Foreground Segmentation Using Adaptive 3 Phase Background Model

Sujoy Madhab Roy, *Student Member, IEEE*, and Ashish Ghosh, *Senior Member, IEEE*

***Abstract*— An extensive collection of algorithms have been proposed over the years to identify the foreground from a video scene, but none of them considers the classification history of previous frames for discovering moving objects. All the existing algorithms focus on a single background model to cope with all types of challenging and complex video environments. In this paper, a real-time pixel level classification method is proposed that uses its previous output history to update its parameters. The model has three components designed to handle various types of challenging background environments. For each pixel, three sub-parts, namely the Adaptive Background Model, the Neighborhood Background Model, and the Change Detection Background Model are constructed to detect various types of complex background changes. A pixel level model updating method uses the previous foreground/background binary classification results to periodically refresh all the three background models. This update method helps to recognize the foreground accurately by adjusting the algorithmic parameters to efficiently detect complex background changes. This novel method shows a significant improvement in performance for a variety of complex video scenes.**

***Index Terms*— Foreground segmentation, change detection, mini-batch, binary classification, neighborhood information, real-time analysis.**

I. Introduction

VARIOUS computer vision applications like visual surveillance often require a video scene to be divided into two distinct segments, namely the foreground and background. The foregrounds or the moving objects in a video scene are of prime interest in surveillance problems. Background subtraction (BS) is a way to filter out the backgrounds from a video scene. A BS algorithm can typically be divided into three major steps: 1. Background model construction, 2. Foreground detection and 3. Updating background model. First a background model is created by using only the first frame or first few frames of the video. Taking this model as reference, in the second step, moving objects are detected by computing the similarity between the new frame and the model with an appropriate threshold. Background model updating is the third major step which makes the model more robust in handling complex background scene changes. Along with these three steps, two more steps are often used to increase the accuracy of a BS procedure. The first is a pre-processing step applied on the input frames of the video mainly to either suppress unwanted distortions or to enhance some image features important for further processing. The other is a post-processing step applied on the foreground detected binary images in order to reduce false detections thereby producing more accurate results.

Over the years, BS has remained a challenging computer vision problem. A BS algorithm that gives outstanding results in the presence of various complex scene changes, such as dynamic background, camouflaged objects, shadow effect, thermal effect, camera motion etc., is difficult to create. Although a few state-of-the-art BS methods exist that perform well over most video scenes, most of the other BS algorithms fail to produce good results for most video scenes. A common reason for failure is using an improper procedure of model update. A BS method without proper model updating is prone to produce less accurate results due to regular scene changes in a complex video environment. Proper identification of the changes in a video scene is one of the main objectives of an efficient BS method. An efficient background model must have different techniques or sub-models to handle different challenging background environments like dynamic background, camera jitter, camera motion, air turbulent motion, etc., but in reality this type of background model is very difficult to construct. A possible solution is to divide the background challenges into various groups and each group is handle by a separate sub-model, and this strategy is used in this article. Proper scene identification crucially depends on a proper background model update strategy. Currently existing BS methods update their background models by using only current pixel values and completely ignoring the history of previous classifications. In the proposed method we try to overcome this limitation by taking into account the entire history of past classifications for updating the background model.

In this article, we propose a new pixel level non-parametric BS algorithm where background model is divided into 3 parts, each having a different purpose. The first part, Adaptive Background Model (ABM), tracks the changes to each pixel independently by analyzing the binary classifier output of the previous frames of a video. The purpose of the ABM is to handle the dynamic background efficiently. The second part, Neighborhood Background Model (NBM), observes the neighborhood information of each pixel from the previous binary classification output. NBMs purpose is to resolve the

Manuscript received October 18, 2018; revised January 18, 2019 and March 20, 2019; accepted April 30, 2019. The Associate Editor for this paper was L. M. Bergasa. *(Corresponding author: Sujoy Madhab Roy.)*

The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: sujoyroy_r@isical.ac.in).

Digital Object Identifier 10.1109/TITS.2019.2915568

ghost effect. The third part, Change Detection Background Model (CDBM), continuously monitors current changes to the background and updates itself accordingly. CDBM can handle a variety of challenges like camera jitter, air turbulent motion, moving camera motion, etc. All three background models together can handle/detect many complex environments in a video scene. The novelty in the proposed algorithm is in considering the previous binary classification results instead of only the current pixel values for updating all three parts of the background model.

This article is organized as follows. Section 2 provides a brief discussion of the existing BS algorithms. Section 3 describes the details of the proposed method. Experimental results are explained in Section 4. Conclusions and future works are given in Section 5.

II. Review of Existing BS Methods

A significant amount of work has been done in the last few decades to solve the BS problem from different perspectives [1], [2]. In statistical point of view, a BS model can be divided into two parts: parametric model and non-parametric model. In machine learning point of view, BS problem can be divided into a classification problem or a clustering problem. Again at image space level, a BS method can be divided into three parts: pixel based, small region based and large region (i.e., frame) based method. In this article, we categorize the BS problem by considering the size, type and updating procedure of the background model.

A simple pixel based background model considers only one element for each pixel position. In [3], background model store only the median value of previous frames. For classification purpose, the median value is used to compare with the pixel value of a new frame. Wren *et al.* [4] associated Gaussian distribution about the mean intensity value for each point on the texture surface. In this method, after each iteration, mean is updated using a simple adaptive filter. In [5], an image area is divided into similar regions (similar intensity) by examining the difference between two consecutive frames with respect to a pre-defined threshold. Then mode is taken as background element. All these models perform well if a video scene has a single-mode background.

In multi-modal background environment, a single-valued background model produces erroneous results. To handle multi-modal behavior of the background (e.g., dynamic background), one approach is to consider as many elements in a model as the number of modes in the background. Friedman and Russell [6] learned a Mixture-of-Gaussian (MoG) model using an unsupervised technique, incremental Expectation Maximization (EM) algorithm. Each mixture component of each class is updated by observing the likelihood of the membership. Stauffer and Grimson also proposed a pixel based parametric MoG model, Gaussian mixture model (GMM) [7]. This model updates the statistical parameters only when a pixel is classified as background. As the maximum size of each model is set beforehand, a new background information can be placed into the model by replacing an element of the model. An improvement over the GMM is proposed by Hayman and Eklundh [8]. The authors mainly consider motion blur, sub-pixel camera motion and mixed pixels at object boundaries to reduce the classification errors of the GMM [7] method. A non-parametric model, Kernel density estimation (KDE) is another Gaussian family distribution, proposed by Elgammal *et al.* [9]. Li *et al.* [10] detect pedestrians in infrared images using GMM soft decomposition and KDE based foreground estimation. Kim *et al.* [11] proposed a color similarity measure to create a pixel level codebook model. Like [7], a new background information can take place of an existing model element.

