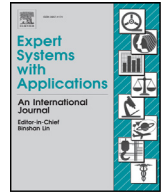




Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Real-time record sensitive background classifier (RSBC)

Sujoy Madhab Roy*, Ashish Ghosh

Machine Intelligence Unit, Indian Statistical Unit, Kolkata 700108, India



ARTICLE INFO

Article history:

Received 15 September 2017

Revised 20 September 2018

Accepted 14 October 2018

Available online 23 October 2018

Keywords:

Background subtraction

Computer vision

Mini-batch

Binary classification

Switch information

Neighbourhood search

Real-time analysis

ABSTRACT

In computer vision, one of the ways to identify moving objects or foregrounds from a complex video scene is background subtraction. Two problems that usually arise in background subtraction are that of *dynamic background* and *ghost effect*. In this article, an efficient, adaptive, real-time, novel background subtraction method is proposed to handle these two problems by utilizing the classification history of previous frames to detect changes in a video scene. The proposed method, coined as Record Sensitive Background Classifier (RSBC) is an amalgamation of two models, the Adaptive Background Model (ABM) and the Neighbourhood Background Model (NBM), which make use of the foreground and background classification information of the previous frames. The novel design of the ABM considers the temporal history of transitions between foreground and background of a given pixel, while the novelty in NBM lies in considering the binary classification results of the local neighbourhood in the preceding frame. Several state-of-the-art methods are compared with the proposed RSBC method, over which the proposed method shows significant improvement. Because of its good performance at fast processing speeds the proposed method can also be used as a preliminary prioritization step for slower but high accuracy methods in an object tracking system.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Background and foreground are complementary parts of an image. Background is the static or non-informative part whereas foreground carries the prime 'informative' part of a video scene. Background subtraction (BS) is the separation of foreground from the background of a video scene. Many computer vision applications use BS as an initial step to identify the foreground objects in a real-time video stream. BS algorithms work by representing the background with an appropriate model. The set of pixels agreeing with the background model are labeled as background and the set of pixels that are significantly different from the background model are labeled as foreground. The first step in constructing a background model is to initialize the model with some existing information, e.g. using first frame or the first few initial frames of a video. Background maintenance is the next step that is used to keep the background model robust and adaptive over time, thus helping to detect the backgrounds accurately. The problems which make BS challenging are dynamic background, moving background objects, long time static foreground etc. An efficient BS algorithm must be updated adaptively to make it robust against the above mentioned complex environments.

Dynamic background and ghost effect are two major concerns for a BS algorithm. Improper identification of these problematic situations may degrade the overall performance of the method. A region in a video scene is said to contain dynamic background when pixel values of that region changes periodically, such as a tree swaying in the wind. In the ghost effect, a background object suddenly begins to move, as a result of which the newly exposed area behind that moving object gets wrongly classified as foreground, which then stays in that place for a long time. The main purpose of the proposed algorithm is to simultaneously solve dynamic background, long time static foreground and ghost effect by constructing two background models that tackle all these cases separately. To achieve this we consider a new type of feature, which until now has been ignored by previous algorithms, namely the binary classification history of the previous frames, which helps in the update process of the background models. By studying the nature of the previous classification results, we focus on correctly identifying dynamic background and ghost effect in a video scene.

In the past decades, many algorithms have been proposed to solve the BS problem from different angles. In the statistical viewpoint, a background is modeled considering a single Gaussian by Wren, Azarbajani, Darrell, and Pentland (1997) or using a Gaussian Mixture Model (GMM) by Stauffer and Grimson (1999). Zhao, Bouwmans, Zhang, and Fang (2012) combined Type - 2 fuzzy GMM method with Markov Random Field (MRF).

* Corresponding author.

E-mail addresses: sujroyroy_r@isical.ac.in (S.M. Roy), ash@isical.ac.in (A. Ghosh).

Subspace learning (Zhou, Yang, Zhao, & Yu, 2014) is another procedure that considers the spatial information for BS. Barnich and Van Droogenbroeck (2011) in their ViBe algorithm solved the BS problem using a probabilistic method. Non-parametric consensus strategy (SubSENSE) is used by St-Charles, Bilodeau, and Bergevin (2015) by considering the similarity of observed samples. Braham and Van Droogenbroeck (2016) introduced a BS method using convolutional neural network (CNN). All these methods consider the pixel intensity values as a feature for background modeling.

In this article, a new BS algorithm is proposed that uses the background-foreground binary classification results of the previous frames to detect certain complex background environments thereby achieving higher performance than other algorithms which do not track their classification history. The classification history carries a lot of information about the background characteristics which can help in the efficient construction of the background model. Only the first frame of a video scene is required to initialize the background model. The model is updated after a fixed time interval but it keeps track of the current status in every frame. One of the novelties of this method is that it keeps track of the total number of foregrounds and backgrounds (in this article foreground and background classification results are denoted by f and b , respectively) as well as the total shift between f 's and b 's. Another novelty of this method is to use the preceding binary classification result to check the neighbourhood of each f and if necessary integrate one new background model in addition to the existing background model to identify the foregrounds efficiently. The proposed method, Record Sensitive Background Classifier (RSBC), uses two different background models for background model construction: Adaptive Background Model (ABM) and Neighbourhood Background Model (NBM). The ABM can handle different challenging video scene but NBM specially used to handle moving background objects and dynamic background. NBM removes the ghost effect efficiently and also produces accurate results in the presence of non-stationary background objects and static foreground objects in a video scene. The method shows better performance in various complex video environments with respect to the state-of-the-art methods.

This article is organized as follows. Section 2 provides a brief discussion of the existing BS algorithms. Motivation of the proposed method are given in Section 3. Section 4 discusses the details of the proposed method. Details of the performance evaluation tools are presented in Section 5 and Section 6 gives details of the experimental results. Conclusions and future works are given in Section 7.

2. Review of existing BS methods

In computer vision, background subtraction is used as a tool to extract foregrounds from a video scene for the purpose of video surveillance, object detection (people, car, boat etc.) and tracking. Over the last few decades a variety of models from various perspectives have been proposed for BS.

A statistical model for BS, using Gaussian distributions with mean μ and variance σ^2 as parameters, was proposed by Stauffer and Grimson (1999). To address the multi-modal behaviour of the background the authors mapped the background information into a fixed number of Gaussian probability density functions (*pdf*) called GMM. This method produces good results when the video scene contains dynamic background. However, the assumption of a fixed number of Gaussian distributions to model the background is unrealistic as the number of modes in the background distribution may vary from scene to scene in the same video sequence and the assumed number of Gaussians may be inadequate to handle all the dynamic back-

ground scenes. After that, many researchers used and modified GMM for better accuracy and effective application. Zivkovic and van der Heijden (2006) computed the exact number of Gaussian *pdfs* at the execution time instead of a pre-assumed number proposed in GMM (Stauffer & Grimson, 1999). Although this method properly models the dynamic background, it produces erroneous results in the case of ghost effect. Tuzel, Porikli, and Meer (2005) considered Gaussian distributions in a layered form and estimated the parameters automatically using a recursive Bayesian learning approach. Rajagopalan (2014) used a genetic algorithm to determine the optimal values of the GMM parameters (learning rate and background ratio) for higher accuracy. A shareable GMM was used Chen, Wang, and Lu (2015). The shareable model considers both foreground and background model and data points switch between the models. This method shows better accuracy over the previous BS methods which use GMM. A Type-2 fuzzy background model was built using the GMM and MRF by Zhao et al. (2012) which gives flexibility to the GMM. The fuzziness helps to properly fit the Gaussian model over the background samples but the execution time increases greatly which is an impediment for real-time processing. A real-time compressive sensing (CS) BS method for embedded camera network is proposed by Shen et al. (2016). The CS-GMM structure uses random projection matrix to reduce the dimensionality of the data while retaining valuable information. A fixed number of Gaussians to model the background is still a concern for dynamic and complex background environments. Elgammal, Harwood, and Davis (2000) proposed a non-parametric model which rely on local intensity observations to estimate background *pdf* using Kernel Density Estimation (KDE). To construct the background model from the histogram of buffered pixel values, a large amount of memory is required for storage. Chen, Chen, Huang, and Hung (2007) construct an asymmetric feed-forward framework which combines coarse-level approach with fine-level approach. The authors maintain the model with Gaussian mixtures and contrast histogram is used as a feature. The asymmetric feed-forward strategy to combine coarse and fine background models help to identify non-stationary background but this strategy unable to remove ghost effect. In the thermal domain, Davis and Sharma (2007) proposed a statistical BS method by fusing contours from thermal and visible imagery. To improve the performance of the method, an A* path-constrained is used to joining the broken segments in the fused contour image. This method is limited its scope to only extracting silhouettes of people in urban settings. A colour difference histogram and a fuzzy c-means clustering algorithm is used by Panda and Meher (2016) to detect moving objects in a video scene. Although this method efficiently handle dynamic background, it shows ghost effect and also produces erroneous results in case of shadows. A pixel based probabilistic integrated object recognition is used by Serratosa, Alquézar, and Amézquita (2012). The method is capable of handling full and partial occlusions with still or slowly moving camera. All GMM based methods discussed above are unable to simultaneously handle both the long time static foreground problem and the ghost effect problem.

