Moving Object Detection Revisited: Speed and Robustness

Hong Han, *Member, IEEE*, Jianfei Zhu, Shengcai Liao, *Member, IEEE*, Zhen Lei, *Member, IEEE*, and Stan Z. Li, *Fellow, IEEE*

***Abstract*— The detection of moving objects in videos is very important in many video processing applications, and background modeling is often an indispensable process to achieve this goal. Most of the traditional background modeling methods utilize color or texture information. However, color information is sensitive to illumination variations and texture information cannot be utilized to separate smooth foreground from smooth background in most cases. Achieving good performance in terms of high foreground detection accuracy and low computational cost is also challenging. In this paper, we propose a new integration framework of texture and color information for background modeling, in which the foreground decision equation includes three parts (one part for color information, one part for texture information, and the left part for the integration of color and texture information). This framework is able to combine the advantages of texture and color features while inhibiting their disadvantages as well. Moreover, we propose a block-based method to accelerate the background modeling. In particular, in the texture information modeling process, a single histogram model is established for each block whose bins indicate the occurrence probabilities of different patterns, which is different from the traditional multihistogram model for block-based background modeling, and then dominant background patterns are selected to calculate the background likelihood of new coming blocks. Dynamic background and multimodal problems can be handled through this technique. To evaluate the foreground detection performance reasonably, a new quality measure is proposed. Extensive experiments on various challenging videos validate the effectiveness of the proposed method over state-of-the-art methods.**

***Index Terms*— Block based, integrated information, moving object, object detection, single histogram model.**

Manuscript received May 2, 2014; revised August 28, 2014; accepted October 30, 2014. Date of publication November 5, 2014; date of current version June 2, 2015. This work was supported in part by the National Natural Science Foundation of China under Projects 61203267, 61375037, 61473291, 61075041, and 61105016; in part by the National Science and Technology Support Program under Project 2013BAK02B01; in part by the Chinese Academy of Sciences, Beijing, China, under Project KGZD-EW-102-2; and in part by the AuthenMetric Research and Development Funds. This paper was recommended by Associate Editor C. Regazzoni.

H. Han is with the School of Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: hanh@mail.xidian.edu.cn).

J. Zhu is an Algorithm Engineer with the Alibaba Group, Hangzhou, China. This work was done when he visited the National Laboratory of Pattern Recognition, Center for Biometrics and Security Research, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

S. Liao, Z. Lei, and S. Z. Li are with the National Laboratory of Pattern Recognition, Center for Biometrics and Security Research, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2014.2367371

I. Introduction

MOVING object detection plays an important role in many video processing applications, such as object tracking, categorization, reidentification, and video condensation. It often serves as preprocessing for higher level video analyses and its performance directly affects the performance of the subsequent applications. For object tracking, if a moving object is detected as two or two moving objects are detected as one, the tracking result may be incorrect. For object categorization, incomplete or adhesive detection of moving objects may lead to wrong categorization, and it is the same case for object reidentification. For video condensation, object tracking is also an indispensable part. It is not the desired result if the head and legs of one person appear at different time in the condensed video. Ideally, a detection method should detect each moving object separately without breaking.

Background modeling is indispensable for moving object detection in many cases and many works have been done in this research area. In early works, the background model was constructed for each pixel independently. In [1], a single Gaussian model was used to model the value variations of each pixel, and the parameters of the Gaussian model were updated recursively with an adaptive filter. It is robust in modeling the static background, but is sensitive to dynamic background variations. To address these problems, the mixture of Gaussian models (MoGs) was proposed in [2]. In addition, a series of variants [3], [4] were proposed to improve the performance of background modeling. However, the Gaussian model-based methods are all based on the assumption that the pixel intensity follows a Gaussian distribution, which is not always correct. In contrast, a nonparametric method was proposed in [5], where each pixel is directly modeled with a probability density function without any assumption on the distribution of the pixel values. Since all the possible pixel values are modeled into the probability density function, dynamic background modeling problem can be handled and illumination changes can be adapted progressively. In [6], a Bayesian decision rule for background and foreground classification was constructed based on different feature selections for static and moving backgrounds, and is able to deal with dynamic background problems better than MoG. In [7] and [8], background pixel values are quantized into a set of codebooks to form the background model which is efficient in memory and speed.

Pixel intensity-based modeling techniques cannot handle the illumination variation problem at feature level since pixel

Fig. 1. Left: busy human stream with ruleless lighting condition changes, so color information may not work well. Right: foreground and background are both smooth, so there may not be evident differences between foreground and background in texture information.

intensity itself is sensitive to the change of lighting conditions. The robustness of the texture feature to illumination variations makes it possible to handle the lighting changing problem. In [9], a set of powerful filter operators was integrated with a linear prediction model to detect foregrounds, where the filter operators are automatically selected. Dynamic background modeling with illumination variations can be solved. In [10], an local binary pattern (LBP) histogram-based method was proposed to handle the illumination changing problem because of the stability of the LBP feature in terms of illumination variations, but the moving cast shadow problem was not well handled yet. Mutihistogram techniques have also been used to deal with multimodal problems. To address the moving cast shadow problem, a new scale-invariant local ternary pattern (SILTP) feature was proposed in [11]. A pattern kernel density estimation technique was also introduced, and multimodal and multiscale background modeling were adopted to deal with complex dynamic scenes.

There also exist methods that fuse color and texture information to get robust background modeling in [12]–[14]. However, their fusion way is just assigning different weight values for color and texture information, which may not be so effective in some cases.

Until now, only a few works have been done on block-based background modeling or background subtraction. The advantage of the block-based strategy over the pixel-based one is that stable foreground detection results can be achieved with less computation and memory resource, while the disadvantage is that the detection boundary will be very coarse, and adjacent moving objects may be connected. Early works of block-based background modeling include [15] and [16]. In [15], the normalized vector distance measure was used for block correlation computing. In [16], an edge histogram was calculated for each block. Later, a multi LBP histogram-based blockwise background model was proposed in [17] to achieve robustness to illumination changes and high processing speed, but the detection boundary is coarse and adjacent moving objects tend to be connected.

