small flying birds with low contrast. Scene 10 (Fig. 13) has heavy waving trees in the wind.

4.2. Parameter selection

While there are eleven parameters, the proposed algorithm is not very sensitive to most of them and there is a wide range from

[2] ftp://ftp.cs.rdg.ac.uk/pub/PETS2006/.

[3] ftp://ftp.cs.rdg.ac.uk/pub/PETS2001/.

Table 1
The parameter settings of our method for the results in Figs. 5–8, 11, 13, 17.

Figure	Scene(s)	β	d	α	λ	ζ	ρ	$T_{\min}$	T_{stay}
5–8	–	2	5	0.001	10	100	20	30	200
11	2–6, 8, 9	2	5	0.001	10	100	20	30	200
11	1, 7	2	5	0.001	10	100	20	16	200
13	10	2	5	0.001	10	100	20	30	200
17	1–7	1	5	0.001	10	100	20	90	100

which each parameter can choose a proper value. The detail settings of all parameters for the following experiments are given in Table 1. We now examine the impacts of each parameter one by one. As one is being inspected, others are kept fixed at the default values in Table 1. The quantitative results in Figs. 9 and 10 are computed over all the nine scenes in Fig. 11 based on ground truth.

Brick size. The brick size is critical to the performance of the proposed algorithm. Through empirical analysis, we find that a small space–time scale (e.g., 4 × 4 × 3 pixels) is enough for effective background modeling. In Fig. 6, the intermediate results of three brick sizes (i.e., 4 × 4 × 1 pixels, 3 × 3 × 2 pixels, and 4 × 4 × 3 pixels) are illustrated. It is apparent that the spatio-temporal information enhances the saliency of foreground object, which is essential to object detection. The ROC curves in Fig. 9 demonstrate the superiority of brick level methods to block level methods quantitatively. In addition, for the brick level methods, most of the false alarms occur on the contour areas of moving objects due to the use of spatio-temporal neighborhood information. While the missing detections in the brick level methods mainly occur on the flat body areas, those in the block level occur not only on the flat regions but also on the moving boundaries, especially in low contrast scenes in Fig. 9(b), which is unacceptable for real applications. Moreover, the computation cost is proportional to the brick size. With consideration of trade-off between the performance and efficiency, we select 4 × 4 × 3 pixels for the following experiments.

Number of initial samples T_{init}. As shown in Fig. 7, we test two values of T_{init} (i.e., 5 and 50) for a very busy traffic scene. It is obvious that our method is not sensitive to the initialization. In the following tests, T_{init} is set to 50 as default.

Subsample ratio β. In Fig. 8 , we run our algorithm on three video resolutions, namely original resolution 352 × 288 pixels ($\beta = 1$), medium resolution 176 × 144 pixels ($\beta = 2$) and low resolution 88 × 72 pixels ($\beta = 4$). In the original resolution, the results are the most accurate but the computation cost is also the highest. The **processing speeds** for the three resolutions are about 3 fps, 12 fps and 48 fps in C++ (Intel Core II 2.66G with 1 GB RAM). In the following, we report results of our method in the medium resolution.

Number of PCs d and learning rate α. The number of PCs d determines the descriptive power of the background model and is also related to the computation complexity. The background model adapts itself to the changes of a scene by a constant learning rate α in (4). The larger the learning rate is, the more weight is given to recent observations and the faster the adaptation. Although a wide range of values for two parameters could be chosen (as shown in Fig. 10), we empirically select $d = 5$ and $\alpha = 0.001$ as their default values.

Adaptive threshold T_{adap}. The adaptive threshold T_{adap} is mainly determined by the main threshold factor λ in (8). The influence of λ to performance is shown in Fig. 10. The two auxiliary factors η and ρ along with C determine the contribution of the new brick to T_{adap}. Generally just as the new brick is very different from the mean brick μ_t (too bright or dark), it will be used to adjust the threshold. So C should not be too small. Here, we set it as the ten times the dimensions of the brick (i.e., $C = 48 \times 10$). The smaller the η and ρ are, the larger the contribution of the new brick to T_{adap} is. In order to keep T_{adap} approximately stable, they should not be too small. Empirically, the η and ρ are fixed at 100 and 20 for all experiments. The minimum threshold $T_{\min}$ is relevant to noise level of the test scene. In the case of low noise level, a small value (e.g., 16) is proper. Otherwise, a high value (e.g., 30) will be needed.

Stay threshold T_{stay}. The threshold T_{stay} controls the stay time of foreground objects in the scene and is application-related. A small T_{stay} means that the background model will be updated by the stay foreground object detected quickly. In our experiments, it is usually set to 200.

4.3. Comparison with other methods

Comparison with two classical methods. When $\tau = 1$ (e.g., 4 × 4 × 1 pixels), our method is reduced to a block-based method, which is similar to early subspace learning methods (Monnet et al., 2003; Wang et al., 2007). Comparison results between block-based methods and brick-based methods are shown in Figs. 6 and 9. Here, we compare the proposed method (STPBM) with two classical approaches, i.e., MOG (Stauffer et al., 2000) and TBMOD (Heikkila and Pietikainen, 2006) on the nine scenes in Fig. 11. The MOG is one of the most widely used background modeling method and used as baseline in our comparison. The TBMOD is a typical block-based method that can deal with considerable illumination changes. Moreover, in the proposed and two compared methods there are not any higher level processing, e.g., illumination estimator, shadow removing, detection of stay object, etc. The MOG and TBMOD methods are evaluated with several sets of parameters according to the suggestions of the authors and the best results are reported for comparison.