Kim, Chalidabhongse, Harwood, and Davis (2005) introduced a Codebook based model, a compressed form of the background information based on a cylindrical colour model to recognize the periodic or dynamic background motion. Although the Codebook model detects dynamic background efficiently, it tends to produce false positives in the case of ghost effect. A parallel implementation of Codebook and GMM models on a supercomputer platform using automatic thread management with Open Multi-Processing (OpenMP) application programming interface is presented by Szwoch, Ellwart, and Czyżewski (2016). Karasulu and Korukoglu (2012) use entropy-based simulated annealing for BS. To increase the performance of BS, this hybrid method spends a large

amount of time to process a frame as a result of which the method is not applicable for real-time BS. Barnich and Van Droogenbroeck (2011) proposed a fast BS method that is based on a probabilistic background model called Visual Background extractor (ViBe). One uniqueness of this pixel based model is that it updates not only by its own pixel information but also by the information of its neighbourhood (chosen randomly) which is useful in ‘ghost’ suppression but static foregrounds get consumed by the background because of it. Van Droogenbroeck and Paquot (2012) modified the ViBe algorithm in their ViBe+ algorithm by using an adaptive distance measure proposed by Kim et al. (2005) in their Codebook algorithm, that inhibits the propagation of background blobs around its internal borders by using the gradient information. Wang and Shi (2016) have proposed a blind updating procedure to quickly adapt dynamic environment changes. Although this method improves the ViBe algorithm, it still produces many false negative results for slowly moving objects. St-Charles et al. proposed the LOBSTER (St-Charles & Bilodeau, 2014) and SuBSENSE (St-Charles et al., 2015) methods which are based upon ViBe, to which they added a Local Binary Similarity Pattern (LBSP) based term into the distance function and they also used pixel level feedback loops to control model adaptation. SuBSENSE shows good performance over a variety of complex background sequences but the method is not applicable in real-time processing because of its high execution time. For BS of flying targets Weng, Wang, Ho, and Sun (2010) applied segmentation operation over two consecutive image frames and obtained two binary images. The authors then subtracted the two binary images to identify the moving edges of the flying target. The detection accuracy of the flying object degrades in cluttered scene due to high uncertainty caused by the clouds. Mandellos, Keramitsoglou, and Kiranoudis (2011) proposed a pixel level system for detecting and extracting vehicles in complex road scenes. The system uses the histogram-based filtering procedure to construct the actual background but on demand background reconstruction makes the system robust under unstable lighting, different view-angles and congestion. A BS algorithm is proposed by Prasad et al. (2018) which detects objects in maritime environment. Spagnolo, Leo, Distante et al. (2006) segment the moving object using BS. In this method foreground segmentation is done by analyzing temporal images with illumination changes. Although this method is capable of detecting slow moving objects with high precision, the method may mistakenly detect objects that are motionless in the image.

Oliver, Rosario, and Pentland (2000) proposed a subspace based BS method using dimensionality reduction by applying eigenvalue decomposition. The authors accumulated a few initial frames and extracted most important features of those frames using Principal Component Analysis (PCA). As the background features are not updated periodically, it is unable to handle various complex video environments including ghost effect. Monnet, Mittal, Paragios, and Ramesh (2003) used an on-line auto-regressive model for computing fast Incremental PCA (IPCA) but it is still computationally expensive and unsuitable for real-time BS. Robust PCA (RPCA) (Candès, Li, Ma, & Wright, 2011) is an extension of PCA that extracts the low-rank structure of the image frames to produce better results than PCA. In this method the background is modelled by a low-rank subspace which can gradually change over time and the moving foreground objects constitute the correlated sparse outliers. Guyon, Zahzah, and Bouwmans (2012) used the low-rank and block-sparse matrix decomposition method to separate foreground and background. These methods does not have a mechanism to remove the ghost effect. An approximate RPCA technique (Kutz, Grosek, & Brunton, 2016) is achieved using a dynamic mode decomposition (DMD) method for evaluating the dynamics of complex systems in BS. The authors used multi-resolution DMD for further separation of dynamic content in the video on different

time scales. This method is presented without the proof of convergence to low-rank and sparse structures.

Braham and Van Droogenbroeck (2016) introduced a BS method using convolutional neural network (CNN). The Background model is estimated by extracting hierarchical features using a deep learning method and the network is trained with a scene-specific dataset. This scene-specific dataset training limits the scope of the model to some particular datasets. Nguyen, Pham, Ha, and Jeon (2018) train a triplet network to detect the changes in a video scene due to object motion. The authors use a data-driven approach which combines a sample-based background model (e.g., PBAS (Hofmann, Tiefenbacher, & Rigoll, 2012), SuBSENSE (St-Charles et al., 2015)) with a feature extractor obtained by training the network. Although this method performs well on different video datasets, it still relies on one of the existing sample-based methods to train the network. Ortego, SanMiguel, and Martínez (2018) proposed a method which shows that the accuracy of a BS methods can be increased using an improvements on the foreground segmentation masks.

The interested reader is referred to survey articles (Bouwmans, 2014; Bouwmans & Zahzah, 2014; Brutzer, Höferlin, & Heidemann, 2011), and BS libraries (Sobral, Bouwmans, & Zahzah, 2016; Sobral & Vacavant, 2014) for details on the BS field.

In the fields of BS, researchers have been using various models like GMM, KDE, Codebook, probabilistic, PCA, low-rank etc. to handle various complex scenarios such as dynamic background motions, camouflaged objects, moving background objects, static foreground objects etc. in a video scene. To the best of the authors knowledge, no BS algorithm in the literature till the present has considered the binary classification history of previous frames, i.e., the nature of the previous foregrounds and backgrounds. In this article a BS method is proposed by considering the previously separated foreground and background classification.

3. Motivation for the proposed algorithm

Two of the biggest weaknesses of pixel based BS techniques are the problem of “dynamic background” and the problem of “ghost effect”. Dynamic background is the effect in which a moving object in the background is classified as foreground. Ghost effect occurs when a stationary object in the background begins to move, as a result of which the old position of that object is wrongly classified as foreground. One of the novel features of the proposed method is to track the classification history of previous frames. This classification history is then used for the purpose of model update which is in contrast to only tracking the history of the past observed pixel values that is currently used by most state-of-the-art algorithms such as Codebook (Kim et al., 2005), ViBe (Barnich & Van Droogenbroeck, 2011), PBAS (Hofmann et al., 2012) and all the PBAS derivatives like SuBSENSE (St-Charles et al., 2015), etc. The classification history of previous frames can help in the detection of dynamic background in a video scene. In Fig. 1, an annotated example shows why the classification history is so important for detecting dynamic background. In the given figure, we plot the absolute difference between the R channel values of the first 300 frames and the R channel value of the first frame, at a fixed pixel position (coordinates (123,200), marked as a red dot inside the green circle in Fig. 1(a)). The image is taken from the fountain01 video scene of “dynamic background” category of CD-net dataset (Goyette, Jodoin, Porikli, Konrad, & Ishwar, 2012). For the first frame, the difference is zero. Threshold value (15) is represented by a black horizontal line. The red star (representing background or b) and green plus (representing foreground or f) signs indicate that the absolute global difference (Roy & Ghosh, 2018) value is more than or less than the threshold value, respectively. In the dynamic background environment red stars and green pluses

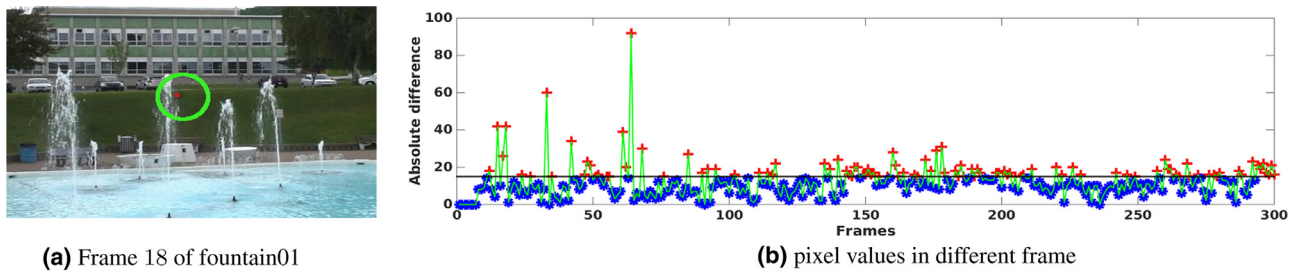


Fig. 1. R channel values (coordinates (123,200)) of the first 300 frames of *Fountain02* video scenes of CDnet dataset. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article).

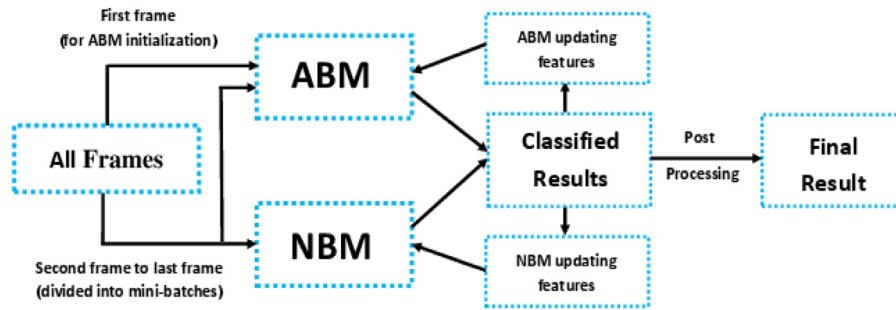


Fig. 2. Pipeline of the proposed RSBC algorithm. The first frame of a video scene is used to initialize the ABM. The rest of the frames are sub-divided into mini-batches. Every pixel of every frame (except the first frame) is classified by both ABM and NBM. The classification history of each mini-batch is used to generate ABM and NBM updating features to update both ABM and NBM adaptively. The final results are created by post-processing the classified results.

alternate frequently (continuous foreground or background span is very small). This figure shows that if the video scene contains a complex environment such as dynamic background where a pixel's value changes very frequently, the foreground and background results also switch very often. We count the number of back and forth switches in the classification history as a feature for adaptively updating the background model which can help in efficiently detecting dynamic background. another novel feature of the proposed algorithm is to tackle the ghost effect problem by considering the neighbouring pixels' classification history in addition to the classification history of each individual pixel. This also helps in detecting dynamic background efficiently.

