FOREGROUND DETECTION IN SURVEILLANCE VIDEOS VIA A HYBRID LOCAL TEXTURE BASED METHOD

Xiaojing Du, Guofeng Qin*
College of Electronics and Information Engineering,
Tongji University, Shanghai, China
Emails: *gfqing@tongji.edu.cn

Submitted: June 13, 2016 ***Accepted: Oct.1, 2016*** ***Published: Dec.1, 2016***

Abstract- Foreground detection is a basic but challenging task in computer vision. In this paper, a novel hybrid local texture based method is presented to model the background for complex scenarios and an image segmentation based denoising processing is applied to reduce noise. We combine the uniform pattern of eXtended Center-Symmetric Local Binary Pattern (XCS-LBP) and Center-Symmetric Local Derivative Pattern (CS-LDP) to generate a discriminative feature with shorter histogram. Retaining the strengths of the two textures, it appears to be robust to dynamic scenes, illumination changes and noise. Based on the hybrid feature, we employ an overlapping block based Gaussian Mixture Model (GMM) framework which makes classifying decision in pixel level. Experimental results on two changeling datasets (Wallflower and I2R dataset) clearly justify the performance of proposed method. Besides, we take the foreground masks obtained by proposed method as input to a tracking system showing notable results.

Index terms*:* Foreground detection, background modeling, derivations of local binary pattern.

I. INTRODUCTION

As people's growing awareness of security, quantities of surveillance cameras are installed in public, for instance, airports, highways, railway stations, etc. Comes along the problem that how to analyze enormous surveillance videos and apply them into practical situations. The major applications, such as object detection and tracking [1], vehicle type detection [2], pedestrian detection [3], anomaly detection [4], focus on the moving objects (usually people and vehicles) in captured videos. Thus, detecting the foreground objects, being a crucial procedure of these applications, is extremely significant.

In the literature, the most popular foreground segmentation algorithms are based on background modeling (background subtraction, BGS). It is conceivable that the problem would be much easier with a known background. Ideally, each frame of video sequences contains nothing moving except foreground objects, and the stationary background would be directly obtained. Nevertheless, the real scene would be more complex for moving trees, changing illumination, noise, weather, adding difficulties to acquire a robust and adaptive background model. Previous researches [5], [6] have proved that background modeling based methods tend to show a stably better performance. Background modeling tends to utilize the first n frames of video sequences to train a reference of image background. Then the foreground objects would be quite easily acquired by simple comparison between the current frame and trained background. Since background in real world would change over time, the obtained reference should be updated correspondingly.

Before establishing a background model, we would have to figure out the feature and minimize unit to be employed. Commonly used features are color features, edge features, motion features and texture features [7]. Among them, color features are not uncommon due to their simple and explicitness. Nevertheless, they are sensitive to noise and illumination change. Motion features provides temporal information between frames. Most texture features are insensitive to illumination changes and shadows. The minimize units may be a pixel [8], a block [9] or a cluster [10]. Different units have its peculiar properties: pixel based modeling methods would be easily affected by noise and computationally intensive; block-based methods turn out to be more robust to noise, but the detected foreground objects always have rough edges.

This paper presents a novel hybrid local texture descriptor based background modeling method for foreground detection in complex scenes. EXtended Center-Symmetric Local Binary Pattern (XCS-LBP) descriptor being a novel derivation of Center-Symmetric Local Binary Pattern (CS-LBP) is first proposed in [11]. It shows a preferable performance in most occasions, but tends to fail in dynamic scenes. Thus we extend it into a more discriminative feature with Center-Symmetric Local Derivative Pattern (CS-LDP) and apply it into a robust background modeling framework. Moreover, we introduces superpixel segmentation into image denoise. Segmentation based denoising algorithm enables to retain more details and edges information than some traditional methods. Generally, the proposed method works as follows. First, operate denoising processing on each frame; then train an image sequence in block level to obtain background reference; compare the incoming frame with obtained reference and finally classify every pixel into background or foreground.

Rest of the paper is organized as follows. Section Ⅱ briefly describes previous work of background modeling. Section Ⅲ would introduce the proposed method in detail. Section Ⅳ presents experimental results and comparison with other algorithms. The last section summarizes the paper and suggests some possible future directions.

II. RELATED WORKS

Background modeling algorithms aim at obtaining a robust and effective background reference for video sequences captured by stationary cameras. Here we would like to introduce some notable methods briefly. Single Gaussian method [12] models each pixel with a Gaussian distribution, calculates the mean and standard deviation of each pixel, and classifies a pixel into foreground when its value is larger than the selected threshold. Whereas, it is prone to fail when coping with the situation containing dynamic factors (swaying branches, flowing water). Thus, here comes Gaussian Mixture Model (GMM) [8], which models each pixel with k Gaussian distributions and continuously updates the weights of each distribution using a learning rate. Since then, many improvements [5] of GMM spring up focusing on the optimizations of the number of Gaussians, learning rate, and threshold. In [13], the authors propose an algorithm to simultaneously select the number of Gaussian; in [14], the authors propose to use particle swarm

optimization to tune the learning rate and threshold. However, the assumption that pixel intensity submits to Gaussian distribution is limited.

