J Ambient Intell Human Comput (2024) 15:1213–1227
DOI 10.1007/s12652-017-0480-x

ORIGINAL RESEARCH

Segmenting foreground objects in a multi-modal background using modified *Z*-score

Suman Kumar Choudhury[1] · Pankaj Kumar Sa[1] · Kim-Kwang Raymond Choo[2,3] · Sambit Bakshi[1]

Received: 4 August 2016 / Accepted: 17 March 2017 / Published online: 9 April 2017

Abstract This article presents a background subtraction method to detect moving objects across a stationary camera view. A hybrid pixel representation is presented to minimize the effect of shadow illumination. A non-recursive background model is developed to address the problem with gradual illumination change. Modified Z-score labeling is employed to analyze the sample variation of the temporal sequence to build a multi-modal background. The same measure is further applied to detect the foreground pixels against the stationary background classes. Morphological filtering is employed to suppress the sensor noise as well as to fill the camouflage holes. A decision rule is formulated that considers the period of being stationary of a foreground object and the period of being absence of a background class to tackle the object relocation problem. The proposed approach along with nine other state-of-the-art methods are compared on various image sequences taken from the Wallflower and the I2R datasets in terms of recall, precision, figure of merit, and percentage of correct classification. The tabular results, as well as the obtained figures demonstrate the efficacy of the proposed scheme over its counterparts.

Keywords Video surveillance · Object detection · Background subtraction · Background modeling · Foreground extraction · Modified *Z*-score

✉ Sambit Bakshi
sambitbaksi@gmail.com; bakshisambit@nitrkl.ac.in

Suman Kumar Choudhury
sumanchoudhury.nitr@gmail.com

Pankaj Kumar Sa
pankajksa@nitrkl.ac.in

Kim-Kwang Raymond Choo
raymond.choo@fulbrightmail.org

1 Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, India

2 Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA

3 School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia

1 Introduction

Visual surveillance aims at extracting useful information from an enormous amount of video data by automatically detecting and tracking objects of interest, analyzing their activities, and producing a semantic description (Wang 2013; Doretto et al. 2011; Albano et al. 2014). It has enormous applications both in public and private establishments, such as land security, traffic management, crime prevention, effective decision-making, accident prediction, monitoring threats, and so forth (De Smedt et al. 2014; Varga and Szirányi 2017). A typical surveillance system comprises a set of cameras alongside the connected computers to process and monitor the ongoing activities. In this article, we focus on the very first step of an automated surveillance system i.e. moving object detection.

In the absence of any *a priori* scene knowledge, the most widely used method for moving object detection is background subtraction (Sajid and Cheung 2015). It consists of three steps; background initialization, foreground extraction, and background maintenance (Kumar and Yadav 2017). A model of the observed scene is estimated using few initial frames during background initialization. The subsequent frames are then compared with the modeled background to detect the foreground objects. The next stage is to update the modeled background to adapt any means of changes that may occur in the observed scene. The performance of background subtraction is usually influenced by a number of factors, such as shadow, camouflage, uninteresting background movement, object relocation, and gradual illumination change.

In this article, we propose a background subtraction model to detect the foreground objects in the presence of the above factors. A hybrid color space is suggested for appropriate pixel representation. The sample variation of pixel sequence along the temporal domain is taken into consideration in the modeling of a dynamic background. A non-recursive outlier labeling methodology is adapted to extract the potential foreground, and the frequency of objects' appearance is applied to update the modeled background.

We will briefly discuss related literature in the next section, prior to presenting our proposed scheme in Sect. 3. The simulation results are presented in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Related literature

Precise background modeling and its periodic maintenance are essential to accurate object detection (Choudhury et al. 2016; Goyal and Singhai 2017; Setitra and Larabi 2014; Sobral and Vacavant 2014; Elgammal 2014). We will now describe the various approaches in the literature.

In approaches using single Gaussian and mixture of Gaussians, pixel sequence along the temporal domain is modeled using Gaussian distribution (Wren et al. 1997). Each background location is parameterized by two model parameters, namely: mean μ and variance σ^2, of the temporal sequence. However, single Gaussian fails to capture waving background due to the presence of non-relevant oscillation. Stauffer and Grimson introduced a multi-label background using a mixture of Gaussians (MoG) that classifies the initialization sequence into multiple numbers of Gaussians (Stauffer and Grimson 1999, 2000). Zivkovic and Heijden also suggested an algorithm to compute the correct number of Gaussian distributions at each location, based on their sample variation over time (Zivkovic 2004; Zivkovic and van der Heijden 2006).

In the codebook model, each background location is modeled using a set of codewords such as the minimum and maximum intensity value, the occurrence frequency, the first and last access times to the codeword, and the maximum negative run length time (Kim et al. 2005; Wu and Peng 2010; Fernandez-Sanchez et al. 2013). For example, in Shah et al. (2015), a self-adaptive codebook model is presented where a modified color space is used for pixel representation, and a block-based initialization algorithm uses a self-adaptive algorithm to update the model parameters.

In buffer based subtraction approaches, each background location is modeled using the recent pixel history in a finite buffer. The absolute difference between the current pixel and buffer median decides whether it is stationary (Lo and Velastin 2001). Subsequently, the median measure is replaced by the medoid to represent the background (Cucchiara et al. 2003; Calderara et al. 2006). Toyama *et al.* apply a linear predictive model using Wiener filter (Toyama et al. 1999), where the coefficients are estimated using the sample covariance. Such modeling is further applied in a relevant subspace using principal component analysis (Zhong and Sclaroff 2003). In another work, Wang and Suter developed a model using the notion of consensus to counter the problem with illumination change and background relocation (Wang and Suter 2006, 2007). Buffer based methods generally adapt well to the slow varying illumination at the cost of high memory overhead.

Non-parametric background models do not assume any prior shape distribution, unlike the default Gaussian model (McHugh et al. 2009; Heikkil and Pietikinen 2006). The kernel density estimation (KDE) techniques usually take sufficient temporal sequences to converge to the underlying target distribution. The kernel bandwidth is inversely related to the number of training samples (i.e. a wider bandwidth yields an over-smoothed distribution, whereas a narrow bandwidth leads to a jagged density estimation). Piccardi and Zan estimated the kernel bandwidth as a function of the median of the absolute difference between the successive frames (Elgammal et al. 2000). Subsequently, the mean-shift paradigm is adapted to estimate the underlying distribution using fewer training samples (Piccardi and Jan 2004). In another work, a fast Gauss transform technique was applied to improve the computation burden (Elgammal et al. 2001). Parag et al. proposed a boosting based ensemble learning to select appropriate features for the KDE methods (Parag et al. 2006).

Zhang and Xu apply the fuzzy Sugeno integral to model the observed scene (Hongxun and De 2006). In a separate work, the Sugeno integral is replaced by the Choquet integral to obtain better results (El Baf et al. 2008). In another work, color, texture, and edge features are fused with Choquet integral for object detection task (Azab et al. 2010). Bouwmans et al. applied another type-2 fuzzy model to compute the correct number of background classes to model a multi-modal scene (Bouwmans and El Baf 2009; El Baf et al. 2009). Kim and Kim apply the fuzzy color histogram to model the dynamic background (Kim and Kim

2012). In another work, color difference histogram is first used to capture the multi-modal background, and then the Fuzzy C-means is employed to reduce the large dimensionality of the histogram bins (Panda and Meher 2016).

For learning model approaches, the initialization pixel sequence is trained across a classifier to learn the shape distribution of the underlying background. Culibrk et al. apply a multi-layered feed forward network with 124 neurons (Culibrk et al. 2007). In other words, a probabilistic neural network (PNN) is learned to create the background model, and a Bayesian classifier is applied to separate the moving objects. In another work, a self-organization map network is trained to learn a background location (Maddalena and Petrosino 2008, 2010). They further apply a spatial coherence analysis to reduce the false alarms.

From the existing literature, it is clear that parametric models generally rely heavily on their underlying assumptions and, thereby, limiting the operating flexibility in varying environments. Non-parametric models, on the other hand, accept sufficient training samples to estimate the underlying distribution. The unimodal background fails to model a dynamic scene, whereas the multi-modal background needs to compute the oscillation periodicity for each location. Recursive models fail to tackle the gradual illumination variation, whereas non-recursive models adapt such eventual variations at the price of high memory overhead. Pixel-based schemes compare the pixel values along the temporal axis, whereas the region-based methods compute the sample variation both along the temporal domain and spatial neighborhood.

3 Proposed scheme

In this section, we present a comprehensive background model (hereafter referred to as CBGM) to extract the set of moving components across a stationary field of view. The proposed framework of background subtraction is shown in Fig. 1.