Instead of creating background model for each pixel separately, another type of background model is constructed by extracting a few important features from the whole image frame. Zhou *et al.* [12] considered the low-rank model of the background. The foreground objects are detected as outlier in low-rank representation. A robust 2D-PCA model is proposed by Sun *et al.* [13] which is more robust to outliers and corrupted data. For BS, a GMM based superpixel approach is used by Chen *et al.* [14]. The authors initially generate a superpixel spanning tree by considering each pixel as a vertex and also consider a GMM for each pixel. They constructed a superpixel hierarchy and in each hierarchical steps the method calculates the motion vector. The GMM's for the similar pixels are merged to reduce the total size of the background model.

In recent years, researchers consider a fixed number of background elements/samples for each pixel separately to construct a non-parametric background model. Wang and Suter [15] proposed a sample consensus (SACON) method, in which the background model is defined by N most recently observed pixel values for each pixel separately, where N varies from 20 to 200. To update the model, background samples are replaced by first-in-first-out manner. A real-time BS method (ViBe) was proposed by Bernich and Van Droogenbroeck [16] where background model size (N) is considered as 20. This non-parametric model updates by a random scheme: a pixel is selected randomly from a background classified pixel's neighbor and updated with its pixel value. Droogenbroeck and Paquot [17] modified ViBe for better accuracy by sacrificing its processing speed. Hofmann *et al.* [18] proposed a background model, pixel based adaptive segmenter (PBAS), where recent history of pixel values are observed to construct the model of size 35. The model is updated like [16] by choosing a neighboring position randomly but only performed with a probability that depends on an adaptive state variable called learning rate. Zhong *et al.* [19] modified the model updating strategy of the PBAS by using both pixel and object level information. St-Charles *et al.* [20] proposed a change detection method (SuBSENSE) that uses a pixel-level feedback loop to update each of the parameters. The value of N is fixed with 50. In SuBSENSE, the model update rule is applicable for both cases when a pixel is classified as background or foreground.

Deep learning methods are also applied in recent years for BS. Yang *et al.* [21] constructed a deep background model using Convolutional Neural Network (CNN), where three atrous convolution branches are used to extract spatial information from different neighborhoods of pixels. Ke *et al.* [22]

detect road congestion using multidimensional visual features and a CNN. The method uses a GMM for background modeling and the CNN is then used to accurately detect the foregrounds. Chen *et al.* [23] extracted the pixel-wise semantic features using a convolutional encoder-decoder network for object detection. The authors also used a long short-term memory (LSTM) model to integrate pixel-wise changes.

In a BS method, researchers mainly considered a single background model to deal with various complex behavior of the background like dynamic background, moving background objects, air turbulent motion, moving camera motion etc., in a video scene. To the best of the authors knowledge, no BS algorithm in the literature till the present day has considered different background models to cope with different challenges of the background motions. Again no BS method considered the binary classification history of previous frames (i.e., the nature of the previous foregrounds and backgrounds to update the background model). In this article a BS method is proposed by considering the previously separated foreground and background classification and also considers three separate background models to handle various background challenges.

III. Proposed Method

A. Overview

Extracting the moving objects from a video scene having a static background is easier than a scene with a complex background. A simple background model with one or two samples per pixel is sufficient to produce good results for a video scene with a static background. However background motions can make a video scene more complex. Background motions can be of two kinds, periodic or aperiodic. An example of periodic background motion is dynamic background whereas moving background, camera jitter, camera motion, turbulent motion etc. are examples of aperiodic background motion. A single background model to cope with every situation is very difficult to construct. In early BS research, authors of [4], [24] considered a separate background model for each pixel with only one element per pixel. This single model element was used as a reference to classify future pixel values. Later [7], [25] also use a separate background model for each pixel but more than one element in the model, to handle complex background environment. In recent years [15], [16], [18], [20] consider a predefined fixed size model for every pixel position. All these methods can handle complex scenes but each of them has its own pros and cons.

In this article, a novel pixel level method is used to solve the BS problem by constructing three different background models for each pixel position. The proposed 3PBM classifier is composed of 3 different models, namely the ABM, the NBM and the CDBM, where each of them has a specific purpose. The ABM and NBM are mainly useful in handling complex environments such as dynamic background and ghost effect. CDBM is constructed in a way similar to [15], [16], [18], [20] in order to incorporate current changes of the background and is useful in handling camera jitter, camera motion and air turbulent motion.

A pipeline of the proposed 3PBM method is shown in Figure 1. The method can be divided into 4 parts:

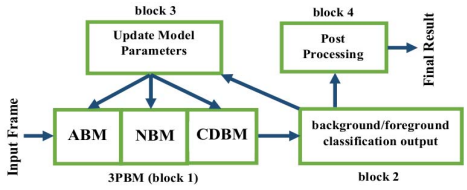

Fig. 1. Pipeline of the proposed 3PBM method.

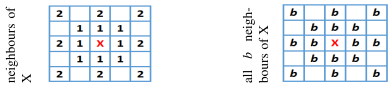

Fig. 2. Neighborhood of a pixel position X.

1. Background model: Three separate background models (ABM, NBM and CDBM) are combined to create 3PBM separately for each pixel position (see block 1). 2. Foreground detection: a pixel-level classifier classifies new pixel values into foreground or background (see block 2). 3. Updating background model: a periodic updating mechanism to independently update all three models of all the pixel positions as well as refresh the model parameters (see block 3). 4. For final results, a post-processing operation applied on classification outputs using basic morphological operations (see block 4).

All the three models, consist of lists of numeric values which are matched with the pixel values of each new frame (using certain matching criteria described below) and if a match is found then the pixel is declared as background, labeled as *b*. Only one match is good enough to declare a pixel as background. If the pixel does not match to any of the models, it is declared as foreground, labeled as *f*. ABM and CDBM models are initialized using only the first frame of the video and NBM is initially empty. All frames except the first frame (henceforth referred to as *test frames*) are divided into mini-batches of 50 frames each. The ABM model is updated after every mini-batch but the NBM and CDBM models and their model parameters are updated after every test frame.

For every mini-batch of classification results, four features (namely, total number of *f*'s, total number of *b*'s, total number of transitions between *f*'s and *b*'s and two-level neighborhood checking (see Figure 2) for *f* pixels) are computed from the classification history of each pixel position. These characteristics help in the adaptive update process of all three models.

All the existing state-of-the-art BS algorithms only consider current pixel values for updating the background model. None of these algorithms consider the classification results of the previous frames for updating the background model. Previous results often contain information about the nature of the video scene. This motivates us to create a background model that updates continually by using the previous classification results.

