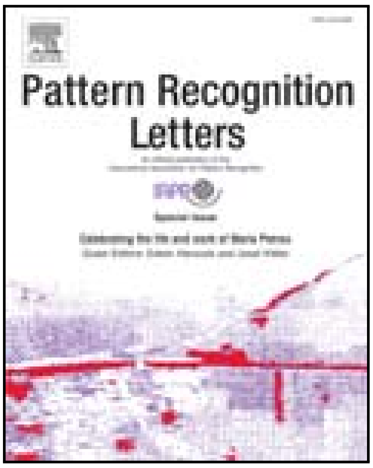

Parallel Multi-modal Background Modeling

Domenico D. Bloisi, Andrea Pennisi, Luca Iocchi

PII: S0167-8655(16)30312-9
DOI: 10.1016/j.patrec.2016.10.016
Reference: PATREC 6666

To appear in: *Pattern Recognition Letters*

Received date: 6 May 2016
Revised date: 21 October 2016
Accepted date: 29 October 2016

Highlights

- Fast and accurate background modeling.
- Robustness to the absence of clean frames.
- Per-pixel parallel computation.
- Quantitative experiments on several benchmark sequences.
- Publicly available source code.

Pattern Recognition Letters
journal homepage: www.elsevier.com

Parallel Multi-modal Background Modeling

Domenico D. Bloisi[a,**], Andrea Pennisi[a,b], Luca Iocchi[a]

[a] *Sapienza University of Rome, Department of Computer, Control, and Management Engineering, via Ariosto, 25 00185 Rome, Italy*
[b] *Vrije Universiteit Brussel (VUB), Department of Electronics and Informatics (ETRO), Pleinlaan 2 - B-1050 Brussel - Belgium*

ABSTRACT

Background subtraction is a widely used technique for detecting moving objects in image sequences. Very often background subtraction approaches assume the availability of one or more clear (i.e., without foreground objects) frames at the beginning of the sequence in input. However, this assumption is not always true, especially when dealing with dynamic background or crowded scenes. In this paper, we present the results of a multi-modal background modeling method that is able to generate a reliable initial background model even if no clear frames are available. The proposed algorithm runs in real–time on HD images. Quantitative experiments have been conducted taking into account six different quality metrics on a set of 14 publicly available image sequences. The obtained results demonstrate a high-accuracy in generating the background model in comparison with several other methods.

1. Introduction

Background subtraction (BS) is a popular and widely used technique that represents a fundamental building block for different Computer Vision applications, ranging from automatic monitoring of public spaces to augmented reality. The BS process is carried out by comparing the current input frame with the model of the scene background and considering as foreground points the pixels that differ from the model. Thus, the fundamental problem is to generate a background model that is as reliable as possible and consistent with the observed scene.

BS has been largely studied and many techniques have been developed for tackling the different aspects of the problem. This interest in BS is demonstrated by the many surveys published on this topic. For example, a survey on statistical background modeling has been conducted by Bouwmans et al. (2010) and a review about methods for multisensor surveillance has been realized by Cristani et al. (2010). A recent survey by Bouwmans (2014) provides a large overview of background models by dividing them in traditional and recent approaches.

In addition to the large literature, open-source software libraries have been released, so that also non-experts can exploit BS techniques for developing Computer Vision based systems.

However, open issues in BS (see Fig. 1) still need to be addressed, including how to deal with:

- Sudden and gradual illumination changes (e.g., due to clouds or time of day).
- Shadows (both hard and soft) and reflections.
- Camera jitter (e.g., due to wind in outdoor scenarios).

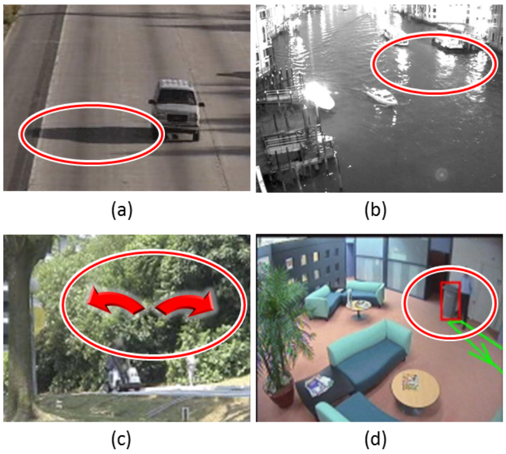

Fig. 1. Challenges for BS methods. a) Dark shadows (ATON data set). b) Reflections on water (MarDCT data set). c) Swaying trees (Perception Test Images Sequences). d) Moved furniture (CANDELA data set).

[**] Corresponding author: Tel.: +39-06-77274-063;
e-mail: bloisi@diag.uniroma1.it (Domenico D. Bloisi)

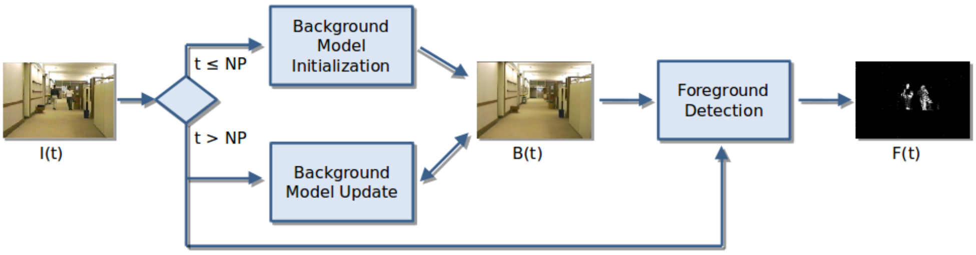

Fig. 2. Background subtraction process. *I(t)* is the input image at time *t*, *B(t)* the background model, and *F(t)* the foreground mask. *N* images, selected with a sampling period *P*, are used to generate and to update *B*.

- Background movement (e.g., waves on the water surface, swaying trees).
- Permanent and temporary changes in the background geometry (e.g., moving furniture in a room, parked cars).

BS Process. According to Bouwmans (2014), the BS process can be divided into three phases:

1. *Background model initialization*. N frames are collected with a sampling period P and analyzed to create the first background model B.
2. *Foreground detection*. The input image I is compared with the current background model B to detect moving objects in the scene.
3. *Background model update*. The background model B is updated over time to reflect possible changes in the scene.

Phase (1) is carried out only once, exploiting N frames at the beginning of the video sequence in input. Phases (2) and (3) are executed repeatedly as time progresses in order to adapt the background model coherently with with the changes in the scene (see Fig. 2).

Background Model Initialization. In contrast to the widely studied background model representation and model maintenance routines, limited attention has been given to the problem of initializing the background model (Bouwmans, 2014). In particular, often BS methods assume the availability of one or more *clean*, i.e., without foreground objects, frames at the beginning of the sequence in input (Maddalena and Petrosino, 2014b). Thus, the background model is initialized using the first frames, presuming that they do not contain foreground objects. This is a strong assumption that is not always true, because of continuous clutter presence.

