IET Image Processing

Research Article

Extended scale invariant local binary pattern for background subtraction

ISSN 1751-9659
Received on 5th December 2016
Revised 8th December 2017
Accepted on 10th March 2018
E-First on 13th April 2018
doi: 10.1049/iet-ipr.2016.1026
www.ietdl.org

Dongdong Zeng[1,2,3] ✉, Ming Zhu[1], Fang Xu[1,3], Tongxue Zhou[1,3]

[1]*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, People's Republic of China*

[2]*Key Laboratory of Airborne Optical Imaging and Measurement, Chinese Academy of Sciences, Changchun 130033, People's Republic of China*

[3]*University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China*

✉ *E-mail: zengdongdong13@mails.ucas.edu.cn*

Abstract: Background subtraction based on change detection is the first step in many video surveillance systems, an effective background subtraction algorithm should distinguish foreground from the background sensitively, and adapt to the variation of background scenes robustly. In this study, the authors propose a robust background subtraction algorithm which takes advantages of local texture features represented by an extended scale invariant local binary pattern and colour intensities to characterise pixel representations. Local texture features achieve good tolerance against illumination variations in rich texture regions but not so efficiently on uniform regions, so a photometric invariant colour measurement is proposed to overcome its limitation. Both quantitative and qualitative evaluations carried out on a well-known change detection dataset are provided to demonstrate the effectiveness of the proposed algorithm.

1 Introduction

Many high-level computer vision applications such as object tracking, video surveillance and activity recognition rely on a pixel-level segmentation of scenes into foreground and background. This task is often referred to as background subtraction and its performance has a huge effect on the performance of these applications.

Generally speaking, a complete background subtraction process has four components: (i) model initialisation, which regards the initialisation process; (ii) model representation, which describes what kind of model to be used to represent the background model; (iii) model update, which concerns the update mechanism used for adapting the model to the changes of the scene over time; (iv) foreground detection, which consists of comparing the current frame with the background frame to label pixels as foreground or background. The background subtraction process needs to deal with many challenging situations such as illumination variations, dynamic backgrounds, camera jitter, bad weather, noise and shadows. Over the recent past, a multitude of algorithms for background subtraction have been developed, mentioning all of them would go beyond the scope of this article, excellent survey papers can be found in [1, 2]. Most of background subtraction algorithms mainly manifest in two aspects, the first one is to take more advanced probabilistic models to represent the background [3–7], the other one is to employ a more powerful feature descriptor [8–10] or combine different features together [11–14]. As we know, colour features are the most widely used features in background subtraction, but they present several limitations when shadows, camouflage and illumination variations occurrence, texture features have been developed to deal with these situations.

In this paper, we propose a single model, single update scheme, spatio-temporal-based background subtraction algorithm. The key aspects of our method are as follows:

i. We propose a new texture feature called extended scale invariant local binary pattern (ESILBP) which derive from scale invariant local ternary patterns [15] (SILTP), comparing with traditional texture features [like local binary pattern (LBP) [8], SILTP [15], local binary similarity patterns (LBSP) [9]], the texture representation of ESILBP is more powerful.
ii. A photometric invariant colour measurements is proposed to overcome illumination variations.
iii. A combination of ESILBP and colour features are embedded into the background subtraction framework, both the colour features and ESILBP features have their own merits and demerits, they can compensate each other for better performance.
iv. A model update mechanism derived from the ViBe [16] algorithm is used for adapting the model to the changes of the scene over time.

The recently published Change Detection dataset [17, 18] provides realistic large-scale videos with accurate hand-labelled ground-truth, giving a balanced coverage of the range of challenges present in the real world. Extensive experiments evaluated on the dataset demonstrate that our method compared favourably to the recent state-of-the-art background subtraction methods.

The rest of this paper is organised as follows. Section 2 gives an overview of the related background subtraction algorithms. Section 3 introduces the proposed texture feature (ESILBP) and photometric invariant colour measurements. Section 4 describes the framework for background subtraction. Experimental results on the Change Detection dataset [17, 18] are reported in Section 5. Finally, conclusions are given in Section 6.

2 Related work

The simplest and earliest algorithms used for background subtraction are based on such idea: when using stationary cameras, the background can be modelled by the observed intensity of every independent pixel, disparities between the current frame and background reference frame are usually indicative of foreground objects. However, due to the complex nature of read-world scenes, finding a good background frame is usually impossible.

Over the recent past, improved adaptive algorithms which used pixel-wise intensity to create background models have been made, such as the ones based on probability density estimation (Gaussian mixture models [4, 5]), they model dynamic background elements at individual pixel locations using a mixture of Gaussian probability density functions. A common problem of this kind of method is to find the balance between the model stability and the

© The Institution of Engineering and Technology 2018

speed of the model adapts to the changing background. Recently, more flexible and adaptive variations of GMM [6, 7] were proposed to improve the update speed and the model stability.

However, the Gaussian assumption for the pixel intensity distribution is not always true in practical applications. To deal with this limitation, a non-parametric approach based on kernel density Estimation (KDE) was proposed in [3], which builds a statistical representation of the scene background by estimating the probability density function directly from the data without any prior assumptions. The KDE method is able to establish the probability density function of arbitrary shape to approximate the distribution of the actual pixel intensity, but it requires more storage spaces and most of them update their models in a first-in-first-out (FIFO) strategy, thus are unable to model both short-term and long-term periodic events.

The Codebook method [19] presented an alternative approach to solve the above-mentioned problems. Each pixel is represented by a codebook, each codebook consists of codewords which contain colours transformed by the innovative colour distortion metric. During the detection phase, if the current pixel is similar with one of the codewords, it is classified as a background pixel; otherwise, it will be considered as a foreground pixel. The Codebook method is believed to be able to capture background variations over a long period of time thus it can deal with periodic variations, however, it ignores the spatial information around pixel, it cannot capture complex spatio-temporal distributions of background.

The algorithms presented in [20, 21] implement a background subtraction based on neural network, each pixel is modelled with a neuronal map of weight vectors. The closest weight vector with the input pixel is updated over time. Such adaptive model can handle scenes containing moving backgrounds, gradual illumination variations and camouflage. However, they require a training period depending on the present time of the foreground objects in the sequences.

The first non-deterministic background subtraction algorithm called ViBe was proposed in [16] and have shown to outperform many existing algorithms. The ViBe is the first algorithm that uses a stochastic maintenance strategy to integrate new information into the model. If the incoming pixel is classified as background, it has the probability to be inserted into the background sample model at the corresponding location, while many other algorithms just replaced the oldest samples with the new sample. The authors showed that the stochastic strategy ensures the expected lifespan of each sample decays exponentially, due to its simplicity and effectiveness, we use a similar model update strategy in our algorithm.

All the above-mentioned background subtraction algorithms are based on colour features; in [8], Heikkila *et al.* developed a novel and powerful approach based on texture features represented by LBP to capture background statistics. Each pixel is modelled as a group of LBP histograms calculated over the neighbourhoods. Their method has shown excellent performance because of its computational simplicity and tolerance against illumination changes. However, these features are computed based on the comparison of the centre pixel with the neighbouring pixels, changes cannot be detected in sufficiently large uniform regions or if the intensity of the centre pixel remains larger than neighbouring pixel after a change in a scene. An improved version of LBP called SILTP was proposed in [15], which is more effective in dealing with illumination variations and shadows in scenes. Despite these advancements, they perform poorly in flat areas and result in some 'holes' in objects. Recently, LBSP was proposed in [9, 22], based on the absolute difference, demonstrated to surpass traditional colour comparisons via Hamming distance thresholding.

Some authors also proposed to use multiple features to improve the robustness of the background model; the idea is to add other features to the colour feature. The most common way is to add texture features to be more robust to the illumination variations and shadows. In [14], Yao *et al.* proposed a multi-layer background subtraction based on colour feature and LBP feature, producing satisfactory results in a large variety of scenes. More recently, St-Charles and Bilodeau [9] performed the background subtraction by integrating the colour feature and the LBSP feature, showing the state-of-the-art performance.

3 Texture and colour features

In this section, we will introduce the proposed texture feature called ESILBP and photometric invariant colour measurements.