Algorithm 1 presents the pseudocode of the 3PBM method.

B. Adaptive Background Model (ABM)

In some video scenes the background pixel values change very frequently, because of periodic movements of the background, such as swaying trees branches and leaves, sparkling

Algorithm 1 3PBM Method

Input: A video sequence.

Initialize: ABM and CDBM are initialized using the first frame.

I: **for** each pixel x of a test frame t **do**

II: Find similarity between pixel value $P^t(x)$ with the elements of $ABM(x)$, $CDBM(x)$ and $NBM(x)$ using $R(x)$.

III: Unsuccessful search update $PABM(x)$.

IV: For a f decision (see Equ. 1), if its two-level neighbors are all b, track x and update $NBM(x)$ using ζ_1, ζ_2.

V: Update parameters $R(x), \upsilon(x), T(x)$ and $\bar{D}_m(x)$.

VI: **end for**

VII: From the binary classified results, update parameters *#Transition* and *AbsDiff*.

VIII: After every *Fwindow* number of frames, for all x, update $ABM(x)$ using $PABM(x)$, τ_1 and τ_2.

Output: Binary classified images.

of a water fountain or ripples on water etc. In such dynamic background environments, without proper modeling, a background region could get wrongly classified as foreground. In a static background environment only one mode (peak in the distribution) is enough to represent the pixel intensity distribution. On the other hand, a region of dynamic background will have multi-modal nature, i.e., it will contain multiple dominant peaks since multiple objects occupy a particular pixel position. In dynamic background, if only one mode is taken as a background, a background classifier will produce a rapidly alternating string of f's and b's. To identify dynamic background properly, a simple but effective way is to follow the nature of the previous output string of f's and b's. A long string of f's or b's indicates foreground or background respectively but a frequent transition between f's and b's indicates dynamic background. We compute three features from the previous classification output in order to recognize dynamic backgrounds. These features are the frequency counts of the previous f's and b's and total transition between f's and b's.

ABM is initialized using the first frame of the video. One ABM is created for each pixel position of the video frame. The first entry of an ABM is the corresponding pixel value of the first frame. Let the pixel value of the x-th pixel of the t-th frame be denoted by $P^t(x)$. Let the ABM of the x-th pixel be denoted as $ABM(x)$ and let its i-th entry be denoted as $ABM^i(x)$. Initially then $ABM(x) = \{ABM^1(x)\}$ and $ABM^1(x) = P^1(x)$. After initialization, ABM is updated after every mini-batch of $Fwindow$ $(= 50)$ consecutive frames. Between two successive updates, in order to append dynamic background information into the ABM, various criteria are computed based upon the previous classification outputs within that mini-batch.

To classify a pixel position x of t-th frame, its value $P^t(x)$ is matched with the elements of the $ABM(x)$ by considering the matching condition $Dist(ABM^i(x), P^t(x)) \leq R(x)$, where $ABM^i(x)$ is the i-th element of $ABM(x)$ and $R(x)$ is the adaptive distance threshold (discussed in Section III-F). If this relation is true for any i, then x is declared as background with label b and the algorithm proceeds to the next pixel position. If $P^t(x)$ does not match with $ABM(x)$ then the algorithm tests $P^t(x)$ with $NBM(x)$ as described in the next section. If the $NBM(x)$ classifies x as background then the pixel is declared background with label b. If the $NBM(x)$ also fails to identify the pixel as background then the algorithm tests $P^t(x)$ with $CDBM(x)$ and if $CDBM(x)$ fails to detect the pixel as background it is then declared as foreground with label f. Therefore pixel is declared foreground only if all three models fail to detect it as background.

To update the ABM, a temporary list of values called Probable ABM (PABM) is maintained. At the beginning of every mini-batch the PABM is initialized to empty. If a pixel value $P(x)$ does not match with $ABM(x)$ elements, then $P(x)$ is added to the $PABM(x)$. At the same time a frequency count buffer is associated with the corresponding $PABM(x)$ entry and initialized with value 1. When a pixel value $P^k(x)$ of test frame k fails to match with $ABM(x)$ elements, then $P^k(x)$ is matched with the $PABM(x)$ elements. A successful match in $PABM(x)$ increases the corresponding frequency count buffer by 1 and an unmatched $P^k(x)$ is appended to the existing $PABM(x)$ with frequency count value 1. After a mini-batch, for each pixel, first the PABM is trimmed by removing every element of PABM whose frequency count is less than 10% of $Fwindow$, then the PABM is conditionally merged with the corresponding ABM, depending upon certain criteria that is described next. To take the decision whether or not to append the elements of PABM into the pre-existing ABM, the classification history is used to compute two parameters. The first one, *#Transition*, is the total number of transitions between f's and b's, and the second one, *AbsDiff*, is the absolute difference between total number of f's and b's during a mini-batch. If a video scene contains dynamic background then *#Transition* will be high and *AbsDiff* will be low. Therefore the condition that we use for appending PABM into ABM is $\#Transition \geq \tau_1$ and $AbsDiff \leq \tau_2$ depending on two predefined threshold values τ_1 and τ_2. If either of these two conditions is not met then the PABM is not added to the ABM and is discarded.

C. Neighborhood Background Model (NBM)

In BS, one of the toughest problems is ghost effect. Ghost effect occurs when an object that had been static from the beginning of the scene starts to move away. The newly exposed background area over which the object had been stationary is now falsely detected as foreground. This area is of the same shape as the object that has moved away and is called the ghost of the object. This ghost of the object adversely affects the classification results and needs to be distinguished from true objects and eliminated. Barnich *et al.* proposed a background model updating technique (spatial diffusion) for reducing the ghost effect in ViBe [16]. In this technique, everytime a pixel x is classified as background, an 8 nearest neighbor of x (say x_{n8}) is randomly selected and a model sample of x_{n8} is replaced with the current pixel value $P(x)$. This technique gives the background a tendency to spread around neighboring regions. Thus newly exposed ghost regions gradually get filled

up by the neighboring background regions. Although this technique reduces the false positive outputs (ghost of the object), it generates false negative outputs (by transforming foreground into background) for long time static foregrounds and for slowly moving objects. This approach is used in many other BS methods including PBAS [18], ViBe+ [17], SuBSENSE [20] to remove the ghost of the object. But these methods are unable to completely solve the problem.

In order to tackle the problem of ghost effect without the aforementioned side-effect of misclassifying static foregrounds, we propose the NBM. In NBM, the identifying feature of the ghost effect is the sudden appearance of a foreground pixel in the midst of background pixel and it continuing to stay as a foreground pixel for a large number of frames. In the case of dynamic background also a foreground pixel may suddenly appear in the midst of background pixels; but it wouldn't stay as foreground for long but would keep switching frequently between f and b. This motivates us to track the future classification results of a pixel which is classified as f in the current frame and its neighbors (marked as 1's and 2's) are all b (see Figure 2) in the previous frame. We continue the tracking for a certain number of frames in order to see whether the classification output remains steady at f or not. After watching the classification history for those certain number of frames a decision is made. This technique differs from ViBe where the decision is taken immediately and the output is not tracked, as a result of which some true foreground objects turn into background.

