Spatio-Temporal Patches for Night Background Modeling by Subspace Learning

Youdong Zhao[1, 2], Haifeng Gong[2], Liang Lin[1, 2] and Yunde Jia[1]
[1]*Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081 PRC*
[2]*Lotus Hill Research Institute, China*
{zyd458, jiayunde, linliang}@bit.edu.cn, hfgong@lotushill.org

Abstract

In this paper, a novel background model on spatio-temporal patches is introduced for video surveillance, especially for night outdoor scene, where extreme lighting conditions often cause troubles. The spatio-temporal patch, called brick, is presented to simultaneously capture spatio-temporal information in surveillance video. The set of bricks of a given background patch, under all possible lighting conditions, lies in a low-dimensional subspace, which can be learned by online subspace learning. The proposed method can efficiently model the background and detect the appearance and motion variance caused by foreground. Experimental results on real data show that the proposed method is insensitive to dramatic lighting changes and achieves superior performance to two classical methods.

1. Introduction

Background modeling plays a key role in video surveillance. In the past decade, various background modeling algorithms are developed and achieve great progress [4]. However, there still remains a bottleneck for system performance for night outdoor scene (Fig.1), where faint lighting, low SNR, low contrast, dramatic illumination changes, etc., all can cause troubles.

The traditional *pixel level* methods [4, 7] cannot cope with global changes in the scenes, since they model the background as a set of independent pixel processes. Recently, there is a tendency to use the neighborhood information of a pixel to improve the performance of background modeling, i.e., spatial neighborhood (block-based) [2, 3, 4 and 8] or temporal neighborhood (optical flow based) [10]. Both these methods have been used with success. However, in a night outdoor scene, the block-based methods often suffer from either heavy false alarm due to dramatic lighting changes or missing detection as the foreground

Fig. 1. Challenging scene examples(refer to Fig.6)

is very similar to the background in local due to low contrast; the estimation of optical flow frame-to-frame also faces troubles due to the dramatic illumination changes, e.g., occasional car lighting.

Intuitively, spatial neighborhood information and temporal neighborhood information are complementary to each other. For example, in a faint lighting and low contrast environment, the motion of foreground supplies the most visual information; whereas when there is a dramatic illumination change, the appearance of foreground gives us the main visual information.

Motivated by above observations, we propose a novel background modeling method in *brick level*, which simultaneously utilizes *spatio-temporal information* to improve performance. In the proposed method, the brick is the fundamental atom of processing. Unlike the block or optical flow feature, brick can capture the appearance information in the spatial domain and the motion information in the temporal domain at the same time. Based on the bricks, the background models are learned by an online subspace learning method, i.e., CCIPCA algorithm [9], which is fast in convergence rate and low in computational complexity. The proposed method has three major properties: 1) It utilizes *spatio-temporal information* for background modeling; 2) It is insensitive to dramatic lighting changes in night outdoor scenes; 3) It is able to detect the non-salient appearance variance of foreground due to low contrast.

Some researchers [5] also use the spatio-temporal information for background modeling. However, their consideration is the problem of developing background

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

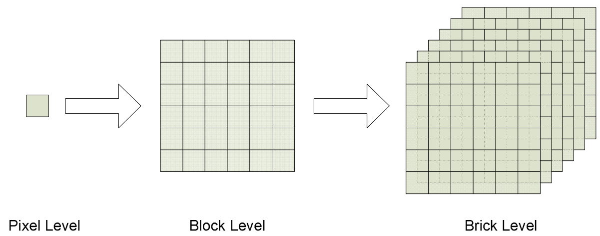

Fig. 2. The three levels of background modeling

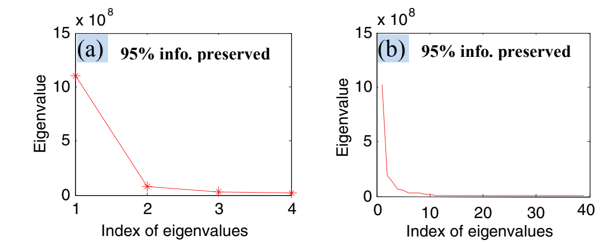

Fig. 3. Eigenvalue curves of different state bricks, (a) background, (b) foreground occlusion

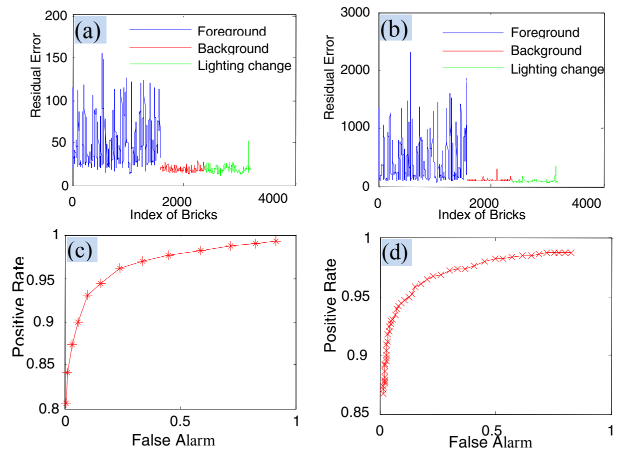

Fig. 4. Reconstruction error curves of different types of bricks and the corresponding ROC curves

models for scenes with consistent background motion, e.g., trees waving in the wind. And they build the background models based on the responses to spatio-temporal derivative filters at each pixel in the video sequence.

