Dynamic Background Subtraction using Spatial-Color Binary Patterns

Wei Zhou*, Yu Liu*, Weiming Zhang*†, Liansheng Zhuang* and Nenghai Yu*
*MOE-Microsoft Key Laboratory of Multimedia Computing and Communication
University of Science and Technology of China, Hefei, China
†Department of Information Research
Zhengzhou Information Science and Technology Institute, Zhengzhou, China
{zhouwei1, liuyu35}@mail.ustc.edu.cn, zwmshu@gmail.com, {lszhuang, ynh}@ustc.edu.cn

***Abstract*—In this paper, an efficient approach for background modeling and subtraction is proposed. It's based on a novel spatial-color feature extraction operator named spatial-color binary patterns(SCBP). As the name implies, features extracted by this operator include spatial texture and color information. In addition, a refine module is designed to refine the contour of moving objects. Using the proposed method, we improve the accuracy of subtracting the background and detecting moving objects in dynamic scenes.**

A data-driven model is used in our method. For each pixel, first, a histogram of SCBP is extracted from the circular region, and then a model consist of several histograms is built. For a new observed frame, each pixel is labeled either background or foreground according to the matching degree between its SCBP histogram and its model, then the label is refined and finally the model of this pixel is updated. The proposed approach is tested on challenging video sequences, which shows that the proposed method performs much better than several texture-based methods.

***Keywords*-local binary patterns; spatial-color binary patterns; dynamic background modeling;**

I. Introduction

With the development of digital capture and storage technology in the past decades, video surveillance is almost everywhere in cities, the collected video data also grows explosively, automatic video analysis is in urgent needs. Moving objects discovery is a very important task for intelligent Video Surveillance, and is the basis for further semantic analysis. Background subtraction is an important methods for moving object detection, and has a very wide range of applications, such as object tracking, activity recognition, behavior understanding and content based compression. In this paper, we aimed to design an efficient algorithm to extract moving objects in surveillance videos. The key of background subtraction is to build and maintain an adaptive background model to represent the background of a video, which is a challenging task owing to that backgrounds of scenes in real-life are usually dynamic, including noise, illumination changes, swaying trees, rippling water and so on.

In order to overcome these problems, many approaches had been proposed. There are some surveies and comparative studies [3][18][16][12] [4] examined a wide-range of background subtraction methods. While some approaches were proposed for videos captured by freely moving camera [13][19] or highly dynamic scenes [15], most of models and approaches were designed for videos captured by static camera.

One of the most famous model for background subtraction is the Gaussian Mixture Model(GMM)[21], extended from single Gaussian model [23], which fits a gaussian distribution to the values of each pixel over a series of frames. GMM models each pixel by a weighted sum of several gaussian components to account for dynamic scenes. Stenger et al.[17] used hidden Markov model to adaptively select the number of components, and Z. Zivkovic [25] proposed an extended version of GMM .

A. Elgammal et al. [6] proposed a nonparametric kernel density estimation(KDE) technique for building statistical representations of the background and the foreground. This method estimates probability density function(pdf) of pixel values directly from previous frames. KDE techniques are powerful but computational and memory expensive. Later, A. Elgammal applied fast Gauss transform to KDE to save computational cost[5]. Y. Sheikh and M. Shah [20] also improved the KDE method by considering both color and location information of pixels, to some degree, spatial information was taken in to account.

Both GMM and KDE are pixel-wise method, these methods model each pixel independently. However, because neighboring pixels are highly relative, assuming independence between pixels is unreasonable and restrict their use in dynamic background. There will be many false alarms if the background changes significantly.

Another possible choice is block-wise framework. For example in [8], frames were divided into blocks, then a model was built for each block by learning a classifier, and the classifier was updated online. For each new frame, blocks were classified into either foreground or background. Block-wise methods take advantage of inter-block neighborhood information, however, the basic unit is block, thus block-wise methods cannot get accurate shape information, but only coarse foreground detection.