The above mentioned technique of identifying the ghost effect is implemented in the following way. We first find all the pixels which are classified as f in the t-th frame such that its neighbors are all classified as b (see Figure 2) in the $(t-1)$-th frame. These pixels could be potential ghost effects and are therefore marked for future tracking. We then follow the future classification results of these marked pixels. According to the previously explained criterion, if a marked pixel continues to remain as foreground for a significant number of frames after being marked, it can then be considered as ghost effect. Therefore if a marked pixel x remains f in ζ_1 number of consecutive frames, then it probably occurred due to the ghost effect and so the $NBM(x)$ is updated with the pixel value of x. If on the other hand x is not classified as f in all ζ_1 consecutive frames, it could still be due to the ghost effect as it is possible that a few frames could get randomly misclassified. In such a case we continue to observe the future classification results up to ζ_2 consecutive frames ($\zeta_2 > \zeta_1$) and if these ζ_2 frames contain not less than ζ_1 number of f's then this probably occurred because of ghost effect and the pixel value of x is appended with $NBM(x)$.

At the beginning, an empty NBM is associated with each pixel position of a frame. An element is added to a NBM only when the ghost effect occurs in a video scene. Thus as the video progresses, it is possible to have some pixels with non-empty NBMs while the remaining have empty NBMs. To match a pixel value $P^t(x)$ at position x of frame t with an element of the corresponding $NBM(x)$, $Dist(NBM^i(x), P^t(x)) \leq R(x)$ is calculated, where $NBM^i(x)$ is the i-th element of the $NBM(x)$.

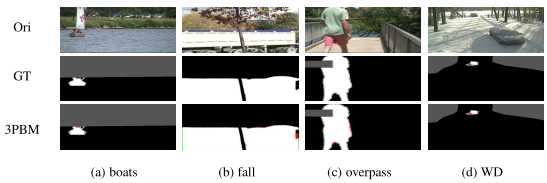

Fig. 3. Original input images (Ori), ground-truth images (GT), output generated using proposed 3PBM method on (a) *boats*, (b) *fall*, (c) *overpass* video scenes from dynamic background category and (d) *winter driveway* (WD) video scene having ghost effect.

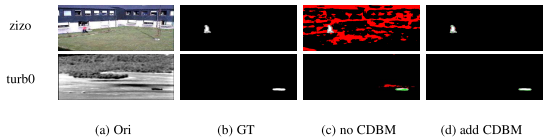

Fig. 4. (a) Original input images (Ori), (b) ground-truth images (GT), (c) output without CDBM (no CDBM) and (d) With CDBM (add CDBM) on *zoomInZoomOut* (zizo) and *turbulence*0 (turb0) video scenes.

The update procedure of $R(x)$ is discussed in Section III-F. If this relation is true for any i, then x is declared as background with label b, otherwise proceed to $CDBM(x)$ to classify x.

D. Change Detection Background Model (CDBM)

ABM and NBM can efficiently identify dynamic background and ghost effect in a video scene (see Figure 3) but complex background motions such as air turbulent motion, camera motion, etc., may degrade the accuracy. Column 3 of Figure 4 shows the classification errors that can occur in such challenging video environments. In this section we propose the CDBM in order to handle these weaknesses of ABM and NBM. A video scene with turbulent motion, camera motion, etc. has neither a periodic motion (like dynamic background) nor some specific nature which can be easily anticipated. In such complex environments the background keeps changing with time, because of which a background model must be regularly updated (preferably in every iteration) by continuously incorporating the current background into the background model. Because of regular updates to the background model, frequent inclusion of new elements with every change of the background may greatly increase the model size, if its size is not capped beforehand. In the SACON [15], ViBe [16], PBAS [18] and SuBSENSE [20] methods, the authors use a fixed length (N) background model for each pixel of the frame. In these methods, if a pixel is classified as background, the model updates itself by replacing an existing model element with the current pixel value. In SACON, the authors show that a value of $N \in [20, 200]$ is good enough to adequately model the background behavior. In ViBe, PBAS and SuBSENSE, N is taken as 20, 35 and 50, respectively. In all of the above methods, the total amount of memory required to store all the model elements is $N \times$ (No. of channels) $\times$ (Frame size), which can become very large, especially in case of high resolution videos. CDBM uses a fixed length background model, similar to the above methods, but which requires far lesser amount of memory and also gives better performance for video sequences containing turbulent motion or camera

motion. The memory requirement for CDBM is reduced by taking $N = 8$. The performance loss introduced by this reduced model size is compensated by introducing a novel modification to the decision threshold and learning rate update functions that are used by SuBSENSE algorithm. This novel alteration of the update functions, when used in conjunction with ABM and NBM, gives us either better or similar performance to the above mentioned algorithms at a fraction of the memory cost. These new modifications to the SuBSENSE update functions are described in detail in Section III-F. The main purpose of CDBM is to detect the non-periodic background movements. The inclusion of CDBM along with ABM and NBM improves the accuracy in the presence of turbulent motion, camera motion, etc. (see column 4 of Figure 4).

Like ABM, CDBM is also initialized for every pixel position by only using data from the first frame. For the purpose of initialization, for pixel position x, the 8 members of the $CDBM(x)$ are selected from the 8 nearest neighbors of x, where $CDBM(x) = \{CDBM^1(x), \ldots, CDBM^8(x)\}$. To classify a pixel value $P^t(x)$ of test frame t, a relation $Dist(CDBM^i(x), P^t(x)) \leq R(x)$ is calculated where $CDBM^i(x)$ is the i-th element of $CDBM(x)$, $i \in \{1, 2, \ldots, 8\}$. If this relation is true for any i, then x is declared as b, otherwise it is declared as f. For each test frame t, if a pixel value $P^t(x)$ is classified as b, an element $CDBM^i(x)$ is randomly picked with equal probability from $CDBM(x)$ and replaced with the current pixel value $P^t(x)$ with probability $1/T(x)$, where $T(x)$ is the *learning rate* of $CDBM(x)$. The update procedures of both $R(x)$ and $T(x)$ are discussed in Section III-F. This stochastic update procedure of the CDBM is similar to the update mechanism used in ViBe and SuBSENSE except that CDBM does not simultaneously update the neighboring pixel. A blind model update strategy for the neighborhood of a background pixel used in ViBe does not consider the foreground and background classification output of the neighbors, which erroneously turns slow moving objects into background. Again the neighborhood update policy used in SuBSENSE also misclassify the slowly moving objects. On the other hand, as the CDBM update mechanism applied on the background pixel only, it does not include static objects information into background model. Thus CDBM update strategy gives a major advantage over the model update mechanism used by the methods ViBe, SuBSENSE.

