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

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

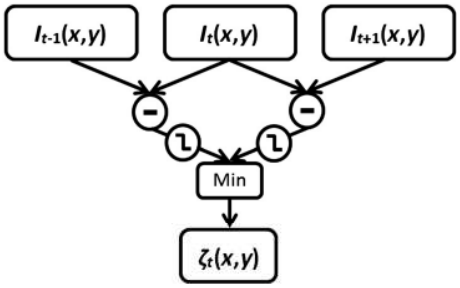

Fig. 1 3-FD method.

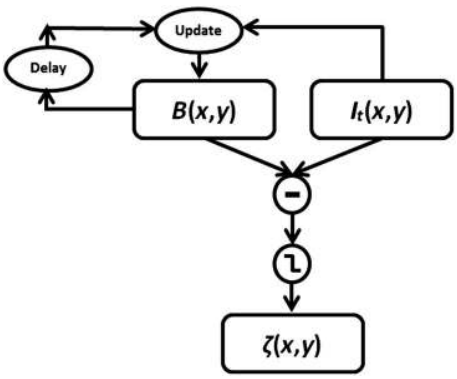

Fig. 2 Adaptive background detection.

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

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)

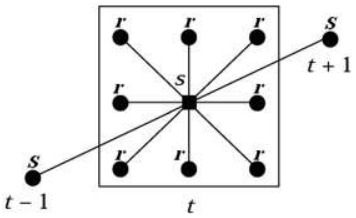

Fig. 3 3×3 spatio-temporal neighborhood.[106]

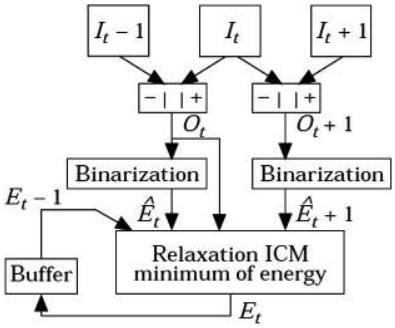

Fig. 4 MRF-based motion detection.[106]

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

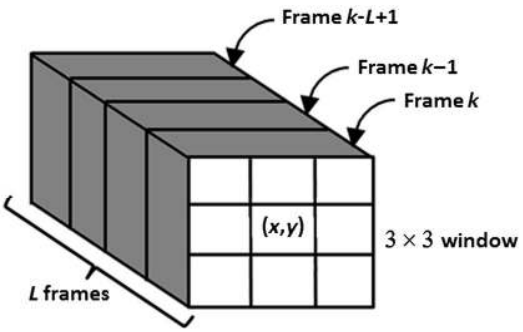

Fig. 5 Pixels used to construct the spatio-temporal histogram of pixel (i,j).