3.1 Texture description with ESILBP

The original LBP was proved to be a powerful local feature descriptor [8]. It labels the pixels in an image block by thresholding the neighbourhood of each pixel with the centre pixel and gets the results as a binary number. Let a pixel c be in a certain location, the coordinate of the pixel is (x_c, y_c), there are N neighbouring pixels spaced on a circle of radius R. The LBP operator applied to $c(x_c, y_c)$ can be expressed as

$$\mathrm{LBP}_{N,R}(c) = \sum_{i=0}^{N-1} S(g_i - g_c)2^i \tag{1}$$

where g_c is the grey value of centre pixel c, g_i is the grey value of its N neighbouring pixels spaced on a circle of radius R and S is a thresholding function which is defined as

$$S(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

The encoding is shown in Fig. 1 (first row). However, the LBP operator is not robust to image noise; a little change of the central pixel value will cause a great effect on the resulting code. To deal with this problem, Tan and Triggs [23] proposed a local ternary pattern (LTP) operator which is more robust by adding a small offset value for comparison

$$S(x) = \begin{cases} 1, & \text{if } x \geq T; \\ -1, & \text{if } x < T; \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

where T is the fixed threshold used to add robustness. An example is shown in the second row of Fig. 1 with $T = 5$. However, adding a small offset value for comparison is not invariant under scale transform of intensity values by a multiplying constant. Suppose all pixel values are multiplied by 2, the LTP descriptor cannot keep its invariance against scale transform. To solve this problem, Liao *et al.* [15] proposed a SILTP operator. Given the centre pixel $c = (x_c, y_c)$, the SILTP operator can be expressed as

$$\mathrm{SILTP}^{\tau}_{N,R}(c) = \bigoplus_{k=0}^{N-1} s_\tau(I_c, I_k) \tag{4}$$

where I_c is the grey value of the centre pixel $c = (x_c, y_c)$, I_k is the grey value of its N neighbouring pixels spaced on a circle of radius R, $\oplus$ denotes concatenation operator of binary strings, τ is a scale factor affecting the tolerant range and s_τ is defined as

$$s_\tau(I_c, I_k) = \begin{cases} 01, & \text{if } I_k > (1+\tau)I_c; \\ 10, & \text{if } I_k < (1-\tau)I_c; \\ 00, & \text{otherwise}. \end{cases} \tag{5}$$

A SILTP encoding example is shown in Fig. 1 (third row, with scale factor $\tau = 0.1$). The most important properties of SILTP are its tolerance against illumination variations and local image noise within a range. However, we should also notice that the SILTP operator presented in [15] only used half of its eight neighbourhoods with four neighbouring pixels information, which might result in loss of the texture information. In this paper, we propose an ESILBP operator by taking more spatial information into consideration, as illustrated in Fig. 2. This pattern is calculated on a 5×5 neighbourhood region, we use the same computational

© The Institution of Engineering and Technology 2018

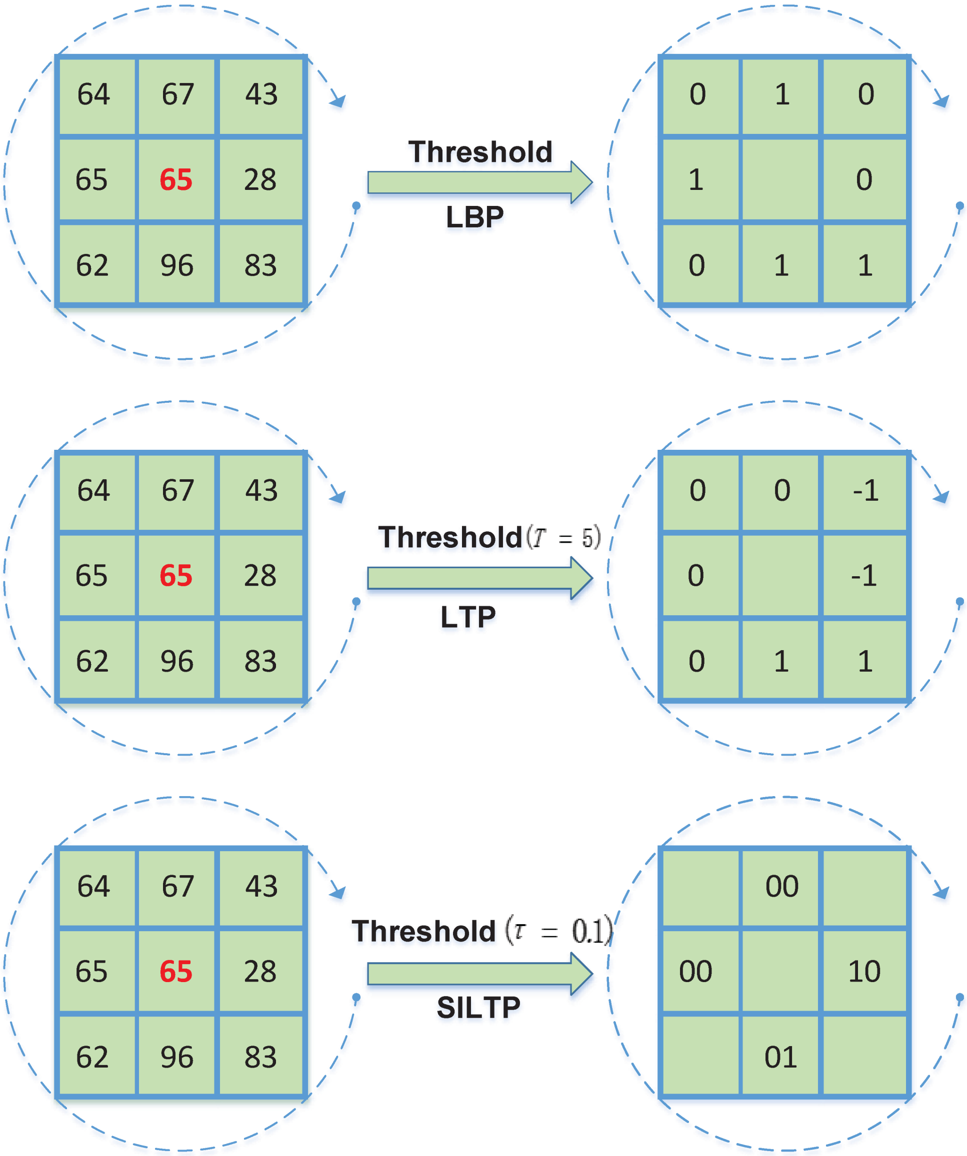

Fig. 1 *Examples of different LBP encoding. First row: LBP. Second row: LTP. Third row: SILTP*

expression as (4) and (5), but we set the parameters $N = 8$ and $\tau = 0.3$. For example, in Fig. 2, we get the ESILBP operator: 1000000001100101, the length is 16 bits.

There are three advantages of the ESILBP texture feature. First, it is computationally efficient, which only need one more comparison than LBP for each neighbouring pixel. Second, it is robust to local image noise and illumination variations by introducing a scale factor. Finally, the ESILBP can represent more texture information compared to LBP and SILTP, although the computational complexity is increased compared with SILTP (8 bits) due to the increased operator length, we will demonstrate that we can get better performance while meeting real-time requirements.

3.2 Photometric invariant colour measurement

Although the texture features can work well in many scenes, the colour features also play a very important role in some scenes with less texture information. As illustrated in Fig. 3, we can see that only relying on the ESILBP feature comparisons can sometimes fail on the uniform and flat regions. This is because the wall as background and clothes as foreground have the same texture information, simply relying on texture feature is difficult to distinguish. To handle these situations, we proposed to use ESILBP features in addition to colour features to create our background model.

Since the RGB colour representation is sensitive to illumination variations, many background subtraction methods exploit the normalised RGB colour representation to deal with this problem, however, they do not work well in the dark regions. In our experiment, we observed how pixel values change over time under illumination variations using a colour panel, we found that the pixel values changes are mostly distributed in the axis going towards the origin colour point (0,0,0) [14, 19]. Based on the observation, we compare the colour difference using the relative angle with respect to the origin colour point and the change range of their brightness.