Despite the density-based method, other features like edge histograms, Discrete Cosine Transform (DCT) coefficients, texture features are employed to background modeling as well. In [15], the authors divide each frame into blocks, compute edge histograms and compare differences between the current frame and background. In [16], the authors divide image into blocks with size of 4×4 and utilize the corresponding DCT coefficient as eigenvector. In [11], the authors propose an extended CS-LBP feature to model background. Here we concentrate on LBP feature. LBP has been attracting many researchers' attention for its simplicity and easiness to compute. It is first proposed by [16] to describe texture features. The original version of LBP chooses a 3×3 window via comparing the gray value of central pixel with every neighbor pixels to encode the central one. Although LBP is invariant to gray-scale changes, it is changeable when rotates. One useful extension to the original LBP is uniform LBP (ULBP), which is capable of reducing the length of the feature vector. A pattern that conforms to ULBP contains at most two bitwise transitions. Another notable extension is CS-LBP, which compares the gray level of pairs of pixels in centered symmetric directions. CS-LBP produces shorter histograms and more robust to noise. The CS-LDP put forward in [17] captures more detailed information and generates histogram with the same length as that of CS-LBP.

More related to our work is the notable effort in [11], [1]. In [11], the authors propose XCS-LBP. Its novelty lies in introducing central pixel and avoiding defining a threshold value which is inevitable in CS-LBP. But it is not robust enough to deal with complex scenes. Besides, traditional pixel-based methods are prone to neglect rich contextual information, while those block-based methods often acquire foreground objects with coarse edges. In pioneering work [1], the authors propose an overlapping block-based classifier cascade modeling method, which models for blocks and makes final decision in pixel level. It demonstrates great performance in many situations. But it is incapable of dealing with the situations where foreground objects shields background in training sequences. One of the primary motivations of our work is to figure out these limitations and develop a more robust framework to eliminate them.

Our method innovatively combines two features into a more discriminative one generating shorter histograms and robust enough when confronting with challenging situations. Moreover, we take pixel dependency into consideration instead of treating the pixels as individuals by

employing overlapping blocks. Each block is modeled by GMM and final pixel classification is made based on the number of blocks containing that pixel classified into background or foreground. We also introduce image segmentation into denoise to reduce the effect of noise while retaining details as much as possible.

III. PROPOSED METHOD

The proposed foreground detection method has three main components:

1) Operate denoise process on each frame, using superpixel segmentation and median filter.

2) Generate hybrid histogram by concatenating the uniform XCS-LBP and uniform CS-LDP histograms.

3) Establish the proposed overlapping block-based modeling method and generate foreground mask for each frame in pixel level.

a. Denoise Processing

Generally, surveillance videos shot outdoors takes lager risk that they might be contaminated by noise. Unfortunately, noise devalues the quality and accuracy of latent image to some extent. The descriptors adopted in proposed method might be sensitive to noise. Thus, denoise operation is adopted as a pre-processing procedure. Traditional denoise solutions are average filter and median filter, both of which blur the edges and details. Thus the proposed approach introduces image segmentation into denoising process. Segmentation aims to divide a digital image into several regions (superpixel). More precisely, segmentation is to assign each pixel a label, and pixels with the same label share certain characteristics. The boundaries of superpixel imply some changes happening between different regions, which should be persevered during denoise operation. The efficient SLIC (Simple Linear Iterative Clustering) [18] algorithm is chosen to process segmentation for it is fast and preserves well edges.

First operate SLIC algorithm on a given image and obtain labels of every pixel; then process each superpixel region with median filter on the whole image with Equation (1) on the condition that the pixels in a template (e.g. $3\times3, 5\times5$) share the same superpixel label, which makes it possible for us to reduce noise and preserve the significant information included by boundaries.

(a) (b) (c) (d)

Figure 1 (a) Original image; (b) Image denoised via average filter with PSNR = 29.995; (c) Image denoised via median filter with PSNR = 31.873; (d) Image denoised via proposed method with PSNR = 32.850.

$$g(x,y) = med\{f(x-k, y-l), (k,l \in W)\} \tag{1}$$

In Equation (1), $f(x,y)$ is the original image; W is 2D template; $g(x,y)$ is the output image.

Figure 1 shows the comparisons between proposed denoise method and two traditional algorithms, average filter and median filter. The original image being segmented into 300 regions by SLIC is operated by proposed denoise method with a 3×3 template. And the proposed method exhibits a promising result with a higher ratio of peak signal to noise (PSNR) than that of other two methods.

b. Hybrid Descriptor

1. Uniform XCS-LBP

CS-LBP, first proposed in [19], compares the value of pixel-pairs in centered symmetric directions, encoding central pixel with a sequence of binary. A pixel c located at (x_c, y_c) is being encoded as:

$$CS-LBP_{P,R}(c) = \sum_{i=0}^{P/2-1} S(G_i - G_{i+P/2})2^i \tag{2}$$

In Equation (2), P is the number of equally spaced pixels on a circle with radius R and center (x_c, y_c), $G_i (i = 0,..., P-1)$ represents the gray values of the P pixels and S being the threshold function is defined as:

$$S(x) = \begin{cases} 1 & x > T \\ 0 & otherwise \end{cases} \tag{3}$$

In equation above, T is a threshold defined by users. Comparing to original LBP, CS-LBP generates shorter histogram ($2^{P/2}$), adds local contrast information and shows robust performance.