A texture based model proposed by M. Heikkilä [10][9] was popular in recent years. The authors used Local binary patterns(LBP)[11] to describe textures, and built a model

978-0-7695-4541-7/11 $26.00 © 2011 IEEE
DOI 10.1109/ICIG.2011.76

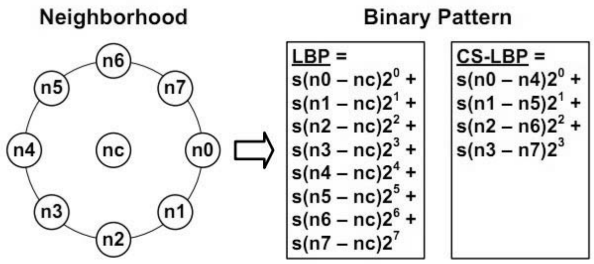


Figure 1. LBP and CS-LBP features for a neighborhood of 8 pixels, from [11].

based on LBP histograms over circular regions for a given pixel. The LBP based model is robust to backgrounds made of animated textures. Two extended texture-based models were proposed to improve the performance, S. Zhang et al. extended this model to temporal and proposed Spatio-temporal LBP based background model [24], and G. Xue et al. used spatial extended center-symmetric LBP(SCS-LBP)[7] to build background model.

However, all the above mentioned texture-based models did not take into account the color information, which is very important and discriminative to our intuition. Another drawback of these model is that they can't depict object contours accurately because of using histograms over regions. In this paper, we propose a novel spatial-color binary patterns(SCBP) operator to fuse texture and color information, and a refine strategy is designed to handle the second problem(Section III-B).

The rest of this paper is organized as follows: in section 2, the proposed SCBP operator is introduced, then we describe our proposed background subtraction in detail in section 3. Experiments and results are given in Section 4, followed by the conclusion of the paper.

II. SCBP OPERATOR

LBP[11] is a popular texture description, which has shown excellent performance on object detection and face recognition [1][2]. LBP has several favorable properties, the most important properties are its computational simplicity and its tolerance for illumination changes. LBP describes a pixel by comparing the value of a pixel to its neighbors. The definition of LBP operator is:

$$LBP_{N,R}(x_c, y_c) = \sum_{i=0}^{N-1} 2^i s(g_i - g_c), \quad s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}, \tag{1}$$

where g_c is the gray value of the center pixel(x_c, y_c), and N denotes the number of neighbors choosed to compare. All these neighbors are evenly distributed on a circle around (x_c, y_c) with radius R. The values of neighbors that do not fall exactly on pixels are estimated by bilinear interpolation. An extended version of LBP is Center-symmetric LBP(CS-LBP), which computes the binary bits by comparing the gray value of a pair of centrosymmetric pixels rather than comparing each neighbor to central pixel, see Fig. 1.

$$\text{CS-LBP}_{2N,R}(x_c, y_c) = \sum_{i=0}^{N-1} 2^i s(g_i - g_{i+N}), \tag{2}$$

Both LBP and CS-LBP are computed on gray scale images; thus only texture information is included. But in real-world surveillance videos, the color of foreground objects is usually different from the color of background, thus besides the intensity, color information is another important factor to distinguish foreground and background. Motivated by this fact, we proposed a novel binary operator named SCBPwhich enhances the LBP with color information.

$$\begin{aligned} SCBP_{2N,R}(x_c, y_c) &= LBP_{N,R}(x_c, y_c) \\ &+ 2^{N+1} f(R_c, G_c \mid \gamma) \\ &+ 2^{N+2} f(G_c, B_c \mid \gamma) \\ &+ 2^{N+3} f(B_c, R_c \mid \gamma), \\ f(a, b \mid \gamma) &= \begin{cases} 1, a > \gamma b \\ 0, \text{otherwise} \end{cases}, \end{aligned} \tag{3}$$

where the R_c, G_c and B_c are the three color channels of the center pixel (x_c, y_c), and $\gamma > 1$ is a factor to suppress the noise. In the rest of this paper, we set the binary to zero unless one color channel is 1.1 times larger than another. By adding color information, the length of binary bits grows, which will lead to exponential growth of patterns, i.e. the dimension of histograms, and will seriously affect the efficiency of our algorithm. So we cut down patterns by using CS-LBP, choosing a small number N and dropping one of the three color bits. In fact, the three color bits are highly correlative, dropping one of them is not critical. The final SCBP we used in this paper is defined as:

$$\begin{aligned} SCBP_{2N,R}(x_c, y_c) &= \text{CS-LBP}_{2N,R}(x_c, y_c) \\ &+ 2^{N+1} f(R_c, G_c \mid \gamma) \\ &+ 2^{N+2} f(G_c, B_c \mid \gamma), \end{aligned} \tag{4}$$