As depicted in Fig. 4, for an input pixel $I^t(p)$ and a background model pixel $I_k^{t-1}(p)$, the colour distance is defined as

$$\begin{aligned}\mathrm{Dist}(I_k^{t-1}(p), I^t(p)) = {} & \max\,(D_A(I_k^{t-1}(p), I^t(p)),\\ & D_R(I_k^{t-1}(p), I^t(p)))\end{aligned} \tag{6}$$

where $D_A(I_k^{t-1}(p), I^t(p))$ denotes the relative angle of $I^t(p)$ and $I_k^{t-1}(p)$, and $D_R(I_k^{t-1}(p), I^t(p))$ is the range within which we allow the colour changes to vary. The D_A is defined as

$$D_A(I_k^{t-1}(p), I^t(p)) = 1 - \mathrm{e}^{-\max(0, \theta - \theta_n)} \tag{7}$$

© The Institution of Engineering and Technology 2018

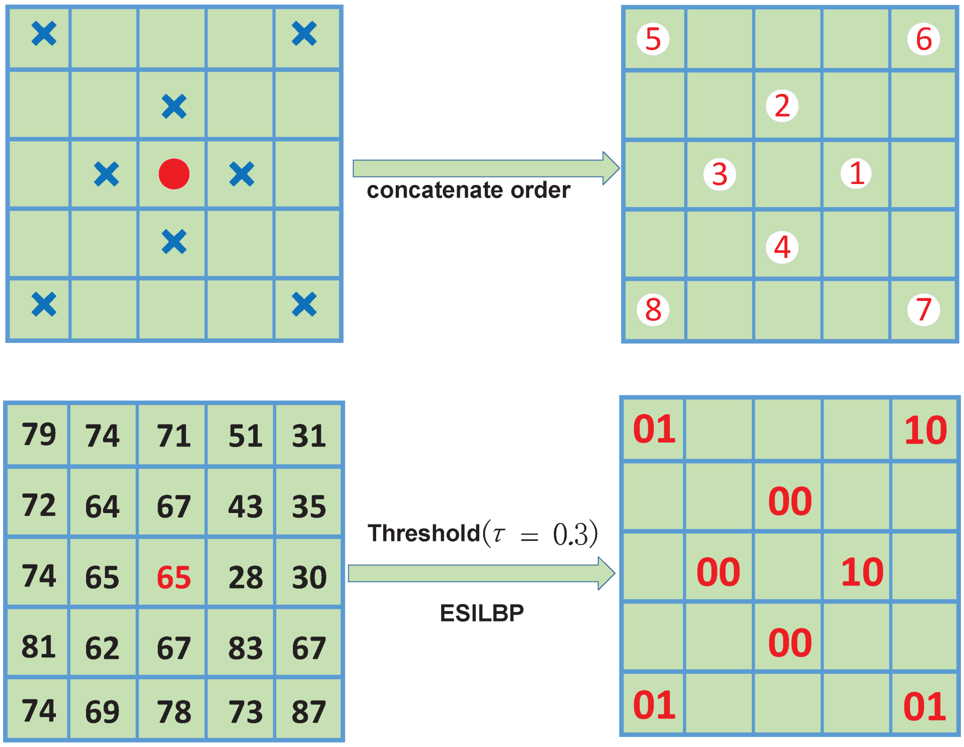

Fig. 2 *Example of ESILBP. First row: pattern used to calculate ESILBP. Second row: an example of ESILBP operator encoding*

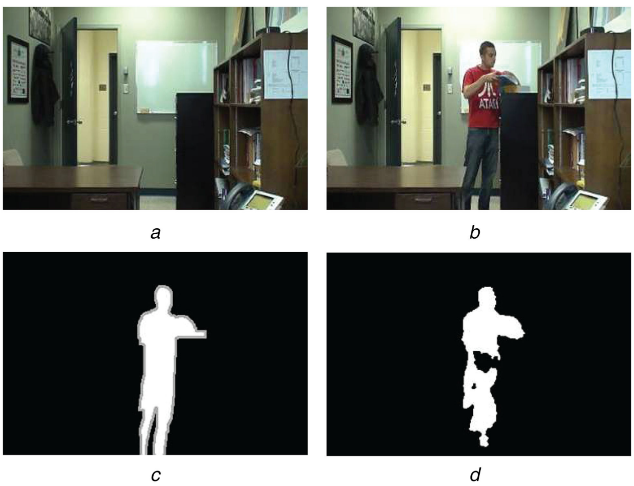

Fig. 3 *Typical failure case when only ESILBP features are used for change detection on CDNet dataset*
(a) Background image, ***(b)*** 812nd frame image from the office sequence, ***(c)*** Ground-truth, ***(d)*** Foreground detection result based on ESILBP features

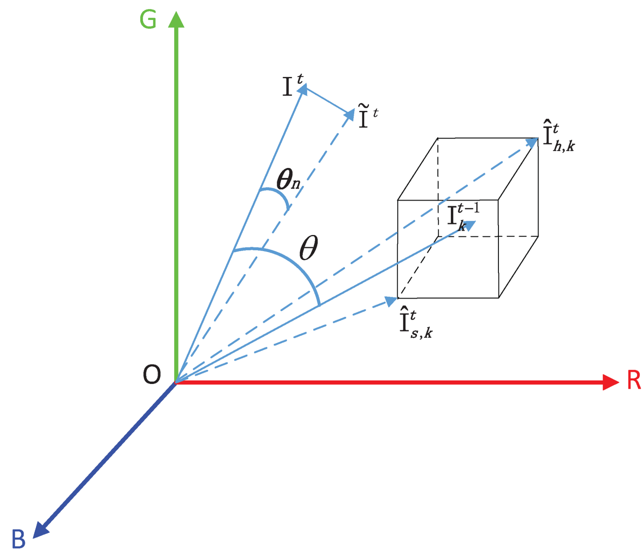

Fig. 4 *Photometric invariant colour model*

where θ is the angle between two RGB vectors I_k^{t-1} and I^t. θ_n is the largest angle between the RGB vector of I^t and the noise RGB vector of $\tilde{I}^t$, and we empirically set it to 3°. The D_R is defined as

$$D_R(I_k^{t-1}, I^t) = \begin{cases} 0, & \text{if } \breve{I}_{s,k} < I^t < \hat{I}_{h,k}; \\ 1, & \text{otherwise}. \end{cases} \quad (8)$$

where $\breve{I}_{s,k} = \min(\lambda I_k^{t-1}, \breve{I}_k^{t-1})$ $(\lambda \in [0.4, 0.7])$, and $\hat{I}_{h,k} = \max(\eta I_k^{t-1}, \hat{I}_k^{t-1})$ $(\eta \in [1, 1.2])$. $\breve{I}_{s,k}$ represents the potentially darkest 'shadow' value that the pixel can take, and $\hat{I}_{h,k}$ represents the brightest 'highlight' colour value that the pixel can take.

4 Background modelling

In this section, we employ ESILBP features and colour features to the statistical model of background and give a detail description of the framework for background subtraction, including background model representation, background model initialisation, foreground detection and background model maintenance. Fig. 5 provides a flowchart of the proposed background subtraction algorithm.

4.1 Background model representation

Most of the background subtraction algorithms rely on probability density functions [4, 7] or statistical parameters [3, 24] of the background generation process. In fact, the choice of a particular probability density function inevitably introduces a bias towards the real probability density function. The innovative mechanism presented in ViBe [16] indicated that the observed pixel samples would have a higher probability to appear again, it relies on the collection and maintenance of background model samples using a random approach.

In this paper, we adopt a sample consensus background modelling approach similar to ViBe. The ViBe is a pixel-based flexible and lightweight method; it has shown excellent performance to handle dynamic backgrounds and gradual

© The Institution of Engineering and Technology 2018

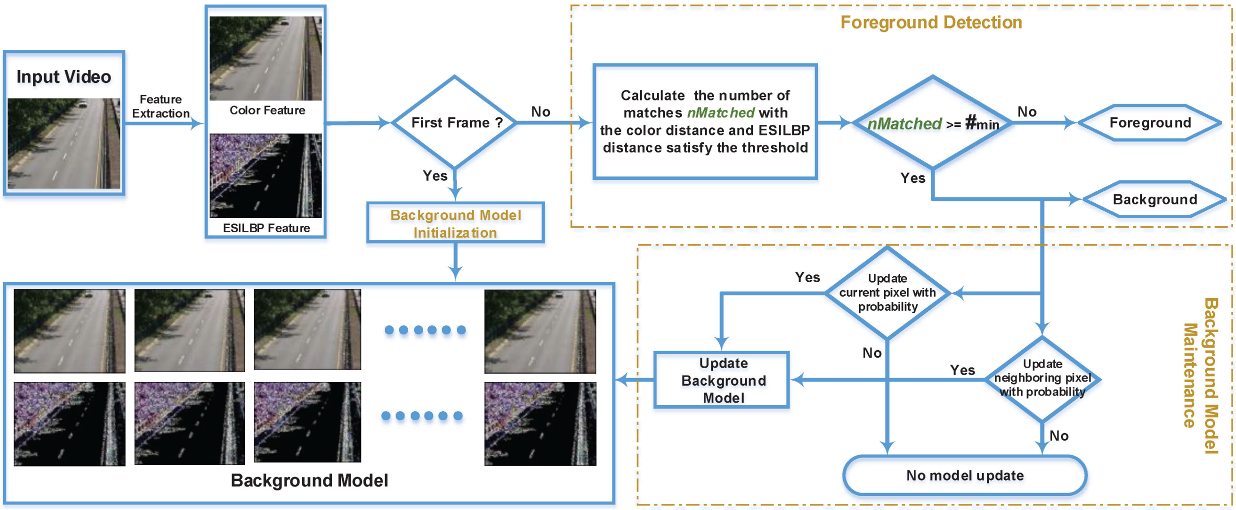

Fig. 5 *Flow chart of the proposed background subtraction algorithm*

illumination changes. However, the original ViBe only takes colour feature into consideration, in our method, we try to integrate the ESILBP features and colour features to characterise pixel representations. That is to say, we use the colour-texture combined features to replace the single colour features in the original ViBe method. For each pixel $p(x, y)$, its background model $B(p)$ contains a set of N recent background samples