Model Update. Two different policies can be used to modify the background model, namely *selective* and *blind* update. In selective (or conditional) update, only pixels classified as belonging to the background are updated. The selective update improves the detection of the targets since foreground information are not added to the background model, thus solving the problem of ghost observations. The use of information coming from the previous background model is highlighted in Fig. 2 by the bi-directional arrow between the update module and the background model B. However, when using selective updating, any incorrect pixel classification (e.g., due to illumination changes) produces a persistent error, since the background model will never adapt to it. This is why it is necessary also to have a blind update, where no update decisions are taken and every pixel in the background model is updated without considering the previous computed models. On the other hand, the blind update has the disadvantage that values not belonging to the background (e.g., stationary foreground objects) can be added to the model.

In this paper, we focus on the background initialization phase of the BS process when dealing with image sequences where no clean frames are available. We describe the results of an on-line and real-time parallel method, called Independent Multimodal Background Subtraction Multi-Thread (IMBS-MT), which is an extended version of the IMBS method, described in (Bloisi and Iocchi, 2012). The main contributions of this work are:

1. A parallel architecture to run in real-time with HD images (1360×768 pixels), with a publicly available source code[1].
2. An incremental background model generation to deal with sudden changes in the scene.

For the experimental evaluation, 14 test sequences provided by the Scene Background Initialization (SBI)[2] data set have been used. Quantitative results, obtained by considering six different quality metrics, demonstrate the capability of IMBS-MT to generate very accurate background models.

The rest of this paper is organized as follows. Related work is discussed in the next Section 2, giving particular emphasis to clustering-based BS methods and to existing software libraries. The proposed method is described in Section 3. The results of the quantitative comparison of IMBS-MT with other methods, carried out on publicly available image sequences, are shown in Section 4. Conclusions and future directions are discussed in Section 5.

2. Related Work

As pointed out in the previous section, BS has been extensively studied and many different approaches for generating accurate foreground masks have been published. Evaluations and comparisons for different BS methods have been presented in recent surveys realized by Sobral and Vacavant (2014) and Xu et al. (2016). From the large literature on BS algorithms, we

[1] http://www.dis.uniroma1.it/~bloisi/sw/imbs-mt.html
[2] http://sbmi2015.na.icar.cnr.it/SBIdataset.html

have decided to discuss here methods adopting the same ideas contained in our approach, namely a clustering algorithm for building the model, an adaptive mechanism to adjust the model in case of global variations in the scene, and a specific implementation to speed-up the foreground mask generation process. Furthermore, this section contains the description of a set of BS approaches for which open-source code and experimental data are available, since we believe that providing the source code for the algorithms and producing publicly available challenging benchmarks are fundamental requirements for achieving more and more reliable BS modules.

Clustering Approaches. One of the first real-time adaptive BS methods based on clustering has been proposed by Butler et al. (2003). The algorithm models each pixel by means of a group of k clusters and adapts the clusters to deal with variations in both the background and the ambient lighting. Incoming pixels are compared and classified against the corresponding cluster group by using the Manhattan distance. However, it is not possible to differentiate between moving objects and their shadows, which often cause the segmentation to blur or to erroneously detect shadows points as separate moving objects. Li et al. (2008) describe a method for background modeling and moving objects detection based on clustering theory. An histogram containing the pixel values over time is used to extract the moving objects by considering each peak in the histogram as a cluster. Fan et al. (2010) perform a k-means clustering and single Gaussian model to reconstruct the background through a sequence of scene images with foreground objects. Then, based on the statistical characteristics of the background pixel regions, the algorithm detects the moving objects. In addition, an adaptive algorithm for foreground detection is used in combination with morphological operators and a region-labeling mechanism. Kumar and Sureshkumar (2013) propose a modification of the k-means algorithm for computing BS in real-time. Their experimental results show that selecting centroids can lead to a better BS with the ability of handling images from dynamic environments.

Differently to the above-cited methods, the concept of *time interval* is a key factor in our approach. Indeed, we build the background model by considering N frame samples that are collected on the basis of a time period P. The details about our clustering algorithm are given in Section 3.

Adaptive Approaches. In order to deal with highly dynamic background (e.g., water scenarios, crowded scenes or dense urban traffic environments), it is crucial to consider a global model of the movement of the scene (Ablavsky, 2003). Indeed, the background model has to be sensitive enough to:

1. Detect moving objects.
2. Adapt to long-term lighting (e.g., time of day).
3. Take care of structural changes (e.g., objects entering the scene and becoming stationary).
4. Adjust to sudden background changes (e.g., clouds passing or light switching).

Combining local (i.e., pixel-wise) and global (i.e., frame-level) models allows to satisfy simultaneously both the sensitivity to foreground motion and the ability to model sudden background changes (Pennisi et al., 2015).

One of the first frame-level algorithm for dealing with global changes in the scene has been written by Toyama et al. (1999). A set of scene background models is maintained in memory and the one used is the model that produces the fewest number of foreground pixels. The approach is suitable only for situations where the scene presents cyclic changes (e.g., in the light switching problem). The authors provide also a set of publicly available sequences with ground-truth annotations called the Wallflower data set. More recently, Vosters et al. (2012) propose a real-time approach, which combines Eigenbackground with a statistical illumination model for coping with rapidly changing illumination conditions. The method is based on two algorithms: The former is used to reconstruct the background frame, the latter improves the foreground segmentation. However, the moved background objects are detected as foreground forever after movement. This is because the object's new location is not incorporated into the Eigenspace background model.

Vehicle traffic monitoring is an example of Computer Vision application that can be strongly affected by sudden illumination changes and weather issues. Nieto et al. (2012) present a vision-based system for vehicle tracking and classification devised for traffic flow surveillance. They propose an adaptive multi-cue segmentation strategy that detects foreground pixels corresponding to moving and stopped vehicles, even with noisy images due to compression. The approach adaptively thresholds a combination of luminance and chromaticity disparity maps between the learned background and the current frame, where the disparity maps are generated by comparing the values of the pixel luminance and chromaticity differences with respect to their corresponding temporal variances. Then, extra features derived from gradient differences are used to improve the segmentation of dark vehicles with casted shadows and to remove headlight reflections on the road. However, the method takes into account the color of the vehicle as a key feature, thus white vehicles can often be included in the background model.

In our approach, we use statistics computed at frame level to take care of global changes in the scene. If a large variation in the total number of foreground pixels is detected, then the background model is re-initialized in order to adapt to the new situation. The details about how our method manages global changes are given in Section 3.

