Journal of Electronic Imaging

JElectronicImaging.org

Comparative study of motion detection methods for video surveillance systems

Kamal Sehairi
Fatima Chouireb
Jean Meunier

SPIE. IS&T imaging.org

Kamal Sehairi, Fatima Chouireb, Jean Meunier, "Comparative study of motion detection methods for video surveillance systems," *J. Electron. Imaging* **26**(2), 023025 (2017), doi: 10.1117/1.JEI.26.2.023025.

Comparative study of motion detection methods for video surveillance systems

Kamal Sehairi,[a,*] Fatima Chouireb,[a] and Jean Meunier[b]
[a]University of Laghouat Amar Telidji, Telecommunications, Signals and Systems Laboratory, Laghouat, Algeria
[b]University of Montreal, Department of Computer Science and Operations Research, Montreal, Canada

Abstract. The objective of this study is to compare several change detection methods for a monostatic camera and identify the best method for different complex environments and backgrounds in indoor and outdoor scenes. To this end, we used the CDnet video dataset as a benchmark that consists of many challenging problems, ranging from basic simple scenes to complex scenes affected by bad weather and dynamic backgrounds. Twelve change detection methods, ranging from simple temporal differencing to more sophisticated methods, were tested and several performance metrics were used to precisely evaluate the results. Because most of the considered methods have not previously been evaluated on this recent large scale dataset, this work compares these methods to fill a lack in the literature, and thus this evaluation joins as complementary compared with the previous comparative evaluations. Our experimental results show that there is no perfect method for all challenging cases; each method performs well in certain cases and fails in others. However, this study enables the user to identify the most suitable method for his or her needs. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.2.023025]

Keywords: motion detection; background modeling; object detection; video surveillance.
Paper 161035 received Dec. 14, 2016; accepted for publication Mar. 29, 2017; published online Apr. 25, 2017.

1 Introduction

Motion detection, which is the fundamental step in video surveillance, aims to detect regions corresponding to moving objects. The resultant information often forms the basis for higher-level operations that require well-segmented results, such as object classification and action or activity recognition. However, motion detection suffers from problems caused by source noise, complex backgrounds, variations in scene illumination, and the shadows of static and moving objects. Various methods have been proposed to overcome these problems by retaining only the moving object of interest. These methods are classified[1–3] into three major categories: background subtraction,[4,5] temporal differencing,[6,7] and optical flow.[8,9] Temporal differencing is highly adaptive to dynamic environments; however, it generally exhibits poor performance in extracting all relevant feature pixels. Therefore, techniques such as morphological operations and hole filling are applied to effectively extract the shape of a moving object. Background subtraction provides the most complete feature data, but is extremely sensitive to dynamic scene changes due to lighting and extraneous events. Several background modeling methods have been proposed to overcome these problems. Bouwmans[10] classified these advanced methods into seven categories: basic background modeling,[11–13] statistical background modeling,[10,14–16] fuzzy background modeling,[17,18] background clustering,[19,20] neural network background modeling,[21–24] wavelet background modeling,[25,26] and background estimation.[27–29] Furthermore, Goyette et al.[30] categorized background modeling techniques into six families: basic,[31–34] parametric,[14,15,19,35–37] nonparametric and data-driven,[16,38–41] matrix decomposition,[42–45] motion segmentation,[46–48] and machine learning.[21,22,49–52] Sobral and Vacavant[53] adopted a similar categorization for motion detection methods: basic [frame difference (FD), mean, and variance over time],[18,54–57] statistical,[14,15,58–60] fuzzy,[17,18,61–63] neural and neuro-fuzzy,[21,22,64,65] and other models [principal component analysis (PCA),[42] Vumeter[66]]. Optical flow can be used to detect independently moving objects even in the presence of camera motion. However, most optical flow methods are computationally complex and cannot be applied to full-frame video streams in real time without specialized hardware.[67] Recent categories have emerged in the last few years, such as advanced nonparametric modeling (Vibe,[39] PBAS,[40] Vibe+[68]), background modeling by decomposition into matrices,[69–74] and background modeling by decomposition into tensors.[75–77]

2 Related Works

2.1 *Survey Papers*

Several surveys on motion detection methods have been presented in the last decade, the authors tried to detail the algorithms used, categorize them, and explain their different steps such as postprocessing, initialization, background modeling, and foreground generation. Bouwmans[78] presented the most complete survey of traditional and recent methods for foreground detection with more than 300 references, and these methods were categorized by the mathematical approach used. In addition, the author explored the different datasets available, existing background subtraction libraries and codes. Elhabian et al.[79] provided a detailed explanation of different background modeling methods. Specifically, they explained how models can be updated and initialized, and explored ways to measure and evaluate their performance.

*Address all correspondence to: Kamal Sehairi, E-mail: k.sehairi@lagh-univ.dz

Morris and Angelov[80] reviewed three pixel-wise subtraction techniques and compared their capabilities in seven points: resource utilization, computational speed, robustness to noise, precision of the output (no numerical results were provided), complexity of the environment, level of autonomy, and scalability. Bouwmans[81] in another work presented a review paper in which he classifies the different improvement techniques made to PCA method [eigen-backgrounds (Eig-Bg)] and compared these techniques with Gaussian detection methods and kernel density estimation (KDE) using Wallflower datasets. Cuevas et al.[82] presented a state of the art of different techniques for detecting stationary foreground objects, e.g., abandoned luggage, people remaining temporarily static or objects removed from the background; the authors addressed the main challenges in this field with different datasets available for testing these methods. Bux et al.[83] gave a complete survey on human activity recognition (HAR), providing the state of the art of foreground segmentation, feature extraction, and activity recognition, which constitute the different phases of HAR systems; in foreground segmentation phase, the authors first categorized all methods to background construction-based segmentation for static cameras and foreground extraction-based segmentation for moving cameras. They detailed the different steps of background construction (initialization, maintenance, and foreground detection) and classified these methods into five models: basic, statistical, fuzzy, neural network, and others. Cristani et al.[84] presented a comprehensive review of background subtraction techniques for mono- and multisensor surveillance systems that considers different kinds of sensors (visible, infrared, and audio). The authors presented a different taxonomy, classifying motion detection methods into three categories: perpixel, perregion, and perframe processing, and from each category emerges subcategories. The authors propose solutions for different challenging situations (adopted from the Wallflower dataset[85]) using the fusion of multisensors.

2.2 *Comparative Papers*

In recent years, many studies have also attempted to compare different motion detection methods. The aim of these studies is to define the accuracy, the speed, memory requirements, and capabilities to handle several situations. For this purpose, different challenging datasets have been developed in order to give a fair benchmark for all methods. Table 1 summarizes some previous comparison studies, the evaluated methods, datasets, and performance metrics used for each comparison.

Table 1 shows more than 60 motion detection methods that were tested on 8 datasets. Other datasets exist like: CMU,[136] UCSD,[137] BMC,[138] SCOVIS,[139] MarDCT.[140] Moreover, new datasets were introduced with depth cameras, such as kinect database[141] and RGB-D object detection dataset.[142] We can also find more specialized video datasets such as Fish4knowledge[143] for underwater fish detection and tracking. Other comparative studies can be found in Refs. 144–148.

Owing to the importance of the motion detection step, it is necessary to examine other motion detection methods that have not been evaluated thus far. In particular, various simple modifications to original methods (preprocessing, thresholding, filtering, etc.) can lead to different results.

The objective of this study is to evaluate and compare different motion detection methods and identify the best method for different situations using a challenging complete dataset. To this end, we tested the following methods: temporal differencing (FD),[86,149,150] three-frame difference (3-FD),[151–153] adaptive background (average filter),[90,154,155] forgetting morphological temporal gradient (FMTG),[156] ΣΔ background estimation,[157,158] spatio-temporal Markov field,[159–161] running Gaussian average (RGA),[14,162,163] mixture of Gaussians (MoG),[15,59,164] spatio-temporal entropy image (STEI),[165,166] difference-based STEI (DSTEI),[166,167] Eig-Bg,[42,168,169] and simplified self-organized map (Simp-SOBS)[24] methods. Many of these methods (3-FD, ΣΔ, FMTG, STEI, DSTEI, Simp-SOBS) have not been previously evaluated on challenging datasets; to this end, we used the CDnet2012[30,115] and CDnet2014[131] datasets and compared them with the well-known and classical algorithm of motion detection in the literature (RGA, MoG, and Eig-Bg). The CDnet2014[131] comprises a total of 53 videos of indoor and outdoor scenes with more than 159,000 images. Each scene represents different moving objects, such as boats, cars, trucks, cyclists, and pedestrians, captured in different scenarios (baseline, shadow, and intermittent object motion) as well as under challenging conditions (bad weather, camera jitter, dynamic background, and thermal). For each video, a ground truth is provided to allow precise and unified comparison of the change detection methods. Furthermore, the following seven metrics were used for evaluation: recall, specificity, false positive rate (FPR), false negative rate (FNR), percentage of wrong classification (PWC), precision, and F-measure.

The remainder of this paper is organized as follows. Section 3 reviews the motion detection algorithms used in this study (FD techniques, background modeling techniques and energy-based methods). Section 4 provides a detailed explanation of the evaluation metrics used to score and evaluate the above-mentioned methods. Section 5 presents and discusses the results for different categories. Finally, Sec. 6 concludes this paper.

3 Motion Detection Methods

Motion detection by a fixed camera poses a major challenge for video surveillance systems in terms of extracting the shape of moving objects. This is due to several problems related to the monitored environment, such as complex backgrounds (e.g., tree leaf movement), weather conditions (e.g., snow or rain), and variations in illumination, as well as the characteristics of the moving object itself, such as the similarity of its color to the background color, its size, and its distance from the camera. Therefore, in recent years, several methods have been developed to overcome these problems. This section reviews the motion detection algorithms used in this comparative study.

3.1 *Frame Difference (Temporal Differencing)*

The FD method is the simplest method for detecting temporal changes in intensity in video frames. In a gray-level image, for each pixel with coordinates (x, y) in frame I_{t-1}, we compute the absolute difference with its corresponding coordinates in the next frame I_t as

Table 1 Comparison studies on motion detection methods.

Comparative studies	Tested methods	Datasets used	Metrics used
Toyama et al.[86]	FD[86] Mean + threshold[86] Mean + covariance (RGA)[14] MoG[15] Normalized block correlation[87] Temporal derivative (MinMax)[88] Bayesian decision[89] Subspace learning-principle component analysis (Eig-Bg)[42] Linear predictive filter[86] Wallflower method[86]	Wallflower dataset[85]	FN FP
Piccardi[4]	RGA[14] Temporal median filter[90,91] MoG[15] KDE[16] Sequential kernel density approximation[92] Subspace learning-principle component analysis (Eig-Bg)[85] Co-occurrence of image variations[93]	—	Limited accuracy (*L*) Intermediate accuracy (*M*) High accuracy (*H*)
Cheung and Kamath[94]	Frame differencing[86,94] Temporal median filter[90,91] Linear predictive filter[86] KDE[16] Approximated median filter[94] Kalman filter[95] MoG[15]	KOGS/-IAKS Universitaet Karlsruhe dataset[96]	Recall Precision
Benezeth et al.[97]	Temporal median filter (basic motion detection)[90,91] RGA (one Gaussian)[14] Minimum, maximum, and maximum inter-FD (MinMax)[88] MoG[15] KDE[16] Codebook[19] Subspace learning-principle component analysis (Eig-Bg)[89]	Synthetic videos Semisynthetic videos and VSSN 2006 dataset[98] IBM dataset[99]	Recall Precision
Bouwmans[10]	MoG[15] MoG with particle swarm optimization (MoG-PSO)[100] Improved MoG[101] MoG with MRF[102] MoG improved HLS color space[103] Spatial-time adaptive per pixel MoG (S-TAP-MoG)[104] Adaptive spatio-temporal neighborhood analysis[105] Subspace learning-principle component analysis (Eig-Bg)[42] Subspace learning-independent component analysis[106] Subspace learning incremental nonnegative matrix factorization[107] Subspace learning using incremental rank-tensor[108]	Wallflower dataset[85]	FN FP
Goyette et al.[109]	Euclidean distance[97] Mahalanobis distance[97] Local-self similarity[110] MoG[15] GMM KaewTraKulPong[58] GMM Zivkovic[60] GMM RECTGAUSS-Tex[111] Bayesian multilayer[36] ViBe[39] KDE[16] KDE Nonaka et al.[112] KDE Yoshinaga et al.[113] Self-organized background subtraction (SOBS)[21] Spatially coherent SOBS (SC-SOBS)[22] Chebyshev probability[28] ViBe+[66] Probabilistic super-pixel Markov random fields (PSP-MRF)[114] Pixel-based adaptive segmenter (PBAS)[40]	CDnet 2012 dataset[115]	Recall Specificity FPR FNR PWCs Precision *F*-measure

Table 1 (*Continued*).

Comparative studies	Tested methods	Datasets used	Metrics used
Wang et al.[116]	Euclidean distance[97] Mahalanobis distance[97] Multiscale spatio-temp BG model[117] GMM Zivkovic[60] CP3-online[118] MoG[15] KDE[16] SC-SOBS[22] *K*-nearest neighbor (KNN) method[60,119] Fast self-tuning BS[120] Spectral-360 (Ref. 121) Weightless neural networks (CwisarDH)[51,122] Majority vote-all[116] Self-balanced local sensitivity (SuBSENSE)[123] Flux tensor with split Gaussian (FTSG) models[124] Majority vote-3 (Ref. 116)	CDnet 2014 dataset[125]	Recall Specificity FPR FNR PWCs Precision *F*-measure
Jodoin et al.[125]	Euclidean distance[97] Mahalanobis distance[97] MoG[15] GMM Zivkovic[60] GMM KaewTraKulPong[58] GMM RECTGAUSS-Tex[111] KDE[16] KDE Nonaka et al.[112] KDE Yoshinaga et al.[113] SOBS[21] SC-SOBS[22] KNN method[119] Spectral-360 (Ref. 121) FTSG models[124] PBAS[40] PSP-MRF[114] Splitting Gaussian mixture model (SGMM)[126] Splitting over-dominating modes GMM (SGMM-SOD)[127] Dirichlet process GMM (DPGMM)[128] Bayesian multilayer[36] Histogram over time[13] Local-self similarity[110]	CDnet 2012 dataset[115]	FPR FNR PWCs
Bianco et al.[129]	IUTIS-1 (Ref. 129) IUTIS-2 (Ref. 129) IUTIS-3 (Ref. 129) FTSG models[124] SuBSENSE[123] Weightless neural networks (CwisarDH)[51,122] Spectral-360 (Ref. 121) Fast self-tuning BS[120] KNN method[119] KDE[16] SC-SOBS[22] Euclidean distance[97] Mahalanobis distance[97] Multiscale spatio-temp BG model[117] CP3-online[118] MoG[15] GMM Zivkovic[60] Fuzzy spatial coherence-based SOBS[65] Region-based MoG (RMoG)[130]	CDnet 2014 dataset[125,131]	Recall Specificity FPR FNR PWCs Precision *F*-measure
Xu et al.[132]	MoG[15] KDE[16] Codebook[19] SOBS[21] ViBe[39] PBAS[40] GMM Zivkovic[60] (adaptive GMM) Sample consensus (SACON)[133,134]	CDnet 2014 dataset[131] Video dataset proposed by Wen et al.[135]	Recall Specificity FPR FNR PWCs Precision *F*-measure

$$\zeta(x,y) = |I_t(x,y) - I_{t-1}(x,y)|. \quad (1)$$

For an RGB color image, we can compute this difference by various means, such as Manhattan distance [Eq. (2)], Euclidean distance [Eq. (3)], or Chebyshev distance [Eq. (4)][97,170–173]

$$\zeta(x,y) = |I_t^R(x,y) - I_{t-1}^R(x,y)| + |I_t^G(x,y) - I_{t-1}^G(x,y)| + |I_t^B(x,y) - I_{t-1}^B(x,y)|, \quad (2)$$

$$\zeta(x,y) = \sqrt{[I_t^R(x,y) - I_{t-1}^R(x,y)]^2 + [I_t^G(x,y) - I_{t-1}^G(x,y)]^2 + [I_t^B(x,y) - I_{t-1}^B(x,y)]^2}, \quad (3)$$

$$\zeta(x,y) = \max\{|I_t^R(x,y) - I_{t-1}^R(x,y)|, |I_t^G(x,y) - I_{t-1}^G(x,y)|, |I_t^B(x,y) - I_{t-1}^B(x,y)|\}, \quad (4)$$

where $I_t^C(x,y)$ represents the pixel value in the C channel.

In spite of its simplicity, this method offers the following advantages. It exhibits good performance in dynamic environments (e.g., during sunrise or under cloud cover) and works well at the standard video frame rate. In addition, the algorithm is easy to implement, with relatively low design complexity, and can be executed effectively when applied to a real-time system.[174]

3.2 *Three-Frame Difference*

The 3-FD[151] method is based on the temporal differencing method. Two-FD operations given by Eqs. (5) and (6) are performed; then the results are thresholded using Eq. (7) and combined using Eq. (8), i.e., the AND logical operator (or the minimum) (see Fig. 1)

$$\zeta_1(x,y) = |I_t(x,y) - I_{t-1}(x,y)|, \quad (5)$$

$$\zeta_2(x,y) = |I_t(x,y) - I_{t+1}(x,y),| \quad (6)$$

$$\psi_t(x,y) = \begin{cases} 0, & \text{if } \zeta_t(x,y) < \mathrm{Th}_t \quad \text{background} \\ 1, & \text{otherwise} \quad\quad\quad\ \text{foreground} \end{cases}, \quad (7)$$

$$\zeta_t(x,y) = \mathrm{Min}[\psi_1(x,y), \psi_2(x,y)]. \quad (8)$$

This method is robust to noise and provides good detection results for slow moving objects.

