IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 22, 2019, accepted July 5, 2019, date of publication July 9, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927745

Local Compact Binary Count Based Nonparametric Background Modeling for Foreground Detection in Dynamic Scenes

WEI HE[1,2], (Member, IEEE), YONG K-WAN KIM[2], HAK-LIM KO[2], JIANHUI WU[1,2], WUJING LI[1], AND BING TU[1], (Member, IEEE)

[1]School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China
[2]Department of Information and Communication Engineering, Hoseo University, Asan 31499, South Korea

Corresponding authors: Jianhui Wu (jhwu@hnist.edu.cn) and Bing Tu (tubing@hnist.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51704115, in part by the Hunan Provincial Natural Science Foundation under Grant 2019JJ40104 and Grant 2019JJ50211, in part by the Open Fund of Education Department of Hunan Province under Grant 18K086, in part by the Science and Technology Program of Hunan Province under Grant 2016TP1021, in part by the project titled ''Development of Distributed Underwater Monitoring and Control Networks'', funded by the Ministry of Oceans and Fisheries, Korea, and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2018-08-01417) supervised by the IITP (Institute for Information & Communications Technology Promotion).

ABSTRACT Background subtraction is one of the most fundamental and challenging tasks in computer vision. Many background subtraction algorithms work well under the assumption that the backgrounds are static over short time periods but degrade dramatically in dynamic scenes, such as swaying trees, rippling water, and waving curtains. In this paper, we propose an effective background subtraction method to address these difficulties by combining color features with texture features in the ViBe framework. Specifically, we present a novel local compact binary count (LCBC) feature that can capture local binary gray-scale difference information and totally discard the local binary structural information. The effective fusion of color and LCBC information significantly improves the performance of the ViBe model, making it very robust to background variations while still highlighting the moving objects. We further embed the total variation (TV) norm regularization technique into the proposed method, which can enhance the spatial smoothness of foreground objects, thereby further improving the accuracy of the method. We evaluate the proposed method against ten sequences containing dynamic backgrounds and show that our method outperforms many state-of-the-art methods in reducing the false positives without compromising the reasonable foreground definitions. The experimental results on challenging well-known data sets demonstrate that the proposed method works effectively on a wide range of dynamic background scenes.

INDEX TERMS Foreground detection, nonparametric background modeling, local compact binary count, dynamic background, video signal processing.

I. INTRODUCTION

Background subtraction is generally regarded as an effective technique for detecting foreground objects in video sequences. Although it has been extensively studied over the years, background subtraction is still a challenging problem, especially in complex scenes. Among many challenges, difficulties caused by dynamic backgrounds are the main aspects. To deal with these problems, many researchers have reported diverse techniques, a large number of conventional algorithms can be found in recent surveys [1], [2]. Recent efforts focus on two major aspects: one is to develop ingenious background models and the other is to propose reliable feature representations.

The choice of background model is critical to the accuracy of foreground detection. In early works, the parametric background model was built for each pixel individually. In the parametric models [3]–[8], each pixel in the scene can be

The associate editor coordinating the review of this manuscript and approving it for publication was Habib Ullah.

characterized by various parameter distributions. However, the parametric models often fail to handle outdoor scenes with strong background motion. This is mainly because they make restrictive assumptions that the pixel intensity follows a Gaussian distribution, which is not always correct. To address this problem, nonparametric models [9]–[17] were proposed and effectively improved the robustness, due to they can overcome some of the problems inherent in parametric models, such as continuous parameter estimation and the choice of suitable probability distribution functions. The landmark method in this regard called ViBe [13] was proposed by Barnich and Droogenbroeck. With the innovative mechanisms background modeling and the stochastic update strategy, ViBe has shown superior performance in computation speed and detection rate than many other state-of-the-art methods. However, it cannot efficiently handle dynamic backgrounds and noise since it uses only color values of pixels to create background models.

Along with the development and improvement of background models, numerous reliable features [18]–[26] and feature selection techniques [27]–[30] were utilized to better address the challenges of background modeling. One of the most widely used features is the local binary pattern (LBP) [18], which has shown excellent performance in background modeling due to its robustness to local variations and its computational simplicity. To further enhance the computational efficiency, the Center Symmetric LBP (CS-LBP) was proposed in [19]. In [20], the authors extended LBP to Local Ternary Pattern (LTP) by using a threshold around zero to improve the robustness to noise. Scale-Invariant Local Ternary Pattern (SILTP) was proposed in [21], which introduces a scale transform factor into LTP. Local Binary Similarity Pattern (LBSP) was proposed to improve the discriminability by employing inter and intra LBSP in background models [22]. Recently, Local SVD Binary Pattern (LSBP) was proposed in [23], which can exploit the potential structure of the local regions to enhance the robustness to illumination changes and noise. However, the LBP and its variants are hand-crafted and not robust to frequent changes in pixels. To overcome these problems, a learning-based local binary descriptor, named a compact binary face descriptor (CBFD) [24], has been proposed. This learned binary code is more adaptable to the data and has a stronger discriminative power than hand-crafted binary codes. He *et al.* [25] introduced the Local Compact Binary Descriptor (LCBD) to model backgrounds, and achieved encouraging performance.

In this paper, we propose a local compact binary count based nonparametric (LcbcBN) method to effectively detect foreground in dynamic scenes. In this method, we improve upon LCBD and present a novel Local Compact Binary Count (LCBC) descriptor to cope with dynamic backgrounds at the feature level. For each given image, pixel difference vectors (PDVs) in local patches are firstly extracted by computing the difference between each pixel and its neighboring pixels. Then, a feature mapping is learned to project these PDVs into low-dimensional binary vectors in an unsupervised manner. Thus, the redundancy information in PDVs is removed and compact binary codes are obtained. Finally, LCBC is obtained by counting the number of ones in the binary vector. Figure 1 illustrates the pipeline of the LCBC feature learning method. As stated before, ViBe cannot efficiently handle noise and dynamic backgrounds, as it uses only the color values of pixels to create background models. Although the texture features can work well for background modeling in many scenes, they may have limitations on large texture-less objects. On the other hand, the RGB color features are sensitive to illumination variations, but they are suitable for handling some scenes lacking texture. To benefit from the strengths of both feature spaces, in this paper, we combine texture features with RGB color features in the ViBe framework. Therefore, each pixel is characterized by its LCBC feature and color intensities, which can compensate for their respective flaws, resulting in better accuracy in terms of foreground detection in comparison with many state-of-the-art methods. To keep our method's implementation simple, there are no additional improvements to ViBe model other than a simplified Total Variation minimization step.

The main contributions of this work are: a novel local binary feature descriptor, LCBC, which can capture local binary gray-scale difference information and is robust to local variations; and an effective ViBe-based method to deal with highly dynamic scenes, relying on color and LCBC features; and the Total Variation (TV) minimization algorithm to enhance the spatial smoothness of foreground objects. To demonstrate the effectiveness of our method, we compare it with several state-of-the-art methods in ten surveillance dynamic scenes from I2R [31] and CDnet2014 [32] datasets.

The rest of this article is organized as follows. Section II introduces a brief review of the previous work. Section III gives a description of the proposed method. The experiments and result comparisons are discussed in Section IV, followed by the conclusions in Section V.

II. RELATED WORK

Over the recent past, various methods for foreground detection have been proposed. In this paper, we broadly divided the related methods into two categories: traditional methods and recent methods.