$$B(p) = \{\phi_1(p), \phi_2(p), \phi_3(p), \ldots, \phi_N(p)\} \tag{9}$$

where $\phi_i(p) = \{I_i(p), \mathrm{ESILBP}_i(p)\}$ is the previously observed background samples and each containing a colour feature $I_i(p)$ and an ESILBP feature $\mathrm{ESILBP}_i(p)$.

4.2 Background model initialisation

Many popular background subtraction algorithms described in the literature such as [3, 19] need a sequence of frames to initialise models. But one may wish to segment the foreground in short initialisation sequence or even from the second frame on. Furthermore, many applications require the ability to refresh or re-initialise the background model in the presence of sudden lighting changes. A more convenient solution is to initialise the background model from a single frame.

Since no temporal information is contained in a single frame, it seems to be no other choice than to take values from the spatial neighbourhood. We make the assumption that the neighbouring pixels share a similar temporal distribution. Suppose that we have a pixel $p(x, y)$ in the initialised image, then the background model $B(p)$ is initialised by random selecting the feature values from the neighbourhood of p for N times

$$B(p) = \{\phi(\bar{p}) \mid \bar{p} \in \mathcal{N}(p)\} \tag{10}$$

where $\mathcal{N}(p)$ is the neighbouring pixel of p and the probability of choosing $\bar{p}$ follows a Gaussian distribution. In our experiments, a 7×7 neighbourhood region has proved to be a good choice.

For example, as shown in Fig. 5, suppose the current input frame I is the first frame, the background model is constructed as follows: first, the colour feature map I_{color} and ESILBP feature map I_{ESILBP} are extracted from current frame, then, for a pixel $p(x, y)$ in the model, randomly select a position $\bar{p}(\bar{x}, \bar{y})$ in its 7×7 neighbourhood region, get the colour feature and ESILBP feature of $\bar{p}(\bar{x}, \bar{y})$ from $I_{\text{color}}(\bar{x}, \bar{y})$ and $I_{\text{ESILBP}}(\bar{x}, \bar{y})$, take these values as the colour feature and ESILBP feature of $p(x, y)$, repeat the process N times for all pixel, we get the final initialised background model.

The number of background samples per pixel model has recommended to take the value of $N = 20$ in [16]. In fact, N controls the balance of sensitivity and precision of the model, using more background samples lead to higher precise but lower

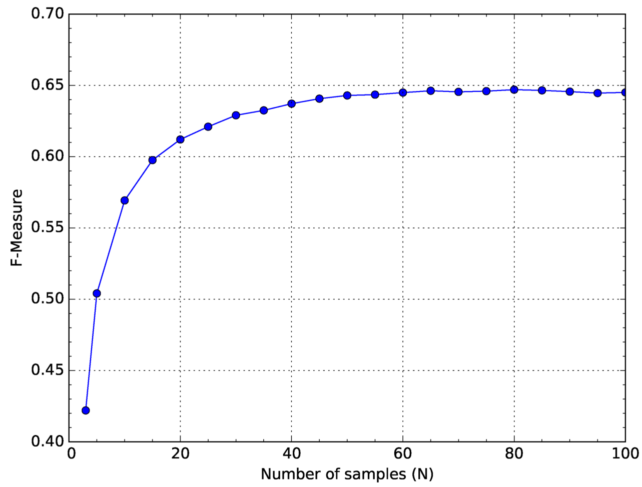

Fig. 6 *F-measure scores obtained on the CDnet 2014 dataset for different background samples*

sensitive, and vice-versa. Due to the larger representation space induced by the texture feature ESILBP, we have to raise the value of N to keep more background samples in the model. We determine this value based on the 2014 CDnet dataset [18], as we can see in Fig. 6, the overall *F-measure* score tends to get maximum when N reaches the value of 80, we determined to set $N = 50$ in this paper, although $N = 80$ is preferred for better precision, larger N value increases memory and computational complexity, but with few performance improvement.

4.3 Foreground detection

In foreground detection procedure, let us denote the input frame at time t as I^t, to classify a pixel $p^t(x, y)$ as foreground or background, we will need to calculate the number of matches between the input pixel $p^t(x, y)$ and its background model $B(p)$. This procedure is presented in the following equation:

$$M(p^t(x, y)) = \#\{i \mid \mathrm{dist}(p^t(x, y), \phi_i(p)) < T, i \in [1, N]\} \tag{11}$$

where $M(p^t(x, y))$ is the number of matches, $\mathrm{dist}(p^t(x, y), \phi_i(p))$ calculated the distance between $p^t(x, y)$ with its background model samples. We should notice that the input pixel $p^t(x, y)$ contains colour feature $I^t(p)$ and ESILBP feature $\mathrm{ESILBP}^t(p)$, we need to get the distances in two different ways.

First, to calculate the colour similarity, L1 or L2 distance are the most commonly used metric due to their simplicity and efficiency [16, 22], however, they perform poorly under illumination

© The Institution of Engineering and Technology 2018

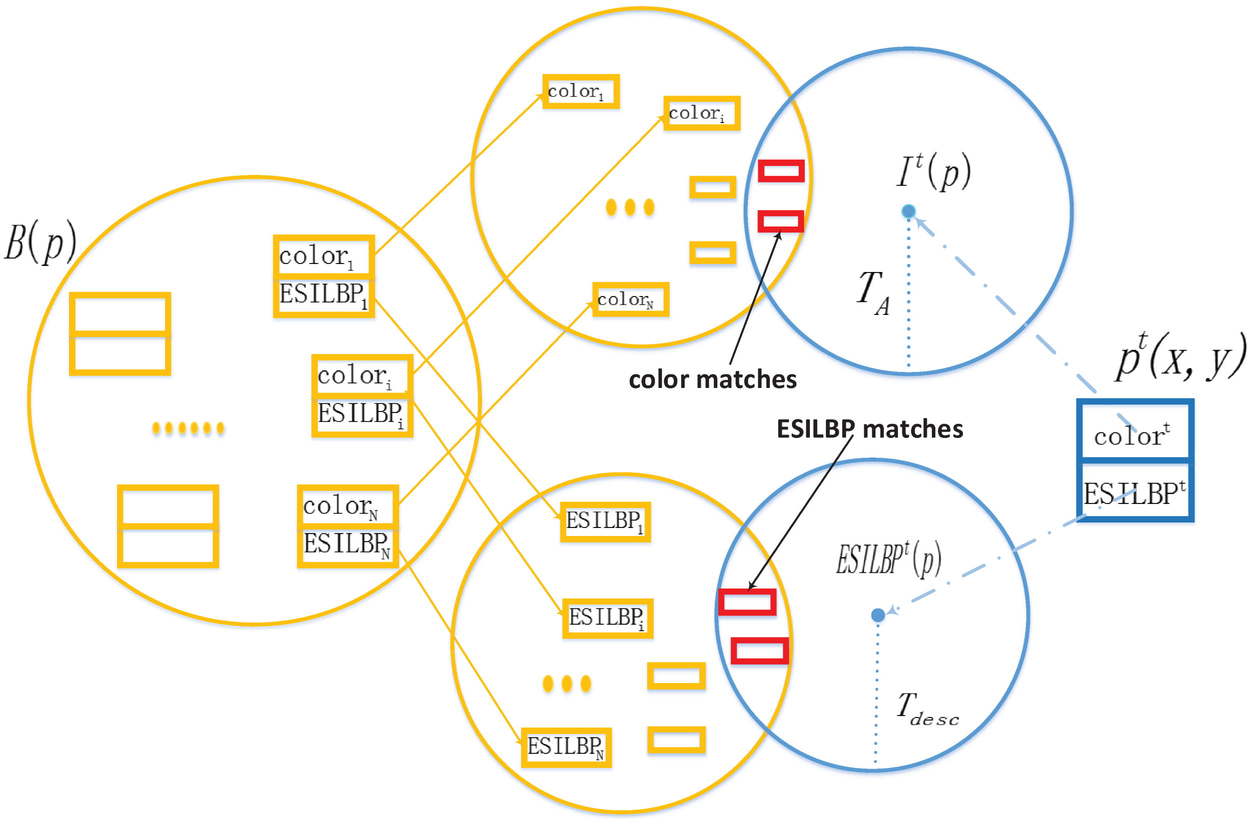

Fig. 7 *Calculate the number of matches between the input pixel and its background model. To find a match, both the colour feature and ESILBP feature must be successfully matched*