Color information is sensitive to illumination variations, while texture information cannot be utilized to separate smooth foreground from smooth background in most cases (Fig. 1). In this paper, we propose a new integration framework of color and texture information, which can inherit their advantages while inhibiting their disadvantages. Since background modeling is usually a pretreatment for higher level video analyses, it should be computationally efficient. Therefore, we use the block-based strategy and construct one model for each block, which is different from the pixel-based strategy that has one model for each pixel. A lot of computational resources can be saved. Traditional block-based methods construct a model of several histograms for each block to deal with dynamic background and multimodal problems and make foreground detection decision for new frames by histogram matching, which are also time consuming. Instead, we construct the background model of just one histogram with its bins indicating the probabilities of their corresponding patterns in the block. Since all the frequently appearing patterns can be dominant in the model histogram, we are able to deal with dynamic background and multimodal problems. As aforementioned, the block-based method has a shortcoming that the detection boundary will be coarse and may connect adjacent moving objects. To deal with this problem, we use two levels of block sizes. Background models are constructed in big blocks for stability, while detection decisions are made for small blocks to achieve finer boundary. Traditional pixel-based quality measurement is not suitable for evaluation of foreground detection results because the cases of close moving objects connecting together, or one moving object breaking into several parts, cannot be reflected directly. Therefore, we propose a new region-based quality measurement to directly show the effectiveness of moving object detection.

The main contributions of our paper are as follows. First, a new integration framework of texture and color information is proposed and both illumination variations and smooth background–foreground problem can be handled. Second, the block-based strategy is used and a single histogram model is established for each block, which makes our modeling process fast with little memory consuming. Dynamic background and multimodal problem can also be handled via our dominant background pattern selection process. Third, two levels of block sizes are used to benefit from the fact that the background model in big blocks can be more stable, while the final foreground detection boundary based on small blocks can be more accurate. Finally, a new quality measure is designed according to the rule that different moving objects should be detected separately without breaking.

The rest of this paper is organized as follows. Section II describes the SILTP information-based background model. Section III describes the color information-based background model. In Section IV, we give the integration way of SILTP information and color information for our foreground detection judgement. A new quality measure is designed to evaluate moving object detection results in Section V. Comprehensive experiments and analyses are demonstrated in Section VI. Finally, the conclusion is drawn in Section VII.

II. Blockwise Background Model Based on SILTP Information

A. Scale Invariant Local Ternary Pattern

Liao *et al.* [11] proposed the SILTP feature representation for background modeling, which is more robust to illumination changes. The SILTP can be encoded as

$$\mathrm{SILTP}^{\tau}_{N,R}(x_c, y_c) = \bigoplus_{k=0}^{N-1} s_\tau(I_c, I_k) \tag{1}$$

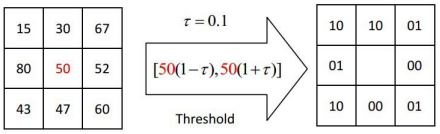


Fig. 2. SILTP operator.

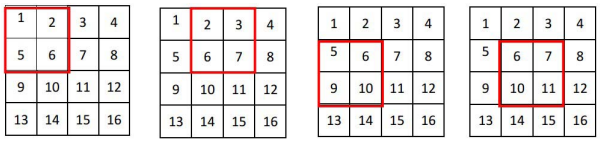


Fig. 3. There are 16 small blocks, and we show four red big blocks here, which are partially overlapping. It can be seen that each small block may belong to several big blocks.

where I_c is the gray intensity value of the center pixel, I_k represents the gray intensity values of the N neighborhood pixels equally spaced on a circle of radius R, $\bigoplus$ denotes the concatenation operator of binary strings, τ is a scale factor indicating the comparing range, and s_τ is a piecewise function defined as

$$s_\tau(I_c, I_k) = \begin{cases} 01, & \text{if } I_k > (1+\tau)I_c \\ 10, & \text{if } I_k < (1-\tau)I_c \\ 00, & \text{otherwise.} \end{cases} \tag{2}$$

Fig. 2 shows an example of the SILTP operator.

B. Block-Based Method

Our block-based strategy is motivated by [17], who divided each video frame into equally sized blocks using a partially overlapping grid structure and constructed a multihistogram model for each block. Their foreground detection decision was also made for each block, which will result in coarse foreground detection boundaries and may connect adjacent moving objects because of the large bock size. Realizing that the size of the overlapped area is smaller than the initial block size, we develop a two-level block size strategy to fix this problem.

In our strategy, each image is divided into small blocks: four small blocks, form a big block, while the big blocks are partially overlapping, as shown in Fig. 3. A background model is calculated for each big block but the final foreground detection decision is made for each small block. The strategy has several advantages: the SILTP histogram model extracted from a big block can be more stable than that from a small one (since the small block is less tolerable to background movements and is more sensitive to noises). The partially overlapping technique is able to obtain more information than the nonoverlapping one. In addition, the final judgement of each small block depending on the information of all the big blocks, which it belongs to is more believable. Moreover, the detection result based on small blocks can be more refined.

A histogram of the SILTP feature is calculated for each big block. As described above, since each neighborhood pixel can be encoded as one of the three possible patterns, there are totally 3^N possible patterns and we can further map the SILTP strings to $[1, 3^N]$. Suppose the size of the big block is $S_b \times S_b$, and the mapped number of the SILTP code of pixel i is M_i. For each pixel i, we calculate the SILTP histogram of the big block H_s as

$$H_s(M_i) = H_s(M_i) + \frac{1}{S_b \cdot S_b} \tag{3}$$

where $H_s(M_i)$ is the M_ith bin of histogram H_s, and there are totally 3^N bins. Then, the SILTP histogram of each big block is obtained.

C. Background Modeling

Heikkila and Pietikainen [10] and Heikkilä *et al.* [17] construct a model of several histograms to represent the appearance of one block in the video sequence and get the foreground detection decision by matching the new block histogram with the existing background histograms. However, there may be too many possible histogram types due to dynamic background and multimodal conditions. Consequently, a limited number of histograms is not enough to represent the background well and useful information may be lost. Realizing this difficulty, we do not try to focus on the typical histograms; instead, we focus on the patterns appearing in the block area. Therefore, we construct the background model of each block with only one histogram whose bins indicating the appearing probabilities of its corresponding patterns in this block. Since moving objects usually appear for a short time in a video sequence compared with the background, those patterns which occur more frequently are more likely to be background patterns. Moreover, the more background patterns a block contains, the more likely it is a background block. Sometimes, foreground patterns may be included in the background patterns only with different distributions or different colors, and our assumption will be invalid in this case. Fortunately, this case does not often happen.

