Pattern Recognition Letters
journal homepage: www.elsevier.com

LaBGen: A method based on motion detection for generating the background of a scene

Benjamin Laugraud[a,**], Sébastien Piérard[a], Marc Van Droogenbroeck[a]

[a] *University of Liège, INTELSIG Laboratory, Montefiore Institute, Quartier Polytech 1, Allée de la Découverte 10, 4000 Liège, Belgium*

ABSTRACT

Given a video sequence acquired with a fixed camera, the generation of the stationary background of the scene is a challenging problem which aims at computing a reference image for a motionless background. For that purpose, we developed our method named LaBGen, which emerged as the best one during the Scene Background Modeling and Initialization (SBMI) workshop organized in 2015, and the IEEE Scene Background Modeling Contest (SBMC) organized in 2016. LaBGen combines a pixel-wise temporal median filter and a patch selection mechanism based on motion detection. To detect motion, a background subtraction algorithm decides, for each frame, which pixels belong to the background. In this paper, we describe the LaBGen method extensively, evaluate it on the SBI 2016 dataset and compare its performance with other background generation methods. We also study its computational complexity, the performance sensitivity with respect to its parameters, and the stability of the predicted background image over time with respect to the chosen background subtraction algorithm. We provide an open source C++ implementation at http://www.telecom.ulg.ac.be/labgen.

1. Introduction

Given a scene viewed from a fixed viewpoint, the problem of generating an image of the background is known as the *background generation problem*. The background is usually defined as the set of elements that are motionless or subject to periodic movements. An important particular case arises when the background is motionless for the duration of the whole sequence (*i.e.* the background is *stationary*). This paper focuses on this case, in which the background image is unique (see Fig. 1). The background generation problem has many applications, including the rendering of non-occluded views of monuments or landscapes that are difficult to capture in crowded places. The *background subtraction* problem is another application, well known in computer vision, that can benefit from the background generation (Cristani et al., 2010).

Estimating the stationary background image is challenging, especially when the background is occluded by moving objects during long time periods. For example, a pixel-wise temporal median filter (denoted as *median method* for convenience hereafter) will not be able to produce the expected result when the background is visible for less than half of the time. Therefore, more sophisticated methods have been proposed in the literature (see the review of Maddalena and Petrosino, 2014). To help clarifying the efficacy of the background generation methods, the SBMI workshop was organized by Maddalena and Bouwmans in 2015, and the first complete benchmarking framework was provided. The original framework comprises a dataset, known as the Scene Background Initialization (SBI) dataset, with 7 video sequences and the corresponding ground truths, an evaluation methodology, and a library to compute a set of dedicated metrics.

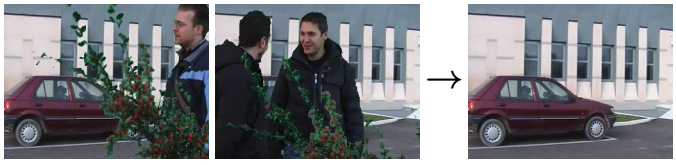


Fig. 1. The generation of a stationary background image is a challenging task, especially when the background is never fully visible. The right image is an example of a stationary background reference image produced from a series of images (such as the ones on the left) by the LaBGen method.

Among the stationary background generation methods submitted to the SBMI workshop, our method (see Laugraud et al., 2015b), named LaBGen, obtained the best performance;

**Corresponding author: Tel.: +32 (0)4 366 26 94;
e-mail: blaugraud@ulg.ac.be (Benjamin Laugraud)

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

LaBGen was also ranked as first during the IEEE Scene Background Modeling Contest (SBMC) organized in 2016. It combines the principles of the median method with a mechanism to select patches based on motion detection. More specifically, the quantity of motion is estimated locally in each frame in order to keep the patches that contains the lowest quantities of motion.

In this paper, we present the LaBGen method extensively, evaluate it on the SBI 2016 dataset (which is an extension of the original one), and compare its performance to other background generation methods. We also study its computational complexity and the performance sensitivity with respect to its parameters. First, we provide an extended description of the method in Section 2. Section 3 describes the experimental setup. Section 4 presents the performance of the method with respect to its parameters. As optimizing the parameters for a given sequence is time consuming, we establish two sets of parameter values suitable for a wide range of applications. In Section 5, we analyze the questions related to the choice of an appropriate background subtraction algorithm, and the stability of the computed background image over time. Section 6 concludes the paper.

2. Description of the LaBGen method

The LaBGen method was first introduced in Laugraud et al. (2015b). It comprises five steps:

1. To cope with difficulties due to short video sequences, the length of the input sequence is increased. The number of passes $\mathcal{P}$ is the parameter that controls the length of the augmented video sequence.
2. A motion detection is performed, frame per frame, to determine which pixels belong to the background. The parameter $\mathcal{A}$ identifies the used motion detection algorithm.
3. Based on the motion detection, the quantity of motion is estimated locally inside of spatial areas whose size is dependent on the parameter $\mathcal{N}$.
4. The quantity of motion is then used to select locally the subset Ω of patches with the least motion. The amount of selected patches is given by the parameter $\mathcal{S}$.
5. The stationary background image B is generated by applying the median method on the subsets of selected patches.

In the following sections, we detail these five steps.

2.1. Step 1: Increasing the length of a video sequence

The motion detection in the second step can be performed by a background subtraction algorithm. To detect motion, most background subtraction algorithms need a long series of images to be initialized and trained. During this period, whose optimal duration is driven by the video content, the motion detection is unreliable (*e.g.* presence of *ghosts*, as illustrated in Fig. 2). Because of the presence of ghosts, bootstrapping, and other effects, the LaBGen background generation method, relying on algorithms of background subtraction for motion detection, would be unsuccessful for short video sequences. Therefore, it was suggested in Laugraud et al. (2015b) to increase the length of any input video sequence artificially by performing $\mathcal{P}$ passes. An odd pass processes the input sequence in the forwards (chronological) order, while an even pass processes it in the backwards (reverse) order to ensure smooth transitions between the end of a pass and the beginning of a new one. In general, considering the input frames in a non-chronological order does not necessarily decrease the performance of background subtraction, and in some cases, it can improve the performance with respect to the forwards order, as shown in Laugraud et al. (2015a). In the current implementation, $\mathcal{P}$ is always chosen to be odd to finish with a forwards pass.

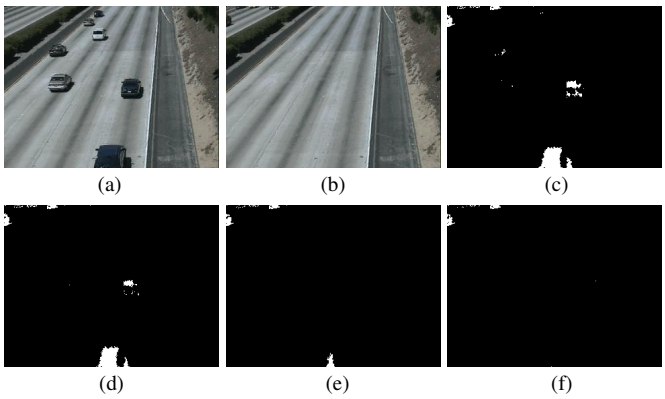


Fig. 2. Running through the video sequence several times helps in removing ghosts with some background subtraction algorithms. In this example, the ViBe algorithm has been initialized with frame 1 of the HighwayII sequence (a). Image (b) is the 446$^{\text{th}}$ image of the sequence. Images (c), (d), (e), and (f) are the segmentation maps respectively returned during pass 1, 2, 3, and 5. As can be seen, several passes help to remove ghosts.

Let T be the initial number of frames in the original sequence. The length of the augmented video sequence is given by $T' = \mathcal{P}(T-1)+1$. If the t^{th} frame is denoted by F^t, the video sequence augmentation step lengthens the sequence in the following way:

$$F^1 \cdots F^T \mapsto F^1 \cdots F^T \left(F^{T-1} \cdots F^2 F^1 \cdots F^T\right)^{\frac{\mathcal{P}-1}{2}}. \qquad (1)$$

Note that the first frame of the augmented video is not processed as no motion can be detected in this frame. However, it is used to initialize the motion detection algorithms. For convenience, the frames of the augmented video sequence are renumbered $F^1 \cdots F^{T'}$, with T' being the index of the last frame.