Input : pixel $p(x, y)$
Output : the FG/BG label of $p(x, y)$
$colorDist = 0$, $textureDist = 0$, $nMatches = 0$, $i = 0$
while $nMathes < \#_{min}$ && $i < N$
  $colorDist = Dist(I_i(p), I^t(p))$
  if $colorDist > T_A$
    goto failedMatch;
  for $c = 1 : nChannels$
    $textureDist$ += $ESILBP_c^t(p) \oplus ESILBP_{i,c}(p)$
  if $textureDist > nChannels \cdot T_{desc}$
    goto failedMatch;
  $nMatches$++;
  failedMatch:
    i++;
if $nMatches < \#_{min}$
  $p(x, y)$ is foreground;
else
  $p(x, y)$ is background;

Fig. 8 *Algorithm 1: foreground detection*

variation or shadow scenes. In this paper, a photometric invariant colour measurement was proposed in Section 3.2 to measure the colour similarity, so we get the colour distance with (6), if the result is less than the pre-defined threshold T_A, a colour match was found. Then, to calculate the similarity between ESILBPs, we should make a measuring strategy that is different from colours as ESILBPs are binary strings described texture feature. The Hamming distance is adapted to compare the similarity, we can use the XOR operator to get the distance effectively. For example, there are two ESILBPs: 01101011 01010010 and 01101010 00010110, the distance is 3. If the distance is less than T_{desc}, a ESILBP match was found. To consider input pixel matches with a background sample, both the colour feature and ESILBP feature must be successfully matched. Fig. 7 displays the process of finding a match. So (11) can be rewritten as follows:

$$M(p^t(x, y)) = \#\{i \mid \text{dist}(I^t(p), I_i(p)) < T_A \quad \&\& \quad \text{dist}(\text{ESILBP}^t(p), \text{ESILBP}_i(p)) < T_{\text{desc}}, i \in [1, N]\} \tag{12}$$

After we obtained the number of matches, the label of pixel $p^t(x, y)$ is classified as follows:

$$S^t(p) = \begin{cases} 1, & \text{if } M(p^t(x, y)) < \#_{\min} \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

where $S^t(p)$ is the output segmentation map and 1 means foreground and 0 means background. $\#_{\min}$ is the minimum number of matches required for a pixel. In this paper, we set $\#_{\min} = 2$ to get a reasonable trade-off between computational complexity and noise resistance as demonstrated in [16].

The pseudocode of foreground detection procedure is shown in Algorithm 1 (see Fig. 8).

4.4 Background model maintenance

Many background model update strategies have been summarised in [2], most of them use FIFO strategy to update their model, but there is no evidence showing that this is optimal. ViBe proposed a conservative, stochastic update strategy. In this paper, we update the model with a similar strategy, it contains two steps: first, when a pixel $p(x, y)$ in the current frame $I^t(p)$ is classified as background, it has a $1/\phi$ probability to replace a randomly picked background sample in $B(p)$, where ϕ is a time subsampling factor as described in [16]. Then, $\bar{p}(\bar{x}, \bar{y})$, one of the neighbours of $p(x, y)$ in 3×3 region also has the same probability $(1/\phi)$ to replace one of its background samples by the features of $p(x, y)$.

The pseudocode of background update is shown in Algorithm 2 (see Fig. 9).

The fact that background samples are replaced randomly instead of replacing the oldest one guarantee that a solid history of long-term and short-term background representation can be remained in our model. This updating strategy cancels the time

© The Institution of Engineering and Technology 2018

window concept and each result of background subtraction is different, combining with a conservative updating strategy, new samples can be incorporated into the background model only if they are classified as background, thus preventing static foreground objects from being absorbed into the background model too fast.

A problem caused by conservative updating strategy is *ghosting* effect, which is commonly defined as falsely classified background pixel regions due to the removal of scene objects, e.g. static objects that suddenly start moving. A popular method to deal with this situation is called 'detection support map ($M(p)$) [25]' which saves the number of times that a pixel has been consecutive classified as foreground, if the value of $p(x, y)$ in $M(p)$ exceed a given threshold, then $p(x, y)$ is inserted into the background model, however this method would add parameters and increase the computational complexity.

In our method, the second update step named *spatial diffusion* allows ghost regions to be automatically absorbed into the background model as time goes by. As neighbouring background pixels share similar spatial distribution, according to this updating strategy, background models hidden by the removed object will be updated with neighbouring pixel samples from time to time. It allows a spatial diffusion of information regarding the background evolution that relies on samples exclusively classified as background.

Moreover, the 'spatial diffusion' step improves the tolerance of the model to the limited camera motion and enhances the spatial coherence. Besides, the ESILBP features prevent the spread of samples across object boundaries. Even if a sample is wrongfully diffusion from one background model to another, the odds that might be matched are much lower due to the use of ESILBP features.

Input : the label of pixel $p(x, y)$
1: **if** $p(x, y)$ is background
2: **if** $rand()\ \%\ \phi == 0$
3: update $B(p)$ with the features of $p(x, y)$;
4: **if** $rand()\ \%\ \phi == 0$
5: update $B(\bar{p})$ with the features of $p(x, y)$;
6: **else**
7: do nothing;

Fig. 9 *Algorithm 2: background update*

Table 1 Overview of CDnet 2012 and 2014 dataset

Category	Videos	Dataset	Total number of pictures
baseline	4	2012 and 2014	6049
cameraJ	4	2012 and 2014	6420
dynamic	6	2012 and 2014	18,871
intermittent	6	2012 and 2014	18,650
shadow	6	2012 and 2014	16,949
thermal	5	2012 and 2014	21,100
badWeather	4	2014	20,900
lowFramerate	4	2014	9400
nightVideos	6	2014	16,609
PTZ	4	2014	8630
turbulence	4	2014	15,700
total	31 + 22	—	88,039 + 71,239

5 Experimental results

5.1 Evaluation datasets

We evaluate our method on the CDnet dataset provided for the Change Detection Challenge [17, 18]. The goal of the Change Detection Challenge is to allow for performance evaluation of recent and future background subtraction methods. The CDnet 2012 dataset consists of 31 videos from realistic scenarios with nearly 90,000 frames. These videos are grouped into six categories namely: *baseline, camera jitter, dynamic background, intermittent object motion, shadow* and *thermal*. Accurate human annotated ground-truth is available for all videos and is used for performance evaluation, thus, an exhaustive competitive comparison is possible on different methods. This dataset was also updated in year 2014, adding 22 videos with nearly 70,000 frames in five new categories: *bad weather, low framerate, night videos, pan-tilt-zoom* and *turbulence* [18]. To our knowledge this is the most complete dataset for background subtraction; a complete overview of the dataset is depicted in Table 1.