A background model will be calculated for each big block. Given a grayscale video sequence, let SILTP histograms of one big block over time $1, 2, \ldots, t$ be $H_s^1, H_s^2, \ldots, H_s^t$. Suppose B_s is the background model histogram of the big block based on SILTP information, and the models over time $1, 2, \ldots, t$ are $B_s^1, B_s^2, \ldots, B_s^t$. Let B_s^0 be the initial value of B_s, in which $B_s^0(1) = B_s^0(2) = \cdots = B_s^0(N_b) = 1/N_b$ ($N_b = 3^N$ is the number of the bins of the histogram). Then, B_s^t is updated as

$$B_s^t(i) = (1-\alpha)B_s^{t-1}(i) + \alpha H_s^t(i) \tag{4}$$

where $B_s^t(i)$ is the ith bin of the background model histogram at time t and α is the learning rate.

D. Moving Object Detection With SILTP Background Model

After background modeling, each big block will have a histogram as its background model. We further evaluate its probability of belonging to the background according to

$$P_b^b = \sum_{i=1}^{N_b} H_s(i) T\left(B_s(i), \frac{\eta}{N_b}\right) \tag{5}$$

where

$$T\left(B_s(i), \frac{\eta}{N_b}\right) = \begin{cases} 1, & \text{if } B_s(i) \geq \frac{\eta}{N_b} \\ 0, & \text{if } B_s(i) < \frac{\eta}{N_b}. \end{cases} \tag{6}$$

P_b^b is the background likelihood of the big block with a value between 0 and 1, and η is a variable controlling the value of the threshold η/N_b. Through this, we can get the background likelihood of all the big blocks of the image. Equation (6) shows that we consider those bins with the occurrence probability larger than η/N_b as dominant background patterns. Equation (5) shows that we get the summation of the occurrence probabilities of the selected dominant background bins as our final probability of the new coming big block belonging to the background. Since the initial value of each bin of the background model histogram is $1/N_b$, the value of the bins with their corresponding patterns appearing more times will be larger than $1/N_b$, and the value of the bins with their corresponding patterns appearing fewer times or even do not appear will be smaller than $1/N_b$. Intuitively said, $\eta = 1$ is a proper value choice, and we have tried different values of η, finding that $\eta = 1$ is indeed a proper value choice. In fact, the occurrence probabilities of most of the foreground patterns and unusual background patterns are smaller than $1/N_b$, so we use those patterns whose occurrence probabilities larger than $1/N_b$ to calculate the background probability of each block. Our single histogram model is able to handle the multimodal background-modeling problem, because all the patterns that appear frequently will be dominant in the model histogram.

The next step is to make the decision whether each small block belongs to the background or foreground. The probability of a small block belonging to the background can be obtained by averaging the background likelihood of all the big blocks it belongs to and the equation can be written as

$$P_b^s = \frac{\left(\sum_{i=1}^{n} P_b^{b(i)}\right)}{n} \tag{7}$$

where P_b^s is the background likelihood of the small block, n is the number of the big blocks which the small block belongs to and $P_b^{b(i)}$ is the background likelihood of the ith one. If $P_b^s > T_s$, we decide that this small block belongs to the background; otherwise, we judge that this small block belongs to the foreground. Here, T_s is the threshold for judging background and foreground.

III. Blockwise Background Model Based on Color Information

The SILTP feature has a shortcoming in dealing with smooth surface. Some common types of backgrounds (e.g., roads) may be smooth, and there will also be smooth foreground types like the body of cars or pedestrians with single-color clothes. Since SILTP features of smooth backgrounds and foregrounds are nearly the same, it is hard to detect smooth foregrounds from smooth backgrounds. In this case, color information is a good supplement. In addition, color information can also be useful in other cases where SILTP does not perform well, such as the case that most of the patterns of foreground parts belong to dominant background patterns but only have a different distribution. To make use of color information, we update a temporary background image and compare it with the new coming frame to judge which parts of the new coming frame are more likely to be background.

A. Temporary Background Image Updating

Once a new video frame comes, the temporary background image is updated according to the following equation:

$$T_b^t = \begin{cases} 0, & \text{if } t = 0 \\ (1-\beta)T_b^{t-1} + \beta T_N, & \text{if } t > 0 \end{cases} \tag{8}$$

where

$$\beta = \begin{cases} \frac{1}{We^{\ln(W)\frac{t-W}{W-1}}}, & \text{if } 1 \le t < W \\ \frac{1}{W}, & \text{if } t \ge W. \end{cases} \tag{9}$$

T_b^t is the tth temporary background image, t is the current frame index, T_N is the new coming frame, β is the updating rate, and W is the updating time window size.

B. Color Information Difference Calculation

When a new video frame comes, a color difference between each of its small blocks and the corresponding small block in the temporary background image is calculated. We first average the differences of each color channel of all the pixels in each small block, and then combine the differences of the three channels to get the final difference measure. The reason why we get the color difference of each small block instead of the big block is that calculation for overlapped big blocks is more time consuming and global color difference is already stable enough for each small block. Supposing there are N_s pixels in each small block, the difference measure of each small block is calculated with

$$D^r = \sum_{i=1}^{N_s} \left(C_b^r(i) - C_n^r(i)\right) \tag{10}$$

$$D^g = \sum_{i=1}^{N_s} \left(C_b^g(i) - C_n^g(i)\right) \tag{11}$$

$$D^b = \sum_{i=1}^{N_s} \left(C_b^b(i) - C_n^b(i)\right) \tag{12}$$

$$D = \left(\left(\frac{D^r}{N_s}\right)^2 + \left(\frac{D^g}{N_s}\right)^2 + \left(\frac{D^b}{N_s}\right)^2\right)\frac{1}{255 \cdot 255 \cdot 3} \tag{13}$$

where D^r is the summation of the r color channel difference between the small blocks of the new coming frame and the temporary background image. $C_b^r(i)$ is the r color channel pixel value of the ith pixel in the small block of the temporary background image, and $C_n^r(i)$ is the r color channel pixel value of the ith pixel in the small block of the new coming video frame. D is the final color information difference of the small block pair of the new coming video frame and the temporary background image. It can be seen from (10) that when computing the r color channel difference, we can first get the summation of the r channel pixel values of all the pixels in the small block, and then get the r color channel difference by subtracting the two summations of the new coming video frame and the temporary background image. This means we only extract the global color change of the small block, and moving background problems like tree waving can be better handled through this strategy. The final color information difference measure D has the value between 0 and 1.