2.2. Step 2: Detecting motion

Different approaches exist to detect motion in a video sequence, such as optical flow or background subtraction. The computational cost of the former makes it unattractive; that is why we prefer to use background subtraction algorithms to detect motion. These algorithms decide, for each pixel $p^t_{x,y}$ of a given frame F^t (with (x,y) being the pixel coordinates, and $t \in [2, T']$), whether it belongs to the foreground or the background of the scene based on motion detection. Typically, they maintain a *model* of the background, updated according to the content of the current frame and an *updating strategy*. To perform a classification, a *segmentation strategy* is applied. For each pixel, the features extracted from the current frame are

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

compared to those stored in the model. This segmentation process results in a *segmentation map* m^t associated to the current input frame F^t with:

$$m^t_{x,y} = \begin{cases} 0 & \text{if } p^t_{x,y} \text{ is in the background,} \\ 1 & \text{if } p^t_{x,y} \text{ is in the foreground.} \end{cases} \quad (2)$$

Note that, while most background subtraction algorithms use an internal representation for the background, LaBGen does not require any access to the model. Exploiting the internal model to derive a spatially coherent background image appeared to be complicated for some types of models. Moreover, the background generation method does not require the knowledge of the difference value before the thresholding operation. In other words, in the LaBGen method, the background subtraction algorithms (or any other technique that could be used to detect motion) are interchangeable for producing the required binary segmentation map for each input frame. Consequently, any background subtraction algorithm can be used, regardless of its model, updating strategy, and segmentation strategy. In the following, we use the parameter $\mathcal{A}$ to refer to the chosen background subtraction algorithm.

2.3. Step 3: Estimating locally the quantity of motion

To motivate the need for an estimation of the quantity of motion, let us underline that applying a pixel-wise median filter on all the values classified in the background to generate a stationary background image would not work when the background is rarely observable. Indeed, most background subtraction algorithms are optimized to have a low false detection rate, even if that leads to a high miss rate. Therefore, they do not guarantee that at least half of the values classified in the background belong to the background. In technical terms, there is no guarantee that the *negative predictive value* is larger than 0.5.

In order to adapt the inadequate behavior of background subtraction algorithms for our needs, we analyze the predictions made by an algorithm in a spatial area $f_i \subseteq F$ (with i being the index of the considered area), instead of considering a single pixel. More precisely, we compute q^t_i, the proportion of pixels classified in the foreground by the background subtraction algorithm in a given area f^t_i of a frame F^t. This measure represents the quantity of motion in the considered area, and it estimates the probability of observing pixels corresponding to moving objects in that area. In our implementation, we arbitrarily divide the width w and height h of the image plane by the parameter $\mathcal{N} \in \mathbb{N}^+$ to create $\mathcal{N} \times \mathcal{N}$ non-overlapping patches, denoted by $f_i \subseteq F$ (with $i \in \left[1, \mathcal{N}^2\right]$ and $\bigcup_i f_i = F$), and whose width/height ratio is approximately identical to the w/h ratio of the original image. Then, we measure the quantity of motion in each patch of each frame as follows:

$$q^t_i = \sum_{(x,y)\in\Psi_i} \frac{m^t_{x,y}}{\#\Psi_i} \quad \text{with } \Psi_i = \left\{(x,y) \,\middle|\, p_{x,y} \in f_i\right\}. \quad (3)$$

2.4. Step 4: Selecting subsets of patches

During the *selection step*, for every spatial area f_i in the image plane, we build a subset Ω_i of maximum $\mathcal{S} \in \mathbb{N}^+$ patches, $\mathcal{S}$ being a parameter of the method. The patches in the subset Ω_i are selected among the set of candidate patches $\left\{f^2_i \cdots f^{T'}_i\right\}$ according to the quantities of motion $q^2_i \cdots q^{T'}_i$. It should be noted that selecting a restricted number of patches is essential for two reasons: (1) it bounds the memory usage, and (2) by keeping $\mathcal{S}$ patches, we avoid the problems related to a bias in determining q^t_i and finding an adequate threshold.

The selection mechanism builds the subset Ω_i iteratively, Ω^t_i being the subset of patches that are selected after processing t frames. We define the initial subset $\Omega^1_i = \emptyset$. While $t \leq \mathcal{S} + 1$, the patches f_i of the first processed frames (starting from the second one) are all selected:

$$\Omega^t_i = \Omega^{t-1}_i \cup \left\{f^t_i\right\} \quad \text{for } t \in [2, \mathcal{S}+1]\,. \quad (4)$$

After that, when $t > \mathcal{S}+1$, we incorporate the candidate patch f^t_i into the subset Ω^t_i if a *selection criterion* is satisfied. Note that, to keep constant the cardinality of Ω^t_i, a patch is removed when a new one is selected. The selection criterion checks whether the quantity of motion q^t_i associated to the candidate patch f^t_i is less or equal to at least one quantity of motion q^α_i associated to a patch $f^\alpha_i \in \Omega^{t-1}_i$. When it is satisfied, the patch $f^\beta_i \in \Omega^{t-1}_i$ with the highest quantity of motion q^β_i is removed from Ω^t_i. If two or more patches in Ω^{t-1}_i share this quantity, we always remove the oldest patch. In summary, we build the subsets Ω^t_i as follows:

$$\Omega^t_i = \begin{cases} \Omega^{t-1}_i \cup \left\{f^t_i\right\} \setminus \left\{f^\beta_i\right\} & \text{if } t > \mathcal{S}+1 \;\wedge\; q^t_i \leq q^\beta_i, \\ \Omega^{t-1}_i & \text{otherwise.} \end{cases} \quad (5)$$

Note that the final set of selected patches Ω_i is the subset of patches $\Omega^{T'}_i$ constructed for frame $F^{T'}$. Thus, $\Omega_i = \Omega^{T'}_i$.

2.5. Step 5: Generating the background image

With our method, an estimation of the stationary background B^t can be generated after the processing of each frame F^t. Once the sets of selected patches Ω^t_i have been built for every spatial area f_i in the image plane, B^t is constructed by applying a pixel-wise median filter on the selected patches:

$$B^t_{x,y} = \text{median}\left(\left\{p^{t'}_{x,y} \,\middle|\, p_{x,y} \in f_i \wedge f^{t'}_i \in \Omega^t_i\right\}\right). \quad (6)$$

In particular, the final background image B corresponds to the stationary background generated after the processing of frame $F^{T'}$. In other words, $B = B^{T'}$. For multi-channel images, we arbitrarily compute the median on each channel independently.

2.6. Implementation of the method

Fig. 3 provides an implementation of the LaBGen method in pseudo code. In this code, the sets of selected patches Ω_i are lists of selected patches f^t_i, stored with their associated quantity of motion q^t_i, and sorted by ascending order of quantities of motion. This helps to insert and remove selected patches efficiently. Furthermore, the stationary background is generated only once, when the selection step has been performed on the final frame $F^{T'}$, although we could output a background image for each time index. We provide an open source C++ implementation at http://www.telecom.ulg.ac.be/labgen.

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

```
for pass p ← 1 to 𝒫 do
    while frames are available do
        if pass p % 2 = 1 then
            F ← get next frame in forwards order;
        else
            F ← get next frame in backwards order;
        if F is the first processed frame then
            initialize model of algorithm 𝒜 with frame F;
            continue;
        m ← segmentation map of frame F returned by 𝒜;
        update model of algorithm 𝒜 with frame F;
        foreach i ∈ [1, 𝒩²] do
            f_i ← patch in spatial area i of frame F;
            m_i ← patch in spatial area i of segmentation map m;
            q_i ← (# foreground classifications in m_i) / (# pixels in f_i);
            foreach pair (f'_i, q'_i) ∈ list of selected patches Ω_i do
                if quantity of motion q_i ≤ q'_i then
                    insert (f_i, q_i) into list Ω_i before (f'_i, q'_i);
                    if #Ω_i > 𝒮 then remove last pair of list Ω_i;
                    break;
            if #Ω_i < 𝒮 and pair (f_i, q_i) ∉ list Ω_i then
                insert (f_i, q_i) at the end of list Ω_i;
foreach list of selected patches Ω_i do
    apply pixel-wise median filter on selected patches f'_i ∈ Ω_i;
```

Fig. 3. An implementation of the LaBGen method in pseudo code.