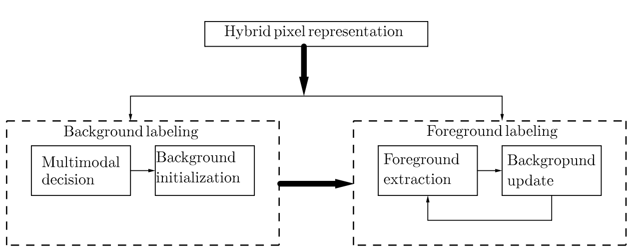


Fig. 1 Framework of the proposed background model

3.1 Hybrid pixel representation

Background models usually suffer from shadow illumination. The underlying region significantly deviates from the modeled background; thereby, falsely appearing as foreground. The light illumination is the primary source of shadow impression, given by

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{pmatrix} \begin{pmatrix} R_o \\ G_o \\ B_o \end{pmatrix} \tag{1}$$

where R_o, G_o, B_o denote the original form of red, green, and blue channel respectively, and α represents the illumination factor. Any invariant form that can nullify the effect of α can be a suitable measure of pixel representation.

In our work, we adapt an invariant color model $c_1c_2c_3$ (Gevers and Smeulders 1999) that depends on the chromatic content only, given by

$$\begin{aligned} c_1 &= \arctan\left(\frac{R}{\max\{G, B\}}\right) \\ c_2 &= \arctan\left(\frac{G}{\max\{R, B\}}\right) \\ c_3 &= \arctan\left(\frac{B}{\max\{R, G\}}\right) \end{aligned} \tag{2}$$

The division operation in Eq. (2) negates the illumination factor; thus, minimizing the shadow effect. However, the $c_1c_2c_3$ model remains in-determinant across the achromatic axis, more generally for low R, G, B pixel values. Therefore, it is necessary to store the original observed intensities to represent pixels across the achromatic axis. Accordingly, we express a pixel value (say q) in a hybrid color space, given by

$$q = \begin{cases} (c_1, c_2, c_3) & \text{if } |R-G| + |G-B| + |R-B| < \beta \\ (R, G, B) & \text{otherwise} \end{cases} \tag{3}$$

We set $\beta = 30$ such that the sum of absolute difference between each pair of RGB channel with less than 30 unit would represent the achromatic axis.

3.2 Multi-modal decision in a dynamic background

Waving of leaves, water flow in fountain, and fluttering of flags, are few real-world examples that can be considered as non-relevant (or uninteresting) movements. A Unimodal system is not capable of addressing the problem with such background oscillation.

We consider few initial training samples (say N), $\mathbf{Q} = \{q_1, q_2, \ldots, q_N\}$ at each location to model the background. Let $\mathbf{L} = \{l_1, l_2, \ldots, l_K\}$ be the required K background classes for each model location due to background oscillation ($K < N$). It can be realized that the oscillation periodicity may not be uniform across the background.

Therefore, the value of K may vary across model location. We devise a procedure to compute a *least background separation threshold* (LBST) to determine the required number of background classes each waving location should have.

The initialization sequence $\mathbf{Q}_{xy}$ may contain *RGB* values as well as $c_1c_2c_3$ values. Accordingly, we need to compute six *Least Background Separation Threshold* (*LBST*) across both color space and for each color channel: $\tau_{xy}^{c_1}, \tau_{xy}^{c_2}, \tau_{xy}^{c_3}, \tau_{xy}^{R}, \tau_{xy}^{G}, \tau_{xy}^{B}$. The estimated thresholds are then applied during background initialization phase (see Algorithm 1 in Sect. 3.3) that evidently distributes the input pixel sequence to the correct number of background classes. Modified *Z*-score labeling is applied during threshold estimation. This measure is again applied during foreground extraction phase (Sect. 3.4), where the reasoning in selecting various parameters is elaborated. The details of *LBST* computation are presented in Fig. 2. Also, in Table 1, the first column reflects the steps numbering as shown in Fig. 2, and the second column describes the significance of the corresponding steps.

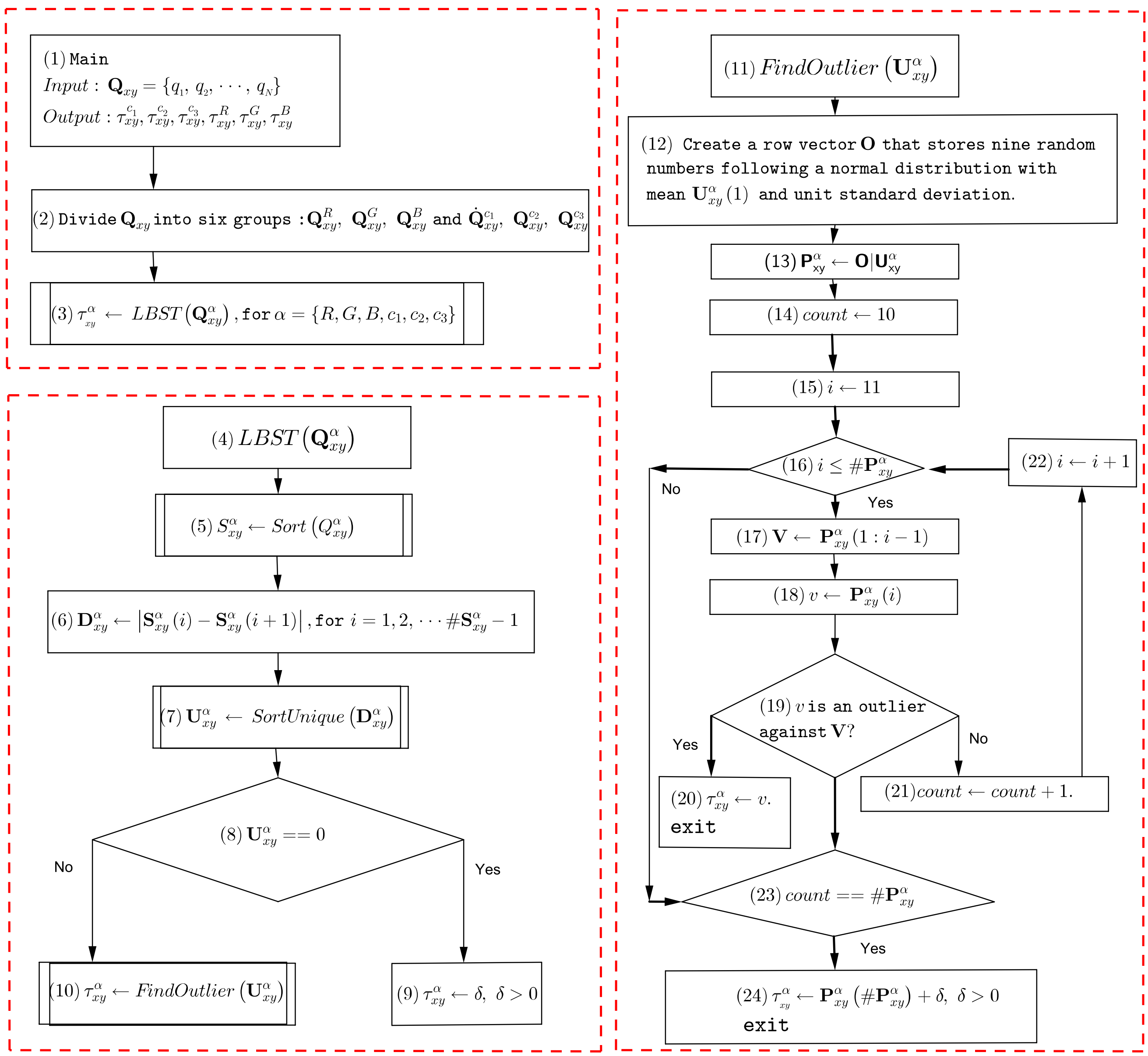


Fig. 2 Flowchart: computing least background separation threshold

Table 1 First column: flowchart step number (as shown in Fig. 2); second column: explanation for the corresponding step