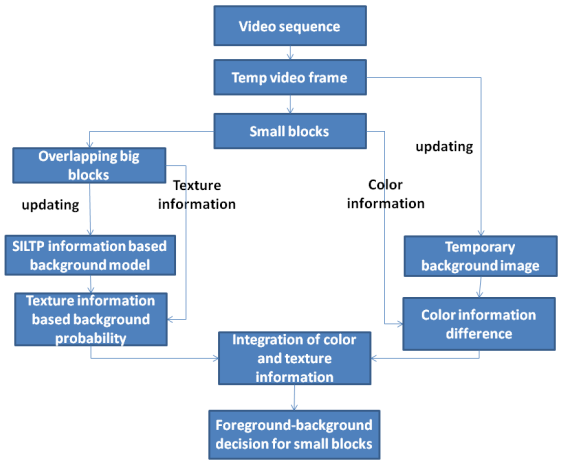


Fig. 4. Framework of our integrated moving object detection method.

IV. Integration of Color Information and SILTP Information for Moving Object Detection

Fig. 4 shows the framework of our integrated moving object detection method. On one hand, we update an SILTP information-based background model for each big block, and get the texture information-based background probability based on the model and the SILTP information of each big block in the new coming frame. On the other hand, we update a color information-based temporary background image, and get the color information difference based on the temporary background image and the color information of the new coming frame. Then, we integrate color information and texture information to get the foreground–background decision for each small block.

We have already got the background likelihood P_b^s of each small block of the new coming frame based on SILTP information and the color information difference D between the small blocks of the temporary background image and those of the new coming frame. Therefore, the only thing left is to make a final decision based on these two values. For SILTP information, we use the threshold T_s to judge whether the small block belongs to the background. For color information, we use the threshold T_c to judge whether the small block belongs to the background. The final decision can be made according to

$$\text{Decision} = \begin{cases} \text{foreground}, & \text{if } P_b^s < T_s, \text{ or } D > T_c, \\ & \text{or } (1-P_b^s)D > \frac{1}{\rho}(1-T_s)T_c \\ \text{background}, & \text{else} \end{cases} \tag{14}$$

where ρ is a scale factor. $(1 - P_b^s)D > (1/\rho)(1 - T_s)T_c$ means when both SILTP feature and color information change, the joint threshold can be smaller than the product of those two thresholds. Our purpose is that we can detect more whole moving objects while restraining false alarm rate at the same time. To detect more whole moving objects, we should increase T_s and decrease T_c, but this will make false alarm rate higher. Therefore, our joint judgement is crucial. We have realized the fact that both the color information and SILTP information may change obviously in the foreground region, while this case does not easily happen in the background region. Thus, the product of SILTP feature change and color information change can be very small in the background area, thus we are able to divide the joint judging threshold by ρ to detect more effective foreground areas, while false foreground detections in the background area can still be restrained. Fig. 5 shows the advantage of our joint judging strategy. $\rho = 1$ means the joint judging condition $(1 - P_b^s)D > (1/\rho)(1 - T_s)T_c$ does not make sense, and the comparison of Fig. 5(a) and (c), and Fig. 5(b) and (d) shows that a proper value of ρ can help detect more effective foreground parts while avoiding false detections at the same time. The reason why we do not use the joint judging strategy only is that sometimes only texture or color information is efficient while the other makes no difference, and in this case, the product of the SILTP feature change and color information change will be too small and unstable, leading to miss detection of the foreground parts. Therefore, all the three parts are needed.

Through this, we can get our final decision based on the integration of SILTP information and color information.

V. Quality Measure Design

Traditional quality measure for moving object detection is pixel based, which means it only cares about how many foreground pixels are correctly detected, and how many background pixels are incorrectly detected as foreground pixels. However, it does not work for moving object detection, and may even be improper. Fig. 6 shows some samples of moving object detection results. According to the traditional measure, the middle result image outperforms the left one, but in fact, in the left one, the moving object can be unbrokenly detected while it is detected broken in the middle one. As for the right one, high performance can be achieved with pixel-based quality measure, but several moving objects are actually detected as one. Therefore, the pixel-based quality measure is not proper for moving object detection since the goal of moving object detection is to detect moving objects separately without breaking.

The main purpose of our background-modeling process is to detect the whole of every moving object without breaking. Therefore, in the input frame, the best result is that there is a one-to-one correspondence between the groundtruth of the moving objects and the detected contiguous regions. In particular, if the foreground mask of one moving object in the detecting result is fractured into several parts, or the foreground masks of several moving objects are connected as one, the performance measure should be degraded.

Suppose the groundtruth of the foreground mask of the frames for evaluation is M^t, and the detection result of the foreground mask is M^d. All the contiguous regions of M^t and M^d are detected, and then, their bounding rectangles are obtained, named as B^t and B^d. The major issue of the evaluation process is to get the match degree between B^t and B^d. Before the matching process, some small bounding rectangles in B^d whose sizes do not meet the requirement of

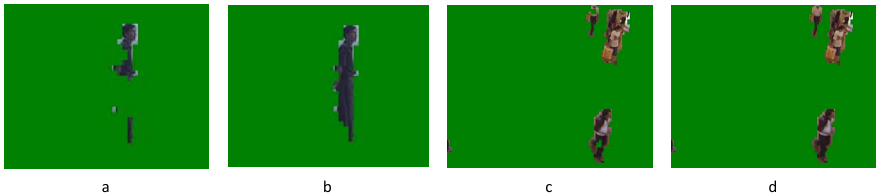


Fig. 5. Foreground detection results. (a) and (b) Detection results of one frame. (c) and (d) Detection results of another frame. $\rho = 1$ for (a) and (c) and $\rho = 9$ for (b) and (d).

an effective object are removed. This requirement is that the height and the width of the bounding rectangle should be both larger than $S_m (S_m = 4$ here) pixels. The matching function of one bounding rectangle in B^t with another bounding rectangle in B^d is designed as

$$\Phi(B_i^d, B_j^t) = \begin{cases} 1, & \text{if } (B_i^d \cap B_j^t) > \kappa(B_i^d \cup B_j^t) \\ 0, & \text{else} \end{cases} \tag{15}$$

where κ is a threshold indicating the minimum overlapping ratio needed for matching, B_i^d is the ith bounding rectangles of B^d, and B_j^t is the jth bounding rectangles of B^t. With background subtraction based on moving object detection methods, one bounding rectangle in B^t or B^d can be matched with one bounding rectangle at most in B^d or B^t when $\kappa \geq 0.5$. $\kappa = 0.5$ is used in our experiment, so the true positives of the detecting result can be calculated as

$$\text{TP} = \sum_{i,j} \Phi\big(B_i^d, B_j^t\big). \tag{16}$$

The false positives and false negatives of the detecting result can be further computed as

$$\text{FP} = \sum_i \phi\big(B_i^d\big) \tag{17}$$

where

$$\phi(B_i^d) = \begin{cases} 1, & \text{if } \sum_j \Phi(B_i^d, B_j^t) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

$$\text{FN} = \sum_j \varphi(B_j^t) \tag{19}$$

where

$$\varphi(B_j^t) = \begin{cases} 1, & \text{if } \sum_i \Phi(B_i^d, B_j^t) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

if $\phi(B_i^d) = 1$, it can be known that no matching is found in the groundtruth for the ith bounding rectangle of B^d. This is similar for $\varphi(B_j^t)$.