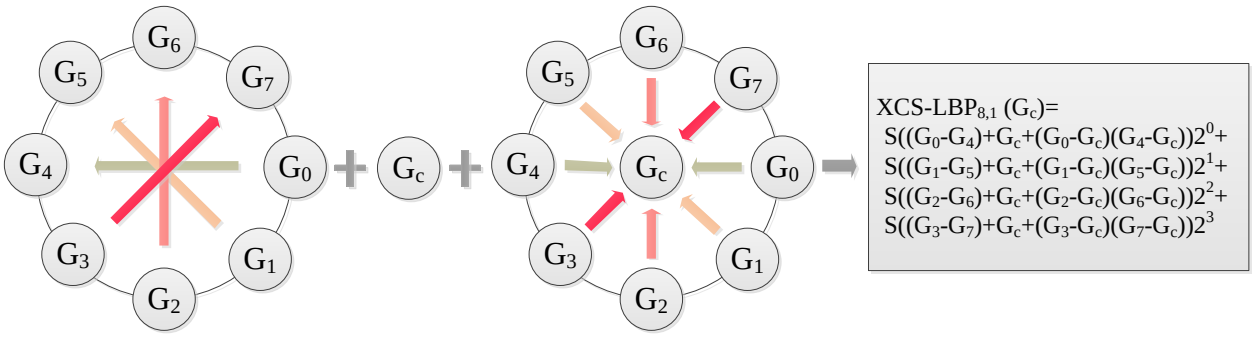


Figure 2 the computation of XCS-LBP descriptor.

However, users have to define the threshold T by themselves, which is quite inconvenient. Similar with uniform CS-LBP proposed in [20], we propose a uniform XCS-LBP (UXCS-LBP) descriptor. XCS-LBP first proposed in [11] differs from original CS-LBP for considering central pixel and adopting a novel threshold function to determine the types of pattern transition. XCS-LBP can be expressed as:

$$XCS-LBP_{P,R}(c)=\sum_{i=0}^{P/2-1}S(G_1(i,c)+G_2(i,c))2^i \tag{4}$$

In Equation (4), the threshold function S is defined as:

$$S(x_1+x_2)=\begin{cases}1 & x_1+x_2\geq 0\\ 0 & otherwise\end{cases} \tag{5}$$

In Equation (5), $G_1(i,c)$ and $G_2(i,c)$ are defined as:

$$G_1(i,c)=G_c+G_i-G_{i+P/2} \tag{6}$$

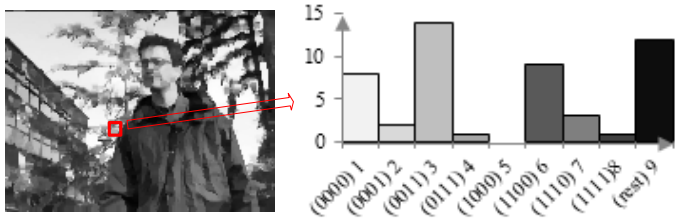
$$G_2(i,c)=(G_i-G_c)(G_{i+P/2}-G_c) \tag{7}$$

G_i, P has the same notation with Equation (2), and G_c is the central pixel's value.

XCS-LBP ingeniously avoids the threshold selection problem and generates the same length histogram as CS-LBP. The encoding procedure of XCS-LBP for a neighborhood size of 8 is illustrated in Figure 2. To transform XCS-LBP into UXCS-LBP, we adopt a rule that a uniform XCS-LBP pattern has no more than one bitwise transition between 0 and 1, which is first formulated in [20] for uniform CS-LBP. For instance, 0011 (one transition) and 0000 (zero transition) are uniform, while 0100 (two transitions) is non-uniform. The uniform decisional function is defined as:

$$U(XCS-LBP_{P,R}(c))=\sum_{i=2}^{P/2}|b_i-b_{i-1}| \tag{8}$$

In Equation (8), $P/2$ denotes the length of binary sequence generated by XCS-LBP, b_i is the binary number in sequence. If $U(x) \le 1$, x is a uniform binary pattern, but otherwise a non-uniform one. The length of histogram extracted by UXCS-LBP equals the number of uniform patterns plus 1 (represents the rest of non-uniform patterns). Specifically, the histogram extracted by $UXCS-LBP_{8,1}$ is illustrated in Figure 3. The original 16 dimensional histogram extracted by XCS-LBP for 8 neighborhoods is reduced into 9 dimensions containing 1 non-uniform and 8 uniform bins.



(a) (b)

Figure 3 (a) The original image with a 8×8 block (red region) to extract UXCS-LBP histogram. (b) Denote the extracted histogram with 8 uniform binary sequences and 1 non-uniform (the rest 8 kind) bins.

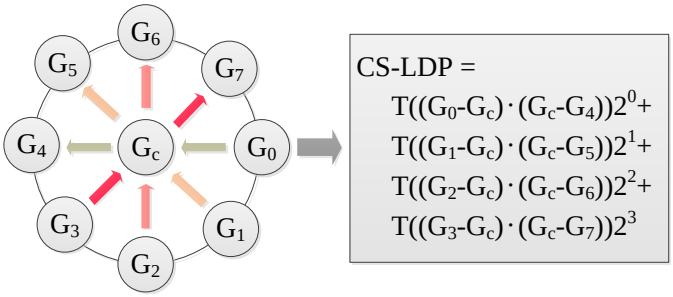


Figure 3 the computation of CS-LDP descriptor

2. Uniform CS-LDP

CS-LDP being a higher order derivative pattern captures more detail information. CS-LDP compares pixel pairs in centered symmetric directions similar to CS-LBP, but it considers central pixel. Specific encoding process is described as:

$$CS-LDP_{P,R}(c) = \sum_{i=0}^{P/2-1} T[(G_i - G_c)\cdot(G_c - G_{i+P/2})]2^i \tag{9}$$

In Equation (9), G_c is the central pixel to be encoded, G_i and $G_{i+P/2}$ are the selected pixels in established ways, and P, R has the same meaning as above. The threshold function t is defined as:

$$T(x_1, x_2) = \begin{cases} 1 & x_1 \cdot x_2 \le 0 \\ 0 & otherwise \end{cases} \tag{10}$$

CS-LDP generates a $2^{P/2}$ long histogram, which is the same with CS-LBP. Figure 4 is the encoding process of CS-LDP for a neighborhood size of 8. Uniform CS-LDP is defined similarly to UXCS-LBP utilizing Equation (8) and Equation (9). The extracting process of histogram by uniform $CS-LDP_{8,1}$ is similar to that extracted by $UXCS-LBP_{8,1}$ mentioned above.

