Texture and Motion Pattern Fusion for Background Subtraction

Bineng Zhong[1], Xiaopeng Hong[1], Hongxun Yao[1], Shiguang Shan[2,3], Xilin Chen[2,3], Wen Gao[1,2,4]

1Department of Computer Science and Engineering, Harbin Institute of Technology,
2Digital Media Research Center, Institute of Computing Tech., CAS, Beijing, China
3Key Laboratory of Intelligent Information Processing, CAS, Beijing,China
4Digital Media Institute, Peking University, Beijing, China
{bnzhong, xphong, yhx}@vilab.hit.edu.cn; {sgshan, xlchen, wgao}@jdl.ac.cn

Abstract

In this paper, we propose a novel background subtraction algorithm, which takes both texture and motion information into account. Texture information is represented by local binary pattern (LBP), which is tolerant of illumination changes and is computational simplicity. Assuming that there is significant structure in the correlations between observations across time, we propose a novel operator to extract motion information. Then, each pixel is modeled as a group of texture pattern histograms and motion pattern histograms respectively. Finally, we combine the texture pattern-based and motion pattern-based background model. Experimental results on challenging videos demonstrate the robustness and effectiveness of the proposed method.

Keywords: Background subtraction, texture pattern, motion pattern

1. Introduction

With the increasing demands of security, automated visual surveillance has become a hot topic. For surveillance video captured by static camera, the first sub-problem is background subtraction to detect moving objects in scene. However, background subtraction turns out to be a challenging problem due to many factors, such as color distortion, shadow caused by moving object, illumination variation and scene motion (e.g. wave).

Aiming at these problems, two types of approaches have been proposed: pixel-wise methods and hybrid methods.

Most of the pixel-wise methods, e.g., temporal difference and median filtering, assume that the observation sequence of each pixel is independent to each other and background scene is static. In the famous method [1], pixel in the scene is modeled as a Mixture of Gaussian. Elgammal utilizes a general nonparametric kernel density estimation technique for building a statistical representation of the scene background [2]. A non-statistical clustering technique to construct a background model is presented in [3]. The background is encoded on a pixel-by-pixel basis and samples at each pixel are clustered into the set of codewords. Although the above background subtraction methods have significantly different modeling schemes, most of them use standard color or intensity information to differ foreground from background, which is a strict assumption and limit their application in dynamic environment. Most of the dynamic scenes exhibit persistent motion characteristics. Therefore, a natural approach to model their behavior is via motion information (i.e. optical flow). In [4], both temporal and spatial information are exploited to improve background subtraction results. Mittal and Paragios [5] use the most recent T frames to build a non-parametric model of color and optical flow. While their approach still views the image as a set of independent pixels, they produce impressive results when the same motions are observed many times in every block of T frames. These pixel-wise algorithms can effectively adapt to smooth behaviors and gradual variations in the background. However, there are still some problems which lead to poor performance when infrequent motions occur, such as trees rustling periodically (but not constantly) due to wind gusts.

Hybrid methods take the correlation between pixels in the spatial vicinity into account. In [6, 7], background modeling is fulfilled in a jointly pixel, region and frame level. The textured-based method proposed in [8] forms an elegant combination of the pixel-wise and region-wise algorithms. Local binary patterns (LBP) are exploited as features and then each pixel is represented as a statistical model of a large region around it over time series. It outperforms the traditional pixel-based methods. Tian [9] integrates intensity and texture information into the GMM model to remove shadows and to enable the algorithm working for quick lighting changes. In [10], scene is coarsely represented as the union of pixel layers and foreground objects are detected by propagating these layers using a maximum-likelihood assignment. However, the limitations of the method are high-computational complexity and the requirement of an extra offline training step. Please refer to [11] for a more complete background subtraction methods review.

In this paper, we endorse the necessity to exploit motion information and thus extend Heikkila's algorithm [8] into the spatial-temporal domain in feature level. The overview of our algorithm is shown in Fig.1. First, texture and motion feature maps are extracted using corresponding operators. Second, for each pixel we compute two histograms over the region surrounding it in these two feature maps respectively. Third, two statistical model are obtained over time, with K most representative histograms respectively. Finally, they are combined in a natural fashion, also with a modeling update mechanism. When a new frame is coming, model subtraction is executed to make a decision. Experimental results indicate that the proposed method is effective for dynamic background modeling and outperforms the work by [1] and [8].