2.7. Classification of the method

According to the taxonomy introduced in the survey by Maddalena and Petrosino (2014), the method is:

- *Hybrid* as we combine classifications made at the pixel-level by a background subtraction algorithm with a region-level selection process extracting patches with the lowest quantities of motion.
- *Non-recursive* as our method stores patches observed in the previous frames in sets during one or more passes, and directly derives the estimated background image by the application of a pixel-wise median filter.
- *Selective* as the pixel-wise median filter is applied on sets of selected patches.

3. Experimental setup

In this section, we present the experimental setup for our experiments. It consists in the SBI 2016 dataset presented in Section 3.1, and metrics used for the evaluation presented in Section 3.2. We also discuss the choice of 13 different background subtraction algorithms in Section 3.3.

3.1. The SBI 2016 dataset

The 2016 version of the SBI dataset (`http://sbmi2015.na.icar.cnr.it/SBIdataset.html`), provided by Maddalena and Petrosino (2015), comprises 14 sequences listed in Table 1. They originate from various other datasets and are provided with a ground truth image considered as the perfect stationary background estimation. Matlab code is also provided to compute the six metrics presented in Section 3.2.

Table 1. Description of the sequences of the SBI 2016 dataset. Each sequence (1st column) is provided with the resolution (2nd column), the number of frames (3rd column), the number of frames Without Foreground objects (WF; 4th column), the Mean Quantity of Motion (MQM; 5th column) estimated from the segmentation maps returned by the oracle (see Section 3.3), and the ability of the median method to provide a background estimation that is acceptable visually (6th column).

Sequence	Resolution	Frames	WF	MQM	Median
Board	200 × 164	228	0	0.341	✘
Candela_m1.10	352 × 288	350	0	0.038	✘
CAVIAR1	384 × 256	610	0	0.055	✘
CAVIAR2	384 × 256	460	0	0.009	✔
CaVignal	200 × 136	258	0	0.103	✘
Foliage	200 × 144	394	5	0.507	✘
Hall&Monitor	352 × 240	296	0	0.039	✘
HighwayI	320 × 240	440	0	0.167	✔
HighwayII	320 × 240	500	0	0.052	✔
HumanBody2	320 × 240	740	0	0.107	✔
IBMtest2	320 × 240	90	0	0.068	✔
People&Foliage	320 × 240	341	0	0.421	✘
Snellen	144 × 144	321	1	0.585	✘
Toscana	800 × 600	6	0	0.218	✘

3.2. Metrics to assess the results

Maddalena and Petrosino (2015) suggest to use six metrics to evaluate background generation methods. The metrics to minimize (resp. maximize) are indicated by a ↓ (resp. ↑) symbol:

1. *Average Gray-level Error (AGE, ↓, [0, 255]):* average of the absolute difference between the gray-scale values of an input image and a ground truth image.
2. *Percentage of Error Pixels (pEPs, ↓):* considers that a difference of gray-scale values larger than 20 is an error.
3. *Percentage of Clustered Error Pixels (pCEPs, ↓):* if a gray-scale value and its 4-connected neighbors are considered as errors according to pEPs, then the corresponding pixel is erratic.
4. *Peak-Signal-to-Noise-Ratio (PSNR, ↑):* defined by Eq. 7, with MSE being the Mean Squared Error between an input gray-scale image and a ground truth gray-scale image:

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}}\, dB. \tag{7}$$

5. *Multi-Scale Structural Similarity Index (MS-SSIM, ↑, [−1, 1]):* metric defined by Wang et al. (2003) using structural distortion as an estimation of the perceived visual distortion.
6. *Color image Quality Measure (CQM, ↑, dB):* metric defined by Yalman and Ertürk (2013) combining per-channel PSNRs computed on an approximated reversible RGB to YUV transformation.

Except for CQM, all the metrics are designed for monochromatic images. In practice, colors images are only evaluated on the luminance component Y. In this paper, we arbitrarily chose pEPs as our reference metric.

3.3. Background subtraction algorithms

In our experiments, we tested 13 background subtraction algorithms. As a huge number of algorithms are available these

days (Bouwmans, 2014), we decided to test a large panel of different models. The simplest algorithm, commonly called as frame difference (F. Diff.), thresholds the distance of colors between consecutive frames. As the noise is not temporally uniform, some algorithms use a probabilistic model estimating a statistical distribution of the background colors adapted over time. Thus, by supposing a Gaussian noise, the Pfinder (Wren et al., 1997) modeled the background with a Gaussian distribution whose mean and variance are adapted over time. Instead of estimating the mean, the $\Sigma - \Delta$ algorithm (Manzanera and Richefeu, 2004) approximates the median using a behavior close to $\Sigma - \Delta$ estimators. Stauffer and Grimson (1999) extended the idea of using a Gaussian distribution to exclude dynamic backgrounds from the foreground. By using a mixture of Gaussians, several modes can be combined and more than one background can be handled (MoG G.). Zivkovic (2004) improved this extension by adapting the number of distributions needed over time (MoG Z.). Goyat et al. (2006) presented VuMeter, which is a non parametric method estimating a probability mass function of the background colors from a temporal color histogram. A different approach, described by Heikkilä and Pietikäinen (2006), leverages the idea of Stauffer et al. by using a mixture of LBP textures.

As an alternative, sample-based algorithms build their model by collecting features sampled over time. The KDE algorithm (Elgammal et al., 2000) collects a set of past color samples and estimates a non parametric statistical distribution by applying Parzen windows with a Gaussian kernel. The ViBe algorithm (Barnich and Van Droogenbroeck, 2011) uses a pure sample-based approach and random policies to sample observed background colors. Moreover, it incorporates a spatial propagation mechanism to ensure spatial consistency. Among others, some variants of ViBe have been developed by Hofmann et al. (2012) with PBAS which adapts the decision thresholds and update rates using controllers, and by St-Charles et al. (2015) with SuBSENSE (SuBS.) by associating the addition of adaptive parameters with the sampling of LBSP strings.

A different approach, based on self organization through artificial neural networks, has been proposed by Maddalena and Petrosino (2008). Their algorithm, named SOBS, builds a neural map that associates each pixel to several HSV weight vectors. Its updating strategy is based on a spatio-temporal selective weighted running average.

For completeness, we designed an *oracle* algorithm leading to a perfect segmentation, according to the pEPs metric, by classifying a pixel in the foreground when the difference of luminance with the ground truth is larger than 20.

Regarding the implementations, the ViBe algorithm has been provided by its authors. The implementations of the other algorithms are in the BGSLibrary maintained by Sobral (2013) and have been used with their default set of parameters. We wrote the code of the oracle algorithm.

4. Performance evaluation

In order to assess the LaBGen method for a large set of parameters, an estimation of the background has been generated for each sequence of the SBI 2016 dataset using each combination of $\mathcal{P} = \{1, 3, 5, ..., 29\}$, $\mathcal{N} = \{1, 2, 3, ..., 50\}$, $\mathcal{S} = \{1, 3, 5, ..., 201\}$, and the background subtraction algorithms $\mathcal{A}$ presented Section 3.3. Remember that, by construction, $\mathcal{P}$ is always an odd positive integer. Moreover, $\mathcal{S}$ is also chosen to be odd to avoid interpolated values in the median filter. The four parameters of our method ($\mathcal{P}$, $\mathcal{N}$, $\mathcal{S}$, $\mathcal{A}$) provide enough flexibility to obtain an almost perfect background estimation for any input video sequence. Therefore, in Section 4.1, we discuss the achievable performance when we choose a particular background subtraction algorithm $\mathcal{A}$ and optimize the $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ parameters. Then, we discuss the optimization of all four parameters $(\mathcal{P}, \mathcal{N}, \mathcal{S}, \mathcal{A})$ on a per sequence basis in Section 4.2. In Section 4.3, we compare the performances of LaBGen with that of other background generation methods. We analyze the computational complexity in Section 4.4 and provide a sensitivity analysis with respect to the parameters in Section 4.5. Note that LaBGen has been assessed on the SBMnet category-driven dataset (`http://www.scenebackgroundmodeling.net`) in Laugraud et al. (2016).