5.2 Evaluation metrics

In order to compare the methods, a total of seven different metrics have been defined to evaluate the performance of different methods. Let TP stands for the *true positives* and holds the number of pixels correctly labelled as foreground, TN stands for the *true negatives* and holds the number of pixels correctly labelled as background, FP stands for the *false positives* and holds the number of pixels incorrectly labelled as foreground, FN stands for the *false negatives* and holds the number of pixels incorrectly labelled as background. According to [17], these metrics are defined as follows:

- Recall (Re) $= (\mathrm{TP}/(\mathrm{TP} + \mathrm{FN}))$
- Specificity (Sp) $= (\mathrm{TN}/(\mathrm{TN} + \mathrm{FP}))$
- False positive rate (FPR) $= (\mathrm{FP}/(\mathrm{FP} + \mathrm{TN}))$
- False negative rate (FNR) $= (\mathrm{FN}/(\mathrm{TP} + \mathrm{FN}))$
- Percentage of wrong classifications (PWC) $= 100 \cdot ((\mathrm{FN} + \mathrm{FP})/(\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}))$
- Precision (Pr) $= (\mathrm{TP}/(\mathrm{TP} + \mathrm{FP}))$
- F-measure (FM) $= 2 \cdot ((\mathrm{Re} \cdot \mathrm{Pr})/(\mathrm{Re} + \mathrm{Pr}))$

The sum of all pixels in every category is used to calculate these metrics and the *overall* category is computed based on the mean of each category. For PWC, FNR and FPR metrics, lower values indicate higher accuracy, while for Re, Sp, Pr and FM metrics, higher values indicate better performance. During these metrics, we are especially interested in the *F-measure*, which is commonly accepted as a good indicator of the overall performance of background subtraction methods. It was found to be closely correlated with the method ranks used on the website [17], the better background subtraction methods usually have higher *F-measure* scores than the worse performing methods.

5.3 Parameters

Our method consists of few parameters; we can obtain optimal performance by adjusting parameters in different scenarios. However, we used a universal parameter set for all videos to respect the Change Detection Challenge [17, 18] competition rules:

- $T_A = 0.2$: colour distance threshold used in (6) to determine if an input pixel matches the background sample.
- $T_{\mathrm{desc}} = 2$: texture descriptor threshold used to determine if an input pixel matches the background sample based on the Hamming distance.
- $N = 50$: number of samples stored in the background model for each pixel.
- $\#_{\min} = 2$: minimum number of samples that match background model to label an pixel as background.
- $\lambda = 0.5, \eta = 1.2$: parameters used in (8).
- $\phi = 16$: time sampling factor used to update the background model.
- $\tau = 0.3$: scale factor used to calculate ESILBP features in (5).

In our proposed method, the segmentation decision is made independently for each pixel, thus the segmentation results can benefit from regularisation step, which is done using median filtering. It combines information from neighbouring pixels and assigns homogeneous labels on uniform regions. In this paper, we use a uniform 7×7 median filter for all evaluated methods.

© The Institution of Engineering and Technology 2018

Table 2 Complete results obtained with the proposed method on the CDnet 2012 and 2014 dataset

Category	Recall	Specificity	FPR	FNR	PWC	Precision	*F*-measure
baseline	0.9223	0.9984	0.0016	0.0773	0.4583	0.9596	0.9403
cameraJ	0.7122	0.9925	0.0074	0.2878	1.9228	0.8465	0.7703
dynamic	0.8179	0.9856	0.0144	0.1821	1.5902	0.6201	0.6216
intermittent	0.6327	0.9668	0.0332	0.3673	5.9732	0.6868	0.6238
shadow	0.9110	0.9922	0.0077	0.0890	1.1018	0.8621	0.8220
thermal	0.7209	0.9958	0.0042	0.2791	1.6658	0.9007	0.7824
Overall (2012)	0.7862	0.9886	0.0114	0.2138	2.1187	0.8125	0.7701
badWeather	0.4562	0.9996	0.0004	0.5438	0.9199	0.9312	0.5934
lowFramerate	0.7200	0.9946	0.0054	0.2800	1.2978	0.6629	0.6561
nightVideos	0.6110	0.9801	0.0213	0.3669	2.941	0.4627	0.4821
PTZ	0.6420	0.8111	0.1889	0.3579	19.1242	0.0413	0.6698
turbulence	0.8161	0.9962	0.0038	0.1839	0.4812	0.6602	0.6699
Overall (2014)	0.7249	0.9734	0.0271	0.2752	3.3871	0.6930	0.6452

Table 3 Per-category *F*-measure comparisons between hybrid methods on the CDnet 2012 dataset

Method	Baseline	CameraJ	Dynamic	Intermittent	Shadow	Thermal	Overall
multi-layer [14]	0.9004	0.7311	**0.6278**	0.4816	0.8099	0.6331	0.6993
LOBSTER [9]	0.9320	0.7462	0.5664	0.5940	0.8696	0.7803	0.7481
ESILBP	**0.9403**	**0.7703**	0.6216	**0.6238**	**0.8820**	**0.7824**	**0.7701**

5.4 Performance evaluation

First, we present the detail performance evaluation results of our method in Table 2. Using the evaluation framework of the CDnet dataset [17, 18], seven metric scores are reported, including *recall, specificity, FPR, FNR, PWC, precision* and *F-measure*. We can see from Table 2 that our method performs well on the CDnet 2012 dataset with a overall *F-measure* of 0.7701. In the *baseline* category, both the *recall* and *precision* metric scores exceed 0.9, and the *F-measure* gets a 0.94 high score. As demonstrated in [26], if a method with a *F-measure* above 0.94 and a *PWC* below 0.9, then the segmentation results may be considered almost as good as the ground-truth, since a simple dilation (or erosion) of one (or two) pixel of the ground-truth may result in the *F-measure* drop from 1.0 to about 0.94. This again shows the efficiency of our method. We also see that the *shadow* category is well handled by our method, this category mainly focuses with the challenges like illumination and camouflage, benefit from the combination of photometric invariant colour measurements and the texture feature ESILBP, the *recall* metric score exceed 0.9 and the *F-measure* is above 0.8. The same can be said for the *thermal* category, which contains numerous camouflage problems. As for the *camera jitter* category, although many sequences contain camera vibration, with the sample consensus background modelling approach, the *precision* is above 0.8, the *recall* and *F-measure* are all above 0.7. However, we also notice that these categories like *dynamic background* and *intermittent* pose a great challenge, with a *F-measure* about 0.6. The *dynamic background* category contains many complex background motion and changes in illumination, the *intermittent* category mostly contains sequences with abandoned objects and static objects suddenly start moving, it mainly focuses on static object detection, both of these categories are very difficult for most of the background subtraction methods. We can also see from Table 2 that the overall *F-measure* score 0.6452 of our method evaluated on the CDnet 2014 dataset is much lower than the score 0.7701 obtained on the CDnet 2012 dataset, as stated earlier, the 2014 dataset is much complex than 2012 dataset which mainly deals with traditional challenges of background subtraction. These new categories are much harder to deal with, including videos captured under outside snowy conditions, low frame rate videos with wavering global lighting conditions, urban traffic surveillance videos captured at night with glare effect caused by car headlights, videos obtained with pan-tilt-zoom cameras and long distance thermal surveillance videos with air turbulence under high temperature environments. The results show that only *lowFramerate, PTZ* and *turbulence* seem to result in acceptable performance with *F-measure* >0.65, both *nigheVideos* and *badWeather*, a more sophisticated, high-level background subtraction method would result in better performance, this is our future goal.

Second, we compare our algorithm with the following state-of-the-art classical algorithms: GMM [4], multi-layer [14], KDE [3], SOBS [20], ViBe [16] and LOBSTER [9]. Among them, GMM, KDE, SOBS and ViBe are pixel-level methods, SILTP is the region-level method. Multi-layer, LOBSTER and our method are hybrid methods. We first give a detail per-category *F-measure* comparisons between multi-layer, LOBSTER and our method on the CDnet 2012 dataset, we choose CDnet 2012 instead of CDnet 2014, as we could not find the source code of multi-layer but we found its evaluation results on CDnet 2012 dataset from [17]. Both of these methods integrate the colour information and texture information to build the background model, the multi-layer combine the colour features and the LBP features to model the background model, the LOBSTER maps the LBSP texture features and colour features into the background subtraction framework. The comparison results are shown in Table 3, we can see that our method gets the highest *F-measure* score in five out of six categories, and the overall *F-measure* score of our method achieves a 10.1% improvement compared with the multi-layer and 2.94% improvement compared with the LOBSTER. We also notice that in the *dynamic background* category, the multi-layer performs a little better than ours, this may be due to intrinsic noise sensitivity of texture features, as the ESILBP considers more spatial texture information and takes 16 bits to represent the texture feature compared with 8 bits of LBP. Next, we present in Table 4 the quantitative comparison results evaluated on the CDnet 2014 dataset, the results of other methods are from the website www.changedetection.net, for a specific metric, if the method obtains the best score on it, the corresponding value is highlighted in bold. We can see that the SOBS gets the best *recall* and *FNR* performance, LOBSTER gets the lowest *FPR* score (the lower, the better), while our method performs best on the other three metrics, the *PWC, precision* and *F-measure*, especially the *F-measure* is much higher than all other methods.