Compared with classical GMM [7] and LBP [3] background modeling, the experimental results on real data demonstrate superior performance of the proposed method for night outdoor scenes.

The rest of the paper is organized as follows. In Section 2, the definition and properties of brick are presented. The proposed method is discussed in Section 3. Experimental results and conclusions are given in Section 4 and 5 respectively.

2. Brick analysis

A *brick*, as shown in Fig.2, is a small *spatio-temporal patch* (e.g., $15 \times 15 \times 7$) within a video sequence [6]. Given a video sequence, a set of brick sequences are obtained. Each brick sequence corresponds to a fixed patch within the video frame.

It is well-known that the set of *n*-pixel images of an object, under all possible lighting conditions, lies in a low-dimensional subspace in $\mathbb{R}^n$ image space [1]. In a night outdoor scene (Fig.1), there are mainly three states for a brick, i.e., normal background bricks, bricks with illumination changes and bricks with foreground occlusion. Here, we make two assumptions about the brick distribution.

Assumption 1: *The set of bricks of a given background patch under various lighting conditions lies in a low-dimensional manifold (background subspace) embedded in the high-dimensional brick space.*

Assumption 2: *Due to the randomness of foreground, the bricks with foreground occlusion are uniformly distributed in the high-dimensional brick space and can be well separated from the background subspace.*

The approximate subspace can be captured by online subspace learning, as there are rich images under various lighting conditions in surveillance application. Two experiments are introduced to demonstrate the above assumptions. Without loss of generality, we randomly select a $15 \times 15 \times 7$ patch as shown in Fig.1 and the first 20,000 bricks containing various changes are obtained. The brick distribution is offline analyzed by batch PCA using supervised information, i.e., the bricks are manually divided into three subsets w.r.t. the three states and analyzed separately.

Fig.3 shows the curves of eigenvalues learned from different subsets. Because the number of normal background bricks is larger than those with lighting changes, when analyzing part of normal background bricks and all the bricks with lighting changes are used for learning in Fig.3a. In Fig.3b, the training samples are those with only foreground occlusion. From these results, we can see that background bricks (with only lighting changes) do lie in a low-dimensional subspace, whereas the fore-ground bricks distribute in a high-dimensional space.

Fig.4 illustrates the curves of reconstruction errors and the corresponding ROC curves. We use part of normal background bricks to learn a subspace. And the rest of background (red), part of the lighting change bricks (green) and all foreground bricks (blue) are projected into the subspace. The residual errors are shown in Fig.4a. In Fig.4b, part of bricks with lighting changes is used for training, the rest together with part of normal background and all foreground for testing. Fig.4c and 4d give the corresponding ROC curves. These results demonstrate that the normal background bricks and the bricks with lighting changes lie in the same subspace, from which the foreground bricks can be well separated.

3. Background modeling on bricks

In this paper, we solve the problem of background modeling by online subspace learning.

3.1. Problem formulation

When a new frame $I_n (n = 1,2,\dots)$ comes (its size is $W \times H$), we divide it into $N = (W \times H)/(h \times h)$ patches $\{P_{i,n}\}_{i=1}^{N}$, where i is the patch index and h is the height and width of the patch (suppose that W and H can be divided exactly by h). For each patch $P_{i,n}$, we combine it with the corresponding previous $t-1$ patches (e.g., $t = 5$) to form a brick (Fig.2), whose size is $h \times h \times t$. Then we will obtain a set of brick sequence $\{B_i\}_{i=1}^{N}$, where $B_i = \{B_{i,1}, B_{i,2}, \dots, B_{i,n}, \dots\}$.

In the i^{th}brick sequence (e.g., corresponding to the patch shown by red rectangle in Fig.1), there are mainly two types of changes, i.e., illumination changes or foreground occlusion. The bricks with only lighting changes are the background bricks and lie in a low-dimensional subspace S_i, which can be learned by an online subspace learning method. Here, we reshape the brick $B_{i,n}$ into a D-dimensional ($D = h \times h \times t$) column vector $x_{i,n}$ (mean removed).

