Computer Vision and Image Understanding 200 (2020) 103032

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Texture collinearity foreground segmentation for night videos

Isabel Martins [a,d,*], Pedro Carvalho [a,b], Luís Corte-Real [b,c], José Luis Alba-Castro [d]

[a] ISEP, Polytechnic of Porto, Porto, Portugal
[b] INESC TEC, Porto, Portugal
[c] Faculty of Engineering, University of Porto, Porto, Portugal
[d] atlanTTic Research Center, University of Vigo, Vigo, Spain

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
Background subtraction
Foreground segmentation
Change detection
GMM
MOG
Night videos
Texture features
Texture matching

ABSTRACT

One of the most difficult scenarios for unsupervised segmentation of moving objects is found in nighttime videos where the main challenges are the poor illumination conditions resulting in low-visibility of objects, very strong lights, surface-reflected light, a great variance of light intensity, sudden illumination changes, hard shadows, camouflaged objects, and noise. This paper proposes a novel method, coined COLBMOG (COLlinearity Boosted MOG), devised specifically for the foreground segmentation in nighttime videos, that shows the ability to overcome some of the limitations of state-of-the-art methods and still perform well in daytime scenarios. It is a texture-based classification method, using local texture modeling, complemented by a color-based classification method. The local texture at the pixel neighborhood is modeled as an N-dimensional vector. For a given pixel, the classification is based on the collinearity between this feature in the input frame and the reference background frame. For this purpose, a multimodal temporal model of the collinearity between texture vectors of background pixels is maintained. COLBMOG was objectively evaluated using the ChangeDetection.net (CDnet) 2014, Night Videos category, benchmark. COLBMOG ranks first among all the unsupervised methods. A detailed analysis of the results revealed the superior performance of the proposed method compared to the best performing state-of-the-art methods in this category, particularly evident in the presence of the most complex situations where all the algorithms tend to fail.

1. Introduction

Segmentation of foreground objects in video sequences is a fundamental step in many computer vision applications and a critical factor for the success of the overall system. Developing robust and universal methods for unsupervised segmentation of moving objects in video sequences has proved to be a hard and challenging task, particularly for uncontrolled environments such as in outdoor scenes (Kim and Jung, 2017). State-of-the-art methods show good performance in a wide range of situations, but systematically fail when facing more complex and challenging scenarios, such as the ones found in outdoor nighttime videos, that include poor illumination conditions resulting in low-visibility of objects, very strong lights, surface-reflected light, a great variance of light intensity, sudden illumination changes, camouflaged objects, hard shadows and noise. These challenges have a great impact on the detection of foreground objects.

Comprehensive reviews of current approaches for foreground segmentation have been presented in Bouwmans (2014), Sobral and Vacavant (2014), Bouwmans (2011), Brutzer et al. (2011), Benezeth et al. (2010), Bouwmans and Zahzah (2014) and Elhabian et al. (2008). Despite the large number of methods proposed in the literature, none has been able to deal with all challenges. Most of the works reported in the literature are focused on daytime environments. Nighttime videos have been considered one of the most difficult scenarios to deal with, in the context of unsupervised segmentation of moving objects (Wang et al., 2014b). Under the difficult illumination conditions typical of the nighttime environment, the obvious features of objects which are effective for segmentation in daytime become invalid. However, the lack of algorithms developed to address the problems specific to the segmentation of moving objects in nighttime videos is evident, with just a few exceptions (Zhao et al., 2008; Li et al., 2011). The method proposed in Zhao et al. (2008) models the background using spatio-temporal patches, called bricks, and the background is learned using online subspace learning. However, the experiments reported were conducted using a private dataset, and no objective evaluation is presented. In Li et al. (2011), the authors combine the background subtraction task and the object detection task into one framework, which also includes the method proposed in Zhao et al. (2008), but the experiments reported were also conducted using a private dataset, and the results for foreground segmentation were compared only with the results obtained with the method in Zhao et al. (2008).