Real-time/Parallel Implementations. Creating and maintaining probabilistic background models for high definition images is computationally expensive and can limit the real-time applications of BS methods to low resolution sequences, far below the acquisition ability of state-of-the-art cameras (Culibrk and Crnojevic, 2010). The Graphics Processing Unit (GPU) can be used to speed-up the computation of the foreground image, achieving real-time performance on high resolution frames.

Yang and Chen (2012) propose to use CPU and GPU as a combined computing unit in order to perform BS in dynamic background. GPU is employed to compute SIFT (Scale Invariant Feature Transform) points in order to match between two input frames, while CPU is used for compensating global motion through an affine transformation. Culibrk and Crnojevic (2010) present a parallel implementation for the Background Modeling Neural Networks (BNNs) method, which uses unsupervised learning. The GPU parallelization allows to handle 720×576 images in real-time. More recently, Wilson and Tavakkoli (2015) utilize Nvidia's CUDA architecture to accelerate their non-parametric background modeling method, which employs three stages of training, classification, and update, specifically designed for the parallel CUDA architecture. An OpenCL algorithm implementation for GPU devices of the GMM method, obtained by taking into account specific features of the GPU architecture, is presented in (Szwoch, 2015). For a video stream of 1920×1080 pixel images captured at 15 fps, it was possible to exceed the source rate only off-line, since the amount of computations was too large for CPUs to process the stream in online mode.

Also optimized implementation of well-known BS algorithms have been proposed to achieve real-time performance. In (Szwoch et al., 2016), Gaussian mixture models (GMM) and Codebook methods are tested on a supercomputer platform. The GMM algorithm proves to be significantly more efficient than the Codebook (about three times faster); however, in case of 1920×1200 images, the GMM algorithm is not able to work in real-time. Zivkovic and van der Heijden (2006) propose a modification of the Kernel Density Estimation (KDE) method, which uses a balloon variable-size kernel approach. The balloon approach leads to a very efficient implementation that is faster than the original KDE.

In this work, we do not use GPU acceleration to speed-up the BS process. Instead, we describe how to compute in parallel parts of the method by means of C++11 threads, obtaining real-time performance on HD images (1360×768 pixels).

Publicly Available Resources. The possibility of having the source code of the BS methods described in the literature represents a key point towards the goals of generating more and more accurate foreground masks and of widely applying this technology. OpenCV[3] is an open source Computer Vision library released under a BSD license and hence free for both academic and commercial use. OpenCV version 3 provides the source code for two BS methods:

1. MOG2: An improved adaptive Gaussian mixture model (Zivkovic, 2004);
2. KNN: K-Nearest Neighbors background subtraction described in (Zivkovic and van der Heijden, 2006).

BGSLibrary[4] is an OpenCV based C++ BS library containing the source code for both native methods from OpenCV and several approaches published in the literature (Sobral, 2013). The author also provides a JAVA graphical user interface (GUI) that can be used for comparing different methods.

A complete and updated collection of BS methods and publicly available data sets can be found in the Background Subtraction website[5]. A section of the website is dedicated to the available implementations of both traditional, e.g., statistical methods (Bouwmans, 2011), and recent emerging approaches, e.g., Fuzzy background modeling (Bouwmans, 2012). Another section of the website contains links and references for available BS data sets.

ChangeDetection (Goyette et al., 2012) is a benchmark data set containing several video sequences annotated with ground truth data. The sequences are grouped into different categories, like "Dynamic Background", "Camera Jitter", "Intermittent Object Motion", and "Shadow", which contain very challenging scenarios.

A database of surveillance videos and image sequences, dedicated to the maritime domain, is MarDCT – Maritime Detection, Classification, and Tracking data set (Bloisi et al., 2015). MarDCT has been developed for evaluating BS techniques on environments characterized by water background and for providing very challenging data (containing reflections, occlusions, waves, and wakes) from real working systems.

In this paper, we use the SBI data set for evaluating our approach and to obtain a quantitative comparison with other state-of-the-art methods. The details concerning the SBI sequences are given in Section 4.

3. Parallel Background Model Initialization

IMBS (Independent Multimodal Background Subtraction) is a BS method that has been designed for dealing with highly dynamic scenarios characterized by non-regular and high frequency noise, such as water background (Bloisi and Iocchi, 2009). IMBS is a per-pixel, non-recursive, and non-predictive BS method, meaning that:

- Each pixel signal is regarded as an independent process (per-pixel).
- A set of input frames is analysed to estimate the background model based on a statistical analysis of those frames (non-recursive).
- The order of the input frames is considered not significant (non-predictive).

The above listed design choices are fundamental for achieving a very fast computation, since *(i)* working at pixel level and *(ii)* considering each background model as independent from the previous computed ones allows for carrying out the BS process in parallel.

In the next sub-section, we briefly summarize the IMBS method, whose details can be found in (Bloisi and Iocchi, 2012). Then, we present an extended version of the original IMBS algorithm, called IMBS-MT (multi-tread), which is designed for parallel computation.

3.1. IMBS

The main idea behind IMBS is the discretization of the color distribution for each pixel, by using an on-line clustering algorithm. More specifically, for each pixel $p(i, j)$ the analysis of a set of N sample image frames is used to determine the

[3] http://opencv.org

[4] https://github.com/andrewssobral/bgslibrary

[5] https://sites.google.com/site/backgroundsubtraction

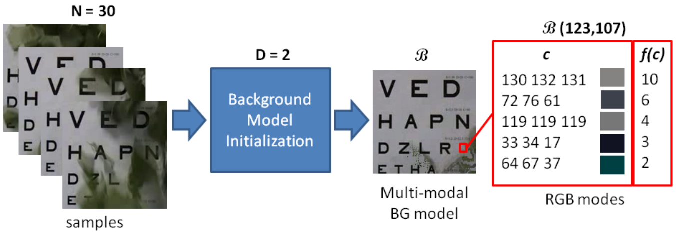

Fig. 3. IMBS stores multiple background values for each pixel. The five RGB modes for the BG model corresponding to the pixel in position (123, 107) are shown in this figure.

background model $\mathfrak{B}(i, j)$ for that pixel. $\mathfrak{B}(i, j)$ is a set of pairs $\langle c, f(c)\rangle$, where c is a value in the chosen color space (e.g., a triple in RGB or HSV space) and $f(c)$ is the number of occurrences of the value c in the sample set (see Fig. 3). After processing all the samples, only those color values that have enough occurrences (i.e., $\geq D$) are maintained in the background model. In this way, the background model contains, for each pixel, a discrete and compact multi-modal representation of its color probability distribution over time.