3.2. Online subspace learning

In this paper, we adopt a fast incremental principal component analysis algorithm, candid covariance-free IPCA (CCIPCA) [9], to compute the principal components of a brick sequence, i.e., background model. The CCIPCA algorithm is fast in convergence rate and low in computational complexity. In our implementation, the algorithm is slightly modified in initialization, learning rate setting and updating scheme.

Given a brick sequence $B = \{x_1, x_2, \dots, x_n, \dots\}$, where x_n (for succinctness of notation, we drop the patch index i in the suffix) is the n^{th}brick vector, the first d dominant eigenvectors are estimated recursively by following two equations:

$$v_{k,n} = \frac{n-1-l}{n} v_{k,n-1} + \frac{1+l}{n} x_{k,n} \langle x_{k,n}, \frac{v_{k,n-1}}{\|v_{k,n-1}\|} \rangle, \quad (1)$$

$$x_{k+1,n} = x_{k,n} - \langle x_{k,n}, \frac{v_{k,n}}{\|v_{k,n}\|} \rangle \frac{v_{k,n}}{\|v_{k,n}\|}, \quad (2)$$

where $\langle \cdot,\cdot \rangle$ denotes inner product, $v_{k,n}$ is the $k^{th}(1 \le k \le d)$ eigenvector updated by the n^{th} brick vector, $x_{1,n} = x_n$, $x_{k+1,n}$ is the residual brick vector after being projected onto the first k estimated eigenvectors and $\frac{1+l}{n}$ is the learning rate. The authors proved that, with the algorithm given by Eqs. (1) and (2), $v_{k,n} \to \pm\lambda_k q_k$ when $n \to \infty$, where λ_k is the k^{th} largest eigenvalues of covariance matrix of B and q_k is the corresponding unit eigenvector [9]. In addition, the mean μ_n is updated by:

$$\mu_n = \frac{n-1-l}{n} \mu_{n-1} + \frac{1+l}{n} x_n. \quad (3)$$

In the implementation, a batch PCA algorithm is first performed on first few (e.g., 200) bricks to initialize the CCIPCA algorithm. In order to adapt the variance of new frames more efficiently, the learning rate α is set with a fixed value (e.g., 0.005), no longer correlating to n as in Eqs. (1) and (3). The new updating equations are given by Eqs. (4) and (5).

$$v_{k,n} = (1-\alpha) v_{k,n-1} + \alpha x_{k,n} \langle x_{k,n}, \frac{v_{k,n-1}}{\|v_{k,n-1}\|} \rangle \quad (4)$$

$$\mu_n = (1-\alpha)\mu_{n-1} + \alpha x_n \quad (5)$$

The incoming bricks do not always belong to the background subspace, so it is necessary to avoid using these to disturb the model. When a new brick $x_n$arrives, the distance L between x_n and the subspace S is recursively computed as follows,

$$x_{k+1,n} = x_{k,n} - \langle x_{k,n}, q_{k,n-1} \rangle \, q_{k,n-1}, \ (k = 1,2,\dots,d) \quad (6)$$

$$L(x_n, S) = \|x_{d+1,n}\|, \quad (7)$$

where $q_{k,n-1} = \frac{v_{k,n-1}}{\|v_{k,n-1}\|}$ is the k^{th} unit eigenvector of the subspace S. If L is less than a threshold T, x_n is marked as background and will be used to update the subspace S, otherwise foreground. Note that the distance L in Eqs. (7) is just the residual error of x_n. But it cannot be computed as traditional manner, i.e.,

$$L(x_n, S) = \|x_n - \textstyle\sum_{k=1}^{d} q_{k,n-1} \langle x_n, q_{k,n-1} \rangle\|, \quad (8)$$

since the eigenvectors $\{q_{k,n-1}\}_{k=1}^{d}$ usually are not orthogonal to each other strictly [9].

Fig.5. Comparison of performances between *patch level* (2nd column) and *brick level* (3rd column)

4. Experiments

We evaluate the proposed algorithm on a number of night outdoor scenes, collected by LHI [11], and the performance is satisfactory. The frame size of the video sequence is 384×288. The frame rate is 25fps. We first show the efficiencies of the *brick* in capturing the appearance and motion changes comparing to *patch level* and then compare the performances with two classical approaches, i.e., GMM [7] and LBP [3].

We first convert the color frame to gray, and then subsample it to a 96×72 small image. From the small image, we extract a patch of 4×4 for each pixel and use them to construct brick. In this case, the context information around each pixel is also being considered. Fig.5 shows the results of background modeling in *patch level* ($4 \times 4 \times 1$) and *brick level* ($4 \times 4 \times 5$) respectively. Note that in *patch level* there are heavy missing detections due to lack of the information of brick in temporal domain. In the follow experiments, the brick size is empirically set to be $4 \times 4 \times 5$.