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

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

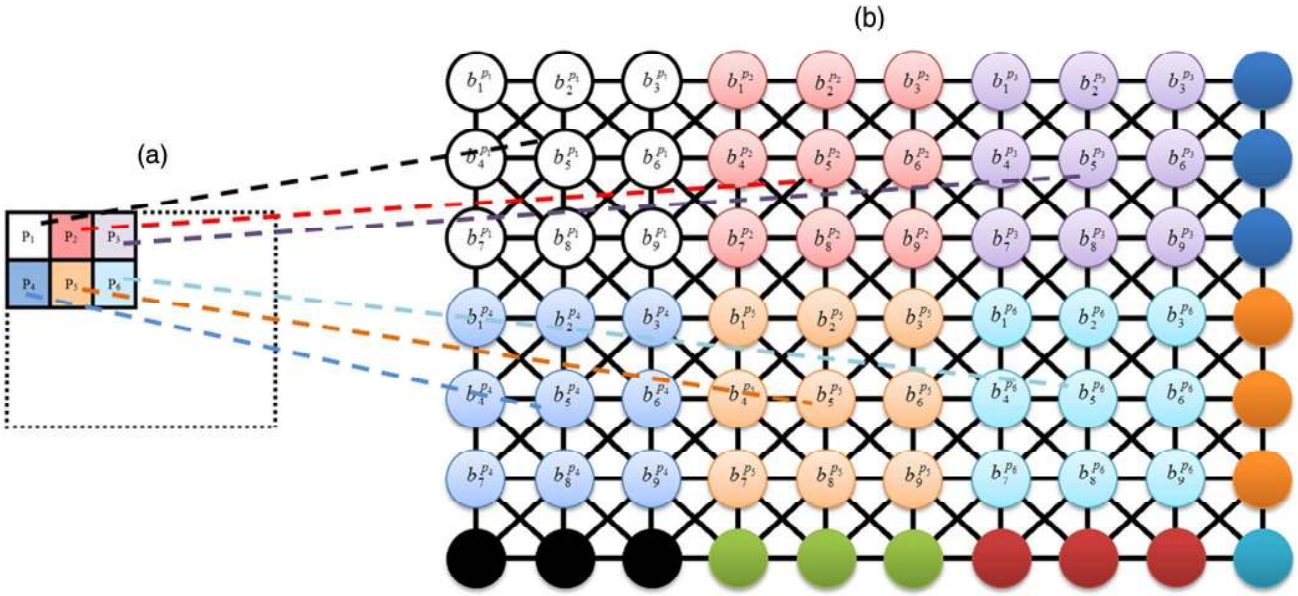

Fig. 6 (a) A simple image I_t and (b) the modeling neuronal map B_t when $n = 3$.

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

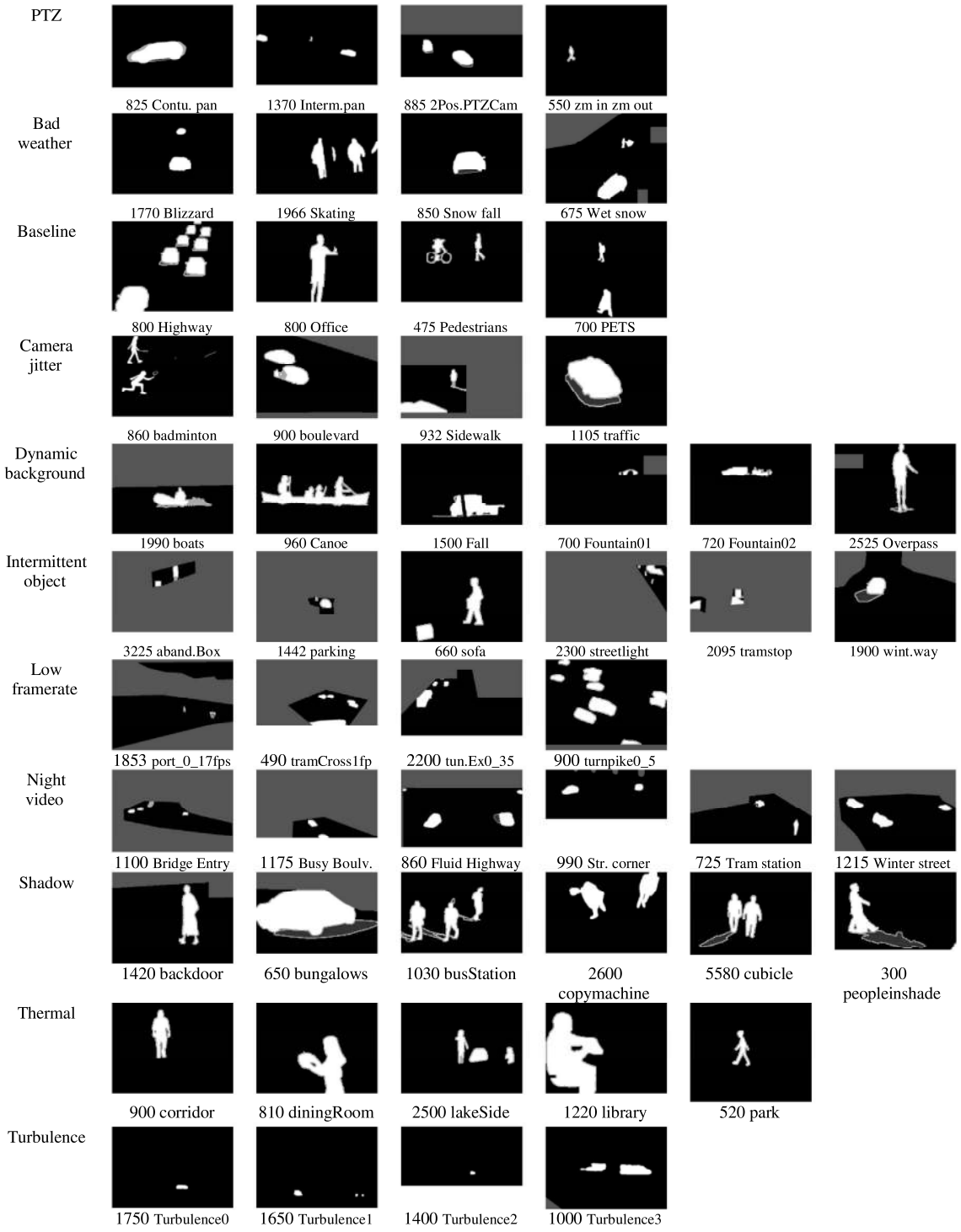

Fig. 8 Corresponding ground truths of the sample frames in Fig. 7.