3.3 *Adaptive Background Subtraction (Running Average Filter)*

The concept underlying this method is to compute the average of the previous N frames to model the background, to update the first background image by considering new static objects in the scene. The background image is obtained as in Ref. 155, where τ is the time required to acquire N images

$$B(x,y) = \frac{1}{\tau}\sum_{t=1}^{\tau} I_t(x,y). \quad (9)$$

From Eq. (9), this method consumes a significant amount of memory, which causes problems for real-time implementation in particular. To overcome these problems, it is better to compute the background recursively (Fig. 2) as

$$B_{t+1}(x,y) = (1-\alpha)B_t(x,y) + \alpha I_t(x,y), \quad (10)$$

where $\alpha \in [0,1]$ is a time constant that specifies how fast new information supplants old observations. The larger the value of α, the higher the rate at which the background frame is updated with new changes in the scene. However, α cannot be too large, because it may cause artificial "tails" behind moving objects.[154] In fact, to prevent tail formation, α must be fixed according to the observed scene, the size and speed of the moving objects, and the distance of these objects from the camera. Furthermore, the problem of continuous movement of small background objects, especially in outdoor scenes (e.g., fluttering flags and swaying tree branches), can be addressed by segmenting such objects with the moving objects.

3.4 *Forgetting Morphological Temporal Gradient*

In this method, which was first introduced by Richefeu and Manzanera,[156] the difference between temporal dilation and temporal erosion defines the change, given by

$$\delta_\tau[I_t(x,y)] = \max_{z\in\tau}\{I_{t+z}(x,y)\}, \quad (11)$$

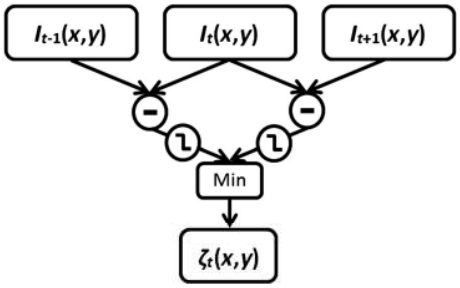

Fig. 1 3-FD method.

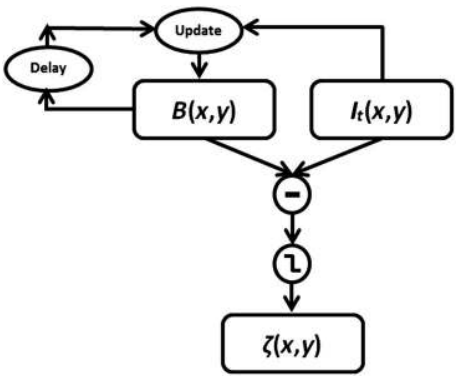

Fig. 2 Adaptive background detection.

$$\varepsilon_\tau[I_t(x,y)] = \min_{z\in\tau}\{I_{t+z}(x,y)\}, \tag{12}$$

where $\tau = [t_1, t_2]$ is the temporal structuring element.

In order to reduce not only the memory consumption linked to the use of the temporal structuring element but also the sensitivity of this method to large sudden variations, the authors used a running average filter (RAF) to recursively estimate the values of the temporal erosion and dilation. Thus, Eqs. (11) and (12), respectively, become

$$M_t(x,y) = \alpha I_t(x,y) + (1-\alpha)\max\{I_t(x,y), M_{t-1}(x,y)\}, \tag{13}$$

$$m_t(x,y) = \alpha I_t(x,y) + (1-\alpha)\min\{I_t(x,y), m_{t-1}(x,y)\}, \tag{14}$$

where $M_t(x,y)$, $m_t(x,y)$ denote the forgetting temporal dilation and the forgetting temporal erosion, respectively.

The FMTG is given by

$$\Gamma_t(x,y) = M_t(x,y) - m_t(x,y). \tag{15}$$

Furthermore, the authors tried to combine this method with the ΣΔ filter to improve the results and automatically define the time constant α.

3.5 ΣΔ *Background Estimation*

Proposed by Manzanera and Richefeu,[157] this method is based on the nonlinear ΣΔ filter used in electronics applications for analogue-to-digital conversion. The principle of this algorithm is to estimate two values, namely the current background image M_t and the time-variance image V_t, using an iterative process to increment or decrement these values. The algorithm is executed in four steps:[158]

(1) Computation of ΣΔ mean

$$\left.\begin{aligned} &M_0(x,y) = I_0(x,y)\\ &M_t(x,y) = M_{t-1}(x,y) + \operatorname{sgn}[I_t(x,y) - M_{t-1}(x,y)] \end{aligned}\right\}, \tag{16}$$

(2) Computation of difference

$$\Delta_t(x,y) = |M_t(x,y) - I_t(x,y)|, \tag{17}$$

(3) Computation of ΣΔ variance

$$\left.\begin{aligned} &V_0(x,y) = \Delta_0(x,y)\\ &\text{if } \Delta_t(x,y) \neq 0, V_t(x,y) = V_{t-1}(x,y)\\ &\qquad + \operatorname{sgn}[N\times\Delta_t(x,y) - V_{t-1}(x,y)] \end{aligned}\right\}, \tag{18}$$

(4) Computation of motion label

$$\begin{cases} D_t(x,y) = 0 & \text{if } \Delta_t(x,y) < V_t(x,y)\\ D_t(x,y) = 1 & \text{else} \end{cases}. \tag{19}$$

The only parameter to be set in this method is N, which represents the amplification factor. However, the application of this method entails several problems such as noise and ghost effects due to moving objects that remain static for long periods in the scene. To overcome this problem, the authors proposed a hybrid geodesic morphological reconstruction filter[157] based on the forgetting morphological operator,[156] and given by

$$\Delta_t' = H\mathrm{Re}\, c_{a'}^{\Delta_t}[\mathrm{Min}(\|\nabla(I_t)\|, \|\nabla(\Delta_t)\|)], \tag{20}$$

where the gradients of I_t and Δ_t are obtained by convolution with Sobel masks, and α is the time constant. Furthermore, the classical geodesic relaxation $\mathrm{Re}\, c^{\Delta_t}$ is defined by the geodesic dilation as $\mathrm{Re}\, c^{\Delta_t}[\mathrm{Min}(\|\nabla(I_t)\|, \|\nabla(\Delta_t)\|)] = \mathrm{Min}\{\delta_B[\mathrm{Min}(\|\nabla(I_t)\|, \|\nabla(\Delta_t)\|), \Delta_t]\}$, where δ is the morphological dilation operator and B is the structuring element.

3.6 *Markov Random Field-Based Motion Detection Algorithm*

Introduced by Bouthemy and Lalande,[159] this algorithm aims to improve image difference using a Markovian process. To this end, the authors have defined motion detection in images as a binary labeling problem, where the appropriate labels are given by

$$\begin{cases} e(x,y,t) = 1 & \text{if the pixel belongs to a moving object}\\ e(x,y,t) = 0 & \text{if the pixel belongs to a static background} \end{cases}, \tag{21}$$

and the observation is the absolute difference between two consecutive frames or between the current frame and a reference image

$$O_t(x,y) = |I_t(x,y) - I_{t-1}(x,y)|. \tag{22}$$

The maximum *a posteriori* (MAP) criterion is used to estimate the appropriate labels of field E given field of observation O interpreted by maximizing the conditional probability

$$\max_e P[E = e|O = o]. \tag{23}$$

Using Bayes' theorem, this is equivalent to

$$\max_e \frac{P[O = o|E = e]P[E = e]}{P[O = o]}, \tag{24}$$

where $P[O = o]$ is constant with respect to the maximization because the observations are inputs. Conversely, the maximization of $P[O = o|E = e]P[E = e]$ is equivalent to the minimization of an energy function derived from the Hammersley–Clifford theorem, which states that MRF exhibit a Gibbs distribution with an energy function as[175,176]

$$P[E = e|O = o] = \frac{e^{-U(o,e)}}{Z}, \tag{25}$$

where Z is a normalizing factor. The energy function U is given by the sum of two terms

$$U(e,o) = U_m(e) + U_a(o,e), \tag{26}$$

where U_m denotes the energy that ensures spatio-temporal homogeneity and U_a denotes the adequacy energy that ensures good coherence of the solution compared to the observed data

$$U_a(o,e)=\frac{1}{2\sigma^2}[o-\psi(e)]^2,$$
$$\psi(e)=\begin{cases}0 & \text{if } |I_t(x,y)-I_{t-1}(x,y)|<\text{Th}\\ \alpha & \text{else}\end{cases}, \tag{27}$$

$$U_m(e)=\sum_{c\in C}V_c(e_s,e_r), \tag{28}$$

where c denotes a set of binary cliques associated with the chosen neighborhood system describing spatio-temporal interactions between the different pixel intensities.[159] In our case, there is a 3×3 spatial neighborhood window and two temporal connections: (x,y,t) to $(x,y,t-1)$ and (x,y,t) to $(x,y,t+1)$ (see Fig. 3)

Furthermore, V_c is given by

$$\begin{cases}V_c(e_s,e_r)=V_s(e_s,e_r)+V_p(\mathrm{e}_s^t,\mathrm{e}_s^{t-1})+V_p(\mathrm{e}_s^t,\mathrm{e}_s^{t+1})\\ V_s(e_s,e_r)=\begin{cases}-\beta_s & \text{if } \mathrm{e}_s=\mathrm{e}_r\\ +\beta_s & \text{if } \mathrm{e}_s\neq e_r\end{cases}\\ V_p(\mathrm{e}_s^t,\mathrm{e}_s^{t-1})=\begin{cases}-\beta_p & \text{if } \mathrm{e}_s^t=\mathrm{e}_s^{t-1}\\ +\beta_p & \text{if } \mathrm{e}_s^t\neq\mathrm{e}_s^{t-1}\end{cases}\\ V_p(\mathrm{e}_s^t,\mathrm{e}_s^{t+1})=\begin{cases}-\beta_f & \text{if } \mathrm{e}_s^t=e_s^{t+1}\\ +\beta_f & \text{if } \mathrm{e}_s^t\neq e_s^{t+1}\end{cases}.\end{cases} \tag{29}$$

After defining the energy U for our Markovian model, the authors considered the problem of minimizing this energy, ultimately using an iterative deterministic relaxation technique[159] (iterated conditional mode method) (see Fig. 4).

3.7 *Running Gaussian Average (One Gaussian)*

In this method, the background is modeled by fitting a Gaussian distribution (μ,σ) over a histogram for each pixel;[4,14] this gives the probability density function (pdf)

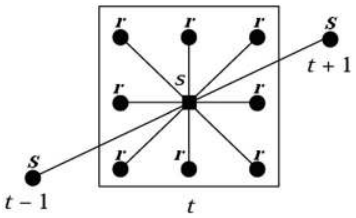

Fig. 3 3×3 spatio-temporal neighborhood.[106]

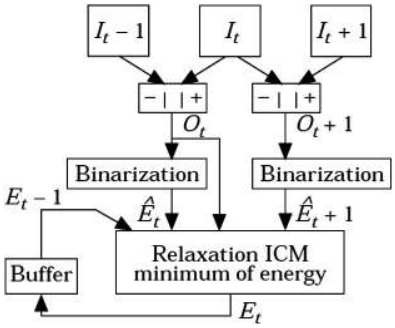

Fig. 4 MRF-based motion detection.[106]

of the background.[14] In order to update this pdf, an RAF is applied to the parameters of the Gaussian

$$\mu_t=\alpha I_t+(1-\alpha)\mu_{t-1}, \tag{30}$$

$$\sigma_t^2=\alpha(I_t-\mu_t)^2+(1-\alpha)\sigma_{t-1}^2. \tag{31}$$

Then, the pixels correspond to a moving object if the following inequality is satisfied

$$|I_t-\mu_t|>D\sigma_t, \tag{32}$$

where D is the deviation threshold (e.g., $D=2.5$).

This method offers the advantages of high execution speed and low memory consumption. However, it suffers from problems associated with the use of the running average (appropriate choices of α and deviation threshold D). Moreover, the use of a single Gaussian to model the background will not give good results for complex backgrounds; in addition, it will favor the extraction of the shadows of moving objects.

3.8 *Gaussian Mixture Model*

In the Gaussian mixture model (GMM) (or MoG), proposed by Stauffer and Grimson,[15] the temporal histogram of each pixel X is modeled using a mixture of K Gaussian distributions in order to precisely model a dynamic background. For example, the periodic or random oscillation of a tree branch that sways in the wind and hides the sun is modeled using two Gaussians. One Gaussian models the temporal variation in the intensity of the pixels when the tree branch obstructs the sun, and the other Gaussian represents the different local intensities produced by the sun. The intensity of each pixel is compared to these Gaussian mixtures, which represent the probability distribution of possible intensities belonging to the dynamic background model. Low probability of belonging to these Gaussian mixtures indicates that the pixel belongs to a moving object. The probability of observing the current pixel value in the multidimensional case is given by

$$P(X_t)=\sum_{k=1}^{K}\omega_{k,t}\eta(\mu_{k,t},\Sigma_{k,t},X_t), \tag{33}$$

where $\omega_{k,t}$ is the estimated weight associated with the k'th Gaussian at time t, $\mu_{k,t}$ is the mean of the k'th Gaussian at time t, and $\Sigma_{k,t}$ is the covariance matrix. Furthermore, η is a Gaussian pdf

$$\eta(\mu_{k,t},\Sigma_{k,t},X_t)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{1}{2}(X_t-\mu_t)^T\Sigma^{-1}(X_t-\mu_t)}. \tag{34}$$

Owing to limited memory and computing capacity, the authors have set K in the range of 3 to 5, and they have assumed that the RGB color components are independent and have the same variances. Hence, the covariance matrix is of the form[59]

$$\sum_{i,k}=\sigma_{i,k}^2 I. \tag{35}$$

The first step is to initialize the parameters of the Gaussians $(\omega_k,\mu_k,\Sigma_k)$. Then, a test is performed to match each new pixel X with the existing Gaussians

$$|\mu_k - X_t| \leq D\sigma_k \qquad k = 1, \ldots, M, \tag{36}$$

where M is the number of Gaussians and D is the deviation threshold.

If a match is found, we update the parameters of this matched Gaussian as

$$\rho = \frac{\alpha}{\omega}, \tag{37}$$

$$\omega_t = (1-\alpha)\omega_{t-1} + \alpha, \tag{38}$$

$$\mu_t = \rho X_t + (1-\rho)\mu_{t-1}, \tag{39}$$

$$\sigma_t^2 = \rho(X_t - \mu_t)^2 + (1-\rho)\sigma_{t-1}^2, \tag{40}$$

where α and ρ are learning rates; here, ρ is taken from the alternative approximation proposed by Power and Schoonees.[177] For the other unmatched distributions, we maintain their mean and variance and update only the weights as in

$$\omega_t = (1-\alpha)\omega_{t-1}. \tag{41}$$

Then, we normalize all the weights as $\omega_k / \sum_{k=1}^{M} \omega_k$.

If no match is found with any of the K distributions, we create a new distribution that replaces the parameters of the least probable one, with the current pixel value as its mean, an initially high variance, and low prior weight

$$\mu_t = X_t, \tag{42}$$

$$\sigma_t^2 = \sigma_0^2 \qquad \text{(largest value)}, \tag{43}$$

$$\omega_t = \min(\omega_t) \qquad \text{(smallest value)}. \tag{44}$$

Then, to distinguish the foreground distribution from the background distribution, we order the distributions by the ratio of their weights to their standard deviations (ω_k/σ_k), assuming that the higher and more compact the distribution, the greater the likelihood of belonging to the background. Then, the first B distributions in the ranking order satisfying Eq. (45) are considered background[4]

$$\sum_{k=1}^{B} \omega_k > T, \tag{45}$$

where T is a threshold value. Finally, each new pixel value X is compared to these background distributions. If a match is found, this pixel is considered to be a background pixel; otherwise, it is considered to be a foreground pixel

$$|\mu_k - X_t| \leq D\sigma_k, \qquad k = 1, \ldots, B. \tag{46}$$

This method can yield good results for dynamic backgrounds by fitting multiple Gaussians to represent the background more effectively. However, it entails two problems: high computational complexity and parameter initialization. Furthermore, additional parameters must be fixed, such as the threshold value and learning values α and ρ. Many improvements on this method, which deal with the issues and complications affecting the standard algorithm, such as the updating process, initialization, and approximation of the learning rate, can be found in the literature. We refer the readers to the following papers: Power and Schoonees[177] explained in detail the standard MoG method used by Stauffer and Grimson; Bouwmans et al.[59] discussed the improvements made to the standard MoG method; Carminati and Benois-Pineau[178] used an ISODATA algorithm to estimate the number of K Gaussians for each pixel, and for the matching test the authors use the likelihood maximization followed by Markov regularization instead of the approximation of MAP; Kim et al.[179] showed that an indoor scene is much closer to a Laplace distribution than to a Gaussian, for which a generalized Gaussian distribution (G-GMM) is proposed instead of a GMM to model the background. Makantasis et al.[180] proposed to use the Student-t mixture model (STMM) rather than the Gaussian, due to the smaller number of parameters to be tuned. However, the use of STMM increases the complexity of calculation and the memory requirements. To solve this problem, the authors used an image grid; if change is detected using FD, the background modeling is applied in the corresponding grid. Many other works tried to use different mixture models like the Dirichlet[128] or hybrid (KDE-GMM) mixture model.[78,181]

3.9 *Spatio-Temporal Entropy Image*

In this method, which was proposed by Ma and Zhang,[165] a statistical approach is adopted to measure the variation of each pixel based on its $w \times w$ neighbors along L accumulated frames. A spatio-temporal histogram is created for each pixel, $H_{x,y,q}$ (Fig. 5), where q denotes the bins of the histogram, and the components of the histogram are $\{H_{x,y,1}, \ldots, H_{x,y,Q}\}$, where Q is the total number of bins. Then, the corresponding pdf for each pixel is given by[166]

$$P_{x,y,q} = \frac{H_{x,y,q}}{N}, \tag{47}$$

where $N = L \times w \times w$.