From the results in Fig. 11, we can see that MOG performs worst for almost all the testing scenes while the proposed method achieves the best performance. For sudden illumination changes, e.g., car lighting in nighttime scenes 1–3, camera gain in scene 6 and sudden sunlight changes in scene 8, our method shows its ability to overcome extreme illumination changes. While TBMOD can tolerate moderate illumination changes, it fails in the extreme conditions. The results of TBMOD on scenes 1–3 and 5 with respect to various parameter settings show that while the false alarms could be reduced slightly by tuning the parameters, the missing detections correspondingly increase rapidly (these video results are available in our Website mentioned previous). In scenes 1, 4 and

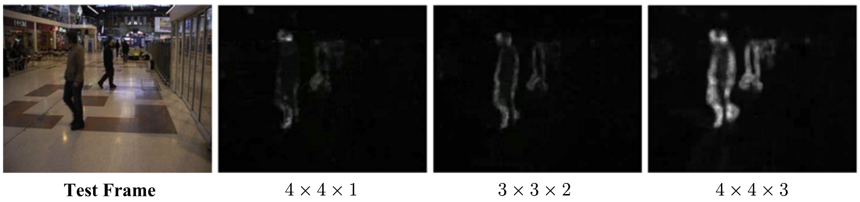


Fig. 6. The residual error maps obtained with three brick sizes. The block level method (4 × 4 × 1 pixels) gives low residual errors for foreground in the regions of both boundaries and bodies, whereas the brick level method (even only using two successive frames, e.g., 3 × 3 × 2 pixels) is sensitive to the moving boundaries. The size of 4 × 4 × 3 pixels is used for experiments hereafter.

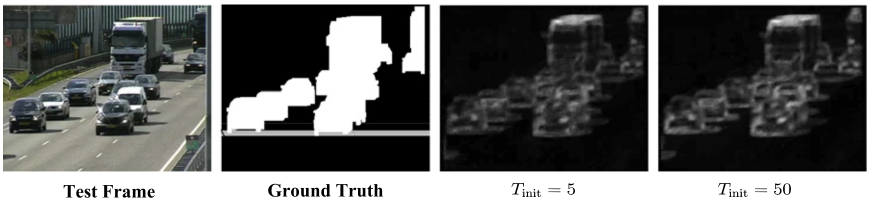


Fig. 7. The residual error maps computed with two different T_{init} illustrate the robustness of our algorithm to the model initialization.

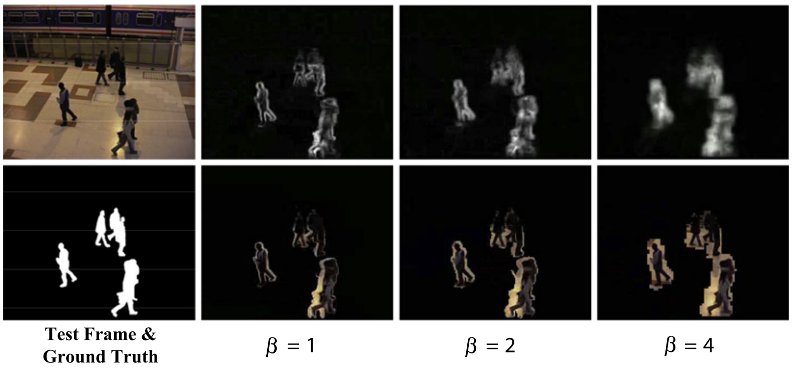


Fig. 8. Comparison of three subsample rates β. Residual error maps and detected objects are shown in row 1 and 2 respectively.

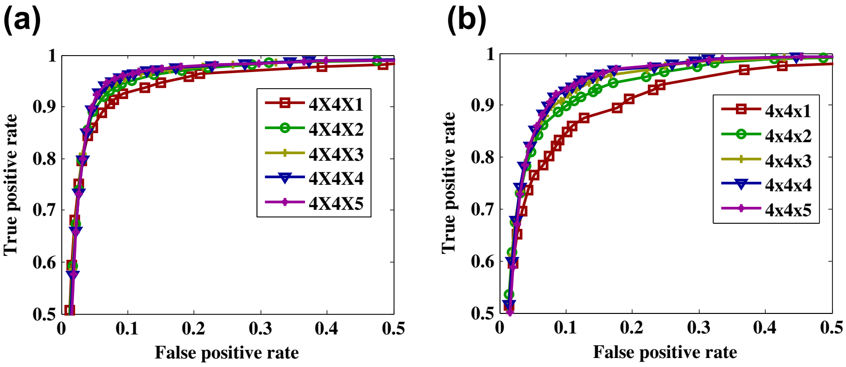
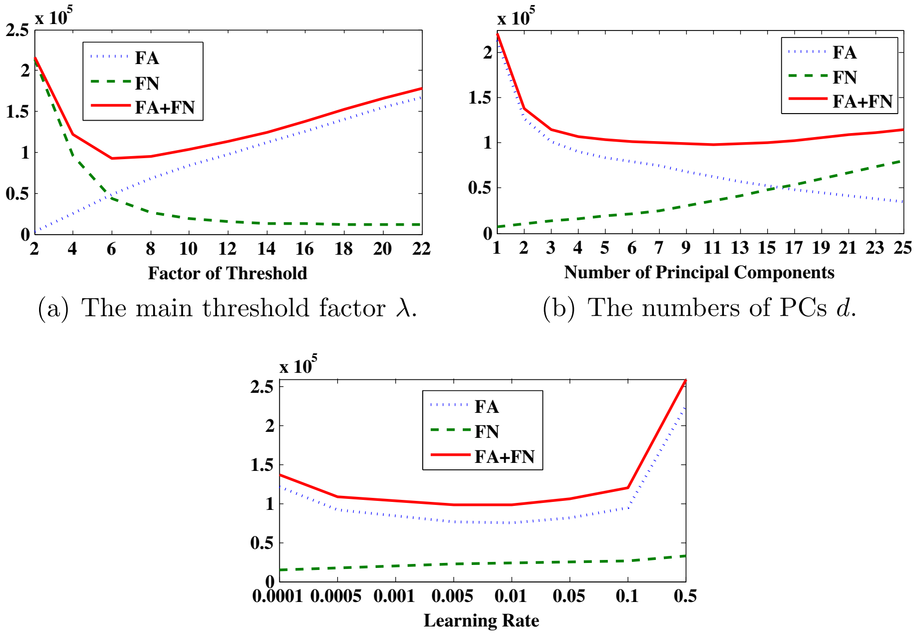


Fig. 9. Performance comparison among different brick sizes: (a) computed over all nine scenes and (b) computed over four scenes with low contrast, i.e., Scenes 1–3 and 6. Here, in order to compute the ROC curves, we simply adopt the fixed threshold scheme rather than the adaptive threshold scheme discussed in Section 3.3.