3. Hybrid UXCS-LBP and uniform CS-LDP

XCS-LBP being a derivation of CS-LBP retains local contrast information; CS-LDP describes detail information. Thus combining the two descriptors would generate a more discriminative feature. For reducing the amount of computation and histogram length, we combine UXCS-LBP and uniform CS-LDP to generate the required feature histogram. Suppose that the pixel lying in (x_c, y_c) is the center and R is the radius of circle R_{circle}. The histogram computed by XCS-LBP and CS-LDP can be represented as $H_{UXCS-LBP}$ and H_{CS-LDP} .

We define $H = H_{UXCS-LBP} + H_{UCS-LDP}$, where + implies concatenation operation. Since we just discuss about the situation where neighborhood pixel is 8, both UXCS-LBP and uniform CS-LDP produces a 9 dimensional histogram. The length of hybrid histogram is 18, which is much shorter than that of LBP or CS-LBP.

c. Overlapping Block-Based Background Modeling

1. Overlapping Mechanism

Each frame is divided into blocks with $N \times N$ size, and each block overlaps its neighbor by a certain number of pixels in both vertical and horizontal directions, which can be regarded as block advancement. The less advancement leads to the larger overlapping between blocks. According to the experimental results, the final mask tends to be smoother and more accurate as overlapping region grows, so are computation and other costs. In this paper, to achieve a relatively high precision and low computation result, here 2 as advancement is adopted. Specifically, a 8×8 block overlaps its neighbor by 6×6 pixels.

Extract histograms H via the proposed hybrid descriptor of each block for further analysis. Specifically, the feature vector of block located at (i, j) could be represented by $H_{(i,j)}$. Here we set P = 8 and R = 1, thus the dimension of the feature vector generated for each block is 18.

2. Block-Based Background Modeling

Background modeling being the most significant part of background subtraction aims at constructing and maintaining a statistical representation of background. It has three main components: 1) Background initialization by training the first N frames; 2) Generate binary mask for each frame by classifying blocks as foreground or background through comparing the current frame with trained background; 3) Background maintenance to update the obtained background over time. In the following we would elucidate the proposed training process, classification and maintenance mechanisms in detail.

We employ a two-component Gaussian mixture model for each of the block and cosine similarity to measure the likeness between two vectors. Cosine similarity is a judgment of orientation rather than magnitude, and experimental results suggest that the angles subtended by feature vectors exposed to varying illumination are hardly variant [21]. When classifying block (i, j), the cosine similarity is computed via:

$$\mathrm{Cos}\,im(H_{(i,j)}, \mu_{(i,j)}) = 1 - \frac{H_{(i,j)}^{T} \mu_{(i,j)}}{\left\| H_{(i,j)} \right\| \left\| \mu_{(i.j)} \right\|} \tag{11}$$

$H_{(i,j)}$ is the hybrid feature vector of block (i, j) and $\mu_{(i,j)}$ is the mean vector for location (i, j).

The first N frames are used for training in order to acquire $\mu_{(i,j)}$. Since complex scenes often contain dynamic factors in background, we employ the parameter estimation strategy proposed in [1]. It trains a two-component Gaussian mixture model for each block and utilizes the absolute difference of the weights of the two Gaussians. When the difference is larger than 0.5, the Gaussian with dominant weight is retained and assume that the Gaussian with smaller weight is modeling for dynamic foreground objects. When the absolute difference is less than 0.5, assume that there are no foreground objects and use all data for that particular block to estimate the parameters of the single Gaussian [1].

If $\mathrm{Cos}\,im(H_{(i,j)}, \mu(i, j)) \le T$ (T is a user-defined threshold), the block (i, j) would be classified as background. Once a block has been viewed as background, the corresponding Gaussian model is updated. Specifically, the mean vector is updated via:

$$\mu_{(i,j)}^{new} = (1-\rho)\mu_{(i,j)}^{old} + \rho H_{(i,j)} \tag{12}$$

3. Pixel-Level Classification

The classification method mentioned above is based on blocks. However, it is hard to tackle with the situation where both foreground and background pixels are in one block. Moreover, block-based classification couldn't preserve smooth edges. As overlapping mechanism, the majority of pixels belong to more than one block. Thus, the pixel with more blocks classified into background belongs to background and vice verses.

Specifically, let $pix(x, y)$ stand for a pixel in position of (x, y); utilize $Block_{(x,y)}^{total}$ represent the total number of blocks containing $pix(x, y)$; and $Block_{(x,y)}^{fg}$ is the number of blocks which are classified as foreground and contain $pix(x, y)$. The classification of $pix(x, y)$ is defined as:

$$Z(pix(x,y)) = \begin{cases} fg & Block_{(x,y)}^{fg} / Block_{(x,y)}^{total} \geq C \\ bg & otherwise \end{cases} \tag{13}$$

where fg is abbreviate of foreground, bg background, and C is a user-defined threshold, defined as 0.9 based on empirical results.