To determine whether this pixel belongs to the background or foreground, an entropy measure $E_{x,y}$ is computed from the pdf

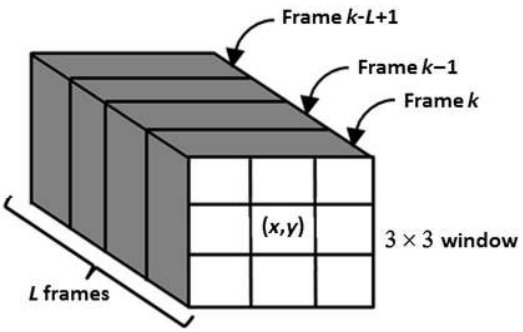

Fig. 5 Pixels used to construct the spatio-temporal histogram of pixel (i,j).

$$E_{x,y} = -\sum_{q=1}^{Q} P_{x,y,q} \log(P_{x,y,q}). \quad (48)$$

Entropy is a measure of disorder; it is thus assumed that the entropy for a change due to noise is small compared to that due to a moving object.

The diversity of the state of each pixel indicates the intensity of motion at its position.[165]

Ma and Zhang[165] first binarized the entropy result using an adaptive threshold and then applied a morphological filter (close–open operation) to enhance the results.

3.10 *Difference-Based Spatio-Temporal Entropy Image*

To overcome the problems associated with spatio-temporal entropy, especially the errors resulting from edge pixels, Jing et al.[166] attempted to use simple temporal differencing as

$$D_t = \Phi(|I_t - I_{t-1}|), \quad (49)$$

where Φ quantizes the 256 gray-level values into Q gray levels. As in the case of the STEI method, a spatio-temporal histogram is constructed for each pixel using a $w \times w$ window along an FD of L as

$$H_{x,y,q}(L) = \frac{1}{L}\sum_{k=1}^{L} h_{x,y,q}(k), \quad (50)$$

where $h_{x,y,q}(k)$ is the spatial histogram of each pixel in frame k.

To reduce memory consumption in a real-time system, the authors proposed recursive computation of the spatio-temporal histogram using Eq. (51), where α is a time constant that determines the influence of the previous frames

$$H_{x,y,q}(k+1) = \alpha H_{x,y,q}(k) + (1-\alpha)h_{x,y,q}(k+1). \quad (51)$$

Then, the pdf of each pixel $P_{x,y,q}$ is obtained by normalizing Eq. (47). Subsequently, the entropy of each pixel is obtained using Eq. (48). Finally, a thresholding method is used to extract the motion region.

3.11 *Eigen-Background Subtraction*

The Eig-Bg method, or subspace learning using principle component learning (SL-PCA), is a background modeling method developed by Olivier et al.[42] The concept underlying this method is that the moving object is rarely found in the same position in the scene across the training frames; hence, its contribution to the eigenspace model is not significant. Conversely, the static objects in the scene can be well described as the sum of various eigenbasis vectors. In this method, an eigenspace is formed using N reshaped training frames, $\mathrm{ES} = [I_1 I_2, \ldots, I_N]$, with mean μ and covariance C

$$\mu = \sum_{k=1}^{N} I_k, \quad (52)$$

$$C = \mathrm{Cov}(\mathrm{ES}) = \mathrm{ES} \cdot \mathrm{ES}^{\mathrm{T}} = \frac{1}{N}\sum_{k=1}^{N}[I_k - \mu] \cdot [I_k - \mu]^{\mathrm{T}}. \quad (53)$$

Then, we compute M principal eigenvectors by PCA, using singular value decomposition or eigendecomposition; the eigendecomposition is given by

$$C = V \Lambda V^{\mathrm{T}}, \quad (54)$$

where $\Lambda = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ is the diagonal matrix that contains the eigenvalues ($\lambda_1 > \lambda_2 > \cdots > \lambda_N$) and V is the eigenvector matrix. Furthermore, V_M consists of the M eigenvectors in V that correspond to the M largest eigenvalues.[169,182] Then, every new image I is normalized to the mean of the eigenspace μ and projected on these M eigenvectors as

$$I' = V_M^{\mathrm{T}}(I - \mu). \quad (55)$$

Subsequently, the background is reconstructed by back-projection as

$$B = V_M I' + \mu. \quad (56)$$

Finally, the foreground can be detected by thresholding the absolute difference as

$$|I - B| > \mathrm{Th}. \quad (57)$$

The authors were very satisfied with the accuracy of the results obtained and specified that their method entails a lower computational load than the MoG method. However, they did not explain how to choose the images that form the eigenspace, because the model is based on the content of these images. If a scene includes a slow or large moving object or crowd movement in the eigenspace, the background model, which should contain only the static objects, will be inappropriate. Furthermore, the authors have not explained how to update the background to consider the possibility of moving objects becoming static. Many methods have been proposed to overcome these problems, e.g., updating the Eig-Bg[43,182] and using a selective mechanism.[169] A complete survey on PCA techniques applied to background subtraction can be found in the work of Bouwmans and Zahzah.[69,81]

3.12 *Simplified Self-Organized Background Subtraction*

The use of a self-organizing map for background modeling was first proposed by Maddalena and Petrosino.[21] They built a model by mapping each color pixel $I_t(x, y)$ into an $n \times n$ weight vector, thus obtaining a neuronal map B_t of size $[n \times W, n \times H]$, where W and H are the width and height of the observed scene. The initial neuronal map B_0 is obtained in the same manner, where I_0 represents the scene containing the static objects (Fig. 6).

Then, for each pixel $I_t(x, y)$ from the incoming frame, a matching test is performed with the corresponding weights b_i, $i = 1, \ldots, n^2$ to find the best match b_m defined by the

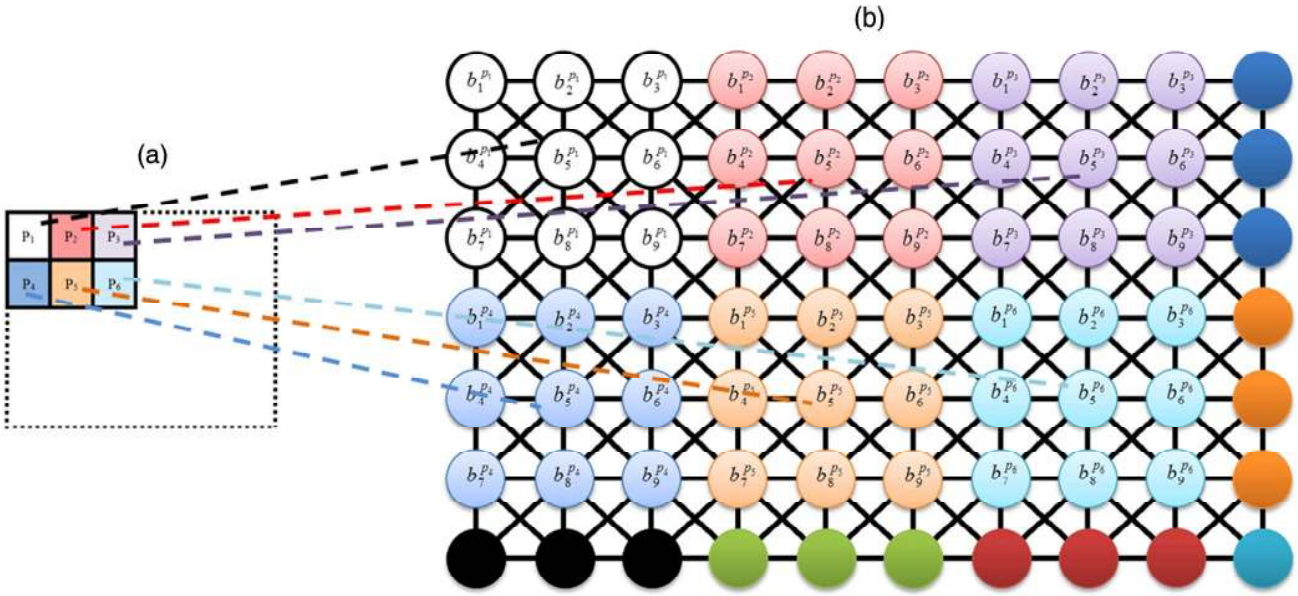

Fig. 6 (a) A simple image I_t and (b) the modeling neuronal map B_t when $n = 3$.

minimum distance between $I_t(x, y)$ and b_i that should not be greater than a predefined threshold Th

$$d_{\min}[b_m, I_t(x, y)] = \min_{i=1,\ldots,n^2} \{d[b_i, I_t(x, y)]\} \leq \text{Th}. \quad (58)$$

The difference is computed per the color space of the image. Maddalena and Petrosino[21] used the HSV hexagonal cone color space; the Euclidean distance in this case is given by[172]

$$d[b_i, I_t(x, y)] = \sqrt{[v_{b_i} s_{b_i} \cos(h_{b_i}) - v_{I_t} s_{I_t} \cos(h_{I_t})]^2 + [v_{b_i} s_{b_i} \sin(h_{b_i}) - v_{I_t} s_{I_t} \sin(h_{I_t})]^2 + (v_{b_i} - v_{I_t})^2}. \quad (59)$$

If the best match is found in background B_t at position $(\bar{x}, \bar{y})$, we consider the pixel $I_t(x, y)$ to be a background pixel; otherwise, we consider it to be a foreground pixel. We update the model around the best match position as

$$b_{t+1}(i, j) = b_t(i, j) + \alpha_{i,j}(t)[I_t(x, y) - b_t(i, j)]. \quad (60)$$

For $i = \bar{x} - \lfloor n/2 \rfloor, \ldots, \bar{x} + \lfloor n/2 \rfloor$, $j = \bar{y} - \lfloor n/2 \rfloor, \ldots, \bar{y} + \lfloor n/2 \rfloor$, $\alpha_{i,j}(t) = \alpha(t)\omega_{i,j}$, where $\omega_{i,j}$ are $n \times n$ Gaussian weights and $\alpha(t)$ is the learning factor given by

$$\alpha(t) = \begin{cases} \alpha_1 - t\left(\frac{\alpha_1 - \alpha_2}{K}\right), & \text{if } 0 \leq t \leq K \\ \alpha_2, & \text{if } t > K \end{cases} \quad (61)$$

where α_1 and α_2 are predefined constants such that $\alpha_2 \leq \alpha_1$ and K is the number of frames required for the calibration phase, which depends on how many static initial frames are available for each sequence.

To reduce the computational load as well as the number of parameters to be tuned, Chacon-Murguia et al.[24] proposed an self organizing map (SOM)-like architecture in which the mapping is one-to-one, i.e., each neuron is associated with its corresponding pixel. Since each pixel $I_t(x, y)$ is represented in the HSV color space, each neuron $b(x, y)$ has three inputs h_b, s_b, v_b, and the matching test is performed as

$$d[b, I_t(x, y)] \leq \text{Th} \wedge |v_{I_t} - v_b| \leq \text{Th}_v, \quad (62)$$

where $d[b, I_t(x, y)]$ is the Euclidean distance in HSV color space, defined previously in Eq. (59), and Th and Th_v are threshold values experimentally set by the authors. The second condition eliminates object shadows. If the result is true, the current pixel is considered to be a background pixel and the weights of the corresponding neuron $b(x, y)$ and its neighbors are updated using Eqs. (63) and (64), respectively

$$b_{t+1}(x, y) = b_t(x, y) + \alpha_1[I_t(x, y) - b_t(x, y)], \quad (63)$$

$$b_{t+1}(x', y') = b_t(x', y') + \alpha_2[I_t(x', y') - b_t(x', y')], \quad (64)$$

where $x' = x - 1, x + 1$ and $y' = y - 1, y + 1$ are the coordinates of the neighboring neurons, and α_1 and α_2 are the learning rates, with $\alpha_1 > \alpha_2$ for nonuniform learning.

If the result of Eq. (62) is not true, the current pixel is considered to be a foreground pixel and no update is required. The authors showed that this simplified model performs satisfactorily in different scenarios.

Many other studies have attempted to improve the original self-organized background subtraction (SOBS) method. For example, Maddalena and Petrosino[65] introduced fuzzy rules for subtraction and process updates. Furthermore, the authors[183,184] used a 3-D neuronal map to model the background; the map consists of n layers of the classical two-grid neuronal map and considers scene changes over time.

4 Evaluation Metrics and Performance Analysis

To correctly evaluate these methods and achieve a fair comparison, the methods were applied to the same dataset. Furthermore, to define the properties of each method, we used the same seven metrics as those used for CDnet2014: recall, specificity, FPR, FNR, PWC, precision, and F-measure. The role of these metrics is to quantify how well each algorithm matches the ground truth. All metrics are based on the following four quantities:[79]

- True positives (TP): number of foreground pixels correctly detected.
- False positives (FP): number of background pixels incorrectly detected as foreground pixels.
- True negatives (TN): number of background pixels correctly detected.
- False negatives (FN): number of foreground pixels incorrectly detected as background pixels (also known as misses).

Recall (the sensitivity or true positive rate) is the ratio of the number of foreground pixels correctly detected by the algorithm to the number of foreground pixels in the ground truth[79]

$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{65}$$

Specificity (the true negative rate) represents the percentage of correctly classified background pixels

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}. \tag{66}$$

FPR is the ratio of the number of background pixels incorrectly detected as foreground pixels by the algorithm to the number of background pixels in the ground truth

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}. \tag{67}$$

FNR is the ratio of the number of foreground pixels incorrectly detected as background pixels by the algorithm to the number of background pixels in the ground truth

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}. \tag{68}$$

PWC is defined as the percentage of wrongly classified pixels

$$\mathrm{PWC} = \frac{\mathrm{FN} + \mathrm{FP}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{69}$$

Precision is defined as the ratio of the number of foreground pixels correctly detected by the algorithm to the total number of foreground pixels detected by the algorithm[79,94]

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{70}$$

F-measure (or $F1$ score) is a measure of quality that quantifies, in one scalar value for a frame, the similarity between the resulting foreground detection image and the ground truth.[59,100] Mathematically, it is a trade-off between recall and precision[185]

$$F1 = 2.\frac{\mathrm{Pr}.\mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}. \tag{71}$$

For comparison, we have adopted the same approach used on CDnet[109,116] to generate results. First, for each method, we computed all the metrics for each video in each category; then a category average metric was computed

$$M_c = \frac{1}{N_c}\sum_{v} M_{v,c}, \tag{72}$$

where M_c represents one of the seven metrics (Re, Sp, FPR, FNR, PWC, Pr, F1), N_c is the number of videos in each category, and v is a video in category c.

We also defined an overall average metric (OAM), which is the simple average of the category averages

$$\mathrm{OAM} = \frac{1}{C}\sum_{c=1}^{C} M_c, \tag{73}$$

where C is the number of categories.

To rank all the methods, for each category c, we computed the rank of each method for metric M. Then, we computed the average rank of this method across all the metrics

$$RM_c = \frac{1}{7}\sum_{n=1}^{7} \mathrm{rank}(M_c). \tag{74}$$

Subsequently, we computed the average over all categories to obtain the average rank across categories (RC) for each method

$$\mathrm{RC} = \frac{1}{C}\sum_{c=1}^{C} RM_c. \tag{75}$$

In addition, we computed the average rank across the OAM for each method

$$R = \frac{1}{7}\sum_{n=1}^{7} \mathrm{rank}(\mathrm{OAM}). \tag{76}$$

5 Results and Discussion

We applied each motion detection method described in Sec. 2 to the CDnet 2014 dataset,[131] which includes different scenarios and challenges. Figure 7 shows sample frames from each video in each category, whereas Fig. 8 shows their corresponding ground truths.

Each motion detection method was followed by an automatic thresholding operation in order to determine region changes and remove small changes in luminosity, except for the RGA, MoG, and MRFMD methods; for the two first methods, the threshold was fixed to 2.5σ, where σ denotes the standard deviation, and for MRFMD, a fixed threshold $\mathrm{Th} = 35$ was used to compute the observation

Fig. 7 Sample frames from each video in each category.

$O(x, y, t)$. We selected Otsu's thresholding method based on a previous study.[67]

For the Eig-Bg method, we set the number of training images to $N = 28$. These training images were equally spaced by 10 frames and the number of Eig-Bg vectors was set to $M = 3$.