4.1. Determination of the best average performance

To evaluate our method, we first tuned the parameters to achieve the best average performance. Table 2 provides the best set of $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ parameters, given a background subtraction algorithm $\mathcal{A}$. These sets of parameters have been found by minimizing the pEPs score averaged over the sequences. Except for the oracle, all the metrics agree to rank the simple frame difference background subtraction algorithm as first. Moreover, it should be noted that, except for the MS-SSIM of some methods, all the results have been improved compared to the ones reported in Laugraud et al. (2015b) for the SBMI workshop. The reason is that a larger parameter space has been explored on an increased number of sequences.

If the results of the oracle are attractive, they are not achievable in practice as it uses the ground truth. Thus, to achieve the best average performance, we advise to use the frame difference algorithm with the following *default set of parameters*:

$$(\mathcal{P}, \mathcal{N}, \mathcal{S}, \mathcal{A})_{\text{default}} = (29, 4, 57, \text{F. Diff.}) . \quad (8)$$

Even though the use of the frame difference algorithm is surprising, we believe that it is particularly efficient in our context due to its total insensibility to its initialization (*i.e.* its insensibility to *bootstrap*). Because it returns the same motion information for any couple of frames at each pass, doubling both $\mathcal{S}$ and $\mathcal{P}$ keeps the median of the selected patches unchanged. Therefore, for a given $\mathcal{N}$, the ratio $\mathcal{S}/\mathcal{P}$ is the main factor influencing the performance when the frame difference algorithm is used. Noting that $\mathcal{S}$ and $\mathcal{P}$ are two odd positive integers, we might need to take large values of $\mathcal{S}$ and $\mathcal{P}$ to reach the optimal $\mathcal{S}/\mathcal{P}$ ratio (especially if the optimal ratio is close to an even integer). This explains why we obtained $\mathcal{P} \neq 1$ in the default set of parameters.

For experimenting with other background subtraction algorithms (*e.g.* to compare the contribution of an untested background subtraction algorithm with the tested ones), we suggest the following *universal set of parameters:*

$$(\mathcal{P}, \mathcal{N}, \mathcal{S})_{\text{universal}} = (1, 2, 19) , \quad (9)$$

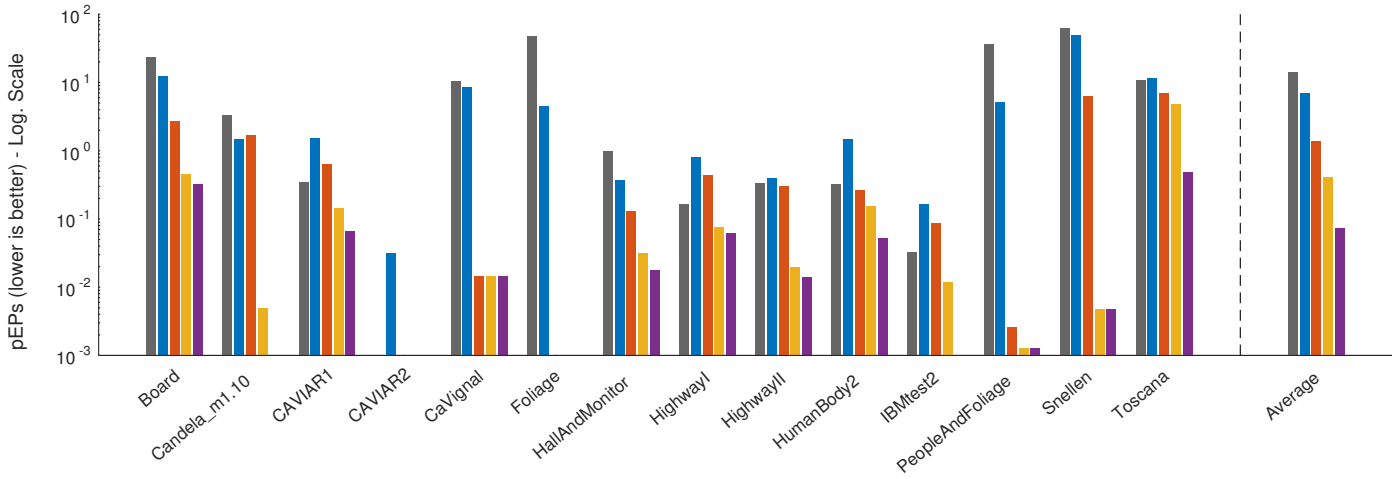


Fig. 4. Performance boost, in terms of averaged pEPs on the 14 SBI 2016 sequences, due to various parameters optimization steps of our method. We consider the simple median method as the baseline ■. The performance of LaBGen with an unspecified background subtraction algorithm and the universal set of parameters (see Eq. 9) is better on average ■. An improvement is obtained with a suitable background subtraction algorithm (here, the frame difference with the default set of parameters; see Eq. 8) ■. A further improvement is obtained by keeping the frame difference and tuning the parameters for each sequence (see Table 3) ■. The best performance is reached when all the parameters are tuned for each sequence (see Table 3) ■.

Table 2. Mean performance on the SBI 2016 dataset with respect to each background subtraction algorithm $\mathcal{A}$ by using the universal set of parameters (left table; see Eq. 9), and parameters tuned for each algorithm $\mathcal{A}$ (right table). The best sets of $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ parameters in the right table have been obtained by minimizing the pEPs score averaged over sequences. The algorithms $\mathcal{A}$ are ranked according to the pEPs metric. Note that the best performance appears in the right table.

Universal set of parameters		
$\mathcal{A}$	Rank	pEPs ↓
Oracle	-	1.4650%
F. Diff.	**1**	**2.6678%**
MoG G.	2	3.6599%
VuMeter	3	4.7470%
KDE	4	6.1272%
PBAS	5	6.7072%
LBP	6	7.1875%
MoG Z.	7	7.2937%
Pfinder	8	7.5703%
Σ-Δ	9	7.5751%
SuBS.	10	8.2697%
ViBe	11	8.7449%
SOBS	12	12.7843%
Median method		14.0500%

		Best parameters			Averaged metrics					
$\mathcal{A}$	Rank	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{S}$	AGE ↓	pEPs ↓	pCEPs ↓	PSNR ↑	MS-SSIM ↑	CQM ↑
Oracle	-	17	49	15	2.3200	0.0148%	0.0006%	38.7113	0.9918	49.7474
F. Diff.	**1**	**29**	**4**	**57**	**2.9945**	**1.3972%**	**0.9246%**	**35.2028**	**0.9764**	**47.2946**
MoG G.	2	17	2	41	4.1857	2.4494%	1.9742%	33.9778	0.9472	45.1503
SuBS.	3	17	3	3	4.1848	2.6674%	1.7270%	33.0554	0.9646	45.0498
PBAS	4	1	2	1	4.8934	3.4566%	2.2524%	30.3995	0.9448	41.0873
VuMeter	5	1	2	27	4.9965	3.9232%	3.1150%	31.5587	0.9362	44.6622
KDE	6	5	2	177	4.5624	4.3616%	3.5576%	32.3596	0.9519	45.5814
MoG Z.	7	29	50	7	5.8876	5.1848%	3.6553%	29.0150	0.9342	41.3826
LBP	8	1	1	39	4.9960	5.2596%	3.9821%	30.6226	0.9381	44.3876
Σ-Δ	9	3	2	7	6.6502	6.1998%	5.1822%	30.9743	0.9202	43.1996
ViBe	10	21	50	7	7.0353	6.2812%	4.0358%	27.8026	0.8996	40.2080
Pfinder	11	1	2	23	7.2657	7.3764%	6.0646%	30.7581	0.9219	43.4202
SOBS	12	7	1	19	8.6817	9.6740%	7.8359%	27.1229	0.8726	40.7849
Median method					10.5889	14.0500%	11.2883%	28.1082	0.8570	42.6332

Table 3. Best set of parameters per SBI 2016 sequence using the frame difference background subtraction algorithm (left table), and the best background subtraction algorithm $\mathcal{A}$ (right table). Each set of $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ and $(\mathcal{P}, \mathcal{N}, \mathcal{S}, \mathcal{A})$ parameters has been obtained by minimizing the pEPs score. Note that the best performance appears in the right table.

$\mathcal{A}$ = F. Diff.				
Sequence	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{S}$	pEPs ↓
Board	3	23	201	0.4543%
Candela_m1.10	1	10	3	0.0049%
CAVIAR1	1	2	199	0.1465%
CAVIAR2	1	13	5	0.0000%
CaVignal	1	2	1	0.0147%
Foliage	1	1	1	0.0000%
Hall&Monitor	1	1	11	0.0320%
HighwayI	3	1	85	0.0768%
HighwayII	1	1	15	0.0195%
HumanBody2	1	17	113	0.1523%
IBMtest2	3	2	185	0.0117%
People&Foliage	1	3	1	0.0013%
Snellen	1	1	1	0.0048%
Toscana	3	1	5	4.7644%

	Best parameters				Metrics					
Sequence	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{S}$	$\mathcal{A}$	AGE ↓	pEPs ↓	pCEPs ↓	PSNR ↑	MS-SSIM ↑	CQM ↑
Board	1	5	99	KDE	4.2793	0.3201%	0.0274%	32.6703	0.9443	51.5876
Candela_m1.10	1	4	1	Σ-Δ	1.5051	0.0000%	0.0000%	41.6544	0.9954	49.6929
CAVIAR1	3	17	149	KDE	3.3136	0.0661%	0.0102%	36.1502	0.9933	52.2087
CAVIAR2	1	2	1	Pfinder	1.4388	0.0000%	0.0000%	42.3633	0.9976	52.5509
CaVignal	1	2	1	F. Diff.	0.5749	0.0147%	0.0000%	44.7687	0.9978	55.1346
Foliage	1	1	1	F. Diff.	1.8351	0.0000%	0.0000%	39.2288	0.9975	45.5876
Hall&Monitor	1	15	111	SOBS	2.0775	0.0178%	0.0000%	38.8801	0.9937	47.1858
HighwayI	1	1	37	MoG G.	1.3731	0.0612%	0.0013%	41.0358	0.9928	59.7805
HighwayII	1	1	23	KDE	1.7793	0.0143%	0.0000%	39.8206	0.9960	48.4193
HumanBody2	23	5	51	PBAS	3.1293	0.0521%	0.0000%	36.2525	0.9975	47.9972
IBMtest2	1	3	5	MoG Z.	2.9613	0.0000%	0.0000%	35.3513	0.9934	48.7203
People&Foliage	1	3	1	F. Diff.	0.9638	0.0013%	0.0000%	43.5511	0.9983	48.4398
Snellen	1	1	1	F. Diff.	0.3776	0.0048%	0.0000%	49.0233	0.9995	55.8155
Toscana	3	13	7	SuBS.	1.3666	0.4850%	0.2281%	35.8936	0.9872	46.9299

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

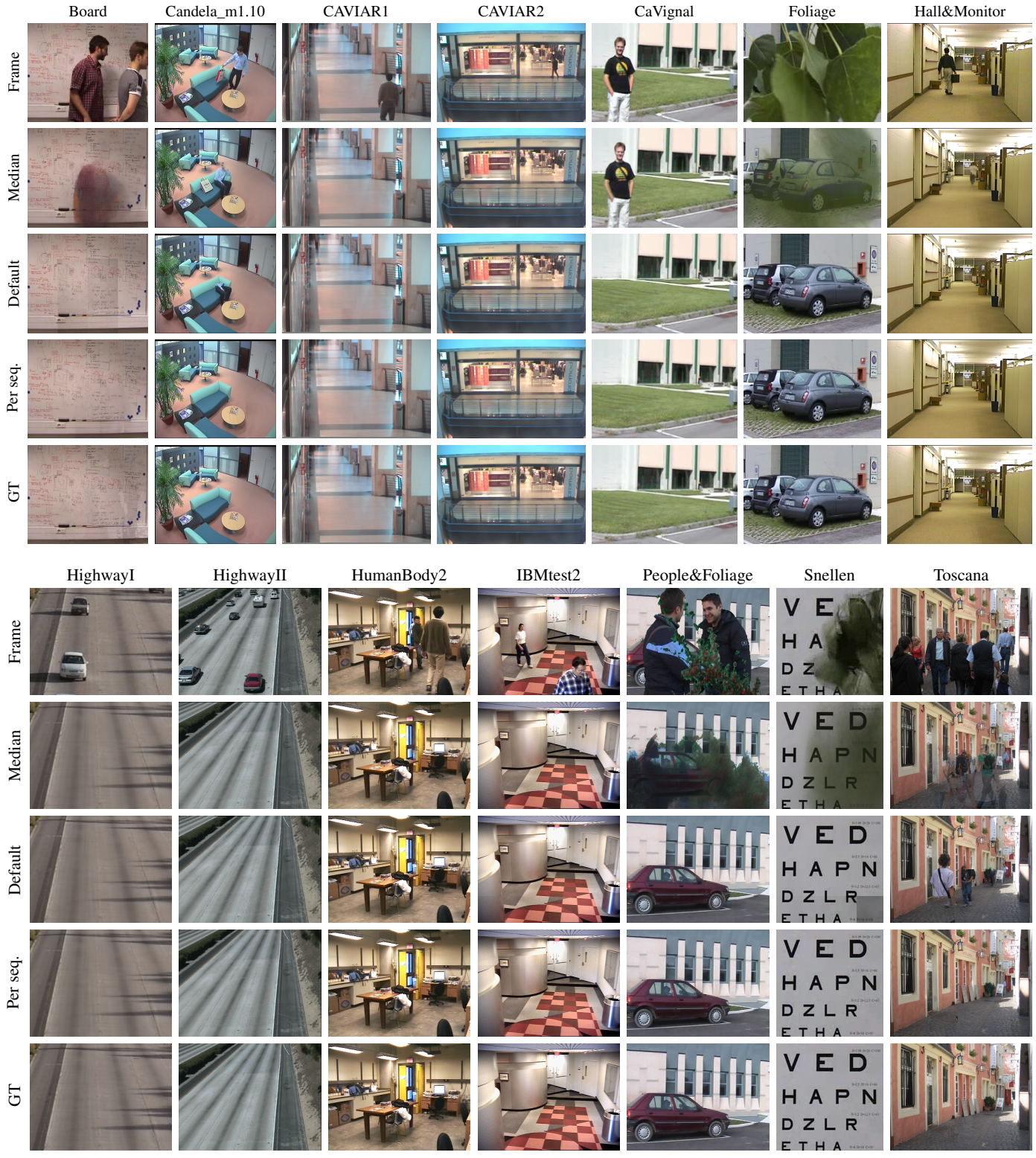


Fig. 5. Results obtained for the sequences of the SBI 2016 dataset (an arbitrary frame is shown on the 1st row) with the median method (2nd row), our method with the default (3rd row; see Eq. 8) and per sequence optimized (4th row; see Table 3) sets of parameters, and the ground truth (5th row).

that minimizes the average pEPs metric ($\simeq$ 6.52%) on the 14 sequences of the SBI 2016 dataset and with the 13 tested background subtraction algorithms. We compare the performance of the LaBGen method using the universal set of parameters and the default set of parameters in Fig. 4 and Table 2.

Fig. 5 shows the background images generated for the SBI 2016 dataset using the default set of parameters (see Eq. 8). Although several results are visually almost perfect (even for the highly cluttered sequences Foliage and People&Foliage), a few imperfections appear for some sequences. For exam-

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

ple, shadows occur in the Board and Snellen sequences, and some patches differ by being darker than other ones in the background images. This suggests that selecting patches without any temporal constraint or appropriate blending paradigm is problematic to handle shadows and illumination changes properly. The robustness to stationary foreground objects is challenged with the four sequences CaVignal, Hall&Monitor, Candela_m1.10, and CAVIAR1. Our method applied on CaVignal and Hall&Monitor produces results that are visually perfect. For the Candela_m1.10 and CAVIAR1 sequences, a silhouette is still partially visible in the estimated background. In fact, LaBGen used with the frame difference is not always able to remove foreground objects that remain static for a long period of time. Another background subtraction algorithm, able to keep such objects in the foreground, is needed to avoid their inclusion into the estimated background. For the HighwayI sequence, camera jitter creates some patch misalignments since our method lacks of a motion compensation mechanism. Finally, as the frame difference algorithm is unable to handle large changes, the LaBGen method fails to produce a usable background estimation for the very short Toscana sequence. Despite of these drawbacks, our method with the default set of parameters succeeds where more complicated methods, such as those assessed by Maddalena and Petrosino (2015), fail.

4.2. Optimizing the performance per sequence

Although the default set of parameters of Eq. 8 leads to a good performance in average (pEPs $\simeq 1.40\%$), we can produce almost perfect results on a per sequence basis.

Table 3 provides, given a video sequence of the SBI 2016 dataset, the best set of $(\mathcal{P}, \mathcal{N}, \mathcal{S}, \mathcal{A})$ parameters according to the pEPs metric. Except for the AGE, PSNR and CQM metrics evaluated on the Hall&Monitor sequence, all the metrics have been improved compared to the results reported in Laugraud et al. (2015b) for the SBMI workshop. Moreover, it is interesting to observe that the $(\mathcal{P}, \mathcal{N}, \mathcal{S}) = (1, 1, 1)$ set of parameters is the best suited for Foliage and Snellen, as these sequences are the only ones containing frames never occluded with foreground objects (see Table 1). Finally, as using the frame difference algorithm provides the best performance in average (see Section 4.1), one can compare the performance of the LaBGen method used with the sets of $(\mathcal{P}, \mathcal{N}, \mathcal{S}, \mathcal{A})$ parameters optimized per sequence, and the sets of $(\mathcal{P}, \mathcal{N}, \mathcal{S}, \text{F. Diff.})$ parameters optimized per sequence, in Fig. 4 and Table 3.

4.3. Comparison to other background generation methods

To complete the performance evaluation of our method, we compared LaBGen to 6 state-of-the-art background generation methods. Among these methods, Photomontage (Agarwala et al., 2004) combines a series of images with respect to an objective function optimized with graph-cuts; for the background generation problem, the objective is the maximum likelihood. RSL2011 (Reddy et al., 2011) estimates the background in a Markov Random Field framework, where the optimal labeling solution is computed iteratively. Instead of formulating an optimal labeling problem, the WS2006 (Wang and Suter, 2006) method looks for the most reliable sub interval of stable intensities for each pixel, and takes its mean as the background value. In comparison to SOBS (see Section 3.3), SC-SOBS (Maddalena and Petrosino, 2012) adds a spatially coherent updating strategy providing further robustness against false detections; the background of a given pixel is extracted by choosing the closest weight vector to the ground truth. BGWiS (De Gregorio and Giordano, 2015) uses a weightless neural network, called WiSARD^{rp}, based on a network of memory nodes composed of several cells. For each pixel, the stimulated cells are increased by a reward while the other ones are decreased by a punishment. The background is then reconstituted from the cells associated to the greatest values. Finally, Sobral et al. (2015) apply a matrix completion algorithm on areas that are in motion. According to these authors, the LRGeomCG matrix completion algorithm performs the best for their method.

Table 4 reports the pEPs scores achieved by LaBGen and the background generation methods described above. Note that we provide results for the subset of 7 sequences composing the SBI 2015 dataset as no data is available for the whole SBI 2016 dataset yet. One can observe that LaBGen used with sets of parameters optimized per sequence is ranked first, and above RSL2011, which was the main challenger of our method during the SBMI workshop. Moreover, LaBGen achieves the best scores for 5 sequences, with an equality with RSL2011 for CaVignal, and Photomontage for Foliage. Although SC-SOBS performs better on the HighwayI and HighwayII sequences, it should be noted that this method uses the ground truth as an input. Except for the SC-SOBS method, LaBGen is better for these two sequences too.

When it uses its default set of parameters, LaBGen is ranked third. Although it is ranked below RSL2011, LaBGen performs better for 4 sequences, equally for CaVignal, but it is penalized by its score on the Snellen sequence. Remarkably, LaBGen with the default set of parameters achieves the best score for 4 out the 7 sequences, compared to other state-of-the-art methods. In addition, only SC-SOBS is better than our method for the HighwayII sequence. Note that for the CaVignal and Foliage sequences, the default set of parameters suffices to reach the best achievable score.

4.4. Computational complexity

The computational complexity of the LaBGen method can be derived by considering its different steps:

1. The first frame is only used to initialize the background subtraction algorithm $\mathcal{A}$. The complexity of this operation is $O(C_i^{\mathcal{A}})$, with $C_i^{\mathcal{A}}$ being the initialization cost of $\mathcal{A}$.
2. The motion detection (see Eq. 2) is applied for each frame. Its complexity is $O(\mathcal{P}TC_s^{\mathcal{A}})$, with $\mathcal{P}$ being the number of passes, T the number of frames of the input sequence, and $C_s^{\mathcal{A}}$ the cost for algorithm $\mathcal{A}$ to segment one frame.
3. The local estimation of the quantity of motion (see Eq. 3) requires to work with each pixel of each frames. Therefore, its complexity is $O(\mathcal{P}Thw)$, with h (resp. w) being the height (resp. width) of the input sequence.
4. In the worst case, selecting subsets of patches requires $\mathcal{S}$ comparisons for each area of the encountered frames (see Eq. 5). Thus, the computational complexity of this step is $O(\mathcal{P}T\mathcal{S}\mathcal{N}^2)$.

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

Table 4. Comparison of the pEPs scores achieved by LaBGen and other background generation methods on the SBI 2015 dataset. The results of SC-SOBS, WS2006, RSL2011, and Photomontage are taken from Maddalena and Petrosino (2015), and the other ones from the corresponding papers. The methods are ranked according to the pEPs score averaged over the 7 sequences. LaBGen is first using sets of parameters optimized per sequence, and third with its default set of parameters.

		Rank	Average pEPs ↓	Per sequence pEPs ↓						
				Hall&Monitor	HighwayI	HighwayII	CaVignal	Foliage	People&Foliage	Snellen
Back. Gen. Method	LaBGen (per seq.)	**1**	**0.0163%**	**0.0178%**	0.0612%	0.0143%	**0.0147%**	**0.0000%**	**0.0013%**	**0.0048%**
	RSL2011	2	0.5753%	0.8321%	0.3477%	1.2448%	**0.0147%**	0.1493%	0.7969%	0.6414%
	LaBGen (default)	3	1.0320%	0.1302%	0.4362%	0.3034%	**0.0147%**	**0.0000%**	0.0026%	6.3368%
	WS2006	4	4.6885%	0.5563%	0.6849%	0.4883%	1.5000%	2.8507%	3.5716%	23.1674%
	Photomontage	5	6.5827%	0.3610%	0.4076%	0.5885%	11.2206%	**0.0000%**	0.0039%	33.4973%
	SC-SOBS	6	8.0175%	0.9801%	**0.0039%**	**0.0091%**	3.1949%	0.5556%	14.0234%	37.3553%
	BGWiS	7	14.1857%	0.5200%	0.1400%	0.4600%	0.0900%	10.6200%	32.8600%	54.6100%
	Median	8	22.5583%	0.9789%	0.1628%	0.3320%	10.4853%	47.7049%	36.0091%	62.2348%
	LRGeomCG	9	35.8343%	0.2200%	0.2500%	0.3500%	13.9300%	62.7600%	83.5300%	89.8000%

Table 5. Run times of LaBGen on five SBI 2016 sequences using the default set of parameters (see Eq. 8), and the same set but with a constrained parameter $\mathcal{P} = 1$, $\mathcal{N} = 1$, or $\mathcal{S} = 1$. The times have been averaged over 100 executions on an Intel Core i7-4790K of an implementation that can still be sped up. Note that $\mathcal{P}$ is the main parameter influencing the run time.

Sequence	Resolution	Frames	Time (in ms)			
			Default	Default $\mathcal{P} = 1$	Default $\mathcal{N} = 1$	Default $\mathcal{S} = 1$
Snellen	144×144	321	718	83	449	462
IBMtest2	320×240	90	624	222	613	366
HumanBody2	320×240	740	7836	530	3219	3258
CAVIAR1	384×256	610	19721	938	6988	3731
Toscana	800×600	6	1220	66	1142	135

5. In our implementation (see Section 2.6), the pixel-wise median filter (see Eq. 6) is applied only once, at the end of the process. Using a selection algorithm, the cost to find the median among $\mathcal{S}$ values is linear in average $O(\mathcal{S}hw)$.

Considering this decomposition, the overall complexity of LaBGen, in the worst case, can be expressed as:

$$O\left(C_i^{\mathcal{A}} + \mathcal{P}T\left(C_s^{\mathcal{A}} + hw + \mathcal{S}\mathcal{N}^2\right) + \mathcal{S}hw\right). \tag{10}$$

In practice, with an Intel Core i7-4790K, a mean pixel throughput of 239×10^6 pixels/s has been measured according to 100 executions on each SBI 2016 sequence using the default set of parameters. Table 5 provides examples of run times which are better than real-time for most sequences.

4.5. Sensitivity analysis

To have a better insight on the performance of the LaBGen method, we study its sensitivity according to pEPs, when the parameters vary around the best sets of $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ parameters tuned for each background subtraction algorithm $\mathcal{A}$ (see Table 2). This sensitivity is illustrated graphically in Fig. 6. We observe that the augmentation step described in Section 2.1, and the local estimation of the quantity of motion described in Section 2.3 are critical elements of LaBGen. Indeed, for most background subtraction algorithms, the best average performance is reached after several passes (when $\mathcal{P} > 1$), and when the quantities of motion are estimated inside large patches (when $\mathcal{N}$ is small) rather than for pixels individually (when $\mathcal{N}$ is very large). More specifically, the $\mathcal{N}$ value leading to the best performance in average is comprised between 1 and 4. The fact that, as long as $\mathcal{N}$ increases and the patch size reduces to a pixel, the oracle performs better, tends to prove that an imperfect motion detection needs a spatial compensation mechanism, and that LaBGen tends to an almost perfect performance if a flawless motion detection is used. Regarding the parameter $\mathcal{S}$, the stability of the performance can be significantly different from one background subtraction algorithm to another. For instance, the performance remains stable on large ranges using the frame difference, while a step away from the local optimum is severely penalized with other algorithms such as ViBe. Note that in addition to the algorithm being used, the parameter $\mathcal{S}$ is strongly dependent on the scene as $\mathcal{S} = 1$ implies that the full background can be observed, free of foreground objects, at least once for each patch. However, $\mathcal{S} > 1$ means that several patches must be combined in order to generate the background in the different spatial areas.

For some parameters, a trade-off between performance and computational cost is possible. For instance, using $\mathcal{A}$ = F. Diff., the number of passes can be dropped from $\mathcal{P} = 29$ to 5 with no significant performance loss ($\simeq 0.36\%$). This can be useful for embedded devices as, according to Section 4.4, a high number of passes is the main bottleneck of the run time. Trade-offs are also possible between the selected background subtraction algorithm and the $\mathcal{S}$ parameter. For example, using KDE, the number of selected patches can be dropped from $\mathcal{S} = 177$ to 95 with no significant performance loss ($\simeq 0.21\%$). However, a decrease of $\mathcal{S}$ is unacceptable using the SOBS algorithm. Such trade-offs seem impossible for the parameter $\mathcal{N}$ since the performance quickly drops for a non optimal value.

5. On the choice of a background subtraction algorithm

Selecting the best background subtraction algorithm to produce a valuable motion detection for the LaBGen method is a difficult task for different reasons. One problem is that, as shown in Fig. 7, there is no negative correlation between the F_1 performance of a background subtraction algorithm (which is the favored metric in the field of background subtraction) and the pEPs performance of LaBGen using this algorithm.

Another important characteristic to consider for the choice of a background subtraction is the resulting stability of the predicted background image over time. To study that stability for a given video sequence of the SBI 2016 dataset, we compute the pEPs score for all the predicted background images B^t after

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.

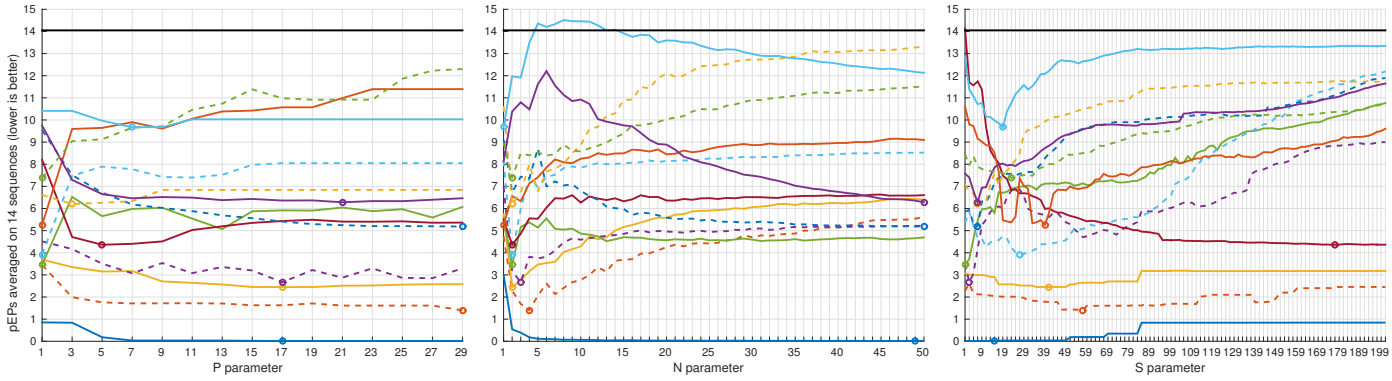


Fig. 6. Sensitivity of the performance, according to pEPs averaged over the 14 SBI 2016 sequences, when the parameters vary around the best sets of $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ parameters tuned for each background subtraction algorithm $\mathcal{A}$ = Oracle —, F. Diff. ---, MoG G. —, SuBS. ---, PBAS —, VuMeter ---, KDE —, MoG Z. ---, LBP —, Σ-Δ ---, ViBe —, Pfinder ---, SOBS — (see Table 2). The local optimum reached using a given algorithm $\mathcal{A}$ is indicated by a circle. Our baseline is the median method —.

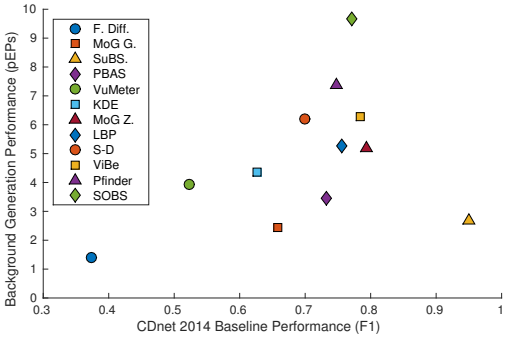


Fig. 7. Comparison between the performance of a background subtraction algorithm on the baseline of the CDnet 2014 dataset (Wang et al., 2014), and the mean performance of the LaBGen method using the best sets of $(\mathcal{P}, \mathcal{N}, \mathcal{S})$ parameters tuned for each algorithm $\mathcal{A}$ (values are taken from the right table of Table 2). As can be seen, there is no obvious relationship between the increase of the F_1 score and the decrease of the pEPs.

each number of frame comprises between 2 and T' (see Eq. 6). By doing so, we observed that some background subtraction algorithms require the presence of precise events in the video sequence to help in finding a correct background image (*e.g.* the observation of frames free of foreground objects). However, as soon as one new frame is processed, this estimation is directly degraded. Fig. 8a illustrates the evolution of the pEPs score for three background subtraction algorithms applied on the Board sequence. One observes that while LaBGen quickly converges towards a good and stable background estimation with the oracle and the frame difference, it never stabilizes its result with VuMeter. This observation raises several questions such as: is a result stable or related to a precise event appearing at the end of a sequence? If events are reorganized in a sequence, will the result be as good as the original one? Or simply, can we trust a result? To provide an insight on theses questions, we made a box plot illustrating the mean variation of the pEPs score over time, for each background subtraction algorithm presented in Section 3.3. As shown in Fig. 8b, it turns out that VuMeter, PBAS, and SuBSENSE tend to lead to unstable background images. Note that even though the stability of the frame difference, KDE, MoG Z., and ViBe algorithms is close to zero in average, there exist some outliers showing that these algorithms are unsuited for at least one sequence.

Finally, we observed in Table 3 that the best background subtraction to be used in our method depends on the considered video sequence. However, the precise relationship between the properties of the video sequences and the characteristics of the background subtraction algorithms is still to be understood (Piérard and Van Droogenbroeck, 2015). Relating to the question of the selection of the background subtraction algorithm, a blind approach that tests several background subtraction algorithms, is the only option in this work.

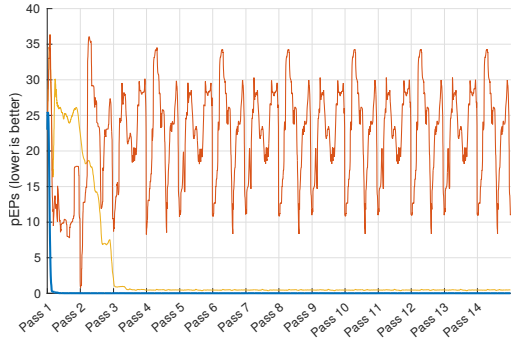
6. Conclusion

In this paper, we present the LaBGen stationary background generation method, first introduced in Laugraud et al. (2015b). This method is simple, flexible, fast, and it performs almost perfectly on the video sequences provided in the SBI 2016 dataset. Moreover, we provide an open source C++ implementation at `http://www.telecom.ulg.ac.be/labgen`.

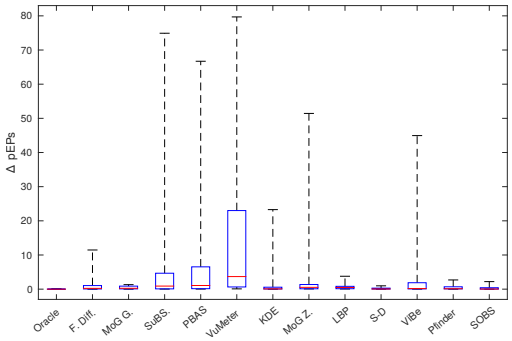
An essential component of LaBGen is its flexible motion detection mechanism, based on interchangeable background subtraction algorithms. Comparisons with 13 background subtraction algorithms show that, although it is counterintuitive, there is no obvious correlation between the known performance of an algorithm and the quality of the background estimation based on it. In particular, one of the most obvious background subtraction algorithm, namely the frame difference, provides a more valuable information for our method than many state-of-the-art algorithms. Another point to consider is that only some background subtraction algorithms guarantee a stable background image with respect to the length of the sequence or the number of passes.

While our results show the importance of tuning the selection of the background subtraction algorithm as well as the other parameters of our method, we suggested two sets of parameters: a default set of parameters, and a universal set of parameters when one wants to test the contribution of an untested background subtraction algorithm.

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.



(a) Evolution of the pEPs score obtained for the Board sequence when an estimation of the background is generated after each frame using the frame difference —, VuMeter —, and oracle — algorithms.



(b) Mean variation of the pEPs score over time. For each sequence of the SBI 2016 dataset, max (pEPs) − min (pEPs) has been computed for each pass (starting from the second one) and averaged.

Fig. 8. With some background subtraction algorithms, the generated background image might not converge while processing the frames of the augmented video sequence. These results were obtained with the parameters optimized per sequence (minimizing the pEPs score).

References

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M., 2004. Interactive digital photomontage. ACM Transactions on Graphics 23, 294–302.

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

Bouwmans, T., 2014. Traditional and recent approaches in background modeling for foreground detection: An overview. Computer Science Review 11-12, 31–66.

Cristani, M., Farenzena, M., Bloisi, D., Murino, V., 2010. Background subtraction for automated multisensor surveillance: A comprehensive review. EURASIP Journal on Advances in Signal Processing 2010, 24 pages.

De Gregorio, M., Giordano, M., 2015. Background modeling by weightless neural networks, in: International Conference on Image Analysis and Processing Workshops (ICIAP Workshops), pp. 493–501.

Elgammal, A., Harwood, D., Davis, L., 2000. Non-parametric model for background subtraction, in: European Conference on Computer Vision (ECCV), Springer, Dublin, Ireland. pp. 751–767.

Goyat, Y., Chateau, T., Malaterre, L., Trassoudaine, L., 2006. Vehicle trajectories evaluation by static video sensors, in: IEEE Intelligent Transportation Systems Conference (ITSC), Toronto, Canada. pp. 864–869.

Heikkilä, M., Pietikäinen, M., 2006. A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 657–662.

Hofmann, M., Tiefenbacher, P., Rigoll, G., 2012. Background segmentation with feedback: The pixel-based adaptive segmenter, in: IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, Rhode Island, USA. pp. 38–43.

Laugraud, B., Latour, P., Van Droogenbroeck, M., 2015a. Time ordering shuffling for improving background subtraction, in: Advanced Concepts for Intelligent Vision Systems (ACIVS), Springer. pp. 58–69.

Laugraud, B., Piérard, S., Braham, M., Van Droogenbroeck, M., 2015b. Simple median-based method for stationary background generation using background subtraction algorithms, in: International Conference on Image Analysis and Processing (ICIAP), Workshop on Scene Background Modeling and Initialization (SBMI), Springer. pp. 477–484.

Laugraud, B., Piérard, S., Van Droogenbroeck, M., 2016. LaBGen-P: A pixel-level stationary background generation method based on LaBGen, in: IEEE International Conference on Pattern Recognition (ICPR), IEEE Scene Background Modeling Contest (SBMC), Cancun, Mexico.

Maddalena, L., Petrosino, A., 2008. A self-organizing approach to background subtraction for visual surveillance applications. IEEE Transactions on Image Processing 17, 1168–1177.

Maddalena, L., Petrosino, A., 2012. The SOBS algorithm: what are the limits?, in: IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, Rhode Island, USA. pp. 21–26.

Maddalena, L., Petrosino, A., 2014. Background model initialization for static cameras, in: Background Modeling and Foreground Detection for Video Surveillance. Chapman and Hall/CRC. chapter 3, pp. 3.1–3.16.

Maddalena, L., Petrosino, A., 2015. Towards benchmarking scene background initialization, in: International Conference on Image Analysis and Processing Workshops (ICIAP Workshops), pp. 469–476.

Manzanera, A., Richefeu, J., 2004. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation, in: Indian Conference on Computer Vision, Graphics and Image Processing, Kolkata, India. pp. 46–51.

Piérard, S., Van Droogenbroeck, M., 2015. A perfect estimation of a background image does not lead to a perfect background subtraction: analysis of the upper bound on the performance, in: International Conference on Image Analysis and Processing (ICIAP), Workshop on Scene Background Modeling and Initialization (SBMI), Springer. pp. 527–534.

Reddy, V., Sanderson, C., Lovell, B., 2011. A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts. EURASIP Journal on Image and Video Processing 164956, 13 pages.

Sobral, A., 2013. BGSLibrary: An OpenCV C++ background subtraction library, in: Workshop de Visao Computacional (WVC), Rio de Janeiro, Brazil. pp. 1–6.

Sobral, A., Bouwmans, T., h. Zahzah, E., 2015. Comparison of matrix completion algorithms for background initialization in videos, in: International Conference on Image Analysis and Processing Workshops (ICIAP Workshops), pp. 510–518.

St-Charles, P.L., Bilodeau, G.A., Bergevin, R., 2015. SuBSENSE: A universal change detection method with local adaptive sensitivity. IEEE Transactions on Image Processing 24, 359–373.

Stauffer, C., Grimson, E., 1999. Adaptive background mixture models for real-time tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Fort Collins, Colorado, USA. pp. 246–252.

Wang, H., Suter, D., 2006. A novel robust statistical method for background initialization and visual surveillance, in: Asian Conference on Computer Vision (ACCV), Berlin, Heidelberg. pp. 328–337.

Wang, Y., Jodoin, P.M., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P., 2014. Cdnet 2014: An expanded change detection benchmark dataset, in: IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, Ohio, USA. pp. 393–400.

Wang, Z., Simoncelli, E.P., Bovik, A.C., 2003. Multiscale structural similarity for image quality assessment, in: Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, USA. pp. 1398–1402.

Wren, C., Azarbayejani, A., Darrell, T., Pentland, A., 1997. Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 780–785.

Yalman, Y., Ertürk, I., 2013. A new color image quality measure based on YUV transformation and psnr for human vision system. Turkish Journal of Electrical Engineering & Computer Sciences 21, 603–613.

Zivkovic, Z., 2004. Improved adaptive Gaussian mixture model for background subtraction, in: IEEE International Conference on Pattern Recognition (ICPR), Cambridge, UK. pp. 28–31.

B. Laugraud, S. Piérard, and M. Van Droogenbroeck. **LaBGen: A method based on motion detection for generating the background of a scene**. In *Pattern Recognition Letters 96 (Special Issue on Scene Background Modeling and Initialization)* 12-21. DOI: 10.1016/j.patrec.2016.11.022.