Flowchart no.	Remarks
(5)	$\mathbf{S}^{\alpha}_{xy}$ arranges input sequence $\mathbf{Q}^{\alpha}_{xy}$ in ascending order so that close pixel values appear consecutively
(6)	$\mathbf{D}^{\alpha}_{xy}$ contains the absolute difference between each neighboring pair of $\mathbf{S}^{\alpha}_{xy}$. $\#\mathbf{S}^{\alpha}_{xy}$: cardinality of $\mathbf{S}^{\alpha}_{xy}$
(7)	First extract the unique elements (removing the duplicate values) from $\mathbf{D}^{\alpha}_{xy}$, and then sort (in ascending order) them in another vector $\mathbf{U}^{\alpha}_{xy}$ $\Rightarrow$ demonstrates the sample variation sequence of input vector $\mathbf{Q}^{\alpha}_{xy}$
(8), (9)	Condition ($\mathbf{U}^{\alpha}_{xy} == 0$): This condition arises, when all pixel values in $\mathbf{Q}^{\alpha}_{xy}$ are equal. In other words, it demonstrates a scenario of zero variation across pixel sequence. In this case, the input pixels with same intensities should be stored in single background class. Accordingly, we can set any positive number (>0) as the *LBST* for such location
(8), (10)	Condition ($\mathbf{U}^{\alpha}_{xy} \neq 0$): This situation arises, when the sample variation is not zero. It can be observed that the deviation becomes more from left to right end in $\mathbf{U}^{\alpha}_{xy}$. Moreover, it can be realized that the first element of the sample variation sequence, i.e, $\mathbf{U}^{\alpha}_{xy}(1)$, is certainly the pixel difference of two close background values. However, any of the rest values in the residual array, i.e $\mathbf{U}^{\alpha}_{xy} - \{\mathbf{U}^{\alpha}_{xy}(1)\}$, may represent the desired *LBST*. It may so happen that this residual set may contain few inliers (difference of two close background pixels) along with the outliers (significant pixel difference between two background values). The inliers obviously lie left to the outliers, because $\mathbf{U}^{\alpha}_{xy}$ is in sorted order. it can be realized that the smallest outlier in the residual array $\mathbf{U}^{\alpha}_{xy} - \{\mathbf{U}^{\alpha}_{xy}(1)\}$ represents the required least background separation threshold
(12)	**O**represents an 1-D array with no outliers; mean $= \{\mathbf{U}^{\alpha}_{xy}(1)\}$ and standard deviation $= 0.1$
(13)	$\mathbf{P}^{\alpha}_{xy}$ is a concatenation of **O** followed by $\mathbf{U}^{\alpha}_{xy}$; no outlier among the first ten values in $\mathbf{Q}^{\alpha}_{xy}$ (nine elements of **O** and $\mathbf{U}^{\alpha}_{xy}(1)$)
(15) to (21)	These steps demonstrate the iterative checking of each $\mathbf{Q}^{\alpha}_{xy}(i)$, $i \geq 11$, from left to right, to find the first (smallest) desired outlier
(19), (20)	Condition (*Is v an outlier with respect to the inlier array* **V**): modified *Z*-score is applied to check whether new pixel *v* is an outlier against the inlier array **V**. If yes, declare *v* as the *least background separation threshold* (*LBST*). Otherwise, include *v* in **V** and repeat the procedure until the last element of $\mathbf{P}^{\alpha}_{xy}$ is encountered
(23), (24)	Condition ($count == \#\mathbf{P}^{\alpha}_{xy}$): this situation holds when the sample variation sequence $\mathbf{U}^{\emptyset}_{xy}$ has no outliers; the sample variation is not exactly zero, however, it is not that significant yet. Only one background class is required to store the entire initialization sequence; the *LBST* should be greater than the maximum sample deviation of the sequence ($LBST > \mathbf{P}^{\alpha}_{xy}(\#\mathbf{P}^{\alpha}_{xy})$)

3.3 Background initialization

The strength of background initialization is relative to its modeled parameters. Most existing schemes consider *mean*(μ) and *standard deviation*(σ) of the temporal sequence to model the background. These two parameters are recursively updated for newly identified background pixels over the frame axis. It can be realized that such recursive update may yield biased model parameters over a longer period (Szwoch et al. 2016). In other words, the model attributes may include the distant past pixel contribution and become skewed towards old observations. Furthermore, recursive parameters (μ and σ^2) are very sensitive to the inclusion of even single exception. As a consequence, a newly declared stationary pixel may appear as an outlier against the biased distribution. In this paper, we suggest a non-recursive background model, based on non-recursive model parameters *median* and *MAD* (median of absolute deviation from median) that stores only the recently accessed background pixels in a finite queue to classify the next pixel.

Each background class l_j contains the recently accessed background pixels in a finite queue (say length W) along with one additional attribute, namely: occurrence frequency of the class f_j. A new pixel is compared with the available background classes at the respective location to find its "belongingness", if any. In the event of a no match, the test pixel is enqueued as the first element in a new background class at the respective location. Background initialization is described in Algorithm 1.

Algorithm 1: Background modeling

```
input  : Q_xy = {q_1, q_2, ..., q_N}. Least background
         separation threshold (LBST) for each color channel
         (x, y): τ_xy^c1, τ_xy^c2, τ_xy^c3, τ_xy^R, τ_xy^G, τ_xy^B.
output: Background model M_xy
Function
InitializeBackground(Q_xy, τ_xy^c1, τ_xy^c2, τ_xy^c3, τ_xy^R, τ_xy^G, τ_xy^B)
    K ← 0;/* Initially no background class                          */
    for t ← 1 to N do
        if K ≠ 0 then
            if q_t is a c1c2c3 triplet then
                /* (q_t = ⟨q_t^c1, q_t^c2, q_t^c3⟩)                     */
                /* find the pixels's belongingness with
                available c1c2c3 classes only.                       */
                Find a class l_m = ⟨l_m^c1, l_m^c2, l_m^c3⟩, where
                |q_t^c1 − median(l_m^c1)| < τ_xy^c1 and
                |q_t^c2 − median(l_m^c2)| < τ_xy^c2 and
                |q_t^c3 − median(l_m^c3)| < τ_xy^c3;
            else
                /* find the pixels's belongingness with
                available RGB classes only.                          */
                /* (q_t = ⟨q_t^R, q_t^G, q_t^B⟩)                        */
                Find a class l_m, such that
                |q_t^R − median(l_m^R)| < τ_xy^R and
                |q_t^G − median(l_m^G)| < τ_xy^G and
                |q_t^B − median(l_m^B)| < τ_xy^B;
        if K = 0 or no matching class found then
            K ← K + 1;
            insert(l_K, q_t);
            /* insert q_t as first element in l_K.                  */
            f_K ← 1;
        else
            // update the model parameters of the matched
            class l_m.
            f_m ← f_m + 1;
            insert (l_m, q_t);
            /* insert q_t at rear end of l_m                       */.
            if #l_m > W then
                /* #l_m is the cardinality of l_m in terms of
                number of pixel values                           */
                delete (l_m); // Overflow condition; remove the
                front end element from l_m.
    return;
```

3.4 Foreground extraction

Moving objects significantly differ from the modeled background in terms of visual appearance. In particular, a foreground behaves as an outlier with respect to all background classes available at the corresponding location. Existing methods, based on recursive model parameters mean μ and standard deviation σ, compute the Z-score to separate the foreground objects (Stauffer and Grimson 1999, 2000). Usually, the absolute value of Z-score ($Z_{\mu,\sigma} = \frac{q-\mu}{\sigma}$, q being the current pixel) that exceeds an empirical threshold 2.5 is declared as foreground. However, it has been observed that the mean and standard deviation of a sequence often become inflated by a few or even a single extreme value(s). It may so happen that the less extreme outliers may remain undetected in the presence of the most extreme outlier and vice versa. In our work, this issue is resolved by using the modified Z-score ($Z_{\text{md,MAD}}$), which is based on median (md) and median of the absolute deviation around median (MAD); median and MAD are directly relative to the number of samples in the observed set rather than the sample itself. Seo has elaborated in his work the superiority of modified Z-score over traditional Z-score (Seo 2006).

If $\mathbf{A} = \{a_1, a_2, \ldots, a_n\}$ represents a sequence of observations, then

$$\text{MAD}(\mathbf{A}) = \text{median}(|\mathbf{A} - \text{median}(\mathbf{A})|) \tag{4}$$

The modified Z-score ($Z_{\text{md,MAD}}$) for a new observation (say a_{new}) with respect to sequence $\mathbf{A}$ is expressed as,

$$Z_{\text{md,MAD}} = \frac{0.6745 \times \left(a_{new} - \text{median}(\mathbf{A})\right)}{\text{MAD}(\mathbf{A})} \tag{5}$$

The MAD approximates 0.6745 times the standard deviation for pseudo-normal observations (Leys et al. 2013).

The use of modified Z-score in foreground labeling requires the knowledge of (1) an empirical threshold that separates a foreground pixel against a background class, and (2) the length of the temporal queue W (maximum size of a background class) based on which the threshold is computed. Iglewicz and Hoaglin (1993) empirically evaluated that a sample for which $|Z_{\text{md,MAD}}| > 3.5$ is labeled as a potential outlier against a sequence of pseudo-normal samples of size ranging from 10 to 40. Accordingly, we take the foreground-labeling threshold = 3.5, and length of temporal queue (W) = 25.

A new pixel q_{xy} at (x, y) may represent either a $(c_1c_2c_3)$ or a (R, G, B) triplet (see Eq. 3). Furthermore, the background model $\mathcal{M}_{xy}$ at (x, y) may contain a mixture of c_1, c_2, c_3 classes as well as R, G, B classes. All these instances need to be taken into consideration when preparing a decision rule (Table 2) for foreground extraction.