Table I shows the comparison of traditional quality measures with ours, which shows the superiority of our quality measure method for moving object detection.

VI. Experiments

Nine data sets are used, provided by [6] to evaluate the performance of our algorithm. These datasets consist of challenging videos containing busy human stream, dynamic background, and illumination variations. Most of the videos have several thousand frames, and 20 frames of each video are labeled as groundtruth. The frame resolution of *Bootstrap* is 160 × 120, and *Campus*, *Curtain*, *Fountain*, *Lobby*, and *WaterSurface* have the same frame resolution 160 × 128, and the frame resolution of *Escalator* is 160 × 130 and *ShoppingMall* is 320 × 256.

Fig. 6. Some samples of moving object detection results.

We compare our results with several state-of-the-art background modeling methods to show the efficacy of our algorithm. These methods include MoG [2], block-wise LBP histogram-based method (LBP-B) [17], pixelwise LBP histogram-based method (LBP-P) [10], and multiblock SILTP-based pattern kernel density estimation (PKDE) methods ($\text{PKDE}^{w=3}_{\text{mb-siltp}}$, $\text{PKDE}^{w=1+2+3}_{\text{mb-siltp}}$) [11]. Block-based background modeling method based on Integration of Texture and Color information is denoted by BITC.

For MoG, the standard implementation in OpenCV2.3.1 is used with default parameters. For LBP-B and LBP-P, we found better parameters than the suggested ones in our tested videos, and further tuned the parameters for every dataset to achieve nearly optimal results. Therefore, it will be more challenging for our algorithm to compare with LBP-B and LBP-P. For $\text{PKDE}^{w=3}_{\text{mb-siltp}}$ and $\text{PKDE}^{w=1+2+3}_{\text{mb-siltp}}$, the suggested parameters in [11] are used, because the algorithms perform very well with those parameters. The suggested parameters are: 1) $\text{SILTP}^{0.05}_{4,1}$ operator; 2) $K = 3$; 3) $T_b = 0.7$; 4) $T_s = 0.01$; 5) $T_m = 0.01$; and 6) $\alpha = 0.005$. For the proposed algorithm, fixed parameters for all the videos are applied, and these parameters are: 1) $\text{SILTP}^{0.05}_{4,1}$ operator; 2) $S_b = 8$; 3) $\alpha = 0.005$; 4) $\eta = 1$; 5) $T_s = 0.55$; 6) $W = 50$; 7) $T_c = 0.1$; 8) $\rho = 9$; 9) $\kappa = 0.5$; and 10) $S_m = 4$. The compared algorithms are tested on all the videos, and all the quality measures are obtained, including those defined in Section V.

Of all the parameters, S_b, T_s, T_c, and ρ play important roles in the performance of our algorithm. If S_b is too small, the histogram model of each big block is unstable and sensitive to noises, and more computation time and memory resource will be cost to get the background model compared with bigger S_b. To the contrary, if S_b is too large, too much local information will be lost and the classification will be too coarse. For static background scenes, T_s can be larger to detect more

TABLE I
COMPARISON OF TRADITIONAL QUALITY MEASURE METHOD WITH OURS

Quality measure	Object	Object breaking	Object adhesion	High-level information	Moving object detection
Traditional	pixel	can not reflect	can not reflect	do not have	not suitable for
Ours	region	can reflect	can reflect	do have	suitable for

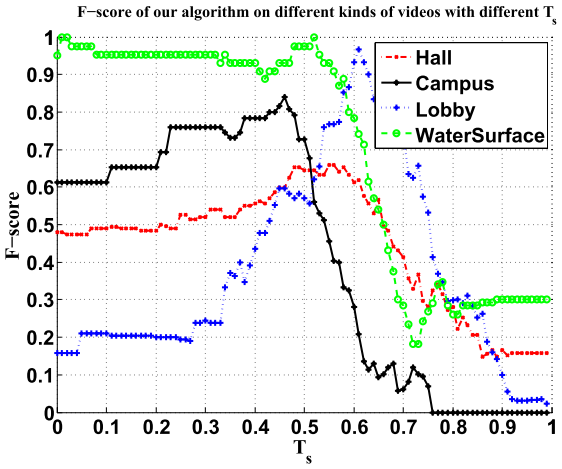


Fig. 7. F-score of our algorithm on different kinds of videos with different T_s.

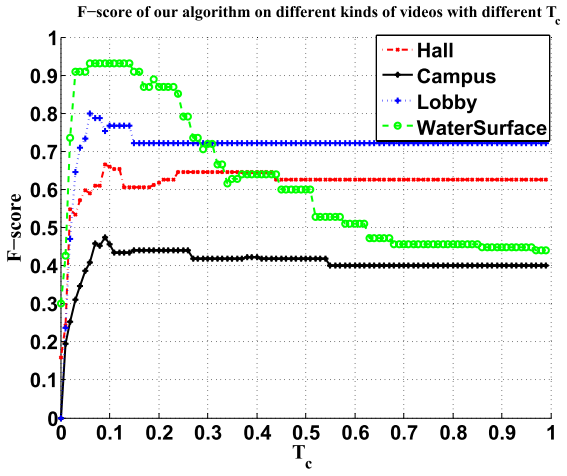


Fig. 8. F-score of our algorithm on different kinds of videos with different T_c.

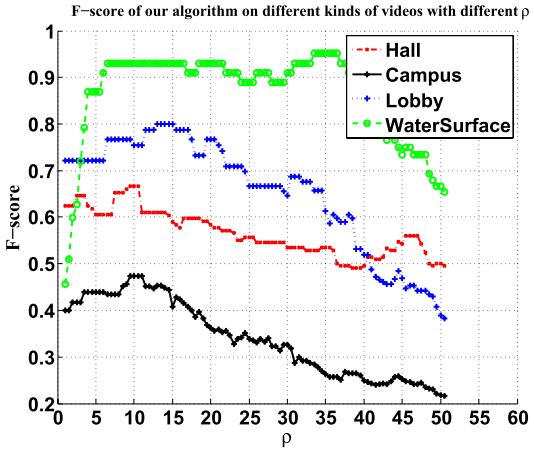


Fig. 9. F-score of our algorithm on different kinds of videos with different ρ.

foreground parts. For dynamic background scene, a large T_s will result in many background parts being detected as the foreground while a small T_s will result in incomplete detection of the foreground. For scenes without frequent illumination variations, T_c could be smaller to detect more foreground parts. For scenes with frequent illumination variations, T_c should be lager to avoid false foreground detection in the background region since color information is just a supplement in the proposed moving object detection. A large ρ can help detect more valid foreground parts but will result in higher false alarm rate while a small ρ loses some foreground parts.