One of the most prominent methods in the first category are those based on statistical models. The pioneering technique in these models was proposed by Wren *et al.* [3], who used a Gaussian distribution to model the background. Although this unimodal Gaussian method is very fast, it produces relatively poor segmentation results, and has a limited capacity for modeling real-world changes. To address this problem, a Gaussian mixture model (GMM) was proposed by Stauffer and Grimson [4]. Since then a series of variants [5]–[8] were proposed to improve the efficiency and robustness of the Gaussian model. Zivkovic and Heijden [5] proposed utilizing a Dirichlet prior to estimate the appropriate number of Gaussians for each pixel dynamically. To improve the

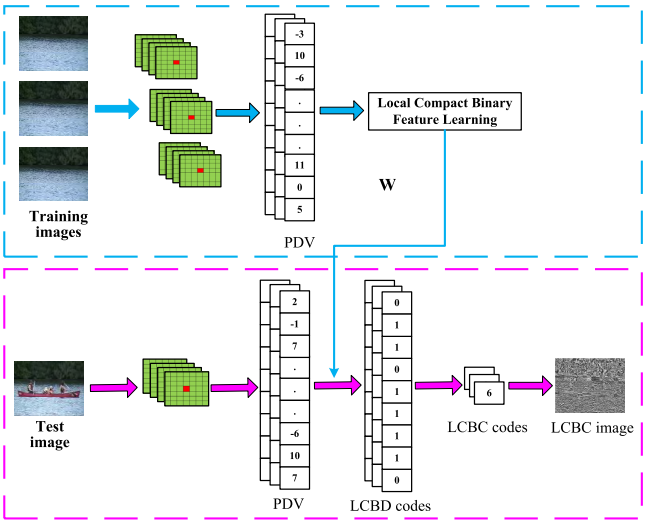


FIGURE 1. The pipeline of LCBC. For each training image, PDVs are first extracted, and then a feature mapping W is learned based on the LCBD. For each test image, the PDVs are also extracted and encoded into binary codes using the learned feature mapping. Finally, we count the number of ones in each LCBD code and extract the corresponding LCBC code.

convergence rate of the Gaussian model, Lee [6] proposed the effective GMM (EGMM). Wang and Miller [7] proposed deriving new update equations by using EM and regularization. Considering the spatial relationship between the neighboring pixels instead of pixels individually, Varadarajan *et al.* [8] proposed a region-based GMM to handle dynamic backgrounds. These Gaussian model-based approaches work well for static scenes even containing gradual changing backgrounds. Kernel density estimation (KDE) is another class of traditional algorithms that has been reported in [9]–[11]. Unlike GMM-based techniques, the KDE based methods utilize the recent history values of each pixel to estimate the probability distribution of the background values. Despite some successes, these methods are often unsuitable for real-time operation, because they are computationally expensive.

One of the most widely used methods in the second category are those based on sample consensus [12]–[17]. Instead of a probability distribution function, Wang and Suter [12] developed a consensus-based method named SACON, which relies on recently observed pixels to determine if the new pixel belongs to foreground or background and then uses a first-in-first-out update strategy for each pixel. As further development, a seminal nonparametric method called ViBe was proposed by Barnich and Droogenbroeck [13]. Although it is superior to the aforementioned methods in speed and memory, ViBe cannot efficiently handle dynamic backgrounds and noise since it uses only color values of pixels to create background models. As we know, color values are not robust to local variations and noise. To date, many improved versions have been proposed [14]–[17] to address its weaknesses. Hofmann *et al.* [14] proposed a pixel-based adaptive segmenter (PBAS) method using an adaptive feedback scheme to cope with more complex environments. St-Charles *et al.* [15], [16] took advantage of feedback mechanisms as well as spatiotemporal binary features and color information resulting in a more accurate background segmentation. More recently, Jiang and Lu [17] proposed a weight-sample-based method, which uses a minimum-weight update strategy and a reward-and-penalty weighting policy to obtain effective change detection. Unfortunately, these sample-based methods are often not well competent under dynamic scenes due to they are sensitive to noise and local variations.

Another class of popular algorithm in the second category is based on fuzzy logic, which has shown promising performance in background subtraction. El Baf *et al.* [33] proposed a specific method combining Type-2 Fuzzy and GMM to handle complex scenes. In another work, Zhao *et al.* [34] proposed the improved Type-2 Fuzzy GMM method by taking the spatial-temporal constraints into account. Chiranjeevi and Sengupta [35] proposed using inter-channel and intra-channel kernel fuzzy correlograms to detect foreground objects. Panda and Meher [36] proposed using fuzzy color difference histogram features to exhibit a significant improvement in the background subtraction.

By considering the low-rank and sparse properties, RPCA has been widely used in foreground detection. Due to dynamic backgrounds, the original RPCA cannot obtain satisfactory results in real scenes. To improve the performance of the original RPCA, new variants of RPCA [37]–[39] have been proposed in recent years. Javed *et al.* presented a spatiotemporal low-rank algorithm [37] for foreground detection, where spatial and temporal graph information is used to achieve excellent performance in dynamic scenes. Liu *et al.* [38] proposed a new low-rank and structured-sparse matrix decomposition method to refine RPCA-based foreground detection. Javed *et al.* [39] encoded spatiotemporal constraints by regularizing spectral graphs for estimating the robust background model.

Recently, unprecedented methods based on deep learning algorithms [40]–[43] have been proposed and obtain excellent results on various change detection scenarios. Braham and Droogenbroeck [40] proposed a novel foreground detection method using convolutional neural networks (CNNs), where a specific network is constructed for a certain scene. Sultana *et al.* [41] proposed a unified method by combining Generative Adversarial Network (GAN) with a semantic inpainting network. Ullah *et al.* [42] proposed a spatio-temporal deep CNN for pedestrian segmentation, which can exploit temporal information for spatial segmentation. A more detailed discussion of these deep learning-based foreground detection techniques can be found in recent surveys [43].

There still exists many other algorithms to segment foreground. Sahin *et al.* [44] proposed using a sliding window and self-regulated learning-based updating strategy for change detection. In addition, several effective methods using social interaction information of different moving entities

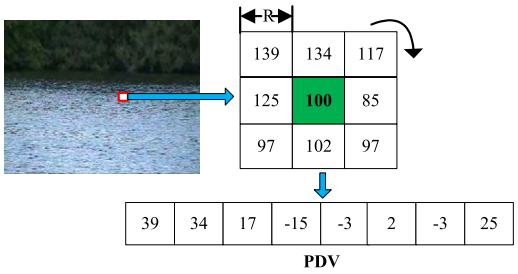


FIGURE 2. The method of extracting a PDV from a given image patch using our method. For each pixel in an image, we first compute the differences between the center pixel and its $(2R+1)*(2R+1)$ neighboring pixels, where R is the radius of the patch, selected as 1 in this figure for easy illustration. These differences then form a vector which becomes the PDV feature of the pixel.

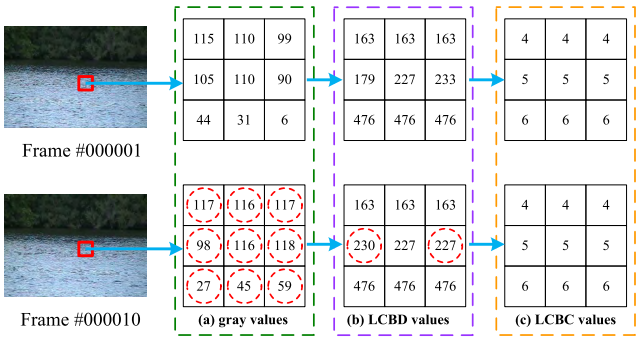


FIGURE 3. Comparison of the robustness of LCBD and LCBC between the first and the tenth frame of the"canoe" sequence, which contains rippling water.

for pedestrian motion segmentation have been proposed in [45]–[47].

III. THE PROPOSED METHOD

A. BRIEF REVIEW OF THE LCBD

As described in [24], the LCBD can be obtained as follows: first, for each given image, pixel difference vectors (PDVs) in local patches are extracted. N PDVs constitute the training set $\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_N]$, where $\mathrm{x}_n \in \mathbb{R}^d$ $(1 \le n \le N)$ is the nth PDV. Figure 2 illustrates an example of the extraction of a PDV from a given patch. Then, K hash functions are learned in an unsupervised manner to map x_n into a K-dimensional binary vector $\mathrm{b}_n = [b_{1n}, b_{2n}, \cdots, b_{Kn}]^T \in \{0,1\}^{K\times 1}$. Let $\mathrm{w}_k \in \mathbb{R}^d$ be the projection vector for the kth function. The kth binary code b_{nk} of x_n can be computed as:

$$\mathrm{b}_{nk} = 0.5 \times (\mathrm{sgn}(\mathrm{w}_k^T \mathrm{x}_n) + 1) \tag{1}$$

where

$$\mathrm{sgn}(v) = \begin{cases} 1, & if\ v \ge 0 \\ -1, & otherwise \end{cases} \tag{2}$$

To make them compact and discriminative, the binary codes can be formulated using the following optimization objective function:

$$\begin{aligned} \min_{\mathrm{W}_k} J(\mathrm{w}_k) &= J_1(\mathrm{w}_k) + \lambda_1 J_2(\mathrm{w}_k) + \lambda_2 J_3(\mathrm{w}_k) \\ &= -\sum_{n=1}^{N} \|b_{nk} - \mu_k\|^2 \\ &\quad + \lambda_1 \sum_{n=1}^{N} \left\| (b_{nk} - 0.5) - \mathrm{w}_k^T \mathrm{x}_n \right\|_F^2 \\ &\quad + \lambda_2 \left\| \sum_{n=1}^{N} (b_{nk} - 0.5) \right\|^2 \end{aligned} \tag{3}$$

where N is the number of PDVs which are extracted from the training samples, μ_k is the mean of the kth binary code of all the PDVs, λ_1 and λ_2 are two parameters to balance the effects of different terms in (3).