3.4.1 *Background update*

Background model might change after initialization with object relocation. An existing background object can either be relocated to another location within the camera view or taken away from the observed view. Similarly, a new background object may be introduced in the view. Such dynamic behavior of the background objects demands an update strategy to relabel the changed location as background. Furthermore, the rapid illumination variation, such as cloud movements or lights ON/OFF events, completely alters the appearance of the observed scene. It may so happen that the entire frame may be significantly deviated from the modeled background, and thereby appear as a single foreground object. In our work,

Table 2 Decision rule for foreground extraction

q_{xy}	$\mathcal{M}_{xy}$	Rule base
$c_1c_2c_3$	*RGB* classes only	Declare q_{xy} as foreground.
RGB	$c_1c_2c_3$ classes only	
RGB	Both *RGB* and $c_1c_2c_3$ classes *RGB* classes only	Apply modified Z-score to 1) find a matching class for the pixel R against the available R classes at $\mathcal{M}_{xy}$ **If** found, $signal_1 = 1$; **Else**, $signal_1 = 0$. 2) find a matching class for the pixel G against the available G classes at $\mathcal{M}_{xy}$ **If** found, $signal_2 = 1$; **Else**, $signal_2 = 0$. 3) find a matching class for the pixel B against the available B classes at $\mathcal{M}_{xy}$ If found, $signal_3 = 1$; **Else**, $signal_3 = 0$ **If** $signal_1 == 1$ and $signal_2 == 1$ and $signal_3 == 1$, Declare q_{xy} as background **Else** Declare q_{xy} as foreground
$c_1c_2c_3$	$c_1c_2c_3$ classes only Both *RGB* and $c_1c_2c_3$ classes	Apply modified Z-score to 1) find a matching class for the pixel c_1 against the available c_1 classes at $\mathcal{M}_{xy}$ **If** found, $signal_1 = 1$; **Else**, $signal_1 = 0$ 2) find a matching class for the pixel c_2 against the available c_2 classes at $\mathcal{M}_{xy}$ **If** found, $signal_2 = 1$; **Else**, $signal_2 = 0$ 3) find a matching class for the pixel c_3 against the available c_3 classes at $\mathcal{M}_{xy}$ If found, $signal_3 = 1$; **Else**, $signal_3 = 0$ **If** $signal_1 == 1$ and $signal_2 == 1$ and $signal_3 == 1$, Declare q_{xy} as background **Else** Declare q_{xy} as foreground

the frequency-rate of appearance is suitably formulated to relabel the background with the advent of any of the above situations.

A foreground model $\mathcal{H}$ is designed, following the similar architecture in congruent to background model $\mathcal{M}$ that stores the labeled foreground pixels along with other model attributes. The occurrence frequency of a relocated or new background object should be high enough to ensure its relabeling as background again. On the contrary, once an existing background object is removed from the underlying scene, its occurrence frequency will no longer increment with time. These two scenarios are taken into consideration to govern a decision rule for background update, given below:

Step-1: For a new pixel q_{xy} at (x, y),

1. If q_{xy} is declared as background, enqueue q_{xy} in the matched background class following the steps (14) through (17) of Algorithm 1.
2. If q_{xy} is declared as foreground, find a matching foreground class at $\mathcal{H}_{xy}$, and update it. For no match, create a new foreground class at $\mathcal{H}_{xy}$ and enqueue q_{xy}.

Step-2: Update $\mathcal{M}$ and $\mathcal{H}$, as given below.

1. Remove the high frequency classes from $\mathcal{H}_{xy}$ and add to $\mathcal{M}_{xy}$

$$\mathcal{M}_{xy} \leftarrow \mathcal{M}_{xy} + \left\{ c_j | c_j \in \mathcal{S}_{xy}, \quad f_j \geq \frac{N}{2} \right\}$$

$$\mathcal{H}_{xy} \leftarrow \mathcal{H}_{xy} - \left\{ c_j | c_j \in \mathcal{H}_{xy}, \quad f_j \geq \frac{N}{2} \right\}$$

2. Remove the background classes that have not been accessed for a defined period from $\mathcal{M}_{xy}$. $\mathcal{M}_{xy} \leftarrow \mathcal{M}_{xy} - \left\{ c_j | c_j \in \mathcal{M}_{xy}, f_j < \frac{t-(N-1)}{2} \right\}$, where t represents the current frame number.

3.5 Morphological refinement

The background noise often leads to some false positives as well as false negatives during foreground extraction. Moreover, parts of the foreground may look identical to that of the underlying background with the same chromatic content, and thereby may possess holes inside the detected object. We apply three morphological filters (square structuring element, size 5×5) to suppress such false alarms.

(a) Morphological opening: to reduce the noise and scattered error pixels.
(b) Morphological closing: to join the disconnected foreground pixels.
(c) Morphological filling: to fill the camouflage holes surrounded by foreground pixels.

4 Simulation results

The proposed model along with some state-of-the-art methods are evaluated using exhaustive simulations on several image sequences. The obtained results are then analyzed in subsequent paragraphs. Prior to this, we briefly describe the benchmark datasets and performance metrics used in our simulation.

4.1 Datasets used

Datasets along with the ground-truth annotations are essential for qualitative as well as quantitative analysis of any algorithm. In the present work, eight video clips from Wallflower (Toyama et al. 1999) and I2R (Li et al. 2003) datasets are used for simulation. Each of these image sequences portrays a typical scenario of video surveillance application. The details of the simulated image sequences along with the associated challenges are described in Table 3.

TimeOfDay: The gradual variation of sunlight illumination across a day is depicted in the *TimeOfDay* sequence. The video shows a relatively dark empty room being brightened gradually and revealing the various objects present in it. Towards the end, a man enters and occupies a couch.

MovedObject: A man enters the room, displaces the chair and phone (background objects) from their original locations, and leaves.

WavingTrees: The waving tree needs to be incorporated into a multi-modal background.

Table 3 Simulated videos and associated challenges

Dataset	Video	Challenge(s) associated
Wallflower	TimeOfDay	Gradual illumination variation
	MovedObject	Object relocation
	WavingTrees	Background osciillation
	Camouflage	Camouflage
I2R	Campus	Background osciillation, shadow
	Hall	Shadow
	Fountain	Background oscillation
	Curtain	Background oscillation, camouflage

Camouflage: A man is walking across the computer monitor. The color of the person's shirt matches with the rolling interference bars on the computer screen. Besides, the foreground object casts shadow on the side wall.

Campus: An outdoor scene wherein a number of objects move on the road in presence of waving leaves.

Hall: The pedestrian movement can be observed from the very first frame of the sequence. In addition, the moving objects cast shadow on the ground surface.

Fountain: The background motion owing to water flow has to be omitted while identifying the true mobile objects under consideration.

Curtain: This video clip portrays the problem of (1) background oscillation owing to the curtain movement, and (2) camouflage due to the chromatic similarity between the underlying background and the mobile foreground.

4.2 Comparative analysis

The proposed method is compared with few state-of-the-art schemes: improved adaptive Gaussian mixture model (IGMM, Zivkovic 2004), Bayesian modeling of dynamic scenes (BMOD, Sheikh and Shah 2005), self organizing background subtraction (SOBS, Maddalena and Petrosino 2008), fuzzy spatial coherence based foreground separation (SOBSCH, Maddalena and Petrosino 2010), two variants of ViBe Barnich and Van Droogenbroeck 2011 (i.e. ViBeR - based on RGB color space, and ViBeG - based on gray color space), intensity range based background subtraction (LIBS, Hati et al. 2013), block-based classifier cascade with probabilistic decision integration (BCCPDI, Reddy et al. 2013), and incremental and multi-feature tensor subspace learning (IMTSL, Sobral et al. 2014).

Background subtraction is a binary classification task in which each pixel of an incoming frame is either labeled as stationary or non-stationary. The following parameters (Table 4), in the form of a confusion matrix, are usually taken into consideration to check the efficacy of any classification model.

A set of four performance metrics, derived from the above parameters, is selected to evaluate the proposed framework.

PCC or percentage of correct classification, defines the percentage of correctly detected pixels over the frame resolution.

$$PCC = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{6}$$

Table 4 Confusion matrix for background subtraction

	Actual value (Ground truth fact)	
Predicted value (Predicted by simulation)	Foreground	Background
Foreground	TP (True Positive)	FP (False Positive)
Background	FN (False Negative)	TN (True Negative)

Recall outputs the proportion of detected true positives as compared to the total number of foreground pixels present in the ground-truth.

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{7}$$

Precision measures the ratio of number of detected true positives to the total number of foreground pixels detected by an algorithm.

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{8}$$

F_1 **− Score** , also known as figure of merit, considers both Precision and Recall to compute the score. Higher the score, better is the algorithm.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

In few scenarios, the ground-truth does not have any foreground pixel. As a result, the count of TP, as well as FN, become zero, which in turn lead to erratic behavior of recall and precision. The recall rate, as expressed in Eq. (7), yields the in-determinant $\frac{0}{0}$ form. On the other hand, the precision rate [see Eq. (8)] results in zero with non-zero false positives. In the present simulation, the above behavior is experienced only in *MovedObject* sequence, where we add 1 (a small quantity) to both the numerator and denominator of Eqs. (7) and (8), respectively. The so formed recall rate, in *MovedObject* sequence, becomes 100% for all simulated methods since all of them have successfully identified zero TP. On the other hand, the modified precision rate is a function of the number of false positives detected by an algorithm.

The proposed scheme, along with state-of-the-art methods, are simulated on the benchmark image sequences listed in Sect. 4.1. A comparative summary for performance (w.r.t. *PCC, Recall, Precision,* and *Figure of merit*) is presented in Tables 5, 6, 7, and 8. The average performance for each metric is presented in Fig. 3. In addition, the obtained binary images are depicted in Figs. 4, 5, 6, 7, 8, 9, 10 and 11.