To determine suitable parameters, the performances of different values of parameters T_s, T_c, and ρ which directly affect our decision on various kinds of videos are analyzed. Four videos are used here: 1) *Hall* for indoor busy scene with moving cast shadows; 2) *Campus* for significant dynamic background scene; 3) *Lobby* for indoor scene with a sudden illumination change; and 4) *WaterSurface* for general dynamic background scene. Figs. 7–9 show the *F-score* (23) results. When considering the performance of one parameter with different values, the other parameters are fixed as before. We first keep two parameters fixed and observe the performance line of the left parameter in all kinds of videos to get a good value of this parameter, then we do the same for the other two parameters to get coarse good values. Gradually, by experimenting many times, we can get the real near optimal values of all the three parameters. From Fig. 7, we can see that the peak point of *Campus* has the smallest T_s, followed by *WaterSurface*, *Hall*, and *Lobby*. The result is consistent with the fact that the more severely the background changes, the smaller T_s should be fixed to achieve good performance (though *Lobby* contains sudden illumination change, the background is very stable actually, and the illumination variation can be adapted after a little while). The *F-score* of our algorithm here mainly relies on the color information when T_s is very small, and the *F-score* rises when T_s increases from a small value for videos *Hall*, *Campus*, and *Lobby*, which means SILTP information is a good supplement for color information for these kinds of videos. The reason why SILTP information does not help for video *WaterSurface* is that background and foreground are both very smooth and of strong color contrast. $T_s = 0.55$ is just a balance between all kinds of videos. Since a bigger T_c will result in less influence of color information on the performance of our algorithm, color information does have benefits considering the result in Fig. 8. However, a too small T_c will result in many false detections, so we choose $T_c = 0.1$ as a tradeoff. From Fig. 9,

it can be seen that neither a small value of ρ nor a big value of ρ is a good choice. We just choose $\rho = 9$ as a tradeoff also.

The next step is to evaluate the performances of all the algorithms on all the nine video sequences. *Recall*, *Precision*, *F-score*, memory usage and *Framerate* will all be considered. TP, FP, and FN are already given in Section V. Then, *Recall*, *Precision*, and *F-score* can be obtained by the following equations:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{22}$$

$$\text{F-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{23}$$

All the experiments are done on a standard PC with 2.93-GHz CPU(dual core), 4G memory, and Windows 8 operation system. All the algorithms are implemented in C++. Table II shows the quality measures for all the algorithms on the test videos of all the methods for comparison. Besides, Table III shows the average performance of all the algorithms on the nine videos.

From Table III, it can be seen that the average performance of our algorithm is outstanding. *Recall* is nearly the best, *Precision* and *F-score* are much better than those of other methods, respectively, memory usage is the lowest, and frame rate is only lower than that of MoG. From Table II, we can see that the memory usage of our method is the smallest for all the videos. The frame rate of MoG is the highest, but $\text{PKDE}^{w=3}_{\text{mb-siltp}}$ and our BITC also perform well. Particulary, we can see that the frame rate of our algorithm in *ShoppingMall*(320×256) is 63 frames/s, which is super real time. Considering the detection performance, the only video on which our method performs much worse than other methods is *Campus*, which contains strongly swinging trees. It has already been discussed before that T_s should be smaller when there exists dynamic background. The more severely the background varies, the smaller T_s should be. Our fixed parameter can handle general dynamic background modeling problem well, which can be seen from videos *Curtain*, *Escalator*, *Fountain*, and *WaterSurface*. However, the trees in *Campus* wave so strongly that the fixed $T_s = 0.55$ is not capable to handle this well. We further tune the parameter T_s to 0.5 and find that the result is improved greatly with *F-score* 0.8136. In *WaterSurface*, LBP-P is a little better than our method in *Recall*, *Precision*, and *F-score*, but its parameters have been tuned to nearly optimal. When T_s of our method is tuned to 0.5, the *F-score* is 0.9756, better than LBP-P. Here, just for the sake of fairness, we use the same parameters for comparison. By the way, we cannot ensure that the *F-score* of our method will always outperform other methods in all cases. What we are doing is to ensure that our method can perform better in most of the cases. Moreover, we get high performance while keeping low memory cost and high processing speed of our method.

We further analyze the performance of our method on different kinds of videos. Videos *Curtain*, *Fountain*, and *WaterSurface* are very similar, since all of them contain big moving objects and dynamic background. The recalls of our method in these three videos are all 1.0 while the precisions are also high, so we can see that big moving objects can be easily detected in this kind of video while there may exist a few false detections of foreground in the background region. Videos *Bootstrap*, *Escalator*, *Hall*, and *ShoppingMall* all have busy pedestrians and moving cast shadows. Though our algorithm performs the best on these four videos, the absolute performance still needs improvement except for that in *ShoppingMall*, which has good illumination conditions and viewing angle. In such busy scene, adjacent moving objects are easily to be detected as one, and one moving object may be split when the illumination condition is bad or when there exist moving cast shadows. Video *Lobby* contains turning OFF and turning ON of the light, but our method can still perform well in such cases, which indicates that our method can adjust to sudden illumination variation quickly.

To analyze the moving object detection result more concretely, we further give qualitative foreground detection results shown in Fig. 10. One typical frame is picked out for each video to show the varied performances of our compared algorithms on it.

For the picked frames of videos *Curtain*, *Fountain*, and *WaterSurface* which contain big moving objects and dynamic background, our BITC method achieves good performance on all the three, and the only big moving object of each video is detected unbroken with no false detection. The foreground and background of the frame of video *Curtain* has strong contrast in color information while the texture information tends to be similar in some parts, and this leads to the good performances of MoG and our BITC. In contrast, LBP-B and LBP-P lose much foreground part resulting in the breakage of the only foreground object and $\text{PKDE}^{w=3}_{\text{mb-siltp}}$ and $\text{PKDE}^{w=1+2+3}_{\text{mb-siltp}}$ have holes (though we do not care about holes in our quality measure design, it is still not good to have holes since foregrounds with holes are more likely to be fractured) in the human body. The foreground object in the fountain frame has been standing still for a short while. Pixel information-based methods MoG, $\text{PKDE}^{w=3}_{\text{mb-siltp}}$ and $\text{PKDE}^{w=1+2+3}_{\text{mb-siltp}}$ are easily to lose foreground parts which will make the foreground object fractured because each pixel is judged separately. In contrast, block information-based methods LBP-B, LBP-P, and our BITC are able to detect the whole foreground object since the information of a big block is considered together. The down part of the feet in the result of MoG is missing due to the similarity of color information between background and foreground there, and the foreground is fractured. After observing the results of these three videos, we can find that our integration framework of texture information and color information is effective and is beneficial to the results, and that block-based information is more stable.