8, while both MOG and TBMOD suffer from false alarms due to shadows or reflections, our method could supress considerable shadows (even moderate reflections). In scene 5 and 8, our method fails in the strong shadows due to the moving boundaries caused by the shadows, which can be solved by specialized shadow removing technique (Huang and Chen, 2009; Finlayson et al., 2006). In low contrast conditions in scenes 1–3, 6, 7 and 9, while the block-based TBMOD suffers from heavy missing detections, our method could detect the dim moving objects successfully thanks to using motion and appearance information simultaneously. The results in scene 6, 7 and 9 demonstrate that our method could cope with dynamic background, such as waving trees and water ripples to a certain degree, which indeed violates our assumptions (see Section 2) of static background. For the busy traffic scene 5, MOG fails to establish background models and TBMOD suffers from both false alarms and missing detections due to the persistent moving cars from the first frame to the end, however, our method succeeds in modeling the background and detects the moving cars successfully. In addition, while our method is sensitive to subtle moving foreground objects, it is robust to various noises. These results further demonstrate the two conjectures on the brick distribution and the discriminative power of background subspaces captured by the online subspace learning method in various real-world conditions.

In Fig. 12, we also compare the performances of the three methods quantitatively based on the ground truth of the nine scenes in Fig. 11. These statistical results should be considered together with the visual results in Fig. 11. Our method gives less false alarms than the two compared methods in the scenes 1–4 and 8–9. In scenes 5 and 6, while MOG or TBMOD achieve similar results in the number of false alarms to our method, they have more missing detections as shown in Fig. 11. In scene 7, though our method produces more false alarms than the other two methods, it gives best results in sense of object detection (Fig. 11). Note that the false alarms of our method mainly occur around the bodies of the moving objects due to the use of spatio-temporal neighborhood information in STPBM. On the false negative, the difference is very small among three methods except the scene 5 where MOG fails to establish background models and TBMOD suffers from both false alarms and false negatives. Most of the false negatives of our method appear on the flat regions of the foreground bodies since the flat bricks with different intensities lie in the same subspace.



(a) The main threshold factor λ.

(b) The numbers of PCs d.

(c) The constant learning rate α.

Fig. 10. Performance evaluation with three parameters. The vertical axis is the number of false alarm (FA) or false negative (FN) pixels.

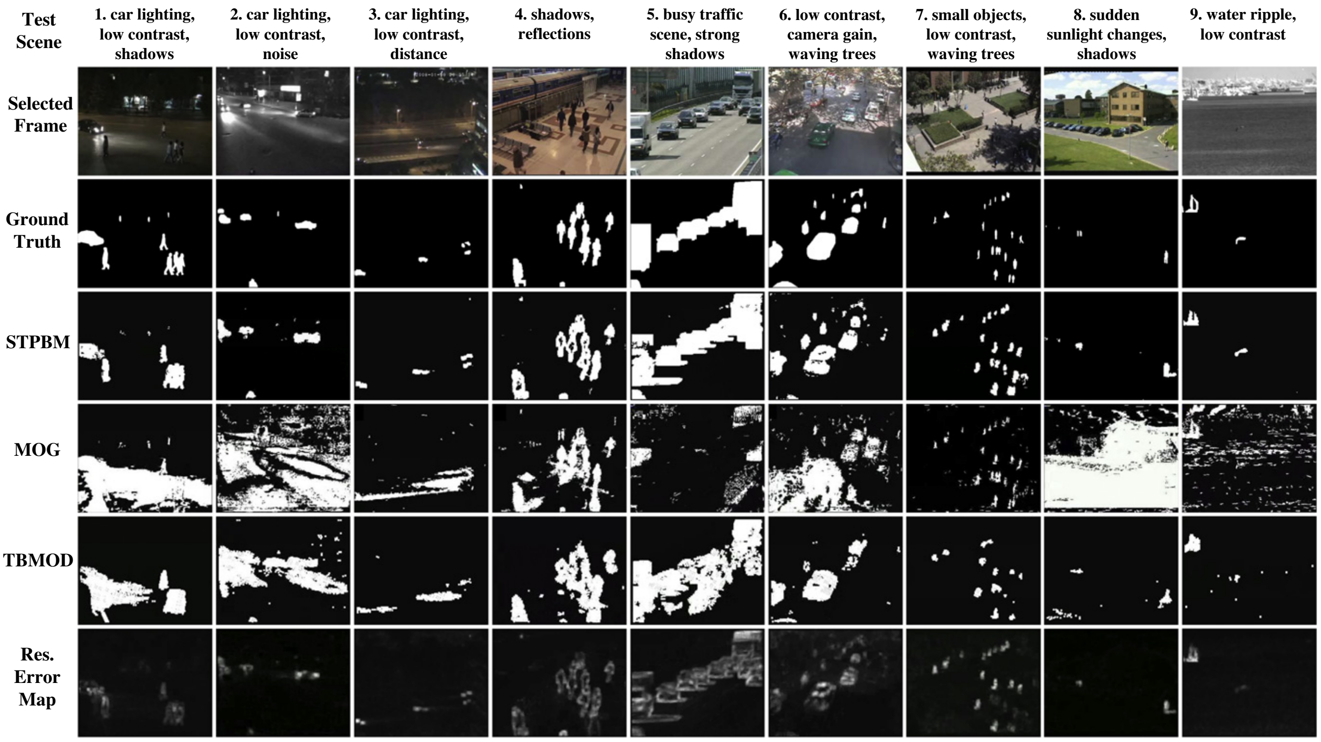

Test Scene
1. car lighting, low contrast, shadows
2. car lighting, low contrast, noise
3. car lighting, low contrast, distance
4. shadows, reflections
5. busy traffic scene, strong shadows
6. low contrast, camera gain, waving trees
7. small objects, low contrast, waving trees
8. sudden sunlight changes, shadows
9. water ripple, low contrast
Selected Frame
Ground Truth
STPBM
MOG
TBMOD
Res. Error Map

Fig. 11. Comparison results of our method (STPBM) with MOG (Stauffer et al., 2000) and TBMOD (Heikkila and Pietikainen, 2006) for nine typical scenes. The results of STPBM are obtained with the adaptive threshold in (8) based on the residual error maps in the bottom. The original resolution is 352 × 288 pixels.