For the MoG method, the parameters used were selected in accordance with the work of Nikolov et al.,[185] who measured the accuracy of the algorithm as a function of each variable parameter. Furthermore, they proposed a set of optimal parameters to improve the performance of the MoG algorithm. Accordingly, we selected the number of Gaussians as $K = 3$, the learning rate as $\alpha = 0.01$, the foreground threshold as $T = 0.25$, the deviation threshold as $D = 2.5$, and the initial standard deviation as $\sigma_{\text{init}} = 20$. Notice that the selected parameters are different from those presented

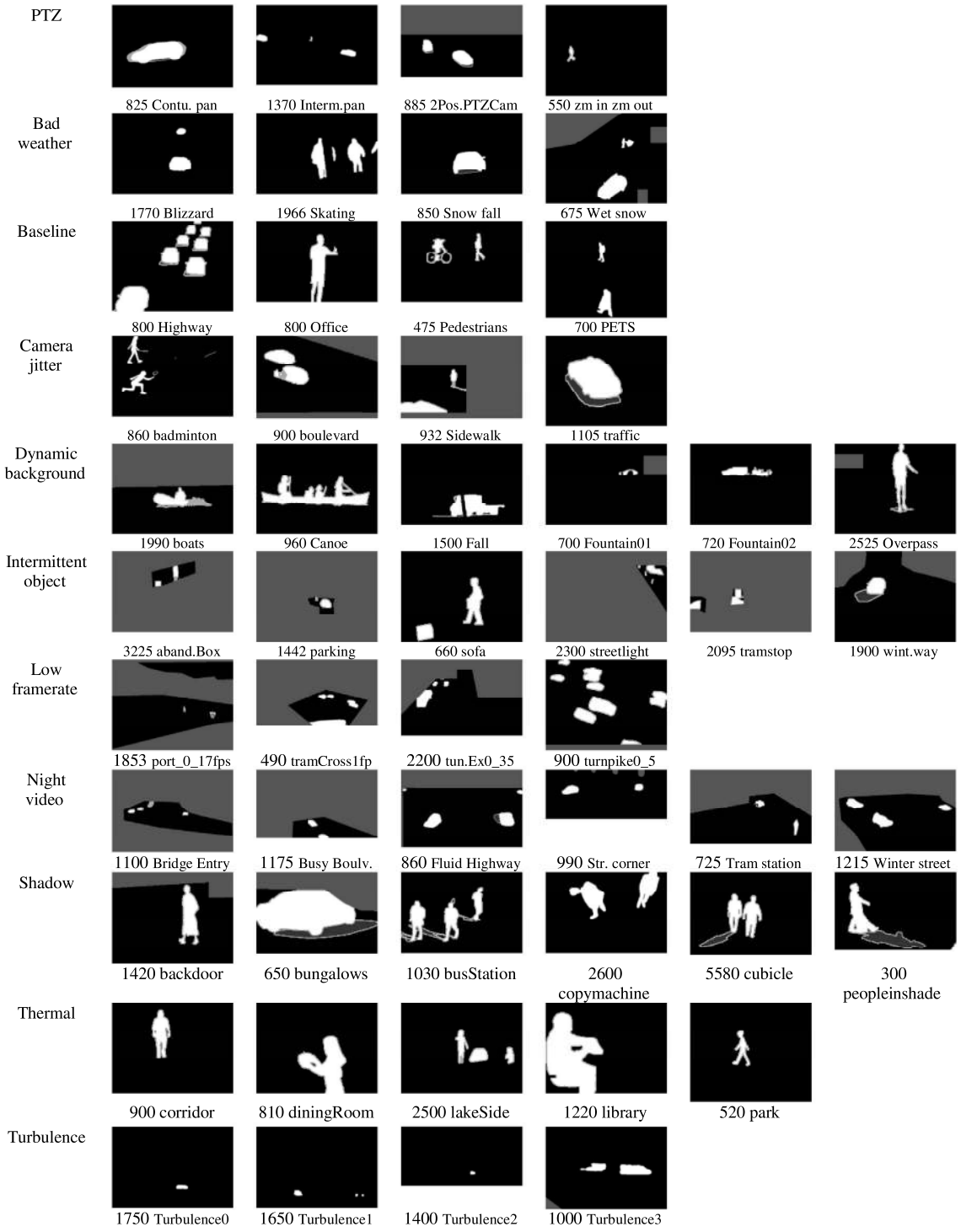

Fig. 8 Corresponding ground truths of the sample frames in Fig. 7.

in CDnet2014;[131] furthermore, we adopted the approximation of Power and Schoonees[177] to compute the learning rate ρ, hence the values of (μ_t, σ_t) were affected and different too.

For the RGA method, the learning rate was set to $\alpha = 0.01$ and the deviation threshold was set to $D = 2.5$. We note that MoG and RGA were applied to gray-level images in all our tests. We also applied the STEI and DSTEI methods to the gray-level images; to construct the spatio-temporal histogram, we selected a 3×3 window with five images and the number of gray levels was set as $Q = 100$. For the MRF-based motion detection algorithm, per the work of Caplier,[160] we set the four parameters to $\beta_s = 20$, $\beta_p = 10$, $\beta_f = 30$, $\alpha = 10$. For the $\Sigma\Delta$ method, the only parameter to be set was N; we selected $N = 3$ (typically, $N = 2$, 3, or 4).[158] For the simplified SOBS (Simp-SOBS), the method was

Table 2 Selected parameters for each method.

Method	Abbrev.	Parameters used
Temporal differencing[7,86]	FD	N/A
Three-frame difference[151,152]	3-FD	N/A
Running average filter[90,154]	RAF	$\alpha = 0.1$
Forgetting morphological temporal gradient[156]	FMTG	$\alpha = 0.1$
$\Sigma\Delta$ background estimation[157]	$\sum\Delta$	$N = 3$
MRF-based motion detection algorithm[159]	MRFMD	$\beta_s = 20$, $\beta_p = 10$, $\beta_f = 30$, $\alpha = 10$
Spatio-temporal entropy image[165]	STEI	$w \times w \times L = 3 \times 3 \times 5$, $Q = 100$
Difference-based STEI[166]	DSTEI	$w \times w \times L = 3 \times 3 \times 5$, $Q = 100$
Running Gaussian average[14]	RGA	$\alpha = 0.01$, $D = 2.5$
Mixture of Gaussians[15]	MoG	$\alpha = 0.01$, $T = 0.25$, $D = 2.5$
Eigen-background[42]	Eig-Bg	$N = 28$, $M = 3$
Simplified self-organized background subtraction[24]	Simp-SOBS	$\alpha_1 = 0.02$ and $\alpha_2 = 0.01$

tested on the HSV color space, where four parameters, Th, Th_v, α_1, and α_2, had to be set; Th and Th_v were set automatically using Otsu's method, and the learning rates were set to $\alpha_1 = 0.02$ and $\alpha_2 = 0.01$, according to Ref. 60. Furthermore, a median filter with $L = 10$ frames was used to initialize the weights of the neuronal map B_0. Finally, for the FMTG method and the adaptive background detection method, the parameter α should take values in the interval [0,1]. In our tests for these last two methods, we chose $\alpha = 0.1$. For FD and 3-FD, we applied these methods on grayscale images using Eq. (1) and Eqs. (5)–(8), respectively.

Table 2 summarizes the selected parameters for each tested method.

The overall results of testing these methods using the CDnet dataset (CDnet2012 and CD2014) are reported in Table 3, where entries are sorted by their average RC.

It is clear from Table 3 that the STEI method generated poor results compared to the other methods; the use of entropy alone as a metric to detect moving objects did not yield good results because the spatio-temporal accumulation window may contain object edges, which can lead to high diversity (high entropy) and thus impair the segmentation result. Moreover, this error can spread to the entire edge region (see Figs. 9 and 10), generating a very high PWC, high FPR and very low percentage of correctly classified background pixels (Sp). STEI is also very sensitive (high recall) due to low misses (FN, see Fig. 11).

Adding the FD to this method (DSTEI) increased its precision and decreased the PWC considerably (Figs. 12 and 13), except for the "camera jitter" category, which still has high FPR, PWC, and low F-measure (Figs. 13–15). Moreover, from the overall results in Table 3, we note that DSTEI did not achieve significant improvement over the FD method, owing to the drawbacks of using the spatio-temporal

Table 3 Overall results across all categories.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	R	R C
Simp-SOBS	0.49362	0.97220	0.02780	0.50638	4.67079	0.51477	0.40097	4.00000	**4.68831**
RGA	0.30123	0.99351	0.00649	0.69877	3.22117	0.49415	0.31465	**3.42857**	5.15584
GMM	0.20606	**0.99593**	0.00407	0.79394	**3.08499**	**0.61021**	0.25420	4.00000	5.44156
Eig-Bg	**0.59669**	0.93715	0.06285	**0.40331**	7.35814	0.41815	**0.41028**	6.57143	6.00000
RAF	0.36107	0.97060	0.02940	0.63893	5.25655	0.44158	0.27924	6.28571	6.05195
FMTG	0.42449	0.95736	0.04264	0.57551	6.37002	0.42101	0.28152	7.14286	6.20779
DSTEI	0.29669	0.96815	0.03185	0.70331	5.63380	0.41881	0.22299	8.14286	6.67532
FD	0.22779	0.97247	0.02753	0.77221	5.43072	0.46649	0.18825	6.85714	6.83117
3-FD	0.08117	0.98815	0.01185	0.91883	4.27114	0.46440	0.08201	8.00000	6.94805
MRFMD	0.08693	0.99056	**0.00944**	0.91307	4.02845	0.42689	0.09293	7.00000	7.37662
$\Sigma\Delta$	0.13762	0.98851	0.01149	0.86238	4.11532	0.37271	0.14229	7.42857	7.49351
STEI	0.45870	0.78646	0.21354	0.54130	22.18321	0.12255	0.12881	9.14286	9.12987

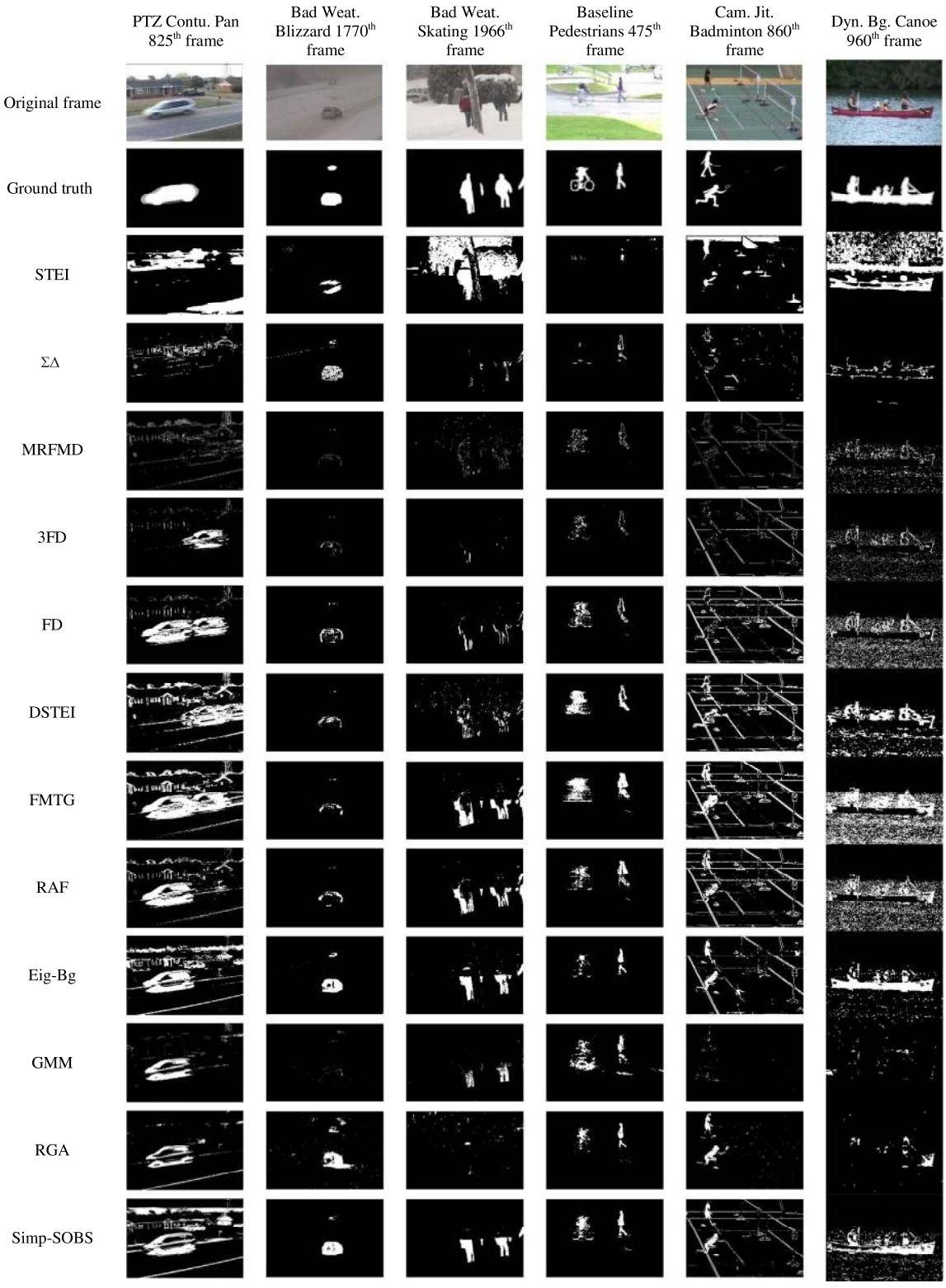

Fig. 9 Samples from the results of tested motion detection methods [pan tilit zoom (PTZ), Bad Wea., Baseline, Cam Jit., Dyn. Bg].

accumulation window and the tails caused by using inappropriate values of α to compute the spatio-temporal histogram recursively (see Figs. 9 and 10). Notably, DSTEI has acceptable percentage of correctly classified background pixels (Sp) in the "dynamic background" category, compared to other methods, see Fig. 16.

From Table 3, we can see that the ΣΔ method produced poor results but achieved significant improvement over the

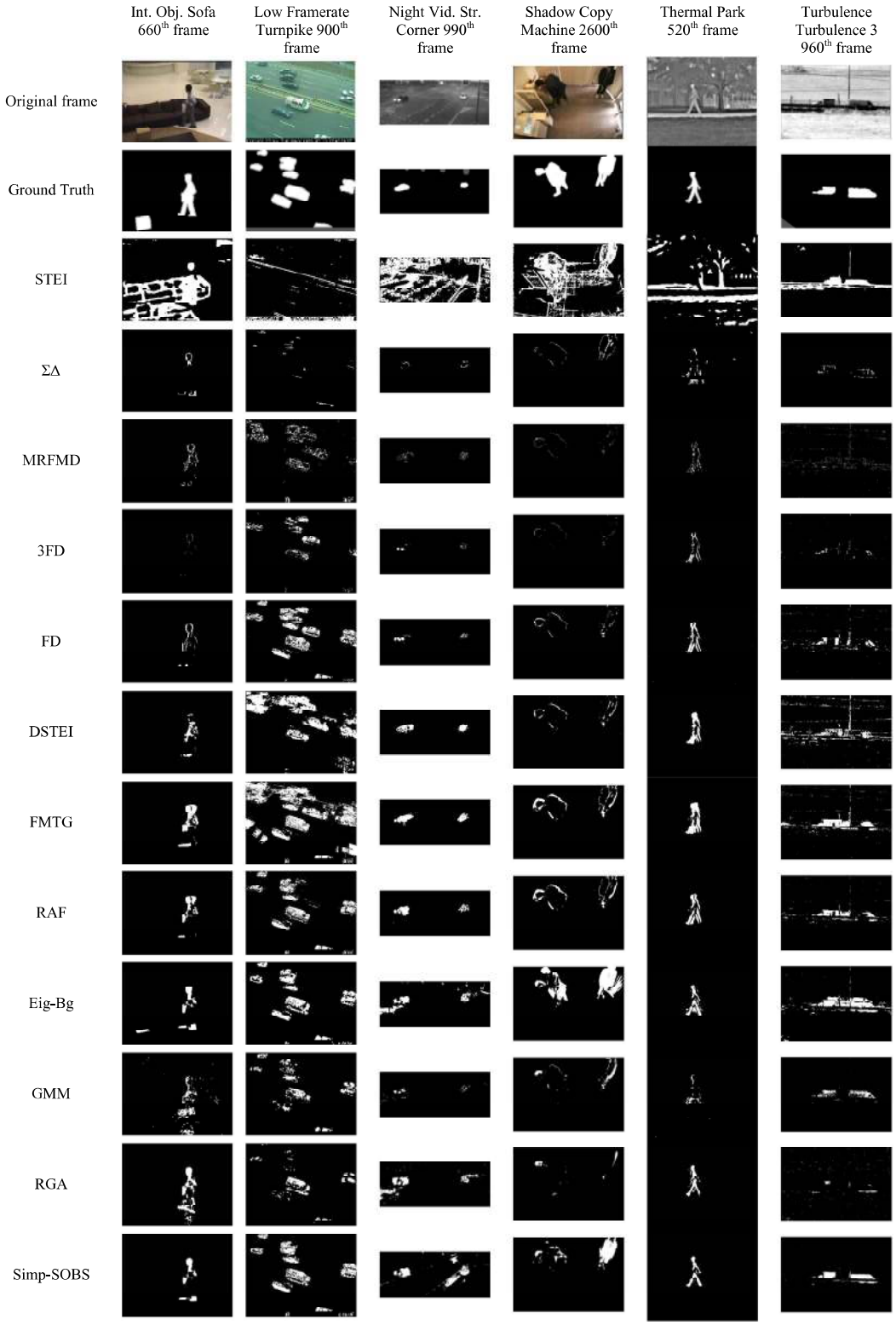

Fig. 10 Samples from the results of tested motion detection methods (Int. Obj., Low. Fr., N.Vid., Shad., Ther., Turb.).

STEI method. The ΣΔ method was characterized by a low FPR (Fig. 14) and high specificity (Fig. 16), i.e., many background pixels were correctly classified, but it still suffered from a high FNR, especially in "PTZ," "camera jitter," and "thermal" categories, and also low precision in "PTZ," "bad weather," "dynamic background," "shadow," and "thermal" categories (See Tables 4, 5, 6, and 7). In Fig. 9, we observe that false detections caused by snowfall in the "skating"

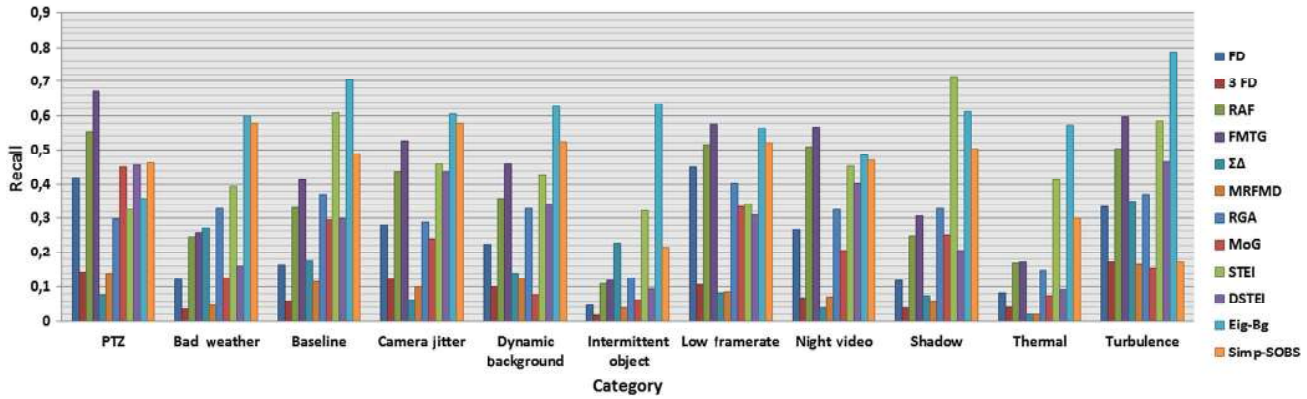

Fig. 11 Recall results for all tested methods over all categories.