The average recognition rate, as shown in Fig. 3, yields the following observations. Eight of the ten methods have at least 90% correct classification rate (PCC rate). BMOD, ViBeR, ViBeG, and IMTSL have high recall rate but low precision rate. BCCPDI and the proposed CBGM, on the other hand, yield satisfactory recall and precision rate. In terms of F_1 metric, CBGM, BCCPDI, and ViBeR are the most promising approaches.

Shadow effects: IGMM, BMOD, SOBS, SOBSCH, and ViBe (both variants) use the luminance measure, and

Table 5 Comparative analysis of PCC

Method	TimeOfDay	MovedObject	WavingTrees	Camouflage	Campus	Hall	Fountain	Curtain
IGMM	96.87	99.88	88.64	74.10	90.81	88.13	95.93	89.32
BMOD	93.98	**100**	98.25	87.32	95.63	92.99	96.63	90.46
SOBS	91.43	92.81	90.14	87.38	87.63	89.63	94.51	**96.98**
SOBSCH	37.83	92.70	91.69	89.05	87.81	89.49	94.85	96.78
ViBeRGB	94.62	96.71	97.05	89.79	95.46	94.89	96.37	95.84
ViBeGray	94.61	97.96	83.92	89.62	94.79	94.95	96.41	95.64
LIBS	15.64	93.85	85.42	89.11	93.58	92.47	95.64	94.65
BCCPDI	85.39	91.38	96.44	85.29	98.10	90.77	97.59	98.92
IMTSL	94.46	99.94	83.60	91.51	95.09	93.48	96.58	92.83
CBGM	**97.32**	**100**	**99.05**	**95.85**	**98.62**	**95.62**	**98.78**	96.45

Table 6 Comparative analysis of recall

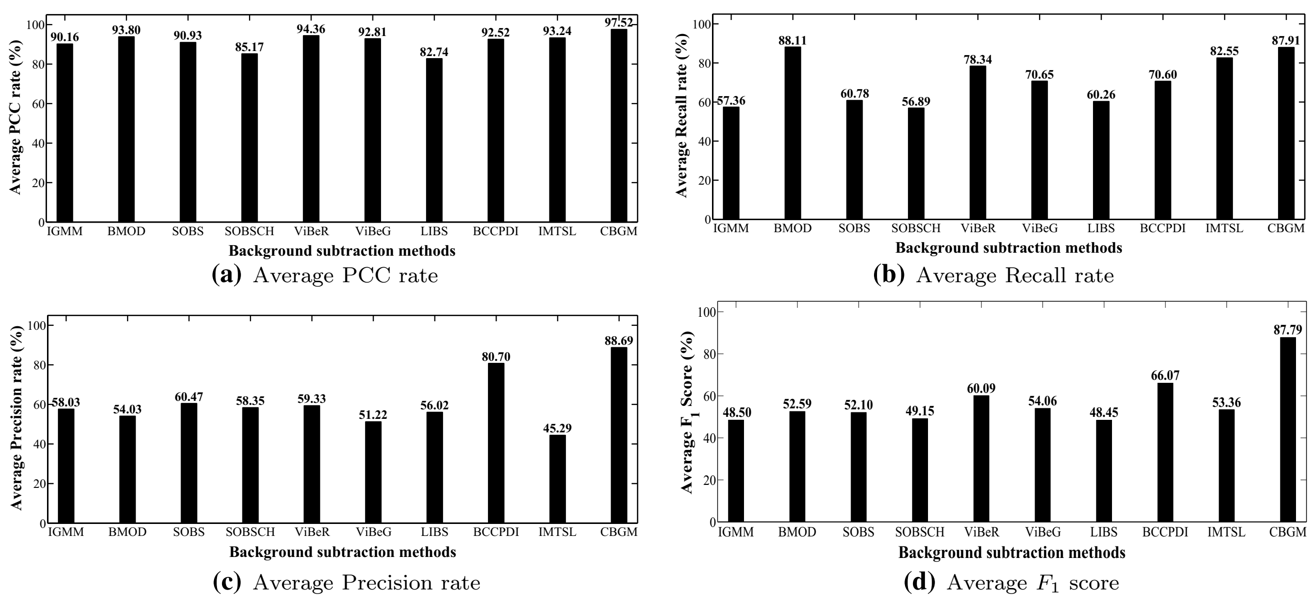
Method	TimeOfDay	MovedObject	WavingTrees	Camouflage	Campus	Hall	Fountain	Curtain
IGMM	83.83	0	86.47	71.32	32.67	34.64	55.49	34.70
BMOD	**100**	**100**	97.35	83.26	**98.20**	**100**	90.20	**100**
SOBS	36.97	0	75.88	82.24	31.89	23.59	45.63	85.58
SOBSCH	2.21	0	79.09	85.11	31.67	22.55	47.50	84.73
ViBeRGB	99.51	0	94.80	91.27	60.73	69.34	65.31	88.87
ViBeGray	99.51	0	86.31	92.76	57.24	71.85	68.11	89.81
LIBS	6.43	0	86.90	95.01	47.20	53.94	53.71	76.18
BCCPDI	33.07	0	91.58	91.86	77.53	46.52	68.28	95.33
IMTSL	61.90	0	96.14	99.62	69.34	73.48	**98.25**	83.29
CBGM	88.55	**100**	**98.06**	**99.90**	84.05	81.08	82.90	74.76

Table 7 Comparative analysis of precision

Method	TimeOfDay	MovedObject	WavingTrees	Camouflage	Campus	Hall	Fountain	Curtain
IGMM	72.11	**100**	74.52	87.29	51.40	52.96	74.51	12.49
BMOD	19.61	**100**	96.92	95.88	26.93	13.24	32.96	0.81
SOBS	20.51	**100**	**99.35**	**97.83**	**95.16**	12.70	79.02	82.54
SOBSCH	16.90	**100**	99.03	96.69	90.64	12.36	75.74	81.12
ViBeRGB	28.30	**100**	95.59	89.74	67.16	**65.95**	50.87	64.87
ViBeGray	28.16	**100**	56.42	87.69	49.34	61.60	46.57	61.73
LIBS	75.80	**100**	61.64	84.34	66.42	46.51	62.23	64.62
BCCPDI	**92.91**	**100**	97.31	79.94	95.73	95.36	92.53	93.30
IMTSL	67.66	**100**	48.37	84.66	31.20	30.19	28.76	31.88
CBGM	73.71	**100**	98.86	92.43	94.75	59.65	**93.76**	**95.18**

Table 8 Comparative analysis of F_1-score

Method	TimeOfDay	MovedObject	WavingTrees	Camouflage	Campus	Hall	Fountain	Curtain
IGMM	77.53	0	80.05	78.50	39.95	41.89	63.61	18.37
BMOD	32.79	**100**	97.13	89.12	42.27	23.38	48.28	1.61
SOBS	26.39	0	86.04	89.36	47.77	16.51	57.85	**84.03**
SOBSCH	3.91	0	87.95	90.54	46.94	15.97	58.38	82.88
ViBeRGB	44.07	0	95.20	90.50	63.78	67.60	57.19	75.00
ViBeGray	43.90	0	68.23	90.15	53.00	66.33	55.32	73.16
LIBS	11.86	0	72.12	89.36	55.18	49.95	57.66	69.93
BCCPDI	48.78	0	94.36	85.48	85.67	62.53	78.57	94.30
IMTSL	64.65	0	64.36	91.53	43.04	42.80	44.50	46.11
CBGM	**80.46**	**100**	**98.46**	**96.02**	**89.08**	**68.73**	**87.99**	83.74



(a) Average PCC rate **(b)** Average Recall rate **(c)** Average Precision rate **(d)** Average F_1 score

Fig. 3 Average results of simulated algorithms computed across all image sequences

unsurprisingly poor results are observed in the *Camouflage* sequence. LIBS is an exception to this, since this approach first computes the minimum and maximum background intensity at each location. The minimum value is further reduced to accommodate the shadow illumination. This alteration in the range works, however, it

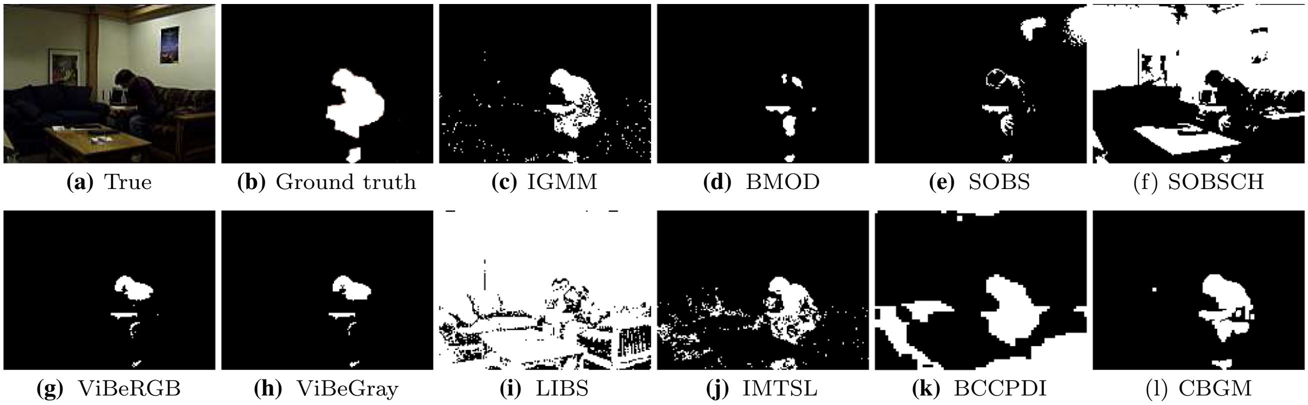

(a) True **(b)** Ground truth **(c)** IGMM **(d)** BMOD **(e)** SOBS (f) SOBSCH
(g) ViBeRGB **(h)** ViBeGray **(i)** LIBS **(j)** IMTSL **(k)** BCCPDI (l) CBGM