Fig. 12 shows precision–recall of all scenes. **Recall** is the ratio of the number of true foreground pixels detected to the number of true foreground pixels. And **precision** is defined as the ratio of the number of true foreground pixels detected to the number of foreground pixels detected. The precision and recall of a perfect result are both 1. The results of STPBM distribute closer to the perfect point (1,1) than other two methods. While our method detects the moving objects in scenes 7 and 8 well in terms of object detection (in Fig. 11), it achieves low precisions (near 0.4) because the foreground objects in the two scenes are so small that the number of false alarm pixels around the objects are comparable to that of ground truth. Moreover, we consider the $F1$ metric, also known as Figure of Merit or F-measure, $F1 = 2 * \text{recall} * \text{precision}/(\text{recall} + \text{precision})$, that is the weighted harmonic mean of precision and recall (Maddalena and Petrosino, 2008a). Such measure allows to obtain a single measure to “rank”

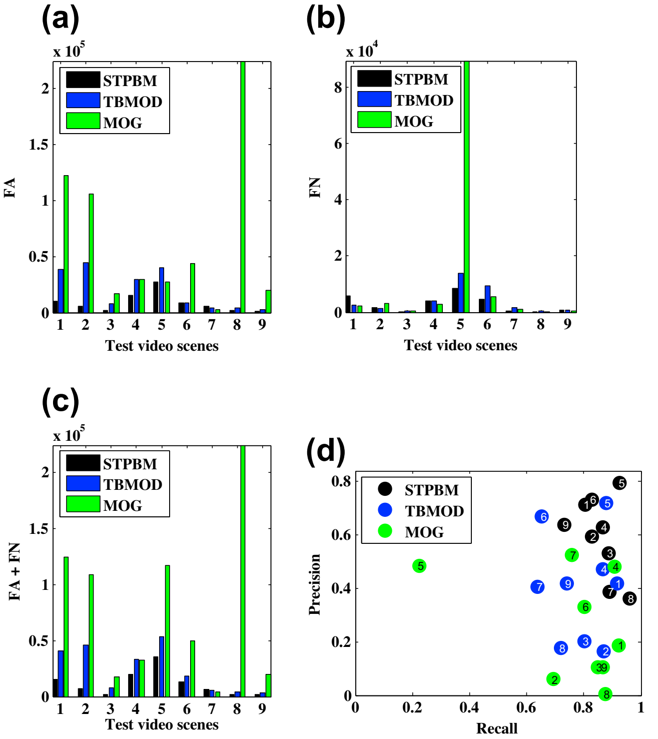


Fig. 12. Quantitative comparison results. The scene numbers correspond to Fig. 11. The vertical axis is the number of false alarm (FA) or false negative (FN) pixels. In the subfigure of precision–recall, a point stands for the result of one method in a scene and an ideal result should be 1 precision and 1 recall rate. The number in each marker corresponds to the scene index.

different methods. Results of three methods on all nine scenes are given in Table 2. Overall, our method outperforms the two compared ones in the test scenes.

As shown in Fig. 11, while our method is robust to waving trees to a certain degree (e.g., scenes 6 and 7), it fails in heavy waving trees in the wind in Fig. 13, which severely disobeys the original hypotheses that background regions were static except illumination changes. Fortunately, this problem could be alleviated by a simple preprocess, i.e., a smoothing operator. After using a Gaussian kernel of 13 × 13 pixels to pre-smooth the incoming frames, the small foreground objects are still salient and detected successfully, whereas most of the waving trees are suppressed. This is comparable to the results in (Heikkila and Pietikainen, 2006).

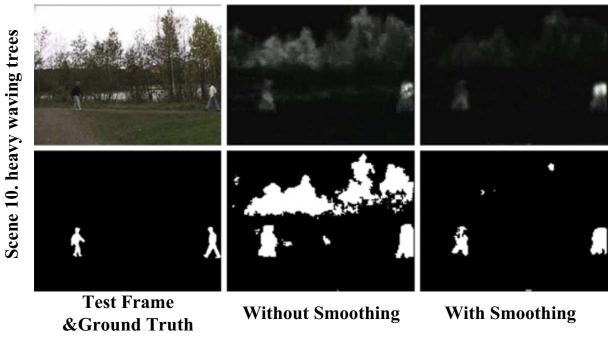


Fig. 13. Results of our method for a scene with heavy waving trees without and with pre-smoothing operator (using the same parameter setting). Residual error maps and foreground masks are shown in row 1 and 2, respectively.

Comparison with the state-of-the-art. We also compare our STPBM with a state-of-the-art method (Pilet et al., 2008), which achieves great progress in coping with sudden illumination changes, such as light switch. They present a novel Gaussian background model on the ratio of intensities between a stored background image and an input image, model the foreground object by a mixture of Gaussian and a uniform distribution on color values, and introduce a spatial prior of foreground/background utilizing spatial correlation and texture information. For each input pixel, its probabilities corresponding to the foreground and the background model are summed respectively. And the segmentation is made by thresholding the probability of background model. In this experiment, for convenient comparison, STPBM also simply thresholds the residual errors of all pixels and all fames by a single threshold rather than the adaptive threshold described in Section 3.3. For Pilet et al. method, besides using their default spatial prior distribution, we also learn a new prior on our own dataset (including the two test scenes in Fig. 15).

We first evaluate two methods with scene 1 and 2, which have clean and static background frames needed by Pilet et al. method. In order to avoid the influence of different thresholds, a series of thresholds are adopted for two methods. Fig. 14 shows the three ROC curves corresponding to STPBM and Pilet et al. method with new prior and default prior respectively. The blue circle on each curve is the cut-off point for best sensitivity and specificity. That is, at this point the algorithm achieves best balance between false alarms and missing detections. With the thresholds corresponding to the three cut-off points, the visual results of foreground detection are shown in Fig. 15. From these results it is apparent that STPBM detects the foreground objects successfully with a few false alarms. However, Pilet et al. method with new or default prior fails to distinguish the foreground objects due to heavy false alarms caused by drastic car lighting, low contrast and low SNR. The above

Table 2
Quantitative results of three methods on all nine scenes.