From a theoretical perspective the importance of tracking the previous classification history is that it allows us to simultaneously handle both ghost effect and dynamic background without sacrificing static foreground. Most of the current state-of-the-art pixelwise BS algorithms can handle dynamic background efficiently. But in doing so they have to either sacrifice handling the ghost effect or sacrifice handling long time static foreground. Current algorithms find it difficult to handle all three effects simultaneously. Some state-of-the-art algorithms handle dynamic background by including the pixel values of the neighbouring pixels into the background model. This also allows them to handle ghost effect but this also destroys long time static foreground by misclassifying it as background. On the other hand some algorithms handle dynamic background by including the pixel values of a few initial frames into the background model. But these models suffer from the disadvantage that they cannot recognize the ghost effect. The proposed model by tracking the past classification history is able to identify dynamic background and ghost effect separately and thus being able to update the background models accordingly so that both dynamic background and ghost effect are classified as background, while static foreground is stays properly classified as foreground.

4. Proposed method

4.1. Overview

A diagram showing the pipeline of the sequence of movements of the proposed RSBC method is given in Fig. 2. In Table 1, we de-

scribe few notations used in this article. The objective of the proposed RSBC method is to identify the moving objects in a video scene, which is done by classifying every pixel of every frame into one of two classes, namely, *foreground* or *background* (in this article foreground and background classification results are denoted by f and b , respectively). To perform this classification we need to construct a *background model*. The term *background model* refers to a mathematical construct which gives a general description the background regions of any video scene and which can be used to distinguish between the background regions from the foreground regions of the video scene. In our pixel based method, we begin by building separate background models for each pixel position of the video sequence. Each of these models are first initialized by using the first frame of the video sequence, in a way similar to some previous methods such as ViBe (Barnich & Van Droogenbroeck, 2011), PBAS (Hofmann et al., 2012), SuBSENSE (St-Charles et al., 2015) algorithms. All pixel values of the subsequent frames (referred to as test frames henceforth) are used for moving object detection and also for updating the background models. To detect moving objects, each pixel of every test frame (referred to as a test pixel henceforth) is classified into f or b by matching it with the corresponding background model (using certain matching criteria described below) and if a match is found then the pixel is declared as background (or b) or if a match is not found then it is declared as foreground (or f). Any BS algorithm must frequently update its background models so that it can maintain accuracy over long periods of time. The background models of the RSBC algorithm corresponding to every pixel position are periodically updated in order to incorporate new background information into the model.

The proposed RSBC model is composed of two separate classifiers, namely, the ABM and the NBM. These models consist of lists of pixel intensity values (or model elements) which are matched with the intensity value of the pixel being tested. Both ABM and NBM model elements are used together to classify every pixel of every test frame. The ABM itself is composed of three separate lists of intensity values, one for each colour channel (in the case of RGB image, a separate list ABM_R , ABM_G and ABM_B for the R, G and B channels, respectively). These three lists for the three colour channels are independently initialized and independently updated and

Table 1
Notations used in the RSBC algorithm with description.

Notation	Description
b	label used to represent a background pixel
f	label used to represent a foreground pixel
$Fwindow$	a mini-batch of 50 consecutive frames
$P_C^t(x)$	pixel value of the x th pixel position of the C th channel of the t th frame
$ABM_C^i(x)$	i th entry of the ABM for the x th pixel of channel C
$PABM_C^i(x)$	i th entry of the PABM for the x th pixel of channel C
$\#Transition$	total number of transitions between b to f or f to b in a mini-batch
$AbsDiff$	absolute difference between the number of f 's and the number of b 's in a mini-batch
τ_1, τ_m, τ_s	parameters used to update ABM
ϑ and γ	parameters used to find similarity between a pixel value with model elements
ζ_1 and ζ_2	parameters used to update NBM

therefore may contain different number of elements at any point of time. Given a test value that has to be matched with the ABM to see if that test value belongs to a background or foreground region, the matching is done by calculating the absolute difference of the test pixel intensity value with all of the corresponding model elements. A given test value is considered as a successful match under two cases. In case 1, if any one of the absolute differences of the test value with the model elements is not more than a predefined threshold ϑ then the test value is considered a successful match. In case 2, if the absolute difference of the test value with the corresponding pixel position of the first frame (absolute global difference) is not more than another predefined threshold γ then also the test value is considered as a successful match. A test pixel is finally declared as a match with the ABM if all of its three colour components (R, G and B values) find a successful match with their respective three colour models (ABM_R , ABM_G and ABM_B). The ABM is updated periodically after every mini-batch of $Fwindow$ ($= 50$) number of frames, depending upon criteria that are derived from the classification history of the previous frames.

NBM on the other hand contains a single combined list of colour triplets where all three colour components are initialized and updated simultaneously. NBM may not necessarily be initialized with the first frame but depending upon certain neighbourhood criteria, it may be initialized at any frame in the middle of the video sequence. The colour triplet values of a test pixel position is declared as a match with NBM if its Chebyshev distance with respect to any one of the triplets within NBM (i.e., NBM elements) is not greater than the threshold ϑ . (Chebyshev distance is a metric which calculates the absolute differences between each component of two vectors and returns the maximum of those. For two vectors \vec{x} and \vec{y} with coordinates x_i and y_i , respectively, the Chebyshev distance is defined as $D_{Chebyshev}(\vec{x}, \vec{y}) = \max_i (|x_i - y_i|)$). The NBM is updated continuously after testing every frame of the video sequence. Just like the ABM, the NBM is also updated depending upon criteria that are derived from the classification history of the previous frames.

Finally, a given pixel is classified as b if the pixel value is matched with either the ABM or the NBM, otherwise it is classified as f .

To update the ABM and the NBM models, we used a few novel features, extracted from the history of previous classification results. To update the ABM, after testing each mini-batch of consecutive frames, we compute three characteristics (or features) from the classification history of each individual pixel. These features are - the total number of f 's, total number of b 's and total number of transitions between f 's and b 's. To initialize and adaptively update the NBM, we used another feature: a two-level neighbourhood (marked by $1/s$ and $2/s$ in Fig. 5(c)) checking for f pixels, obtained from the previous classification result. The detailed update procedures is described below separately for ABM and NBM.

To increase the accuracy of the RSBC method, post-processing (consisting of a few basic morphological operations and a median

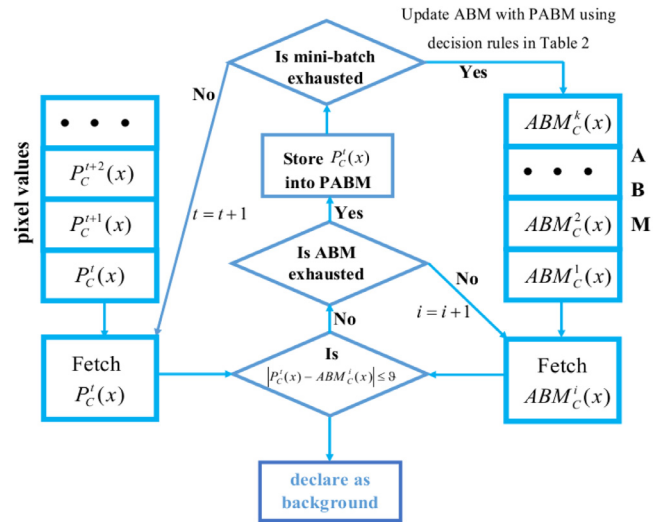


Fig. 3. Block diagram of the functioning of the ABM.

filter) is applied over the classification results to generate the final result.

4.2. Adaptive Background Model (ABM)

Fig. 3 depicting the functioning of the ABM.

4.2.1. Motivation

If a video scene contains dynamic background (such as a swaying tree or ripples on water etc.), then the background region changes very frequently. From a statistical point of view we can say that the intensity distribution of a pixel in dynamic region will have multi-modal nature, i.e., the distribution will contain multiple dominant peaks since multiple objects occupy that particular pixel position. In such an environment, intensity values are distributed among multiple modes. Now if a background classifier considers a fixed value as a background model element, it will output a rapidly alternating string of foregrounds (or f 's) and backgrounds (or b 's). A similar situation is shown in Fig. 1(b) where a fixed value 15 is considered as a background model element and as a result we can find a frequently alternating sequence of foregrounds (green plus sign) and backgrounds (red star sign). In such cases the classifier should adaptively update itself by including all the background modes into the background model. On the other hand, if the video scene contains static background, the frequency count of a single mode will be very high with respect to the other modes (if at all they exist because in static region more than one mode is rarely present) and the classifier should output a long string of either f 's for objects or b 's for backgrounds. Thus to identify dynamic background in a video scene, we count the frequencies of f 's and b 's,

and also the frequency of transitions between f 's and b 's from the past classification history. These frequency counts can be used to obtain a criterion that distinguishes dynamic background from true foreground. These criterion can then be used to update the background model accordingly. If the criterion indicates that the previous f 's were due to dynamic background then we update the background model accordingly to include those pixel values so that in future those pixels are classified as background. Otherwise if our criterion indicates that the previous f 's were not because of dynamic background then we do not update the background model.

4.2.2. Initialization

An ABM is associated with every pixel position and for each of these ABMs a separate ABM_C is constructed for each colour channel $C \in \{R, G, B\}$. ABM_C consists of a list of intensity values (or elements) of the channel C . Let the pixel value of the x th pixel of the C th channel of the t th frame be denoted by $P_C^t(x)$, where the first frame starts with $t = 0$, and let $ABM_C^i(x)$ be the i th entry of the ABM for the x th pixel of channel C , where the first entry begins with $i = 1$. The ABMs for every pixel of every channel are initialized using the pixel values of the first frame. Thus we initialize the ABM of pixel x by setting $ABM_C(x) = ABM_C^1(x)$ where $ABM_C^1(x) = P_C^0(x)$ for each channel C .

Once initialized, each ABM_C is updated after every mini-batch of $Fwindow$ number of consecutive frames ($Fwindow = 50$ for this article). At each update some new pixel values (or elements) may or may not be appended to the existing list of elements in each of the ABM_C 's. In between two successive updates, various criteria (or features) are computed based upon the previously classified results within that mini-batch, which determine how to update the ABM.