Fig. 4 Results of various schemes for the *TimeOfDay* video from Wallflower dataset (Toyama et al. 1999)

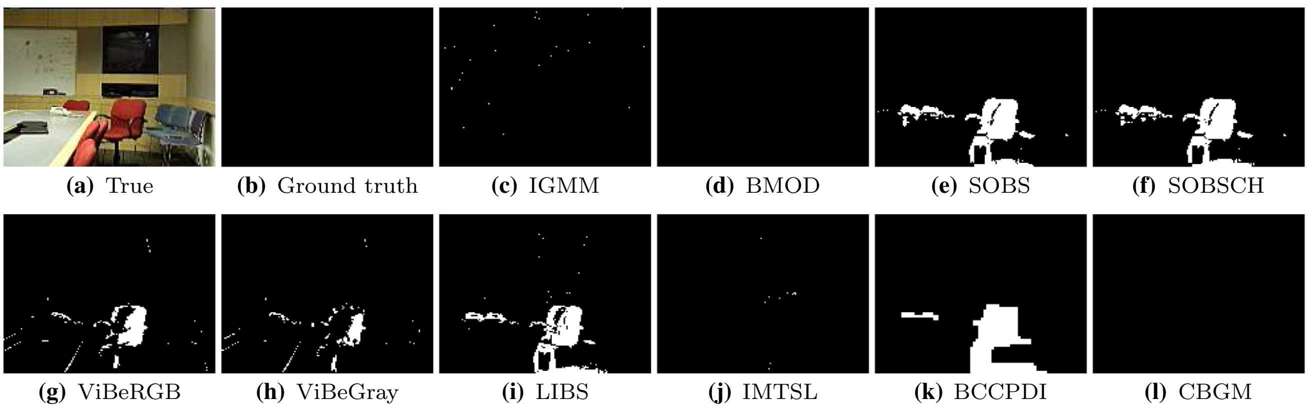

(a) True **(b)** Ground truth **(c)** IGMM **(d)** BMOD **(e)** SOBS **(f)** SOBSCH
(g) ViBeRGB **(h)** ViBeGray **(i)** LIBS **(j)** IMTSL **(k)** BCCPDI **(l)** CBGM

Fig. 5 Results of various schemes for the *MovedObject* video from Wallflower dataset (Toyama et al. 1999)

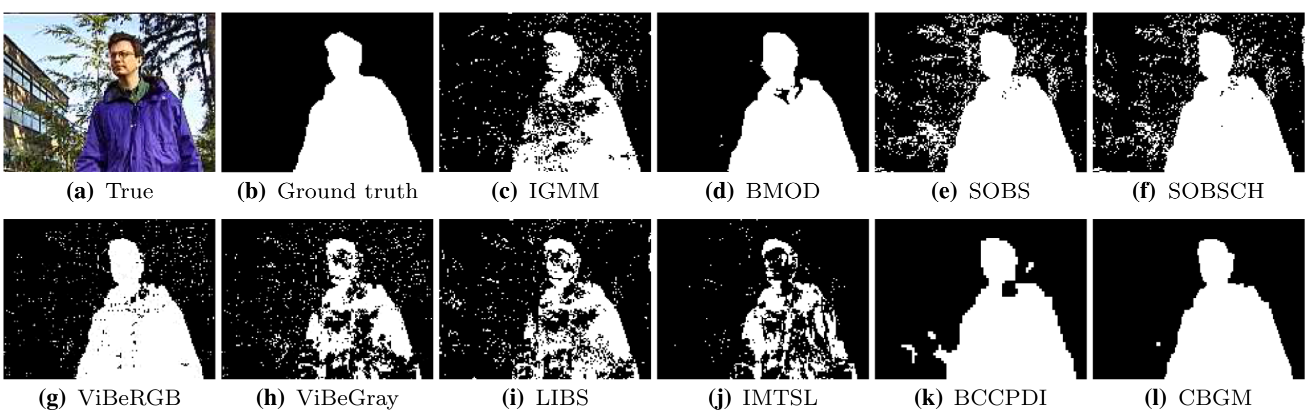

(a) True **(b)** Ground truth **(c)** IGMM **(d)** BMOD **(e)** SOBS **(f)** SOBSCH
(g) ViBeRGB **(h)** ViBeGray **(i)** LIBS **(j)** IMTSL **(k)** BCCPDI **(l)** CBGM

Fig. 6 Results of various schemes for the *WavingTrees* video from Wallflower dataset (Toyama et al. 1999)

is highly parametric and does not perform equally in all scenarios. In our approach, we took the $C_1C_2C_3$ measure, a function of the chromatic content, to represent each pixel, and hence, nullifies the effect of shadow illumination.

Uninteresting movement: *Campus, Fountain, Curtain*, and *WavingTrees* are the sequences, where the possibility of oscillating backgrounds being recognized as foreground objects are high. Most of the methods use a multi-modal background to solve this problem. The least background

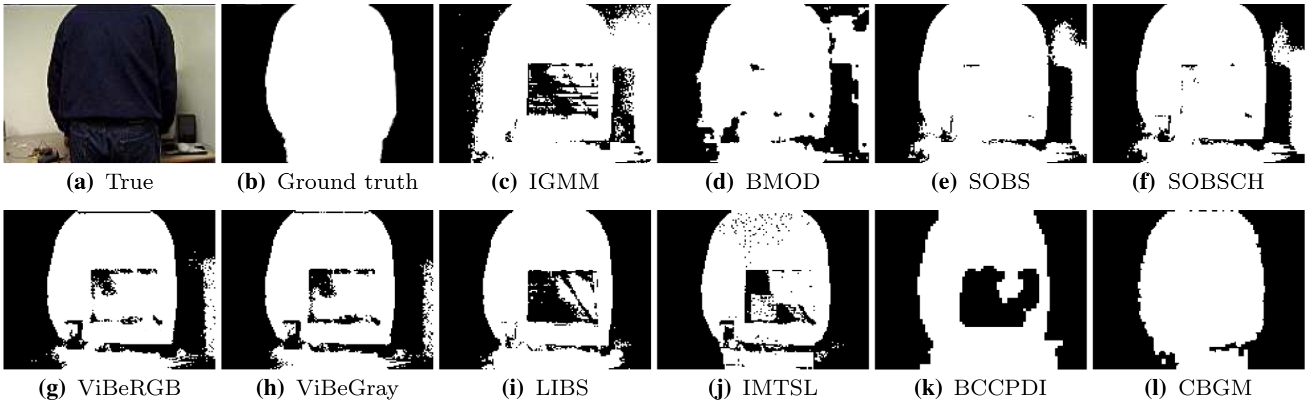


Fig. 7 Results of various schemes for the *Camouflage* video from Wallflower dataset (Toyama et al. 1999)

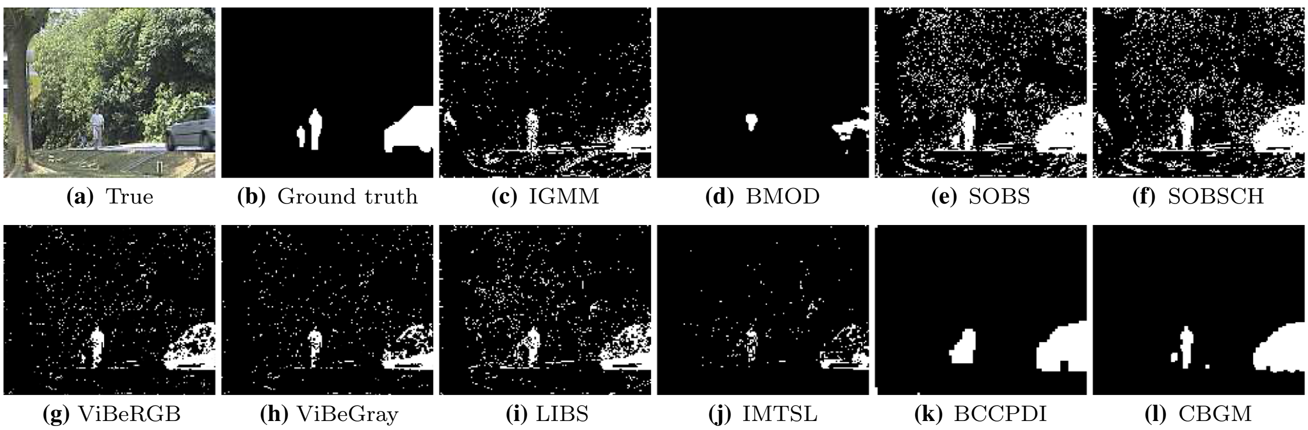


Fig. 8 Results of various schemes for the *Campus* video from I2R dataset (Li et al. 2003)