If we set $N = 4$, the total number of SCBP patterns is 64, which is just appropriate. The SCBP histogram computed over a circular region of radius R_{region} around the pixel is used as the feature vector to represent a pixel, and background model is built based on these feature vectors, here R_{region} is a parameter set by the user.

III. BACKGROUND SUBTRACTION USING SCBP

In this section, we introduce a new background subtraction methods based on proposed SCBP. Our algorithm consists of three modules: background modeling, foreground detection and contour refinement. The first two modules are

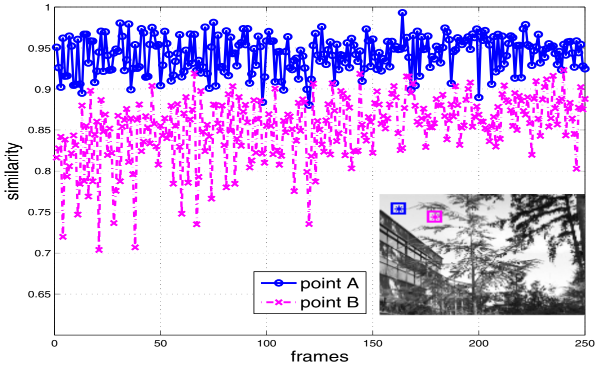


Figure 2. The matching similarity of two background pixel. The curve of Pixel A(the left one in the image) lies above B's, thus it's better to set A a higher threshold.

similar with [9]. The novelties of our algorithm include updating decision threshold adaptively and contour refinement.

In the background modeling module, a model, which is composed of a ground of K weighted SCBP histograms, is built for each pixel, and the top $B < K$ histograms are considered as background. Given the value of the pixel in a new frame, algorithm first computes the feature vector, i.e. the SCBP histogram, and then calculates the similarities between the feature vector and the pixel's model. Similarities larger than the threshold T_p indicate match, and finally both the histograms and weights are updated differently according to the matching status. In the foreground detection module, a pixel is classified into foreground if there is no match occurs between feature vector and the background histograms, otherwise the pixel is labeled as background. The output of the detection module is a binary image showing foreground pixels. For more details, see literature [9].

In the following subsections, we first explain adaptive decision threshold and then introduce our refinement module in detail.

A. Adaptive decision thresholds

Earlier texture-based methods [9][24][7] use a global const threshold T_p to determine match or not for all pixels in their model, but in fact, different pixels of a frame have significantly different instability, and instability of a same pixel changes over time. Thus a global constant threshold may miss some positives and introduce some false alarm. For example, Fig. 2 shows the best match at each frame for two pure background pixels picked from a video, in which we can see that the similarity curve of pixel A is higher than B's, because A is in a flat region while B is covered by waving trees from time to time. So a higher threshold is suitable for A.

In our model, each pixel (x, y) has its own threshold $T_p(x, y)$, which is initialized as global value T_p. At each time, after updating the background model, the threshold is updated similarly:

$$T_p(x, y) = (1 - \alpha)T_p(x, y) + \alpha(s(x, y) - 0.05) \ , \qquad (5)$$

where $s(x, y)$ is the largest similarity between feature vector and background histograms, and α is a learning rate close to one. In this way, the thresholds for static pixels will increase and decrease for dynamic pixels. Thus our background subtraction method is more sensitive in static region and more tolerant in dynamic region.

B. Contour refinement

The texture-based model mentioned above is built on histograms computed over surrounding regions, though each pixel is modeled identically, it's still block-wise. On one hand, it's robust to dynamic background such as waving trees and rippling water; on the other hand it has common drawbacks of block-wise models. A major problem is that the contour of detected object is illegible. Because of using histogram over regions, not only the real foreground, but also the background pixels near the edges of foreground will be classified into foreground, and thus the contour of foreground objects is obscured and false alarm rise, see Fig. 3(b).