E. Final Foreground-Background Classification

The main objective of a BS is to classify every pixel of every frame into foreground or background. A test pixel is considered as a b if any one of the models ABM, NBM or CDBM classify the pixel as background. Only one match is good enough to declare a pixel as b. Otherwise it is declared as f. Thus

$$x \in \begin{cases} \mathbf{b}, & \text{if a match in } ABM(x) \text{ or } NBM(x) \text{ or } CDBM(x) \\ \mathbf{f} & \text{otherwise.} \end{cases} \quad (1)$$

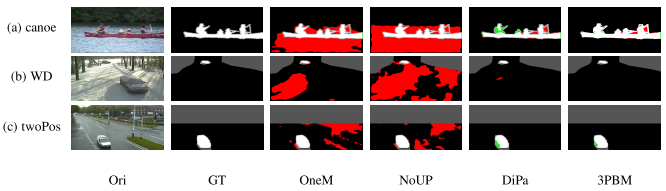

Fig. 5. Original input images (Ori), ground-truth images (GT), output generated by selecting only one sub-model (either of ABM, NBM or CDBM) in column 3 (OneM), without updating the 3PBM in column 4 (NoUP), updating the 3PBM with different parameters in column 5 (DiPa) and the proposed 3PBM method in column 6, on (a) *canoe*, (b) *winter driveway* (WD) and (c) *twoPositionPTZCam* (twoPos) video scenes from CDnet2014 dataset.

As our pixel based BS method takes the classification decision independently for each pixel, it contains some misclassified errors in the binary output. A post-processing operation often improves the performance of a BS method. In this article, to enhanced the accuracy, we use post-processing including median blurring and basic morphological operations.

In Figure 5, we show few results of the "incomplete" 3PBM (either using only one of the sub-models at a time or using all 3 sub-models without updating them). Column 3 shows large number of false positives when we select only one sub-model at a time, and column 4 shows plenty of errors when the 3PBM model is not updated periodically. In column 6, we show the results when 3PBM method is applied.

F. Per Pixel Updates

It happens so often that a video scene starts with a static background but changes rapidly due to dynamic behavior of the background. Without proper identification of the background behavior degrades the accuracy of the BS method. The 4-th column of the Figure 5 shows a large number of classification errors when the background model is not updated. Thus, the background model must be updated adaptively (include new background features into the model) to cope with complex background motions.

In ViBe, pre-defined fixed values are used for the model parameters R (radius of the sphere which is used to collect background samples) and T (model learning rate). In complex video scenes, a fixed global parameter may not be an ideal choice as the behavior of the background changes very frequently. In PBAS and SuBSENSE, parameters R and T update adaptively by following the dynamic changes in a video scene. In this article, we use similar parameters to periodically update the background model.

To test a pixel x, decision threshold $R(x)$ is used to measure the similarity between a background model element and the value of the pixel being tested. $T(x)$ is used as a learning parameter that helps to include new background information into the $CDBM(x)$ by replacing an older element. Like SuBSENSE, we use the parameter $\upsilon(x)$ to control the changes of both the parameters $R(x)$ and $T(x)$. To update these parameters, distance $\bar{D}_m(x)$ is considered, where

$$\bar{D}_m^t(x) = \bar{D}_m^{t-1}(x)(1 - \varphi) + d_m^t(x) * \varphi,$$

t is the current frame, $\varphi \in [0, 1]$ is a fixed value and $d_t(x)$ is the normalized minimum city-block distance between the

test pixel value $P^t(x)$ and 3PBM model element. Distance $\bar{D}_m$ is an indicator that is useful to separate static region from dynamic region. The value of $\bar{D}_m$ is close to zero in the static area but higher in the dynamic region. Again $\bar{D}_m$ is higher in the region where foreground is static for a long time.

A decision threshold (R) is required to classify a pixel into foreground or background. This threshold is often used upon the L1 or L2 distance of a test pixel value with the elements present in the background model. In static region, the current pixel value is very close to the model elements, so R must be small. On the other hand, in dynamic region, R must be large enough to properly classify the changes in the background. In this article, for pixel x, $R(x)$ updates recursively using function

$$R(x) = \begin{cases} R(x) + v(x) \times (5 - 4 \times \bar{D}_m(x)) \\ \qquad\qquad \text{when } R(x) < (1 + k \times \bar{D}_m(x)), \\ R(x) - 1/v(x) \quad \text{otherwise.} \end{cases} \tag{2}$$

where $k = 14$.

To increment R, we consider a function which depends on the distance $\bar{D}_m$. As stated earlier $\bar{D}_m$ is very small in static region, but it suddenly jumps to a higher value in the presence of complex background movements like dynamic background, turbulent motion, camera motion, etc. Again $\bar{D}_m$ is very high in the presence of a long time static object. To handle all such cases, we use different rate of increment of the R value by introducing a $\bar{D}_m$ dependent term $(5-4\times\bar{D}_m)$ (see Equation 2. In the static region, as $\bar{D}_m$ is small R is also small (by following the condition $R(x) < (1+k\times\bar{D}_m(x))$). To quickly adapt complex background changes as early as possible, R is increment with a higher rate because of the factor $5-4\times\bar{D}_m$. This value is reduced when $\bar{D}_m$ is high which again perfectly classify the static or slow moving objects.

In PBAS, learning rate (T) is updated in each iteration in the range $[2, 200]$ by using only the distance value. In SuBSENSE, T is updated periodically in the range $[2, 256]$ using the parameter v and the distance value. In this article, we consider parameter T similar to SuBSENSE but its range is bounded in the interval $[2, 25]$. A small T rapidly replace the older model elements with the current background information and a large T slowly updates the model. As the CDBM model size is fixed with only 8 elements (very much less than 50 used in SuBSENSE), to update the model more often (i.e., with high probability) by including current background changes, we set the maximum value of T to 25.

IV. Experimental Results

In this section, we perform an objective evaluation, to show the robustness of the proposed 3PBM method (object code: see the website www.isical.ac.in/~sujoyroy_r/bsPBM). Initially we present the purpose of the parameters used in the 3PBM method. Change detection (CDnet 2014) [26] and BMC [27] datasets, and a few standard evaluation metrics are used to evaluate the BS methods. For grading the performance of different methods, a subjective evaluation on several state-of-the-art methods are also presented in this section. Finally, we measure the average execution time of the 3PBM method.