IMBS does not need to fit the data in some predefined distributions (e.g., Gaussian). This is the main difference with respect to a Mixture of Gaussians based approach (Stauffer and Grimson, 1999; Zivkovic, 2004), where fitting Gaussian distributions is required and typically the number of Gaussians is limited and determined *a priori*. Once the background model $\mathfrak{B}$ is computed, the foreground mask is built by using a quick thresholding method: A pixel $p(i, j)$ is considered as a foreground point if the current color value is not within the distribution represented in the model, i.e., its distance from all the color values in $\mathfrak{B}(i, j)$ is above a given threshold A. IMBS requires a time $R = NP$ for creating the first background model. Then a new model, independent from the previous one, is built continuously, according to the same refresh time R.

The functional architecture for IMBS is shown in Fig. 4, where the parameters in input to each module are highlighted.

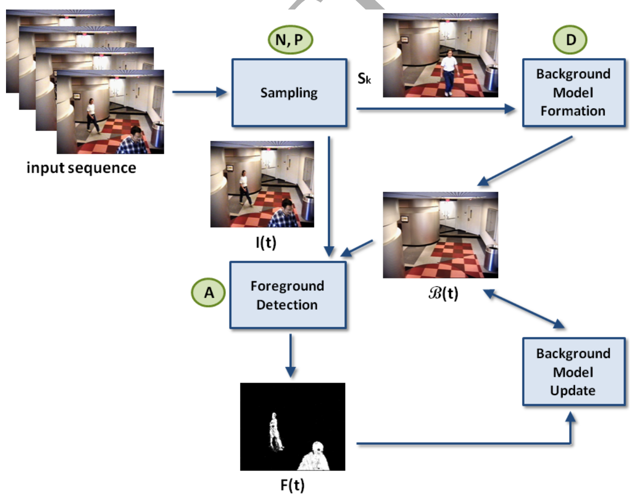

Fig. 4. IMBS functional architecture. The parameters used by the algorithm are circled.

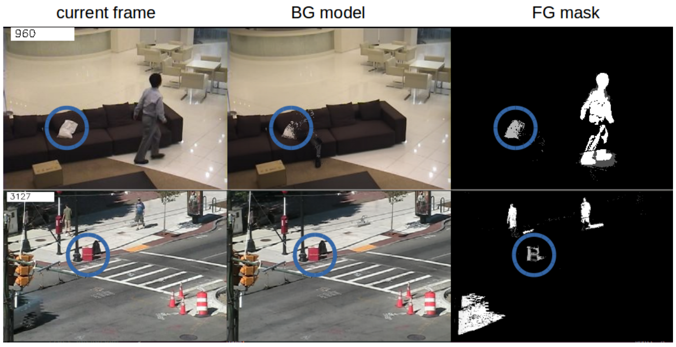

Fig. 5. Use of the "foreground modes" for detecting abandoned objects.

Background Model Update. IMBS adopts a hybrid update policy that allows for highlighting the pixels in the current foreground mask that represent not moving foreground regions. Indeed, IMBS uses the foreground mask $F(t)$ as feedback information to influence the model update, thus being able to point out the presence of stationary foreground objects in the scene.

Given a scene sample S_k and the current foreground binary mask F, if $F(i, j) = 1$ and $S_k(i, j)$ is within one of the modes in the new background model under development, then that mode is labeled as a "foreground mode". Once the background model is completed, if the pixel $p(i, j)$ of the current frame is associated with a foreground mode, then p is not considered as a simple background point, instead it is classified as a potential foreground pixel. If no changes happen in the scene, foreground modes are then absorbed in the successive background model.

The results of our hybrid update on two sequences from the ChangeDetection database are shown in Fig. 5. When an object is abandoned in the scene (first column in Fig. 5), even if the new background model after the event contains the representation of that object (second column), IMBS is able to discern between reliable (white colored) and potential (grey colored) foreground pixels (third column).

IMBS can achieve real-time computation (i.e., about 25 frames per second) for 640×480 input images (Bloisi et al., 2014). In the next sub-section, the parallel method IMBS-MT is described, which has been designed to achieve real-time performance on HD images (1360×768 pixels), as demonstrated by the experiments shown in Section 4.

3.2. *Parallel Method IMBS-MT*

IMBS-MT differs from the original IMBS in two aspects:

1. The background formation and foreground extraction processes are carried out in parallel on a disjoint set of subimages from the original input frame;
2. The background model is initialized incrementally, i.e., the quality of the model is increased as soon as more frame samples are available.

Multi-thread Implementation. If the number of available CPU cores is r, then the input image is split into r non-overlapping regions and each region is assigned to a thread. Instead of creating new images, the r regions are represented as references to the parts of the original whole image. This parallel procedure

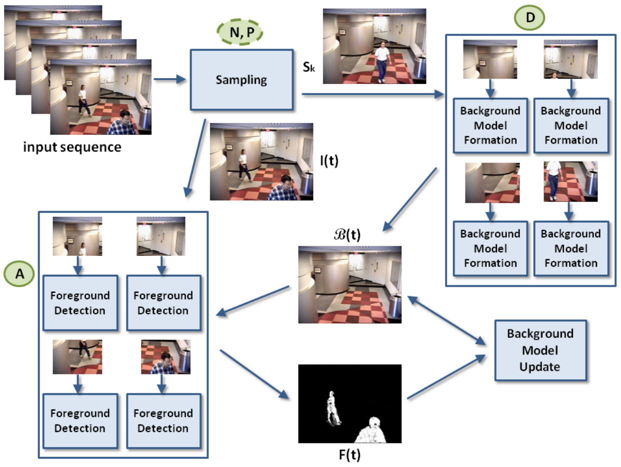

Fig. 6. IMBS-MT (multi-thread) functional architecture. This figure shows the configuration with four CPU cores available. ***N*** **and** ***P*** **(circled in dashed lines) parameters changes over time, due to the incremental model generation.**

is applied both for generating the foreground mask F and the background model $\mathfrak{B}$. When all the threads complete the processing, all the regions automatically join F and $\mathfrak{B}$ thanks to the used *reference* approach. The scheme in Fig. 6 shows the computation flow for IMBS-MT.

Incremental Background Model Generation. IMBS-MT uses an incremental process for initializing the BG model $\mathfrak{B}$, which is created by analyzing a sample set having a variable size. Fig. 7 shows the incremental background modeling mechanism that is carried out before reaching the steady state. In order to have a foreground mask available as quickly as possible, the first background model is built by considering N_1 images, with $N_1 \ll N$, for example $N_1 = N/5$. Next, the second model is generated after N_2P ms, where $N_1 < N_2 < N$, for example $N_2 = N/3$. The first stable model is obtained when $t_k + N_kP$ reaches the final value NP, after which a new model is generated every NP ms.

This incremental process is fundamental for dealing with sudden changes in the scene, since it allows for quickly recovering from situation where the collected scene samples are no more useful, due to changes in the environment. A way for triggering the incremental mechanism is to monitor the number of foreground pixels: If a large increment of the foreground pixels is detected, then the background model is replaced by starting the incremental background model initialization process.