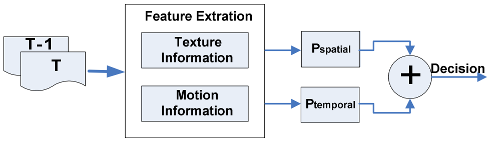


Fig.1. Illustration of the background subtraction framework based on texture and motion pattern.

The remainder of paper is organized as follows. In section 2, we introduce the spatial and temporal operator to extract the texture and motion pattern respectively. The background modeling procedure is described in section 3. In section 4, experimental results are given. Finally, conclusion and future work are drawn in section 5.

2. Texture and Motion Pattern

To capture the texture and motion pattern, a spatial and a temporal operator are introduced in this section, as shown in Fig.2. Here, we denote O_S and O_T for them respectively. Given the frame with gray level at time T by $F_T(x)$, we calculate the spatial feature map $f_{S_T}(x)$ and the temporal feature map $f_{m_T}(x)$ through the O_S and O_T separately, as shown in Fig.3.

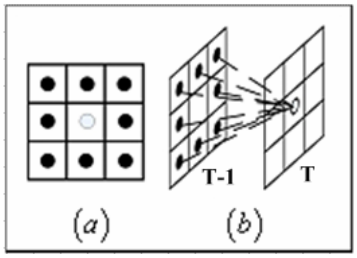


Fig.2. (a) Spatial operator. (b) Temporal operator.

2.1. Texture Pattern with Spatial Operator

Similar to [8], we adopt ordinary LBP as the spatial operator O_S (see Fig. 2(a)), because it is simple, effective, and what is more, invariant to gray-scale changes. Please refer to [8] for a detailed description of LBP.

2.2. Motion Pattern with Temporal Operator

As mentioned above, the motion information is very important for background modeling. Here, we introduce a new temporal operator O_T to obtain the motion pattern, as shown in Fig. 2(b). The motion pattern extracted in the pixel x at time T can be calculated as Eq.1:

$$f_T(x) = \sum_{p=0}^{7} bt_p(x) 2^p \qquad (1)$$

The function $bt_p(x)$ keeps the sign of the difference between the central pixel x at time T and its neighboring pixel x_p in previous T-1th frame. It is formularized as follows:

$$bt_p(x) = \begin{cases} 1, I_{T-1}(x_p) \geq I_T(x) \\ 0, else \end{cases} \qquad (2)$$

, where $I_T(x)$ is the intensity value in the pixel x at time T.

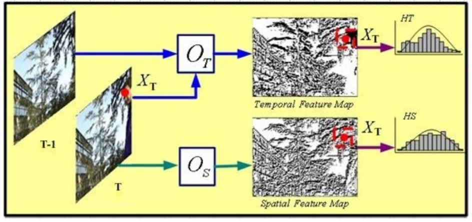


Fig.3. Illustration of the texture and motion pattern histograms computed over a rectangular region.

2.3. Discussion

The temporal operator O_T in Section 2.2 is somewhat inspired by volume local binary patterns (VLBP) and local binary patterns on three orthogonal planes (LBP-TOP) proposed by [12], which is quite similar to our approach at the first glance. However, they are different as follows: (1) VLBP and LBP-TOP combine the spatial and temporal texture patterns in a feature level, thus it forms a very high dimensional feature vector (even the dimensionality of simplified LBP-TOP is 3×2^8), which may cause high-computation complexity. In contrast, the operators used in our paper maintain the motion pattern separately to the texture pattern. This leads to a much lower dimensional representation which is suitable for on-line learning tasks and is computational efficient. (2) To get VLBP and LBP-TOP, one needs information from posterior neighboring frames, which limit their application to online processing task, e.g., background modeling. On the other hand, our temporal operator O_T only requires information from previous adjacent frame. (3) Our texture and motion pattern are first model separately followed by a combining schema, which is different from [12].