A. Parameter Settings

Any parameter must be selected in a proper way such that performance is better frequently. In BS, a parameter with a fixed value may not give good results for every challenging videos. Although we present a fixed initial value for the parameters in all the video scenes, tuning a few parameters by applying the domain knowledge may give better results on some specific scenario. In this article, we consider a fixed set of parameter values which are experimentally tuned.

To update the ABM, previous classification outputs of a mini-batch of *Fwindow* number of consecutive frames are analyzed to calculate the number of transitions (#*Transition*) between f's and b's and the absolute difference (*AbsDiff*) between total number of f's and b's. #Transition and Abs-Diff values are compared with the parameters τ_1 and τ_2, respectively, for assigning new background information into the ABM. In this article we consider $Fwindow = 50$, $\tau_1 = 24$ and $\tau_2 = 12$ in all our experiments.

To identify the ghost effect and update the NBM, we mark every pixel which appears as f suddenly in the midst of b pixels. The marked pixels are tracked for the subsequent ζ_1 or ζ_2 ($\zeta_2 > \zeta_1$) number of frames to identify moving background or ghost of the object. In all experiments, we consider $\zeta_1 = 15$ and $\zeta_2 = 25$.

Local distance parameter R is used to match a test pixel value with the list of elements in ABM, PABM, NBM and CDBM. To match a pixel with the model elements, resemblance of color distortion used by Zeng *et al.* [28] and L1 distance (*Dist(.,.)*, used in Sections III-B, III-C and III-D) is compared with R. Model learning rate T is used to update the CDBM only. In all videos R and T are initialized with 1 and 2, respectively, for every pixel. To update CDBM more frequently with the current changes of the background, the maximum value of T used in this article is 25. A different set of parameter values may degrade the performance of the proposed 3PBM method. In column 5 of the Figure 5 we show some results where different parameter values increase the errors than the 3PBM results (see column 6).

B. Datasets

An efficient BS method can cope with a variety of complex and challenging background environments. Thus to show the robustness of a BS method, a variety of challenging video sequences are required. CDnet 2014 (CDnet) dataset provides one such benchmark dataset which contains an extensive collection of real world indoor and outdoor video sequences. CDnet has a total of 53 video sequences divided into 11 categories. The whole dataset contains nearly $160,000$ frames where most of the video frames are accompanied with ground-truth frames. Frame size of a video in the CDnet dataset varies from 320×240 to 720×576 pixels.

As the CDnet dataset contains only real videos, we select the BMC dataset which contains 20 synthetic videos along with 9 real videos having various background challenges.

TABLE I

RE, SP, FPR, FNR, PWC, PR, F1, MCC AND ACC VALUES OF THE PROPOSED 3PBM METHOD ON CDNET 2014 DATASET

Category	Re	Sp	FPR	FNR	PWC	Pr	F1	MCC	ACC
baseline	0.874	0.997	0.003	0.126	0.841	0.896	0.882	0.879	0.992
camera jitter	0.765	0.983	0.017	0.235	2.378	0.772	0.727	0.737	0.976
dynamic background	0.924	0.999	0.001	0.076	0.183	0.877	0.899	0.899	0.998
interm. object motion	0.663	0.990	0.010	0.337	2.866	0.797	0.686	0.693	0.971
shadow	0.900	0.991	0.009	0.100	1.358	0.838	0.865	0.860	0.986
thermal	0.914	0.988	0.012	0.086	1.562	0.793	0.841	0.839	0.984
bad weather	0.754	0.999	0.001	0.246	0.554	0.922	0.829	0.831	0.994
low framerate	0.751	0.977	0.023	0.249	3.510	0.587	0.535	0.552	0.965
night videos	0.716	0.955	0.045	0.284	5.139	0.360	0.421	0.454	0.949
pan-tilt-zoom	0.635	0.990	0.010	0.365	1.289	0.460	0.501	0.517	0.987
turbulence	0.725	1.000	0.000	0.275	0.211	0.895	0.793	0.800	0.998

The synthetic video sequences are divided into 2 sets, namely *learning* and *evaluation*. Two outdoor scenes ("a street" and "a rotary") with various climate types like cloudy, sunny, foggy and windy conditions constitute the synthetic videos. BMC also contains 9 real videos with casted shadows, fast light changes, etc. Frame size varies from 320 × 240 to 640 × 480 pixels. This dataset also provides ground-truth images for evaluation.

C. Evaluation Metrics and Colour Codes

To examine the effectiveness of a BS algorithm, we consider nine metrics: *Recall* (Re), *Specificity* (Sp), *False Positive Rate* (FPR), *False Negative Rate* (FNR), *Percentage of Wrong Classifications* (PWC), *Precision* (Pr), *F-Measure* (F1) (metrics are defined in [26]), Matthew's Correlation Coefficient (MCC) = $\frac{(TP\cdot TN)-(FP\cdot FN)}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}}$ and accuracy ACC = $\frac{TP+FP}{(TP+FP+TN+FN)}$, where TP = true positives, FP = false positives, TN = true negatives and FN = false negatives. In this article, different color codes are used to visualize all binary classification results. The color codes are white, red, green and black to represent true positives, false positives, false negatives and true negatives, respectively.

D. Existing Techniques Used for Comparison

To test the efficiency of the 3PBM method, we used several BS methods: (i) GMM [7], (ii) KDE [9], (iii) Codebook (CB) [11], (iv) PBAS [18], (v) LOBSTER [29], (vi) ViBe [16], (vii) Spectral-360 [30], $(viii)$ AMBER+ [31], (ix) ViBe+ [17], (x) Adaptive model (Zhong17) [19], (xi) Dirichlet [32], (xii) SuBSENSE [20], $(xiii)$ CP3-online (CP3) [33], (xiv) AAPSA [34], (xv) BMOG [35] and (xvi) CL-VID [36].

E. CDnet 2014 and BMC Results

To assess the performance, we apply the 3PBM method over all the video sequences of the CDnet dataset. The dataset contains 11 categories (*bad weather, baseline, camera jitter, dynamic background, intermittent object motion, low framerate, night videos, pan-tilt-zoom, shadow, thermal and turbulence*) of videos ideal to evaluate the performance of different BS methods.

In Table I, we show the performance of the proposed method 3PBM in all different categories of the CDnet dataset. In all video scene, Precision and Recall values are high except *night video* and *pan-tilt-zoom* categories. In *night video*, FP and FN increases due to wrong classifications of low-visibility of vehicles with their very strong headlights in a night scene. Sp and ACC is very high in all the results which proves the robustness of the 3PBM method. FM and MCC measures are higher in *baseline, camera jitter, dynamic background, shadow, and thermal* video sequences comparative to the other videos. The proposed method shows excellent results on *dynamic background and thermal* categories in all of the metrics.