Methods	MoG			TBMOD			STPBM		
Num.	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
Scene1	0.19	0.94	0.31	0.42	0.92	0.58	0.72	0.83	**0.77**
Scene2	0.06	0.72	0.11	0.16	0.89	0.27	0.59	0.83	**0.69**
Scene3	0.11	0.85	0.19	0.20	0.80	0.32	0.53	0.89	**0.66**
Scene4	0.47	0.93	0.62	0.47	0.88	0.61	0.64	0.88	**0.74**
Scene5	0.48	0.24	0.32	0.71	0.89	0.79	0.79	0.94	**0.86**
Scene6	0.32	0.83	0.46	0.66	0.67	0.66	0.73	0.81	**0.77**
Scene7	0.50	0.79	**0.61**	0.40	0.64	0.50	0.39	0.89	0.54
Scene8	0.01	0.91	0.01	0.18	0.69	0.28	0.37	0.95	**0.54**
Scene9	0.08	0.89	0.15	0.42	0.76	0.54	0.65	0.75	**0.69**

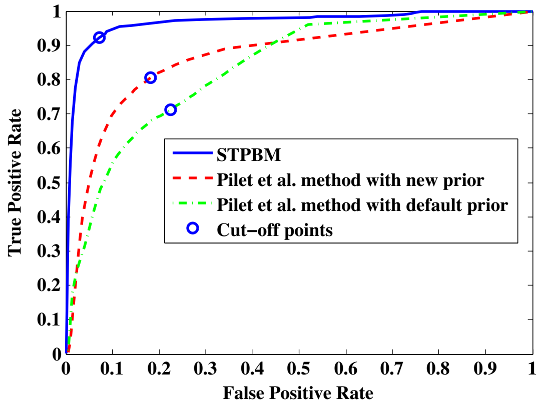


Fig. 14. Performance comparison between our STPBM and Pilet et al. method (Pilet et al., 2008). Three ROC curves are computed over the seven labeled representative frames in scenes 1 and 2. The cut-off point is position for best sensitivity and specificity. The visual results corresponding to the three cut-off points are shown in Fig. 15. Note that in this experiment our STPBM simply employs a single fixed thresholds for all pixels and all frames rather than the adaptive threshold setting as discussed in Section 3.3.

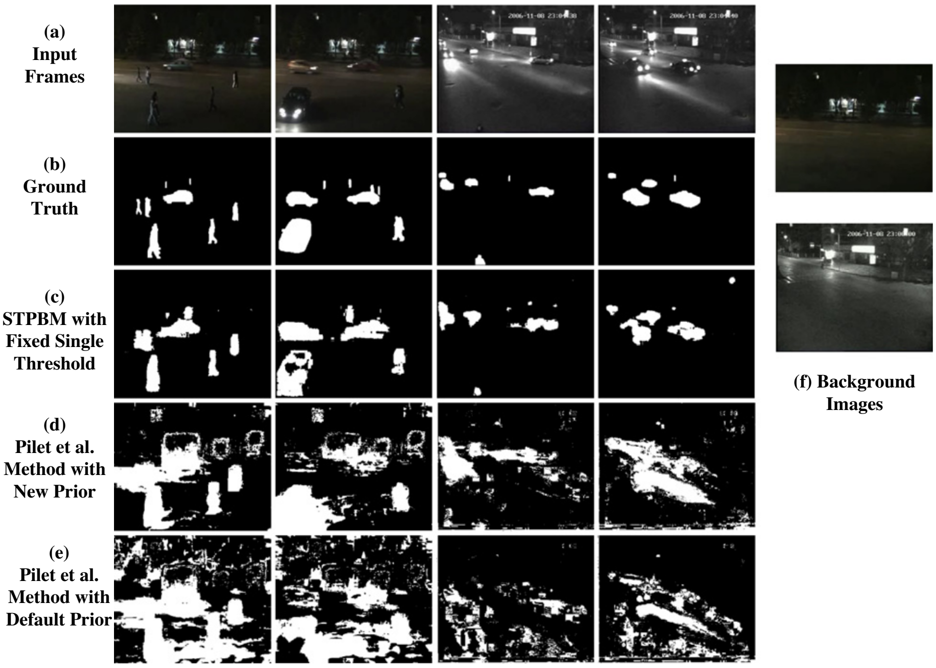


Fig. 15. Performance comparison between our STPBM and Pilet et al. method (Pilet et al., 2008). (a) Input frames (scene 1: 4485 and 4856 frames, and scene 2: 6970 and 7010 frames). (b) Ground truth. (c)(d)(e) are the results of STPBM with a single fixed threshold, Pilet et al. method with new priors and Pilet et al. method with default priors respectively, which correspond to the three cut-off points in Fig. 14. (f) The background models used by Pilet et al. method are the means of the first six frames in two scenes. The original resolution is 352 × 288 pixels.

quantitative and qualitative analysis demonstrates that our STPBM is superior to the state-of-the-art in handling extremely difficult real surveillance scenarios, e.g., nighttime traffic scenes.

In Fig. 16, we also compare two methods on two daytime scenes, namely, scenes 4 and 6. The corresponding quantitative results are listed in Table 3. In these two scenes, the clean and static background images are not available. Thus, we simply use the average images of the first five frames as the background images shown in Fig. 16. Note that there are some foreground objects in the "background images". The results of Pilet et al. method are computed with default prior. In these two daytime scenes, the performances of Pilet et al. method with default prior and new prior are similar. The results of our STPBM are the same as those shown in Fig. 11. We can find that in Scene 4 Pilet et al. method achieves satisfactory performance except the false alarms caused by the non-clean background images. In Scene 6, however, Pilet et al. method fail to detect most of foreground objects due to low contrast. Additionally, besides the false alarms caused by non-clean background images, there are many false alarms of small spots in the upper part of the scene.

Results on two public datasets. In Fig. 17, our method is evaluated on the **Wallflower dataset** (Toyama et al., 1999). The scene of "Waving Trees" is pre-smoothed. All the seven scenes use the same parameter setting in Table 1, though better results could be obtained by tuning the parameters carefully. In the "Light Switch" scene, the performance of our method is superior to those in (Heikkila and Pietikainen, 2006; Toyama et al., 1999; Pilet et al., 2008), and other results are comparable. There are often missing detections on the black flat areas of foreground objects, which is caused by the fact that the pure black bricks lie in any background subspaces. Fortunately, the moving contours of foreground objects are detected reliably, which is enough from the object detection point of view.

We also evaluate the proposed method on the **SABS dataset** (Brutzer et al., 2011). The SABS (Stuttgart Artificial Background Subtraction) dataset is an artificial dataset for pixel-wise evaluation

Table 3
Quantitative results of two methods on two daytime scenes.