B. LOCAL COMPACT BINARY COUNT (LCBC)

Although it has proven to be a powerful local binary feature descriptor [24] and has shown excellent performance in background modeling [25], LCBD may fail to deal with strong background motion. Therefore, we introduce other more robust characteristics to extend LCBD, making it more suitable for application to dynamic backgrounds.

In the original LCBD, each PDV is projected into a low-dimensional binary vector using a learned feature mapping. In the present LCBC, we only count the number of ones in the binary vector instead of encoding them (Figure 1). The main difference between LCBD and LCBC is that LCBD uses a local binary pattern to encode each pixel while LCBC merely counts the number of ones in the LCBD code. However, their meanings are very different: LCBD can extract local structure information, while LCBC merely focuses on the local binary difference information. Compared to LCBD, LCBC is more robust to dynamic backgrounds because it only extracts the local binary grayscale difference information and ignores the local binary structural information. The performance of LCBD and LCBC is illustrated in Figure 3 for "canoe" sequence from CDnet2014 dataset. As can be seen, two pixel blocks are labeled by red at the same positions of the two frames. Due to the rippling water, all these gray values are changed (the circled red pixels in Figure 3 (a)). LCBD can cope with most of these variations, but two errors still occur (the circled red pixels in Figure 3 (b)). As can be seen in Figure 3 (c), LCBC is more robust than LCBD and produced no errors.

To illustrate the discriminative power of LCBD and LCBC, we test them on another dynamic background sequence. In Figure 4, two video frames are taken from the "overpass" sequence from CDnet2014 dataset, where two pixel blocks are labeled by red and blue squares at the same position in the two frames. The red blocks both contain background motion, while in the blue blocks one contains static background and the other contains foreground. For each pixel in a given pixel block, the LCBD and the LCBC can be extracted. The histogram can then be computed for each block. In Figure 4(b), the four histograms of LCBD are labeled with red and green, corresponding to the red and blue squares, respectively.

The four histograms of LCBC are illustrated in Figure 4(c). Finally, the intersection of the histograms is used to measure the similarity of the two blocks. For LCBD, the inter-frame similarity of the red and the blue blocks are 51 and 17, respectively. For LCBC, the inter-frame similarity is 97 for the red blocks and 43 for the blue blocks. Although the red blocks for the two frames are both dynamic backgrounds, the inter-frame similarity based on LCBD does not reveal this well, but LCBC can detect this similarity. Although the LCBC codes do not represent micro-structure, the LCBC features can distinguish the different distributions of local pixels, which is similar to Local Binary Count (LBC) [48].

C. REPRESENTATION OF BACKGROUND MODEL

To overcome the problem of ViBe algorithm being susceptible to noise and illumination changes, we model each pixel using both LCBC and RGB intensities and combine them with ViBe for segmenting the foreground from the background accurately. The ViBe model is well suited to the multi-feature description of pixels. In our method, the background model $B(x_i)$ of each pixel is composed of an array of M recently observed background samples:

$$B(x_i) = \{B_1(x_i), \cdots, B_k(x_i), \cdots, B_M(x_i)\} \tag{4}$$

A pixel x_i is classified as background if its RGB value $Int(x_i)$ and LCBC value $LCBC(x_i)$ are both closer than the corresponding thresholds of at least $\#min$ of the M background samples.

$$\begin{aligned}(dist(LCBC(x_i), BLCBC_k(x_i)) < R_{LCBC}) \\ \&\&(L1dist(Int(x_i), BInt_k(x_i)) < R_{Int})\end{aligned} \tag{5}$$

when Equation 5 comes to 1 we get a match. $\#min$ is the minimum count of matches required for a background classification. If the matches are less than $\#min$, pixel is classified as foreground. We use Euclidean distance for LCBC comparison, and L1 distance for color comparison, for computational efficiency.

Updating the background model B is critical, to account for changes in background. We use a random update strategy, as used in ViBe. For a pixel x_i classified as background, the corresponding background model values $BInt_k(x_i)$ and $BLCBC_k(x_i)$; $k \in [1, M]$, chosen at random from a uniform distribution, are replaced by the current pixel value $Int_k(x_i)$ and $LCBC_k(x_i)$, respectively. This update is performed with probability $p = 1/T$. The parameter T defines the update rate: the higher the value of T, the less likely it is that a pixel will be updated. As in [13], we set $T = 16$. A randomly chosen pixel from the eight-pixel neighborhood is also updated with a probability of $1/T$, with the background model at this neighboring pixel being replaced by its current color intensity and LCBC value. The detailed procedure of the proposed method using both LCBC feature and color intensities is described in Algorithm 1.

Algorithm 1 Background Subtraction Using Color and LCBC Features

Initialization:
for *thefirstN* frames **do**
  Extract the projection matrices $\boldsymbol{W}$ using Equation (3)
end for
for the first frame **do**
  Extract the LCBC feature
  Extract M LCBC values and M color intensities for each pixel, and push them into $BLCBC_k(x_i)$ and $BInt_k(x_i)$ respectively, as the background model
end for
Mainloop
for each new video frame **do**
  for each pixel of the current frame **do**
    Extract $LCBC(x_i)$ and $Int(x_i)$
    $matches = 0$
    $index = 0$
    while (($index$ <= M) && ($matches < \#min$)) **do**
      compute $dist(LCBC(x_i), BLCBC_k(x_i))$ and $L1dist(Int(x_i), BInt_k(x_i))$
      if $dist(x_i) < R_{LCBC}$ && $L1dist(x_i) < R_{Int}$) **then**
        $matches = matches + 1$
      end if
      $index = index + 1$
    end while
    if ($matches < \#min$) **then**
      Foreground
    else
      Background
    end if
  end for
  TV-minimization
end for

D. TV-NORM MINIMIZATION

The proposed method is a pixel-wise method, which means that spatial correlations between pixels are ignored. In the proposed method, spatial smoothness is considered to further improve its accuracy on foreground object detection. There are many strategies to enhance the smoothness of an image, such as L2 regularization [49], Total Variation (TV) minimization [50], and Markov random field (MRF) [58]. Considering effectiveness and efficiency, TV-minimization approach is employed in the proposed method. The resulting output can benefit from spatial smoothing, which we performed using Total Variation (TV) minimization algorithm. As in previous work [50], for a foreground frame obtained using our method, we calculate the TV-minimization problem as follows:

$$FG' = \arg\min_F \frac{1}{2}\|F - FG\|_2^2 + \lambda\|F\|_{TV} \tag{6}$$

where $\|\cdot\|_{TV}$ is the TV norm, FG is the foreground obtained by our method, F is the image to be recovered, and λ is a pos-