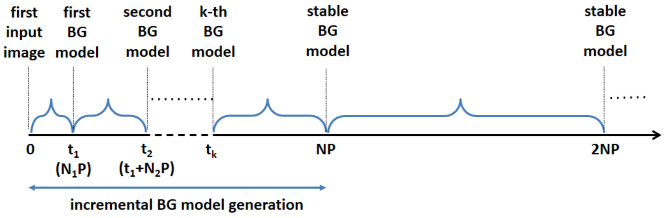

Fig. 7. Incremental BG model generation. In order to obtain a foreground mask as quickly as possible, the first BG models are computed processing a limited number of samples.

4. Experimental Results

In order to experimentally evaluate the performance of our method, 14 different image sequences, provided in the Scene Background Initialization (SBI) data set, have been used. The SBI sequences have been extracted from multiple publicly available sequences that are frequently used in the literature to evaluate background initialization algorithms.

We use for comparison the results generated by nine other BS methods, i.e., Median (Maddalena and Petrosino, 2014a), SC-SOBS (Maddalena and Petrosino, 2012), WS2006 (Wang and Suter, 2006), RSL2011 (Reddy et al., 2010), Photomontage (Agarwala et al., 2004), CA2008 (Chen and Aggarwal, 2008), KNN, and MOG2. For the KNN and MOG2 methods, we have used the implementation available in OpenCV 3 for computing the results, while, for the other methods, we show the results provided by Maddalena and Petrosino (2015), published in occasion of the Scene Background Modeling and Initialization (SBMI2015) Workshop, held in conjunction with ICIAP 2015. However, since in (Maddalena and Petrosino, 2015) only a subset of all the SBI sequences are considered, the quantitative experimental results are split in two tables: Table 1 contains the results of the experiments performed also by Maddalena and Petrosino (2015) on seven out of the total fourteen SBI sequences, while Table 2 shows the results obtained on the remaining seven sequences.

4.1. Accuracy Evaluation

The proposed method has been evaluated both qualitatively and quantitatively, by using the SBI scripts[6] for computing the results. We have maintained the default OpenCV parameters for KNN and MOG2 on all the sequences, while we have used the parameters $P = 500$ ms, $N = 30$, $D = 2$, and $A = 5$ for IMBS-MT on all the sequences except for "Toscana", which contains only six frames and thus we set $N = 5$.

The qualitative evaluation is illustrated in Fig. 8, where the first column contains a sample frame for each sequence. The second column contains the ground truth images provided by the SBI authors, which have been manually obtained by either choosing one of the sequence frames free of foreground objects or by stitching together empty background regions from different sequence frames. The background model images computed with the MOG2, KNN and IMBS-MT methods are shown in the second, third, and forth columns, respectively. It is worth noting that, for MOG2 and KNN, we created the model images with the OpenCV function *getBackgroundImage*. Instead, for IMBS-MT, the model image to be used with the SBI scripts has been obtained by selecting, for each pixel p, the mode with the minimum distance d from the ground truth:

$$d = \arg\min_k |r_k - r_{GT}| + |g_k - g_{GT}| + |b_k - b_{GT}|$$

where (r_k, g_k, b_k) is one of the modes in $\mathfrak{B}$ for the pixel p and (r_{GT}, g_{GT}, b_{GT}) is the corresponding ground truth value.

[6] `http://sbmi2015.na.icar.cnr.it/MODLab/BckgInit/MATLAB/EvaluateBckgInit3.zip`

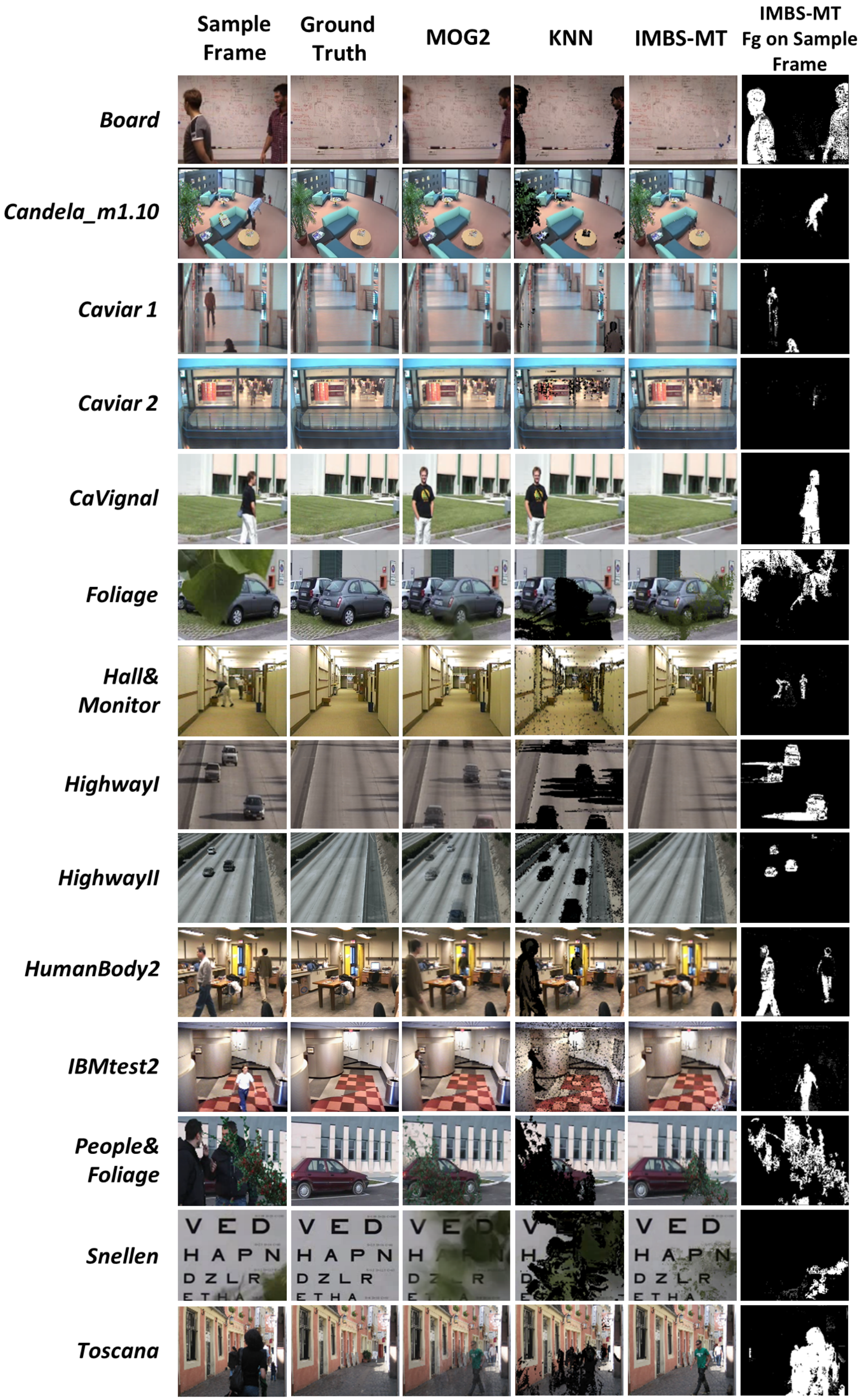

Fig. 8. Qualitative results on the Scene Background Initialization (SBI) data set.