4.2.3. Classification

Each pixel of each test frame is first classified by matching its three colour components (R, G and B) with the respective three ABMs (ABM_R , ABM_G and ABM_B). For pixel x , matching a given colour component (or intensity value) of channel $C \in \{R, G, B\}$ of t th test frame, $P_C^t(x)$, with its corresponding $ABM_C(x)$ is done in two steps. First to test if there exists an element in $ABM_C(x)$ (say the i th entry, see Fig. 3), we compute the expression $|P_C^t(x) - ABM_C^i(x)| \leq \vartheta$, where $|\cdot|$ denotes the absolute value and ϑ is a predefined threshold. If this expression is true, then that colour component is considered as a successful match. If the expression is false, then in the second step we test another expression $|P_C^t(x) - P_C^0(x)| \leq \gamma$ (γ is a predefined constant) to test the absolute global difference. If this expression is true then the match is declared successful, and otherwise it is declared unsuccessful. If a successful match is found for all three colour components then that pixel is declared as b . If however a successful match is not found for even one of the three colour components, the pixel is marked as *probable foreground* and the algorithm proceeds to NBM as described in the next subsection. If NBM also fails to recognize the pixel as background then the pixel is declared as f , else if NBM recognizes the pixel as background then the pixel is declared as b .

4.2.4. Update

The ABM is updated at the end of every mini-batch of $Fwindow$ number of consecutive frames. For this purpose of updating, we maintain another temporary list of numeric pixel intensity values, which we term as *Probable ABM* (PABM) for each channel C , i.e., a temporary $PABM_C$ is associated with every ABM_C . Every element of $PABM_C$ is associated with a frequency value, which counts the number of occurrences of that particular $PABM_C$ entry within the mini-batch. For every mini-batch, a new list of $PABM_C(x)$ is created for each colour channel $C \in \{R, G, B\}$ of each pixel x , as described next. If a test pixel value, $P_C^t(x)$, of a test frame (say t th frame) does not match with the elements of the $ABM_C(x)$, then that test

Table 2

Various decision rules used to adaptively update the ABM.

Case	Decision rules	Decision
1	$\#Transition \geq \tau_l$	include in the ABM
2	$\#Transition \geq \tau_m$ and $AbsDiff \leq \tau_s$	include in the ABM
3	$\#Transition \geq \tau_s$ and $AbsDiff \geq \tau_m$	include in the ABM
4	Else	ABM stays unchanged

value is matched with the elements of $PABM_C(x)$. If the matching is successful then the corresponding frequency count is increased by 1. If the matching is unsuccessful then the test value is appended to the $PABM_C(x)$ list and its frequency is set to 1. Matching is done by calculating the expression $|P_C^t(x) - PABM_C^i(x)| \leq \vartheta$ where ϑ is a predefined threshold and $PABM_C^i(x)$ is the i th entry.

At the end of a mini-batch, when all the previous test frames have been classified, each $PABM_C(x)$ is used to update corresponding $ABM_C(x)$ for each pixel x of channel C . Initially each list of pixel values in $PABM_C(x)$ is trimmed by discarding the values which have frequency counts less than 10% of $Fwindow$ ($= 10\%$ of $50 = 5$ for this article). Then to take the decision, whether or not to merge the remaining elements of the $PABM_C(x)$ with the existing $ABM_C(x)$, we look at the binary classification history of f and b results of the previous frames within that mini-batch. These binary classification results are used to compute the following two quantities: first is the number of transitions between f to b or b to f in that mini-batch, which we term as $\#Transition$, and the second is the absolute difference between the number of f 's and the number of b 's (i.e., $|\#f - \#b|$), which we term as $AbsDiff$. Conditioned upon these two variables it is decided, as shown in Table 2, whether or not to merge the $PABM_C(x)$ elements with the pre-existing $ABM_C(x)$ list.

Table 2 shows the rules used to make this decision. In Table 2, $\#Transition$ stands for the number of transitions between f to b or b to f , $AbsDiff$ is absolute difference between the number of f 's and b 's within a single mini-batch. Three predefined threshold values τ_l , τ_m , and τ_s are used to make update decisions of the ABM. τ_l , τ_m , and τ_s stand for large, medium and small thresholds for the quantities $\#Transition$ and $AbsDiff$. We choose the values of these thresholds so as to be able to effectively distinguish dynamic background from true foreground. We separate the identification of dynamic background into three separate cases, with a rule attached to each case. In case1, a high value of $\#Transition$ indicates a large number of transitions between f to b or b to f , i.e., an indication of dynamic background, as discussed in Section 3. Case2 is for identifying those cases when the value of $\#Transition$ is not as high as in case1 but has a moderate value and along with that the value of $AbsDiff$ is small, indicating roughly equal numbers of f 's and b 's, which also signifies dynamic background. In case3, if $AbsDiff$ is moderate and if $\#Transition$ is more than a small threshold, we also sense the dynamic nature of the background. Thus in all these three cases we add the associated pixel values into the ABM, thus correctly classifying the dynamic background as a background region in future frames. Whenever the rules of Table 2 indicate the presence of dynamic background we include the corresponding pixel values into the ABM. In all other cases the ABM is kept unchanged.

4.3. Neighbourhood Background Model (NBM)

4.3.1. Motivation

Through the NBM we propose a novel feature to efficiently suppress ghost effect. Ghost effect occurs when an object that is part of the background begins to move. The position at which the object had been stationary is now misclassified as f and it seems as if the object has left its ghost behind.

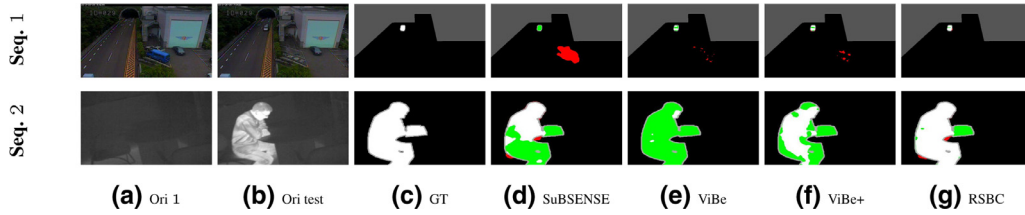


Fig. 4. Original first input image (Ori1), Original test input image (Ori test), ground-truth images (GT), results corresponding to SuBSENSE, ViBe, ViBe+ and RSBC on *tunnelExit_0_35fps* and *library* video sequence. White, red, green and black colours represent the true positives, false positives, false negatives and true negatives, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

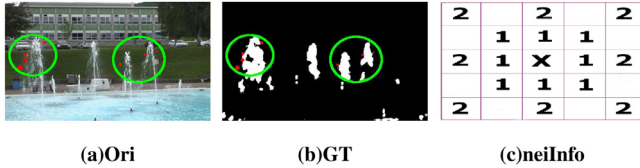


Fig. 5. Fountain01 and the nearest neighbour info.

To suppress the ghost effect from the binary classification results, Barnich and Van Droogenbroeck (2011) in their ViBe algorithm proposed a ghost removal technique (spatial diffusion) which was later used by many authors including those of PBAS (Hofmann et al., 2012), LOBSTER (St-Charles & Bilodeau, 2014) and SuBSENSE (St-Charles et al., 2015) to improve the accuracy of the results. In this technique, during the updating stage of the background model, the pixel values of a background pixel is added to the background model of one of its 8 nearest neighbours and that neighbour is chosen randomly. This random selection may end up with different results on the same input dataset for different runs of the method. Although this technique does reduce ghost effect from video scenes that contain moving backgrounds, it suffers when a video scene contains objects with small motions. This scheme transforms a foreground object into background if the object has little motion (see Seq. 2 of Fig. 4(d) and (e)). As a result, accuracy decreases rapidly due to the increasing numbers of false negatives. To remedy this situation, Barnich and Van Droogenbroeck (2011) modified the ViBe method in ViBe+ (Van Droogenbroeck & Paquot, 2012) to improve the accuracy of the binary results; but even this method fails to solve the problem completely (see Seq. 2 of Fig. 4(f)).

To detect the ghost effect, our algorithm detects f pixels that suddenly appear in the middle of b pixels. Since this can also arise because of dynamic background, our algorithm tracks the future behaviour of that pixel to determine whether the f appeared because of ghost effect or because of dynamic background. In case of dynamic background environment, pixels frequently switch back and forth between f 's and b 's (see Fig. 1(b)). On the other hand in case of ghost effect a pixel position will steadily be classified as f 's for the next few consecutive frames. Therefore our scheme for removing the ghost effect is use the frequencies of f 's and b 's to obtain a criterion for identifying the ghost effect. If our criterion indicates ghost effect then we update the background model to include those pixel values so that in future the model classifies it as background. Otherwise if our criterion does not indicate ghost effect then we do not update the background model. The details of the update process are described below.

4.3.2. Initialization and Classification

To begin with all the pixels in a video sequence have an empty NBM associated with them. As the NBMs are updated some pixels have non-empty NBMs while others continue to have empty

NBMs. A video sequence without any dynamic background or moving background will have empty NBMs for all pixels. If a pixel has empty NBM then the classification output produced by the ABM corresponding to that pixel is the final classification. For a pixel with non-empty NBM, the final classification is f if and only if both ABM and NBM classify the pixel as f . A test colour triplet is classified by the NBM by matching that triplet with the list of existing triplets in NBM. A successful match occurs when the Chebyshev distance of that triplet with respect to any one of the triplets in NBM is not more than the threshold ϑ . If a successful match occurs the pixel is declared as b . If a match is unsuccessful NBM declares the pixel as f , since, as mentioned earlier, NBM is checked if and only if ABM fails to classify the pixel as b .

4.3.3. Update

The update procedure of NBM is as follows: if a pixel position is classified as f and if the pixel x including its neighbouring pixels (see Fig. 5) in the previous frame were all classified as b , then we track the future classifications of pixel x . If pixel x is classified as f for all subsequent ζ_1 consecutive frames, then it probably occurred due to ghost effect and therefore the NBM is updated by appending the current colour components with the existing NBM. If on the other hand all future ζ_1 consecutive frames are not f then we continue tracking up to ζ_2 ($\zeta_2 > \zeta_1$) future consecutive frames and if these ζ_2 frames contain not less than ζ_1 number of foregrounds then this probably occurred because of dynamic background and so once again we append the colour components of the current pixel to the existing NBM. In this technique no random or blind update takes place like ViBe (Barnich & Van Droogenbroeck, 2011), PBAS (Hofmann et al., 2012), SuBSENSE (St-Charles et al., 2015) algorithms.