3. Background Modeling based on Texture and Motion Pattern

In this section, we introduce background modeling mechanism based on texture and motion pattern described above. The goal is to construct and maintain a statistical representation of the scene that the camera sees. In the following, we explain the background modeling procedure for one pixel. The procedure is identical for each pixel, which allows for a high-speed parallel implementation if necessary.

For one particular pixel x at time T, our algorithm combines a texture pattern-based with a motion pattern-based background model in the following manner:

$$P(x_T) = (1-\gamma)P_{spatial}(x_T) + \gamma P_{temporal}(x_T) \quad (3)$$

, where $P(x_T)$ is the probability that the pixel x at time T belongs to background, $P_{spatial}(x_T)$ is the probability using texture pattern information only, $P_{temporal}(x_T)$ is the probability using motion pattern information only, and γ is a mixture factor to control the influence of the texture pattern-based background model and the motion pattern-based background model. Both models have complementary strengths to each other. Thus, our choice of combination of these two models accentuates the advantages of each. Then an incoming pixel is detected as foreground if $P(x_T) \le Th_p$, where Th_p is a threshold.

For the particular pixel x, we calculate its texture pattern histogram $HS(x,T)$ over a user-tunable rectangular region of $fs_t(x)$ at time T (see Fig.3). For efficient calculation, integral histogram [13] is used here. The distribution of the histograms by time $T-1$, denoted by $\{HS(x,t) \mid t = 1,...,T-1\}$, is modeled by K most representative histograms $\{HS_{i,(T-1)}(x) \mid i = 1,...,K\}$. Each $HS_{i,(T-1)}(x)$ have a weight $ws_{i,(T-1)}(x)$ for $i = 1,...,K$ and $\sum_{i=1}^{K} ws_{i,(T-1)}(x) = 1$. In other words, $P_{spatial}(x_T)$ can be represented as follows:

$$P_{spatial}(x_T) = \sum_{i=1}^{K} ws_{i,(T-1)}(x)\delta_{i,T}^{s}(x) \quad (4)$$

The Dirac delta function $\delta_{i,T}^{s}$ is defined as:

$$\delta_{i,T}^{s} = \begin{cases} 1, i = \min\{j \mid S(HS(x,T), H_{s_{j,(T-1)}}(x)) > Th_s\} \\ 0, otherwise \end{cases} \quad (5)$$

, where Th_s is a similarity threshold for texture pattern-based background model. The Dirac delta function $\delta_{i,T}^{s}$ equals one when its index i equals the index of the first representative histogram (of K) which is similar enough to the current texture pattern histogram $HS(x,T)$. Otherwise, $\delta_{i,T}^{s}$ equals zero. The similarity function S between two histograms H_1 and H_2 is calculated using the histogram intersection operation:

$$s(H_1, H_2) = \sum_{b=1}^{B} \min(H_{1,b}, H_{2,b}) \quad (6)$$

, where B is the number of histogram bins. $P_{temporal}(x_T)$ is then defined in a similar manner.

In the background updating phase, if none of the K most representative texture pattern histograms $\{Hs_{i,(T-1)}(x) \mid i = 1,...,K\}$ match the current texture pattern histogram $HS(x,T)$, the representative texture pattern histogram with lowest weight is replaced with the current texture pattern histogram $HS(x,T)$ weighted by a low prior weight β. A match is defined as the similarity above a threshold Th_s. The weights of the K most representative texture pattern histograms at time T are adjusted with the new data as follows:

$$ws_{i,T}(x) = (1-\alpha)ws_{i,(T-1)}(x) + \alpha\delta^{s}_{i,T}(x) \qquad (7)$$

, where α is the learning rate. The representative texture pattern histogram which matches the new observation is updated as follows:

$$Hs_{i,T}(x) = (1-\alpha)Hs_{i,(T-1)}(x) + \alpha HS(x,T) \qquad (8)$$

As a last stage of the updating procedure, we sort the K most representative texture pattern histograms in decreasing order according to their weights. The motion pattern-based background model is updated in a similar manner.

4. Experiments and Discussion

The performance of the proposed method for background subtraction is evaluated in this section. The algorithm is implemented using C++, on a computer with Intel-Core 2 1.86 GHz processor. It achieves a processing speed of 10 fps at a resolution of 160×120 pixels. We compare the performance of our method with the widely used methods of GMM [1] and LBP-based method [8]. Both qualitative and quantitative comparisons are performed to evaluate our approach. The quantitative comparison is done in terms of the number of false negatives (the number of foreground pixels that are missed) and false positives (the number of background pixels that are marked as foreground).