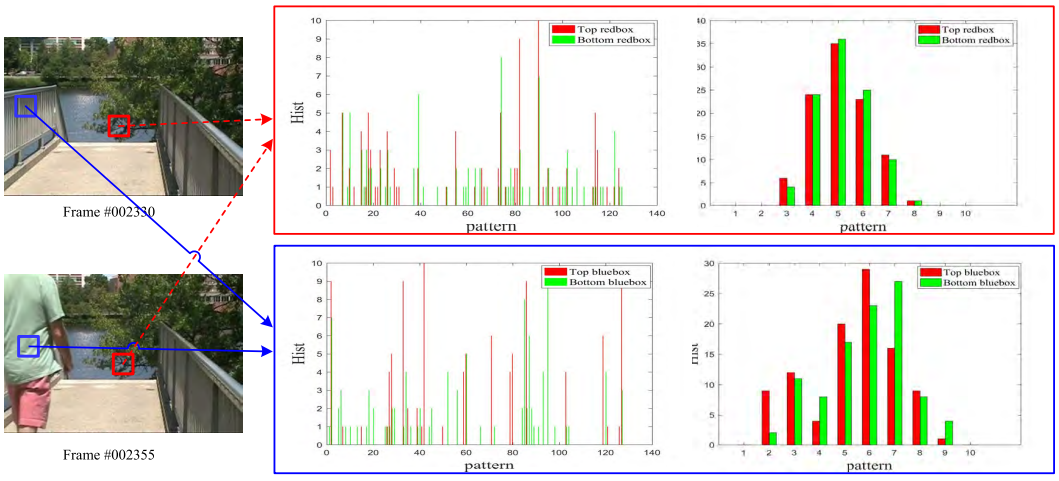


FIGURE 4. Comparison of discriminative power of LCBD and LCBC on the "overpass" sequence containing strongly swaying branches. (The "Top redbox" and "Bottom redbox" indicate the red box in Frame #002330 and Frame #002355, respectively, and these are also true for "Top bluebox" and "Bottom bluebox".).

itive weight for balancing the corresponding terms in (6). The optimization function (6) can be solved by a TV-thresholding algorithm, with the parameters having the similar settings as those in [49].

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed method (LcbcBN), using typical dynamic scenes. We implemented LcbcBN in MATLAB, running on the Windows10 platform with Intel Core i7-7700 CPU@3.60GHz, 3.60GHz and 16GB RAM. The effectiveness of the proposed method is demonstrated on publicly available challenging video sequences containing strong background motion, including "Curtain", "Fountain", "Campus", "WaterSurface", "boats", "canoe", "fall", "fountain01", "fountain02", and "overpass", obtained from the I2R [31] and CDnet2014 [32] datasets. The details of these ten dynamic background sequences are shown in Table 1.

A. EVALUATION METRICS

To get an accurate evaluation of the effectiveness of the proposed method, the following seven evaluation metrics introduced in [32] are utilized:

1) Recall (Re): $Re = \frac{TP}{TP+FN}$
2) Specificity (Sp): $Sp = \frac{TN}{TN+FP}$
3) False Positive Rate (FPR): $FPR = \frac{FP}{FP+TN}$
4) False Negative Rate (FNR): $FNR = \frac{FN}{FP+TN}$
5) Percentage of Wrong Classifications (PWC): $PWC = 100 * \frac{(FN+FP)}{(FN+FP+TN+TP)}$
6) Precision (Pr): $Pr = \frac{TP}{TP+FP}$
7) F-Measure (FM): $FM = \frac{2\times Pr\times Re}{Pr+Re}$

where TP is the total number of true positives, FN is the total number of false negatives, TN is the total number of true negatives, and FP is the total number of false positives. Recall, also known as detection rate, represents the percentage of detected true positives as compared to the total number of true positives in the ground truth. Specificity, also known as true negative rate, represents the percentage of correctly detected negatives over the total number of negatives. FPR, known as fall-out, measures the percentage of false positives out of the total number of negatives. FNR, known as miss rate, gives the proportion of false negative over the total number of positives. PWC represents the percentage of wrongly detected pixels over the total pixels. Precision, also known as positive prediction, that gives the percentage of detected true positives as compared to the total number of pixels detected by the method. In this paper, FM is used primarily, as it takes both the recall and precision into account, and has been widely accepted for the evaluation of detection results.

B. PARAMETER SETTINGS

Since the proposed method consists of several tunable parameters, a good set of parameters for optimal system performance need to be adjusted. For LCBC, we set one unique optimal set of parameters as follows: the radius of the neighborhood $R = 3$, extracted a 48-dimensional PDV for each pixel, and mapped them into K-bit (where $K = 9$) binary codes, and the parameter $\lambda_1 = 0.001$, and $\lambda_2 = 0.0001$. For ViBe algorithm, the parameter $\#min = 2$, since this value has been demonstrated to be optimal [13]–[16], and the other three parameters could be tuned for optimal performance: the number of components of the background model $M \in \{20, 25, 30, 40, 50, 60\}$, the Euclidean distance thresholding $R_{LCBC} \in \{2, 3\}$ and L1 distance thresholding $R_{Int} \in \{20, 25, 30, 35\}$. Figure 5 shows the FM performance of the proposed method on each sequence with different M, R_{Int}, and R_{LCBC} parameter settings.

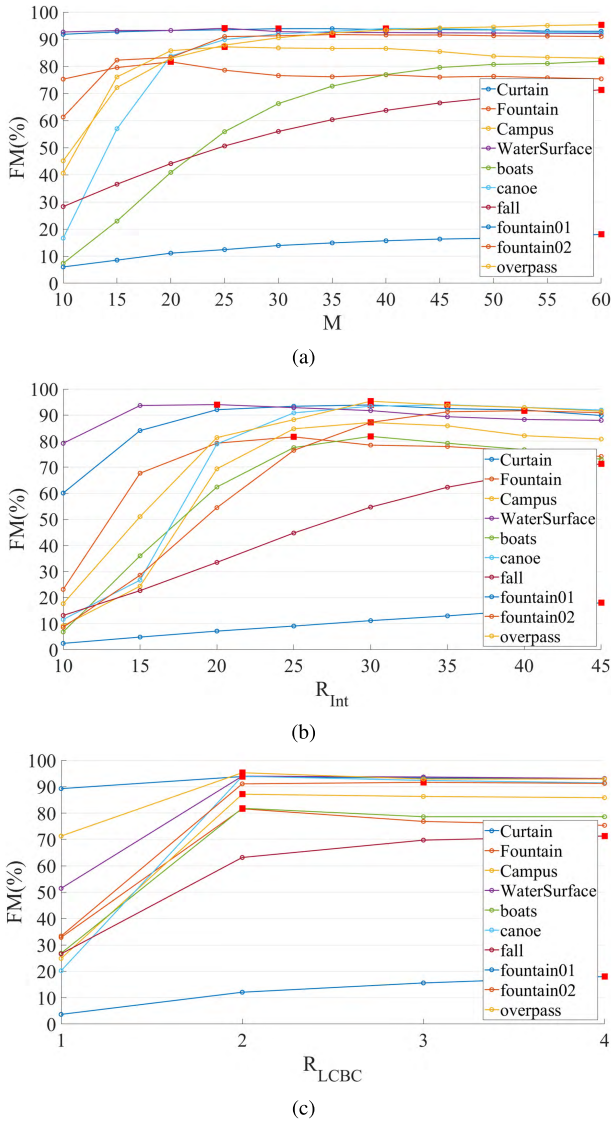


FIGURE 5. *FM* performance of the proposed method on ten sequences with different parameter settings, and the optimal *FM* values are marked by the red squares.

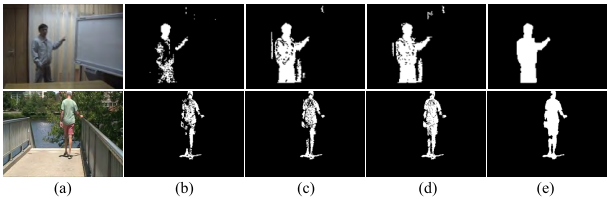
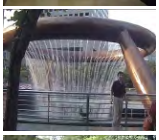
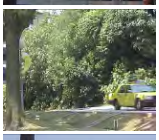
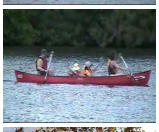
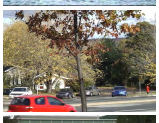
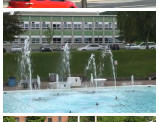


FIGURE 6. Segmentation results obtained with our method on the "Curtain#23786" (top row) sequence from I2R dataset [31] and the "overpass#2508" (bottom row) sequence from CDnet2014 dataset [32]. These sequences both contain important dynamic background elements. (a) Input frame, (b) used Gray only, (c) used RGB only, (d) used RGB and LCBC but no TV, (e) used RGB, LCBC, and TV.