4.4. Final foreground-background classification

A test pixel is classified as f only if both ABM and NBM classify it as f . Otherwise it is declared as b . For every test pixel x the ABM is tested first followed by the NBM. If the ABM recognizes x as a background then the pixel is declared as b and the NBM is skipped altogether since the NBM need not even be updated for b pixels.

$$x \in \begin{cases} f, & \text{if no match in both ABM \& NBM} \\ b & \text{otherwise.} \end{cases} \quad (1)$$

After classifying a video frame into foreground and background, we use some post-processing to improve the accuracy of the results. This post-processing involves a 9×9 median filter and basic morphological operations to refine the detected results by eliminating noise.

5. Performance evaluation

To establish the robustness of the proposed RSBC algorithm, we compare it with some state-of-the-art algorithms using the CDnet

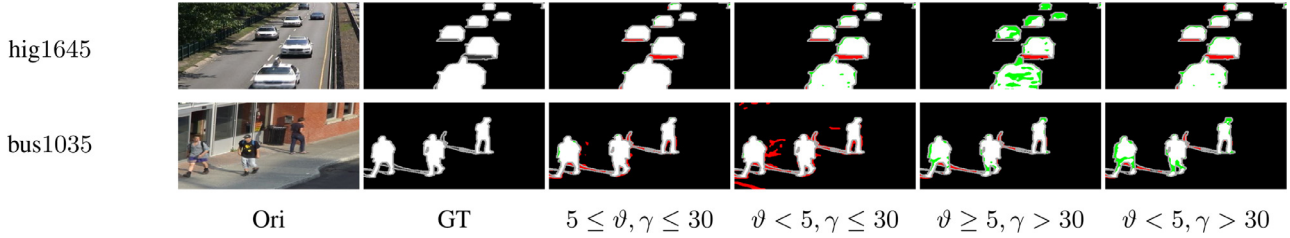


Fig. 6. Original input images (Ori), ground-truth images (GT), results applying proposed RSBC method using different values of ϑ and γ on frame number 1645 of *highway* (hig1645) and frame number 1035 of *busStation* (bus1035) video sequences from CDnet dataset. White, red, green and black colours represent the true positives, false positives, false negatives and true negatives, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3

Parameters ϑ and γ with F1 score in *highway* and *busStation* video sequences.

Parameters ϑ and γ	F1 score of <i>highway</i> sequence	F1 score of <i>busStation</i> sequence
$5 \leq \vartheta, \gamma \leq 30$	0.941	0.859
$\vartheta < 5, \gamma \leq 30$	0.882	0.803
$\vartheta \geq 5, \gamma > 30$	0.805	0.759
$\vartheta < 5, \gamma > 30$	0.858	0.723

dataset and evaluated with some standard evaluation metrics. Descriptions of all these are given below.

5.1. Parameter settings

All complex video scenes containing dynamic background or moving background may not be optimally handled by a fixed set of parameter values. To extract the best results the proposed method uses a few tunable parameters. The parameter values used in the present work have been experimentally tuned to make the algorithm as robust as possible. Our algorithm uses a set of 8 parameters, whose details are listed individually below. Some of these parameters are involved in the classification of each pixel into foreground or background while others are involved in updating the model. Out of these 8 parameters 6 have been assigned fixed values, as the output does not show much sensitivity to their values. For the remaining 2 parameters, namely ϑ and γ , the output shows high sensitivity to the values used, and different values work well for different video sequences. These 2 parameters ϑ and γ are used to classify a test pixel into foreground or background by checking the similarity between a test pixel value with the background model elements. These parameters are chosen in such a way which reduces the classification errors. A very small value of these parameters increases the number of false positives and a very large value incorrectly includes many foreground pixels into the background thus increasing the number of false negatives. However, we have experimentally found that for most video sequences we can obtain optimal results by assigning these 2 parameters one or more values in the range [5, 30]. In Fig. 6, we show few results for different combination of ϑ and γ , which shows that any values of ϑ and γ outside the range [5, 30] almost always produces suboptimal output. This observation is also shown in the Table 3, where we find the F1 scores for different combination of ϑ and γ chosen in columns 3,4,5 and 6 of Fig. 6. Even within the interval [5, 30], we have found that one of the 4 values {5, 10, 20, 30} is sufficient to produce results as good as any other value within that interval. Fig. 7 shows the variation of the F1 score with respect to the ϑ and γ parameters. From Fig. 7 we can see that the rate of change of the F1 score with the parameter values is slow enough that one of the 4 values {5, 10, 20, 30} is sufficient to produce results as good as any other value within the interval [5, 30]. The parameters used

in the proposed algorithm are described in short below. For a full description and motivation see Section 4.

(P1). $\vartheta \in \{5, 10, 20, 30\}$: This parameter is a threshold value that is used to test whether a test pixel value matches with the list of values in ABM, PABM and NBM. If the absolute difference of the test pixel value from the value being matched with is not more than ϑ then the match is considered successful. Different values of ϑ are effective in handling different video scenes. A variety of ϑ values makes the proposed method more robust in handling various complex video scenes, but only a single value is used in a given video sequence. As discussed above, for most video scenes one of the values in the set {5, 10, 20, 30} will almost always produce good results (see Fig. 7(b) and (d)), depending upon the video sequence.

(P2). $\gamma \in \{5, 10, 20, 30\}$: If a test pixel value at position x does not match with ABM using the condition mentioned above, then in the next step we compute the absolute difference of the test pixel with the pixel value of the first frame at position x and compare this difference with the threshold parameter γ . If this difference is not more than the threshold parameter γ , then the test pixel is declared as background. The purpose of this parameter is to give special status to the first frame of the video sequence by using a relaxed value of the threshold for the matching criteria. This parameter γ is assigned a value not less than ϑ because this test is only performed only upon failure of the test mentioned in P1 above and if γ is assigned a value less than ϑ then this test will also be guaranteed to fail, which will be futile. Just like the parameter ϑ above, we have found that for most video sequences any one of the values in {5, 10, 20, 30} is enough to produce optimal results (see Fig. 7(a) and (c)).

(P3). $Fwindow = \{50\}$: To update the ABM, $Fwindow$ number of consecutive frames (a mini-batch) are used to analyse the previous classification history. In all our experiments, we used a single value of $Fwindow$.

(P4). $\zeta_1 = \{20\}$: If a f pixel appears suddenly in the midst of b pixels at location x , then we track that pixel location x for the subsequent ζ_1 number of frames and if all those ζ_1 frames are classified as f then we add that pixel value to the NBM. This value is fixed in the algorithm.

(P5). $\zeta_2 = \{30\}$: In continuation from the above, if all the ζ_1 frames are not classified as f then we continue to track that pixel location for upto ζ_2 number of frames and if out of these ζ_2 , at least ζ_1 number of frames are classified as f then also we add the pixel value to the NBM.

(P6). $\tau_l = \{30\}$, (P7). $\tau_m = \{20\}$ and (P8). $\tau_s = \{12\}$: These three parameters are used to decide whether to update the ABM or not using the decision rules shown in Table 2. τ_l , τ_m , and τ_s stand for large, medium and small thresholds for the quantities $\#Transition$ and $AbsDiff$. If the value of $\#Transition$ (the number of transitions between f 's and b 's in a mini-batch) is more than large threshold τ_l , then the pixel is considered as belonging to dynamic back-

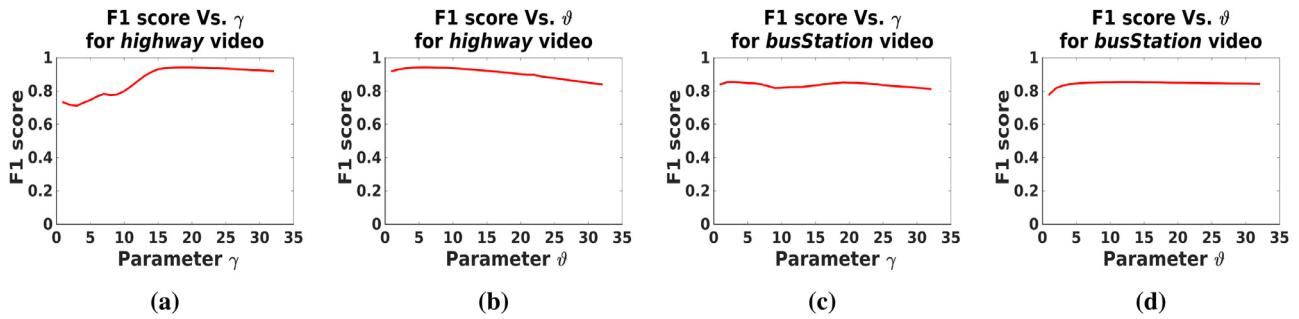


Fig. 7. Variation of F1 score with respect to algorithm parameters (see Section 5.1) for two video sequences, namely *highway* and *busStation*.

ground region and therefore the ABM is updated by merging the values of PABM with it. If it is less than τ_l then depending upon the value of *AbsDiff* (the absolute difference between total number of *f*'s and *b*'s) we may or may not update the ABM, as specified by the decision rules in Table 2.

5.2. Datasets

We used the CDnet dataset for evaluation and comparison purposes. The dataset, initially produced in 2012, has 6 categories with a total of 31 video sequences which contain many real life indoor and outdoor scenarios. Most of the video frames are accompanied with ground-truth frames. In 2014, the authors expanded the old dataset with 5 more categories with 22 more video sequences and corresponding ground-truths. Size of a frame in the CDnet dataset varies from 320×240 to 720×576 pixels.