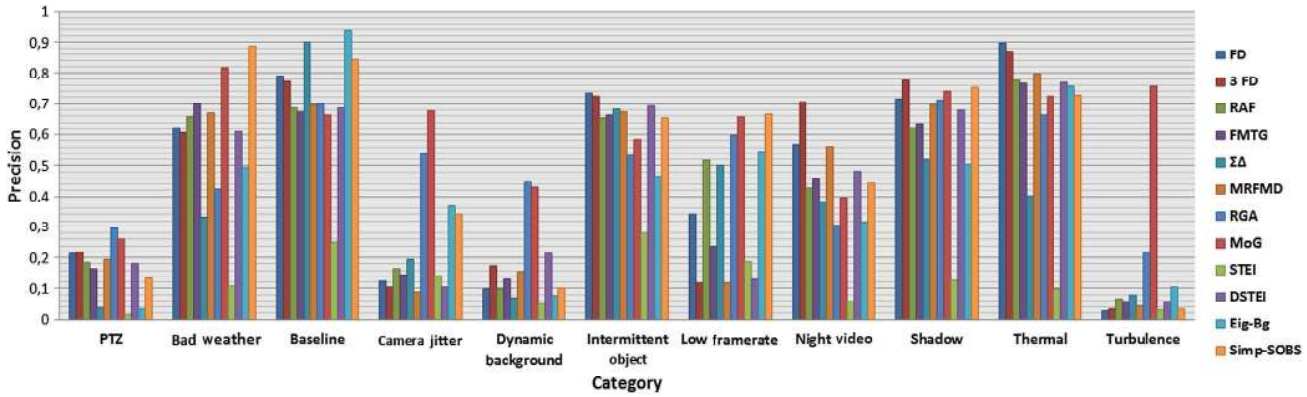

Fig. 12 Precision results for all tested methods over all categories.

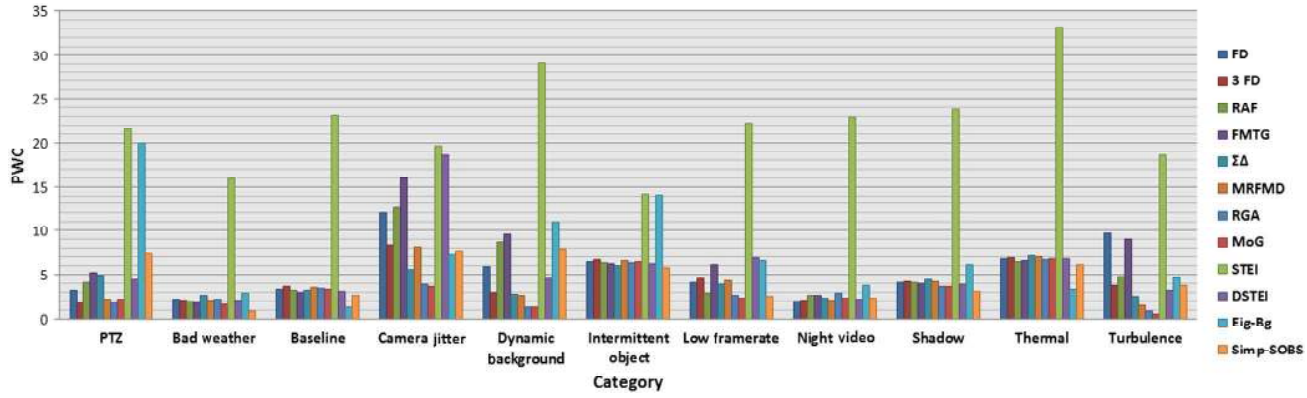

Fig. 13 PWC results for all tested methods over all categories.

video were eliminated, whereas the objects in motion were not well segmented. We note also that this method has a high precision in the case of "baseline" category, Fig. 12; however, results on this category, Fig. 15, show holes left in the segmented objects (high misses) which produce a low recall (Fig. 11 and Table 8), and this problem is owed to the initialization step based on temporal differencing.

The MRFMD method also yielded poor results and was similar to the ΣΔ method, with a high FNR and low recall (Figs. 11 and 17), due to its dependence on the initialization step based on the temporal differencing method, leading to incompletely segmented moving objects (holes). Nevertheless, this method performed well by enhancing the image difference in terms of specificity (Fig. 16), and PWC (Fig. 13) has acceptable precision in the cases of "bad weather," "shadow," "night video," and "thermal" (see Fig. 12 and Tables 5, 9–11).

In addition, this method is parametric because we had to define the values of the model energy $(\beta_s, \beta_p, \beta_f, \alpha)$. Furthermore, the iterative conditional mode technique is

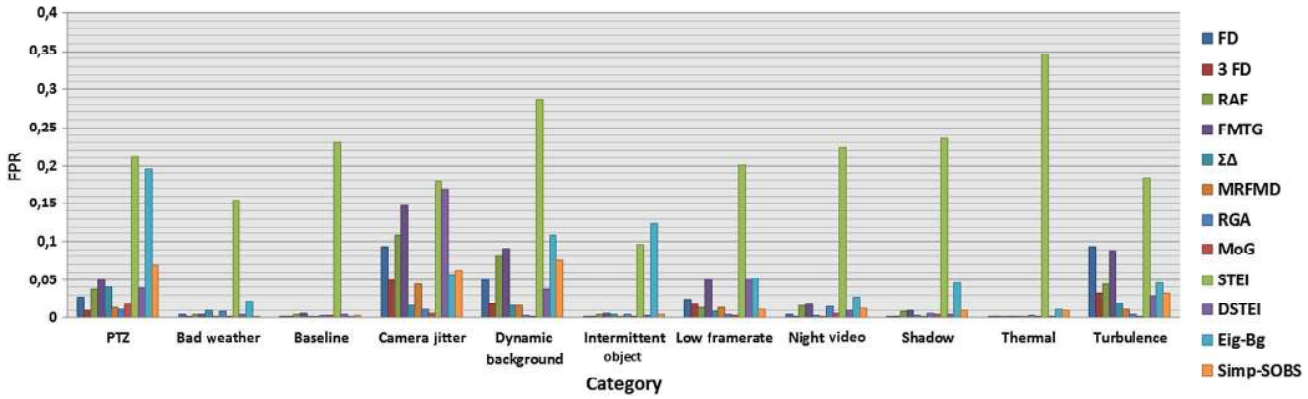

Fig. 14 FPR results for all tested methods over all categories.

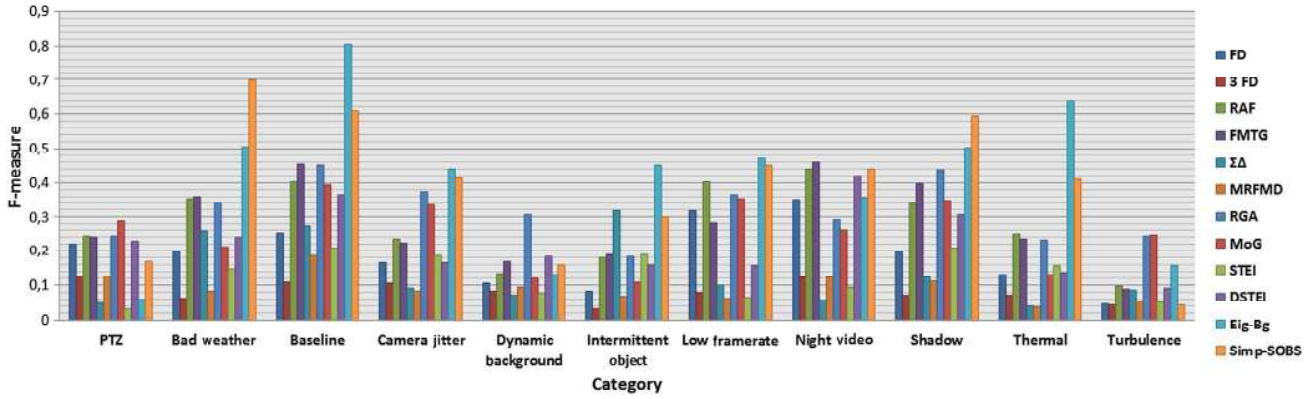

Fig. 15 *F*-measure results for all tested methods over all categories.

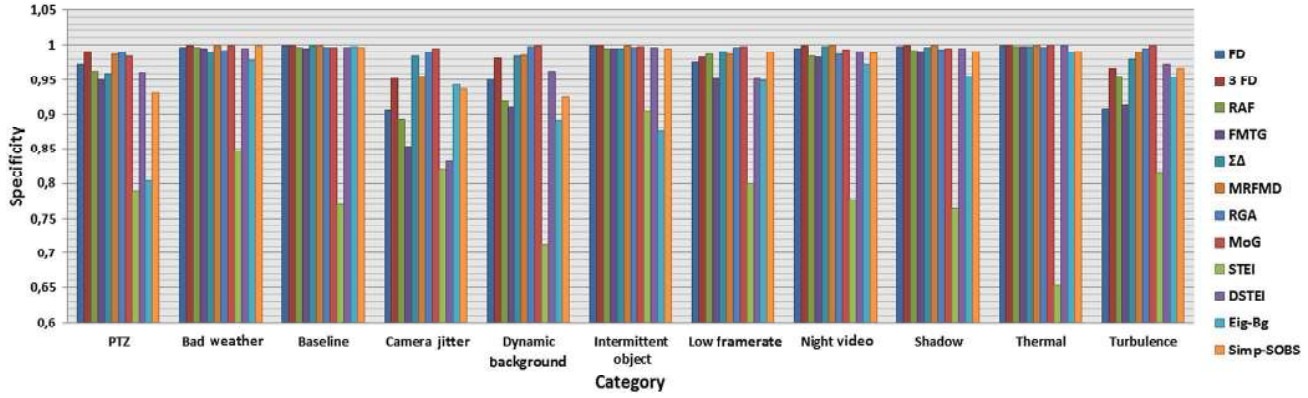

Fig. 16 Specificity results for all tested methods over all categories.

a suboptimal algorithm that may converge to local minima, but its computational time is considerably shorter than that of a stochastic relaxation scheme (i.e., simulated annealing).[159]

As expected, the FD and 3-FD methods did not show good results, except for high precision, mainly because of the incomplete segmentation of the shape of moving objects, preserving only the edges. Moreover, these methods suffered from overlap of slow moving objects and poor detection of objects far from the camera. To overcome the problem of incomplete segmentation, the threshold operation in the FD method is usually followed by morphological operations to link the edges of the moving objects. Then, regions and holes in the image are filled. Another solution is to combine FD methods with a background subtraction method.[5,186] We also note that these methods demonstrate good detection of foreground pixels in night-time videos compared to background subtraction techniques, see Fig. 12 and Table 9,

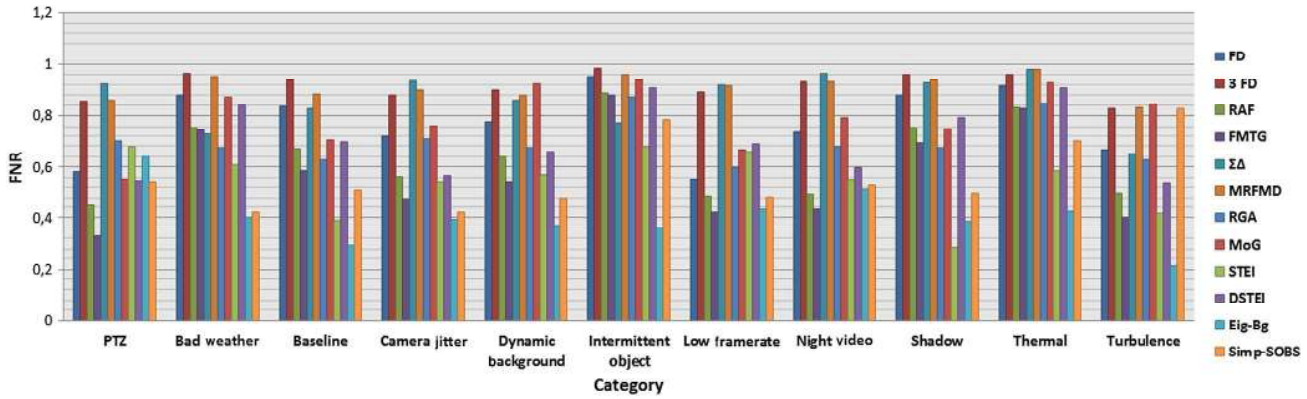

Fig. 17 FNR results for all tested methods over all categories.

owing to the change of light (vehicle or street light), which impairs the modeled background.

The FMTG method yielded acceptable results with observable low FNR (Fig. 17) and high recall (Fig. 11) for "PTZ," "camera jitter," "dynamic background," "low frame rate," "night video," and "turbulence" categories (Tables 12 and 13). However, it was characterized by high FPR, caused by artificial tails due to an inappropriate value of α. Moreover, this method was conducive to strong background motion, as in the "dynamic background" category, Fig. 9.

As with FMTG, the RAF method yielded acceptable results. This result was expected because FMTG is based on a recursive operation of the RAF, i.e., Eq. (10). However, FMTG gave good detection results in bad weather conditions (see Figs. 12 and 15, Table 5) owing to the morphological temporal gradient filter, Eq. (15). Compared to FD and 3-FD, the RAF method outperforms them for almost all categories (except night videos) in terms of F-measure, FNR, and sensitivity, but with high FPR.

The fourth-best method according to the evaluation results (Table 3) was the Eig-Bg method, which yielded good detection results as well as silhouettes of objects. We noted that it had a high recall among all categories, Fig. 11 (except "low frame rate" category), and low FPR, Fig. 14. For "intermittent object" and "PTZ" categories, the Eig-Bg shows high FPR and PWC (Figs. 13, 14 and Tables 4, 14), and this is due to the absence of an update process in the original algorithm. Different techniques have been developed to resolve this problem; to this end we refer the reader to these Refs. 43, 69, 81, and 181. Remarkably, this method had a great precision and F-measure for the "baseline" category, which makes this method the ideal approach for easy and mild challenging

Table 4 Pixel-based evaluation of different motion detection methods applied to the "PTZ" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
GMM	0.44955	0.98300	0.01700	0.55045	2.18306	0.26124	**0.28955**	**3.57143**
RGA	0.29935	0.98946	0.01054	0.70065	**1.71521**	**0.29488**	0.24364	3.71429
RAF	0.55135	0.96246	0.03754	0.44865	4.10254	0.18547	0.24236	4.42857
3-FD	0.14254	**0.99055**	**0.00945**	0.85746	1.74923	0.21983	0.12676	5.00000
FD	0.42000	0.97292	0.02708	0.58000	3.23362	0.21225	0.22099	5.28571
FMTG	**0.67296**	0.94991	0.05009	**0.32704**	5.22171	0.16603	0.24073	5.85714
DSTEI	0.45699	0.96035	0.03965	0.54301	4.42049	0.18210	0.22979	5.85714
MRFMD	0.13974	0.98662	0.01338	0.86026	2.14460	0.19404	0.12624	6.42857
Simp-SOBS	0.46153	0.93149	0.06851	0.53847	7.29140	0.13466	0.16815	7.42857
Eig-Bg	0.35897	0.80386	*0.19614*	0.64103	19.98432	0.03295	0.05791	9.71429
ΣΔ	*0.07621*	0.95886	0.04114	*0.92379*	4.92846	0.03709	0.04640	9.85714
STEI	0.32390	*0.78885*	0.21115	0.67610	*21.53317*	*0.01634*	*0.03047*	*10.85714*

Table 5 Pixel-based evaluation of different motion detection methods applied to the "bad weather" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	*F*-measure	RM_c
Simp-SOBS	0.58117	0.99862	0.00138	**0.41883**	**0.87063**	**0.88652**	**0.69965**	**2.14286**
GMM	0.12651	**0.99966**	**0.00034**	0.87349	1.67592	0.81834	0.21127	4.57143
FMTG	0.25595	0.99546	0.00454	0.74405	1.89175	0.70015	0.35741	5.00000
RAF	0.24383	0.99563	0.00437	0.75617	1.94594	0.65861	0.35216	5.57143
Eig-Bg	**0.60037**	0.97762	0.02238	0.39963	2.91987	0.49428	0.50623	6.57143
MRFMD	0.04452	0.99863	0.00137	0.95548	2.07388	0.67153	0.08175	6.85714
RGA	0.32721	0.99209	0.00791	0.67279	2.19833	0.42407	0.33892	7.14286
FD	0.12061	0.99620	0.00380	0.87939	2.18648	0.62279	0.19972	7.57143
DSTEI	0.15961	0.99509	0.00491	0.84039	2.11153	0.61012	0.24112	7.57143
3-FD	*0.03300*	0.99895	0.00105	*0.96700*	2.09177	0.60550	*0.06235*	7.71429
$\Sigma\Delta$	0.27268	0.98994	0.01006	0.72732	2.61408	0.33206	0.25832	8.14286
STEI	0.39261	*0.84577*	*0.15423*	0.60739	*15.89976*	*0.10713*	0.14777	*9.14286*

scenes. However, its results were strongly dependent on the images that form the eigenspace; the presence of moving objects in this space could alter the detection results. The execution time of this method depended on the number of eigenvectors. In our case, the execution time seemed acceptable for real-time application; however, memory requirements make it unsuitable for this type of application.