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

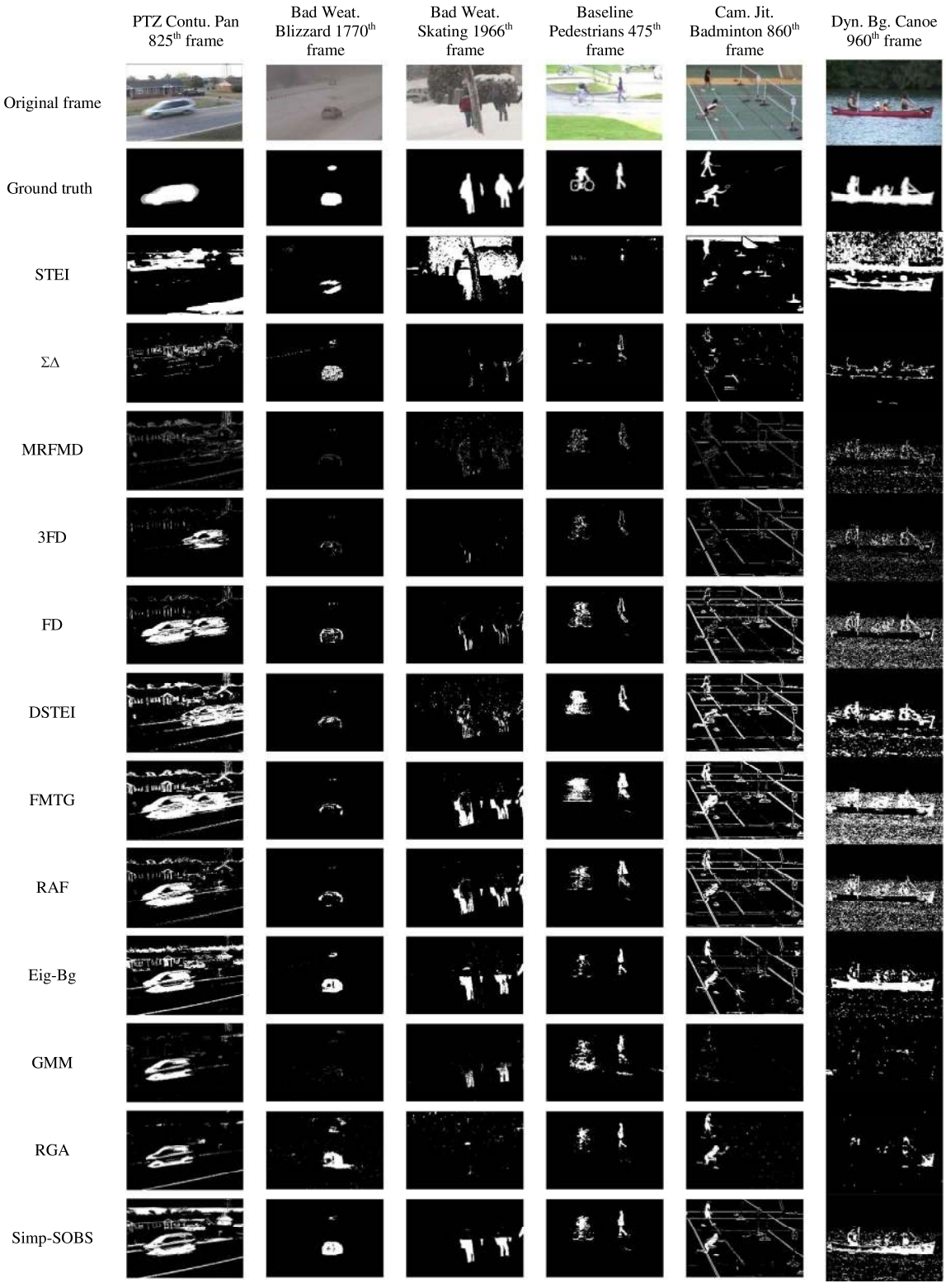

Fig. 9 Samples from the results of tested motion detection methods [pan tilit zoom (PTZ), Bad Wea., Baseline, Cam Jit., Dyn. Bg].