5.3. Evaluation metrics

To evaluate the performance of the proposed RSBC method, four metrics Precision, Recall, F-measure (F1), and Matthew's Correlation Coefficient (MCC) are used defined as,

Precision = $\frac{TP}{TP+FP}$, Recall = $\frac{TP}{TP+FN}$, F-measure = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ and MCC = $\frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}$, where TP = true positives, FP = false positives, TN = true negatives and FN = false negatives.

5.4. Existing techniques used for comparison

To compare the efficiency of the proposed method, we used several state-of-the-art BS methods. Such as, (i) Gaussian mixture model (GMM) (Stauffer & Grimson, 1999), (ii) Type-2 fuzzy MRF (T2MRF) method (Zhao et al., 2012), (iii) Codebook (CB) model (Kim et al., 2005), (iv) Pixel-based adaptive segmenter (PBAS) method (Hofmann et al., 2012), (v) Vehicle trajectories evaluation (VuMeter) method (Goya, Chateau, Malaterre, & Trassoudaine, 2006), (vi) Local Binary Similarity Patterns (LOBSTER) method (St-Charles & Bilodeau, 2014), (vii) ViBe universal BS method (Barnich & Van Droogenbroeck, 2011), (viii) ViBe+ (Van Droogenbroeck & Paquot, 2012), (ix) Self-balanced sensitivity segmenter (SuBSENSE) method (St-Charles et al., 2015), (x) Adaptive Background Modeling (Zhong17) (Zhong, Zhang, Lu, Zhao, & Xu, 2017), (xi) FCDH (Panda & Meher, 2016) (xii) ViBeF (Wang & Shi, 2016).

For these methods, to generate the comparative results, we use the same parameter settings as provided by the authors in their corresponding papers.

5.5. Colour codes used for visualizing experimental results

The ground-truth (GT) images shown in the Figs. 4, 6 and 8, as provided in the CDnet dataset, use white and black colours

Algorithm 1 RSBC method (Code: https://www.isical.ac.in/~sujroyroy_r/RSBCbs).

Input: A sequence of video frames.

Initialize: First frame is used to initialize the ABM.

```

1: for all test frames do
2:   for each pixel position x do
3:     Search the pixel value in the ABM using parameter  $\vartheta$ .
4:     Unsuccessful search further considers the absolute global
       difference. If this value is more than parameter  $\gamma$ , create PABM.
5:   end for
6:   From the binary classified results,
7:   If there is a transition from b to f or vice-versa, increment
       the parameter #Transition.
8:   If the decision is f, check its two-level neighbours (see Fig-
       ure 5c). If all the neighbours are b, track the future results of
       the pixel x.
9:   For all such pixels, initialize or update the NBM using pa-
       rameters  $\zeta_1, \zeta_2$ .
10:  After a mini-batch of Fwindow frames, modify the ABM us-
       ing PABM and parameters  $\tau_l, \tau_m$ , and  $\tau_s$ .
11: end for

```

Output: Binary classified images.

to represent foreground and background pixels, respectively. Non-ROI pixels that are half-occluded or corrupted by motion blur are coloured with dark gray and light gray colours, respectively. In addition to the above, the output images from all the state-of-the-art methods under comparison use the colour red for false positives (background detected as foreground) and green for false negatives (foreground detected as background).

6. Experimental results

To analyze the performance of the proposed method with respect to the state-of-the-art methods, same video sequences are used for all the methods. Qualitative and quantitative evaluations are used to examine and grade the performance of the various methods. We chose 10 video sequences from various categories of the CDnet 2014 dataset, which contain a variety of challenging indoor and outdoor video scenes. The video scenes used are *blizzard* from "badWeather", *highway* from "baseline", *badminton* from "cameraJitter", *fall* from "dynamicBackground", *streetLight* from "intermittentObjectMotion", *tunnelExit_0_35fps* from "lowFramerate", *busyBoulevard* from "nightVideos", *busStation* from "shadow", *library* from "thermal" and *turbulence3* from "turbulence" category. We tested our algorithm on many other video sequences and those results are available in the supplementary material¹.

¹ https://www.isical.ac.in/~sujroyroy_r/RSBCbs.

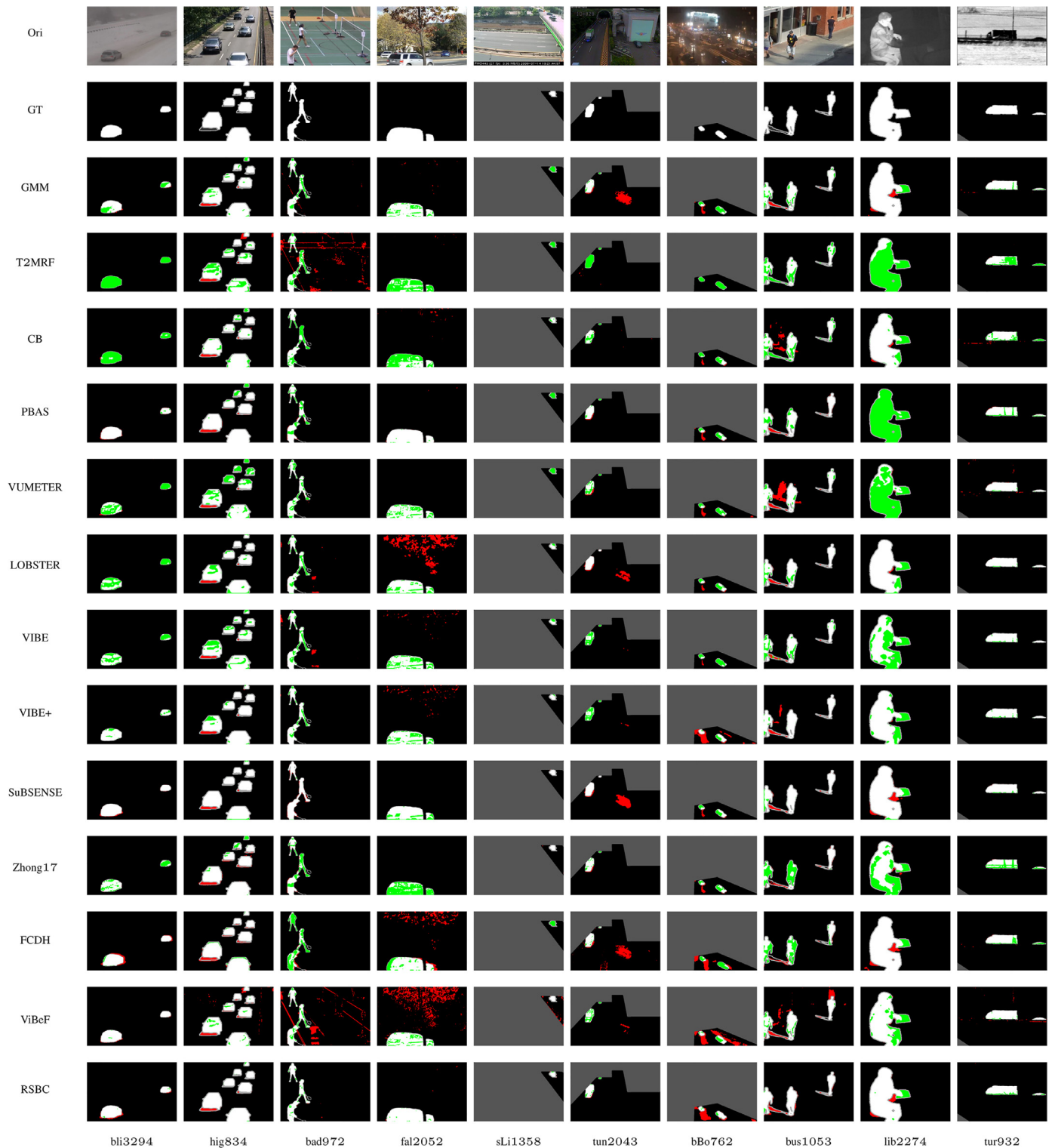


Fig. 8. Sample foreground detection output of 10 frames taken from 10 video clips of the CDnet 2014 dataset. Each column depicts figures of original input images (Ori), ground-truth images (GT), results corresponding to GMM, T2MRF, CB, PBAS, VUMETER, LOBSTER, ViBe, ViBe+, SuBSENSE, Zhong17, FCDH, ViBeF and the proposed method RSBC. Columns represent, from left to right, 3294th frame of *blizzard* (bli3294), 834th frame of *highway* (hig834), 972th frame of *badminton* (bad972), 2052th frame of *fall* (fal2052), 1358th frame of *streetLight* (sLi1358), 2043th frame of *tunnelExit_0_35fps* (tun2043), 762th frame of *busyBoulevard* (bBo762), 1053th frame of *busStation* (bus1053), 2274th frame of *library* (lib2274) and 932th frame of *turbulence3* (tur932), respectively. White, red, green and black colours represent the true positives, false positives, false negatives and true negatives, respectively. The explanation of this figure is given in Section 6.1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4

Precision (Pre), Recall (Rec), F1 score and MCC values of different methods. In each column, red, green and blue colour represent the best, the second best and the third best measurements.