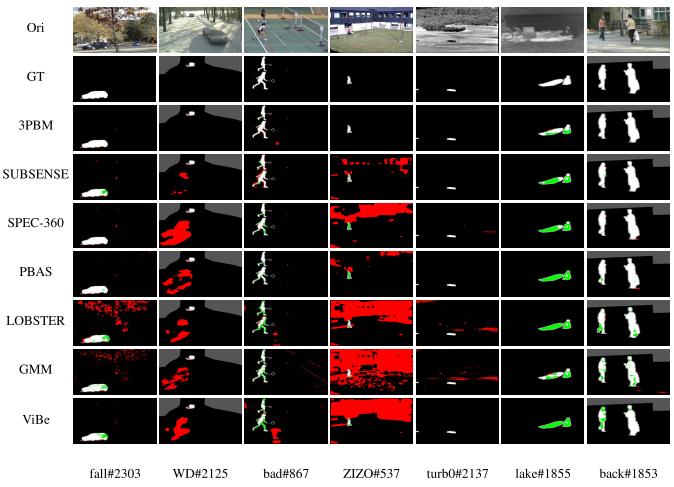

Fig. 6. Foreground detection on 7 video clips taken from CDnet dataset. Each row depicts figures of original input images (Ori), ground-truth images (GT), results corresponding to the proposed method 3PBM, SuBSENSE, SPECTRAL-360 (SPEC-360), PBAS, LOBSTER, GMM and ViBe. Columns represent 2303th frame of *fall* (fall#2303), 2125th frame of *winter driveway* (WD#2125), 867th frame of *badminton* (bad#867), 537th frame of *zoomIn-ZoomOut* (ZIZO#537), 2137th frame of *turbulence0* (turb0#2137), 1855th frame of *lakeside* (lake#1855) and 1853th frame of *backdoor* (back#1853) from left to right.

In Figure 6, we present a quantitative evaluation of the 3PBM method over a subset of videos of CDnet dataset. The videos are "fall" from *dynamic background*, "winter driveway" from *intermittent object motion*, "badminton" from *camera jitter*, "zoom-in-zoom-out" from *PTZ*, "turbulence0" from *turbulence*, "lakeside" from *thermal* and "backdoor" from *shadow* category. In fall, badminton and turbulence0 video scene 3PBM, SuBSENSE, SPECTRAL-360 and PBAS show better results than LOBSTER, GMM and ViBe where the other BS methods produce many false positives. In winter driveway and zoom-in-zoom-out videos, all the methods except 3PBM does not model the background efficiently due to moving background and camera motion and as a results produce plenty of false positives. In backdoor video, almost all the methods give good accuracy except ViBe and LOBSTER. In lakeside video all the methods misclassify some object region (false negative results are high) but 3PBM shows the best accuracy.

In Table II, we show the performance of all the BS methods. In *dynamic background, thermal, pan-tilt-zoom and turbulence* categories, 3PBM produces the best F1 scores among all the other methods. In *bad weather* sequence 3PBM and SuBSENSE show significant performance over all the other methods. In *baseline* category, although 3PBM obtains high

TABLE II

F1 SCORES OF DIFFERENT METHODS ON CDNET 2014 DATASET. IN EACH COLUMN, RED, GREEN AND BLUE COLOUR REPRESENT THE BEST, THE SECOND BEST AND THE THIRD BEST MEASUREMENTS

Method	F1 score for various categories of CDnet 2014 dataset										
	baseline	*camera jitter*	*dyn. background*	*int. obj. motion*	*shadow*	*thermal*	*bad weather*	*low framerate*	*night videos*	*pan-tilt-zoom*	*turbulence*
3PBM	0.882	0.727	0.899	0.686	0.865	0.841	0.828	0.535	0.421	0.501	0.793
SuBSENSE [20]	0.950	0.815	0.818	0.657	0.899	0.817	0.862	0.645	0.560	0.348	0.779
Spectral-360 [30]	0.933	0.716	0.787	0.566	0.884	0.776	0.757	0.644	0.483	0.365	0.543
PBAS [18]	0.924	0.722	0.683	0.575	0.860	0.756	0.780	0.540	0.373	0.121	0.706
ViBe+ [17]	0.871	0.715	0.720	0.509	0.815	0.665	0.703	0.477	0.362	0.060	0.783
KDE [9]	0.909	0.572	0.596	0.409	0.803	0.742	0.757	0.548	0.436	0.037	0.448
ViBe [16]	0.870	0.600	0.565	0.507	0.803	0.665	0.609	0.352	0.357	0.061	0.779
GMM [7]	0.825	0.597	0.633	0.520	0.737	0.662	0.738	0.537	0.410	0.152	0.466
AMBER+ [31]	0.881	0.711	0.843	0.721	0.813	0.760	0.767	0.469	0.380	0.135	0.755
CB [11]	0.654	0.635	0.616	0.490	0.533	0.623	0.370	0.415	0.266	0.042	0.362
LOBSTER [29]	0.924	0.724	0.568	0.577	0.837	0.825	0.572	0.195	0.287	0.054	0.363
Dirichlet [32]	0.929	0.748	0.814	0.542	0.813	0.813	0.729	0.325	0.389	0.133	0.717
Zhong17 [19]	0.874	0.494	0.301	0.824	0.650	0.727	0.411	0.642	0.618	0.411	0.625
CP3 [33]	0.885	0.521	0.611	0.618	0.704	0.792	0.777	0.555	0.348	0.279	0.472
AAPSA [34]	0.918	0.721	0.671	0.510	0.795	0.703	0.789	0.513	0.418	0.355	0.558
BMOG [35]	0.830	0.749	0.793	0.529	0.841	0.635	0.810	0.589	0.463	0.244	0.782
CL-VID [36]	0.937	0.521	0.552	0.518	0.841	0.719	0.722	0.567	0.436	0.038	0.480

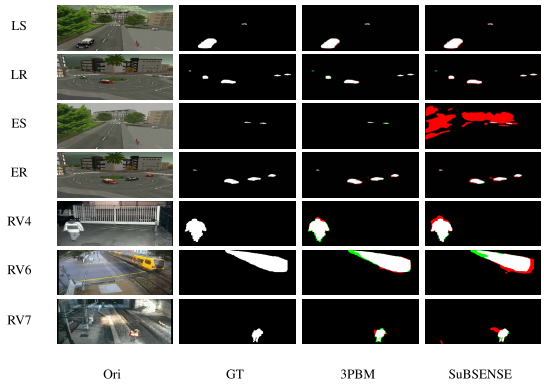

Fig. 7. Foreground detection results of BMC dataset using different methods. Each column depicts figures of original input images (Ori), ground-truth images (GT), results corresponding to the proposed method 3PBM and SuBSENSE method. Rows represent 475th frame of video 211 (LS), 1204th frame of video 521 (LR), 1050th frame of video 412 (ES), 1140th frame of video 122 (ER), 212th frame of real video 4 (RV4), 394th frame of real video 6 (RV6) and 1458th frame of real video 7 (RV7).