This procedure allows to produce a bi-dimensional image by considering the "best" background mode, which is in any case a mode produced by our algorithm without introducing ground truth information. Indeed, IMBS-MT labels a pixel as a foreground point only if it differs from all the background color values. Therefore, no background modes are more important than the other ones. The last column in Fig. 8 contains the foreground masks generated by IMBS-MT on the sample frames shown in the first column.

Table 1 and Table 2 show the quantitative experimental results obtained on the fourteen SBI sequences (bold font is used to denote the best performance). To compute the results, we have used the quality metrics suggested in the SBI web page and listed below, where GT and CB denote the true and estimated background, respectively.

1. Average Gray-level Error (AGE) is the average of the gray-level absolute difference between GT and CB images. Its values range in $[0, L-1]$, where L is the maximum number

Table 1. Results on seven image sequences from the SBI data set.

Sequence	Method	AGE	pEPs	pCEPS	MS-SSIM	PSNR	CQM
Hall & Monitor	KNN	3.9413	0.0121	0.0021	0.9519	28.2208	37.4907
	MOG2	2.4506	**0.0109**	0.0045	0.9833	34.3943	45.9714
	Median	2.7105	0.9931	0.5339	0.9640	30.4656	42.6705
	SC-SOBS	2.4493	0.9801	0.3220	0.9653	30.4384	43.1867
	WS2006	2.6644	0.5563	0.0308	0.9821	30.9313	40.0949
	RSL2011	3.2687	0.8321	0.4711	0.9584	28.4428	37.9971
	Photomontage	2.7986	0.3610	0.0817	0.9819	33.3715	41.7323
	CA2008	2.4737	0.3989	**0.0000**	0.9878	32.2503	41.2399
	IMBS-MT	**1.5350**	0.0923	**0.0000**	**0.9954**	**38.6214**	**48.5224**
HighwayI	KNN	6.1277	0.0616	0.0003	0.8506	25.1521	34.8174
	MOG2	2.6031	0.0023	0.0002	0.9753	35.8635	58.2889
	Median	1.4275	0.1563	0.0143	0.9924	40.1432	62.5723
	SC-SOBS	**1.2286**	**0.0039**	**0.0000**	**0.9949**	**42.6868**	**65.5755**
	WS2006	2.5185	0.6849	0.0247	0.9816	35.6885	56.9113
	RSL2011	2.8139	0.3477	0.0430	0.9830	36.0290	51.9835
	Photomontage	2.1745	0.4076	0.0482	0.9830	37.1250	59.0270
	CA2008	2.9477	1.1654	0.0846	0.9752	33.9800	56.1319
	IMBS-MT	1.4913	0.0612	0.0026	0.9939	41.7728	58.8328
HighwayII	KNN	3.2112	0.0085	0.0001	0.9851	32.0981	39.6454
	MOG2	2.0893	**0.0040**	**0.0000**	0.9946	36.1190	45.2643
	Median	1.7278	0.3190	0.0013	0.9961	34.6639	42.3162
	SC-SOBS	**0.6536**	0.0091	**0.0000**	**0.9982**	**44.6312**	**54.3785**
	WS2006	2.4906	0.4883	0.0130	0.9927	33.9515	40.5088
	RSL2011	5.6807	1.2448	0.4115	0.9766	28.6703	35.0821
	Photomontage	2.4306	0.5885	0.0052	0.9909	34.3975	41.7656
	CA2008	2.434	0.6328	0.0560	0.9919	33.5545	39.4813
	IMBS-MT	1.8684	0.0260	**0.0000**	0.9960	40.1098	48.8094
CaVignal	KNN	15.9267	0.0813	0.0127	0.8241	18.2332	30.9930
	MOG2	16.9327	0.1114	0.0837	0.8136	18.5891	34.5104
	Median	10.3082	10.4632	8.1066	0.7984	18.1355	33.1438
	SC-SOBS	4.0941	3.1949	1.6029	0.8779	21.8507	42.2652
	WS2006	2.5403	1.5000	0.4743	0.9289	27.1089	37.0609
	RSL2011	1.6132	**0.0147**	**0.0000**	0.9967	41.3795	52.5856
	Photomontage	11.2665	11.2206	8.8529	0.7919	17.6257	32.0570
	CA2008	9.2569	0.0625	**0.0000**	0.9932	27.5197	39.7879
	IMBS-MT	**0.7692**	**0.0147**	**0.0000**	**0.9982**	**45.9202**	**57.1044**
Foliage	KNN	34.5615	0.3962	0.0385	0.6281	14.1761	25.6845
	MOG2	32.3624	0.6685	0.5526	0.8038	16.5991	31.5282
	Median	27.0135	47.3125	30.4583	0.6444	16.7842	28.7321
	SC-SOBS	3.8215	0.5556	**0.0000**	0.9900	31.7713	39.1387
	WS2006	6.8649	2.8507	0.0069	0.9754	27.2438	34.9776
	RSL2011	2.2773	0.1493	0.0382	0.9951	36.7450	43.1208
	Photomontage	**1.8592**	**0.0000**	**0.0000**	**0.9974**	**39.1779**	**45.6052**
	CA2008	18.3613	11.5521	4.3681	0.9092	18.7767	29.9137
	IMBS-MT	7.5809	9.8507	3.1319	0.9090	22.7278	34.0028
People & Foliage	KNN	48.4920	0.4718	0.2966	0.4238	10.9196	19.8121
	MOG2	33.8442	0.7108	0.6134	0.8584	16.2252	27.4728
	Median	24.4211	32.2396	25.3203	0.6114	15.1870	27.4979
	SC-SOBS	15.1031	14.0234	5.0117	0.7561	16.6189	35.3667
	WS2006	5.4243	3.5716	0.0924	0.9269	22.6952	31.3847
	RSL2011	2.0980	0.7969	0.5651	0.9905	32.5550	37.0598
	Photomontage	**1.4103**	**0.0039**	**0.0000**	**0.9973**	**41.0866**	**47.1517**
	CA2008	19.7347	12.2409	6.1914	0.8220	17.1567	25.9970
	IMBS-MT	8.3982	7.3568	3.2305	0.8514	20.0658	32.5231
Snellen	KNN	61.9389	0.6832	**0.4328**	0.4493	10.6164	22.5804
	MOG2	58.8159	0.7615	0.6839	0.5336	11.4143	27.0312
	Median	42.3981	62.2010	56.9734	0.6932	13.6573	36.0691
	SC-SOBS	16.8898	37.3553	24.3779	0.9303	21.2571	44.7498
	WS2006	23.0010	23.1674	12.2685	0.7481	15.6158	24.9930
	RSL2011	**1.8095**	**0.6414**	0.4774	**0.9979**	**38.0295**	**50.2600**
	Photomontage	29.9797	33.4973	30.4688	0.5926	14.1466	26.9210
	CA2008	40.5218	44.2371	30.6665	0.6886	12.9428	24.0239
	IMBS-MT	14.4480	25.3279	19.7290	0.8668	19.7436	40.1151

of grey levels.