	blizzard				highway				badminton				fall				streetLight			
	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC
GMM	0.689	0.597	0.640	0.636	0.916	0.770	0.837	0.841	0.686	0.561	0.617	0.621	0.707	0.589	0.643	0.651	0.618	0.308	0.412	0.422
T2MRF	0.339	0.091	0.144	0.144	0.766	0.467	0.580	0.579	0.381	0.276	0.320	0.309	0.338	0.226	0.271	0.293	0.442	0.105	0.170	0.184
CB	0.539	0.148	0.232	0.235	0.930	0.752	0.831	0.838	0.544	0.368	0.439	0.432	0.593	0.301	0.399	0.407	0.726	0.593	0.653	0.664
PBAS	0.817	0.704	0.756	0.763	0.853	0.915	0.883	0.893	0.737	0.803	0.769	0.775	0.852	0.699	0.768	0.766	0.633	0.605	0.619	0.638
VUMETER	0.497	0.282	0.360	0.352	0.741	0.499	0.597	0.603	0.754	0.609	0.674	0.690	0.478	0.403	0.437	0.418	0.684	0.398	0.503	0.507
LOBSTER	0.584	0.474	0.523	0.523	0.897	0.807	0.850	0.846	0.788	0.636	0.704	0.714	0.370	0.779	0.502	0.517	0.670	0.603	0.635	0.640
ViBe	0.676	0.503	0.577	0.577	0.892	0.792	0.839	0.822	0.691	0.512	0.588	0.585	0.663	0.449	0.535	0.537	0.673	0.612	0.641	0.643
ViBe+	0.884	0.726	0.797	0.783	0.926	0.837	0.879	0.880	0.787	0.658	0.717	0.723	0.749	0.609	0.672	0.683	0.741	0.680	0.710	0.723
SuBSENSE	0.852	0.795	0.823	0.819	0.944	0.920	0.932	0.933	0.807	0.877	0.841	0.839	0.791	0.759	0.775	0.788	0.743	0.682	0.712	0.721
Zhong17	0.693	0.715	0.704	0.715	0.892	0.823	0.856	0.851	0.637	0.716	0.674	0.673	0.215	0.382	0.275	0.270	0.709	0.742	0.725	0.723
FCDH	0.805	0.703	0.751	0.747	0.912	0.881	0.896	0.898	0.534	0.662	0.591	0.595	0.549	0.586	0.567	0.571	0.679	0.472	0.557	0.622
ViBeF	0.834	0.772	0.802	0.793	0.879	0.804	0.840	0.835	0.483	0.556	0.517	0.518	0.471	0.517	0.493	0.492	0.705	0.661	0.682	0.687
RSBC	0.872	0.836	0.854	0.847	0.962	0.921	0.941	0.942	0.845	0.785	0.814	0.810	0.799	0.884	0.839	0.843	0.693	0.771	0.730	0.725
	tunnelExit_0_35fps				busyBoulevard				busStation				library				turbulence3			
	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC	Pre	Rec	F1	MCC
GMM	0.269	0.624	0.395	0.378	0.293	0.359	0.323	0.343	0.751	0.620	0.679	0.683	0.859	0.787	0.821	0.833	0.728	0.676	0.701	0.713
T2MRF	0.132	0.099	0.113	0.103	0.417	0.084	0.140	0.146	0.664	0.501	0.571	0.574	0.621	0.319	0.421	0.441	0.607	0.434	0.506	0.499
CB	0.554	0.382	0.452	0.456	0.406	0.272	0.326	0.340	0.553	0.732	0.630	0.601	0.896	0.697	0.784	0.773	0.791	0.633	0.703	0.721
PBAS	0.669	0.421	0.517	0.536	0.323	0.391	0.354	0.350	0.655	0.790	0.716	0.725	0.593	0.393	0.473	0.490	0.807	0.681	0.738	0.745
VUMETER	0.525	0.335	0.409	0.414	0.380	0.311	0.342	0.367	0.513	0.854	0.641	0.640	0.768	0.340	0.471	0.472	0.743	0.718	0.730	0.751
LOBSTER	0.541	0.369	0.438	0.438	0.491	0.443	0.466	0.478	0.765	0.850	0.805	0.792	0.857	0.690	0.764	0.744	0.837	0.729	0.778	0.792
ViBe	0.481	0.348	0.403	0.405	0.430	0.376	0.401	0.429	0.828	0.707	0.763	0.782	0.660	0.410	0.506	0.526	0.770	0.678	0.721	0.732
ViBe+	0.535	0.387	0.449	0.458	0.402	0.544	0.462	0.452	0.908	0.704	0.793	0.805	0.702	0.530	0.604	0.612	0.766	0.682	0.722	0.721
SuBSENSE	0.747	0.628	0.681	0.680	0.575	0.471	0.518	0.523	0.782	0.952	0.859	0.848	0.901	0.719	0.800	0.787	0.842	0.749	0.793	0.790
Zhong17	0.605	0.429	0.502	0.503	0.447	0.457	0.452	0.450	0.756	0.684	0.718	0.716	0.618	0.550	0.582	0.585	0.789	0.642	0.708	0.705
FCDH	0.622	0.593	0.607	0.601	0.392	0.436	0.413	0.408	0.730	0.660	0.693	0.686	0.781	0.736	0.758	0.751	0.782	0.686	0.731	0.734
ViBeF	0.504	0.370	0.427	0.431	0.273	0.321	0.295	0.295	0.784	0.760	0.772	0.769	0.758	0.731	0.744	0.742	0.794	0.744	0.768	0.770
RSBC	0.665	0.625	0.644	0.629	0.463	0.619	0.530	0.524	0.795	0.934	0.859	0.850	0.801	0.950	0.869	0.853	0.860	0.723	0.787	0.781

6.1. Qualitative evaluations

To visually appreciate the performance of the proposed RSBC method against that of the other state-of-the-art methods, in Fig. 8, we present some binary classified results from various categories of the CDnet 2014 dataset. The figure shows a one frame from original videos (Ori) in row 1, the ground truth (GT) images in row 2 and output images generated by applying various BS methods from row 3 to row 13. In the output images, white and black colours represent true positives and true negatives, whereas green and red colours indicate the false negatives and false positives, respectively.

Detailed qualitative observations: In Fig. 8, the *blizzard* video scene in the first column, where cars are moving in a low-visibility winter storm environment, a few of the methods including PBAS, SuBSENSE, ViBeF and RSBC show better performance over others. The *tunnelExit_0_35fps* video scene in the sixth column, shows ghost effect due to the sudden movement of a parked car which was stationary in the scene for a long time. In this scene, the methods GMM, LOBSTER, SuBSENSE, FCDH and ViBeF produce many false positives in the region where the car had been parked. RSBC can properly identify the ghost region and produces the best visual result. Due to camera jitter, in the *badminton* video (third column) most of the methods except SuBSENSE and RSBC produce high numbers of false detections. In the *fall* video scene (fourth column) dynamic background is created by periodic movements of tree branches. In this scene the LOBSTER, FCDH and ViBeF methods show large number of false positives whereas GMM, T2MRF, CB, VUMETER, ViBe, ViBe+, SuBSENSE and Zhong17 produce high false negatives. This type of event can lead many of the existing methods to misclassify the background region. Many BS methods, such as T2MRF, PBAS, VUMETER, ViBe, ViBe+, Zhong17 and ViBeF, incorporate a long time stationary foreground object (the *library* sequence in ninth column) into the background which increases the false negatives. But RSBC is able to identify the foreground object with high accuracy because the ABM component of RSBC is specially equipped to handle such type of dynamic background events. In the *streetLight* sequence in the seventh column, which contains regular movements of cars in a street at night, all the methods produce large classification errors because of sudden illumination changes in the background which none of the existing methods can handle. On the other hand, all the methods show good performance in *highway* (in second column) and *turbulence3* (in last column), but RSBC method produces superior results.

By observing the various visual results, we find that a majority of the state-of-the-art methods cannot efficiently handle either dynamic background or ghost effect. Most of the methods also do not produce consistent performance over various challenging video environments. In general the SuBSENSE and RSBC methods produce consistently better qualitative results compared to others and particularly in the *highway*, *fall*, *tunnelExit_0_35fps* and *busStation* scenes RSBC outperforms SuBSENSE. Therefore, overall it can be claimed that the proposed RSBC method produced better quality visual results than the other algorithms which signifies that our algorithm is robust enough to handle various challenging circumstances.

6.2. Quantitative evaluations

The well performance of the proposed method is confirmed by statistical measurements such as Precision (Pre), Recall (Rec), F-measure (F1 score) and MCC, all of which are given in Table 4. To show the top three performers for various datasets, in the table we use red, green and blue colours to signify the best, the second best and the third best measurements, respectively. This table is also available at https://www.isical.ac.in/~sujroy_r/RSBCbs with 4 digit precision results.

For the PBAS and Zhong17 methods, the results are consistent across many video scenes except for the *busyBoulevard*, *tunnelExit_0_35fps* and *library* scenes where they exhibit very poor results. ViBe+ outperforms ViBe and ViBeF in most complex scenes but RSBC shows higher F1 and MCC scores than ViBe+ in all the video scenes. The T2MRF and CB methods produce comparatively lower Recall, F1 and MCC values in all datasets. As the method FCHD does not handle the shadow and ghost effect, in the *busyBoulevard*, *streetLight* and *tunnelExit_0_35fps* video sequences it shows lower scores. Due to the challenging environments like dynamic background in *fall* and ghost effect in *busyBoulevard*, the F1 and MCC scores are small for a majority of the methods but RSBC produces relatively better scores. In the *library* video scene, except for GMM and RSBC, the Recall values are low for all other methods due to inclusion of the foreground into the background. The methods T2MRF, VUMETER and LOBSTER produce inconsistent results across the various video scenes. SuBSENSE is the only method other than RSBC which gives steady F1 and MCC scores.

The statistical measurements in Table 4 show that for many complex video environments the proposed RSBC method reduces the false positives and false negatives to achieve better results compared to the other state-of-the-art methods. Thus we can conclude that the proposed RSBC method is very flexible in that it can efficiently handle many challenging video environments without compromising its performance.

6.3. System settings and execution time

The proposed RSBC method has proven to be efficient in foreground separation in various video sequences with complex environments. Both qualitative and quantitative evaluations highlight the effectiveness of RSBC. We implemented the RSBC algorithm in C++ using the OpenCV library. We executed all the algorithms on an 3.40 GHz Intel i7 CPU with 8 MB cache, 8 GB RAM, running on Linux operating system.

The processing speed of an algorithm must be high for real-time BS. Table 5 shows the processing speed of various algorithm along with the proposed RSBC method in terms of frames per second (FPS). In the table, we found that the RSBC algorithm processed about 85 frames per second on an average over the video sequences taken in this article. ViBe is the only method that processes more frames than RSBC per unit time but the other methods are slower compared to RSBC.

Table 5
Comparison of Average Frames Per Second (FPS).