F1 measure, a few of the state-of-the-art methods show better results. Overall we can found that the 3PBM method shows comparative performance with the state-of-the-art methods in all video categories.

Figure 7 shows the quantitative results on BMC dataset. The selected sequences are 211 and 412 from "a street", 521 and 122 from "a rotary" and real videos 4, 6 and 7. In the figure, 3PBM shows better scores than SuBSENSE in majority of the videos. Table III shows qualitative results of various methods in different categories of the BMC dataset. Although SuBSENSE shows slightly better results over the 3PBM on *evaluation* category, in *learning* and real video categories 3PBM gives high F1 score.

F. Processing Speed

All the algorithms are written in C++ using the OpenCV library on a computer with a 3.3 GHz Intel i5 CPU with 8 GB RAM, running on Linux operating system. Table IV shows the processing speed of 3PBM on various image size of CDnet and BMC datasets which is reasonable to satisfy for real-time BS applications.

TABLE III

F1 SCORES OF DIFFERENT METHODS APPLIED ON THE BMC DATASET. BMC CONTAINS SYNTHETIC VIDEOS FOR *Learning* (LS AND LR) AND *Evaluation* (ES AND ER), AND ALSO REAL VIDEOS (RV). IN EACH COLUMN, RED, GREEN AND BLUE COLOUR REPRESENT THE BEST, THE SECOND BEST AND THE THIRD BEST MEASUREMENTS

Method	F1 score for various categories of BMC dataset				
	LS	LR	ES	ER	RV
3PBM	0.802	0.795	0.852	0.738	0.742
SuBSENSE [20]	0.733	0.771	0.866	0.774	0.736
PBAS [18]	0.787	0.708	0.770	0.716	0.664
ViBe [16]	0.725	0.692	0.767	0.679	0.682
BMOG [35]	0.558	0.570	0.742	0.758	0.573

TABLE IV

AVERAGE FRAMES PER SECOND (FPS) OF THE 3PBM METHOD

Frame size	320×240	640×480	720×576
FPS of the 3PBM	51.72	24.58	13.07

V. CONCLUSION AND FUTURE WORKS

A BS method with high accuracy and real-time processing speed is essential in many video processing applications. In this article, we propose a real-time BS method that can handle a variety of challenging indoor and outdoor, real and synthetic video sequences using three different background models. To update the background models by analyzing the nature of the background dynamics, the proposed method focuses on the previous binary classification results. Note that the previous results are simple but effective feature to properly characterize the nature of the background. The proposed method produces significant results in most of the video sequences and the method can adapt to many real-time applications because of

the lower computational time. The method gives less accurate results for frequent halos and reflections on the street due to very strong headlights of vehicles in night video scene. To overcome such complex background environments, in future, a few more features will be considered to update the model efficiently. As the 3PBM method does not explicitly handle the complex environments like shadow effect, camouflaged objects, traffic in night videos, etc., 3PBM is not stable in such events. In future more object detection databases will be used to validate the 3PBM method.

ACKNOWLEDGEMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor and the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article. Sujoy Madhab Roy thanks Mr. Ashish Bakshi for his contributions to improve this article.

REFERENCES

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

[2] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand*, vol. 122, pp. 4–21, May 2014.

[3] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Brit. Mach. Vis. Appl.*, vol. 8, no. 3, pp. 187–193, 1995.

[4] 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.

[5] J. Zheng, Y. Wang, N. L. Nihan, and M. E. Hallenbeck, "Extracting roadway background image: Mode-based approach," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1944, no. 1, pp. 82–88, Jan. 2006.

[6] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proc. 13th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 1997, pp. 175–181.

[7] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 1999, pp. 246–252.

[8] E. Hayman and J. O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1, Oct. 2003, pp. 67–74.

[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. 6th Eur. Conf. Comput. Vis. (ECCV)*, Jun. 2000, pp. 751–767.

[10] L. Li, F. Zhou, and X. Bai, "Infrared pedestrian segmentation through background likelihood and object-biased saliency," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2826–2844, Sep. 2018.

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

[12] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[13] Y. Sun, X. Tao, Y. Li, and J. Lu, "Robust 2D principal component analysis: A structured sparsity regularized approach," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2515–2526, Aug. 2015.

[14] M. Chen, X. Wei, Q. Yang, Q. Li, G. Wang, and M.-H. Yang, "Spatiotemporal GMM for background subtraction with superpixel hierarchy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1518–1525, Jun. 2018.

[15] 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.

[16] 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.

[17] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 32–37.

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

[19] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, and Y. Xu, "An adaptive background modeling method for foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1109–1121, May 2017.

[20] 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.

[21] L. Yang, J. Li, Y. Luo, Y. Zhao, H. Cheng, and J. Li, "Deep background modeling using fully convolutional network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 254–262, Jan. 2018.

[22] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2157–2170, Jun. 2019.

[23] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixel-wise deep sequence learning for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[24] B. P. L. Lo and S. A. Velastin, "Automatic congestion detection system for underground platforms," in *Proc. Int. Symp. Intell. Multimedia, Video Speech Process. (ISPACS)*, May 2001, pp. 158–161.

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

[26] 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 Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2014, pp. 393–400.

[27] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre, "A benchmark dataset for outdoor foreground/background extraction," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 291–300.

[28] Z. Zeng, J. Jia, Z. Zhu, and D. Yu, "Adaptive maintenance scheme for codebook-based dynamic background subtraction," *Comput. Vis. Image Understand.*, vol. 152, pp. 58–66, Nov. 2016.

[29] 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.

[30] M. Sedky, M. Moniri, and C. C. Chibelushi, "Spectral-360: A physics-based technique for change detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 405–408.

[31] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 401–404.

[32] T. S. F. Haines and T. Xiang, "Background subtraction with Dirichlet process mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 670–683, Apr. 2014.

[33] D. Liang, S. Kaneko, M. Hashimoto, K. Iwata, and X. Zhao, "Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes," *Pattern Recognit.*, vol. 48, pp. 1374–1390, Apr. 2015.

[34] 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.

[35] I. Martins, P. Carvalho, L. Corte-Real, and J. L. Alba-Castro, "BMOG: Boosted Gaussian mixture model with controlled complexity," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.* Cham, Switzerland: Springer, 2017, pp. 50–57.

[36] 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, Jun. 2018, Art. no. 1750056.