IV. EXPERIMENTAL RESULTS

To improve that the proposed method is significant to foreground detection as well as higher level applications, we conducted two series of experiments in this section. First, we compared the proposed algorithm with other two foreground detection methods based on given ground truth segmentations; then we applied the proposed method into object tracking to validate its practicability.

a. Comparison Results Based on Ground Truth

In order to evaluate the performance of proposed method, we conducted quantities of experiments on two representative datasets: Wallflower and I2R. The Wallflower dataset has seven image sequences, with each sequence presenting a potentially challenging scenario for background modeling, including swaying branches, sudden light switch, gradual illumination changing and etc. Each sequence has only one ground truth image available for evaluation. In our experiment, we operate our training, testing and evaluation process on the seven sequences according to their given introductions. The I2R Dataset contains nine image sequences captured in both outdoor and indoor situations. The sequences are considerably challengeable for their

complex background, e.g. glittering water surface, switched lights, spouting fountain, a crowded hall. The nine sequences have twenty hand-segmented ground truth images randomly chosen from thousands frames for evaluation.

The proposed method is compared with algorithms based on GMM [22] and XCS-LBP [11] descriptor. The GMM based method is implemented with OpenCV 2.0 and XCS-LBP descriptor based method is obtained from the authors. Our method is written in C++ language utilizing the Armadillo and OpenCV 2.0 libraries. Most parameter setting confronts with default values except that the learning rate adopted in GMM is 0.001. For XCS-LBP based method and proposed method, P (neighborhood pixels) = 8 and R (radius) = 1. The parameters C (pixel-level decision threshold) and T (cosine similarity threshold) in proposed method are defined as 0.9 and 0.03 respectively. Besides, we did not employ any morphological post-processing operations for the results obtained by three methods.

To evaluate the performance of the methods, we exhibit both visual and quantitative comparison results. For quantitative evaluation, we adopt the F-Measure metric which quantifies the similarity between ground truth and obtained mask. It is defined as:

$$F-measure = 2\frac{recall \times precision}{recall + precision} \tag{14}$$

$$recall = \frac{TP}{TP + TN} \tag{15}$$

$$precision = \frac{TP}{TP + FN} \tag{16}$$

TP, short for true positive, is the number of foreground pixels correctly detected as foreground ones; FP, false positive, is the number of background pixels detected as foreground ones; FN, false negative, is the number of foreground pixels detected as background ones.

We present foreground masks obtained by the three methods in Figure 5 on the Wallflower dataset. In Figure 5, the LS sequence records a room scene changing with the lights on and off. The TD sequence describes a room changing gradually form dark to bright. As is shown in Figure 5, GMM based method shows a less preferable result on the two sequences; it is sensitive to illumination change. The B sequence shows a busy cafeteria and each frame contains people to The TD sequence describes a room changing gradually form dark to bright. As is shown in Figure 5, GMM based method shows a less preferable result on the two sequences; it is sensitive to illumination change. The B sequence shows a busy cafeteria and each frame contains people to evaluate the performance of the three methods when training set contains foreground objects.

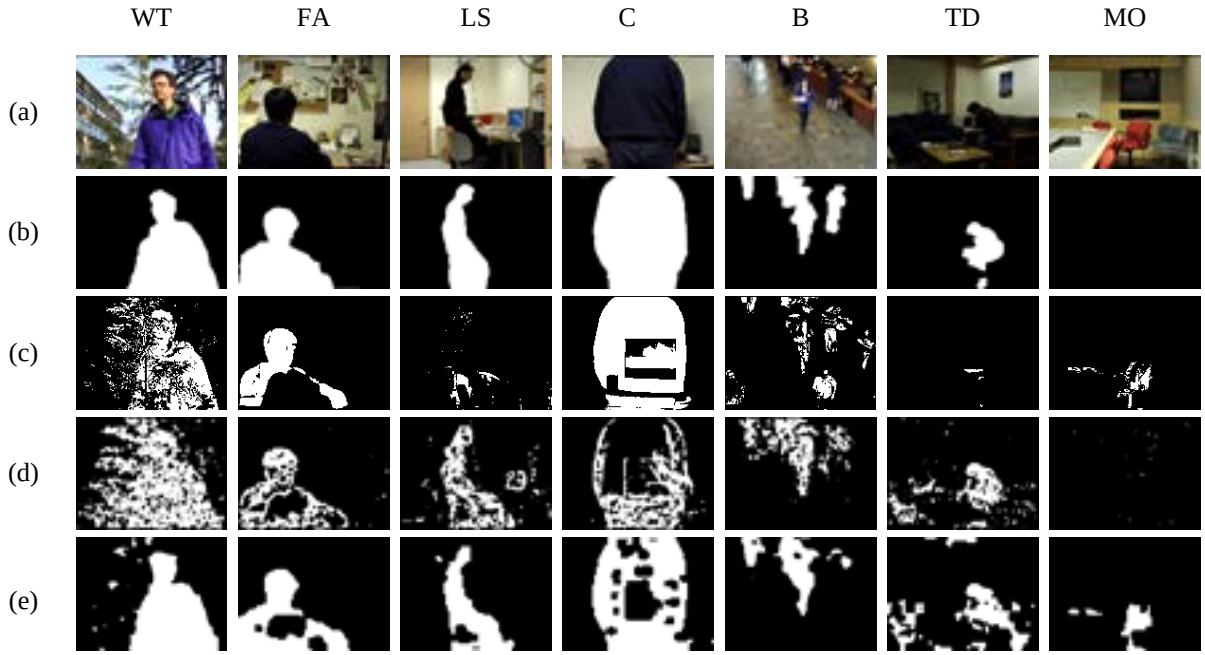


Figure 4 (a) Example frames from the Wallflower dataset. (b) Ground-truth foreground mask and foreground mask estimation using: (c) GMM based, (d) XCS-LBP based method, (e) Proposed method.