* Corresponding author at: ISEP, Polytechnic of Porto, Porto, Portugal.
E-mail address: mis@isep.ipp.pt (I. Martins).

https://doi.org/10.1016/j.cviu.2020.103032
Received 3 April 2019; Received in revised form 23 June 2020; Accepted 24 June 2020
Available online 29 June 2020

This paper proposes a new method, coined COLBMOG (COLlinearity Boosted MOG), to address the problem of unsupervised foreground segmentation in nighttime videos, that relies upon a texture-based change detection method which exploits a local texture feature. Texture features, such as Local Binary Pattern (LBP) (Heikkilä and Pietikäinen, 2006) and its many variants, have shown to be more robust to illumination changes than color features. We propose a richer representation of the local texture at the pixel neighborhood by looking at the values of the luminance of the pixel and its N-1 neighbors as a vector in an N-dimensional space. In the absence of structural changes in the background scene, the difference between the texture of a pixel neighborhood in the current frame and a reference background frame can essentially result from illumination variations or noise. Thus, measuring collinearity results in increased robustness to uniform illumination variations while improving the odds of detecting moving objects, including camouflaged objects, when their texture differs from the background texture. The proposed method, therefore, builds and maintains an updated GMM-based temporal model of the collinearity between texture vectors of background pixels, allowing the detection of foreground pixels as those that do not match this model. This method is complemented by a color-based background model (Martins et al., 2018) that explores the characteristics of color spaces that separate luminance from chrominance. This color-based background model provides not only a reference background image for the texture-matching process but also a color-based classification mechanism that, in specific situations, is complementary to the texture-based approach.

We evaluated COLBMOG using the ChangeDetection.net (CDnet) 2014, Night Videos category, benchmark (ChangeDetection.NET, 2014). COLBMOG ranks first among all the unsupervised methods.[1] We compared in detail our results with the two top-performing methods in this category. The results obtained show that the superior performance of COLBMOG is particularly evident in the presence of the most challenging situations, where all the algorithms tend to fail.

The remainder of the paper is organized as follows. Section 2 presents an overview of current trends in background subtraction methods and the motivation for our work. Section 3 details the proposed COLBMOG method to segment foreground objects, including the local texture modeling, the texture-based classification and the complementary mechanisms implemented to deal with very dark areas and textureless foreground objects. The experimental setup is detailed in Section 4 and the results are presented and discussed in Section 5. Final conclusions are presented in Section 6.

2. Trends in moving object segmentation

2.1. Background subtraction methods

Background subtraction (BS) is a widely used approach for detecting foreground objects in video sequences (Bouwmans, 2014; Sobral and Vacavant, 2014; Bouwmans, 2011; Brutzer et al., 2011; Benezeth et al., 2010; Elhabian et al., 2008). The principle for this approach is that foreground objects can be detected from the difference between the current frame and a reference frame or a background model. As a result, background modeling has become a widely used approach, and many new methods have been proposed in the last decades, for the robust and efficient modeling of the background.

In general, background modeling methods exploit the temporal variation of one or more features of each pixel to maintain an updated model of the background and extract the pixels belonging to the foreground objects, as those whose associated features do not match this model. The type of features selected and the type of background model used, conform a plethora of different methods in the literature.

[1] When this paper was submitted for reviewing.

2.1.1. Features

Most methods use low-level features in the pixel domain, with the most common being color, edges, texture or motion (Bouwmans et al., 2018).

Color features are among the most popular and are usually defined in the RGB color space because it is directly available from the sensor or the camera. However, it is well known that the R, G, and B color components are correlated and this results in the increased sensitivity to illumination changes (López-Rubio and López-Rubio, 2015). Other color spaces have been explored (Kristensen et al., 2006) with the best results being obtained with color spaces that separate luminance from chrominance, such as YCrCb or L*a*b* (Balcilar et al., 2014; Martins et al., 2018). Color features work well in many scenarios, but their discriminative ability decreases considerably in the presence of illumination changes, camouflaged objects, and shadows. Alternative features that proved to be more robust to these challenges are edge features and texture features.

Edge features are generally computed, from the gray level image or from each color component, using a gradient approach (López-Rubio and López-Rubio, 2015; Li et al., 2004; Kim et al., 2015; Azab et al., 2010; Holtzhausen et al., 2015). Although edge features can be considered more robust to illumination changes, edge position, shape and length may change in consecutive frames due to noise. Edge features can be used alone (Jain et al., 2007; Allebosch et al., 2015) but are usually combined with color features (Lindström J. Lindgren et al., 2006; Allebosch et al., 2016).