In order to validate the proposed technique, two different types of scenes are firstly considered. The first is the challenging outdoor scene (see Fig.4), which involves swaying trees and rain. The second is the scene (see Fig.5) which contains a moving duck in foreground, with dynamic background composed of ripples in the water with reflection. Even in these difficulty scenes, we observe that the algorithm is able to robustly detect the objects of interest with extremely low false alarm rate. In Fig.5, there are some false positives, occurring due to the reflection of the moving duck in the water. Please see Table 1 for the parameter values used, where K is the number of the most representative texture or motion pattern histograms, $N \times N$ the size of the rectangular region over which texture or motion pattern histogram is computed, Bins the number of histogram bins by using the equalization method, Th_s the threshold of histogram similarity, α the learning rate, β a low prior weight for newly added representative histogram, γ the mixture factor and Th_p the threshold of the minimum portion of the weight that should be accounted for by the background.

Table 1. The parameter values of the proposed method for the results in Fig.4, 5 and 6.

Fig	K	$N \times N$	Bins	Th_s	α	β	γ	Th_p
4	3	15×15	32	0.8	0.01	0.05	0.5	0.8
5	3	19×19	32	0.75	0.01	0.05	0.5	0.8
6	3	15×15	32	0.8	0.01	0.05	0.5	0.8

In Fig. 6(a), we show the results of the proposed method using other three test sequences. The sequences used in the experiment include dynamic background, and illumination changes. The frames on the first column are from background subtraction competition in VSSN2006 [14], which contain heavily swaying trees. The next frames on the second column are from [5] where the challenge was due to the vigorous motion of the trees and bushes. The proposed method robustly handles these situations and the moving object is detected correctly because it combines texture and motion information of background image variations. The last frames on the third column are from [6] where a person walks in front a swaying tree. The proposed method also gives good results. Identical parameters are used in these sequences, although it is possible to adapt the values for better performance. Please see Table 1 for the parameter values used. In experiments, for all parameters, we find that a good value can be chosen across a wide range of values.

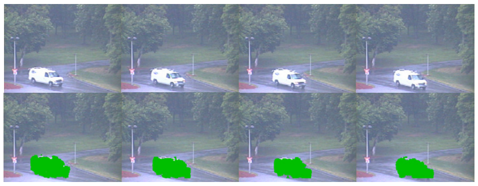


Fig.4. Detection results from a outdoor sequence, which contains swaying trees and rain.

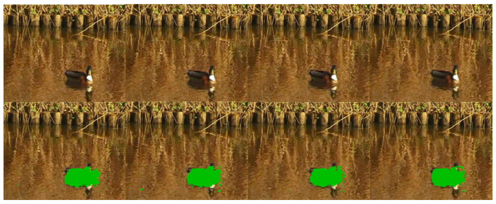
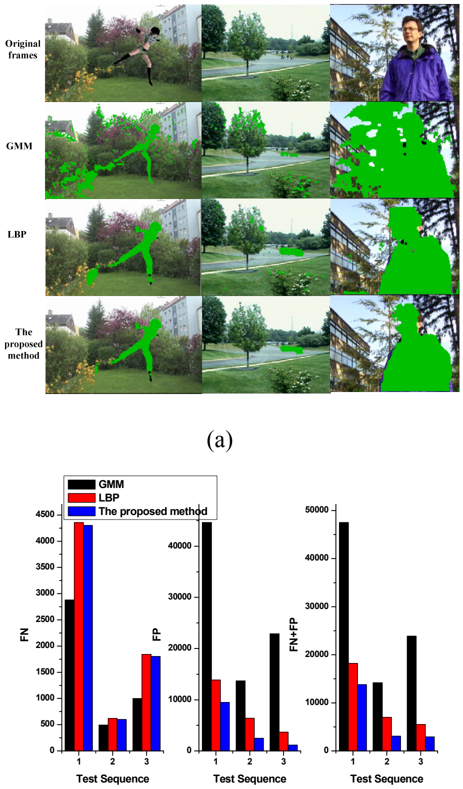


Fig.5. Detection results from a sequence, which contains a moving duck in foreground, with dynamic background composed of ripples in the water with reflection.