2. Percentage of Error Pixels (pEPs) is the ratio between the EPs and the number N of image pixels. Its values range in $[0, 1]$.
3. Percentage of Clustered Error Pixels (pCEPs) is the ratio between the CEPs and the number N of image pixels. Its values range in $[0, 1]$.
4. Multi-Scale Structural Similarity Index (MS-SSIM) uses structural distortion as an estimate of the perceived visual distortion. It assumes values in $[0, 1]$.
5. Peak-Signal-to-Noise-Ratio (PSNR) is defined as:

$$PSNR = 10 \log_{10} \frac{(L-1)^2}{MSE}$$

where L is the maximum number of grey levels and MSE is the Mean Squared Error between GT and CB images. It assumes values in decibels.

6. Color image Quality Measure (CQM) is based on a reversible transformation of the YUV color space and on the PSNR computed in the single YUV bands. As for the

Table 2. Results on seven image sequences from the SBI data set.

Sequence	Method	AGE	pEPs	pCEPS	MS-SSIM	PSNR	CQM
Board	KNN	31.1259	26.8963	17.2561	0.7734	13.4368	21.3434
	MOG2	21.5981	23.3689	15.2652	0.8433	17.0541	29.2805
	IMBS-MT	**2.2537**	**0.3201**	**0.0061**	**0.9836**	**36.8244**	**52.4920**
Candela m1.10	KNN	11.2176	10.0507	5.8179	0.8158	17.3467	23.1109
	MOG2	1.7044	0.7694	0.6185	**0.9914**	34.0895	**46.9634**
	IMBS-MT	**1.3823**	**0.4705**	**0.0957**	0.9893	**35.4288**	44.2374
CAVIAR1	KNN	4.6259	4.0415	2.8463	0.9348	24.3250	29.9878
	MOG2	3.1274	2.8412	2.3621	0.9722	29.3055	41.4591
	IMBS-MT	**1.2267**	**0.0539**	**0.0214**	**0.9967**	**42.2244**	**55.0816**
CAVIAR2	KNN	7.1935	4.8910	1.4404	0.8469	18.9970	25.5783
	MOG2	1.4154	0.1658	0.0997	0.9974	40.1120	53.4368
	IMBS-MT	**1.2948**	**0.0102**	**0.0000**	**0.9986**	**43.0235**	**53.7161**
HumanBody2	KNN	20.9423	18.5130	15.2188	0.7783	14.5871	21.4805
	MOG2	11.2767	13.4609	9.9427	0.8752	19.5258	32.1251
	IMBS-MT	**1.9190**	**0.5794**	**0.0534**	**0.9958**	**34.0997**	**45.2074**
IBMtest2	KNN	21.3572	16.2995	2.3099	0.6671	14.1235	20.5705
	MOG2	**3.1981**	**1.5039**	0.7083	0.9717	**30.5180**	**38.1524**
	IMBS-MT	7.3508	3.2734	**0.1328**	**0.9721**	24.6275	36.4310
Toscana	KNN	19.0935	22.7581	17.5800	**0.7034**	16.3492	16.2754
	MOG2	9.5929	12.8060	8.3773	0.8947	**23.5968**	**23.1972**
	IMBS-MT	**7.4109**	**6.9096**	**5.2394**	0.8903	22.5367	22.0319

PSNR, it assumes values in decibels.

For the metrics AGE, pEPs, and pCEPs the lower the value, the better is the background estimate, while for MS-SSIM, PSNR, and CQM the higher the value, the better is the background estimate.

Discussion. The experimental results demonstrate that IMBS-MT achieves very good results, comparable with the other considered methods. It is also possible to note that no one method is able to obtain the best performance over all the sequences.

The main advantages of IMBS-MT over the other methods are:

1. IMBS-MT maintains good performance also when the nature of the background is highly dynamic. This is due to the specific capacity of IMBS-MT to model scenes with dynamic background, since IMBS-MT does not consider a predefined distribution of the pixel values in the background. When stationary objects are in the scene, the model update mechanism of IMBS-MT produces very good results, as demonstrated by the results on the "CaVignal" sequence.
2. IMBS-MT is a very fast algorithm, able to achieve good results without the need of a long processing time (as demonstrated by the computational performance analysis given below).

4.2. Computational Performance

The functional architecture shown in Fig. 6 has been implemented to take advantage of parallel execution by using the class Thread provided by C++11. In particular, *r* threads runs in parallel for creating the background model, since each thread process one of *r* the regions the image is split into. In the same way, it is also possible to obtain a fast computation of the foreground mask by exploiting the parallel execution of *r* threads.

Table 3. Computational load in terms of FPS on different computer display standards for KNN (with TBB), MOG2 (with TBB), IMBS (mono-thread), and IMBS-MT (multi-thread).

Video Standard	Frame Size	KNN	MOG2	IMBS	IMBS-MT
Video CD	352×240	144.702	80.0249	35.13	150.32
HD	1360×768	18.5798	6.2748	12.49	25.31
HD+	1600×900	13.7526	3.8475	8.42	20.72
Full HD	1920×1080	9.7816	2.9693	3.45	13.52

In order to ensure real-time performance on high resolution images, we collected 9 video sequences of an urban scenario captured at four different resolutions and measured the computational speed of IMBS-MT on an Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz, 8 GB RAM. The results are shown in Table 3, where the performance of IMBS-MT is compared with the TBB parallel implementation of MOG2 and KNN algorithms and with the mono-thread implementation of IMBS. Our IMBS-MT method achieves a computational speed of more than 13 frame per seconds on Full High-Definition (Full HD) images, real-time performance on HD frames, and a very high processing speed, i.e., more than 150 frames per second, on 352×240 images.

It is worth nothing that, the possibility of working with Full HD data allows for using high-level image processing routines after the foreground extraction, such as face recognition and plate identification. Moreover, with the computational speed achieved by IMBS-MT, it is possible to process simultaneously up to four HD video streams in real-time on a single PC.