C. EFFECTS OF SEPARATE STAGES

In order to better understand the performance of the proposed method, the effects of each stage of the pipeline are analyzed. Figure 6 shows some segmentation results when only using gray information (Figure 6 (b)), only using the color

TABLE 1. Details of ten video sequences.

Sequence	frame	size	Number of frames
Curtain [31]		128 × 160	2964
Fountain [31]		128 × 160	523
Campus [31]		128 × 160	1439
WaterSurface [31]		128 × 160	633
boats [32]		240 × 320	7999
canoe [32]		240 × 320	1189
fall [32]		480 × 720	4000
fountain01 [32]		288 × 432	1184
fountain02 [32]		288 × 432	1499
Overpass [32]		240 × 320	3000

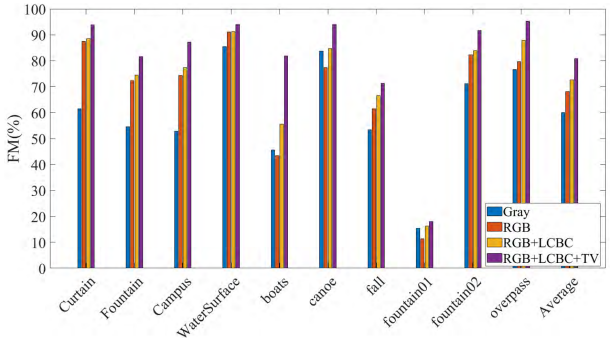


FIGURE 7. *FM* comparisons of sequences with dynamic background from the I2R [31] and CDnet2014 [32] datasets for demonstrating the effects of each stage of the pipeline, using Gray only, RGB only, RGB and LCBC, and RGB, LCBC, and TV configurations of our method.

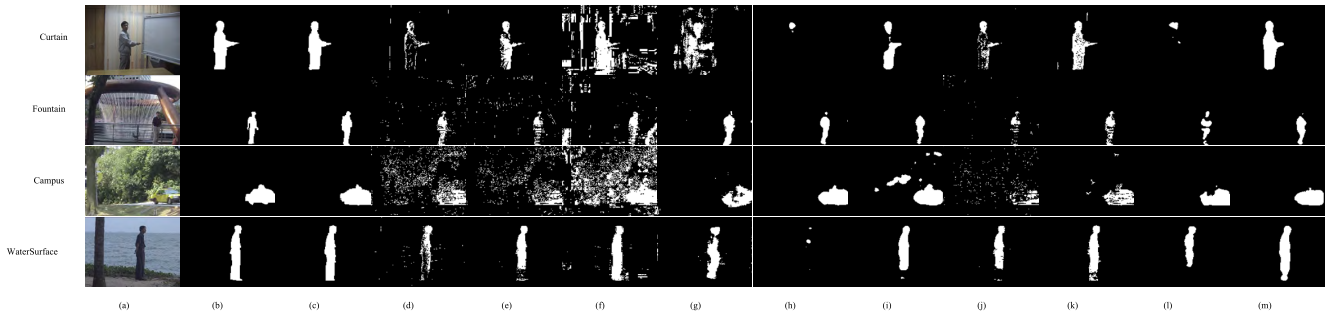


FIGURE 8. Segmentation results of four dynamic background sequences from I2R dataset [31]. (a) Input video frames, (b) ground truth, (c) LcbcBN, (d) GMM [4], (e) IGMM [5], (f) KDE [9], (g) LBP [18], (h) MultiLayer [51], (i) LOBSTER [52], (j) ViBe [13], (k) ViBe-rgb [13], (l) PBAS [14], (m) SuBSENSE [15].

information (Figure 6 (c)), using both LCBC and color information (Figure 6 (d)), and using LCBC, color information and TV processing (Figure 6 (e)). Since the page limitation, in Figure 6, a short qualitative analysis on ''Curtain'' and ''overpass'' sequences is provided. In fact, all ten dynamic background sequences were tested, and a full quantitative (*FM*) evaluation is presented in Figure 7.

The ''Curtain'' is an indoor dynamic background sequence containing a set of strongly waving curtains, and the ''Overpass'' is an outdoor dynamic background sequence with swaying trees and shimmering water surface. To handle those typical dynamic background changes, methods should tolerate background variations while maintaining sensitivity to detect real foreground objects. As it can be seen in Figure 6(b), the algorithm using only gray failed to deal with such dynamic scenes, as many foregrounds are absorbed into background. Figure 6(c) shows that the algorithm using color intensities can detect most of the foregrounds, but it also produces some bad foreground blobs. After combining color features with LCBC features in ViBe algorithm (Figure 6(d)), we obtain better performance, as they can compensate for their respective defects. However, we also observe that the problem of foreground false positives and false negatives cannot be completely solved. It is noticed that after adding TV-minimization algorithm (Figure 6(e)), the version with the TV, is much more reliable. According to the average *FM* shown in Figure 7, the RGB+LCBC+TV can promote the detection results and obtain superior performance over the other configurations. To better illustrate the key contributions of the proposed method, we compared the results of different ViBe-based model configurations in Table 2. Note that these scores are the averages obtained over ten sequences. It is noticeable that ViBeRGB+LCBC+TV has the highest average *FM*, which is a more than 20% improvement compared to the original ViBe algorithm (ViBe-Gray). Since LCBC is based on the local binary grayscale difference information rather than local binary grayscale magnitudes, it is more robust to local variations than LCBD. From Table 2 it can be seen that the RGB+LCBC configuration (fourth row) has achieved more than 6% improvement in accuracy compared to the RGB+LCBD configuration (third row). Among all tested feature configurations, RGB+LCBC+TV is the best choice for dynamic background subtraction.

TABLE 2. Average performance comparison of different ViBe-based model configurations on ten dynamic scenes.

Configuration	Pr	Re	FM
Gray	0.5315	0.7926	0.6003
RGB	0.7245	0.6848	0.6808
RGB+LCBD	0.6170	0.8074	0.6667
RGB+LCBC	0.7700	0.7041	0.7312
RGB+LCBC+TV	**0.8555**	**0.7953**	**0.8088**

D. QUALITATIVE RESULTS

To properly evaluate our method, we compared the proposed method with ten popular state-of-the-art background methods: GMM [4], IGMM [5], KDE [9], LBP [18], SWCD [44], Multilayer [51], LOBSTER [52], AAPSA [53], IUTIS-2 [54], and CL-VID [55]; and six methods based on sample consensus: ViBe [13], ViBe-rgb [13], PBAS [14], SuBSENSE [15], WeSamBE [17], and SBBS [56]. We have used the BGSlibrary [57] for the following algorithms: GMM, IGMM, KDE, Multilayer, LOBSTER, ViBe-gray, PBAS, and SuBSENSE. We have implemented the LBP method and ViBe-rgb method ourselves. The other results can be downloaded online via the CDnet website (http://www.changedetection.net/).

The first row of Figure 8 gives the detection results of the ''Curtain'' sequence, which contains significant motion of the curtains. Obviously, the proposed method worked very well in such a scene and the detection results are almost the same to the ground truth, and are superior to the other methods. SuBSENSE can detect most of the foregrounds, but it also produced some false positives. GMM, IGMM, MultiLayer, LOBSTER, ViBe, and PBAS failed to correctly detect the person, while KDE and LBP produced many false positives. The second row depicts a person walking in front of a spouting fountain. One can see that the proposed method can suppress the dynamic background and obtained the best results. GMM, IGMM, and KDE detected many false detections, and the other seven methods lost many true foregrounds. The third row gives the detection results of the sequence ''Campus'', which contains swaying tree branches. As illustrated,

TABLE 3. The evaluation results of the proposed method on the ten dynamic sequences.