Methods based on Gaussian distributions showed better performance than other methods. The GMM method is characterized by its very high precision and high $F1$-score in difficult challenging categories ("bad weather, dynamic background, camera jitter, low frame rate, shadow, and turbulence"), and has low PWCs and low FPR in all categories; this is due to the number of K Gaussians used to model the

Table 6 Pixel-based evaluation of different motion detection methods applied to the "camera jitter" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	*F*-measure	RM_c
Eig-Bg	**0.6076**	0.9421	0.0579	**0.3924**	7.1837	0.3663	**0.4391**	**3.1429**
RGA	0.2905	0.9894	0.0106	0.7095	3.8901	0.5399	0.3741	3.5714
GMM	0.2374	**0.9944**	**0.0056**	0.7626	**3.6016**	**0.6779**	0.3344	3.7143
Simp-SOBS	0.5808	0.9373	0.0627	0.4192	7.7127	0.3411	0.4147	4.1429
RAF	0.4378	0.8926	0.1074	0.5622	12.6178	0.1664	0.2351	6.8571
FMTG	0.5246	0.8533	0.1467	0.4754	16.0025	0.1457	0.2236	7.0000
$\Sigma\Delta$	*0.0637*	0.9837	0.0163	*0.9363*	5.5428	0.1953	0.0898	7.0000
FD	0.2809	0.9061	0.0939	0.7191	12.0018	0.1227	0.1668	8.1429
3-FD	0.1188	0.9513	0.0487	0.8812	8.4001	0.1040	0.1065	8.2857
STEI	0.4609	*0.8205*	*0.1795*	0.5391	*19.6615*	0.1436	0.1872	8.4286
MRFMD	0.0987	0.9554	0.0446	0.9013	8.1616	*0.0888*	*0.0824*	8.5714
DSTEI	0.4354	0.8314	0.1686	0.5646	18.5179	0.1041	0.1648	*9.1429*

dynamic backgrounds. Notably, the RGA method outperforms the MoG method (Table 3), with higher recall (Fig. 11), low FNR values (Fig. 17), and higher F-measure (Fig. 15) in almost all categories. This is owed to the large number of parameters required to set for the MoG algorithm $(K, \alpha, T, D, \sigma)$, which differ with the challenging conditions presented by a video (day/night, indoor/outdoor, complex/simple background, with/without noise). Thus, in some cases, the RGA method seemed to be sufficient. Moreover, its computational complexity was lower than that of the MoG method, as was found by Piccardi[4] and Benezeth et al.[97]

Table 7 Pixel-based evaluation of different motion detection methods applied to the "dynamic background" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
RGA	0.32604	0.99739	**0.00261**	0.67396	**1.25771**	**0.44585**	**0.30555**	**3.00000**
DSTEI	0.34040	0.96264	0.03736	0.65960	4.53876	0.21356	0.18193	5.00000
GMM	*0.07816*	**0.99884**	0.00116	0.92184	1.28203	0.43209	0.11900	5.28571
Simp-SOBS	0.52337	0.92515	0.07485	0.47663	7.89821	0.10130	0.16129	5.57143
MRFMD	0.11941	0.98458	0.01542	0.88059	2.65240	0.15423	0.09444	6.14286
FMTG	0.46085	0.90959	0.09041	0.53915	9.53249	0.12996	0.16795	6.42857
RAF	0.35967	0.91939	0.08061	0.64033	8.73891	0.09714	0.13346	7.14286
3-FD	0.09905	0.98135	0.01865	*0.90095*	2.95550	0.17437	0.08254	7.28571
Eig-Bg	**0.62761**	0.89217	0.10783	**0.37239**	11.02858	0.07684	0.13219	7.28571
ΣΔ	0.13895	0.98350	0.01650	0.86105	2.75199	0.07037	*0.07060*	7.57143
FD	0.22397	0.95006	0.04994	0.77603	5.86099	0.09671	0.10502	7.71429
STEI	0.42841	*0.71230*	*0.28770*	0.57159	*29.06152*	*0.05091*	0.07804	*9.57143*

Table 8 Pixel-based evaluation of different motion detection methods applied to the "baseline" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
Eig-Bg	**0.70521**	0.99832	0.00168	**0.29479**	**1.32715**	**0.93552**	**0.80366**	**2.14286**
Simp-SOBS	0.48942	0.99649	0.00351	0.51058	2.63660	0.84190	0.60852	3.85714
ΣΔ	0.17359	0.99946	0.00054	0.82641	3.23513	0.90046	0.27642	5.42857
RGA	0.37160	0.99672	0.00328	0.62840	3.39816	0.69864	0.44997	5.85714
FMTG	0.41467	0.99483	0.00517	0.58533	3.00519	0.67255	0.45341	6.57143
FD	0.16392	0.99883	0.00117	0.83608	3.35138	0.79205	0.25216	6.71429
RAF	0.33089	0.99618	0.00382	0.66911	3.21227	0.68705	0.40236	7.14286
DSTEI	0.30211	0.99629	0.00371	0.69789	3.09809	0.68536	0.36369	7.42857
3-FD	*0.06106*	**0.99947**	**0.00053**	*0.93894*	3.61961	0.77192	*0.10796*	7.71429
GMM	0.29563	0.99640	0.00360	0.70437	3.30918	0.66375	0.39360	8.00000
MRFMD	0.11522	0.99872	0.00128	0.88478	3.51801	0.69588	0.18603	8.28571
STEI	0.60964	*0.76918*	*0.23082*	0.39036	*23.14295*	*0.25211*	0.20983	*8.85714*

Finally, Simp-SOBS showed the best results of all the methods, with high precision, recall, and F-measure, owing to the use of the HSV color space and the condition on the V value component that significantly reduced object shadows. Furthermore, we note from Table 15 that the results of this method in challenging categories ("bad weather," "low frame rate," and "shadow") were as good as those in simple categories ("baseline"). From the previous results, we can note that the most challenging categories for this method are: the "turbulence" category characterized by low precision

Table 9 Pixel-based evaluation of different motion detection methods applied to the "night video" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
FD	0.26524	0.99547	0.00453	0.73476	**2.01185**	0.56769	0.35088	**4.71429**
DSTEI	0.40097	0.99043	0.00957	0.59903	2.24922	0.48116	0.42043	5.14286
3-FD	0.06949	**0.99921**	**0.00079**	0.93051	2.03676	**0.70609**	0.12229	5.28571
FMTG	**0.56808**	0.98238	0.01762	**0.43192**	2.63505	0.45533	**0.46018**	5.28571
Simp-SOBS	0.47254	0.98834	0.01166	0.52746	2.29129	0.44207	0.43995	5.28571
MRFMD	0.07002	0.99876	0.00124	0.92998	2.07952	0.56291	0.12237	5.57143
RAF	0.50793	0.98401	0.01599	0.49207	2.63158	0.42673	0.43843	5.71429
GMM	0.20595	0.99388	0.00612	0.79405	2.26189	0.39635	0.26142	7.00000
Eig-Bg	0.48550	0.97250	0.02750	0.51450	3.77828	0.31558	0.35541	7.71429
ΣΔ	*0.03625*	0.99749	0.00251	*0.96375*	2.27967	0.38100	*0.05685*	8.14286
RGA	0.32350	0.98592	0.01408	0.67650	2.84418	0.30709	0.29126	8.28571
STEI	0.45413	*0.77635*	*0.22365*	0.54587	*22.85890*	*0.05566*	0.09322	*9.85714*

Table 10 Pixel-based evaluation of different motion detection methods applied to the "shadow" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
Simp-SOBS	0.50198	0.99044	0.00956	0.49802	**3.07260**	0.75207	**0.59353**	**4.28571**
RGA	0.32784	0.99372	0.00628	0.67216	3.61823	0.71265	0.43611	4.57143
GMM	0.24914	0.99518	0.00482	0.75086	3.61844	0.74109	0.34635	5.00000
FD	0.11852	0.99764	0.00236	0.88148	4.08282	0.71392	0.19762	6.14286
DSTEI	0.20761	0.99541	0.00459	0.79239	3.92670	0.67913	0.30582	6.28571
FMTG	0.30569	0.99083	0.00917	0.69431	3.94654	0.63377	0.39529	6.42857
3-FD	*0.03776*	**0.99962**	**0.00038**	*0.96224*	4.24996	**0.77620**	*0.07113*	6.85714
Eig-Bg	0.61292	0.95488	0.04512	0.38708	6.08321	0.50480	0.50255	7.14286
MRFMD	0.06100	0.99894	0.00106	0.93900	4.24903	0.69451	0.11048	7.28571
RAF	0.24641	0.99171	0.00829	0.75359	4.09855	0.62128	0.33949	7.42857
ΣΔ	0.07324	0.99648	0.00352	0.92676	4.39514	0.51865	0.12251	8.28571
STEI	**0.71537**	*0.76417*	*0.23583*	**0.28463**	*23.76070*	*0.12635*	0.20898	*8.28571*

(Fig. 12) and high FNR (Fig. 17), and the "PTZ" category characterized by high PWC (Fig. 13) and high FPR (Fig. 14).

Figure 18 shows the frame rate (execution time) of each method, applied on "PETS2006" video from the "baseline" category, with a resolution of 720×576. Tests were carried on Intel I7 2.3 GHz with 16 GB RAM, and parts of the code were nonvectorized. This figure shows that the fastest methods were DF, 3-FD, RAF, and FMTG because of their

Table 11 Pixel-based evaluation of different motion detection methods applied to the "thermal" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	*F*-measure	RM_c
Eig-Bg	**0.57527**	0.98983	0.01017	**0.42473**	**3.34477**	0.75660	**0.63940**	**4.71429**
RAF	0.16894	0.99785	0.00215	0.83106	6.46159	0.77678	0.24840	4.85714
FD	0.08293	0.99952	0.00048	0.91707	6.76412	**0.89736**	0.13254	5.28571
FMTG	0.17280	0.99771	0.00229	0.82720	6.53223	0.76712	0.23493	5.42857
Simp-SOBS	0.30052	0.99103	0.00897	0.69948	6.14208	0.72833	0.40935	5.42857
DSTEI	0.09116	0.99884	0.00116	0.90884	6.80556	0.77026	0.13743	6.00000
3-FD	0.03946	0.99975	0.00025	0.96054	6.90147	0,.87329	0.06998	6.42857
RGA	0.14817	0.99652	0.00348	0.85183	6.69029	0.66375	0.23215	7.14286
MRFMD	**0.01977**	**0.99979**	**0.00021**	*0.98023*	6.97932	0.79833	*0.03642*	7.28571
GMM	0.07448	0.99857	0.00143	0.92552	6.79791	0.72397	0.13242	7.57143
STEI	0.41382	*0.65387*	*0.34613*	0.58618	*33.18293*	*0.09735*	0.15688	8.28571
ΣΔ	0.02021	0.99792	0.00208	0.97979	7.12201	0.40127	0.03848	*9.57143*

Table 12 Pixel-based evaluation of different motion detection methods applied to the "low frame rate" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	*F*-measure	RM_c
Simp-SOBS	0.51954	0.98959	0.01041	0.48046	2.56486	**0.66767**	0.44959	**2.71429**
RGA	0.40264	0.99610	0.00390	0.59736	2.58883	0.59523	0.36304	3.71429
GMM	0.33327	**0.99736**	**0.00264**	0.66673	**2.31369**	0.65832	0.35273	3.71429
RAF	0.51335	0.98607	0.01393	0.48665	2.90861	0.51617	0.40144	4.57143
Eig-Bg	0.56639	0.94749	0.05251	0.43361	6.54513	0.54566	**0.46997**	5.85714
FD	0.45044	0.97505	0.02495	0.54956	4.09714	0.34237	0.31730	6.42857
FMTG	**0.57796**	0.95039	0.04961	**0.42204**	6.06386	0.23980	0.28403	6.57143
ΣΔ	*0.08207*	0.99134	0.00866	*0.91793*	3.90652	0.50163	0.10016	7.14286
MRFMD	0.08464	0.98657	0.01343	0.91536	4.31021	0.11816	*0.06137*	8.85714
3-FD	0.10544	0.98281	0.01719	0.89456	4.58337	*0.11600*	0.08011	9.14286
DSTEI	0.31088	0.95063	0.04937	0.68912	6.88456	0.13012	0.15748	9.28571
STEI	0.34130	*0.79972*	*0.20028*	0.65870	*22.17830*	0.18730	0.06510	*10.00000*

simplicity. Eig-Bg shows also a good execution time, and this is interpreted by the use of a subset of singular values and vectors, which overcomes the long time required to compute the N eigenvectors using eigendecomposition. ΣΔ has a slow frame rate owing to the postprocessing step required to eliminate the ghost effect. The slowest methods were MoG, STEI, and DSTEI; MOG is slow because of the computing complexity linked to the use of K Gaussian distributions,

Table 13 Pixel-based evaluation of different motion detection methods applied to the "turbulence" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
RGA	0.37053	0.99522	0.00478	0.62947	0.89475	0.22000	0.24396	**3.42857**
GMM	*0.15444*	**0.99975**	**0.00025**	*0.84556*	**0.46990**	**0.75562**	**0.24731**	4.14286
Eig-Bg	**0.78752**	0.95435	0.04565	**0.21248**	4.71277	0.10395	0.15637	4.71429
Simp-SOBS	0.78287	0.95072	0.04928	0.21713	5.07985	0.11308	0.16392	5.28571
DSTEI	0.46520	0.97167	0.02833	0.53480	3.17950	0.05695	0.09098	5.71429
ΣΔ	0.34950	0.97966	0.02034	0.65050	2.48953	0.07998	0.08646	5.85714
RAF	0.50183	0.95535	0.04465	0.49817	4.79166	0.06753	0.09773	6.14286
MRFMD	0.16698	0.98938	0.01062	0.83302	1.56510	0.04329	0.05261	7.00000
FMTG	0.59875	0.91203	0.08797	0.40125	9.05480	0.05586	0.08904	7.28571
3-FD	0.17101	0.96715	0.03285	0.82899	3.76600	0.03516	*0.04327*	8.57143
STEI	0.58499	*0.81651*	*0.18349*	0.41501	*18.58043*	0.03174	0.05193	9.28571
FD	0.33346	0.90664	0.09336	0.66654	9.71757	*0.02896*	0.04452	*10.57143*

Table 14 Pixel-based evaluation of different motion detection methods applied to the "intermittent object" category.

	Recall	Specificity	FPR	FNR	PWC	Precision	F-measure	RM_c
ΣΔ	0.22742	0.99530	0.00470	0.77258	6.00318	0.68200	**0.31732**	**4.28571**
Simp-SOBS	0.21610	0.99506	0.00494	0.78390	**5.81846**	0.65376	0.30207	5.42857
DSTEI	0.09327	0.99691	0.00309	0.90673	6.23954	0.69406	0.15943	5.85714
RGA	0.12612	0.99609	0.00391	0.87388	6.33705	0.53360	0.18253	6.28571
FD	0.04566	0.99873	0.00127	0.95434	6.43023	**0.73456**	0.08313	6.42857
FMTG	0.11714	0.99452	0.00548	0.88286	6.18409	0.66480	0.19018	6.42857
RAF	0.10976	0.99541	0.00459	0.89024	6.31262	0.65421	0.18066	6.71429
Eig-Bg	**0.63616**	*0.87557*	*0.12443*	**0.36384**	14.03169	0.46714	0.45021	7.00000
3-FD	*0.01516*	**0.99954**	0.00046	*0.98484*	6.62883	0.72605	*0.02929*	7.14286
MRFMD	0.03628	0.99878	**0.00122**	0.96372	6.57922	0.67419	0.06807	7.28571
GMM	0.06211	0.99820	0.00180	0.93789	6.42132	0.58364	0.10818	7.28571
STEI	0.32070	0.90380	0.09620	0.67930	*14.15520*	*0.27958*	0.18753	*7.85714*

Table 15 Top three methods for all categories based on the average RC.

	First	Second	Third
PTZ	MoG	RGA	RAF
Bad weather	Simp-SOBS	MoG	FMTG
Baseline	Eig-Bg	Simp-SOBS	ΣΔ
Camera jitter	Eig-Bg	RGA	MoG
Dynamic background	RGA	DSTEI	MoG
Intermittent object motion	ΣΔ	Simp-SOBS	DSTEI
Low frame rate	Simp-SOBS	RGA	MoG
Night video	FD	3-FD	DSTEI
Shadow	Simp-SOBS	RGA	MoG
Thermal	Eig-Bg	RAF	FD
Turbulence	RGA	MoG	Eig-Bg

the time required to update their parameters, and to order them; and the slowness of STEI and DSTEI is more closely related to the time needed to compute the histogram from the spatio-temporal window. MRF, Simp-SOBS, and RGA also had long execution times, but they were not as slow as the previous methods.

6 Conclusion

In this paper, we review and compare motion detection methods using one of the most recent, complete, and challenging datasets: CDnet2012 and CDnet2014. Detailed pixel evaluation was performed using different metrics to enable a user to determine the appropriate method for his or her needs.