Texture features are more robust to illumination changes than color features. Some common texture features include the Local Binary Pattern (LBP) (Heikkilä and Pietikäinen, 2006), the Local Ternary Pattern (LTP) (Liao et al., 2010) and a number of variants (Bilodeau et al., 2013; Davarpanah et al., 2016). Since the LBP descriptor checks the relative difference between spatially neighboring pixels and not the absolute values of each pixel, illumination changes can be efficiently overcome. But texture features can produce false detections due to textures artificially generated by local illumination effects or noise and may fail to detect textureless foreground regions from a textureless background even when they have different intensity values.

The motion features provide temporal information and are usually obtained via optical flow to deal with irrelevant motion in the background. The main drawback of most of the optical flow algorithms is that they are computationally slow, although some faster algorithms have been proposed (Bao et al., 2014).

A proper combination of visual features (color, texture, motion) modeling temporal and spatial pixel variations can improve performance, if the employed features are uncorrelated, making them suitable to address multiple challenges (Zhang et al., 2011; Han and Davis, 2012; Li et al., 2004). The most common approach is to combine two features, and one of them is usually color. Examples are color-gradient (Holtzhausen et al., 2015), color-texture (Chua et al., 2012) or color-motion (Wang and Suter, 2007; Martins et al., 2016).

Another aspect that should be taken into account is the selection of the size of the picture element used to model the background. Pixel-based approaches only use the current pixel value (Stauffer and Grimson, 1999), whereas the block-based (Yang et al., 2016) or region-based (Varadarajan et al., 2015) ones combine its neighboring pixels according to the spatial proximity. Pixel-based modeling and comparison is usually faster and enables a pixel-based precision, but it is less robust to noise than block-based or region-based approaches.

2.1.2. Background modeling

In background subtraction, the selected feature or features are compared against a background model to be classified as either foreground or background. To achieve this, an accurate and up-to-date background model has to be initialized and maintained. The initialization aspect of background modeling is a topic that has emerged recently (Jodoin et al., 2017) but most of the methods described in the literature focus on the representation and the adaption issues of background modeling. The most used approaches to model the background are based on statistical models.

Parametric models. Gaussian Mixture Model (GMM), or Mixture of Gaussians (MoG), has been well explored and it is probably the most popular strategy to model the background. It is a parametric model capable of handling several modes in a pixel value (Stauffer and Grimson, 1999). It can deal with slow lighting changes, periodical motion in the cluttered background, slow-moving objects, long-term scene changes, and camera noise. It is widely used due to its computational efficiency and good performance in a large number of applications. These traits inspired many improvements and extensions (Zivkovic, 2004; Lee, 2005; Martins et al., 2017). Among the most popular and widely used methods is the adaptive GMM background model proposed in Zivkovic (2004), which achieves increased performance in multi-modal backgrounds, without penalizing computational performance, by adaptively determining the number of Gaussians for each pixel on-line and, in this way, automatically adapt to the scene. The performance of this method was significantly boosted by the solution presented in Martins et al. (2017, 2018) which explores a novel classification mechanism that combines color space discrimination capabilities with hysteresis and a dynamic learning rate for background model update. The complexity of the method is kept low, proving its suitability for real-time applications.

The method proposed in Wang et al. (2014a) uses GMM to model the background scene and, besides, single gaussians are employed for foreground modeling, resulting in more reliable change detections. However, in the presence of drastic illumination changes, the method often fails to distinguish changes due to object motion from light changes, producing a large number of pixels misclassified as foreground. To tackle dynamic background movement, the method proposed in Chen et al. (2015) applies GMM in a local region, with models for foreground and background being learned with pixel-wise GMM. To label a given pixel, a region around the pixel is searched for the GMM that shows the highest probability for the center pixel. On the challenging scenario of nighttime videos, the method becomes very noisy and, at the same time, fails to detect camouflaged objects as it relies only on color features.

Most approaches are designed assuming a static camera and fail when used with moving cameras. An edge-based background estimation method is proposed in Allebosch et al. (2015) to cope with camera viewpoint changes. It is based on an edge descriptor, calculated from Local Ternary Patterns (LTPs), that is compared with a GMM background model that uses a dynamic learning rate. Optical flow is used to detect and compensate for camera viewpoint changes automatically. The performance of this method was further improved in Allebosch et al. (2016) by the addition of color information as an extra feature. These methods have shown not only to be able to deal with camera motion but also to be more stable in difficult illumination conditions, such as in nighttime videos, than other competing methods. Although not designed specifically for nighttime videos, the edge descriptor used is calculated from the LTP texture descriptor, that is an illumination-invariant feature, resulting in increased robustness to illumination-variations. This is confirmed as they were the best performing methods, among the unsupervised methods, for both the Pan-Tilt-Zoom (PTZ) and Night Videos (NV) categories in the CDnet 2014 benchmark (ChangeDetection.NET, 2014) before the submission of COLBMOG to CDnet.