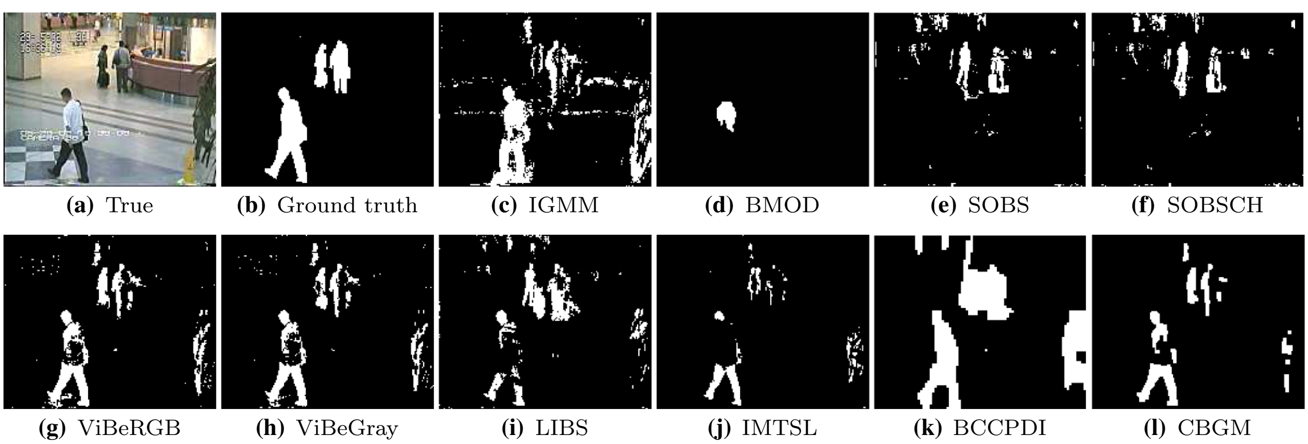


Fig. 9 Results of various schemes for the *Hall* video from I2R dataset (Li et al. 2003)

deviation threshold, in our work, approximates the limit to the number of background classes a model location should have.

Sunlight illumination: Variation of sunlight illumination over a day is a perfect example of gradual illumination variation. As evident from Fig. 4, most of the methods perform moderately in tackling this situation. Recursive models may fail to cope with such eventual variations owing to biased model parameters over a longer surveillance duration. In the present work, we follow a non-recursive architecture, where a background class stores only the recent pixel history in a sliding queue to classify a forthcoming pixel.

Background relocation: The *MovedObject* sequence depicts the background relocation problem as shown in Fig. 5. IGMM, BMOD, IMTSL and CBGM produce a satisfactory result. We approach this problem by using an additional foreground model to store the non-stationary pixels. The high miss-rate of an existing background class and high hit-rate of a foreground class ensure the necessary relabeling.

Running time: The proposed algorithm is executed with MATLAB R2014a on a computer having configuration Intel Core i7 (64-bit, 3.40 GHz), 8 GB RAM. The average running time of each of the methods is listed in Table 9. IGMM, ViBeR, and ViBeG followed by SOBS, SOBSCH have very low processing time. Our proposed scheme consumes around 14 ms per frame that comes next to the above methods. The processing time can be further reduced with codes optimization and parallelization.

Parameter selection: One of the major concerns in background modeling is the choice of initialization

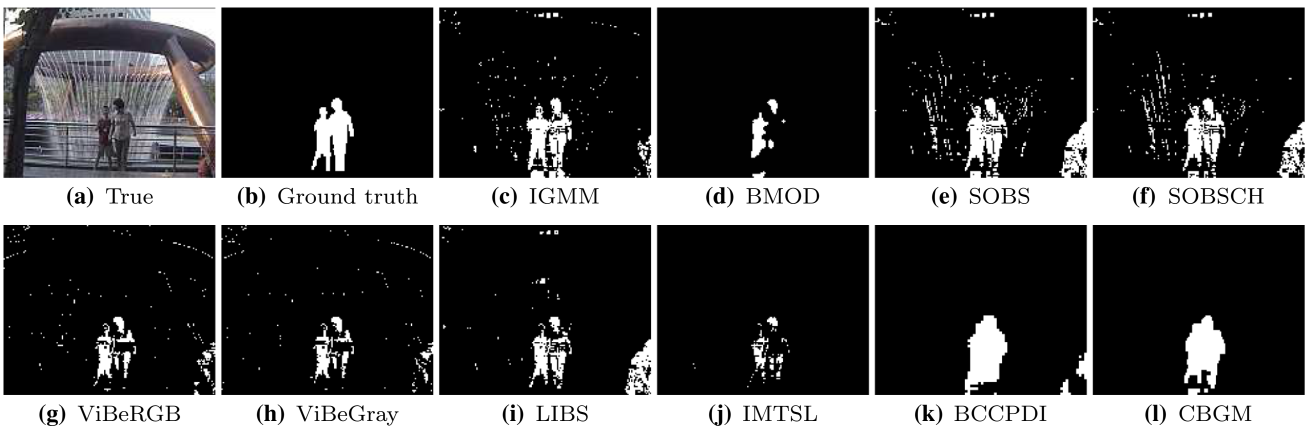


Fig. 10 Results of various schemes for the *Fountain* video from I2R dataset (Li et al. 2003)

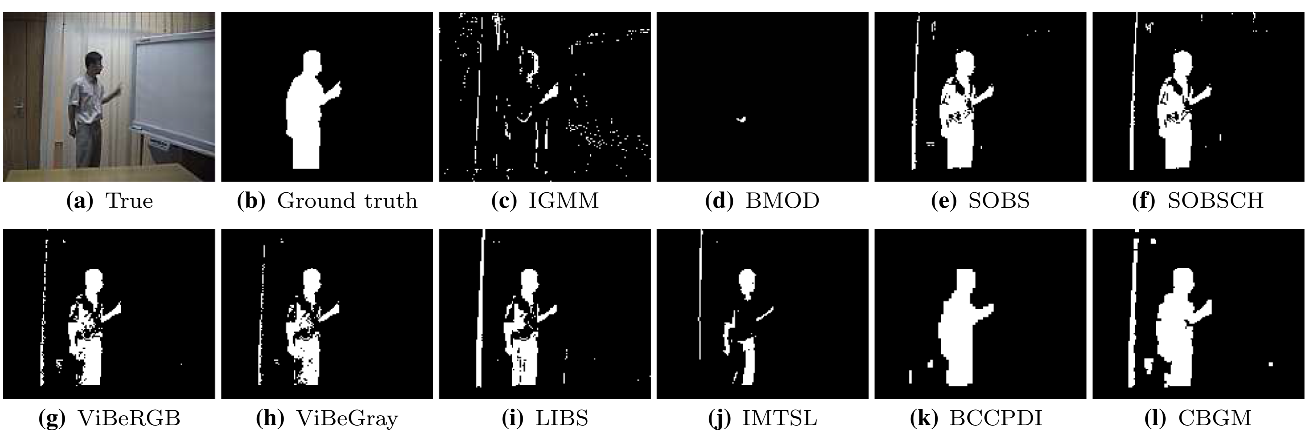


Fig. 11 Results of various schemes for the *Curtain* video from I2R dataset (Li et al. 2003)

Table 9 Average processing time in millisecond per frame

Method	IGMM	BMOD	SOBS	SOBSCH	ViBeR	ViBeG	LIBS	BCCPDI	IMTSL	CBGM
Time (ms/frame)	01	241.33	03.11	03.11	01	01	71.89	269.56	272.56	13.89

frames (N), which is influenced by the following environmental factors. Higher foreground density during initialization requires more number of frames to capture the rearward background locations. The spread of background oscillation is the second factor that is again directly relative to the number of frames enforced to record all possible variations at a pixel location. The third factor is the maximum duration of frames, where a foreground remains stationary during its course of movement. These factors lead to an understanding that suitable choice of N requires some prior knowledge of the scene under observation. In this work, we varied the value of N from 20 to 150 at a discrete interval and found that the first 50 frames are adequate to model the background for the image sequences available in Wallflower and I2R datasets.