Methods	Pilet et al. method			STPBM		
Num.	Prec.	Rec.	F_1	Prec.	Rec.	F_1
Scene 4	0.62	0.90	0.73	0.64	0.88	0.74
Scene 6	0.34	0.64	0.44	0.73	0.81	0.77

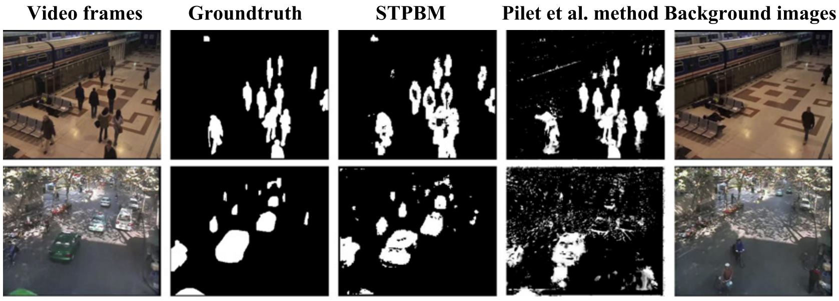


Fig. 16. Performance comparison between our STPBM and Pilet et al. method (Pilet et al., 2008) on two daytime scenes. The original resolution is 352 × 288 pixels.

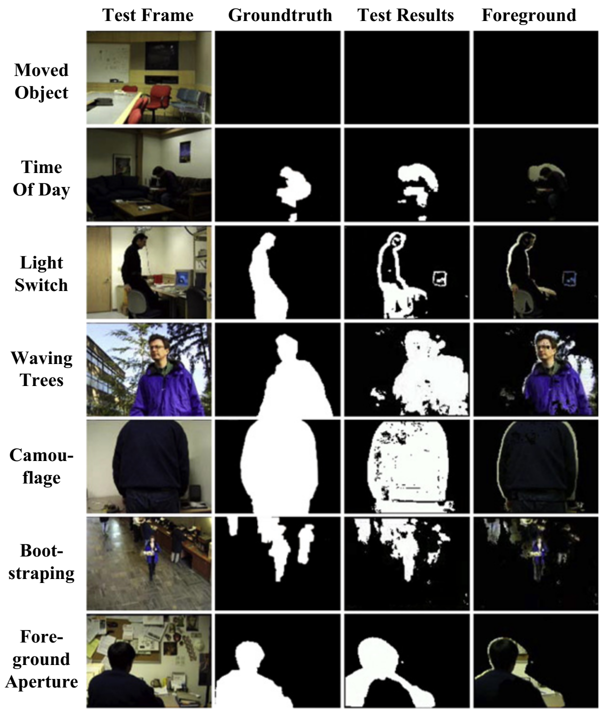


Fig. 17. Background modeling results of our method on Wallflower dataset (Toyama et al., 1999). The original resolution is 160 × 120 pixels.

of background models. In contrast to manually annotated ground-truth data, the SABS dataset does not suffer from imperfect labels or only a small number of annotated frames. The use of artificial data also makes it possible to separably judge the performance of background subtraction methods for typical challenges. For evaluation they consider the following typical challenges: gradual illumination changes, sudden illumination changes, dynamic background, camouflage, shadows, bootstrapping, and video noise. Nine different test scenarios are provided to cover these challenges, which are basic, dynamic background (DyBg), bootstrapping (Bts), darkening (Dark), light switch (LS), noisy night (NN), camouflage (Cam), and video compression (H264). The dataset consists of nine video sequences for the nine different scenarios. These sequences are further split into training and test data. For every frame of each test sequence ground-truth annotation is provided. The sequences have a resolution of 800 × 600 pixels and are captured from a fixed viewpoint. For some scenarios, a detail of a sequence is considered to focus on regions with high impact to a specific problem. (More details about the dataset and the evaluation process can be found in (Brutzer et al., 2011).)

Table 5
The parameter settings of the proposed method for the results in Table 4.

Scene(s)	β	d	α	λ	ζ	ρ	$T_{\min}$	T_{stay}
Noisy night (NN)	4	5	0.001	10	100	20	30	200
Other 8 scenes	4	5	0.001	10	100	20	16	200

The performance of background modeling methods are measured by the *F*-measure on pixel-level. Comparison results with other nine background modeling methods are listed in Table 4. The results of the nine compared methods directly come from the website of SABS dataset (http://www.vis.uni-stuttgart.de/index.php?id=sabs). For the proposed method, the parameter settings for the results in Table 4 are listed in Table 5. Note that all the test scenes share the same parameter setting except the minimum threshold $T_{\min}$ that is tuned to adapt the noise level of the test scenes. In addition, there is not any pre-processing step (e.g., pre-smoothing) adopted by the proposed method. Again, the false alarms of our method mainly occur around the bodies of the moving cars due to the use of spatio-temporal neighborhood information and most of the false negatives appear on the flat regions of the cars' bodies, as shown in Fig. 18. The best results of the proposed method on the bootstrapping (Bts), light switch (LS), and noisy night (NN) are consistent with the previous experiments and further demonstrate the pretty properties of the STPBM. The results on the dynamic background (DyBg) also show the ability of the proposed method to deal with dynamic background to a certain degree.

5. Discussion and conclusions

We present a novel method for background modeling by online subspace learning on spatio-temporal patches (or video bricks). The proposed method models the variations of video bricks via a low-dimensional background subspace in the high-dimensional brick space. Both off-line and on-line experimental results demonstrate the two conjectures on the space distribution of video bricks. In the proposed method there is no more extra post-processing except the pre-smoothing for scenes with very heavy waving trees. All the promising results are obtained by our simple and effective framework.

The proposed method has three major appealing properties:

(1) Robust to illumination changes. Due to the fact that all video bricks of a static background patch under arbitrary illumination conditions can be explained well by a low-dimensional background subspace that is learnt and updated by online subspace learning, our STPBM method is robust to various illumination changes in real surveillance scenario.

Table 4
Quantitative results of the proposed method (STPBM) and nine other methods (McFarlane and Schofield, 1995; Stauffer et al., 2000; Oliver et al., 2000; McKenna et al., 2000; Li et al., 2003; Kim et al., 2005; Zivkovic and van der Heijden, 2006; Maddalena and Petrosino, 2008b; Barnich and Vibe, 2009) on the SABS dataset (Brutzer et al., 2011).