In this section, we propose a simple strategy to reduce the false alarm by generating a pixel-wise binary mask Ω and applying it to the output of background model. The binary mask Ω must satisfy:

1) Almost all true foreground pixels are covered;
2) Allow some false alarm, but the less the better.

These requirements are not difficult to meet and there are many approaches to generate such a mask. We consider this problem in two aspects:

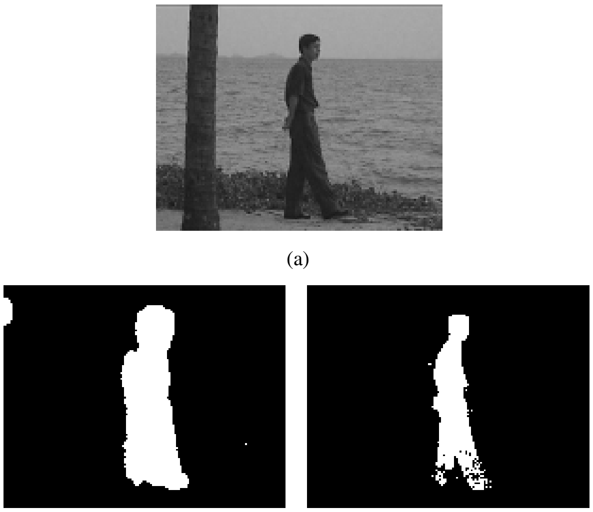
1) If the intensity deviate a lot from average, the pixel is probably belongs to foreground.
2) If the color composition changes obviously, the pixel is probably belongs to foreground.

According to the above considerations, intensity and color rate are used to generate the mask. let $\bar{.}$ and $std(.)$ indicate the average and standard deviation, we calculate the mask Ω_i for i_{th} pixel by the following formulation:

$$\Omega_i = \begin{cases} 1, & \text{if } [d_i \geq \xi std d_i]\&[d_i/\bar{g}_i \geq \varepsilon_1], \\ 1, & \text{if } \|(r_i, g_i, b_i) - (\bar{r_i}, \bar{g}_i, \bar{b_i})\|_2 \geq \varepsilon_2, \\ 0, & \text{otherwise} \ , \end{cases} \qquad (6)$$

Here,$d_i = abs(g_i - \bar{g}_i)$ is the absolute deviation of intensity from average. Given the three color channels R, G and B, (r, g, b) are chromaticity coordinates calculated by $r = R/(R + G + B)$, $g = G/(R + G + B)$ and $b = B/(R + G + B)$. We set parameters $\xi = 2.5$ and $\varepsilon_1 = \varepsilon_2 = 0.2$ empirically in this paper.

Another advantage of this formulation is that it can suppress shadow by constraining $d_i/\bar{g}_i \geq \varepsilon$, which brings the



(a)

(b) (c)

Figure 3. An example of contour refinement:(a) original image, (b) detected foreground,(c) final result after refinement.

similar effect as Ahmed Elgammal chose a subset of samples affected by shadows to produce the observed lightness of a pixel in [6].

Let Ψ is the output of foreground detection model, the final result of our algorithm is:

$$isFG = \Psi \ \& \ \Omega, \tag{7}$$

Eq. (7) means that a pixel is labeled foreground if and only if both foreground detection module and refine module agree. Then the average and standard deviation are updated for background pixels identified by $isFG$.

$$\begin{aligned} \bar{g_i} &= (1-\beta)\bar{g_i} + \beta g_i \ , \\ std_i &= \sqrt{(1-\beta)std_i^2 + \beta(g_i - \bar{g_i})^2} \ , \end{aligned} \tag{8}$$

The chromaticity coordinates $(\bar{r_i}, \bar{g_i}, \bar{b_i})$ are updated the same as g_i. Fig. 3 shows the effectiveness of our refine strategy.