Dataset	Category	Re	Sp	FPR	FNR	PWC	Pr	FM
I2R[31]	Curtain	0.9117	0.9414	0.0586	0.0883	7.8025	0.9672	0.9387
	Fountain	0.8781	0.8382	0.1618	0.1219	14.697	0.7626	0.8163
	Campus	0.9000	0.9167	0.0833	0.1000	8.8867	0.8448	0.8716
	WaterSurface	0.9312	0.9190	0.0810	0.0688	7.3458	0.9494	0.9402
CDnet2014[32]	boats	0.7806	0.9992	0.0008	0.2194	0.2174	0.8599	0.8183
	canoe	0.9394	0.9978	0.0022	0.0606	0.4277	0.9398	0.9396
	fall	0.8270	0.9911	0.0089	0.1730	1.1791	0.6267	0.7131
	fountain01	0.5402	0.9963	0.0037	0.4598	0.4076	0.1082	0.1803
	fountain02	0.8939	0.9999	0.0001	0.1061	0.0349	0.9408	0.9168
	Overpass	0.9524	0.9994	0.0006	0.0476	0.1255	0.9538	0.9531
	average(I2R)	0.9053	0.9038	0.0962	0.0948	9.6830	0.8810	0.8917
	average(CDnet2014)	0.8223	0.9973	0.0027	0.1778	0.3987	0.7382	0.7535
	average(I2R+CDnet2014)	0.8555	0.9599	0.0401	0.1446	4.1124	0.7953	0.8088

TABLE 4. The performance comparison of the proposed method and ten state-of-the-art algorithms on the four dyanmic background sequences from I2R dataset [31].

Algorithms	Curtain			Fountain			Campus			WaterSurface		
	Re	Pr	FM	Re	Pr	FM	Re	Pr	FM	Re	Pr	FM
GMM [4]	0.5534	0.7432	0.6344	0.7092	0.8435	0.7705	0.6764	0.5548	0.6096	0.5256	0.8298	0.6436
IGMM [5]	0.5624	0.9805	0.7148	0.5824	0.6860	0.6300	0.4897	0.5315	0.5098	0.7753	0.9911	0.8700
KDE [9]	0.8710	0.3127	0.4598	0.7390	0.4515	0.5605	0.9585	0.1692	0.2876	0.9578	0.8171	0.8818
LBP [18]	0.6367	0.5115	0.5671	0.8604	0.7298	0.7897	0.6374	0.7877	0.7046	0.8402	0.7791	0.8085
MultiLayer [51]	0.3600	0.9427	0.5210	0.7161	0.8699	0.7855	0.8844	0.8330	0.8579	0.3453	0.9465	0.5060
LOBSTER [52]	0.6546	0.9804	0.7850	0.6372	0.7721	0.6982	0.8260	0.5410	0.6538	0.8266	0.9454	0.8820
ViBe [13]	0.4439	0.9970	0.6143	0.3902	0.9091	0.5460	0.3712	0.9175	0.5286	0.7482	0.9980	0.8553
ViBe-rgb [13]	0.8439	0.9083	0.8749	0.6819	0.7714	0.7239	0.7130	0.7757	0.7431	0.8806	0.9425	0.9105
PBAS [14]	0.4051	0.9608	0.5699	0.6501	0.8194	0.7250	0.7436	0.8171	0.7786	0.6272	0.9228	0.7468
SuBSENSE [15]	0.8521	0.9642	0.9047	0.7656	0.8606	0.8103	0.6543	0.8942	0.7557	0.8810	0.9165	0.8984
LcbcBN	0.9117	0.9672	**0.9387**	0.8781	0.7626	**0.8163**	0.9000	0.8448	**0.8716**	0.9312	0.9494	**0.9402**

the results of the proposed method, SuBSENSE, and MultiLayer outperformed the other approaches. GMM, IGMM, KDE, and LOBSTER produced many false positives, and LBP, ViBe, ViBe-rgb, and PBAS failed to obtain complete foreground. The last row represents a scene containing a large area of water rippling. The results have shown that the proposed method can obtain better performance in dealing with such dynamic background than others. It is noted that GMM, MultiLayer, and PBAS cannot handle such a scene resulting in most foregrounds to be absorbed into the background.

To provide a better evaluation, as shown in Figure 9, the proposed method was further tested on the other six typical dynamic background sequences, which are from the dynamic background category of the CDnet 2014 dataset [32]. These sequences depict outdoor scenes where there is significant dynamic motion in the background such as boats moving on shimmering water (boats sequence and canoe sequence), cars passing behind spouting fountains (fountain01 and fountain02 sequences), and pedestrians, cars, and trucks passing through scenes with trees shaking in the wind (fall sequence and overpass sequence). The first two columns of the Figure 9 represent boats on the shimmering water, one can see that only our method and SWCD performed relatively well, as it can remove water motion without sacrificing real foreground. LBP and ViBe produced many false alarms, while the other methods lost many foregrounds. In the ''fall'' scene, the proposed method, SWCD, and the other five sample-based methods (ViBe-rgb, PBAS, SuBSENSE, SBBS, and WeSamBE) can tolerate such background motions and obtained better results than the other methods. The fourth and fifth columns of Figure 9 depict cars passing behind fountains, one can see that SWCD, SuBSENSE, SBBS, and WeSamBE can obtain better results than the other methods on both sequences. The proposed method, GMM, IGMM, KDE, MultiLayer, LOBSTER, IUTIS-2, and CL-VID cannot always remove the dynamic background motions. These methods yielded good results on ''fountain02'' sequence but gave inaccurate segmentation with large false positives on ''fountain01'' sequence. The last sequence, ''overpass'', depicts a pedestrian passing in front of swaying trees, only the proposed method, SWCD, CL-VID, and PBAS can obtain complete foreground regions and are found to be better than the others.

Overall, these sample consensus-based methods perform better than the others in most sequences, but they still produce relatively poor segmentation results in some dynamic backgrounds, such as shimmering water (''boats'', ''canoe'', and ''WaterSurface''), spouting fountain (''Fountain'' and ''fountain01''), and strongly swaying branches (''Campus'' and ''overpass''). In contrast to these approaches, our method

TABLE 5. The performance comparison of the proposed method and sixteen state-of-the-art algorithms on the six dynamic background sequences from CDNET 2014 dataset [32].