For the picked frames of videos *Bootstrap*, *Escalator*, *Hall*, and *ShoppingMall* which are indoor scenes with busy pedestrian stream and moving cast shadows, our BITC method works well with few false detections except for that of *Escalator* whose foreground is too busy which leads to the connection of different moving objects. Moreover, most of the cast shadows are removed by our method. The MoG

TABLE II
DETECTING RESULT QUALITY MEASURES FOR ALL THE TEST VIDEOS OF ALL THE METHODS FOR COMPARISON

Algorithms		MoG [2]	LBP-B [17]	LBP-P [10]	$PKDE^{w=3}_{mb-siltp}$ [11]	$PKDE^{w=1+2+3}_{mb-siltp}$ [11]	BITC
Bootstrap	Recall	0.4341	0.4828	0.3793	0.5345	**0.6552**	0.6545
	Precision	0.3944	0.5833	0.3548	0.4627	0.5278	**0.6429**
	F-score	0.4828	0.5283	0.3667	0.4960	0.5846	**0.6486**
	Memory usage	9.9	11.6	39.8	19.1	179	**7.2**
	Frame rate	**390**	150	12.4	265	33	245
Campus	Recall	0.7214	0.6207	0.6552	0.7241	**0.8276**	0.7241
	Precision	0.6000	**0.8182**	0.7037	0.2917	0.3111	0.3333
	F-score	0.6563	**0.7059**	0.6786	0.4158	0.4528	0.4565
	Memory usage	10.1	11.8	42.5	20.1	189	**7.4**
	Frame rate	**320**	135	11.6	225	30	225
Curtain	Recall	0.7000	0.8000	0.3000	**1**	**1**	**1**
	Precision	0.3182	0.5517	0.1538	0.6667	0.9091	**1**
	F-score	0.4375	0.6531	0.2034	0.8000	0.9524	**1**
	Memory usage	10.1	11.8	42.5	20.1	189	**7.4**
	Frame rate	**330**	150	12.2	250	32	**330**
Escalator	Recall	0.3333	0.3333	0.5000	0.6111	**0.6852**	0.6400
	Precision	0.2118	0.4091	0.6429	0.4853	0.4458	**0.6275**
	F-score	0.2590	0.3673	0.5625	0.5410	0.5401	**0.6337**
	Memory usage	10.2	12.0	46.6	20.5	191	**7.4**
	Frame rate	**330**	142	11.4	240	32	245
Fountain	Recall	0.1500	0.9500	0.8500	0.9500	0.9500	**1**
	Precision	0.0517	0.8261	0.7391	0.7037	0.7917	**0.9091**
	F-score	0.0765	0.8837	0.7907	0.8085	0.8636	**0.9524**
	Memory usage	10.1	8.4	46.0	20.1	188	**7.4**
	Frame rate	**300**	150	121.5	235	32	240
Hall	Recall	0.4643	0.3750	0.3929	0.5536	0.5536	**0.5714**
	Precision	0.3824	0.4773	0.3607	0.6200	0.4844	**0.7805**
	F-score	0.4194	0.4200	0.3761	0.5849	0.5167	**0.6598**
	Memory usage	10.9	12.6	56.1	23.7	232	**8.2**
	Frame rate	**270**	120	8.8	200	25	220
Lobby	Recall	0.1333	0.7333	0.7667	0.7667	0.7333	**0.7667**
	Precision	0.1379	0.6875	0.6765	0.7188	0.6667	**0.7667**
	F-score	0.1356	0.7097	0.7188	0.7419	0.6984	**0.7667**
	Memory usage	10.1	11.9	45.9	20.1	188	**7.4**
	Frame rate	**330**	150	11.7	245	32	250
ShoppingMall	Recall	0.5246	0.6721	0.5738	0.7213	0.8115	**0.8571**
	Precision	0.2909	0.7069	0.5645	0.6519	0.6972	**0.7786**
	F-score	0.3743	0.6891	0.5691	0.6848	0.7500	**0.8160**
	Memory usage	20.9	23.6	180.6	67.7	741	**17.7**
	Frame rate	**82**	31	2.25	59	7.7	63
WaterSurface	Recall	**1**	**1**	**1**	**1**	**1**	**1**
	Precision	0.8333	0.8333	**0.9091**	0.7143	0.8333	0.8696
	F-score	0.9091	0.9091	**0.9524**	0.8333	0.9091	0.9302
	Memory usage	10.1	10.5	45.9	20.1	188	**7.4**
	Frame rate	**330**	227	11.5	240	32	245

misses a lot of foreground parts in all the picked frames of these videos since it cannot handle illumination variations and the color difference between foreground and background is small. The LBP-B method can detect most of the foreground parts, but many background parts around the boundary of the foreground are detected as the foreground, which may lead to the connection of adjacent moving objects. In addition, there also exist some false detections and foreground fractures.

TABLE III
AVERAGE DETECTING RESULT QUALITY MEASURES OF ALL THE TEST VIDEOS OF ALL THE METHODS FOR COMPARISON

Algorithms		MoG [2]	LBP-B [17]	LBP-P [10]	$PKDE^{w=3}_{mb-siltp}$ [11]	$PKDE^{w=1+2+3}_{mb-siltp}$ [11]	BITC
Average	Recall	0.4957	0.6630	0.6020	0.7624	**0.8018**	0.8015
	Precision	0.3578	0.6548	0.5672	0.6077	0.6297	**0.7454**
	F-score	0.4168	0.6518	0.5798	0.6562	0.6964	**0.7627**
	Memory usage	11.4	12.7	60.7	25.7	254	**8.6**
	Frame rate	**298**	139	10.4	219	28	229

Fig. 10. Moving object detection results of all the compared algorithms on selected frames from all the tested videos.