Methods	Basic	DyBg	Bts	Dark	LS	NN	Cam	nCam	H264
STPBM	0.709	0.594	**0.714**	0.644	**0.552**	**0.593**	0.724	0.718	0.702
McFarlane	0.614	0.482	0.541	0.496	0.211	0.203	0.738	0.785	0.639
Stauffer	0.800	0.704	0.642	0.404	0.217	0.194	0.802	0.826	0.761
Oliver	0.635	0.552	–	0.300	0.198	0.213	0.802	0.824	0.669
McKenna	0.522	0.415	0.301	0.484	0.306	0.098	0.624	0.656	0.492
Li	0.766	0.641	0.678	0.704	0.316	0.047	0.768	0.803	0.773
Kim	0.582	0.341	0.318	0.342	–	–	0.776	0.801	0.551
Zivkovic	0.768	0.704	0.632	0.620	0.300	0.321	0.820	0.829	0.748
Maddalena	0.766	0.715	0.495	0.663	0.213	0.263	0.793	0.811	0.772
Barnich	0.761	0.711	0.685	0.678	0.268	0.271	0.741	0.799	0.774

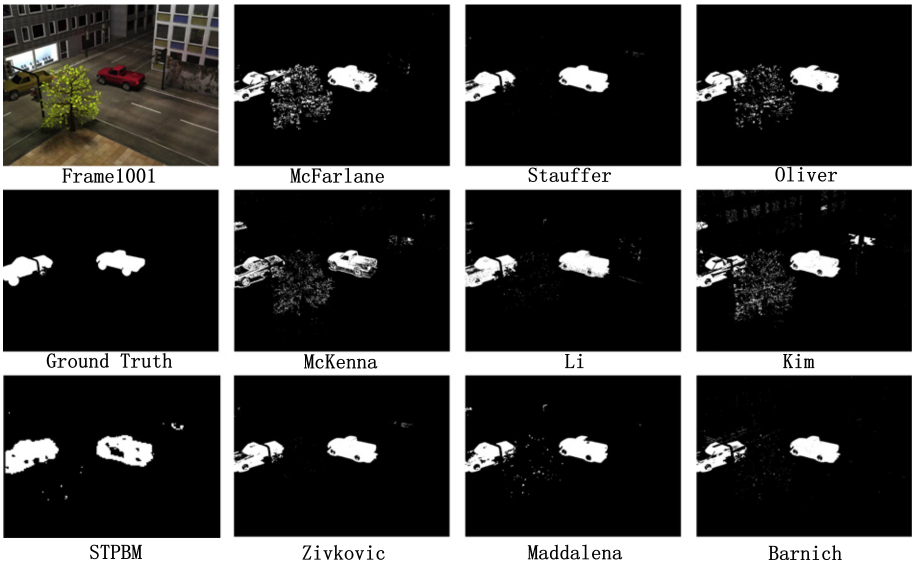


Fig. 18. Representative foreground masks at best (averaged) *F*-measure for gradual illumination change sequence. Ground truth is depicted including do not care boundary pixels. The original resolution is 800 × 600 pixels.

(2) Sensitive to dim moving objects. Since the basic processing unit "video brick" encodes appearance variations and motion dynamics jointly in a straightforward manner, our STPBM method is able to capture subtle variations of foreground objects in a low contrast environment effectively. It is this property that distinguishes the proposed STPBM method from the conventional block-based methods that often suffer from heavy missing detections in low contrast.

(3) Adapt busy scenes well and quickly. The proposed method is capable to adapt a challenging busy scene quickly and achieves satisfactory performance. This mainly should be due to two merits: (i) the online subspace learning method adopted can well capture the background subspace quickly and (ii) the background subspace adapted on-the-fly is not disrupted by outliers (e.g., bricks with foreground occlusion).

Moreover, from the systematic experimental results, we can find that the proposed method can deal with specular reflection, shadows, waving trees, water ripple, etc. to a certain degree, despite the violation of the original assumptions that background regions are static and have Lambertian surface. These properties make the proposed method be ready for many real scenes.

For dynamic background (e.g., heavy waving trees), a simple pre-smoothing operator is introduced. However, coping with the dynamic background is still an open problem in the proposed method. This is due to the fact that the proposed method is unimodal. The illumination changes of a static (or near static) background can be modeled well by one background subspace, whereas the appearance variations of dynamic backgrounds are difficult to model only by one background subspace. The algorithm also fails in the surface with strong specular reflection as it breaks the Lambertian assumption. This problem can be alleviated by using the geometry contextual information (Hu et al., 2008).

Acknowledgment

The authors thank M. Heikkilä and M. Pietikäinen for their source code of the TBMOD algorithm. The authors also would like to thank Dr. Tianfu Wu and Wenze Hu at UCLA for discussions. The main part of this work was done at the Lotus Hill Institute. This work was supported by the NSFC Grants 90920009 and 60832004.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2012.01.012.

References

Barnich, O., Vibe, M.V.D., 2009. A powerful random technique to estimate the background in video sequences. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing.

Basri, R., Jacobs, D.W., 2003. Lambertian reflectance and linear subspaces. IEEE Trans. Pattern Anal. Machine Intell. 25 (2), 218–233.

Belhumeur, P., Kriegman, D., 1998. What is the set of images of an object under all possible illumination conditions? Internat. J. Computer Vision 28, 245–260.

Bouwmans, T., 2009. Subspace learning for background modeling: a survey. Rec. Patents Comput. Sci. 2 (3), 223–234.

Brutzer, S., Hoferlin, B., Heidemann, G., 2011. Evaluation of background subtraction techniques for video surveillance. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S., Duraiswami, R., Harwood, D., 2002. Background and foreground modeling using nonparametric kernel density for visual surveillance. In: Proc. IEEE, pp. 1151–1163.

Elhabian, S., El-Sayed, K., Ahmed, S., 2008. Moving object detection in spatial domain using background removal techniques – state-of-art. Rec. Patents Comput. Sci. 1, 32–54.

Finlayson, G., Hordley, S., Lu, C., Drew, M., 2006. On the removal of shadows from images. IEEE Trans. Pattern Anal. Machine Intell. 28 (1), 59–68.