From the results reported, we can conclude that there is no ideal method for all situations; each method performs well in some cases and fails in others. However, it is worth mentioning here that methods based on FDs are not really designed to detect a complete silhouette of the moving object and thus are underrated here; they aim to detect motion and typically must be combined with other methods to achieve full segmentation. If we had to choose two methods based on the different challenging categories, they would be the Simp-SOBS[24] and the RGA[14] methods; the former for the "bad weather," "baseline," "intermittent object motion," "low frame rate," "night video," "shadow," and "thermal" categories, and the latter for the "PTZ," "camera jitter," "dynamic background," and "turbulence" categories. This choice is justified by the high ranks of the methods for nearly all categories based on the average RC as well as their acceptable execution times. The good performance of Simp-SOBS can be explained by the simple but powerful competitive learning used in SOM with an appropriate HSV color space that separates chromaticity from brightness information. The surprising results of RGA are linked to its low complexity with only a few parameters to adjust (e.g., compared to MoG[15]) that was sufficient for most areas of the images tested (only small image portions would have required more complex methods such as MoG), because the whole image background is not always dynamic except for the bad weather condition or

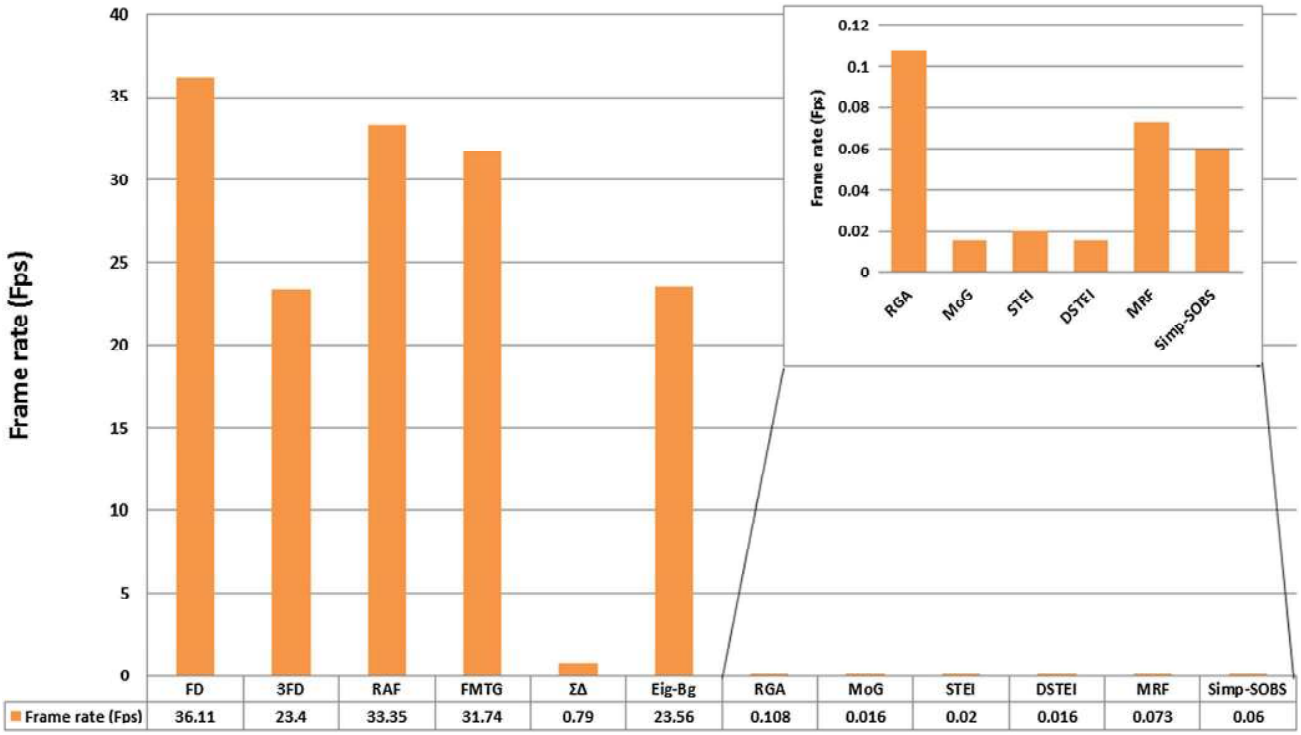

Fig. 18 Computational time for each method (presented in frames per second).

maritime applications, where we found that the MoG is superior to the RGA in the tested categories). In the future, we will test other methods in order to expand the scope of this study and provide users with a complete benchmark of motion detection methods.

References

1. I. S. Kim et al., "Intelligent visual surveillance—a survey," *Int. J. Control Autom. Syst.* **8**(5), 926–939 (2010).
2. M. Paul, S. M. Haque, and S. Chakraborty, "Human detection in surveillance videos and its applications—a review," *EURASIP J. Adv. Signal Process.* **2013**(1), 176 (2013).
3. W. Hu et al., "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst. Man Cybern. Part C* **34**(3), 334–352 (2004).
4. M. Piccardi, "Background subtraction techniques: a review," in *2004 IEEE Int. Conf. on Systems, Man and Cybernetics*, Vol. 4, pp. 3099–3104 (2004).
5. D. A. Migliore, M. Matteucci, and M. Naccari, "A revaluation of frame difference in fast and robust motion detection," in *Proc. of the 4th ACM Int. Workshop on Video Surveillance and Sensor Networks*, pp. 215–218 (2006).
6. S. Yalamanchili, W. N. Martin, and J. K. Aggarwal, "Extraction of moving object descriptions via differencing," *Comput. Graph. Image Process.* **18**(2), 188–201 (1982).
7. R. Jain, W. Martin, and J. Aggarwal, "Segmentation through the detection of changes due to motion," *Comput. Graph. Image Process.* **11**(1), 13–34 (1979).
8. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, California, pp. 674–679 (1981).
9. B. K. Horn and B. G. Schunck, "Determining optical flow: a retrospective," *Artif. Intell.* **59**(1–2), 81–87 (1993).
10. T. Bouwmans, "Recent advanced statistical back ground modeling for foreground detection-a systematic survey," *Recent Pat. Comput. Sci.* **4**(3), 147–176 (2011).
11. W.-X. Kang, W.-Z. Lai, and X.-B. Meng, "An adaptive background reconstruction algorithm based on inertial filtering," *Optoelectron. Lett.* **5**(6), 468–471 (2009).
12. S. Jiang and Y. Zhao, "Background extraction algorithm base on Partition Weighed Histogram," in *3rd IEEE Int. Conf. on Network Infrastructure and Digital Content (IC-NIDC 2012)*, pp. 433–437 (2012).
13. J. Zheng et al., "Extracting roadway background image: mode-based approach," *Transp. Res. Rec.* **1944**, 82–88 (2006).
14. C. R. Wren et al., "Pfinder: real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7), 780–785 (1997).
15. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 246–252 (1999).
16. A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European Conf. on Computer Vision*, pp. 751–767 (2000).
17. F. El Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of Gaussians model: application to background modeling," in *Int. Symp. on Visual Computing*, pp. 772–781 (2008).
18. M. H. Sigari, N. Mozayani, and H. Pourreza, "Fuzzy running average and fuzzy background subtraction: concepts and application," *Int. J. Comput. Sci. Network Secur.* **8**(2), 138–143 (2008).
19. K. Kim et al., "Real-time foreground–background segmentation using codebook model," *Real-Time Imaging* **11**(3), 172–185 (2005).
20. M. Xiao, C. Han, and X. Kang, "A background reconstruction for dynamic scenes," in *2006 9th Int. Conf. on Information Fusion*, pp. 1–7 (2006).
21. L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.* **17**(7), 1168–1177 (2008).
22. L. Maddalena and A. Petrosino, "The SOBS algorithm: what are the limits?," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW 2012)*, 21–26 (2012).
23. M. I. Chacon-Murguia and G. Ramirez-Alonso, "Fuzzy-neural self-adapting background modeling with automatic motion analysis for dynamic object detection," *Appl. Soft Comput.* **36**, 570–577 (2015).
24. M. I. Chacon-Murguia et al., "Simplified SOM-neural model for video segmentation of moving object," in *Int. Joint Conf. on Neural Networks*, pp. 474–480 (2009).
25. B. Antic, V. Crnojevic, and D. Culibrk, "Efficient wavelet based detection of moving objects," in *16th Int. Conf. on Digital Signal Processing*, pp. 1–6 (2009).
26. Y.-p. Guan, X.-q. Cheng, and X.-l. Jia, "Motion foreground detection based on wavelet transformation and color ratio difference," in *3rd Int. Congress on Image and Signal Processing (CISP 2010)*, pp. 1423–1426 (2010).
27. S. Messelodi et al., "A Kalman filter based background updating algorithm robust to sharp illumination changes," in *Int. Conf. on Image Analysis and Processing*, pp. 163–170 (2005).
28. A. Morde, X. Ma, and S. Guler, "Learning a background model for change detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW 2012)*, pp. 15–20 (2012).
29. G. T. Cinar and J. C. Príncipe, "Adaptive background estimation using an information theoretic cost for hidden state estimation," in *Int. Joint Conf. on Neural Networks (IJCNN 2011)*, pp. 489–494 (2011).
30. N. Goyette et al., "A novel video dataset for change detection benchmarking," *IEEE Trans. Image Process.* **23**(11), 4663–4679 (2014).
31. A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(3), 257–267 (2001).
32. D. Kit, B. Sullivan, and D. Ballard, "Novelty detection using growing neural gas for visuo-spatial memory," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2011)*, pp. 1194–1200 (2011).
33. J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proc. Ninth IEEE Int. Conf. on Computer Vision*, pp. 44–50 (2003).
34. F. J. Hernandez-Lopez and M. Rivera, "Change detection by probabilistic segmentation from monocular view," *Mach. Vision Appl.* **25**(5), 1175–1195 (2014).
35. H. Kim et al., "Robust foreground extraction technique using Gaussian family model and multiple thresholds," in *Asian Conf. on Computer Vision* 758–768 (2007).
36. F. Porikli and O. Tuzel, "Bayesian background modeling for foreground detection," in *Proc. of the Third ACM Int. Workshop on Video Surveillance & Sensor Networks*, pp. 55–58 (2005).
37. P. Jaikumar, A. Singh, and S. K. Mitra, "Background subtraction in videos using Bayesian learning with motion information," in *British Machine Vision Conf. (BMVC)*, pp. 615–624 (2008).
38. A. Elgammal et al., "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE* **90**(7), 1151–1163 (2002).
39. O. Barnich and M. Van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.* **20**(6), 1709–1724 (2011).
40. M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW 2012)*, pp. 38–43 (2012).
41. B. Stenger et al., "Topology free hidden Markov models: application to background modeling," in *Proc. Eighth IEEE Int. Conf. on Computer Vision (ICCV 2001)*, pp. 294–301 (2001).
42. N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 831–843 (2000).
43. J. Rymel et al., "Adaptive eigen-backgrounds for object detection," in *Int. Conf. on Image Processing (ICIP 2004)*, pp. 1847–1850 (2004).
44. F. Porikli, "Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images," in *Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTIONS 2005)*, pp. 20–27 (2005).
45. C. Zhao, X. Wang, and W.-K. Cham, "Background subtraction via robust dictionary learning," *EURASIP J. Image Video Process.* **2011**(1), 1–12 (2011).
46. L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 774–780 (2000).
47. X. Lu and R. Manduchi, "Fast image motion segmentation for surveillance applications," *Image Vision Comput.* **29**(2), 104–116 (2011).
48. D. Zhou and H. Zhang, "Modified GMM background modeling and optical flow for detection of moving objects," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 2224–2229 (2005).
49. L. Cheng et al., "Real-time discriminative background subtraction," *IEEE Trans. Image Process.* **20**(5), 1401–1414 (2011).
50. B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(5), 1017–1023 (2012).
51. M. De Gregorio and M. Giordano, "A WiSARD-based approach to CDnet," in *BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC 2013)*, pp. 172–177 (2013).
52. X. Cheng et al., "Improving video foreground segmentation with an object-like pool," *J. Electron. Imaging* **24**(2), 023034 (2015).
53. A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vision Image Understanding* **122**, 4–21 (2014).
54. A. H. Lai and N. H. Yung, "A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence," in *Proc. of the 1998 IEEE Int. Symp. on Circuits and Systems (ISCAS 1998)*, Vol. 4, pp. 241–244 (1998).

55. N. J. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Mach. Vision Appl.* **8**(3), 187–193 (1995).
56. S. Calderara et al., "Reliable background suppression for complex scenes," in *Proc. of the 4th ACM Int. Workshop on Video Surveillance and Sensor Networks*, pp. 211–214 (2006).
57. X. Jian et al., "Background subtraction based on a combination of texture, color and intensity," in *9th Int. Conf. on Signal Processing (ICSP 2008)*, pp. 1400–1405 (2008).
58. P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, P. Remagnino et al., Eds., pp. 135–144, Springer US, Boston, Massachusetts (2002).
59. T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection-a survey," *Recent Pat. Comput. Sci.* **1**(3), 219–237 (2008).
60. Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.* **27**(7), 773–780 (2006).
61. M. Sivabalakrishnan and D. Manjula, "Adaptive background subtraction in dynamic environments using fuzzy logic," *Int. J. Comput. Sci. Eng.* **2**(2), 270–273 (2010).
62. T. Bouwmans, "Background subtraction for visual surveillance: a fuzzy approach," *Handb. Soft Comput. Video Surveillance* **5**, 103–138 (2012).
63. Z. Zhao et al., "A fuzzy background modeling approach for motion detection in dynamic backgrounds," in *Multimedia and Signal Processing: Second Int. Conf., CMSP 2012, Shanghai, China, December 7-9, 2012. Proc.*, F. L. Wang et al., Eds., pp. 177–185, Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
64. D. Culibrk et al., "Neural network approach to background modeling for video object segmentation," *IEEE Trans. Neural Networks* **18**(6), 1614–1627 (2007).
65. L. Maddalena and A. Petrosino, "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection," *Neural Comput. Appl.* **19**(2), 179–186 (2010).
66. Y. Goya et al., "Vehicle trajectories evaluation by static video sensors," in *IEEE Intelligent Transportation Systems Conf. (ITSC 2006)*, pp. 864–869 (2006).
67. K. Sehairi, F. Chouireb, and J. Meunier, "Comparison study between different automatic threshold algorithms for motion detection," in *4th Int. Conf. on Electrical Engineering (ICEE 2015)*, pp. 1–8 (2015).
68. M. Van Droogenbroeck and O. Paquot, "Background subtraction: experiments and improvements for ViBe," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW 2012)*, pp. 32–37 (2012).
69. T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance," *Comput. Vision Image Understanding* **122**, 22–34 (2014).
70. T. Bouwmans et al., "Decomposition into low-rank plus additive matrices for background/foreground separation: a review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.* **23**, 1–71 (2017).
71. R. Cabral et al., "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 2488–2495 (2013).
72. N. Wang and D.-Y. Yeung, "Bayesian robust matrix factorization for image and video processing," in *Proc. of the IEEE Int. Conf. on Computer Vision*, pp. 1785–1792 (2013).
73. C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection via robust low rank matrix factorization including spatial constraint with iterative reweighted regression," in *21st Int. Conf. on Pattern Recognition (ICPR 2012)*, pp. 2805–2808 (2012).
74. S. Javed et al., "Robust background subtraction to global illumination changes via multiple features-based online robust principal components analysis with Markov random field," *J. Electron. Imaging* **24**(4), 043011 (2015).
75. B. Zhou, F. Zhang, and L. Peng, "Background modeling for dynamic scenes using tensor decomposition," in *6th Int. Conf. on Electronics Information and Emergency Communication (ICEIEC 2016)*, pp. 206–210 (2016).
76. W. Cao et al., "Total variation regularized tensor RPCA for background subtraction from compressive measurements," *IEEE Trans. Image Process.* **25**(9), 4075–4090 (2016).
77. A. Sobral et al., "Online stochastic tensor decomposition for background subtraction in multispectral video sequences," in *Proc. of the IEEE Int. Conf. on Computer Vision Workshops*, pp. 106–113 (2015).
78. T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: an overview," *Comput. Sci. Rev.* **11**, 31–66 (2014).
79. S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent Pat. Comput. Sci.* **1**(1), 32–54 (2008).
80. G. Morris and P. Angelov, "Real-time novelty detection in video using background subtraction techniques: state of the art a practical review," in *IEEE Int. Conf. on Systems, Man and Cybernetics (SMC 2014)*, pp. 537–543 (2014).
81. T. Bouwmans, "Subspace learning for background modeling: a survey," *Recent Pat. Comput. Sci.* **2**(3), 223–234 (2009).
82. C. Cuevas, R. Martínez, and N. García, "Detection of stationary foreground objects: a survey," *Comput. Vision Image Understanding* **152**, 41–57 (2016).
83. A. Bux, P. Angelov, and Z. Habib, "Vision based human activity recognition: a review," in *Advances in Computational Intelligence Systems: Contributions Presented at the 16th UK Workshop on Computational Intelligence, September 7–9, 2016, Lancaster, UK*, P. Angelov et al., Eds., pp. 341–371, Springer International Publishing, Cham (2017).
84. M. Cristani et al., "Background subtraction for automated multisensor surveillance: a comprehensive review," *EURASIP J. Adv. Signal Process.* **2010**(1), 343057 (2010).
85. http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm.
86. K. Toyama et al., "Wallflower: principles and practice of background maintenance," in *the Proc. of the Seventh IEEE Int. Conf. on Computer Vision*, pp. 255–261 (1999).
87. T. Matsuyama, T. Ohya, and H. Habe, "Background subtraction for non-stationary scenes," in *Proc. of Asian Conf. on Computer Vision*, pp. 662–667 (2000).
88. I. Haritaoglu, D. Harwood, and L. S. Davis, "W4S: a real-time system for detecting and tracking people in 2 1/2D," in *European Conf. on Computer Vision*, pp. 877–892 (1998).
89. H. Nakai, "Non-parameterized Bayes decision method for moving object detection," in *Proc. Asian Conference on Computer Vision (ACCV)*, pp. 447–451 (1995).
90. B. Lo and S. Velastin, "Automatic congestion detection system for underground platforms," in *Proc. of 2001 Int. Symp. on Intelligent Multimedia, Video, and Speech Processing*, pp. 158–161 (2001).
91. R. Cucchiara et al., "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(10), 1337–1342 (2003).
92. B. Han, D. Comaniciu, and L. Davis, "Sequential kernel density approximation through mode propagation: applications to background modeling," in *Asian Conf. on Computer Vision (ACCV 2004)*, pp. 818–823 (2004).
93. M. Seki et al., "Background subtraction based on cooccurrence of image variations," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 65–72 (2003).
94. S.-C. S. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video," *EURASIP J. Adv. Signal Process.* **2005**(14), 726261 (2005).
95. K. Karmann and A. Brandt, "Moving object recognition using and adaptive background memory," in *Time-Varying Image Processing and Moving Object Recognition*, V. Cappellini, Ed., Vol. **2**, pp. 289–307 (1990).
96. http://i21www.ira.uka.de/image_sequences/.
97. Y. Benezeth et al., "Comparative study of background subtraction algorithms," *J. Electron. Imaging* **19**(3), 033003 (2010).
98. http://imagelab.ing.unimore.it/vssn06/.
99. L. M. Brown et al., "Performance evaluation of surveillance systems under varying conditions," in *Proc. IEEE PETS Workshop*, pp. 1–8 (2005).
100. B. White and M. Shah, "Automatically tuning background subtraction parameters using particle swarm optimization," in *IEEE Int. Conf. on Multimedia and Expo*, pp. 1826–1829 (2007).
101. Hanzi W. and D. Suter, "A re-evaluation of mixture of Gaussian background modeling [video signal processing applications]," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, Vol. **1012** pp. ii/1017–ii/1020 (2005).
102. K. Schindler and H. Wang, "Smooth foreground-background segmentation for video processing," in *Asian Conf. on Computer Vision*, pp. 581–590 (2006).
103. N. A. Setiawan et al., "Gaussian mixture model in improved HLS color space for human silhouette extraction," in *Advances in Artificial Reality and Tele-Existence*, pp. 732–741, Springer (2006).
104. M. Cristani and V. Murino, "A spatial sampling mechanism for effective background subtraction," in *Proc. 2nd Int. Conf. on Computer Vision Theory and Applications (VISAPP 2007)*, pp. 403–412 (2007).
105. M. Cristani and V. Murino, "Background subtraction with adaptive spatio-temporal neighborhood analysis," in *Proc. 2nd Int. Conf. on Computer Vision Theory and Applications (VISAPP 2007)*, pp. 484–489 (2008).
106. D.-M. Tsai and S.-C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.* **18**(1), 158–167 (2009).
107. S. S. Bucak, B. Günsel, and O. Gursoy, "Incremental non-negative matrix factorization for dynamic background modelling," in *Pattern Recognition in Information Systems (PRIS)*, pp. 107–116 (2007).
108. X. Li et al., "Robust foreground segmentation based on two effective background models," in *Proc. of the 1st ACM Int. Conf. on Multimedia Information Retrieval*, pp. 223–228 (2008).
109. N. Goyette et al., "Changedetection.net: a new change detection benchmark dataset," in *IEEE Computer Society Conf. on Computer*