LBP-P can achieve better foreground boundary than LBP-B, but is still not good enough. Furthermore, when some parts of the foreground are similar to the background in LBP information, the resulting detected foreground may be cataclastic as observed in the result of *Bootstrap*. $\text{PKDE}^{w=3}_{\text{mb-siltp}}$ and $\text{PKDE}^{w=1+2+3}_{\text{mb-siltp}}$ perform well in these video frames. Compared with these two methods, our BITC does better in dealing with the moving cast shadow problem. As shown in the detection results of *ShoppingMall*, some cast shadows become separate false detections by $\text{PKDE}^{w=3}_{\text{mb-siltp}}$ and $\text{PKDE}^{w=1+2+3}_{\text{mb-siltp}}$, but this does not happen in our BITC method. One foreground object in the middle is broken by our method, because both the color information and the SILTP information of the feet of that pedestrian are similar to those of the background there. All the methods do not perform well in the selected frame of *Escalator*. The block-based methods tend to connect different moving objects because these objects are too small and close to each other, while the pixel information-based methods tends to miss much foreground part since the color information or the texture information of some foreground parts is similar to the background there. The detection result of the frame of *Hall* also shows that block-based methods are easily to connect different moving objects than pixel-based methods. To sum up, our BITC method can detect most of the foreground parts in busy scenes and can handle the moving cast shadow problem but tends to connect different moving objects when they are too close to each other.

From the picked frame of video *Lobby*, which is an indoor scene with light switching, we can see that all the methods can adapt to global illumination variation after a while. The two

Fig. 11. Foreground detection result of the picked frame of video *Campus* by our BITC method when T_s is tuned to a smaller value.

pedestrians in the frame have been standing there for a little while, and as stated before, the block information-based method trends to be more stable in such conditions since the information of a block is considered together.

The last video is Campus, which contains strongly swinging trees. The detection result of our method is not good based on the chosen fixed parameters since there exist many false detections. However, the result can be much better when we tune the parameter T_s to a smaller value, as shown in Fig. 11. For practical use, we can further figure out a rule for the tuning of T_s and T_c for the users to handle the extreme conditions.

VII. Conclusion

In this paper, we have proposed a fast blockwise background modeling algorithm with the integration of SILTP and color information. A block-based model with single SILTP histogram has been proposed and is able to handle dynamic background and multimodal problems. Dominant background patterns are selected from the SILTP histogram model for calculating the background likelihood of the new coming block. A detection judgement is given on smaller blocks to get more accurate detection boundary than judging big blocks. A temporary background image is updated for the calculation of the color information change of each small block in the new coming frame. The SILTP information and color information have been integrated for much more effective detection of moving objects than separately applied. A new quality measure is proposed for evaluating the performance of our method on various challenging videos, and the result is quite outstanding compared with the other state-of-the-art methods. The memory consumption is low while the processing speed can be superreal time in videos of resolution 320×256. Further detailed analysis shows that our method is robust to illumination variations, dynamic background, and moving cast shadow problems.

References

[1] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 1999.

[3] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in *Computer Vision*. Berlin, Germany: Springer-Verlag, 2002, pp. 543–560.

[4] D.-S. Lee, J. J. Hull, and B. Erol, "A Bayesian framework for Gaussian mixture background modeling," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 3. Sep. 2003, pp. III-973–III-976.

[5] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[6] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 2–10.

[7] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. Int. Conf. Image Process.*, vol. 5. Oct. 2004, pp. 3061–3064.

[8] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, 2005.

[9] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1305–1312.

[10] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[11] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1301–1306.

[12] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.

[13] Z. Zhang, C. Wang, B. Xiao, S. Liu, and W. Zhou, "Multi-scale fusion of texture and color for background modeling," in *Proc. IEEE 9th Int. Conf. Adv. Video Signal-Based Surveill. (AVSS)*, Sep. 2012, pp. 154–159.

[14] E. Learned-Miller, M. Narayana, and A. Hanson, "Background modeling using adaptive pixelwise kernel variances in a hybrid feature space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2104–2111.

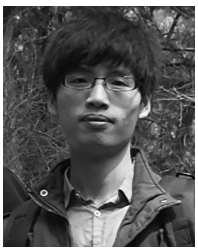
[15] T. Matsuyama, T. Ohya, and H. Habe, "Background subtraction for non-stationary scenes," in *Proc. Asian Conf. Comput. Vis.*, 2000, pp. 662–667.

[16] M. Mason and Z. Duric, "Using histograms to detect and track objects in color video," in *Proc. 30th Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2001, pp. 154–159.

[17] M. Heikkilä, M. Pietikäinen, and J. Heikkilä, "A texture-based method for detecting moving objects," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1. 2004, pp. 187–196.

Hong Han (M'06) was born in 1974. She received the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003.

She is a Senior Researcher with the School of Electronic Engineering, Xidian University, Xi'an. Her research interests include computer vision, information fusion, and machine learning.

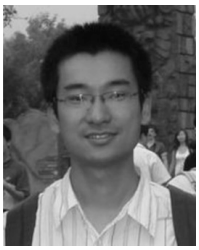


Jianfei Zhu received the M.S. degree in pattern recognition and intelligent systems from the School of Electronic Engineering, Xidian University, Xi'an, China, in 2014.

He is an Algorithm Engineer with the Alibaba Group, Hangzhou, China. His research interests include computer vision, pattern recognition, and image processing, with a focus on face recognition, video analysis, object detection, and classification.

Shengcai Liao (M'13) received the B.S. degree in mathematics and applied mathematics from Sun Yat-sen University, Guangzhou, China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010.

He was a Post-Doctoral Fellow with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA, from 2010 to 2012. He is currently an Associate Professor with CASIA. His research interests include computer vision and pattern recognition, with a focus on face recognition and intelligent video analysis.



Zhen Lei (S'08–M'11) received the B.S. degree in automation from University of Science and Technology of China, Hefei, China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010.

He is an Associate Professor with CASIA. He has authored over 80 papers in international journals and conferences. His research interests include computer vision, pattern recognition, image processing, and in particular, face recognition.

Dr. Lei was the Area Chair of the International Joint Conference on Biometrics in 2014, the IAPR/IEEE International Conference on Biometric in 2015, and the IEEE International Conference on Automatic Face and Gesture Recognition in 2015.

Stan Z. Li (M'92–SM'99–F'09) received the Ph.D. degree from Surrey University, Surrey, U.K.

He is a Professor and the Director of the Center for Biometrics and Security Research with Institute of Automation, Chinese Academy of Sciences, Beijing, China. He has authored over 200 papers in international journals and conferences, and authored and edited eight books. His research interests include pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance.