

A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos



Andrews Sobral^a, Antoine Vacavant^{b,*}

^a Lab MIA, Université de La Rochelle, France

^b ISIT, UMR6284, CNRS/Université d'Auvergne, F-63001 Clermont-Ferrand, France

ARTICLE INFO

Article history:

Received 19 April 2013

Accepted 10 December 2013

Keywords:

Background subtraction

Motion detection

Foreground segmentation

ABSTRACT

Background subtraction (BS) is a crucial step in many computer vision systems, as it is first applied to detect moving objects within a video stream. Many algorithms have been designed to segment the foreground objects from the background of a sequence. In this article, we propose to use the BMC (Background Models Challenge) dataset, and to compare the 29 methods implemented in the BGSLibrary. From this large set of various BG methods, we have conducted a relevant experimental analysis to evaluate both their robustness and their practical performance in terms of processor/memory requirements.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Background subtraction (BS) is a crucial step in many computer vision systems, as it is first applied to detect moving objects within a video stream, without any *a priori* knowledge about these objects [40]. BS has been widely studied since the 1990s, and mainly for video-surveillance applications, since they first need to detect persons, vehicles, animals, etc. before operating more complex processes for intrusion detection, tracking, people counting, etc. Many algorithms have been designed to segment the foreground objects from the background of a sequence, and generally share the same scheme [4]:

Background initialization: They first aim to build a background model thanks to a fixed number of frames. This model can be designed by various ways (statistical, fuzzy, neuro-inspired, etc.).

Foreground detection: In the next frames, a comparison is processed between the current frame and the background model. This subtraction leads to the computation of the foreground of the scene.

Background maintenance: During this detection process, images are also analyzed in order to update the background model learned at the initialization step, with respect to a learning rate. An object not moving during long time should be integrated in the background for example.

Even if the evaluation of such algorithms is an important issue, only a few articles deal with this problematic in the literature [2,13,33]. Moreover, since they are conducted on a reduced period of time, these works do not cover a large set of algorithms of the literature. Recently, this lack of durable reference has lead to the emergence of several benchmarks, fully available on the Web, as ChangeDetection.net [20], SABS (Stuttgart Artificial Background Subtraction Dataset) [8], or BMC (Background Models Challenge) [42]. These datasets allow authors to download challenging videos, and to compare their work with both classical and recent contributions.

In this article, we propose to use the BMC dataset, which has the originality to contain both synthetic and real sequences (more precisely, it is composed of 20 synthetic videos and 9 real videos). In this benchmark, a software (BMC Wizard) is employed to compute four quality measures. Some of them are very significant for BS, and not usual in the other benchmarks (namely, SSIM [43] and D-Score [28]). We have compared the 29 methods implemented in the BGSLibrary (Background Subtraction Library) [36]. Thanks to this library, covering a large set of algorithms from the

* Corresponding author.

E-mail address: antoine.vacavant@udamail.fr (A. Vacavant).

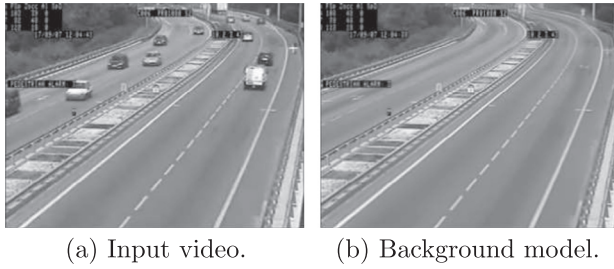


Fig. 1. Adaptive background learning after 750 iterations with $\alpha = 0.01$.

literature, we thus propose here a very large comparison of BS methods, with various methodologies.

In Section 2, we present a pragmatical state of the art about background subtraction, related to BGS library. Both BMC and BGSLibrary are described shortly in Section 3. The evaluation process is explained in Section 4. We finally discuss the results obtained and possible further works in Section 5.

2. State of the art

In this section, we propose an overview of the most famous methods for BS, related to the BGS library. Any reader could also refer to more complete states of the art in [3,7]. In the following, we adopt that bold letters represent vectors or matrices in the equations, upper case letters are collections (matrices, images, etc.), while lower case letters mean canonical elements (like pixels, scalars), and that we consider by default colored pixels in the presentation of the algorithms, meaning that pixels are composed of three components (R, G, B for example). The associated variables may thus be vectors, and images are always considered as matrices. The use of a special colorspace (as gray scale images for example) will be specified in the text if necessary.

2.1. Basic models

Essentially the BS process consists in creating a background model. In a simple way, this can be done by setting manually a static image that represents the background, and having no moving object. For each video frame, we then compute the absolute difference between the current frame and the static image. This method is called *Static Frame Difference* in the rest of the article. However, a

Table 1

Background subtraction algorithms available in BGSLibrary.

Method ID	Method name	Author(s)
<i>Basic methods, mean and variance over time</i>		
StaticFrameDifferenceBGS	Static Frame Difference	–
FrameDifferenceBGS	Frame Difference	–
WeightedMovingMeanBGS	Weighted Moving Mean	–
WeightedMovingVarianceBGS	Weighted Moving Variance	–
AdaptiveBackgroundLearning	Adaptive Background Learning	–
DPMeanBGS	Temporal Mean	–
DPAdaptiveMedianBGS	Adaptive Median	[31]
DPPratiMediodBGS	Temporal Median	[9]
<i>Fuzzy based methods</i>		
FuzzySugenoIntegral	Fuzzy Sugeno Integral	[46]
FuzzyChoquetIntegral	Fuzzy Choquet Integral	[15]
LBFuzzyGaussian	Fuzzy Gaussian	[34]
<i>Statistical methods using one gaussian</i>		
DPWrenGABGS	Gaussian Average	[44]
LBSimpleGaussian	Simple Gaussian	[2]
<i>Statistical methods using multiple Gaussians</i>		
DPGrimsonGMMBGS	Gaussian Mixture Model	[38]
MixtureOfGaussianV1BGS	Gaussian Mixture Model	[25]
MixtureOfGaussianV2BGS	Gaussian Mixture Model	[49]
DPZivkovicAGMMBGS	Gaussian Mixture Model	[49]
LBMixtureOfGaussians	Gaussian Mixture Model	[6]
<i>Type-2 Fuzzy based methods</i>		
T2FGMM_UM	Type-2 Fuzzy GMM-UM	[16,17,5]
T2FGMM_UV	Type-2 Fuzzy GMM-UV	[16,17,5]
T2FMRF_UM	Type-2 Fuzzy GMM-UM with MRF	[47]
T2FMRF_UV	Type-2 Fuzzy GMM-UV with MRF	[47]
<i>Statistical methods using color and texture features</i>		
MultiLayerBGS	Multi-Layer BGS	[45]
<i>Non-parametric methods</i>		
PixelBasedAdaptiveSegmenter	Pixel-Based Adaptive Segmenter	[22]
GMG	GMG	[18]
VuMeter	VuMeter	[19]
<i>Methods based on eigenvalues and eigenvectors</i>		
DPEigenbackgroundBGS	Eigenbackground/ SL-PCA	[32]
<i>Neural and neuro-fuzzy methods</i>		
LBAdaptiveSOM	Adaptive SOM	[29]
LBFuzzyAdaptiveSOM	Fuzzy Adaptive SOM	[30]

static image is not the best choice, if the ambient lighting changes, then the foreground segmentation may fail dramatically. Alternatively, it is possible to use the previous frame rather than a static

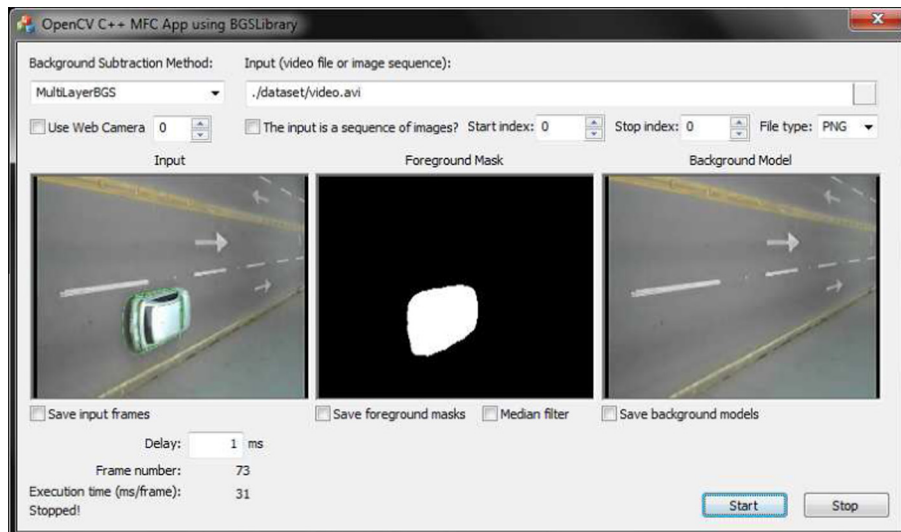


Fig. 2. An OpenCV C++ MFC Application using BGSLibrary.

Table 2

Possible combinations of synthetic data generated for BMC, and their respective number.

Parameter	Value	Description	Number
Scenes	1	Rotary	10
	2	Street	10
Event types	1	Cloudy, without acquisition noise, as normal mode	4
	2	Cloudy, with salt and pepper noise during the whole sequence	4
	3	Sunny, with noise, which generates moving cast shadows	4
	4	Foggy, with noise, making both background and foreground hard to analyze	4
	5	Wind, with noise, to produce a moving background	4
Use cases	1	10 s without objects, then moving objects during 50 s	10
	2	20 s without event, then event (e.g. sun uprising or fog) during 20 s, finally 20 s without event	10

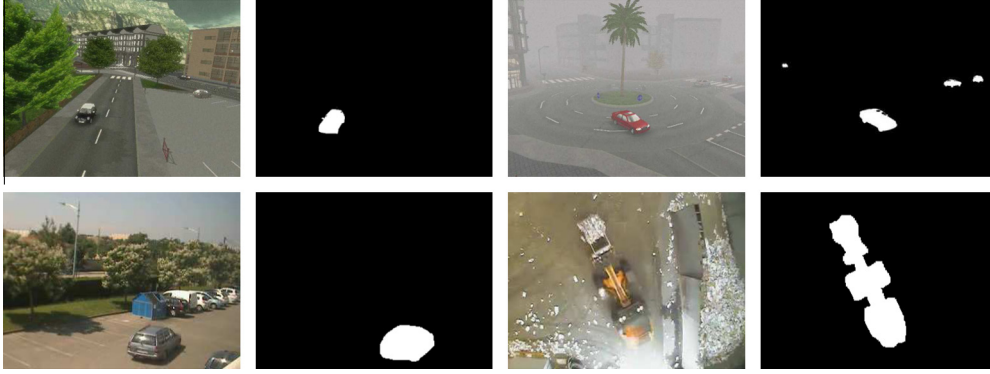
**Fig. 3.** Examples of synthetic (top) and real (bottom) videos and their associated ground truth in the BMC benchmark.

image. This approach, called further *Frame Difference*, works with some background changes but fails if the moving object stops suddenly. In [27], the authors suggest the initialization and maintenance of the background model by the arithmetic mean (or weighted mean) of the pixels between successive images. So, given a video \mathbf{V} with length l containing gray scale images defined by $\mathbf{V} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_l\}$, the background model \mathbf{B} can be defined by:

$$\mathbf{B} = \frac{1}{l} \sum_{t=1}^l \mathbf{Z}_t. \quad (1)$$

Typically Eq. (1) is used to initialize the background model.¹ However, after the initialization, to perform the background model maintenance, it is also common to use Eq. (1) recursively by:

$$\mathbf{B}_t = (1 - \alpha)\mathbf{B}_{t-1} + \alpha\mathbf{Z}_t, \quad (2)$$

where \mathbf{B}_t is the background model at time $t \in \{1, l\} \subset \mathbb{Z}$ and $\alpha \in [0, 1] \subset \mathbb{R}$ is the learning rate. The main advantage of this method is the adaptive maintenance of the background model while changes occur in the scene (see Fig. 1). Afterwards, [34] clarify that some foreground pixels are included in the background model update. To solve this issue, an adaptive-selective method is proposed. In this approach, only the regions with no moving object are updated.

After building the background model, the next step is the foreground detection. The first and most common way is to compute the absolute difference between the current frame and the background model, similarly to *Static Frame Difference* method. However, in this case the background model is continuously adapted instead of a static image. The foreground detection can be performed in other ways. More recent methods, such as [1,24,23,26,46] suggest the use of color, texture and edges features to improve the foreground detection. In [46], the authors present a

Table 3

Parameter settings of each BS algorithm.

Method ID	Settings
<i>Basic methods, mean and variance over time</i>	
StaticFrameDifferenceBGS	$T = 15$
FrameDifferenceBGS	$T = 15$
WeightedMovingMeanBGS	$T = 10$
WeightedMovingVarianceBGS	$T = 15$
AdaptiveBackgroundLearning	$T = 15, \alpha = 0.5$
DPMeanBGS	$T = 2700, \alpha = 10^{-7}, LF = 30$
DPAdaptiveMedianBGS	$T = 20, LF = 30, SR = 10$
DPPratiMediodBGS	$T = 30, SR = 5, HS = 16, \gamma = 5$
<i>Fuzzy based methods</i>	
FuzzySugenolIntegral	$T = 0.67, LF = 10, \alpha_{learn} = 0.5, \alpha_{update} = 0.05, RGB + LBP$
FuzzyChoquetIntegral	$T = 0.67, LF = 10, \alpha_{learn} = 0.5, \alpha_{update} = 0.05, RGB + LBP$
LBFuzzyGaussian	$T = 160, LR = 150, \rho = 100, \sigma = 195$
<i>Statistical methods using one Gaussian</i>	
DPWrenGABGS	$T = 12.15, LF = 30, \alpha = 0.05$
LBSimpleGaussian	$LR = 50, \rho = 255, \sigma = 150$
<i>Statistical methods using multiple gaussians</i>	
DPGrimsonGMMBGS	$T = 9, \alpha = 0.05, n = 3$
MixtureOfGaussianV1BGS	$T = 10, \alpha = 0.01$
MixtureOfGaussianV2BGS	$T = 5, \alpha = 0.01$
DPZivkovicAGMMBGS	$T = 20, \alpha = 0.01, n = 3$
LBMixtureOfGaussians	$T = 80, \alpha = 60, \rho = 120, \sigma = 210$
<i>Type-2 Fuzzy based methods</i>	
T2FGMM_UM	$T = 1, K_m = 2.5, n = 3, \alpha = 0.01$
T2FGMM_UV	$T = 1, K_v = 0.6, n = 3, \alpha = 0.01$
T2FMRF_UM	$T = 1, K_m = 2.0, n = 3, \alpha = 0.01$
T2FMRF_UV	$T = 1, K_v = 0.9, n = 3, \alpha = 0.01$
<i>Statistical methods using color and texture features</i>	
MultiLayerBGS	Original default parameters from [45]
<i>Non-parametric methods</i>	
PixelBasedAdaptiveSegmenter	Original default parameters from [22]
GMG	$T = 0.7, LF = 20$

¹ Other authors such as McFarlane and Schofield [31], Prati et al. [33] and Calderara et al. [9] suggests to use the median filter instead of the mean or average filter.

Table 3 (continued)

Method ID	Settings
VuMeter	$T = 0.03$, $\alpha = 0.995$, $\text{bin}_{\text{size}} = 8$
<i>Methods based on eigenvalues and eigenvectors</i>	
DPEigenbackgroundBGS	$T = 255$, $HS = 10$, $ED = 10$
<i>Neural and neuro-fuzzy methods</i>	
LBAdaptiveSOM	$LR = 180$, $LR_{\text{training}} = 255$, $\rho = 100$, $\rho_{\text{training}} = 240$, $TS = 40$
LBuzzyAdaptiveSOM	$LR = 180$, $LR_{\text{training}} = 255$, $\rho = 100$, $\rho_{\text{training}} = 240$, $TS = 40$

T = threshold, LF = learningFrames, SR = samplingRate, HS = historySize, γ = weight, α = (alpha or learningRate), ρ = sensitivity, σ = noiseVariance, n = gaussians, ED = embeddedDim, TS = trainingSteps.

novel approach that fuses the texture and color features for BS. In [26], the authors investigate how different color spaces affect the segmentation result in terms of noise and shadow sensitivity. In [23] is proposed an approach to model the background of images in a video sequence based on sub-pixel edge map. A new background subtraction algorithm based on a combination of texture, color and intensity information is presented in [24]. In [1] the authors propose a new technique for background modeling and subtraction by fusing texture feature, RGB color feature and Sobel edge detector.

2.2. Statistical models

One of the most popular BS method is based on a parametric probabilistic background model proposed by Stauffer and Grimson [38], and improved by Hayman and Eklundh [21]. In this algorithm, distributions of each pixel color is represented by a sum of weighted Gaussian distributions defined in a given colorspace: the *Gaussian Mixture Model* (or GMM). These distributions are generally updated using an online expectation-minimization algorithm, and enhance the use of one Gaussian distribution, as in [44] (see also [2]).

Table 4

Precision, recall and F-Measure metrics.

Metrics	Description
Precision (Pr)	$TP/(TP + FP)$
Recall (Re)	$TP/(TP + FN)$
F-Measure	$2 (Pr \cdot Re) / (Pr + Re)$
$TP = n^\circ$ of foreground pixels classified as foreground	
$FP = n^\circ$ of background pixels classified as foreground	
$TN = n^\circ$ of background pixels classified as background	
$FN = n^\circ$ of foreground pixels classified as background	

More precisely, as a new image is processed, the GMM parameters (for all pixels) are updated to explain the colors variations. In fact, at time t , we consider that the model \mathbf{m}_t generated for each pixel from the measures $\{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{t-1}\}$ of a pixel is correct. The likelihood that a pixel is a background pixel is:

$$P(\mathbf{z}_t | \mathbf{m}_t) = \sum_{n=1}^N \frac{\alpha_n}{(2\pi)^{d/2} |\Sigma_n|^{1/2}} e^{-\frac{1}{2}(\mathbf{z}_t - \mu_n)^T \Sigma_n^{-1} (\mathbf{z}_t - \mu_n)} \quad (3)$$

where d is the dimension of color space of the measures \mathbf{z}_t and each Gaussian n is described by its mean μ_n and covariance matrix Σ_n . The Gaussians are weighted by factors α_n where $\sum_{n=1}^N \alpha_n = 1$. $|\cdot|$ denotes the matrix determinant. The channels (e.g. R, G, B) of each pixel are considered independent, that is:

$$\Sigma_n = \begin{pmatrix} 1\sigma_n^2 & 0 & 0 \\ 0 & 2\sigma_n^2 & 0 \\ 0 & 0 & 3\sigma_n^2 \end{pmatrix} \quad (4)$$

In the original paper, left subscripts refer to the channel number. However, it is common to use a vector to describe the variance of the three components of a pixel. In this case, to update the GMM, we first associate the measures \mathbf{z}_t to one Gaussian n' if

$$\|\mathbf{z}_t - \mu_{n'}\| < k\sigma_{n'} \quad (5)$$

where k is 2 or 3, and σ_n a vector representing the variance of the Gaussian distribution of index n . The operator $<$ is true if all vector

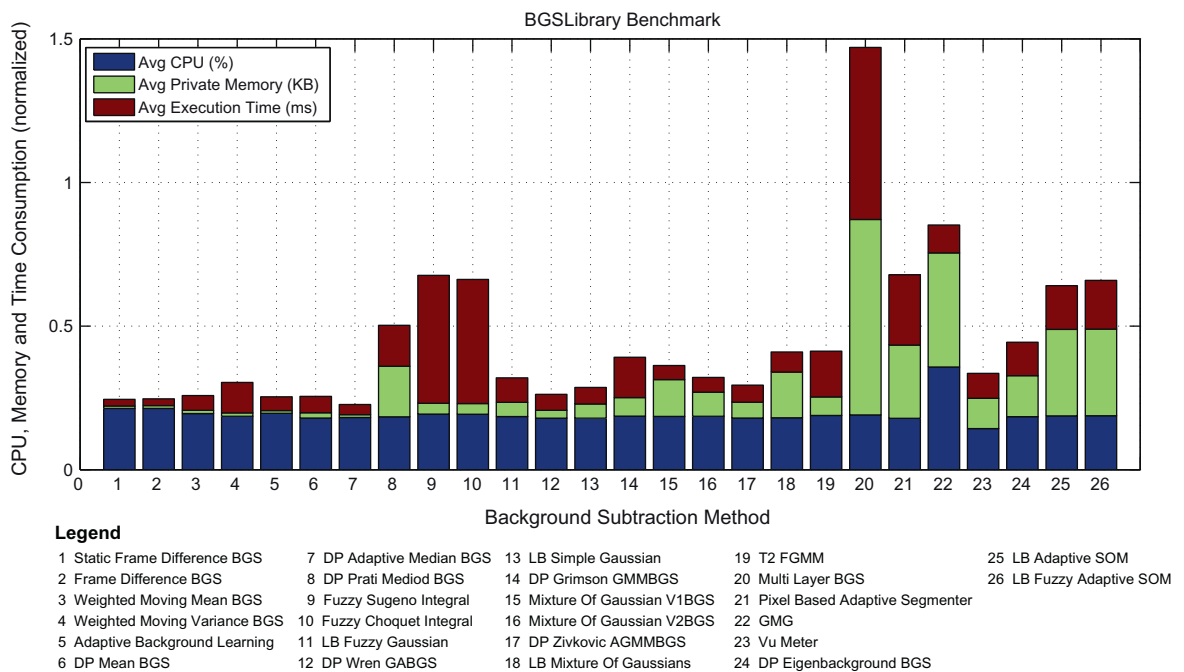
**Fig. 4.** Benchmark of BGSLibrary methods.

Table 5

Global score of BS algorithms evaluated in BMC data set. The Top 5 best methods are highlighted in yellow. The best methods in each group are in red. The bold metric values show the best score of the *F*-Measure, *D*-Score and SSIM.

Method ID	Recall	Precision	<i>F</i> -Measure	PSNR	<i>D</i> -Score	SSIM	FSD
<i>Basic methods, mean and variance over time</i>							
StaticFrameDifferenceBGS	0.885	0.660	0.750	32,238	0.011	0.884	0.119
FrameDifferenceBGS	0.702	0.925	0.798	51,626	0.002	0.993	0.799
WeightedMovingMeanBGS	0.723	0.915	0.807	51,454	0.002	0.993	0.818
WeightedMovingVarianceBGS	0.721	0.912	0.805	51,427	0.002	0.993	0.814
AdaptiveBackgroundLearning	0.808	0.884	0.844	50,684	0.002	0.993	0.896
DPMeanBGS	0.597	0.935	0.729	51,881	0.002	0.992	0.642
DPAptiveMedianBGS	0.829	0.779	0.795	43,267	0.003	0.967	0.691
DPPratiMediodBGS	0.814	0.871	0.837	49,580	0.001	0.991	0.888
<i>Fuzzy based methods</i>							
FuzzySugenIntegral	0.778	0.897	0.832	50,976	0.001	0.993	0.874
FuzzyChoquetIntegral	0.805	0.876	0.837	50,366	0.001	0.992	0.884
LBFuzzyGaussian	0.909	0.740	0.808	42,364	0.003	0.974	0.738
<i>Statistical methods using one Gaussian</i>							
DPWrenGABGS	0.795	0.922	0.853	51,394	0.001	0.993	0.922
LBSimpleGaussian	0.855	0.770	0.805	45,073	0.002	0.982	0.767
<i>Statistical methods using multiple Gaussians</i>							
DPGrimsonGMMBGS	0.717	0.913	0.802	51,445	0.002	0.993	0.808
MixtureOfGaussianV1BGS	0.793	0.912	0.847	51,107	0.001	0.993	0.910
MixtureOfGaussianV2BGS	0.893	0.813	0.850	48,383	0.002	0.992	0.899
DPZivkovicAGMMBGS	0.665	0.928	0.774	51,717	0.002	0.992	0.746
LBMixtureOfGaussians	0.868	0.834	0.848	48,794	0.001	0.991	0.907
<i>Type-2 Fuzzy based methods</i>							
T2FGMM_UM	0.661	0.935	0.774	51,792	0.002	0.992	0.745
T2FGMM_UV	0.800	0.747	0.762	43,395	0.003	0.971	0.628
T2FMRF_UM	0.852	0.670	0.743	31,732	0.012	0.872	0.030
T2FMRF_UV	0.678	0.888	0.763	50,900	0.002	0.991	0.716
<i>Statistical methods using color and texture features</i>							
MultiLayerBGS	0.893	0.863	0.875	49,398	0.001	0.993	0.974
<i>Non-parametric methods</i>							
PixelBasedAdaptiveSegmenter	0.923	0.852	0.885	49,412	0.002	0.994	0.985
GMG	0.947	0.703	0.803	41,826	0.003	0.979	0.730
VuMeter	0.722	0.842	0.775	50,296	0.002	0.992	0.735
<i>Methods based on eigenvalues and eigenvectors</i>							
DPEigenbackgroundBGS	0.879	0.658	0.747	32,843	0.011	0.891	0.114
<i>Neural and neuro-fuzzy methods</i>							
LBAdaptiveSOM	0.838	0.907	0.867	50,553	0.001	0.992	0.952
LBFuzzyAdaptiveSOM	0.877	0.811	0.836	46,182	0.002	0.982	0.848

components at the left are smaller than $k\sigma_n$. This measure represents the background if the Gaussian n' explains the background of the scene. In fact, the weight $\alpha_{n'}$ is high. This Gaussian is then updated:

$$\alpha'_n \leftarrow (1 - \delta)\alpha'_n + \delta \quad (6)$$

$$\mu'_n \leftarrow (1 - \rho'_n)\mu'_n + \rho'_n \mathbf{z}_t \quad (7)$$

$$\sigma_n^{2'} \leftarrow (1 - \rho'_n)\sigma_n^{2'} + \rho'_n(\mathbf{z}_t - \mu'_n)^T(\mathbf{z}_t - \mu'_n) \quad (8)$$

$$\rho'_n \leftarrow \delta \mathcal{N}(\mathbf{z}_t | \mu'_n, \sigma_n^{2'}) \quad (9)$$

where δ is the learning coefficient, and represents adaptation time of the model. For all Gaussians $n \neq n'$, mean and variance are not modified, but: $\alpha_n \leftarrow (1 - \delta)\alpha_n$. If the test of Eq. (5) fails, pixel is associated to the foreground. The Gaussian with the smallest weight is reinitialized with current measure: $\alpha_n = \delta$, $\mu_n = \mathbf{z}_t$, $\sigma_n^2 = \bar{\sigma}^2$, where $\bar{\sigma}^2$ is a high variance. We also apply those affectations for GMM initialization.

GMM has shown good performance for the analysis of outdoor scenes, and has become a very popular BS algorithm. Even if this method is able to handle with low illumination variations, rapid variations of illumination and shadows are still problematic. Furthermore, the learning stage can be inefficient if it is realized with

noisy video frames. To tackle these problems, many authors have studied the various ways to improve GMM-based BS.

Kaewtrakulpong and Bowden [25] propose to modify the update equations in this model to improve the adaptation of the system to illumination variations. Then, the layer model introduced by Tuzel et al. [41] is based on 3D multivariate Gaussian distributions. With a recursive Bayesian learning approach, they are able to estimate the distribution of the variance and the mean of the model. Chen et al. [10] propose a hierarchical and block-based BS algorithm combining GMM and a contrast histogram. Zivkovic [48,49] automatically computes the correct numbers of Gaussian distributions for each pixel, instead of setting it constant. More than simply use color to describe pixels' contents, [45] use a combination of features.

2.3. Fuzzy models

Recently, some authors have introduced *fuzzy* concepts in the different steps of the background subtraction process [4]. In [46], the authors perform the background subtraction by the similarity measure of color and texture features of the input image and the background model using Sugeno Integral [39]. Later, [14] got better results using Choquet Integral [11]. Subsequently [1] have used the

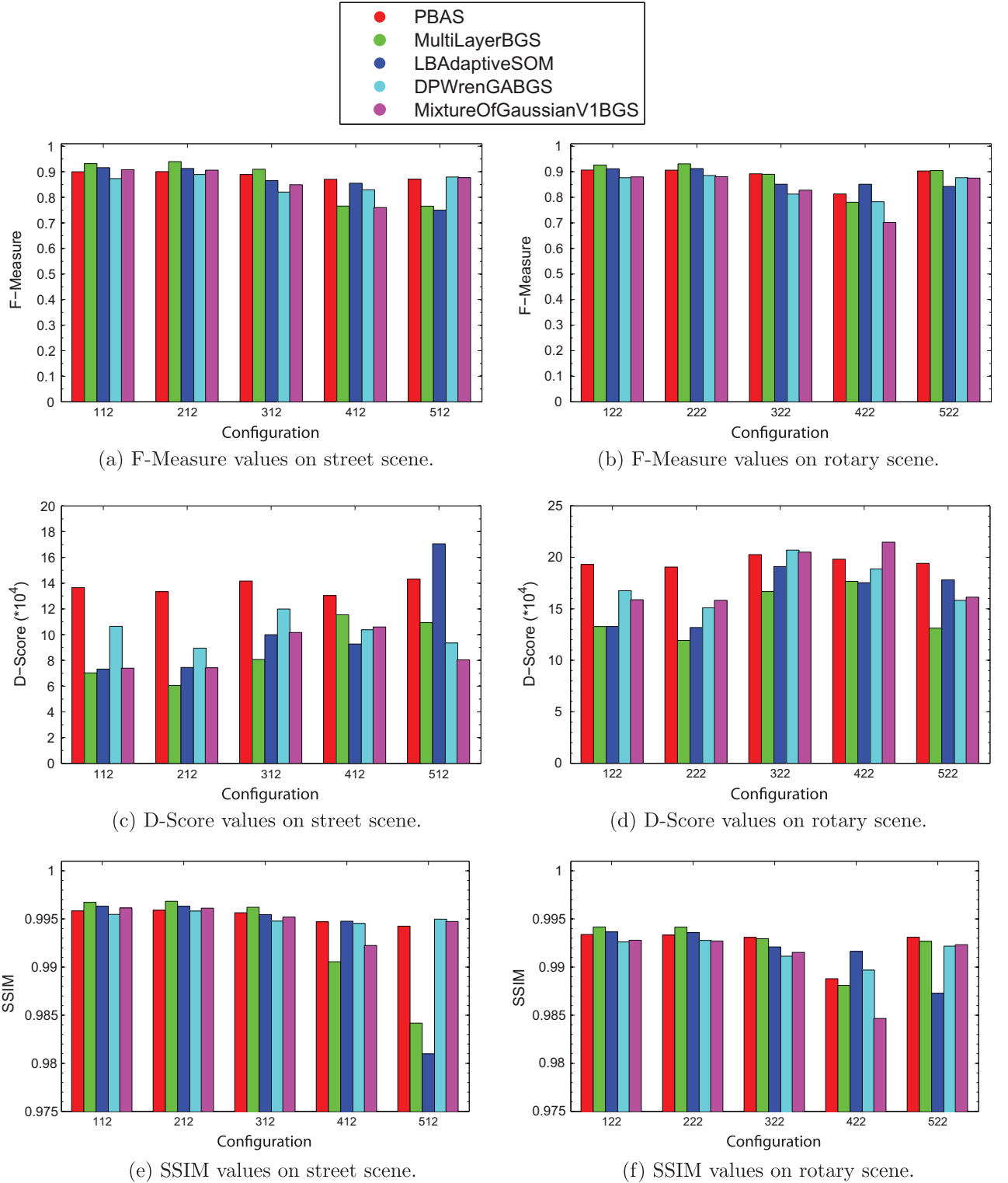


Fig. 5. F-Measure, D-Score and SSIM values of the top 5 best methods on BMC data set. PBAS in red, MultiLayerBGS in green, LBAdaptiveSOM in blue, DPWrenGABGS in cyan and MixtureOfGaussianV1BGS in magenta. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Choquet Integral with color, edge and texture features. In [34], the authors have proposed a fuzzy function to compute the foreground extraction and to update the background model. To update the background model with *Fuzzy Running Average*, [34] suggests the use of a linear saturation function instead of the crisp limiter function defined by applying:

$$\mathbf{S}_n(x, y) = \begin{cases} 1 & \text{if } d(\mathbf{Z}_t(x, y), \mathbf{B}_{t-1}(x, y)) > T \\ \left| \frac{\mathbf{Z}_t(x, y) - \mathbf{B}_{t-1}(x, y)}{T} \right| & \text{otherwise} \end{cases}, \quad (10)$$

where T is the threshold. However, Eq. (10) results in a foreground mask with real values $[0, 1]$. To determine the binary foreground mask, [34] suggests the use of a low-pass filter (LPF) by:

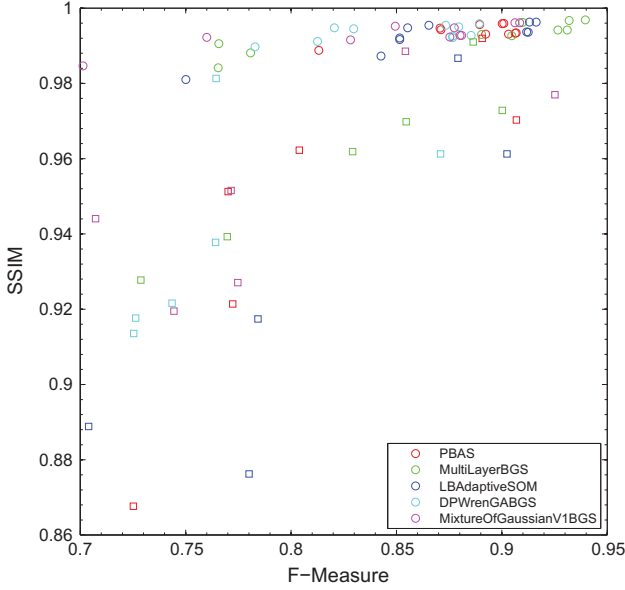


Fig. 6. F-Measure versus SSIM values from the top 5 best BS algorithms on all videos of BMC data set. The rounded points are the values from the synthetic videos and the squared points from the real world videos.

$$\mathbf{F}_t(x, y) = \begin{cases} 1 & \text{if } |LPF(\mathbf{S}_t(x, y))| > T \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The advantage of the Eq. (11) is the noise reduction in the foreground mask. In [5,16,17], the authors suggest a type-2 fuzzy method to deal with uncertainties of the multimodal background. [35] have used a set of fuzzy rules to improve the detection of moving objects. One could consult [4] for a full review of the fuzzy based approaches in the BS field.

2.4. Neural and neuro-fuzzy methods

Basically the neural network learns to classify each pixel of the image. For each pixel, the neural network determines if one pixel belongs to the foreground or to the background [3]. In [12], the authors have used a multilayered feed-forward neural network with 124 neurons. The background segmentation approach proposed by [12] relies on an adapted PNN (Probabilistic Neural Networks). The background model is learned by neural network while a Bayesian classifier identifies if one pixel belongs to the foreground or to the background. [29] have used a Self-Organizing Map (SOM) network to perform BS. Each pixel has one neural map given by weight vectors. Later, [30] has improved the previous work by adding a fuzzy function in background learning step. The previous authors also apply a spatial coherence analysis on the SOM network of each pixel to enhance robustness against false detections and to deal with decision problems typically arising when crisp settings are involved.

2.5. Other models

BS has also been achieved by many other methodologies. We can cite for example the computation of eigen values and eigen vectors [32] or the combination of histograms and Bayesian inference [18]. An original approach, *VuMeter*, proposed by [19], a non-parametric model, is based on a discrete estimation of the probability distribution. It is a probabilistic approach to define the image background model. We recall that \mathbf{Z}_t is an image at time t , and \mathbf{z}_t gives the color vector RGB of a given pixel in \mathbf{Z}_t . A pixel can take two states, ω_1 if the pixel is background, ω_2 if the pixel is foreground. This method tries to estimate $p(\omega_1|\mathbf{z}_t)$. With 3 color

component i (Red, Green, Blue for example), the probability density function can be approximated by:

$$p(\omega_1|\mathbf{z}_t) = \prod_{i=1}^3 p(\omega_1|z_t^i), \quad (12)$$

with

$$p(\omega_1|z_t^i) \approx k_i \sum_{j=1}^n \pi_t^{ij} \delta(b_t^i - j), \quad (13)$$

where δ is the Kronecker delta function, \mathbf{b}_t gives the bin index vector associated to \mathbf{z}_t , j is a bin index, and k_i is a normalization constant to keep at each moment

$$\sum_{j=1}^n \pi_t^{ij} = 1. \quad (14)$$

π_t^{ij} is a discrete mass function which is represented by a bin. At first image ($t = 1$), we set bins values, $\pi_0^{ij} = 1/n$ to have a sum to 1 like in Eq. (14). At each new pixel, its value match to a bin π_t^{ij} . The level of this bin is updated as follows:

$$\pi_{t+1}^{ij} = \pi_t^{ij} + \alpha \cdot \delta(b_{t+1}^i - j), \quad (15)$$

where α is the learning rate. After the learning phase, the bins which are modeling the background have a high value. To choose at each moment if a pixel is background or not, a threshold t is set by an supervised way. Each new pixel with corresponding bins under this threshold will be detected as background. In RGB mode each pixel will be modeled by 3 *VuMeter* (1 for each color). To consider a pixel as background, it must be detected as background with each *VuMeter*.

3. Materials and methods

3.1. The BGSLibrary

The BGSLibrary [36] provides an easy-to-use C++ framework to perform background subtraction. Currently, the library offers 29 background subtraction algorithms. The source code is available under GNU GPL v3 license and the library is free, open source and platform independent. The BGSLibrary also provides one Java based GUI (Graphical User Interface) allowing the users to configure the input video-source, region of interest, and the parameters of each BS algorithm. To use the library, it is necessary to have the OpenCV² library installed.

Fig. 2 shows one Windows MFC (Microsoft Foundation Class) application using the BGSLibrary. Table 1 shows all algorithms with respective author(s) available in the library. The algorithms are grouped by similarity. The column Method ID is the implementation reference of the method name. Some algorithms of BGSLibrary have been used successfully in [37].

3.2. The BMC data set

The BMC (Background Models Challenge) data set proposed by Vacavant et al. [42] are composed of both synthetic and real videos, representing urban scenes acquired from a static camera. The data set is composed of 20 urban video sequences rendered with the SiVIC simulator [13]. These videos are representations of two scenes: a street and a rotary. We sum up in Table 2 the different situations that are generated from these two scenes in the BMC data set. The *Learning* set is composed of the 10 synthetic videos representing the use case 1. Each video is numbered according to scene number (1 or 2), the presented event type (from 1 to 5),

² <http://opencv.org/>.

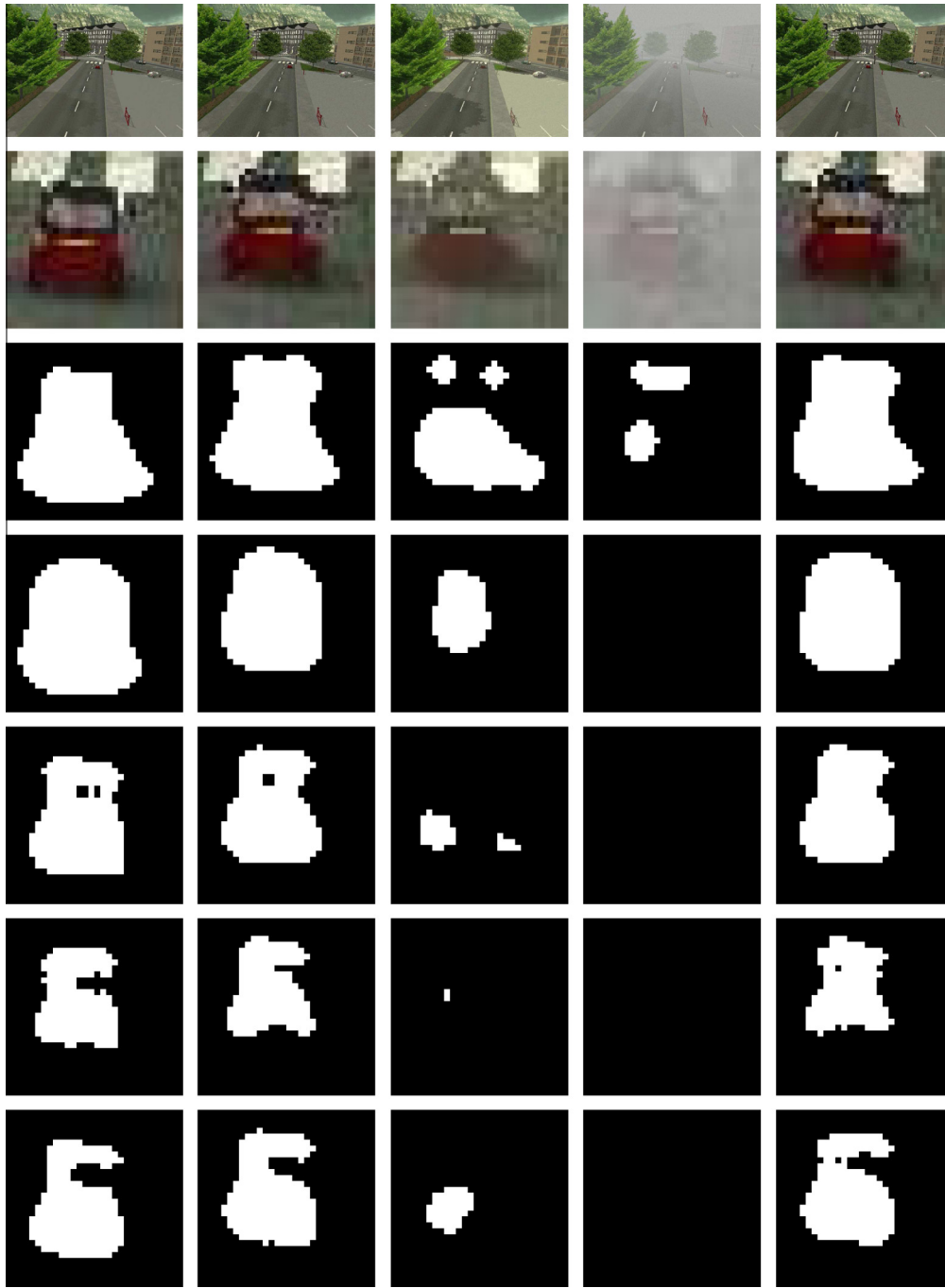


Fig. 7. Foreground masks obtained from the top 5 BS algorithms on all sequences of the street scene. From the first to last row: input frame, region of interest of the frame, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and the use case (1 or 2). For example, the video 311 of our benchmark describes a sunny street, under the use case 1, as illustrated in Fig. 3 (top-left).

The *Evaluation* set first contains the 10 synthetic videos with use case 2. This set is also composed of real videos acquired from static cameras in video-surveillance contexts. This dataset has been built in order to test the algorithms reliability during time and in difficult situations such as outdoor scenes. Real long videos (about one hour and up to four hours) are available, and they may present long time change in luminosity with small density of objects in time compared to previous synthetic ones. This dataset allows to test the influence of some difficulties encountered

during the object extraction phase. Those difficulties have been sorted according to:

1. The ground type (bitumen, ballast or ground).
2. The presence of vegetation (trees for instance).
3. Casted shadows.
4. The presence of a continuous car flow near to the surveillance zone.
5. The general climatic conditions (sunny, rainy and snowy conditions).
6. Fast light changes in the scene.
7. The presence of big objects.

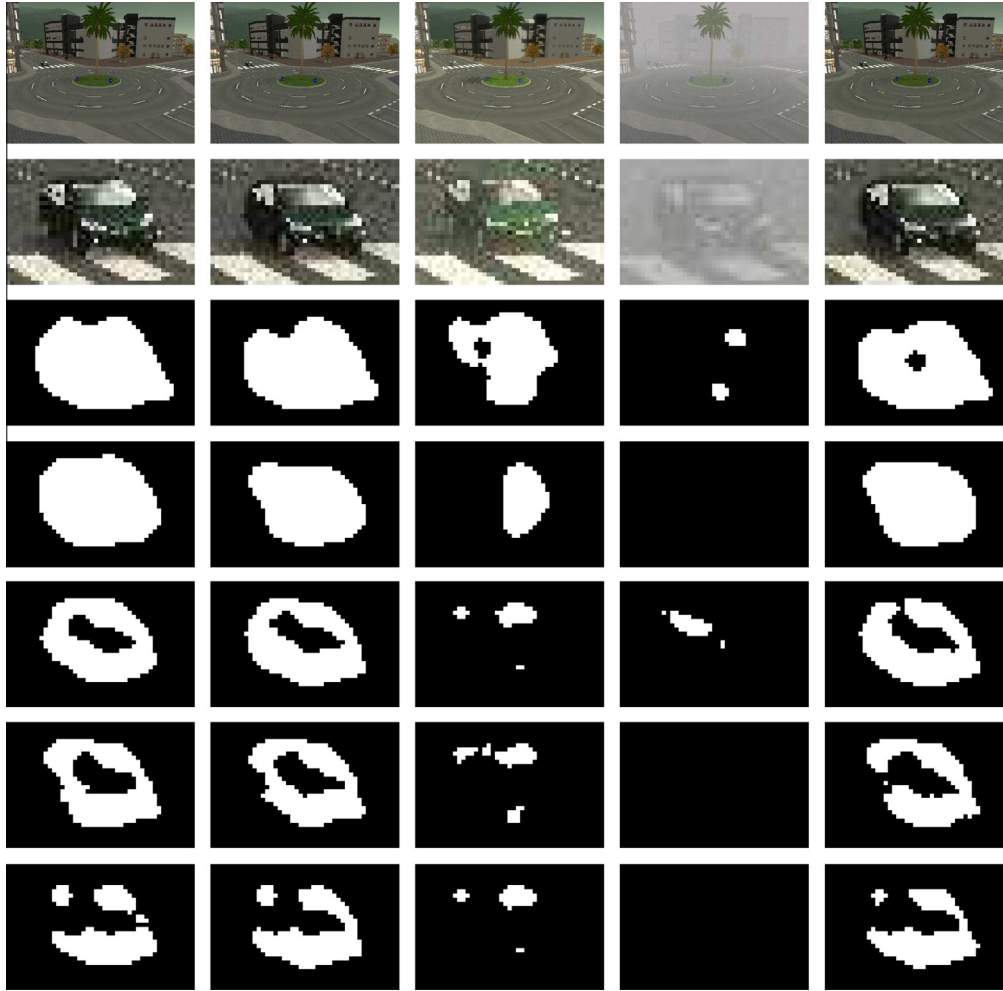


Fig. 8. Foreground masks obtained from the top 5 BS algorithms on all sequences of the rotary scene. From the first to last row: input frame, region of interest of the frame, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS.

Samples of images extracted from those real videos are presented in Fig. 3. For each of these videos have been manually segmented some representative frames that can be used to evaluate a BS algorithm. In the evaluation phase of the BMC data set, no ground truth image is publicly available, and authors should test their BS method with the parameters they have set in the learning phase.

4. Experimental evaluation

In this section, the benchmark and performance evaluation of all BS algorithms are shown. Firstly, the parameter tuning process is presented, followed by the benchmark and performance evaluations.

4.1. Parameter tuning

As explained above in Section 3.2, the BMC data set is composed of two distinct sets of sequences: learning and evaluation. The learning videos are used for parameter tuning of the BGSLibrary algorithms. The BMC data set provides the ground truth of each video, where background pixel are set to 0, and foreground to any other value. The foreground masks generated by each BS method were stored, and then the BMC Wizard software³ has been used

to compute all the quality measures. The parameters of each BS algorithm were adjusted manually to get the best measures. Table 3 shows the parameter settings of each BS algorithm after the learning phase.

4.2. Benchmark evaluation

When selecting a background subtraction method, it is important to evaluate its memory consumption, execution time and CPU occupancy. In embedded systems or real time applications, these features can be critical. In our experiments, we have used an Intel Core i5-2410M 2.3 GHz processor with 4 GB DDR3 RAM and Windows 7 Home Premium x64 SP1. In Fig. 4 is presented the average occupancy of CPU, RAM and execution time of each BS algorithm with all features normalized. Note that in Fig. 4 there are 26 methods rather than 29 methods available in BGSLibrary. In this experiment the benchmark of T2FGMM method includes T2FGMM_UM and T2FGMM_UV, and the T2FMRF method was not evaluated because its implementation has a memory leak failure⁴ with some very long videos provided by BMC.

⁴ A memory leak is unnecessary memory consumption by a computer program. If a program with a memory leak runs long enough, it can eventually run out of usable memory.

³ <http://bmc.univ-bpclermont.fr/?q=node/7>.

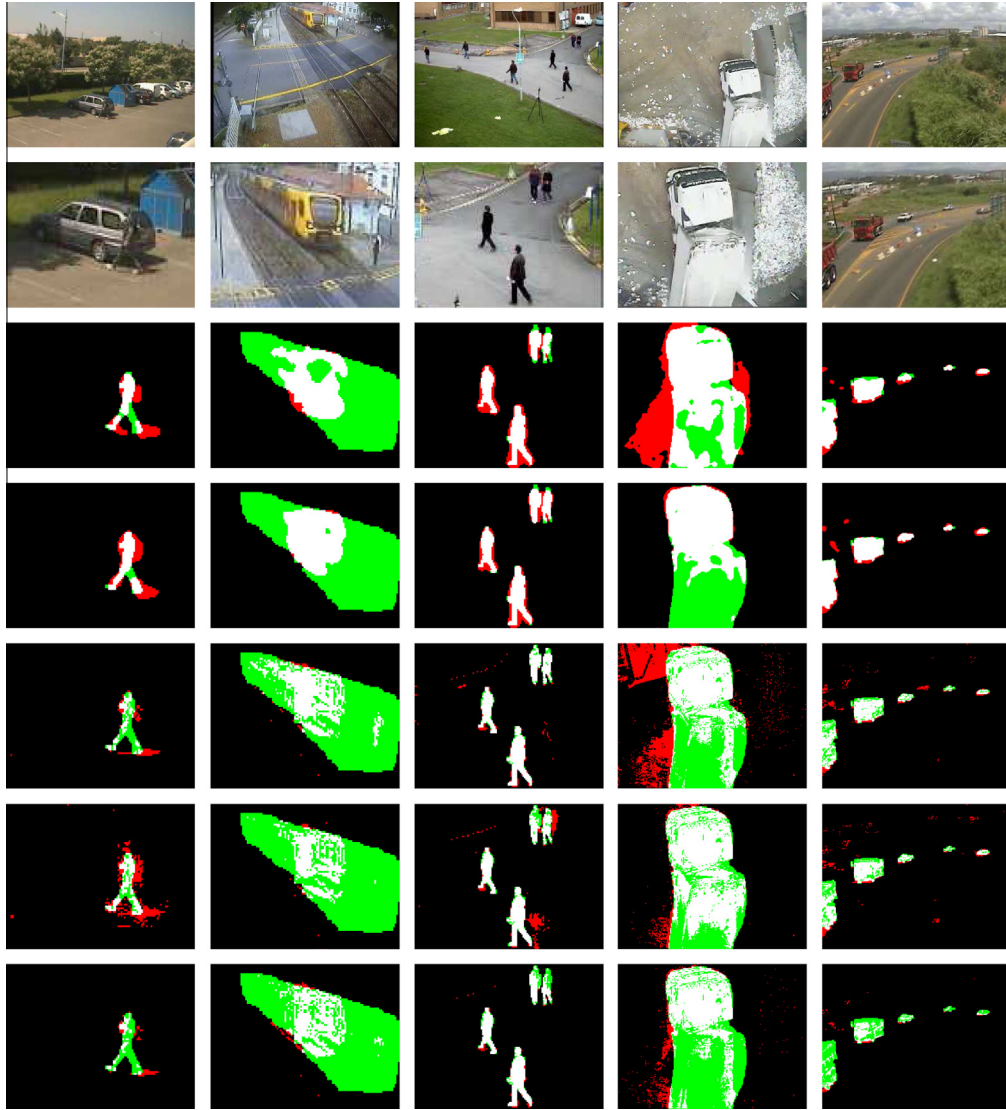


Fig. 9. Foreground masks obtained from the top 5 BS algorithms on all sequences of the real world videos. From the first to last row: input frame, region of interest in the frame, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. The TP pixels are in white, TN pixels in black, FP pixels in red and FN pixels in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6

Performance of the best BS algorithms in each group while moving objects stop suddenly. Each metric was split into two columns where the results with default parameters are shown in the first column and the second column presents the results after parameter tuning.

Methods	TP		FN		FP		TN	
PBAS	0	785	888	103	105	569	75,807	75,343
$N = 40$, $T_{init} = 180$								
MultiLayerBGS	0	825	888	63	1	1131	75,911	74,781
$SW = 0.5$, $HL = 10.2$, $FD = 1$, $DA = 20$								
LBAdaptiveSOM	700	581	188	307	264	86	75,648	75,826
$\rho = 100$, $LF = 50$								
DPWrenGABGS	328	652	560	236	30	508	75,882	75,404
$T = 5.25$, $\alpha = 0.001$, $LF = 40$								
MixtureOfGaussianV1BGS	0	92	888	796	0	0	75,912	75,912
$T = 1$, $\alpha = 0$								
AdaptiveBackgroundLearning	40	704	848	184	43	407	75,869	75,505
$\alpha = 0.01$								
FuzzyChoquetIntegral	43	323	845	565	440	641	75,472	75,271
$T = 0.77$								
T2FGMM_UM	0	459	888	429	0	219	75,912	75,693
$T = 1$, $K_m = 0.5$								
DPEigenbackgroundBGS	748	536	140	352	548	70	75,364	75,842
$T = 1000$								

SW = shadowRate, HL = highlightRate, FD = frameDuration, DA = detectAfter.

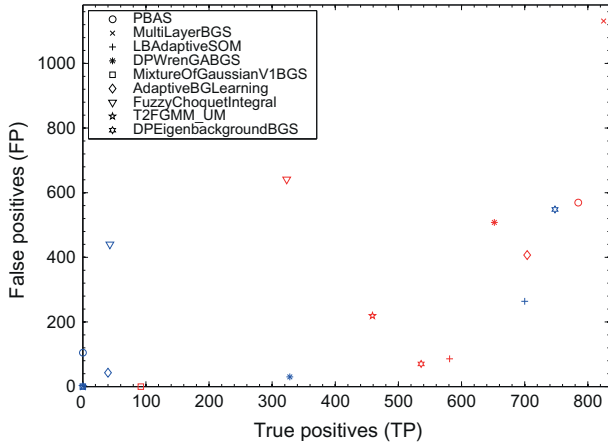


Fig. 10. True positive versus false positive pixels from the best BS algorithms in each group while moving objects stop suddenly. The results with default parameters is shown in blue and the results after the parameter tuning is depicted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Quality evaluation

In general, the basic BS methods are fast (low computational cost), but have a low precision. In our performance evaluation, several criteria have been considered, and represents different kinds of quality of a BS algorithm. We have taken into account the measures computed thanks to the BMC Wizard: *F*-score, PSNR (Peak Signal–Noise Ratio), SSIM (Structural SIMilarity) [43], and *D*-Score [28].

4.3.1. Static quality metrics

The Precision, Recall and *F*-Measure are based on the amount of false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN) as shown in Table 4.

We also compute the PSNR, defined by:

$$PSNR = \frac{1}{n} \sum_{i=1}^n 10 \log_{10} \frac{m}{\sum_{j=1}^m \|S_i(j) - G_i(j)\|^2} \quad (16)$$

where $S_i(j)$ is the j th pixel of image i (of size m) in the sequence S (with length n).

4.3.2. Application quality metrics

We also consider the problem of background subtraction in a visual and perceptual way. To do so, we use the gray-scale images of the input and ground truth sequences to compute the perceptual measure SSIM (Structural SIMilarity), given by [43]:

$$SSIM(S, G) = \frac{1}{n} \sum_{i=1}^n \frac{(2\mu_{S_i}\mu_{G_i} + c_1)(2cov_{S_iG_i} + c_2)}{(\mu_{S_i}^2 + \mu_{G_i}^2 + c_1)(\sigma_{S_i}^2 + \sigma_{G_i}^2 + c_2)}, \quad (17)$$

where μ_{S_i}, μ_{G_i} are the means, $\sigma_{S_i}, \sigma_{G_i}$ the standard deviations, and $cov_{S_iG_i}$ the covariance of S_i and G_i . In our benchmark, we set $c_1 = (k_1 \times L)^2$ and $c_2 = (k_2 \times L)^2$, where L is the size of the dimension of the signal processed (that is, $L = 255$ for gray-scale images), $k_1 = 0.01$ and $k_2 = 0.03$ (which are the most used values in the literature).

We finally use the *D*-Score [28], which consists in considering localization of errors according to real object position. As Baddeleys distance, it is a similarity measure for binary images based on distance transform. To compute this measure we only consider mistakes in BSA results. Each error cost depends on the distance with the nearest corresponding pixel in the ground-truth. As a

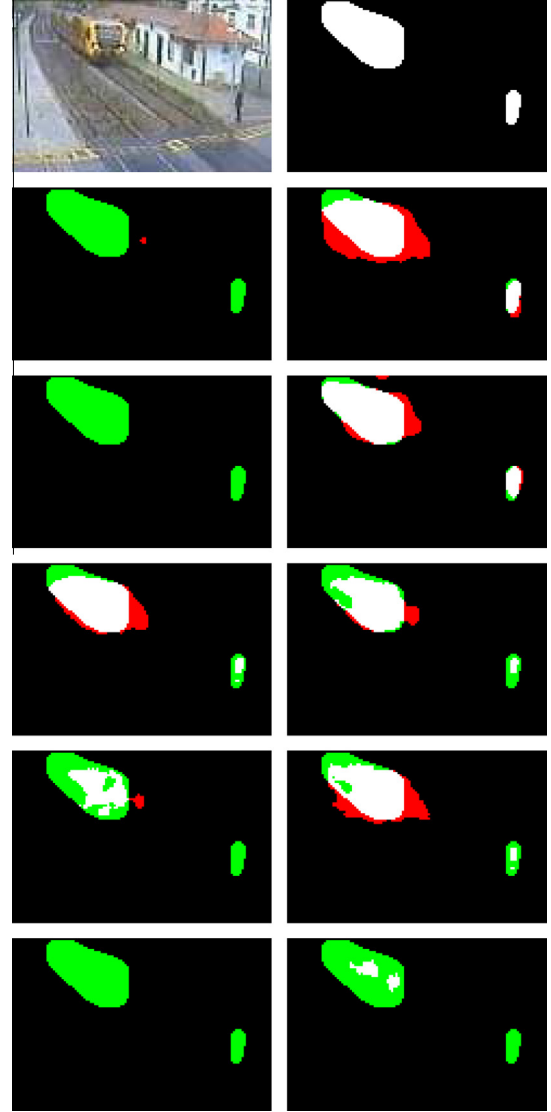


Fig. 11. Foreground masks obtained from the best BS algorithms in each group while moving object stops suddenly (Part 1). From the first to last row: input frame and ground truth, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. The first column shows the foreground with default parameters and the second column presents the foreground masks after tuning.

matter of fact, for object recognition, short or long range errors in segmentation step are less important than medium range error, because pixels on medium range impact greatly on object's shape. Hence, the penalty applied to medium range errors is heavier than the one applied to those in a short or large range.

More precisely, the *D*-Score is computed by using:

$$D - score(S_i(j)) = \exp \left((-\log_2(2.DT(S_i(j)) - 5/2)^2) \right) \quad (18)$$

where $DT(S_i(j))$ is given by the minimal distance between the pixel $S_i(j)$ and the nearest reference point (by any distance transformation algorithm). With such a function, we punish errors with a tolerance of 3 pixels from the ground-truth, because these local errors do not really affect the recognition process. For the same reason, we allow the errors that occur at more than a 10 pixels distance. Details about such metric can be found in [28]. Few local/far errors will produce a near zero *D*-Score. On the contrary, medium range errors will produce high *D*-Score. A good *D*-Score has to tend to 0.

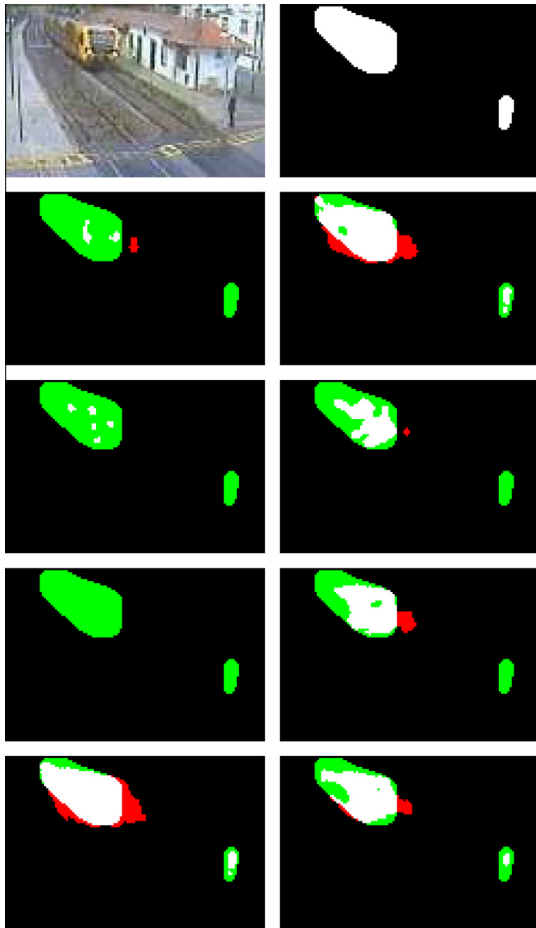


Fig. 12. Foreground masks obtained from the best BS algorithms in each group while moving objects stop suddenly (Part 2). From the first to last row: input frame and ground truth, AdaptiveBackgroundLearning, FuzzyChoquetIntegral, T2FGMM_UM and DPEigenbackgroundBGS. The first column shows the foreground with default parameters and the second column presents the foreground masks after tuning.

4.3.3. Combination of metrics

To perform the ranking of all BS algorithms evaluated here, in the first place the *F*-Measure, *D*-Score and SSIM have been normal-

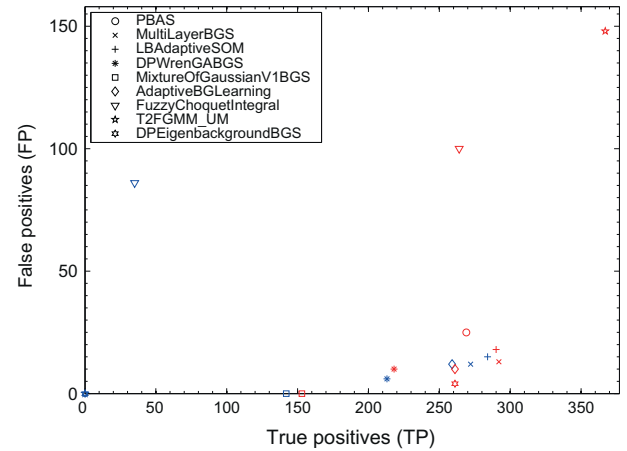


Fig. 13. True positive versus false positive pixels from the best BS algorithms in each group while sudden light changes occur. The results with default parameters is shown in blue and the results after the parameter tuning is depicted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ized in the range $[0, \dots, 1]$. The global quality of a BS algorithm is then obtained thanks to the FSD score defined by:

$$FSD(a) = \frac{n(\overline{FMeasure}(a)) + n(\overline{SSIM}(a)) + (1 - n(\overline{DScore}(a)))}{3}$$

where a is the BS method, $\overline{FMeasure}(a)$, $\overline{SSIM}(a)$ and $\overline{DScore}(a)$ are the average *F*-Measure, SSIM and *D*-Score of the BS algorithm throughout the data set.

Table 5 shows global score of the BS algorithms. Moreover, Fig. 5 shows the *F*-Measure, *D*-Score and SSIM values of the top 5 best methods on BMC data set, and Fig. 6 shows the *F*-Measure versus SSIM values. Finally, Fig. 7 and Fig. 8 display the foreground masks obtained from the top 5 BS algorithms on all sequences of the street scene.

4.3.4. Performance analysis across specific complex situations

In this section we analyze the performance of the best BS algorithms in four specific situations. We evaluate them while moving objects stop suddenly, while sudden light changes occur, when complex cast shadows appear and with moving trees or dynamic

Table 7

Performance of the best BS algorithms in each group while sudden light changes occur. Each metric was split into two columns where the results with default parameters is shown in the first column and the second column presents the results after parameter tuning.

Methods	TP		FN		FP		TN	
PBAS	297	269	90	118	1099	25	75,314	76,388
$N = 40$								
MultiLayerBGS	272	292	115	95	12	13	76,401	76,400
$MLR = 0.1, WLR = 0.1, MW = 0.01$								
LBAdaptiveSOM	284	290	103	97	15	18	76,398	76,395
$\rho = 70$								
DPWrenGABGS	213	218	174	169	6	10	76,407	76,403
$T = 10.025, \alpha = 0.009, LF = 100$								
MixtureOfGaussianV1BGS	142	153	245	234	0	0	76,413	76,413
$T = 5, \alpha = 0.01$								
AdaptiveBackgroundLearning	259	261	128	126	12	10	76,401	76,403
$\alpha = 0.05$								
FuzzyChoquetIntegral	35	264	352	123	86	100	76,327	76,313
$T = 0.87, \alpha_{learn} = 0.5, \alpha_{update} = 0.05$								
T2FGMM_UM	0	367	387	20	0	148	76,413	76,265
$T = 1, \alpha = 0.05, K_m = 0.5$								
DPEigenbackgroundBGS	275	261	112	126	4991	4	71,422	76,409
$HS = 40, ED = 20$								

MLR = modeLearnRate, WLR = weightLearnRate, MW = modeWeight.

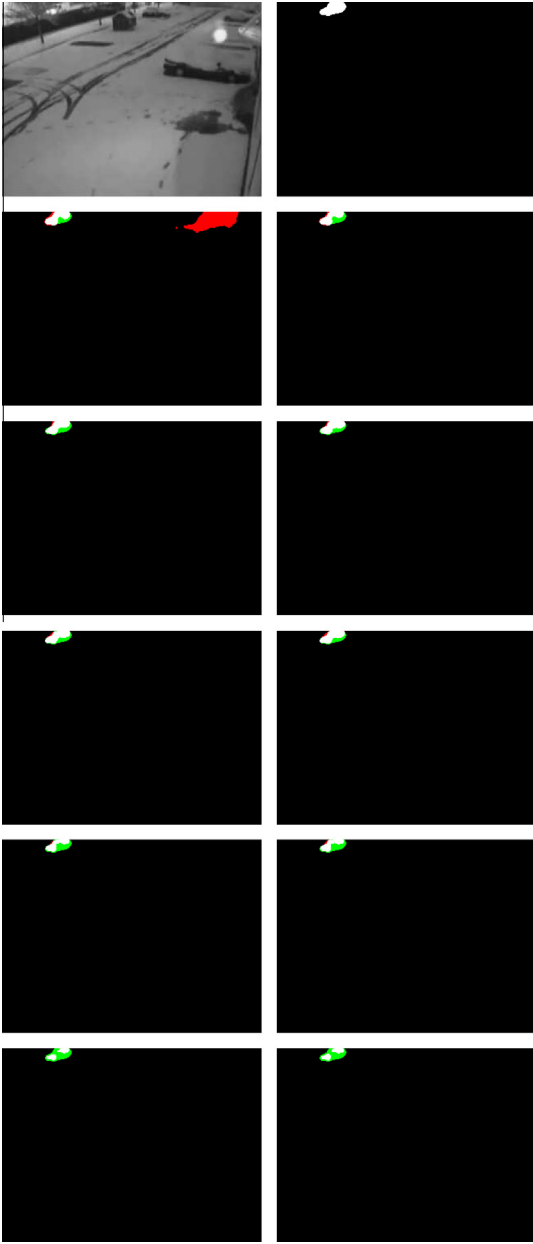


Fig. 14. Foreground masks obtained from the best BS algorithms in each group while moving objects stops suddenly (Part 1). From the first to last row: input frame and ground truth, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. The first column shows the foreground with default parameters and the second column shows the foreground masks after tuning.

background. These four specific situations are very hard to deal and very common in indoor and outdoor environments. All methods are evaluated with their default parameters and we propose then to tune them so that they handle these pathological cases at best. The parameter tuning has been performed experimentally in order to maximize the number of true positives pixels and minimize the number of false positive pixels.

4.3.4.1. While moving objects stop suddenly. In this evaluation we have used one video where a train stops for a certain period of time. Several background subtraction algorithms fail in this situation because the train can be included in the background model while it is updated. Table 6 shows the number of true positive (TP), false negative (FN), false positive (FP) and true negative

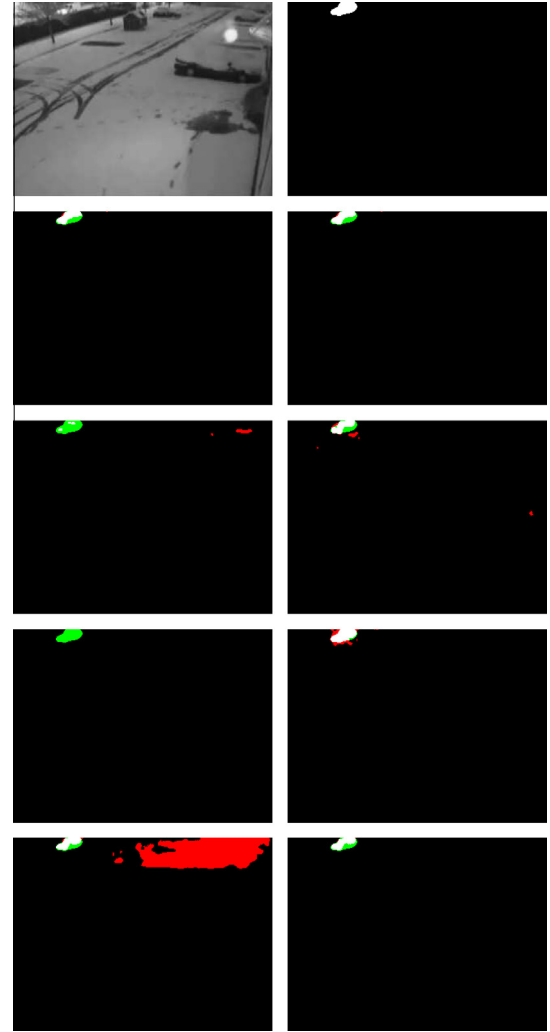


Fig. 15. Foreground masks obtained from the best BS algorithms in each group while moving objects stops suddenly (Part 2). From the first to last row: input frame and ground truth, AdaptiveBackgroundLearning, FuzzyChoquetIntegral, T2FGMM_UM and DPEigenbackgroundBGS. The first column shows the foreground with default parameters and the second column shows the foreground masks after tuning.

(TN) pixels before and after the parameter tuning. The Fig. 10 shows the number of TP pixels versus FP pixels for each algorithm. As it can be observed in Table 6, the number of TP pixels have increased after parameter tuning as expected. However, sometimes the number of FP pixels increases also (see MultiLayerBGS). The best method must have few FP and many TP pixels. In this experiment, the LBAdaptiveSOM and AdaptiveBackgroundLearning algorithms show interesting results, since it is possible to improve their good detection without increasing false ones too much. However, no algorithm was completely able to handle this very hard event in the video, even with an adequate adjustment of their parameters. Figs. 11 and 12 illustrate the foreground masks obtained from the BS algorithms.

4.3.4.2. While sudden light changes occur. In this evaluation we have used one video where a sudden light changes occur. Some BS algorithms fail in this situation due their low noise tolerance. As we can notice in Table 7, the number of TP pixels has increased after parameter tuning as expected and most of FP pixels have decreased. All algorithms (except the PBAS and DPEigenback-

Table 8

Performance of the best BS algorithms in each group while cast shadows appear. Each metric was split into two columns where the results with default parameters is shown in the first column and the second column presents the results after parameter tuning.

Methods	TP	FN	FP	TN
PBAS	821	483	242	580
$R_{lower} = 36$, $T_{init} = 36$				
MultiLayerBGS	737	586	326	477
$SW = 0.3$				
LBAdaptiveSOM	748	437	315	626
$\rho = 135$				
DPWrenGABGS	528	412	535	651
$T = 22.25$				
MixtureOfGaussianV1BGS	405	265	658	798
$\alpha = 0.1$				
AdaptiveBackgroundLearning	809	628	254	435
$T = 30$, $\alpha = 0.1$				
FuzzyChoquetIntegral	271	526	792	537
$T = 8.7$, $LF = 30$, $\alpha_{learn} = 0.5$, $\alpha_{update} = 0.05$, $CS = YCrCb$				
T2FGMM_UM	74	261	989	802
$T = 2$, $K_m = 0.5$, $\alpha = 0.05$				
DPEigenbackgroundBGS	789	483	274	580
$T = 1225$, $HS = 40$, $ED = 20$				

SW = shadowRate, CS = colorSpace.

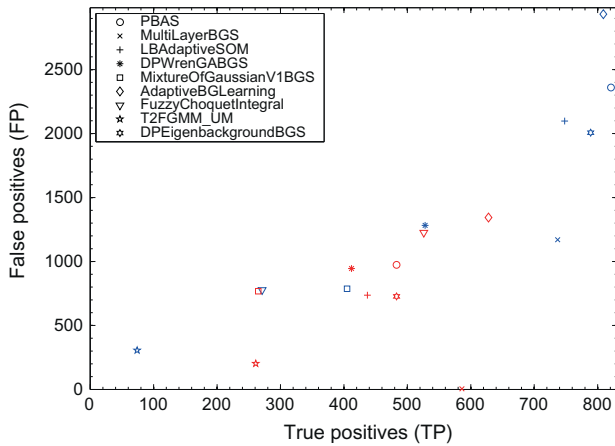


Fig. 16. True positive versus false positive pixels from the best BS algorithms in each group while a cast shadow appears. The results with default parameters is shown in blue and the results after the parameter tuning is depicted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

groundBGS) have achieved good results with their default parameters. The PBAS only have achieved a better result after the parameter tuning. In this experiment, the LBAdaptiveSOM and MultiLayerBGS algorithms have shown interesting results, since it has been possible to find a good compromise between increase of TP pixels and the increase of FP ones. Fig. 13 shows the number of TP pixels versus FP pixels for each algorithm, Figs. 14 and 15 show the foreground masks obtained from these BS algorithms.

4.3.4.3. While cast shadows appear. The occurrence of hard and soft shadows is also one of the challenging situations for many BS algorithms. In this evaluation we have used one video where one person walks in a dark environment and the lights produce a large shadow. In Table 8, we can observe that the number of FP pixels have decreased after parameter tuning as expected but the number of TP pixels have decreased slightly. In this experiment, the MultiLayerBGS algorithm have shown interesting results, as in

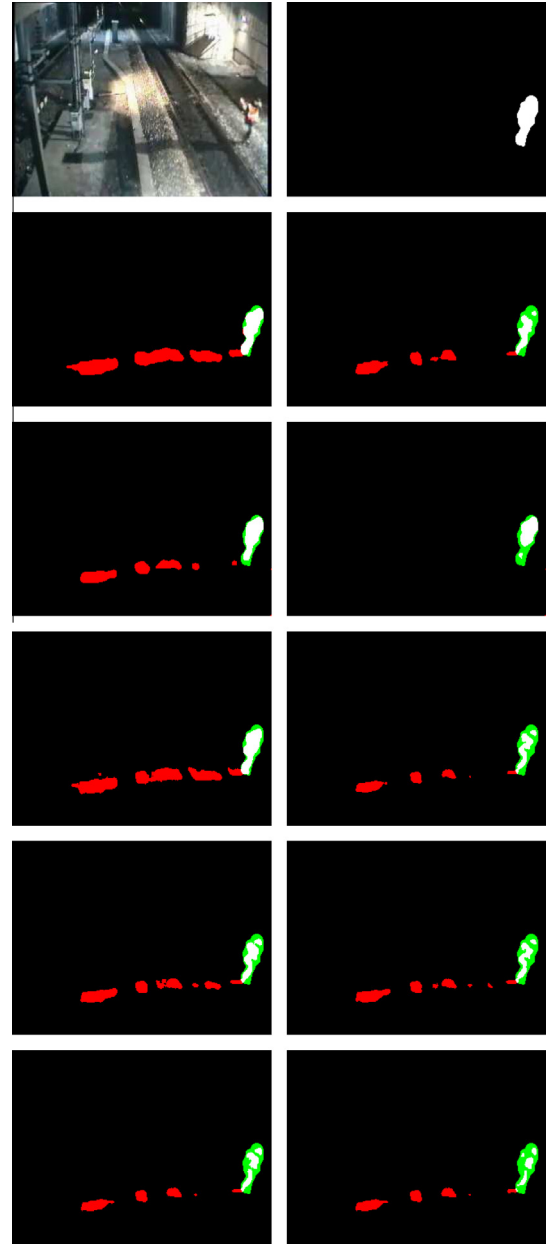


Fig. 17. Foreground masks obtained from the best BS algorithms in each group while a cast shadow appears (Part 1). From the first to last row: input frame and ground truth, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. The first column shows the foreground with default parameters and the second column presents the foreground masks after tuning.

the previous complex cases we have evaluated. It is important to note that MultiLayerBGS algorithm has one parameter called SW (Shadow Rate) specific for the occurrence of shadows. The MultiLayerBGS has successfully discarded the shadow in this experiment. The other methods have achieved only slight better results. For all the tested methods, we have not succeeded to remove the shadow completely without harming the detection of the TP pixels. The Fig. 16 shows the number of TP pixels versus FP pixels for each algorithm, Figs. 17 and 18 present the foreground masks obtained from the BS algorithms.

4.3.4.4. Moving trees or dynamic background. Dynamic backgrounds and moving trees are still an open challenge in BS. In this evaluation we have used one video where moving trees produce a

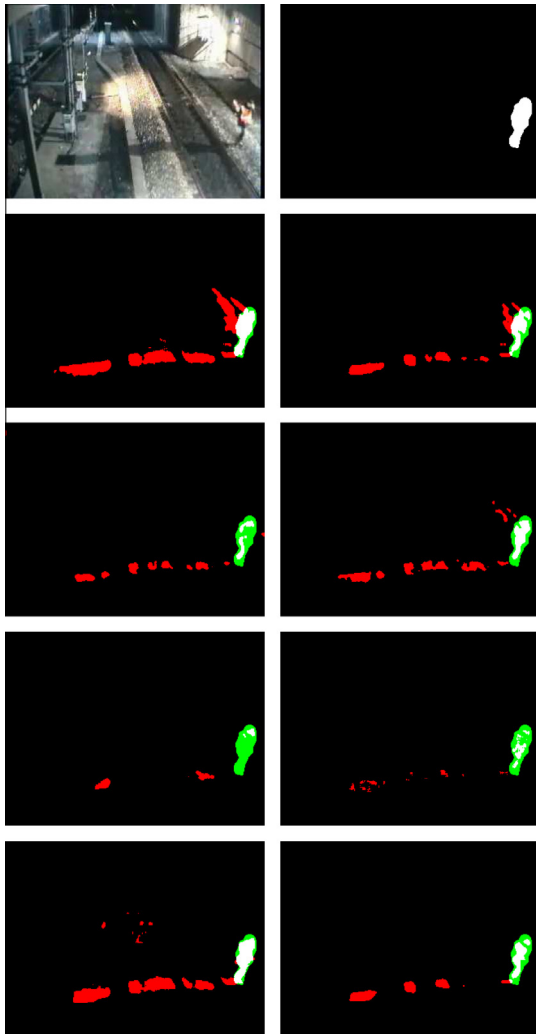


Fig. 18. Foreground masks obtained from the best BS algorithms in each group while a shadow occur (Part 2). From the first to last row: input frame and ground truth, AdaptiveBackgroundLearning, FuzzyChoquetIntegral, T2FGMM_UM and DPEigenbackgroundBGS. The first column shows the foreground with default parameters and the second column presents the foreground masks after tuning.

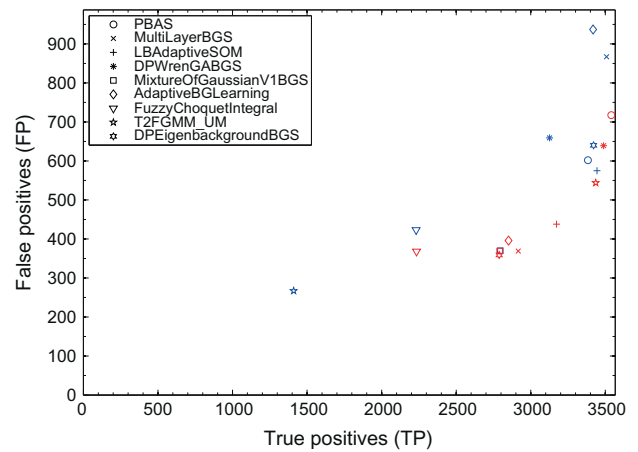


Fig. 19. True positive versus false positive pixels from the best BS algorithms in each group when background contains moving trees. The results with default parameters is shown in blue and the results after the parameter tuning is depicted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

dynamic background. We can notice in Table 9 that the number of FP pixels have decreased after parameter tuning as expected but the number of TP pixels have changed slightly for some methods. In this experiment, the PBAS, DPWrenGABGS, LBAdaptiveSOM and T2FGMM_UM algorithms have shown interesting results since a good compromise was found between the increase of TP and FP pixels. The other methods have achieved only a moderate improvement of their quality. Fig. 19 shows the number of TP pixels versus FP pixels for each algorithm, while Figs. 20 and 21 illustrate the foreground masks obtained from these BS algorithms.

5. Discussion

In this article, we have compared a wide range of BS algorithms from the literature, thanks to the BGSLibrary. From the benchmark provided by BMC, we have extracted a set of five methods that clearly overtake the other ones, as we can observe in Table 5. The ranking function we have introduced, combining the F , SSIM and D -Score measures allows us to determine the best methods of the tests. These methods cover a large period of time in the

Table 9

Performance of the best BS algorithms in each group when the background contain moving trees. Each metric was split into two columns where the results with default parameters is shown in the first column and the second column presents the results after parameter tuning.

Methods	TP		FN		FP		TN	
PBAS	3383	3541	234	76	602	717	72,581	72,466
$T_{inc} = 10$								
MultiLayerBGS	3508	2916	109	701	867	369	72,316	72,814
$BMP = 0.9$, $BPT = 0.4$, $BPUT = 0.4$								
LBAdaptiveSOM	3444	3173	173	444	575	438	72,608	72,745
$\rho = 100$								
DPWrenGABGS	3126	3488	491	129	659	639	72,524	72,544
$\alpha = 0.05$								
MixtureOfGaussianV1BGS	2796	2791	821	826	370	369	72,813	72,814
$\alpha = 0.1$								
AdaptiveBackgroundLearning	3418	2851	199	766	937	396	72,246	72,787
$T = 35$								
FuzzyChoquetIntegral	2231	2234	1386	1383	423	368	72,760	72,815
$\alpha_{learn} = 0.5$, $\alpha_{update} = 0.05$								
T2FGMM_UM	1410	3436	2207	181	267	544	72,916	72,639
$\alpha = 0.05$								
DPEigenbackgroundBGS	3421	2788	196	829	640	359	72,543	72,824
$T = 1000$								

BMP = bgModePercent, BPT = bgProbThreshold, $BPUT$ = bgProbUpdatingThreshold.

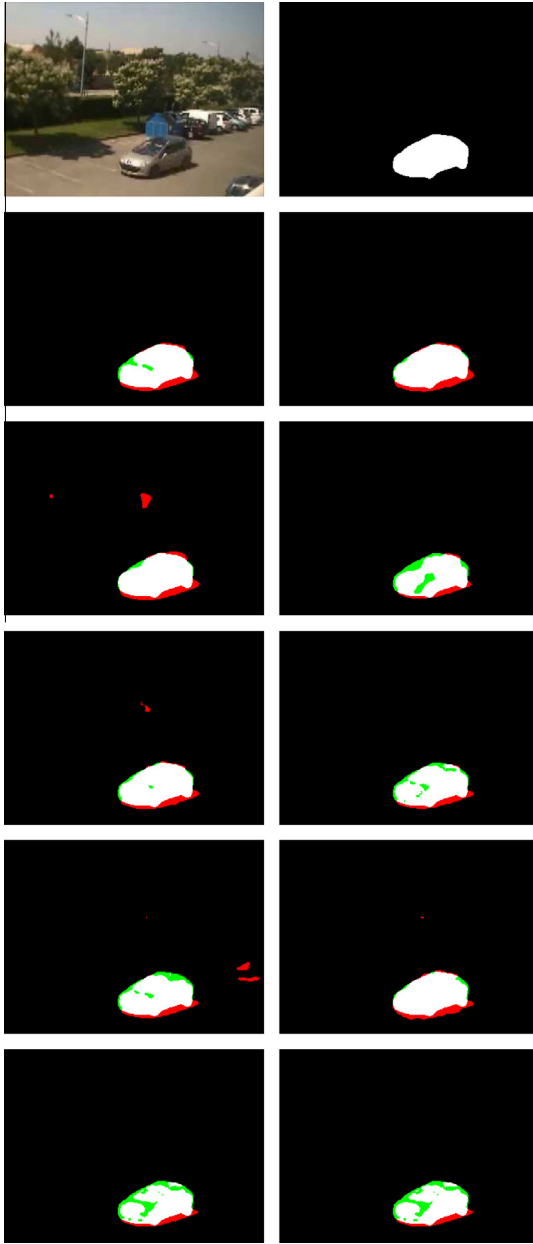


Fig. 20. Foreground masks obtained from the best BS algorithms in each group when background contains moving trees (Part 1). From the first to last row: input frame and ground truth, PBAS, MultiLayerBGS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. The first column shows the foreground with default parameters and the second column presents the foreground masks after tuning.

literature, Pfinder (DPWrenGABGS) introduced in 1997 [44] and MixtureOfGaussianV1BGS, the GMM improvement proposed by [25] in 2001 are very good BS methods. More recent algorithms like MultiLayerBGS [45], PixelBasedAdaptiveSegmenter [22] and LBAdaptiveSOM [29] have also shown an interesting robustness. This top 5 ranking can be confirmed thanks to the visual analysis we can conduct thanks to Figs. 7–9.

Fig. 5 informs us that the more discriminative measures in our ranking functions are D -Score and SSIM, whereas the F -measure is very similar for the five best methods. This can be explained by the fact that the F -measure is a statistical metric, only based on the count of well classified pixels (background/foreground). The SSIM is a structural measure taking into account more image information, and D -Score permits to highlight the distance between misclassified pixels and the ground truth.

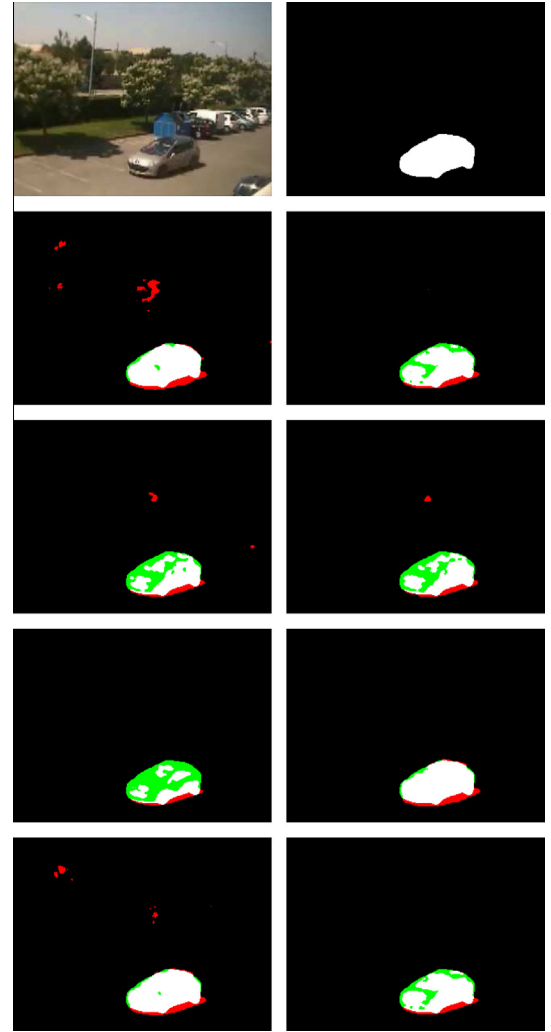


Fig. 21. Foreground masks obtained from the best BS algorithms in each group when background contains moving trees (Part 2). From the first to last row: input frame and ground truth, AdaptiveBackgroundLearning, FuzzyChoquetIntegral, T2FGMM_UM and DPEigenbackgroundBGS. The first column shows the foreground with default parameters and the second column presents the foreground masks after tuning.

We can observe from Fig. 6 that the real world videos are the hardest cases of the benchmark, leading to lower values of both SSIM and F -measure. We can also see that the best method like PixelBasedAdaptiveSegmenter do not always handle this very hard situations (associated square dots represent low F and SSIM values). The high robustness of the best methods is mostly due to their consistency in computing a good segmentation on a large set of videos.

Thanks to the visual illustrations we provide in Figs. 7–9, we can confirm the ranking of the five best methods of our tests. The PixelBasedAdaptiveSegmenter clearly produces the best foreground masks. Most algorithms critically fail for very complex situations, as fog and illuminations due to sun rising (third and fourth column of Figs. 7 and 8), casted shadows (first column of Fig. 9), or very high lighting changes (second column of Fig. 9). The learning of the background and foreground detection are very challenging in these scenes, and even the top 5 BS algorithms of our study are not completely able to handle them.

The very precise experiments we have conducted to study the complex situations in BS have shown that we can improve the robustness of the tested algorithms with an adequate parameter

tuning. However, we have also observed that these pathological cases constitute an open challenge for BS. As an illustration, Figs. 17 and 18 clearly demonstrate that even the best algorithms we have evaluated are not able to handle correctly the cast shadow appearing in the sequence. New mathematical and algorithmic ideas have to be found to compute BS, and mostly to solve this kind of complicated configurations.

All these remarks should be modulated by the processor and memory requirements needed for these methods. Indeed, Fig. 4 illustrates that memory and time consumption are very high for the best methods of our comparison (PixelBasedAdaptiveSegmenter, LBAdaptiveSOM and mostly MultiLayerBGS). A first future work that we would like to study is to integrate these performance evaluation into the ranking function we have developed in this article. We are convinced that the best BS algorithm should be efficient and fast (that is, tending to real-time functioning).

Besides the development of more BS algorithms in the BGSLibrary, we also plan to investigate the test of these methods on moving backgrounds. A scene acquired from a mobile camera (embedded in a car for example) is a real challenge for BS. Most algorithm are designed for static background, and the way of adapting them to dynamic scenes is an important task in the BS field.

We would also like to compare the BS approaches from BGSLibrary with the other benchmarks now available in the Web, like SABS or ChangeDetection.net. We plan to study the connections between all these benchmarks, and their relevance in the evaluation of BS techniques.

References

- [1] M.M. Azab, H.A. Shedeed, A.S. Hussein, A new technique for background modeling and subtraction for motion detection in real-time videos. In: IEEE International Conference on Image Processing (ICIP), 2010, pp. 3453–3456.
- [2] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, Review and evaluation of commonly-implemented background subtraction algorithms. In: IEEE International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4.
- [3] T. Bouwmans, Recent advanced statistical background modeling for foreground detection: A systematic survey, *Recent Patents Comput. Sci.* 4 (3) (2011) 147–176.
- [4] T. Bouwmans, Background subtraction for visual surveillance: a fuzzy approach, in: *Handbook on Soft Computing for Video Surveillance*, Taylor and Francis Group, 2012.
- [5] T. Bouwmans, F. El Baf, Modeling of dynamic backgrounds by type-2 fuzzy Gaussian mixture models, *MASJUM J. Basic Appl. Sci.* (2009).
- [6] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of Gaussians for foreground detection - a survey, *Recent Patents Comput. Sci.* 1 (3) (2008) 219–237.
- [7] T. Bouwmans, F. El Baf, B. Vachon, Statistical background modeling for foreground detection: a survey, *Handbook of Pattern Recognition and Computer Vision*, vol. 4, World Scientific Publishing, 2010, pp. 181–199, Ch. 3.
- [8] S. Brutzer, B. Höferlin, G. Heidemann, Evaluation of background subtraction techniques for video surveillance, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1937–1944.
- [9] S. Calderara, R. Melli, A. Prati, R. Cucchiara, Reliable background suppression for complex scenes, in: ACM International Workshop on Video Surveillance and Sensor Networks, 2006.
- [10] Y.-T. Chen, C.-S. Chen, C.-R. Huang, Y.-P. Hung, Efficient hierarchical method for background subtraction, *Pattern Recogn.* 40 (10) (2007) 2706–2715.
- [11] G. Choquet, Theory of capacities, *Ann. Inst. Fourier (1953)* 131–295.
- [12] D. Culibrk, O. Marques, D. Socek, H. Kalva, B. Furht, Neural network approach to background modeling for video object segmentation, *IEEE Trans. Neural Networks* 18 (6) (2007) 1614–1627.
- [13] Y. Dhome, N. Tronson, A. Vacavant, T. Chateau, C. Gabard, Y. Goyat, D. Gruyer, A benchmark for background subtraction algorithms in monocular vision: a comparative study, in: IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA), 2010, pp. 66–71.
- [14] F. El Baf, T. Bouwmans, B. Vachon, Foreground detection using the choquet integral, in: International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 2008, pp. 187–190.
- [15] F. El Baf, T. Bouwmans, B. Vachon, Fuzzy integral for moving object detection, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2008, pp. 1729–1736.
- [16] F. El Baf, T. Bouwmans, B. Vachon, Type-2 fuzzy mixture of Gaussians model: Application to background modeling, in: International Symposium on Advances in Visual Computing (ISVC), 2008.
- [17] F. El Baf, T. Bouwmans, B. Vachon, Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009, pp. 60–65.
- [18] A.B. Godbehere, A. Matsukawa, K. Goldberg, Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation, in: American Control Conference (ACC), 2012.
- [19] Y. Goyat, T. Chateau, L. Malaterre, L. Trassoudaine, Vehicle trajectories evaluation by static video sensors, in: IEEE International Conference on Intelligent Transportation Systems, 2006.
- [20] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, P. Ishwar, Changelog.net: a new change detection benchmark dataset, in: Workshop on Change Detection (CDW) at the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1–8.
- [21] E. Hayman, J.-O. Eklundh, Statistical background subtraction for a mobile observer, in: IEEE International Conference on Computer Vision (ICCV), 2003, pp. 67–74.
- [22] M. Hofmann, P. Tiefenbacher, G. Rigoll, Background segmentation with feedback: The pixel-based adaptive segmenter, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, pp. 38–43.
- [23] V. Jain, B.B. Kimia, J.L. Mundy, Background modeling based on subpixel edges, in: IEEE International Conference on Image Processing (ICIP), 2007.
- [24] X. Jian, D. Xiao-Qing, W. Sheng-Jin, W. You-Shou, Background subtraction based on a combination of texture, color and intensity, in: International Conference on Signal Processing (ICSP), 2008.
- [25] P. Kaewtrakulpong, R. Bowden, An improved adaptive background mixture model for realtime tracking with shadow detection, *European Workshop on Advanced Video Based Surveillance Systems (AVSS)*, 2001.
- [26] F. Kristensen, P. Nilsson, V. Owall, Background segmentation beyond RGB, in: Asian Conference on Computer Vision (ACCV), 2006, pp. 602–612.
- [27] A. Lai, N. Yung, A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence, in: IEEE International Symposium on Circuits and Systems (ISCAS), 1998, pp. 241–244.
- [28] C. Lallier, E. Renaud, L. Robinault, L. Tougne, A testing framework for background subtraction algorithms comparison in intrusion detection context, in: IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS), 2011.
- [29] L. Maddalena, A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, *IEEE Trans. Image Process.* 17 (7) (2008) 1168–1177.
- [30] L. Maddalena, A. Petrosino, A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection, *Neural Comput. Appl.* 19 (2) (2010) 179–186.
- [31] N.J. B. McFarlane, C.P. Schofield, Segmentation and tracking of piglets in images, *Mach. Vis. Appl.* 8 (3) (1995) 187–193.
- [32] N.M. Oliver, B. Rosario, A.P. Pentland, A Bayesian computer vision system for modeling human interactions, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8) (2000) 831–843.
- [33] A. Prati, I. Mikic, M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (7) (2003) 918–923.
- [34] M. Sigari, N. Mozayani, H. Pourreza, Fuzzy running average and fuzzy background subtraction: concepts and application, *Int. J. Comput. Sci. Network Security* 8 (2) (2008) 138–143.
- [35] M. Sivabalakrishnan, D. Manjula, Adaptive background subtraction in dynamic environments using fuzzy logic, in: *International Journal of Image Processing*, 2010.
- [36] A. Sobral, BGSLibrary: an opencv c++ background subtraction library, in: IX Workshop de Viso Computacional (WVC'2013). Rio de Janeiro, Brazil, 2013, Software available at <<http://code.google.com/p/bgslibrary/>>.
- [37] A. Sobral, L. Oliveira, L. Schnitman, F. Souza, Highway traffic congestion classification using holistic properties, in: International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA'2013). Innsbruck, Austria, 2013, more information available at <<http://www.behance.net/andrewssobral>>.
- [38] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for a real-time tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 1999.
- [39] M. Sugeno, Theory of fuzzy integrals and its applications. Ph.D. thesis, Tokyo Institute of Technology, 1974.
- [40] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, in: International Conference on Computer Vision (ICCV), 1999, pp. 255–261.
- [41] O. Tuzel, F. Porikli, P. Meer, A Bayesian approach to background modeling, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
- [42] A. Vacavant, T. Chateau, A. Wilhelm, L. Lequière, A benchmark dataset for foreground/background extraction, in: *Background Models Challenge (BMC) at Asian Conference on Computer Vision (ACCV)*, LNCS, vol. 7728, Springer, 2012, pp. 291–300.
- [43] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [44] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfunder: real-time tracking of the human body, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (7) (1997) 780–785.

- [45] J. Yao, J. marc Odobez, Multi-layer background subtraction based on color and texture, in: *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 2007.
- [46] H. Zhang, D. Xu, Fusing color and texture features for back ground model, in: *International Conference on Fuzzy Systems and Knowledge Discovery*, 2006.
- [47] Z. Zhao, T. Bouwmans, X. Zhang, Y. Fang, A fuzzy background modeling approach for motion detection in dynamic backgrounds, in: *International Conference Communications in Computer and Information Science*, vol. 346, 2012, pp. 177–185.
- [48] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, in: *IEEE International Conference on Pattern Recognition (ICPR)*, 2004.
- [49] Z. Zivkovic, F.V. D. Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, *Pattern Recog. Lett.* 27 (7) (2006) 773–780.