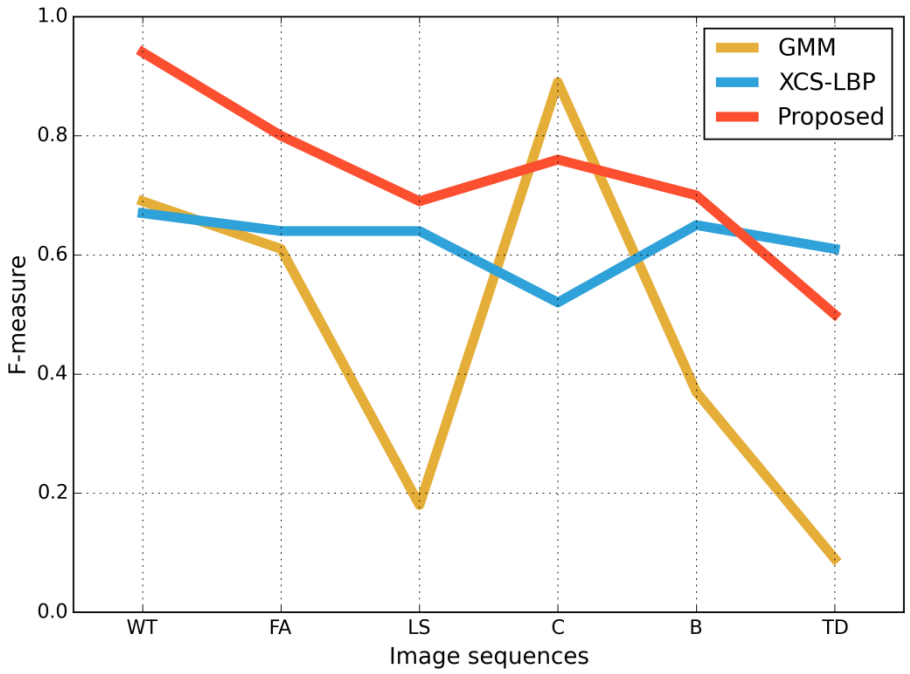


Figure 5 Comparison of F-measure values of foreground masks obtained on Wallflower dataset using GMM, XCS-LBP and proposed method

And GMM based performs not as well as expected. XCS-LBP descriptor based method acts not bad in the whole dataset. But its performance is less satisfactory in C sequence and WT sequence. WT sequence describes a person walking against a background consisting of waving branches. C sequence is a person walks in front of a monitor, with rolling bars on the screen which include similar color to the person's clothing.

We note that the XCS-LBP based shows a poor performance on the two sequences where background involves uninteresting dynamic factors or shares similar texture information with foreground objects. Our method clearly exhibits a better performance in general. And the corresponding quantitative evaluation is shown in Figure 6. Except the blue curve, the other two representing proposed and XCS-LBP method are relatively smooth. The average F-measure value of GMM is 0.48; that of XCS-LBP is 0.62; the value of proposed method is 0.72. Since the higher value of F-measure denotes the better performance of corresponding algorithm, our method outperforms the other two in most occasions.

In Figure 7, MR sequence is a meeting room with a moving curtain with teacher writing on blackboard. BR sequence is the same as B sequence in Wallflower dataset, recording a busy cafeteria. People are walking in front of a fountain in FT sequence. SC sequence presents a busy shopping mall and people show up in every frame. In LB sequence, people walk in an office with

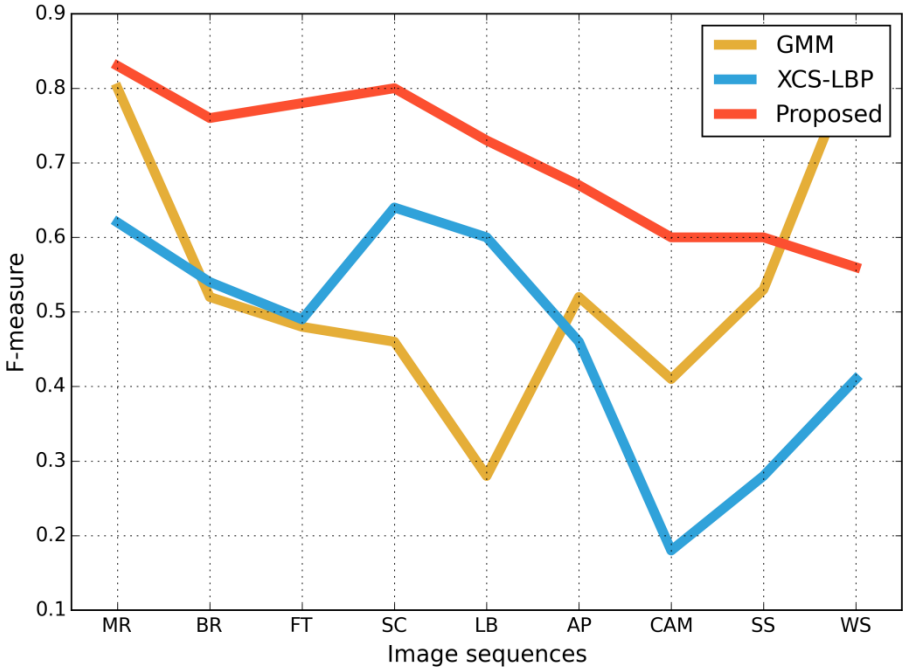


Figure 6 Comparison of F-measure values of foreground masks obtained on I2R dataset using GMM, XCS-LBP and proposed method