The C++ source code of IMBS-MT is publicly available and can be downloaded from the following repository:
`https://github.com/dbloisi/imbs-mt`

5. Conclusions

In this paper, we have described a fast clustering-based background subtraction method, called IMBS-MT. The key aspect of IMBS-MT is the capacity of generating an accurate background model even if no clear frames (i.e., images without foreground objects) are present in the image sequence in input. IMBS-MT includes a mechanism for computing the background model incrementally and it has the capacity of carrying out in parallel the background formation and foreground extraction processes.

Experimental results, obtained on the challenging sequences of the SBI data set, demonstrate that IMBS-MT can generate highly accurate initial background models with a very high speed. Quantitative results obtained by IMBS-MT have been compared with eight state-of-the-art BS methods, i.e., KNN, MOG2, Median, SC-SOBS, WS2006, RSL2011, Photomontage, and CA2008, obtaining good results with respect to six different quality metrics. As a difference with other methods, IMBS-MT has been designed for maintaining a good accuracy with real-time computational speed on HD images and it can be used to process up to four HD video streams on a single CPU at the same time.

As future work, we intend to further improve the speed of the algorithm by creating a GPU-based implementation for IMBS-MT.

References

Ablavsky, V., 2003. Background models for tracking objects in water, in: ICIP (3), pp. 125–128.

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M., 2004. Interactive digital photomontage. ACM Trans. Graph. , 294–302.

Bloisi, D.D., Iocchi, L., 2009. ARGOS - A video surveillance system for boat trafic monitoring in venice. IJPRAI 23, 1477–1502.

Bloisi, D.D., Iocchi, L., 2012. Independent multimodal background subtraction, in: CompIMAGE, pp. 39–44.

Bloisi, D.D., Iocchi, L., Pennisi, A., Tombolini, L., 2015. ARGOS-Venice boat classification, in: AVSS, pp. 1–6.

Bloisi, D.D., Pennisi, A., Iocchi, L., 2014. Background modeling in the maritime domain. Machine Vision and Applications 25, 1257–1269.

Bouwmans, T., 2011. Recent advanced statistical background modeling for foreground detection: A systematic survey. Recent Patents on Computer Science 4, 147–176.

Bouwmans, T., 2012. Background subtraction for visual surveillance: A fuzzy approach, in: Handbook on Soft Computing for Video Surveillance. Taylor and Francis Group. chapter 5, pp. 103–138.

Bouwmans, T., 2014. Traditional and recent approaches in background modeling for foreground detection: An overview. Computer Science Review 1112, 31–66.

Bouwmans, T., El Baf, F., Vachon, B., 2010. Statistical background modeling for foreground detection: A survey, in: Handbook of Pattern Recognition and Computer Vision. World scientific Publishing, pp. 181–199.

Butler, D., Sridharan, S., Bove, V.M.J., 2003. Real-time adaptive background segmentation, in: ICASSP, pp. 349–352.

Chen, C.C., Aggarwal, J.K., 2008. An adaptive background model initialization algorithm with objects moving at different depths., in: ICIP, pp. 2664–2667.

Cristani, M., Farenzena, M., Bloisi, D.D., Murino, V., 2010. Background subtraction for automated multisensor surveillance: A comprehensive review. EURASIP J. Adv. Sig. Proc. , 1–24.

Culibrk, D., Crnojevic, V., 2010. Gpu-based complex-background segmentation using neural networks, in: IMVIP, pp. 262–275.

Fan, T., Li, L., Tian, Q., 2010. A novel adaptive motion detection based on k-means clustering, in: ICCSIT, pp. 136–140.

Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P., 2012. Changedetection.net: A new change detection benchmark dataset, in: CVPR Workshops, pp. 1–8.

Kumar, A., Sureshkumar, C., 2013. Background subtraction based on threshold detection using modified k-means algorithm, in: PRIME, pp. 378–382.

Li, Q., He, D., Wang, B., 2008. Effective moving objects detection based on clustering background model for video surveillance, in: CISP, pp. 656–660.

Maddalena, L., Petrosino, A., 2012. The SOBS algorithm: What are the limits?, in: CVPR Workshops, pp. 21–26.

Maddalena, L., Petrosino, A., 2014a. The 3dsobs+ algorithm for moving object detection. Computer Vision and Image Understanding 122, 65–73.

Maddalena, L., Petrosino, A., 2014b. Background model initialization for static cameras, in: Handbook on Background Modeling and Foreground Detection for Video Surveillance. Chapman and Hall/CRC, pp. 3–1–3–16.

Maddalena, L., Petrosino, A., 2015. Towards benchmarking scene background initialization, in: ICIAP Workshops, pp. 469–476.

Nieto, M., Cortes, A., Barandiaran, J., Otaegui, O., Sanchez, P., 2012. Adaptive multicue background subtraction for robust vehicle counting and classification. IEEE Transactions on Intelligent Transportation Systems 13, 527–540.

Pennisi, A., Previtali, F., Bloisi, D.D., Iocchi, L., 2015. Real-time adaptive background modeling in fast changing conditions, in: AVSS, pp. 1–6.

Reddy, V., Sanderson, C., Lovell, B.C., 2010. A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts. EURASIP Journal on Image and Video Processing , 1–14.

Sobral, A., 2013. BGSLibrary: An OpenCV C++ background subtraction library, in: WVC, pp. 1–6.

Sobral, A., Vacavant, A., 2014. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Computer Vision and Image Understanding 122, 4–21.

Stauffer, C., Grimson, W., 1999. Adaptive background mixture models for real-time tracking, in: ICCV, pp. 246–252.

Szwoch, G., 2015. Performance evaluation of parallel background subtraction on gpu platforms. Elektronika: konstrukcje, technologie, zastosowanian 56, 23–27.

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

Toyama, K., Krumm, J., Brumitt, B., Meyers, B., 1999. Wallflower: principles and practice of background maintenance, in: ICCV, pp. 255–261.

Vosters, L., Shan, C., Gritti, T., 2012. Real-time robust background subtraction under rapidly changing illumination conditions. Image Vision Comput. 30, 1004–1015.

Wang, H., Suter, D., 2006. A novel robust statistical method for background initialization and visual surveillance, in: ACCV, pp. 328–337.

Wilson, B., Tavakkoli, A., 2015. An efficient non-parametric background modeling technique with cuda heterogeneous parallel architecture, in: ISVC. Springer International Publishing, pp. 210–220.

Xu, Y., Dong, J., Zhang, B., Xu, D., 2016. Background modeling methods in video analysis: A review and comparative evaluation. CAAI Transactions on Intelligence Technology 1, 43–60.

Yang, Y., Chen, W., 2012. Parallel algorithm for moving foreground detection in dynamic background, in: ISCID, pp. 442–445.

Zivkovic, Z., 2004. Improved adaptive gaussian mixture model for background subtraction, in: ICPR, pp. 28–31.

Zivkovic, Z., van der Heijden, F., 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters 27, 773–780.