Finally, we present some qualitative comparisons between these methods under different challenging situations [27]. Fig. 10 shows the comparison results between GMM, ViBe and our method on the CDnet 2012 dataset. The first column is the input frame from difference sequences, the second column is the corresponding ground-truth, the third column is the segmentation results of our method, the fourth column represents the segmentation results of ViBe [16] and the last column shows the segmentation results of GMM [4]. As we can see, in highway sequence, our method almost

© The Institution of Engineering and Technology 2018

Table 4 Comparison of the results on the CDnet 2014 dataset by different methods

Method	Recall	Specificity	FPR	FNR	PWC	Precision	*F*-measure
SILTP [15]	0.5173	0.9581	0.0418	0.4827	5.5643	0.5021	0.4278
GMM Zivkovic [4]	0.6604	0.9725	0.0275	0.3396	3.9953	0.5973	0.5566
KDE Elgammal [3]	0.7375	0.9519	0.0481	0.2625	5.6262	0.5811	0.5688
viBe [16]	0.6147	0.9735	0.0265	0.3853	3.8616	0.6601	0.5755
SOBS [20]	**0.7621**	0.9547	0.0453	**0.2379**	5.1498	0.6091	0.5961
LOBSTER [9]	0.6836	0.9770	**0.0230**	0.3164	3.4310	0.6867	0.6230
ESILBP	0.7249	0.9734	0.0271	0.2752	**3.3871**	**0.6930**	**0.6452**

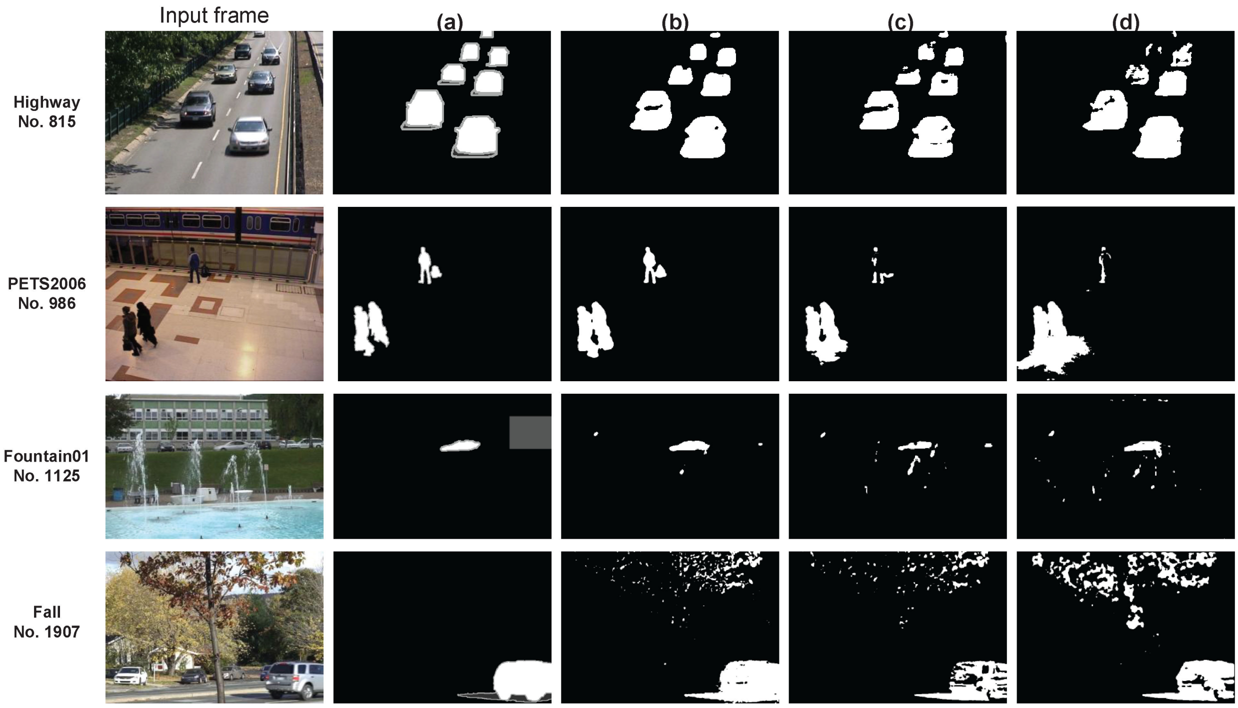

Fig. 10 *Qualitative performance comparison for various sequences of the CDnet 2012 dataset*
(a) Ground-truth, ***(b)*** Segmentation results of our method, ***(c)*** Segmentation results of ViBe, ***(d)*** Segmentation results of GMM

perfectly extracts the moving cars. In the shadowed PETS2006 sequence, one can notice that the GMM and ViBe approaches are susceptible to shadows, whereas our method is visibly better. This is attributed to the complete usage of photometric invariant colour measurement and ESILBP features. In fountain01 sequence, where water rippling and waving trees occurs, our approach mitigates both of the challenges in a better way than others. In the case of dynamic background fall sequence, a significant improvement in the segmentation result can be observed for our method over others, especially compared to GMM, where many noisy background pixels are classified in clusters. We also compare the segmentation results of GMM, LOBSTER and our method on the CDnet 2014 dataset, which contains more challenging situations compared with CDnet 2012 dataset; the results are shown in Fig. 11. The first column is the input frame from difference sequences, the second column is the corresponding ground-truth, the third column is the segmentation results of our method, the fourth column represents the segmentation results of LOBSTER [9] and the last column shows the segmentation results of GMM [4]. In foreground aperture challenge (the snowfall sequence), which is to segment a moving object in a uniform coloured regions, our method extracts the complete object than the other methods, where many false negatives are detected. In FluidHighway sequence, which is taken from urban traffic monitoring at night, our method handles the glare effect from car headlights better than others. In ContinusPan sequence from PTZ category, although the camera is not static, the overall performance of our method is also better than LOBSTER and GMM. In turbulence sequence, we observe that our method captures less of background turbulence and segments moving objects perfectly, this again demonstrates the effectiveness of our method.

5.5 Processing speed and memory usage

Background subtraction is often the first stage of many computer vision applications, processing speed and memory requirements are critical information for researches to be considered before choosing which method to implement. To achieve real-time performance, the background subtraction methods must be fast, light and efficient. In this section, we give a detailed analysis of the time and space complexity of our method. To do so, our method has been implemented in C++ and use the OpenCV [28] image processing library. All the experiments are carried out on a 4.0 Ghz Intel Core-i7 6700K with 32 GB RAM and Ubuntu 14.04 operating system. As the sequences do not have the same size, we reported the number of FPS over different frame size in Table 5 and we can see the results show the real-time performance. Since our method operates at the pixel level, it has the potential to achieve much higher FPS for hardware implementation or parallel implementation.

Another key information is the memory usage of our method. Consider a colour image I with the size of $W \times H$, assuming that the background model samples for each pixel are N, then the space complexity of our method is $\mathcal{O}(NWH)$. Each background sample requires three bytes of memory to store the colour information and six bytes of memory to store the ESILBP information. For a colour sequence with a frame size of 720×576 (e.g.: PETS2006 sequence), if the number of background samples is set to 50, we can calculate the memory requirements of our method would be around 180 MB. For embedded platform, decrease the number of background samples can dramatically reduce the memory usage.