lights switched on and off. And in AP sequence a busy hall in airport is presented and each sequence contains people. In the CAM sequence, people and several cars pass on a road in front of strongly waving trees. SS is about people taking escalator in a subway station. In the WS sequence, a person walks along a lake and its background contains many waves and blue sky. As is shown in Figure 7, GMM based method tends to fail when illumination changes for its poor performance in LB sequence. Besides, XCS-LBP based method performs worse than expected on some sequences, for instance WS and CAM sequence where background images contain disturbing dynamic factors. Moreover, this method also acts poorly when the background and foreground objects contain similar texture or no texture. Unfortunately, the proposed method is unable to handle this problem as well, thus it acts not so well in MR, WS and C sequence. Specifically, the average F-measure value of GMM is 0.53; the value of XCS-LBP is 0.47; the value of proposed method is 0.70. To sum up, the proposed method shows a stably better performance than the other two on I2R dataset as well.

The comparison results shown in Figure 6 and Figure 7 denote that the proposed method outperforms the other two methods in most occasions. It performs better than XCS-LBP under circumstance that background contains uninterested moving objects; and it has an advantage over GMM when lighting changes. Nevertheless, the proposed method has weakness as well. It inclines to be not as well as expected when background and foreground objects share similar texture.

b. Application on Object Tracking

In this section, we integrated the proposed method with object tracking. The obtained masks are delivered into tracking mechanisms which is based on mean shift algorithm with FG using implemented by OpenCV 2.0. Three sequences are chosen for our further experiments. Seq1 is about a worker wondering along the railway in dim light and only one object for us to track; Seq2 records several slow-moving vehicles on the street and more than one target; Seq3 sequence demonstrates a small crowd hanging around on a street corner, which is much more complex than Seq1 and Seq2 for containing multiple interested objects and severe occlusions.

Figure 8 shows partial tracking results on three sequences, from which we can see that the tracker with proposed foreground detection method achieves quite well performance. Since our foreground detection method performs well even in dim light, the tacker performs well in Seq1.

The images sampled for Seq 1 is 43th, 57th, 81th and 108th show that the railway worker is continuously and accurately tracked. In Seq2, passing cars are also been tracked accurately. The 152th and 170th frames demonstrate the tracking result of a white car. And the 625th and 636th images show the result of a taxi. For the objects in the former two sequences, they are much easier to track for their simple scene with no more than two targets to track at the same time. The situation in Seq3 is much more complex with more than 5 objects to track and occlusion existing between those objects. The sampled frames 683th, 692th, 697th and 705th of Seq3 accurately track the original 5 objects and immediately track the new coming person with tracking ID as 049. From the experimental result on Seq3, we note that the objects would be deemed as a whole to track when they are close and the method prone to fail when occlusion happens.

The tracking system integrated with the proposed foreground detection method shows great performance on the three sequences. However, the biggest limitation for our method to be applied into practical tracking is time. Honestly, it could not achieve real time performance so far, but we would try our best to reduce its computation.