Fig.6 shows some results of background modeling with a long video sequence. The dimension of subspace d is set to be 8. The learning rate α and matching threshold T (for all subspaces we simply set the same threshold) in our method is 0.0005 and 50. For GMM [7] and LBP [3], their parameters are all set with proper values. From the results, we can see that GMM performs poorest when there are dramatic illumination changes. While LBP operators [3] can tolerate some illumination changes, it fails with extreme lighting changes on the road. The proposed method achieves the best performance, although it seems to be sensitive to the tiny changes in the roadside beyond the street. In Fig.7, more comparison results on another scene are given. Note that both missing detection and false alarm exist in the results of LBP, whereas our method shows ability of tolerating extreme illumination changes and being sensitive to the appearance variance caused by foreground, though there are some "trails".

The proposed method in above experiments runs in about 15fps on a standard PC (Inter Pentium 2.8G with 1GB RAM). Using some sophistical schemes, e.g., updating part of all the models in turn with each new frame, the frame rate can be improved to 40fps with performance reduced slightly.

5. Conclusions

We present a novel method for background modeling in *brick level*, which utilizes *space-time information* simultaneously, is insensitive to dramatic lighting changes and can capture the tiny appearance and motion variance of foreground efficiently. The algorithm also has some limitations. For example, it fails in the surface with specular reflection, e.g., marble wall. This problem can be solved by using the geometry context information [4] which can also speed up the algorithm.

Acknowledgements

The authors would like to thank Tianfu Wu and Wenze Hu for stimulating discussions. This work is done at Lotus Hill Institute and is partially supported by the NSFC (NO. 60675021).

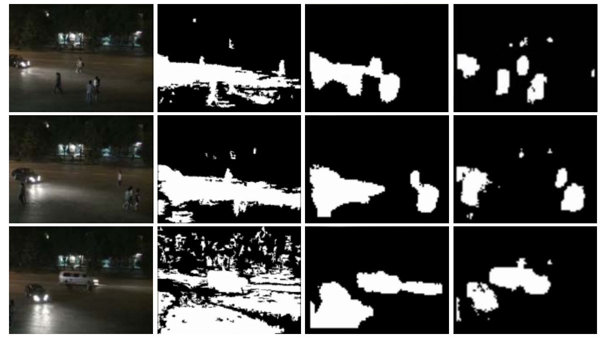

Fig.6. Performances comparison among GMM, LBP and the proposed method

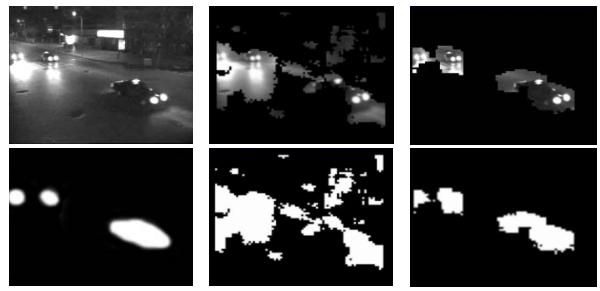

Fig.7. Comparison between LBP (2nd column) and the proposed method (3rd column) with Ground truth (bottom-left)

References

[1] P. Belhumeur and D. Kriegman. What's the set of images of an object under all possible illumination conditions? *Int. J. Computer Vision*, 28(3):1-16, 1998.

[2] M. Cristani, M. Bicego and V. Murino. Integrated region- and pixel-based approach to background modelling. *MOTION*'2002.

[3] M. Heikkil and M. Pietik. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. PAMI* , 28(4):657-661, Apr. 2006.

[4] W. Z. Hu, H. F. Gong, S. C. Zhu and Y. T. Wang. An integrated background model for video surveillance based on primal sketch and 3D scene geometry. *CVPR*'2008.

[5] R. Pless. Spatio-temporal background models for outdoor surveillance. *Journal on Applied Signal Processing*, pp. 2281–2291, 2005.

[6] E. Shechtman and M. Irani. Space-time behavior-based correlation. *IEEE Trans. PAMI*, 29(11):2045-2056, 2007.

[7] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*'1999.

[8] Lu Wang, L. Wang, M. Wen, Q. Zhuo, and W. Wang. Background subtraction using incremental subspace learning. *ICIP*'2007.

[9] J. Weng, Y. Zhang and W. Hwang. Candid covariance-free incremental principal components analysis. *IEEE Trans. PAMI* , 25(8):1034-1040, Aug. 2003.

[10] L. Wixson. Detecting salient motion by accumulating directionally consistent flow. *IEEE Trans. PAMI*, 22(8):774-780, Aug. 2000.

[11] Z. Yao, X. Yang and S. C. Zhu. Introduction to a large scale general purpose groundtruth database: methodology, annotation tools, and benchmarks. *EMMCVPR*'2007.