5 Conclusion

In this paper, we presented an effective background model to detect moving objects across a fixed camera view. An invariant color model is suggested to counter the shadow illumination. Its in-determinant behavior along the achromatic axis is addressed with the intensity feature. The uninteresting movement addressed using a multi-modal background. Unlike traditional equi-distribution methods, the proposed solution analyzes the temporal pixel sequence and assigns appropriate number of classes to each location. The problem of gradual illumination variation is addressed using a non-recursive architecture, where each background class is represented by a temporal queue that stores only the recently accessed background pixels. A modified Z-score is employed to separate the foreground pixels against the developed model. The duration of a foreground object being stationary and the period of absence of a background class have been mathematically formulated to counter the object relocation issues. Morphology is applied as a post-improvisation module to remove the cluttered noise, to join the disconnected foreground pixels, and to fill the camouflage holes.

Our approach is validated through extensive simulations on standard image sequences and the results are compared with some of the state-of-the-art methods. Accuracy measures such as Precision, Recall, Figure of merit, and Percentage of correct classification substantiate the efficacy of the proposed method over its counterparts.

Acknowledgements This work is supported by the Science and Engineering Research Board (SERB) of India under Grant Number SB/FTP/ETA-0059/2014.

References

Wang X (2013) Intelligent multi-camera video surveillance: a review. Pattern Recogn Lett 34(1):3–19

Doretto G, Sebastian T, Tu P, Rittscher J (2011) Appearance-based person reidentification in camera networks: problem overview and current approaches. J Ambient Intell Hum Comput 2(2):127–151

Albano P, Bruno A, Carpentieri B, Castiglione A, Castiglione A, Palmieri F, Pizzolante R, Yim K, You I (2014) Secure and distributed video surveillance via portable devices. J Ambient Intell Hum Comput 5(2):205–213

De Smedt F, Struyf L, Beckers S, Vennekens J, De Samblanx G, Goedemé T (2014) Faster and more intelligent object detection by combining opencl and kr. J Ambient Intell Hum Comput 5(5):635–643

Varga D, Szirányi T (2017) Robust real-time pedestrian detection in surveillance videos. J Ambient Intell Hum Comput 8(1):79–85

Setitra I, Larabi S (2014) Background subtraction algorithms with post-processing: a review. In: 22nd international conference on pattern recognition (ICPR), IEEE, pp 2436–2441

Sobral A, Vacavant A (2014) A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput Vis Image Underst 122:4–21

Elgammal A (2014) Background subtraction: theory and practice. Synth Lect Comput Vis 5(1):1–83

Wren C, Azarbayejani A, Darrell T, Pentland A (1997) P finder: real-time tracking of the human body. IEEE Trans Pattern Anal Mach Intell 19(7):780–785

Stauffer C, Grimson W (1999) Adaptive background mixture models for real-time tracking. IEEE Comput Soc Conf Comput Vis Pattern Recogn 2:246–252

Stauffer C, Grimson W (2000) Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal Mach Intell 22(8):747–757

Zivkovic Z (2004) Improved adaptive Gaussian mixture model for background subtraction. Int Conf Pattern Recogn IEEE 2:28–31

Zivkovic Z, van der Heijden F (2006) Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recogn Lett 27(7):773–780

Kim K, Halidabhongse T, Harwood D, Davis L (2005) Real-time foreground-background segmentation using codebook model. Real Time Imaging 11(3):172–185

Wu M, Peng X (2010) Spatio-temporal context for codebook-based dynamic background subtraction. AEU Int J Electron Commun 64(8):739–747

Fernandez-Sanchez E, Diaz J, Ros E (2013) Background subtraction based on color and depth using active sensors. Sensors (Basel, Switzerland) 13(7):8895–8915

Lo B, Velastin S (2001) Automatic congestion detection system for underground platforms. International symposium on intelligent multimedia. video and speech processing, IEEE, pp 158–161

Cucchiara R, Grana C, Piccardi M, Prati A (2003) Detecting moving objects, ghosts, and shadows in video streams. IEEE Trans Pattern Anal Mach Intell 25(10):1337–1342

Calderara S, Melli R, Prati A, Cucchiara R (2006) Reliable background suppression for complex scenes. In: ACM international multimedia conference and exhibition, pp 211–214

Toyama K, Krumm J, Brumitt B, Meyers B (1999) Wallflower: principles and practice of background maintenance. IEEE Int Conf Comput Vis 1:255–261

Zhong J, Sclaroff S (2003) Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. IEEE Int Conf Comput Vis 1:44–50

Wang H, Suter D (2006) Background subtraction based on a robust consensus method. Proc Int Conf Pattern Recognn 1:223–226

Wang H, Suter D (2007) A consensus-based method for tracking: modelling background scenario and foreground appearance. Pattern Recogn 40(3):1091–1105

McHugh J, Konrad J, Saligrama V, Jodoin PM (2009) Foreground-adaptive background subtraction. IEEE Signal Process Lett 16(5):390–393

Heikkil M, Pietikinen M (2006) A texture-based method for modeling the background and detecting moving objects. IEEE Trans Pattern Anal Mach Intell 28(4):657–662

Elgammal A, Harwood D, Davis L (2000) Non-parametric model for background subtraction. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 1843:751–767

Piccardi M, Jan Z (2004) Mean-shift background image modelling. Int Conf Image Process IEEE 5:3399–3402

Elgammal A, Duraiswami R, Davis LS (2001) Efficient non-parametric adaptive color modeling using fast Gauss transform. IEEE Comput Soc Conf Comput Vis Pattern Recogn IEEE 2:II-563

Parag T, Elgammal A, Mittal A (2006) A framework for feature selection for background subtraction. IEEE Comput Soc Conf Comput Vis Pattern Recogn IEEE 2:1916–1923

Hongxun Z, De X (2006) Fusing color and texture features for background model. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 4223 LNAI:887–893

El Baf F, Bouwmans T, Vachon B (2008) Foreground detection using the Choquet integral. In: International workshop on image analysis for multimedia interactive services, IEEE, pp 187–190

Azab MM, Shedeed H, Hussein AS, et al (2010) A new technique for background modeling and subtraction for motion detection in real-time videos. In: 17th IEEE international conference on image processing, IEEE, pp 3453–3456

Bouwmans T, El Baf F (2009) Modeling of dynamic backgrounds by type-2 fuzzy Gaussians mixture models. MASAUM J Basic Appl Sci 1(2):265–276

El Baf F, Bouwmans T, Vachon B (2009) Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos. In: IEEE computer society conference on computer vision and pattern recognition workshops, IEEE, pp 60–65

Kim W, Kim C (2012) Background subtraction for dynamic texture scenes using fuzzy color histograms. IEEE Signal Process Lett 19(3):127–130

Culibrk D, Marques O, Socek D, Kalva H, Furht B (2007) Neural network approach to background modeling for video object segmentation. IEEE Trans Neural Netw 18(6):1614–1627

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

Maddalena L, Petrosino A (2010) A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. Neural Comput Appl 19(2):179–186

Li L, Huang W, Gu I, Tian Q (2003) Foreground object detection from videos containing complex background. In: ACM international multimedia conference and exhibition, pp 2–10

Gevers T, Smeulders A (1999) Color-based object recognition. Pattern Recogn 32(3):453–464

Leys C, Ley C, Klein O, Bernard P, Licata L (2013) Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median. J Exp Soc Psychol 49(4):764–766

Iglewicz B, Hoaglin DC (1993) How to detect and handle outliers, vol 16. American Society for Quality Control Press, Milwaukee

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

Barnich O, Van Droogenbroeck M (2011) ViBe: a universal background subtraction algorithm for video sequences. IEEE Trans Image Process 20(6):1709–1724

Hati K, Sa P, Majhi B (2013) Intensity range based background subtraction for effective object detection. IEEE Signal Process Lett 20(8):759–762

Reddy V, Sanderson C, Lovell B (2013) Improved foreground detection via block-based classifier cascade with probabilistic decision integration. IEEE Trans Circuits Syst Video Technol 23(1):83–93

Sobral A, Baker C, Bouwmans T, Zahzah EH (2014) Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 8814:94–103

Choudhury SK, Sa PK, Bakshi S, Majhi B (2016) An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios. IEEE Access 4:6133–6150

Seo S (2006) A review and comparison of methods for detecting outliers in univariate data sets. Dissertation of M. Sc., Graduate School of Public Health, University of Pittsburgh, Pittsburgh, Pennsylvania, US

Goyal K, Singhai J (2017) Review of background subtraction methods using Gaussian mixture model for video surveillance systems. Artif Intell Rev:1–19. doi:10.1007/s10462-017-9542-x

Kumar S, Yadav J S (2017) Background Subtraction Method for Object Detection and Tracking. In: Proceeding of international conference on intelligent communication, control and devices:1057–1063

Szwoch G, Ellwart D, Czyżewski A (2016) Parallel implementation of background subtraction algorithms for real-time video processing on a supercomputer platform. J Real Time Image Process 11(1):111–125

Panda DK, Meher S (2016) Detection of moving objects using fuzzy color difference histogram based background subtraction. IEEE Signal Process Lett 23(1):45–49

Sajid H, Cheung S.C.S (2015) Background subtraction for static and moving camera. IEEE Int Conf Image Process:4530–4534

Shah M, Deng JD, Woodford BJ (2015) A Self-adaptive CodeBook (SACB) model for real-time background subtraction. Image Vis Comput 38:52–64