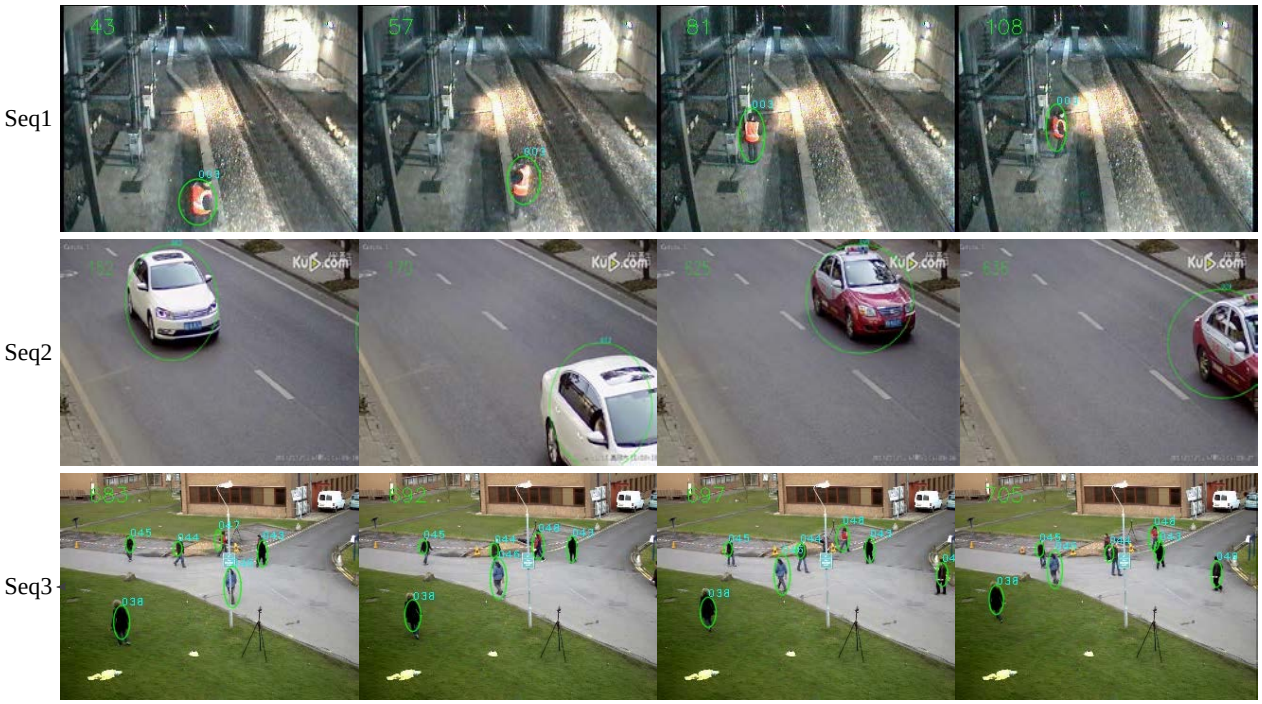


Figure 7 Tracking results on the three chosen sequences

V. CONCLUSION

In this paper, we combine the histograms extracted by UXCS-LBP and uniform CS-LDP, generating a much shorter histogram and retaining the strengths of the two descriptors. It has been proved to be robust under most circumstances ranging from dynamic background to changing illuminations. Besides, image segmentation is brought in our work as a pre-processing technique. In our experiment, we compared the proposed method with typical GMM and XCS-LBP based method qualitatively and quantitatively. Results on the two challenging datasets demonstrate that the proposed method show robust performance.

Further work would focus on how to improve the proposed framework for situations where foreground objects and background share similar texture or no texture and its further applications.

REFERENCES

[1] V. Reddy, C. Sanderson and B.C. Lovell, “Improved foreground detection via block-based classifier cascade with probabilistic decision integration”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 23, pp. 83-93, January 2013.

[2] E.A.J. Abadi, S.A. Amiri, M. Goharimanesh and A. Akbari, “Vehicle model recognition based on using image processing and wavelet analysis”, International Journal on Smart Sensing and Intelligent Systems, Vol. 8, No.4, pp. 2212-2230, December 2015.

[3] Y.L. Tian, A. Senior and M. Lu, “Robust and efficient foreground analysis in complex surveillance videos”, Machine Vision and Applications, Vol. 23, pp. 967-983, September 2012.

[4] S.H Kim, K. Sekiyama, T. Fukuda, “Pattern Adaptive and Finger Image-guided Keypad Interface for In-vehicle Information Systems”, International Journal on Smart Sensing and Intelligent Systems, Vol. 1, No. 3, pp. 572-591, September 2008.

[5] T. Bouwmans, F.E. Baf, and B. Vachon, “Background modeling using mixture of gaussians for foreground detection: a survey”, Recent Patents on Computer Science, Vol. 1, pp. 219-237, 2008.

[6] S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance”, Computer Vision and Pattern Recognition (CVPR), Vol.32, pp.1937-1944, 2011.

[7] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: an overview”, Computer Science Review, Vol. 11, pp. 31–66, May 2014.

[8] C. Stauffer and, W. E. L. Grimson, “Adaptive background mixture models for real-time tracking”, Computer Vision and Pattern Recognition, Vol. 2, 1999.

[9] X. H. Fang, W. Xiong, B. J. Hu and L. T, Wang, “A moving object detection algorithm based on color information”, Journal of Physics: Conference Series, Vol. 48, pp. 384, October 2006.

[10] H. Bhaskar, L. Mihaylova and A. Achim, “Video foreground detection based on symmetric alpha-stable mixture models”, Circuits and Systems for Video Technology, Vol. 20, pp. 1133-1138, 2010.

[11] C. Silva, T. Bouwmans and C. Frélicot, “An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos”, 2014.

[12] K. Kim and L.S. Davis, “Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering”, pp. 98-109, 2006.

[13] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction”, Pattern Recognition, Vol. 2, pp. 28-31, August 2004.

[14] B. White and M. Shah, “Automatically tuning background subtraction parameters using particle swarm optimization”, Multimedia and Expo, pp. 1826-1829), July, 2007.

[15] M. Mason and Z. Duric, “Using histograms to detect and track objects in color video”, Applied Imagery Pattern Recognition Workshop, pp. 154-159, October, 2001

[16] T. Ojala, M. Pietikainen and D. Harwood, “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions”, Pattern Recognition, Vol. 1, No. 1, pp. 582-585, November, 1994.

[17] G. Xue, L. Song, J. Sun and M. Wu, “Hybrid center-symmetric local pattern for dynamic background subtraction”, Multimedia and Expo (ICME), pp. 1-6, July, 2011.

[18] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods”, Pattern Analysis and Machine Intelligence, Vol. 34, No. 11, pp. 2274-2282, 2012.

[19] M. Heikkilä, M. Pietikäinen and C. Schmid, “Description of interest regions with local binary patterns”, Pattern recognition, Vol. 42, No. 3, pp. 425-436, 2009.

[20] Y. Zheng, C. Shen, R. Hartley and X. Huang, “Pyramid center-symmetric local binary/trinary patterns for effective pedestrian detection”, ACCV, pp. 281-292, 2011.

[21] K. Kim, T.H. Chalidabhongse, D. Harwood and L. Davis, “Real-time foreground–background segmentation using codebook model”, Real-time imaging, Vol. 11, No. 3, pp. 172-185, 2005.

[22] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection”, Video-based surveillance systems, pp. 135-144, 2012.