Algorithms	boats			canoe			fall		
	Re	Pr	FM	Re	Pr	FM	Re	Pr	FM
GMM [4]	0.7582	0.7014	0.7287	0.8659	0.8982	0.8818	0.8838	0.2892	0.4358
IGMM [5]	0.7004	0.8012	0.7474	0.8533	0.9194	0.8851	0.8560	0.2817	0.4239
KDE [9]	0.6575	0.6089	0.6323	0.8315	0.8823	0.8823	0.8721	0.1875	0.3086
LBP [18]	0.6410	0.3049	0.4132	0.7966	0.5164	0.6266	0.9524	0.3084	0.4659
SWCD [44]	0.9238	0.7877	**0.8503**	0.9402	0.8963	0.9177	0.8515	0.9117	**0.8806**
MultiLayer [51]	0.5425	0.9217	0.6830	0.8518	0.9929	0.9170	0.8952	0.2983	0.4475
LOBSTER [52]	0.4174	0.9857	0.5865	0.8737	0.9882	0.9274	0.8287	0.1360	0.2337
AAPSA [53]	0.6284	0.9757	0.7645	0.7977	0.9973	0.8864	0.6841	0.8376	0.7531
IUTIS-2 [54]	0.5549	0.6319	0.5909	0.6900	0.7397	0.7140	0.8589	0.1783	0.2953
CL-VID [55]	0.7236	0.9101	0.8062	0.9607	0.9014	0.9301	0.9674	0.1300	0.2292
ViBe [13]	0.3464	0.6649	0.4555	0.7449	0.9573	0.8378	0.6660	0.4447	0.5333
Vibe-rgb [13]	0.4237	0.4463	0.4347	0.7899	0.7577	0.7735	0.7902	0.5029	0.6146
PBAS [14]	0.2213	0.9808	0.3611	0.5625	0.9986	0.7196	0.9474	0.8067	0.8714
SuBSENSE [15]	0.5596	0.9106	0.6932	0.6590	0.9933	0.7923	0.8567	0.8758	0.8661
SBBS [56]	0.3316	0.9532	0.4920	0.8846	0.9943	0.9362	0.9130	0.8445	0.8774
WeSamBE [16]	0.4806	0.9578	0.6400	0.4433	0.9935	0.6131	0.7366	0.9089	0.8137
LcbcBN	0.7806	0.8599	0.8183	0.9394	0.9398	**0.9396**	0.8270	0.6267	0.7131
Algorithms	fountain01			fountain02			overpass		
	Re	Pr	FM	Re	Pr	FM	Re	Pr	FM
GMM [4]	0.7973	0.0401	0.0764	0.8717	0.7451	0.8034	0.8294	0.9191	0.8719
IGMM [5]	0.7506	0.0431	0.0815	0.8437	0.7459	0.7918	0.8076	0.9366	0.8673
KDE [9]	0.7930	0.0565	0.1055	0.8528	0.7955	0.8232	0.8003	0.8512	0.8250
LBP [18]	0.7187	0.0244	0.0472	0.7377	0.1014	0.1783	0.8825	0.5306	0.6627
SWCD [44]	0.7122	0.8127	**0.7591**	0.9327	0.9285	0.9306	0.8550	0.8429	0.8489
MultiLayer [51]	0.5877	0.0381	0.0716	0.8099	0.7213	0.7630	0.8631	0.9072	0.8846
LOBSTER [52]	0.7518	0.0895	0.1600	0.8432	0.8054	0.8239	0.8500	0.6103	0.7105
AAPSA [53]	0.5128	0.3880	0.4418	0.9222	0.2220	0.3579	0.7046	0.9808	0.8201
IUTIS-2 [54]	0.9060	0.0368	0.0707	0.9560	0.8317	0.8895	0.8505	0.9203	0.8842
CL-VID [55]	0.9327	0.0256	0.0498	0.9671	0.2919	0.4484	0.9393	0.7772	0.8506
ViBe [13]	0.4091	0.0950	0.1542	0.5591	0.9773	0.7113	0.6362	0.9649	0.7668
Vibe-rgb [13]	0.5562	0.0630	0.1132	0.7726	0.8802	0.8229	0.7931	0.8002	0.7966
PBAS [14]	0.8638	0.2751	0.4173	0.9077	0.9651	0.9355	0.6704	0.9690	0.7925
SuBSENSE [15]	0.8771	0.6599	0.7532	0.9232	0.9658	**0.9440**	0.7852	0.9437	0.8572
SBBS [56]	0.7870	0.6778	0.7283	0.9035	0.9654	0.9334	0.8431	0.9869	0.9094
WeSamBE [16]	0.8886	0.6236	0.7329	0.9394	0.9474	0.9434	0.5890	0.9289	0.7209
LcbcBN	0.5402	0.1082	0.1803	0.8939	0.9408	0.9168	0.9524	0.9538	**0.9531**

copes with dynamic backgrounds reliably, removes background motions effectively, and produces better segmentation results, which are the closest to ground-truth references.

E. QUANTITATIVE EVALUATIONS

To properly evaluate the performance of the proposed method, two illustrations are presented to show our final results. The complete results of the proposed method on all ten dynamic background sequences are shown in Table 3. We can see that the overall performance in the ''Curtain'', ''WaterSurface'', ''canoe'', ''fountain02'', and ''overpass'' sequences are very good, as in these scenes, the *Re*, *Pr*, and *FM* are all above 0.89. Yet, we would like to point out that the ''fountain01'' sequence poses the biggest difficulty, as it consists of many scattered fountains, which cause serious interference with foreground detection. In this case, both the *Pr* score and *FM* score are very low, less than 0.2, which seriously affects the overall *Pr* score and *FM* score of the proposed method.

Table 4 shows the detected results on four dynamic background sequences from I2R. We compared the proposed method with classic and recent methods. Obviously, our method works much better than other methods and obtains the best the average *FM* in all these dynamic background scenes. Note that our method is significantly better than the ViBe-rgb method, this because LCBC can capture stable information in dynamic scenes. These results demonstrate the impact of LCBC features on the performance of ViBe-rgb. In particular, compared with ViBe-rgb method, the improvements made by our method on the ''Campus'' sequence is more than 12%, which is impressive. On the clearly, the LCBC features improve the detected results by color features.

Table 5 reports the performances of the proposed method and other sixteen state-of-the-art methods on the dynamic category of CDnet 2014 dataset [32]. One can see that in three out of six sequences, our method is among top 2 with 1st position in two of them. In ''fountain02'' video sequence, our method ranks sixth among sixteen methods. This is a very good result, according to [15], $FM > 0.8$ is considered an acceptable result. However, since it doesn't add the complex improvements such as the feedback scheme, and a reward-and-penalty weighting strategy, our method cannot work very well in the ''fall'' sequence compared with other sample-based methods (PBAS [14], SuBSENSE [15], SBBS [56], and WeSamBE [16]). For the ''fountain01'' scene, no method obtained an acceptable result ($FM > 0.8$). Unfortunately, our

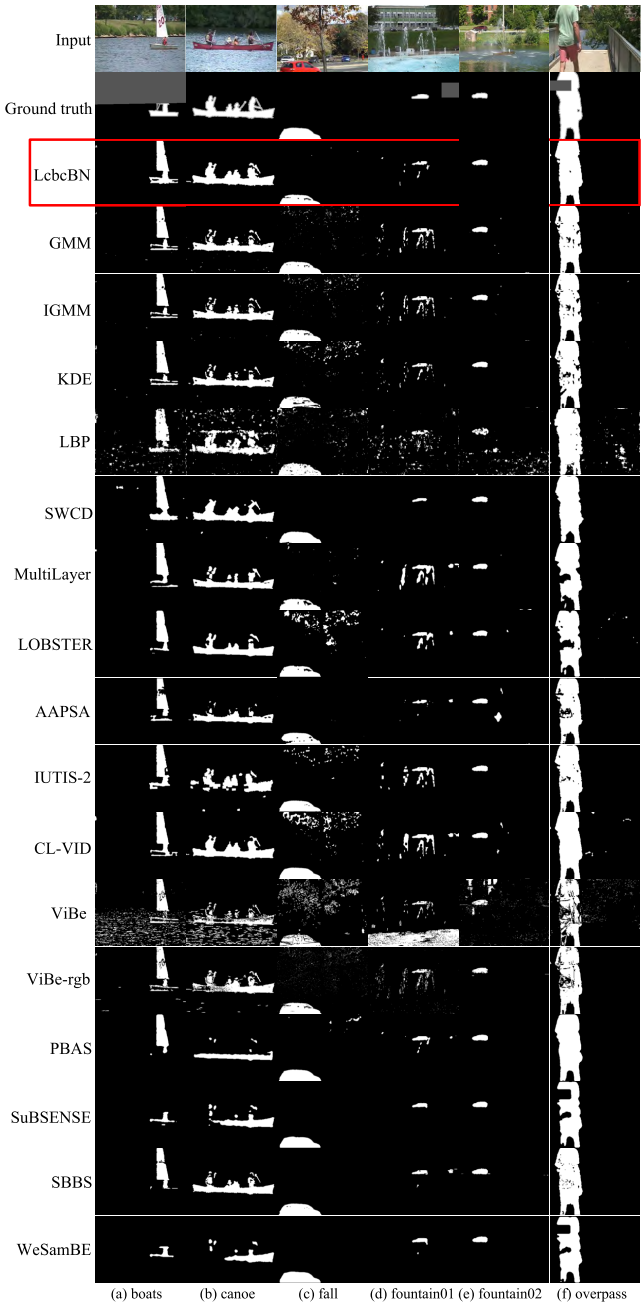


FIGURE 9. Segmentation results on six dynamic background sequences from CDnet2014 dataset [32].

method failed to handle the "fountain01" scene and got a very bad result ($FM < 0.2$). This is due to the fact that the many lasting fountains produce a lot of serious noise, which cannot be handled by ViBe-based methods, resulting in very low Pr and also the FM.

Overall, the proposed method can effectively cope with background motions and significantly outperforms the other approaches for most sequences containing dynamic backgrounds. Note that compared with the original ViBe methods (ViBe and ViBe-rgb), the proposed method obtained encouraging improvement on all video sequences. In particular, combining color features with LCBC in the ViBe framework results in an overall FM 5% improvement (Table 2). This is because LCBC information is considered a good complement to the color features. Moreover, we also benefit from the strengths of the TV algorithm.