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

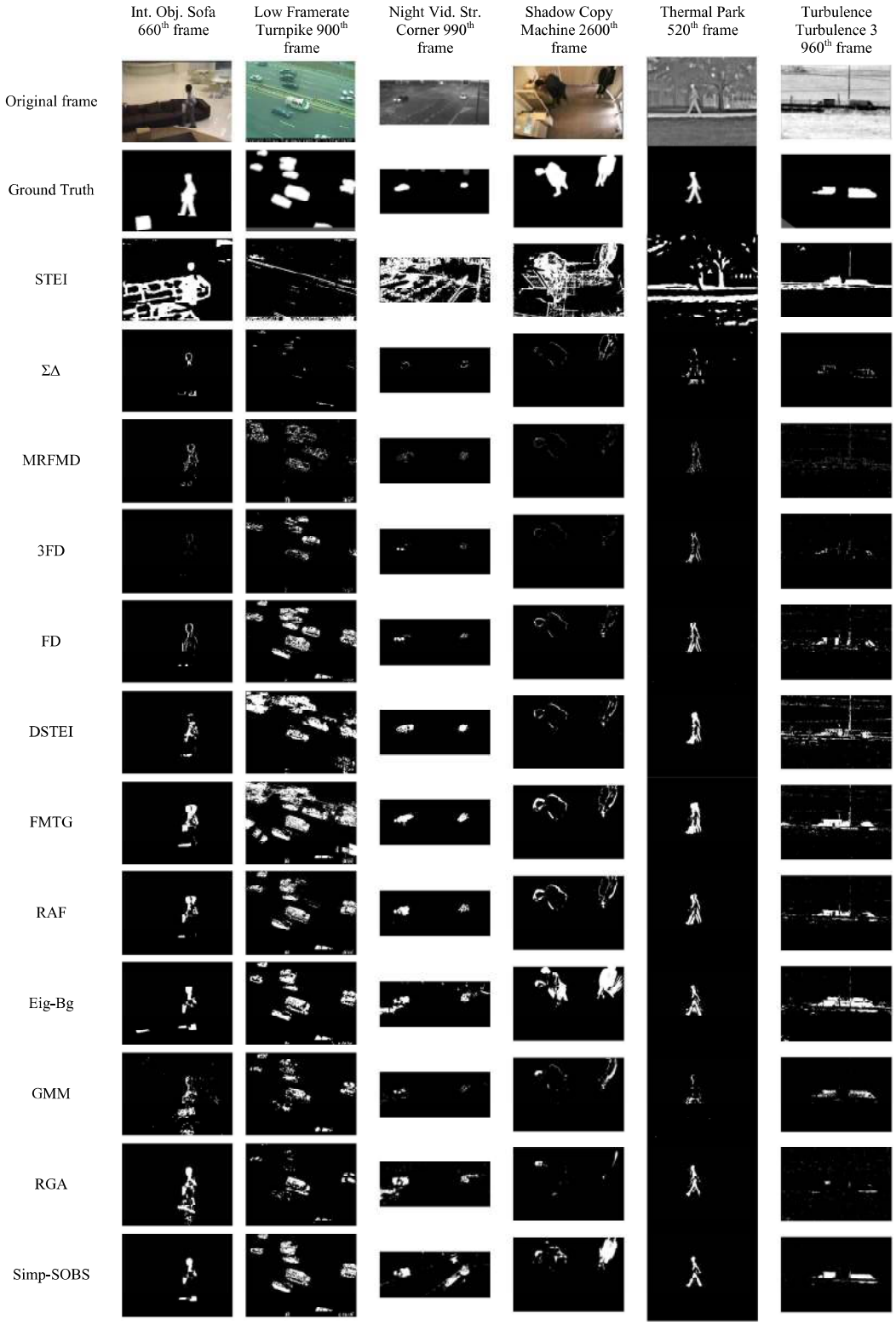

Fig. 10 Samples from the results of tested motion detection methods (Int. Obj., Low. Fr., N.Vid., Shad., Ther., Turb.).

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"