In order to provide a quantitative perspective about the quality of foreground detection with our approach, we manually mark the foreground regions in five frames from each sequence in Fig.6 to generate ground truth data, and make comparison with GMM and LBP. The numbers of error classifications are achieved by summing the errors from the frames corresponding to the ground truth frames.



(a)

(b)

Fig.6. Comparision results of GMM, LBP and the proposed method. a) is the original test sequences and some detection results of the GMM, LBP and the proposed method. b) is the test results. FN and FP stand for false negatives and false positives, respectively.

The corresponding quantitative comparison is reported in Fig. 6(b). For all sequences, the proposed method achieves excellent performance in terms of false positives, and false negatives are acceptable. Since the proposed method is obtained by fusing texture and motion pattern, it is robust against dynamic background and illumination changes. It should be noticed that, for the proposed method, most of the false negatives occur on the contour areas of the foreground objects (see Fig. 6(a)). This is mainly because the features are extracted from the pixel neighborhood. Overall, the proposed method outperforms the comparison methods for the used test sequences.

5. Conclusion and Future Work

In this paper, we propose an algorithm to combine texture pattern with motion pattern for background subtraction under difficult conditions. Both texture and motion information can be obtained using two LBP-like operators, which are invariant to illumination changes and computational simplicity. Experiments show that the proposed background subtraction algorithm is robust to dynamic movement in natural scenes such as swaying vegetation, waving trees, ripples in the water and rain.

Our future work will focus on how to fuse texture and motion pattern for other computer vision applications, such as behavior analysis and visual speech recognition.

6. Acknowledgements

This paper is partially supported by National Natural Science Foundation of China under contract No.60775024 and Program for New Century Excellent Talents in University NCET-05-03 34.

7. References (This is "Header 1" style)

[1] C. Stauffer and W.E.L. Grimson. *Learning Patterns of Activity Using Real-Time Tracking*. TPAMI, vol. 22, no. 8, pp. 747-757, August 2000.

[2] A. Elgammal, D. Harwood, and L. Davis. *Non-parametric Model for Background Subtraction*. ECCV, vol.2, pp.751-767, June 2000.

[3] K. Kim, T.H. Chalidabhongse, D. Harwood and L. Davis. *Real-time Foreground-BackgroundSegmentation using Codebook Model.* Real-Time Imaging, vol.11, issue 3, pp.167-256, June 2005.

[4] Y. Sheikh and M. Shah. *Bayesian Modeling of Dynamic Scenes for Object Detection.* TPAMI, vol. 27, no. 11, pp. 1778-1792, November 2005.

[5] A. Mittal and N. Paragios. *Motion-based background subtraction using adaptive kernel density estimation.* CVPR, vol.2, pp.302-309, July 2004.

[6] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. *Wallflower: Principles and Practice of Background Maintenance*. ICCV, vol.1, pp.255-261, 1999.

[7] O. Javed, K. Shafique, and M. Shah. *A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information*. IEEE Workshop on Motion and Video Computing, pp.22-27, 2002.

[8] M. Heikkila and M. Pietikainen. *A Texture-Based Method for Modeling the Background and Detecting Moving Objects*. TPAMI, vol. 28, no. 4, pp. 657-662, April 2006.

[9] Y.L. Tian, M. Lu, A. Hampapur. *Robust and efficient foreground analysis for real-time video surveillance*. CVPR, vol.1, pp.1182-1187, June 2005.

[10] K.A. Patwardhan, G. Sapiro and V. Morellas. *Robust Foreground Detection in Video Using Pixel Layers*. TPAMI, vol. 30, no. 4, pp. 746-751, April 2008.

[11] M.Piccardi. *Background subtraction techniques: a review*. SMC (4), pp. 3099-3104, 2004.

[12] G.Y. Zhao and M. Pietikainen. *Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions*. TPAMI, vol. 29, no. 6, pp. 915-928, June 2007.

[13] F. Porikli. Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces. CVPR, vol.1, pp. 829-836, June 2005.

[14]http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/#testvideo.