IV. Experiments

In order to confirm the capability of proposed background subtract approach, we perform our algorithm on several public available video sequences. Five methods are compared in out experiments, including the GMM method [21], LBP based method [9], SCS-LBP based method [7], the proposed SCBP based method without refinement(basic SCBP) and SCBP based method with refinement(SCBP). Both visual and numeric methods are used to evaluate the performance. We use detection rate(DR) and false alarm rate(FAR) to evaluate the accuracy of methods.

Figure 4. Foreground detection results on waving trees, columns correspond to the 247^{th}, 251^{th}, 255^{th} and 261^{th} frames. The top two rows are the original frames and the ground truthes, the 3^{rd} to 5^{th} rows are the results obtained by GMM, LBP, SCS-LBP respectively. The 6^{th} row is the results of proposed basic SCBP , the bottom row is the results of SCBP with refinement. Note that we didn't apply any morphological filter to these results.

$$\begin{aligned} DR &= \frac{\#ture\ positives}{\#ture\ positives + \#false\ negatives} \ , \\ FAR &= \frac{\#false\ positives}{\#false\ positives + \#ture\ positives} \ , \end{aligned} \tag{9}$$

Same as LBP and SCS-LBP based method, there is a set of parameters need to be set, including region radius R_{region}, number of histograms K, threshold to detect foreground T_P and threshold to estimate background histograms T_B, R and N for LBP operator, and the learning rates α and β. In our experiments, these parameters are set as follows: $R_{region} = 9$, $R = 2$, $N = 4$ (for SCS-LBP and SCBP, $N = 6$ for LBP), $K = 4$, $T_P = 0.65$, $T_B = 0.7$, $\alpha = \beta = 0.01$.

The first video named waving trees is from [22], and the second video named water surface is from [14], both of them are typical scenes of dynamic background. Visual comparisons are shown in Fig. 4 and Fig. 5, while the numeric comparisons are listed in Table I and Table II.

The visual results are very impressive that the detected foreground of the proposed SCBP method is almost the same as the manually segmented ground truth. From the

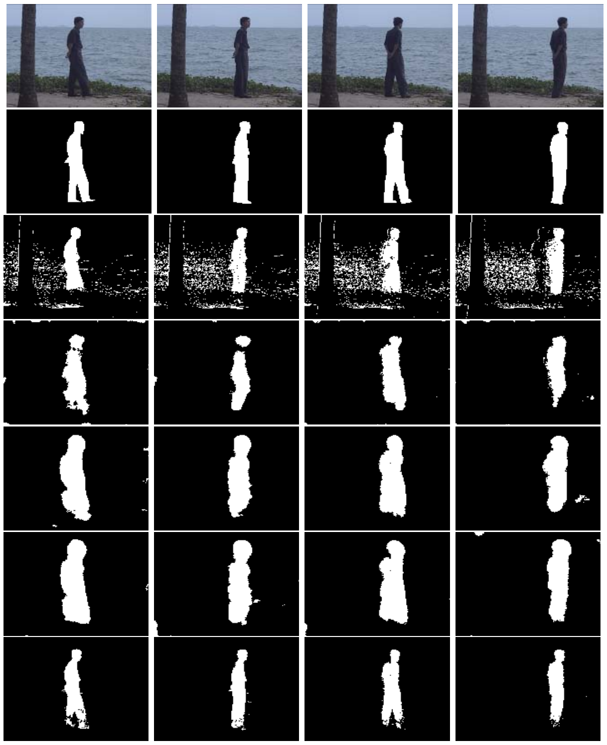


Figure 5. Comparison results on water surface, columns correspond to the 1515^{th}, 1548^{th}, 1559^{th} and 1594^{th} frames. All rows stand the same meaning as Fig. 4

Table I
NUMERIC COMPARISON ON THE WAVING TREES SEQUENCE.

frame	251^{th}		255^{th}	
	DR(%)	FAR(%)	DR(%)	FAR(%)
GMM	64.9	42.0	60.0	47.3
LBP	74.2	5.17	70.5	4.81
SCS-LBP	85.7	9.00	83.5	8.88
baisc SCBP	100	7.98	99.7	10.2
SCBP	99.5	2.33	99.4	2.22

figures, we can see that the basic SCBP method is robust to dynamic background and detects foreground with really few false negatives, and SCBP based method with refinement detected the contours of objects accurately. In the numeric evaluation, basic SCBP improved DR significantly while keeping FAR at a comparable level with SCS-LBP based method, and the proposed SCBP method reduced the false alarm rate to a much lower level than others. According to the test results over all, our SCBP method is powerful and outperforms GMM, LBP and SCS-LBP.