V. CONCLUSION

In this paper, we presented a simple and highly effective background subtraction method for foreground detection in highly dynamic scenes. Combining RGB color with LCBC features in the ViBe framework results in a significant increase in accuracy. Moreover, to further improve the efficiency and accuracy of the proposed method, the TV-norm regularization technique is then considered. Experimental results have demonstrated that the proposed method is quite capable of handling scenes with dynamic backgrounds effectively and performs favorably against many recent competitive algorithms when applying challenging dynamic background sequences. Since the proposed method is relatively simple and effective, it will be applied to other computer vision applications, such as human detection, traffic monitoring, and visual tracking.

REFERENCES

[1] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vols. 11-12, pp. 31–66, May 2014.

[2] S. K. Choudhury, P. K. Sa, S. Bakshi, and B. Majhi, "An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios," *IEEE Access*, vol. 4, pp. 6133–6150, 2016.

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

[4] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1999, pp. 2246–2252.

[5] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 2, 2004, pp. 28–31.

[6] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[7] H. Wang and P. Miller, "Regularized online Mixture of Gaussians for background subtraction," in *Proc. Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2011, pp. 249–254.

[8] S. Varadarajan, P. Miller, and H. Zhou, "Region-based mixture of Gaussians modelling for foreground detection in dynamic scenes," *Pattern Recognit.*, vol. 48, no. 11, pp. 3488–3503, Nov. 2015.

[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2000, pp. 751–767.

[10] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.

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

[12] H. Wang and D. Suter, "A consensus-based method for tracking: Modelling background scenario and foreground appearance," *Pattern Recognit.*, vol. 40, no. 3, pp. 1091–1105, Mar. 2007.

[13] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[14] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2012, pp. 38–43.

[15] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[16] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 990–997.

[17] S. Jiang and X. Lu, "WeSamBE: A weight-sample-based method for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2105–2115, Sep. 2018.

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

[19] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, 2009.

[20] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.

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

[22] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier, "Change detection in feature space using local binary similarity patterns," in *Proc. Int. Comf. Comput. Robot Vis.*, May 2013, pp. 106–112.

[23] L. Guo, D. Xu, and Z. Qiang, "Background subtraction using local SVD binary pattern," in *Proc. Conf. Comput. Vis. Pattern Recogn. Workshops (CVPRW)*, Jun./Jul. 2016, pp. 1159–1167.

[24] J. Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2041–2056, Oct. 2015.

[25] W. He, Y. Kim, J. Wu, G. Zhang, Q. Qi, L. Guo, B. Tu, and F. Huang, "Local compact binary patterns for background subtraction in complex scenes," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1518–1523.

[26] L. Maddalena and A. Petrosino, "Background subtraction for moving object detection in RGBD data: A survey," *J. Imag.*, vol. 4, no. 5, p. 71, 2018.

[27] T. Bouwmans, C. Silva, C. Marghes, M. S. Zitouni, H. Bhaskar, and C. Frelicot, "On the role and the importance of features for background modeling and foreground detection," *Comput. Sci. Rev.*, vol. 28, pp. 26–91, May 2018.

[28] M. Braham and M. Van Droogenbroeck, "A generic feature selection method for background subtraction using global foreground models," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst. (ACIVS)*, 2015, pp. 717–728.

[29] C. Silva, T. Bouwmans, and C. Frélicot, "Online weighted one-class ensemble for feature selection in background/foreground separation," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2216–2221.

[30] C. Silva, T. Bouwmans, and C. Frélicot, "Superpixel-based online wagging one-class ensemble for feature selection in foreground/background separation," *Pattern Recognit. Lett.*, vol. 100, pp. 144–151, Dec. 2017.

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

[32] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2014, pp. 387–394.

[33] F. El Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of Gaussians model: Application to background modeling," in *Proc. ISVC*, Las Vegas, NV, USA, Dec. 2008, pp. 772–781.

[34] Z. Zhao, T. Bouwmans, X. Zhang, and Y. Fang, "A fuzzy background modeling approach for motion detection in dynamic backgrounds," in *Proc. Int. Conf. Multi. Signal Process.*, Shanghai, China, Dec. 2012, pp. 177–185.

[35] P. Chiranjeevi and S. Sengupta, "Detection of moving objects using multichannel kernel fuzzy Correlogram based background subtraction," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 870–881, Jun. 2014.

[36] D. K. Panda and S. Meher, "Detection of moving objects using fuzzy color difference histogram based background subtraction," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 45–49, Jan. 2016.

[37] S. Javed, T. Bouwmans, M. Sultana, and S. K. Jung, "Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA," in *Proc. Int. Conf. Image Anal. Process. (ICIAP)*, Catani, Italy, Sep. 2017, pp. 230–241.

[38] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, Aug. 2015.

[39] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, "Spatiotemporal low-rank modeling for complex scene background initialization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1315–1329, Jun. 2018.

[40] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. IEEE Int. Conf. Syst., Signals, Image Process. (IWSSIP)*, Bratislava, Slovakia, Jul. 2016, pp. 1–4.

[41] M. Sultana, A. Mahmood, S. Javed, and S. K. Jung, "Unsupervised deep context prediction for background estimation and foreground segmentation," *Mach. Vis. Appl.*, vol. 30, no. 3, pp. 375–395, Apr. 2019.

[42] M. Ullah, A. Mohammed, and F. A. Cheikh, "PedNet: A spatio-temporal deep convolutional neural network for pedestrian segmentation," *J. Imag.*, vol. 4, no. 9, p. 107, 2018.

[43] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *J. Neural Netw.*, vol. 117, pp. 8–66, Sep. 2019.

[44] S. Isik, K. Özkan, S. Günal, and Ö. N. Gerek, "SWCD: A sliding window and self-regulated learning-based background updating method for change detection in videos," *J. Elect. Image*, vol. 27, no. 2, 2018, Art. no. 023002.

[45] P. Rota, H. Ullah, N. Conci, N. Sebe, and F. G. B. De Natale, "Particles cross-influence for entity grouping," in *Proc. 21st Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2013, pp. 1–5.

[46] H. Ullah and N. Conci, "Structured learning for crowd motion segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 824–828.

[47] H. Ullah, M. Ullah, and M. Uzair, "A hybrid social influence model for pedestrian motion segmentation," *Neural Comput. Appl.*, pp. 1–17, 2018.

[48] Y. Zhao, D.-S. Huang, and W. Jia, "Completed local binary count for rotation invariant texture classification," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4492–4497, Oct. 2012.

[49] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing." 2019, *arXiv:1902.02918*. [Online]. Available: https://arxiv.org/abs/1902.02918

[50] J. Wang, Q. Li, S. Yang, W. Fan, P. Wonka, and J. Ye, "A highly scalable parallel algorithm for isotropic total variation models," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 235–243.

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

[52] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2014, pp. 509–515.

[53] G. Ramírez-Alonso and M. I. Chacón-Murguía, "Auto-adaptive parallel SOM architecture with a modular analysis for dynamic object segmentation in videos," *Neurocomputing*, vol. 175, pp. 990–1000, Jan. 2016.

[54] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?" in *Proc. Int Conf. Image Anal. Process. (ICIAP)*, 2017, pp. 96–107.

[55] E. López-Rubio, M. A. Molina-Cabello, R. M. Luque-Baena, and E. Domínguez, "Foreground detection by competitive learning for varying input distributions," *Int. J. Neural Syst.*, vol. 28, no. 5, pp. 1750056-1–1750056-16, 2018.

[56] A. Varghese and G. Sreelekha, "Sample-based integrated background subtraction and shadow detection," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 25, pp. 1–12, 2017.

[57] A. Sobral, "BGSLibrary: An OpenCV C++ background subtraction library," in *Proc. 9th Workshop Video Comput. (WVC)*, Rio de Janeiro, Brazil, Jun. 2013. [Online]. Available: https://github.com/andrewssobral/bgslibrary

[58] A. W. Senior, Y. Tian, and M. Lu, "Interactive motion analysis for video surveillance and long term scene monitoring," in *Proc. Comput. Vis.-ACCV Workshops*, Queenstown, New Zealand, 2010, pp. 164–174.