Methods	GMM	T2MRF	CB	PBAS	VUMETER	LOBSTER	ViBe	ViBe+	SuBSENSE	Zhong17	FCDH	ViBe F	RSBC
Frames per second (FPS)	48.7	10.2	45.8	42.5	35.9	12.1	152	37.7	7.4	32.7	7.2	40.2	85.1

7. Conclusion and future works

In this article we proposed a real-time BS method named RSBC to separate foreground objects from the background in a video scene. The novelty of the proposed method is that it adaptively updates the background model by analyzing the nature of the previous binary classification results and this strategy is applied for the first time in BS literature. The proposed RSBC method detects the foreground objects accurately by being able to simultaneously and efficiently identify dynamic background, ghost effect and long term static foreground. Many state-of-the-art algorithms tend to misclassify static foregrounds in an effort to remove ghost effect, while other algorithms that correctly identify static foreground cannot handle ghost effect. RSBC produces better results in most of the video sequences and requires less time with respect to most of the existing state-of-the-art methods. For this reason a good application for RSBC is as a fast pre-processing step for prioritizing video frames that could be further analyzed by slower but more accurate algorithms.

RSBC also has some limitations where it may produce poor results for the following cases. Firstly the method may produce poor results for video scenes such as traffic on a highway where new foreground objects keep entering and exiting the video frame causing frequent transitions between foreground and background, which misleads the ABM to mistake it for dynamic background. If a large number of traffic passes through the highway the algorithm gradually detects all of the traffic as background instead of foreground. Secondly, since the ABM model updates itself only after every $F_{window} = 50$ number of frames, the algorithm may produce false positives during these $F_{window} = 50$ frames, if the background suddenly changes within this interval. Thirdly the proposed method may produce erroneous results when the background changes irregularly without following any specific nature such as in the case of turbulent motion. Fourthly the proposed method fails for video scenes with camera motions. For this reason RSBC may be applied to static surveillance cameras but cannot be applied to pan-tilt-zoom (PTZ) type of surveillance cameras.

To improve the performance of the proposed method, in future, a few more features will be considered to distinguish dynamic background from recurrently appearing foreground. The proposed method can also be improved by adding more features to handle camouflaged objects and PTZ camera motions. A separate model (along with the existing models) can be used to identify irregular changes like turbulent motion or to identify PTZ camera motions. The execution time could be reduced by combining three colour channels in an efficient way without sacrificing classification accuracy. As the proposed method does not explicitly handle hard shadows, this model needs to be extended to prevent hard shadows from getting classified as foreground. The RSBC algorithm could also be modified to work with a universal set of parameters independent of the type of complexity in the video sequence.

Acknowledgement

The authors would like to thank the Editor-in-Chief, the Associate Editor and the anonymous reviewers for their valuable comments and detailed suggestions for helping to clarify the article's presentation and also improving the contents. Sujoy Madhab Roy gratefully acknowledges Mr. Ashish Bakshi for his careful reading and helpful discussion to improve this article.

References

Barnich, O., & Van Droogenbroeck, M. (2011). ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20, 1709–1724.

- Bouwman, T. (2014). Traditional and recent approaches in background modeling for foreground detection: an overview. *Computer Science Review*, 11, 31–66.
- Bouwman, T., & Zahzah, E. H. (2014). Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122, 22–34.
- Braham, M., & Van Droogenbroeck, M. (2016). Deep background subtraction with scene-specific convolutional neural networks. In *Proceedings of the IEEE international conference on systems, signals and image processing (IWSSIP)* (pp. 1–4). IEEE.
- Brutzer, S., Höferlin, B., & Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1937–1944). IEEE.
- Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58, 11.
- Chen, Y., Wang, J., & Lu, H. (2015). Learning sharable models for robust background subtraction. In *Proceedings of the IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). IEEE.
- Chen, Y.-T., Chen, C.-S., Huang, C.-R., & Hung, Y.-P. (2007). Efficient hierarchical method for background subtraction. *Pattern Recognition*, 40, 2706–2715.
- Davis, J. W., & Sharma, V. (2007). Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding*, 106, 162–182.
- Elgammal, A., Harwood, D., & Davis, L. (2000). Non-parametric model for background subtraction. *Computer Vision—ECCV 2000*, 751–767.
- Goya, Y., Chateau, T., Malaterre, L., & Trassoudaine, L. (2006). Vehicle trajectories evaluation by static video sensors. In *Proceedings of the intelligent transportation systems conference (ITSC), 2006* (pp. 864–869). IEEE.
- Goyette, N., Jodoin, P., Porikli, F., Konrad, J., & Ishwar, P. (2012). Changedetection.net: a new change detection benchmark dataset. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 1–8).
- Guyon, C., Zahzah, E.-H., & Bouwmans, T. (2012). *Robust principal component analysis for background subtraction: systematic evaluation and comparative analysis*. Citeseer.
- Hofmann, M., Tiefenbacher, P., & Rigoll, G. (2012). Background segmentation with feedback: the pixel-based adaptive segmenter. In *Proceedings of the 2012 IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 38–43). IEEE.
- Karasulu, B., & Korukoglu, S. (2012). Moving object detection and tracking by using annealed background subtraction method in videos: performance optimization. *Expert Systems with Applications*, 39, 33–43.
- Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-time Imaging*, 11, 172–185.
- Kutz, J. N., Grosek, J., & Brunton, S. L. (2016). Dynamic mode decomposition for robust PCA with applications to foreground/background subtraction in video streams and multi-resolution analysis. *Handbook on robust low-rank and sparse matrix decomposition: applications in image and video processing*. CRC.
- Mandellos, N. A., Keramitsoglou, I., & Kiranoudis, C. T. (2011). A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications*, 38, 1619–1631.
- Monnet, A., Mittal, A., Paragios, N., & Ramesh, V. (2003). Background modeling and subtraction of dynamic scenes. In *Proceedings of ninth IEEE international conference on computer vision* (pp. 1305–1312). IEEE.
- Nguyen, T. P., Pham, C. C., Ha, S. V.-U., & Jeon, J. W. (2018). Change detection by training a triplet network for motion feature extraction. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Oliver, N. M., Rosario, B., & Pentland, A. P. (2000). A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 831–843.
- Ortego, D., SanMiguel, J. C., & Martínez, J. M. (2018). Hierarchical improvement of foreground segmentation masks in background subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Panda, D. K., & Meher, S. (2016). Detection of moving objects using fuzzy color difference histogram based background subtraction. *IEEE Signal Processing Letters*, 23, 45–49.
- Prasad, D. K., Prasath, C. K., Rajan, D., Rachmawati, L., Rajabally, E., & Quek, C. (2018). Object detection in a maritime environment: Performance evaluation of background subtraction methods. *IEEE Transactions on Intelligent Transportation Systems*.
- Rajagopalan, R. (2014). A genetic algorithm for optimizing background subtraction parameters in computer vision. *British Journal of Applied Science & Technology*, 4, 41–48.
- Roy, S. M., & Ghosh, A. (2018). Real-time adaptive histogram min-max bucket (HMMB) model for background subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 28, 1513–1525.
- Serratos, F., Alquézar, R., & Amézquita, N. (2012). A probabilistic integrated object recognition and tracking framework. *Expert Systems With Applications*, 39, 7302–7318.
- Shen, Y., Hu, W., Yang, M., Liu, J., Wei, B., Lucy, S., & Chou, C. T. (2016). Real-time and robust compressive background subtraction for embedded camera networks. *IEEE Transactions on Mobile Computing*, 15, 406–418.
- Sobral, A., Bouwmans, T., & Zahzah, E.-h. (2016). LRSLibrary: Low-Rank and Sparse tools for Background Modeling and Subtraction in Videos. *Robust low-rank and sparse matrix decomposition: applications in image and video processing*. CRC Press.

- Sobral, A., & Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122, 4–21.
- Spagnolo, P., Leo, M., Distante, A., et al. (2006). Moving object segmentation by background subtraction and temporal analysis. *Image and Vision Computing*, 24, 411–423.
- St-Charles, P.-L., & Bilodeau, G.-A. (2014). Improving background subtraction using local binary similarity patterns. In *Proceedings of the IEEE winter conference on applications of computer vision (WACV)* (pp. 509–515). IEEE.
- St-Charles, P.-L., Bilodeau, G.-A., & Bergevin, R. (2015). SuBSENSE: a universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24, 359–373.
- Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE international conference on computer vision and pattern recognition (CVPR)*: 2 (pp. 246–252). IEEE.
- Szwoch, G., Ellwart, D., & Czyżewski, A. (2016). Parallel implementation of background subtraction algorithms for real-time video processing on a supercomputer platform. *Journal of Real-Time Image Processing*, 11, 111–125.
- Tuzel, O., Porikli, F., & Meer, P. (2005). A Bayesian approach to background modeling. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR'05)-workshops* (p. 58). IEEE.
- Van Droogenbroeck, M., & Paquot, O. (2012). Background subtraction: experiments and improvements for Vibe. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 32–37). IEEE.
- Wang, H., & Shi, L. (2016). Foreground model for background subtraction with blind updating. In *Proceedings of the IEEE international conference on signal and image processing (ICSIP)* (pp. 74–78). IEEE.
- Weng, T. L., Wang, Y. Y., Ho, Z. Y., & Sun, Y. N. (2010). Weather-adaptive flying target detection and tracking from infrared video sequences. *Expert Systems with Applications*, 37, 1666–1675.
- Wren, C., Azarbayejani, A., Darrell, T., & Pentland, A. (1997). Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 780–785.
- Zhao, Z., Bouwmans, T., Zhang, X., & Fang, Y. (2012). A fuzzy background modeling approach for motion detection in dynamic backgrounds. In *Multimedia and signal processing* (pp. 177–185). Springer.
- Zhong, Z., Zhang, B., Lu, G., Zhao, Y., & Xu, Y. (2017). An adaptive background modeling method for foreground segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 18, 1109–1121.
- Zhou, X., Yang, C., Zhao, H., & Yu, W. (2014). Low-rank modeling and its applications in image analysis. *ACM Computing Surveys*, 47, 36:1–36:33.
- Zivkovic, Z., & van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27, 773–780.