V. CONCLUSION

In this paper, we aimed at subtracting background and detecting moving objects from videos. A novel background subtraction method based on spatial and color textures is proposed. The contributions of this paper include:1) the SCBP operator was proposed to take into account both the texture and color information; 2) adaptively decision thresholds were used, which improved sensitivity in static regions while enhancing tolerance in dynamic regions; 3) a contour refinement was proposed, which extracted the true contours of objects and reduced false alarms.

Table II
NUMERIC COMPARISON ON WATER SURFACE.

frame	1515^{th}		1559^{th}	
	DR(%)	FAR(%)	DR(%)	FAR(%)
GMM	69.4	47.1	69.6	56.8
LBP	83.5	25.5	81.9	15.6
SCS-LBP	94.1	37.3	89.9	28.7
baisc SCBP	96.2	37.8	91.9	32.2
SCBP	85.2	1.74	82.2	0.81

ACKNOWLEDGMENT

This work is partially supported by the National Science Foundation of China (NO. 60933013), the National Science and Technology Major Project (NO. 2010ZX03004-003), and the Fundamental Research Funds for the Central Universities(NO. WK210023002, NO. WK2101020003).

REFERENCES

[1] T. Ahonen, A. Hadid, et al. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2037–2041, 2006.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *European Conference on Computer Vision*, pages 469–481, 2004.

[3] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *IEEE 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[4] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3):033003, 2010.

[5] A. Elgammal. Efficient kernel density estimation using the fast gauss transform with applications to color modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1499–1514, 2003.

[6] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[7] J. S. G. Xue and L. Song. Dynamic background subtraction based on spatial extended center-symmetric local binary pattern. In *IEEE International Conference on multimedia and expo*, pages 1050–1054, 2010.

[8] H. Grabner and H. Bischof. On-line boosting and vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 260–267, 2006.

[9] M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):657–662, 2006.

[10] M. Heikkilä, M. Pietikäinen, and J. Heikkilä. A texture-based method for detecting moving objects. In *British Machine Vision Conference*, volume 1, pages 187–196, 2004.

[11] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern recognition*, 42(3):425–436, 2009.

[12] S. Herrero and J. Bescós. Background subtraction techniques: Systematic evaluation and comparative analysis. In *Advanced Concepts for Intelligent Vision Systems*, pages 33–42. Springer, 2009.

[13] Y. Jin, L. Tao, H. Di, N. Rao, and G. Xu. Background modeling from a free-moving camera by multi-layer homography algorithm. In *15th IEEE International Conference on Image Processing*, pages 1572–1575, 2008.

[14] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[15] V. Mahadevan and N. Vasconcelos. Background subtraction in highly dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 1–6.

[16] S. Panahi, S. Sheikhi, S. Hadadan, and N. Gheissari. Evaluation of background subtraction methods. In *Digital Image Computing: Techniques and Applications*, pages 357–364. IEEE, 2008.

[17] S. Paragios, B. Stenger, V. Ramesh, N. Paragios, and F. Buhmann. Topology Free Hidden Markov Models: Application to Background Modeling. In *In IEEE International Conference on Computer Vision*, volume 1, pages 294–301, 2001.

[18] D. Parks and S. Fels. Evaluation of background subtraction algorithms with post-processing. In *IEEE 5th International Conference on Advanced Video and Signal Based Surveillance*, pages 192–199, 2008.

[19] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *IEEE 12th International Conference on Computer Vision*, pages 1219–1225, 2009.

[20] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1778–1792, 2005.

[21] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[22] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE International Conference on computer vision*, pages 255–261, 1999.

[23] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[24] S. Zhang, H. Yao, and S. Liu. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In *15th IEEE International Conference on Image Processing*, pages 1556–1559, 2008.

[25] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *IEEE 17th International Conference on Pattern Recognition*, volume 2, pages 28–31, 2004.