Vision and Pattern Recognition Workshops (CVPRW 2012), pp. 1–8 (2012).
110. J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier, "Background subtraction based on local shape," arXiv preprint arXiv:1204.6326 (2012).
111. D. Riahi, P. St-Onge, and G. Bilodeau, "RECTGAUSS-tex: block-based background subtraction," in *Proce Dept. genie Inform. genie logiciel, Ecole Polytechn. Montr. Montr.*, QC, Canada, pp. 1–9 (2012).
112. Y. Nonaka et al., "Evaluation report of integrated background modeling based on spatio-temporal features," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW 2012)*, pp. 9–14 (2012).
113. S. Yoshinaga et al., "Background model based on intensity change similarity among pixels," in *19th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV 2013)*, pp. 276–280 (2013).
114. A. Schick, M. Bäuml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *IEEE Workshop on Change Detection* (2012).
115. http://wordpress-jodoin.dmi.usherb.ca/dataset2012/.
116. Y. Wang et al., "CDnet 2014: an expanded change detection benchmark dataset," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 387–394 (2014).
117. X. Lu, "A multiscale spatio-temporal background model for motion detection," in *IEEE Int. Conf. on Image Processing (ICIP 2014)*, pp. 3268–3271 (2014).
118. D. Liang and S. i. Kaneko, "Improvements and experiments of a compact statistical background model," arXiv preprint arXiv:1405.6275 (2014).
119. B. Tamás, "Detecting and analyzing rowing motion in videos," in *BME Scientific Student Conf.*, Budapest, pp. 1–29 (2016).
120. B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 395–398 (2014).
121. M. Sedky, M. Moniri, and C. C. Chibelushi, "Spectral-360: a physics-based technique for change detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 399–402 (2014).
122. M. De Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 403–407 (2014).
123. P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Flexible background subtraction with self-balanced local sensitivity," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 408–413 (2014).
124. R. Wang et al., "Static and moving object detection using flux tensor with split Gaussian models," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 414–418 (2014).
125. P.-M. Jodoin et al., "Overview and benchmarking of motion detection methods," in *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans et al., Eds., CRC Press, Boca Raton, Florida (2014).
126. R. H. Evangelio and T. Sikora, "Complementary background models for the detection of static and moving objects in crowded environments," in *8th IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS 2011)*, pp. 71–76 (2011).
127. R. H. Evangelio, M. Pätzold, and T. Sikora, "Splitting Gaussians in mixture models," in *IEEE Ninth Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS 2012)*, pp. 300–305 (2012).
128. T. S. Haines and T. Xiang, "Background subtraction with dirichlet processes," in *European Conf. on Computer Vision*, pp. 99–113 (2012).
129. S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?," arXiv preprint arXiv:1505.02921 (2015).
130. S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of Gaussians for dynamic background modelling," in *10th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS 2013)*, pp. 63–68 (2013).
131. http://wordpress-jodoin.dmi.usherb.ca/dataset2014/.
132. Y. Xu et al., "Background modeling methods in video analysis: a review and comparative evaluation," *CAAI Trans. Intell. Technol.* **1**(1), 43–60 (2016).
133. H. Wang and D. Suter, "A consensus-based method for tracking: modelling background scenario and foreground appearance," *Pattern Recognit.* **40**(3), 1091–1105 (2007).
134. H. Wang and D. Suter, "Background subtraction based on a robust consensus method," in *18th Int. Conf. on Pattern Recognition (ICPR 2006)*, pp. 223–226 (2006).
135. J. Wen et al., "Joint video frame set division and low-rank decomposition for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.* **24**(12), 2034–2048 (2014).
136. Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(11), 1778–1792 (2005).http://www.cs.cmu.edu/~yaser/#software.
137. V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(1), 171–177 (2010). http://www.svcl.ucsd.edu/projects/background_subtraction/ucsdbgsub_dataset.htm.
138. A. Vacavant et al., "A benchmark dataset for outdoor foreground/background extraction," in *Asian Conf. on Computer Vision*, pp. 291–300 (2012). http://bmc.iut-auvergne.com/.
139. A. Doulamis et al., "An architecture for a self configurable video supervision," in *Proc. of the 1st ACM Workshop on Analysis and Retrieval of Events/Actions and Workflows in Video streams*, pp. 97–104 (2008).
140. D. D. Bloisi et al., "ARGOS-Venice boat classification," in *12th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS 2015)*, pp. 1–6 (2015). http://www.dis.uniroma1.it/~labrococo/MAR/index.htm.
141. J. Gallego Vila, "Parametric region-based foreround segmentation in planar and multi-view sequences" (2013). https://imatge.upc.edu/web/resources/kinect-database-foreground-segmentation.
142. M. Camplani and L. Salgado, "Background foreground segmentation with RGB-D Kinect data: an efficient combination of classifiers," *J. Visual Commun. Image Represent.* **25**(1), 122–136 (2014). http://eis.bristol.ac.uk/~mc13306/.
143. B. J. Boom et al., "A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage," *Ecol. Inf.* **23**, 83–97 (2014). http://groups.inf.ed.ac.uk/f4k/index.html.
144. S. K. Choudhury et al., "An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios," *IEEE Access* **4**, 6133–6150 (2016).
145. D. H. Parks and S. S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *IEEE Fifth Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS 2008)*, pp. 192–199 (2008).
146. S. Joudaki, M. S. B. Sunar, and H. Kolivand, "Background subtraction methods in video streams: a review," in *4th Int. Conf. on Interactive Digital Media (ICIDM 2015)*, pp. 1–6 (2015).
147. Y. Dhome et al., "A benchmark for background subtraction algorithms in monocular vision: a comparative study," in *2nd Int. Conf. on Image Processing Theory Tools and Applications (IPTA 2010)*, pp. 66–71 (2010).
148. D. K. Prasad et al., "Video processing from electro-optical sensors for object detection and tracking in maritime environment: a survey," in *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–24 (2017).
149. Z. Wang, J. Xiong, and Q. Zhang, "Motion saliency detection based on temporal difference," *J. Electron. Imaging* **24**(3), 033022 (2015).
150. S. S. M. Radzi et al., "Extraction of moving objects using frame differencing, ghost and shadow removal," in *5th Int. Conf. on Intelligent Systems, Modelling and Simulation (ISMS 2014)*, pp. 229–234 (2014).
151. C. Chen and X. Zhang, "Moving vehicle detection based on union of three-frame difference," in *Advances in Electronic Engineering, Communication and Management Vol.2: Proceedings of 2011 International Conference on Electronic Engineering, Communication and Management (EECM 2011), held on December 24–25, 2011, Beijing, China*, D. Jin and Lin S., Eds., pp. 459–464, Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
152. Z. Wang, "Hardware implementation for a hand recognition system on FPGA," in *5th Int. Conf. on Electronics Information and Emergency Communication (ICEIEC 2015)*, pp. 34–38 (2015).
153. Y. Zhang, X. Wang, and B. Qu, "Three-frame difference algorithm research based on mathematical morphology," *Proc. Eng.* **29**, 2705–2709 (2012).
154. J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians," *Image Vision Comput.* **22**(7), 563–570 (2004).
155. X.-j. Tan, J. Li, and C. Liu, "A video-based real-time vehicle detection method by classified background learning," *World Trans. Eng. Technol. Edu.* **6**(1), 189 (2007).
156. J. Richefeu and A. Manzanera, "A new hybrid differential filter for motion detection," in *Computer Vision and Graphics: Int. Conf., ICCVG 2004, Warsaw, Poland, September 2004, Proc.*, K. Wojciechowsk et al., Eds., pp. 727–732, Springer, Netherlands, Dordrecht (2006).
157. A. Manzanera and J. C. Richefeu, "A new motion detection algorithm based on $\Sigma - \Delta$ background estimation," *Pattern Recognit. Lett.* **28**(3), 320–328 (2007).
158. L. Lacassagne et al., "High performance motion detection: some trends toward new embedded architectures for vision systems," *J. Real-Time Image Process.* **4**(2), 127–146 (2009).
159. P. Bouthemy and P. Lalande, "Recovery of moving object masks in an image sequence using local spatiotemporal contextual information," *Opt. Eng.* **32**(6), 1205–1212 (1993).
160. A. Caplier, F. Luthon, and C. Dumontier, "Real-time implementations of an MRF-based motion detection algorithm," *Real-Time Imaging* **4**(1), 41–54 (1998).
161. J. Denoulet et al., "Implementing motion Markov detection on general purpose processor and associative mesh," in *Proc. Seventh Int. Workshop on Computer Architecture for Machine Perception (CAMP 2005)*, pp. 288–293 (2005).

162. Z. Tang, Z. Miao, and Y. Wan, "Background subtraction using running Gaussian average and frame difference," in *Entertainment Computing (ICEC 2007)*, pp. 411–414, Springer (2007).
163. S. Jabri et al., "Detection and location of people in video images using adaptive fusion of color and edge information," in *15th Int. Conf. on Pattern Recognition Proc.*, pp. 627–630 (2000).
164. J. S. Lumentut and F. E. Gunawan, "Evaluation of recursive background subtraction algorithms for real-time passenger counting at bus rapid transit system," *Proc. Comput. Sci.* **59**, 445–453 (2015).
165. Y.-F. Ma and H. Zhang, "Detecting motion object by spatio-temporal entropy," in *IEEE Int. Conf. Multimedia and Expo*, pp. 265–268 (2001).
166. G. Jing, C. E. Siong, and D. Rajan, "Foreground motion detection by difference-based spatial temporal entropy image," in *2004 IEEE Region 10 Conf. TENCON*, pp. 379–382 (2004).
167. M.-C. Chang and Y.-J. Cheng, "Motion detection by using entropy image and adaptive state-labeling technique," in *IEEE Int. Symp. on Circuits and Systems (ISCAS 2007)*, pp. 3667–3670 (2007).
168. M. Celenk et al., "Change detection and object tracking in IR surveillance video," in *Third Int. IEEE Conf. on Signal-Image Technologies and Internet-Based System (SITIS 2007)*, pp. 791–797 (2007).
169. Y. Tian et al., "Selective eigenbackground for background modeling and subtraction in crowded scenes," *IEEE Trans. Circuits Syst. Video Technol.* **23**(11), 1849–1864 (2013).
170. R. J. Radke et al., "Image change detection algorithms: a systematic survey," *IEEE Trans. Image Process.* **14**(3), 294–307 (2005).
171. R. Fisher, "Change detection in color images," in *Proc. of 7th IEEE Conf. on Computer Vision and Pattern* (1999).
172. N. Selvarasu, A. Nachiappan, and N. Nandhitha, "Euclidean distance based color image segmentation of abnormality detection from pseudo color thermographs," *Int. J. Comput. Theory Eng.* **2**(4), 514–516 (2010).
173. Q. Zhou and J. K. Aggarwal, "Tracking and classifying moving objects from video," in *Proc. of IEEE Workshop on Performance Evaluation of Tracking and Surveillance* (2001).
174. T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *37th IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2008)*, pp. 1–8 (2008).
175. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, 721–741 (1984).
176. R. C. Dubes and A. K. Jain, "Random field models in image analysis," *J. Appl. Stat.* **16**(2), 131–164 (1989).
177. P. W. Power and J. A. Schoonees, "Understanding background mixture models for foreground segmentation," in *Proc. Image and Vision Computing New Zealand*, pp. 10–11 (2002).
178. L. Carminati and J. Benois-Pineau, "Gaussian mixture classification for moving object detection in video surveillance environment," in *IEEE Int. Conf. on Image Processing (ICIP 2005)*, Vol. 3, pp. 113–116 (2005).
179. H. Kim et al., "Robust silhouette extraction technique using background subtraction," in *10th Meeting on Image Recognition and Understand (MIRU)* (2007).
180. K. Makantasis, A. Doulamis, and N. F. Matsatsinis, "Student-t background modeling for persons' fall detection through visual cues," in *2012 13th Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pp. 1–4 (2012).
181. Z. Liu, K. Huang, and T. Tan, "Foreground object detection using top-down information based on EM framework," *IEEE Trans. Image Process.* **21**(9), 4204–4217 (2012).
182. R. Li, C. Yu, and X. Zhang, "Fast robust eigen-background updating for foreground detection," in *IEEE Int. Conf. on Image Processing*, pp. 1833–1836 (2006).
183. L. Maddalena and A. Petrosino, "Stopped object detection by learning foreground model in videos," *IEEE Trans. Neural Networks Learn. Syst.* **24**(5), 723–735 (2013).
184. L. Maddalena and A. Petrosino, "The 3DSOBS+ algorithm for moving object detection," *Comput. Vision Image Understanding* **122**, 65–73 (2014).
185. B. Nikolov, N. Kostov, and S. Yordanova, "Investigation of mixture of Gaussians method for background subtraction in traffic surveillance," *Int. J. Reasoning-based Intell. Syst.* **5**(3), 161–168 (2013).
186. H. Liu et al., "Combining background subtraction and three-frame difference to detect moving object from underwater video," in *OCEANS 2016*, Shanghai, pp. 1–5 (2016).

Kamal Sehairi received his BS degree and engineering degree in electronics from Amar Telidji Laghouat University in 2004 and 2009, respectively, and his magister degree in advanced techniques in signal processing from École Militaire Polytechnique, Algiers, in 2012. He is a PhD student at Amar Telidji Laghouat University and a member of the LTSS Laboratory. He is an assistant lecturer at École Normale Supérieure, Laghouat (ENS-L). His current research interests include image processing and video processing, real-time implementation, FPGAs, classification and recognition, and video surveillance systems. He is a member of SPIE.

Fatima Chouireb received her diploma degree in electrical engineering in 1992 and her magister and PhD degrees from Saad Dahlab University of Blida, Algeria, in 1996 and 2007, respectively. In 1997, she joined the Electrical Engineering Department, Laghouat University, Algeria, as an assistant lecturer. Since 2015, she has been full professor in the same departement. She is also a team leader of the Signal, Image, and Speech Research Group of LTSS Laboratory, Laghouat University, Algeria. Her current research interests include signal, image, and speech processing, computer vision, localization/mapping/SLAM, and mobile robotics problems.

Jean Meunier received his BS degree in physics from the Université de Montréal, Montréal, QC, Canada, in 1981, his MScA degree in applied mathematics in 1983, and his PhD degree in biomedical engineering from the Ecole Polytechnique de Montréal, Montréal, in 1989. In 1989, after postdoctoral studies with the Montreal Heart Institute, Montréal, he joined the Department of Computer Science and Operations Research, Université de Montréal, where he is currently a full professor. He is a regular member of the Biomedical Engineering Institute at the same institution. His current research interests include computer vision and its applications to medical imaging and health care.