Garg, R., Du, H., Seitz, S.M., Snavely, N., 2009. The dimensionality of scene appearance. In: IEEE Internat. Conf. Computer Vision.

Grabner, H., Bischof, H., 2006. On-line boosting and vision. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Heikkila, M., Pietikainen, M., 2006. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Machine Intell. 28 (4), 657–662.

Hu, W., Gong, H., Zhu, S.-C., Wang, Y., 2008. An integrated background model for video surveillance based on primal sketch and 3d scene geometry. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Huang, J.-B., Chen, C.-S., 2009. Moving cast shadow detection using physics-based features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Ivanov, Y., Bobick, A., Liu, J., 2000. Fast lighting independent background subtraction. Internat. J. Computer Vision 37, 199–207.

Jabri, S., Duric, Z., Wechsler, H., Rosenfeld, A., 2000. Detection and location of people in video images using adaptive fusion of color and edge information. In: Proc. Internat. Conf. on Pattern Recognition.

Kaewtrakulpong, P., Bowden, R., 2001. An improved adaptive background mixture model for realtime tracking with shadow detection. In: Proc. of Advanced Video Based Surveillance Systems.

Kim, K., Chalidabhongse, T., Harwood, D., Davis, L., 2005. Real-time foreground-background segmentation using codebook model. Real Time Imaging 11 (3), 172–185.

Lee, D., 2005. Effective gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Machine Intell. 27 (5), 827–832.

Li, L., Huang, W., Gu, I., Tian, Q., 2003. Foreground object detection from videos containing complex background. In: ACM Internat. Conf. on Multimedia.

Li, Y., 2004. On incremental and robust subspace learning. Pattern Recognition 37, 1509–1518.

Lim, S.-N., Mittal, A., Davis, L., Paragios, N., 2005. Fast illumination-invariant background subtraction using two views: error analysis, sensor placement and applications. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 1071–1078.

Lin, H.H., Liu, Y.L., Chuang, J.H., 2009. Learning a scene background model via classification. IEEE Trans. on Signal Process. 57 (5), 1641–1654.

Maddalena, L., Petrosino, A., 2008a. A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans. Image Process. 17 (7), 1168–1177.

Maddalena, L., Petrosino, A., 2008b. A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans. Image Process. 17 (7), 1168–1177.

McFarlane, N., Schofield, C., 1995. Segmentation and tracking of piglets in images. Machine Vision Appl. 8 (3), 187C193.

McKenna, S.J., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H., 2000. Tracking groups of people. Computer Image and Vision Understanding 80 (1), 42C56.

Messelodi, S., Modena, C.M., Segata, N., Zanin, M., 2005. A kalman filter based background updating algorithm robust to sharp illumination changes. In: Internat. Conf. on Image Analysis and Processing.

Monnet, A., Mittal, A., Paragios, N., Ramesh, V., 2003. Background modeling and subtraction of dynamic scenes. In: Proc. IEEE Internat. Conf. on Computer Vision, pp. 1305–1312.

Noriega, P., Bernier, O., 2006. Real time illumination invariant background subtraction using local kernel histograms. In: British Machine Vision Conf., vol. 1, pp. 1071–1078.

Oliver, N.M., Rosario, B., Pentland, A.P., 2000. A bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Machine Intell. 22, 831–843.

Patwardhan, K.A., Sapiro, G., Morellas, V., 2008. Robust foreground detection in video using pixel layers. IEEE Trans. Pattern Anal. Machine Intell. 30 (4), 746–751.

Piccardi, M., 2004. Background subtraction techniques: a review. In: Proc. IEEE Internat. Conf. on Systems, Man and Cybernetics.

Pilet, J., Strecha, C., Fua, P., 2008. Making background subtraction robust to sudden illumination changes. In: European Conf. on Computer Vision.

Pless, R., 2005. Spatio-temporal background models for outdoor surveillance. J. Appl. Signal Process.

Ross, D., Lim, J., Lin, R.-S., Yang, M.-H., 2008. Incremental learning for robust visual tracking. Internat. J. Computer Vision 25 (8), 1034–1040.

Seki, M., Wada, T., Fujiwara, H., Sumi, K., 2003. Background subtraction based on cooccurrence of image variations. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 65–72.

Skočaj, D., Leonardis, A., 2008. Incremental and robust learning of subspace representations. Image Vision Comput. 26 (1), 27–38.

Skočaj, D., Leonardis, A., Bischof, H., 2007. Weighted and robust learning of subspace representations. Pattern Recognition 40 (5), 1556–1569.

Stauffer, C., Eric, W., Grimson, W.E.L., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Machine Intell. 22, 747–757.

Toyama, K., Krumm, J., Brumitt, B., Meyers, B., 1999. Wallflower: principles and practice of background maintenance. In: Proc. IEEE Internat. Conf. on Computer Vision.

Wang, L., Wang, L., Wen, M., Zhuo, Q., Wang, W., 2007. Background subtraction using incremental subspace learning. In: Proc. IEEE Internat. Conf. on Image Processing.

Wang, Y., Loe, K.-F., Wu, J.-K., 2006. A dynamic conditional random field model for foreground and shadow segmentation. IEEE Trans. Pattern Anal. Machine Intell. 28 (2), 279–289.

Weng, J., Zhang, Y., Hwang, W., 2003. Candid covariance-free incremental principal components analysis. IEEE Trans. Pattern Anal. Machine Intell. 25 (8), 1034–1040.

Wixson, L., 2000. Detecting salient motion by accumulating directionally consistent flow. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 774–780.

Wren, C., Azarbayejani, A., Darrell, T., Pentland, A., 1997. Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Anal. Machine Intell. 19, 780–785.

Yang, Y.-H., Levine, M.D., 1992. The background primal sketch: an approach for tracking moving objects. Machine Vision Appl. 5, 17–34.

Yao, J., Odobez, J., 2007. Multi-layer background subtraction based on color and texture. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Yao, Z., Yang, X., Zhu, S.C., 2007. Introduction to a large scale general purpose groundtruth database: methodology, annotation tools, and benchmarks. In: Proc. Internat. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition.

Zhao, Y., Gong, H., Lin, L., Jia, Y., 2008. Spatio-temporal patches for night background modeling by subspace learning. In: Proc. Internat. Conf. on Pattern Recognition.

Zivkovic, Z., van der Heijden, F., 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Lett. 27, 773–780.