© The Institution of Engineering and Technology 2018

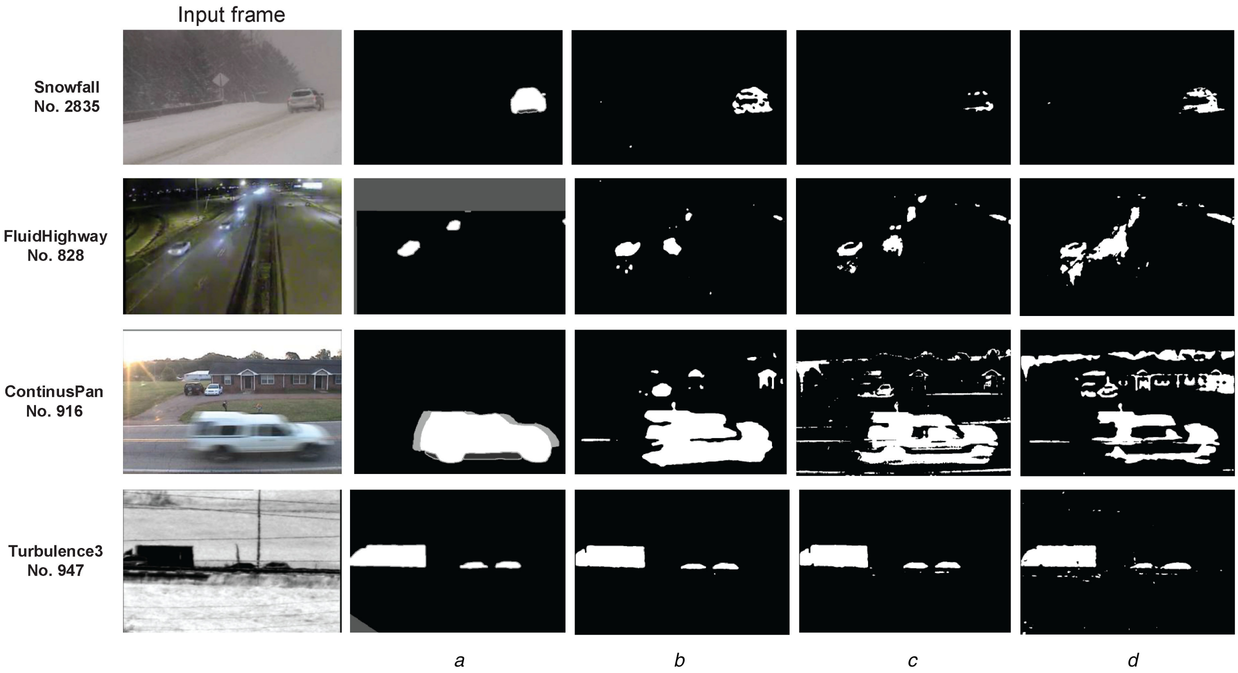

Fig. 11 *Qualitative performance comparison for various sequences of the CDnet 2014 dataset*
(a) Ground-truth, ***(b)*** Segmentation results of our method, ***(c)*** Segmentation results of LOBSTER, ***(d)*** Segmentation results of GMM

Table 5 ESILBP processing speed in terms of frame per seconds (FPS)

Sequence	Frame size	FPS
highway	320×240	97
pedestrians	360×240	89
port_0_17fps	640×480	28
blizzard	720×480	25
PETS2006	720×576	19

6 Conclusion

In this paper, we present a hybrid background subtraction algorithm which combines texture features and colour features. First, a new texture feature called ESILBP is proposed to work against illumination variations and shadows. Second, to overcome the limits of texture features on uniform regions, a photometric invariant colour measurement is proposed. The colour feature and ESILBP feature can compensate each other to achieve better performance. Experiments evaluated on the CDnet2012 and 2014 dataset show that our algorithm outperforms many recent state-of-the-art background subtraction algorithms in the metric of *F-measure* scores. In our future work, we will integrate our pixel representations with more complex feedback model update strategy. Also, region-level or object-level analyses could be used to improve the foreground consistency, more sophisticated post-processing operations like Markov random field could also help refining our segmentation result.

7 Acknowledgments

The authors thank the anonymous reviewers for their helpful feedback. This research is supported by the National Science Foundation of China under grant no. 61401425.

8 References

[1] Bouwmans, T.: ‘Traditional and recent approaches in background modeling for foreground detection: an overview’, *Comput. Sci. Rev.*, 2014, **11**, pp. 31–66

[2] Sobral, A., Vacavant, A.: ‘A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos’, *Comput. Vis. Image Underst.*, 2014, **122**, pp. 4–21

[3] Elgammal, A., Duraiswami, R., Harwood, D.*, et al.*: ‘Background and foreground modeling using nonparametric kernel density estimation for visual surveillance’, *Proc. IEEE*, 2002, **90**, (7), pp. 1151–1163

[4] Stauffer, C., Grimson, W.E.L.: ‘Adaptive background mixture models for real-time tracking’. IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 1999, vol. 2, pp. 246–252

[5] Kaewtrakulpong, P., Bowden, R.: ‘An improved adaptive background mixture model for real-time tracking with shadow detection’. Video-Based Surveillance Systems, Springer, Boston, MA, 2002, pp. 135–144

[6] Zivkovic, Z., Van Der Heijden, F.: ‘Efficient adaptive density estimation per image pixel for the task of background subtraction’, *Pattern Recognit. Lett.*, 2006, **27**, (7), pp. 773–780

[7] Lee, D.S.: ‘Effective Gaussian mixture learning for video background subtraction’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (5), pp. 827–832

[8] Heikkila, M., Pietikainen, M.: ‘A texture-based method for modeling the background and detecting moving objects’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (4), pp. 657–662

[9] St-Charles, P.L., Bilodeau, G.A.: ‘Improving background subtraction using local binary similarity patterns’. IEEE Winter Conf. on Applications of Computer Vision, Steamboat Springs, CO, USA, 2014, pp. 509–515

[10] Ma, F., Sang, N.: ‘Background subtraction based on multi-channel SILTP’. Asian Conf. on Computer Vision, Springer, Berlin, Heidelberg, 2012, pp. 73–84

[11] Narayana, M., Hanson, A., Learned-Miller, E.: ‘Background modeling using adaptive pixelwise kernel variances in a hybrid feature space’. IEEE Conf. on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2104–2111

[12] Shen, J., Yang, W., Lu, Z.*, et al.*: ‘Information integration for accurate foreground segmentation in complex scenes’, *IET Image Process.*, 2012, **6**, (5), pp. 596–605

[13] Ji, Z., Wang, W.: ‘Detect foreground objects via adaptive fusing model in a hybrid feature space’, *Pattern Recognit.*, 2014, **47**, (9), pp. 2952–2961

[14] Yao, J., Odobez, J.M.: ‘Multi-layer background subtraction based on color and texture’. IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 2007, pp. 1–8

[15] Liao, S., Zhao, G., Kellokumpu, V.*, et al.*: ‘Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes’. IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 1301–1306

[16] Barnich, O., Van Droogenbroeck, M.: ‘Vibe: a universal background subtraction algorithm for video sequences’, *IEEE Trans. Image Process.*, 2011, **20**, (6), pp. 1709–1724

[17] Goyette, N., Jodoin, P.M., Porikli, F.*, et al.*: ‘Changedetection. net: a new change detection benchmark dataset’. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 2012, pp. 1–8

[18] Wang, Y., Jodoin, P.M., Porikli, F.*, et al.*: ‘CDnet 2014: an expanded change detection benchmark dataset’. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 2014, pp. 393–400

[19] Kim, K., Chalidabhongse, T.H., Harwood, D.*, et al.*: ‘Real-time foreground-background segmentation using codebook model’, *Real-Time Imaging*, 2005, **11**, (3), pp. 172–185

[20] Maddalena, L., Petrosino, A.: ‘A self-organizing approach to background subtraction for visual surveillance applications’, *IEEE Trans. Image Process.*, 2008, **17**, (7), pp. 1168–1177

[21] Maddalena, L., Petrosino, A.: ‘The SOBS algorithm: what are the limits?’. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 2012, pp. 21–26

© The Institution of Engineering and Technology 2018

[22] St-Charles, P.L., Bilodeau, G.A., Bergevin, R.: 'Subsense: a universal change detection method with local adaptive sensitivity', *IEEE Trans. Image Process.*, 2015, **24**, (1), pp. 359–373

[23] Tan, X., Triggs, B.: 'Enhanced local texture feature sets for face recognition under difficult lighting conditions', *IEEE Trans. Image Process.*, 2010, **19**, (6), pp. 1635–1650

[24] Sheikh, Y., Shah, M.: 'Bayesian modeling of dynamic scenes for object detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (11), pp. 1778–1792

[25] Haritaoglu, I., Harwood, D., Davis, L.S.: 'W4: real-time surveillance of people and their activities', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, **22**, (8), pp. 809–830

[26] Wang, Y., Luo, Z., Jodoin, P.M.: 'Interactive deep learning method for segmenting moving objects', *Pattern Recognit. Lett.*, 2016, **96**, pp. 66–75

[27] Toyama, K., Krumm, J., Brumitt, B., *et al.*: 'Wallflower: principles and practice of background maintenance'. IEEE Int. Conf. on Computer Vision, Kerkyra, Greece, Greece, 1999, pp. 255–261

[28] OpenCV: 'Open source computer vision library', available at http://www.opencv.org

© The Institution of Engineering and Technology 2018