Non-parametric models. Non-parametric kernel density estimation approaches do not assume any underlying distribution and determine a density function directly from the data (Elgammal et al., 2002). These methods avoid the difficulty of identifying the appropriate shape of the p.d.f. and can deal with fast changes in the background. However, they are computationally heavy and have large memory requirements. Improvements have been proposed to overcome these problems such as Zivkovic and van der Heijden (2006) and Tanaka et al. (2007).

Another non-parametric strategy that has been successfully used to model background pixels is sample consensus, proposed in Wang and Suter (2007). Sample consensus determines if a sample should be classified as foreground or background by comparing its current value to a history of recent values. This scheme has been refined, and many variants have been proposed. The method proposed in Barnich and Droogenbroeck (2011) uses color features (RGB) and randomly selects which values to substitute from the background model. It is faster but it cannot address dynamic backgrounds and noise efficiently. In St-Charles et al. (2015) color and texture features, called local binary similarity patterns (LBSP) (Bilodeau et al., 2013), are used along with a feedback mechanism to continuously improve the pixel's modeling. The method shows a very good performance in many scenarios, including illumination variations and shadows, but its performance decreases substantially in the challenging scenario of nighttime videos. A similar method, proposed in St-Charles et al. (2016), works with codewords called "background words" that combine pixel intensities, LBSP features and temporal features. All thresholds are dynamically updated in a feedback loop using a measure of the background dynamics. The method is sensitive to noise and color variations in low contrast background regions, such as the ones found in nighttime videos. Most of these methods have large memory requirements. A weight-sample-based method, proposed recently in Jiang and Lu (2018), employs the notion of "weight" instead of a "consensus" to build the background model. Detection accuracy is improved by assigning variable weights to a few samples. To rapidly adapt to changing scenarios, a minimum-weight update policy is proposed to replace the most inefficient sample instead of the oldest sample or a random sample. An adaptive feedback technique is also incorporated. However, its performance in nighttime videos decreases significantly.

Other approaches. Among the many other approaches that have been proposed in the literature, subspace learning is a family of background modeling methods that aims at modeling the background at the frame-level while reducing dimension significantly. One of the most used approaches in this category is eigenvalue decomposition that uses Principal Component Analysis to determine the background from the most descriptive eigenvectors (Oliver et al., 2000).

Neural network-based solutions (Culibrk et al., 2007; Maddalena et al., 2008; López-Rubio et al., 2011; Gregorio and Giordano, 2014) have received considerable attention and, lately, some researchers have applied deep neural networks (DNN) to the learning method for the maintenance of the background model (Porikli et al., 2016). These methods can model a wide range of variations in its layer structure and thus can cope with the great variability of real-world outdoor scenarios. However, most methods require a human intervention (Wang et al., 2017) and, consequently, cannot be considered unsupervised methods. In Braham and Droogenbroeck (2016), the authors propose the use of an existing background subtraction algorithm for the generation of a scene-specific dataset for training, but the performance is upper bounded by the classification performance of the dataset generator.

An excellent review of DNN-based approaches for detection of moving objects in video taken by a static camera can be found in Bouwmans et al. (2019). Most of the methods are based on Convolutional Neural Networks (CNN) or GAN (Generative Adversarial Networks). Both types need to be trained on a subset of labeled frames of the scene. CNN-based BGS yield very good results when manual annotation is provided but dropping to accuracies similar to unsupervised methods when using annotations provided by these methods. Also, generalization on unseen scenes is compromised in general. GAN-based BGS methods, however show a better generalization behavior when testing over unseen scenes (Zheng et al., 2019), probably due to the generative part of the GAN. So, as concluded in Bouwmans et al. (2019), the gap of performance obtained by DNNs based methods is essentially due to their supervised nature. In addition, their current computation times are too slow to be currently employed in real applications without a GPU card.

Table 1
Average *F-measure* across the overall set of videos and for the "Night Videos" category for some of the best performing unsupervised state-of-the-art methods.

Method	Overall	Night videos
SemanticBGS[a] (Braham et al., 2017)	0.7892	0.5014
IUTIS-5 (Bianco et al., 2017)	0.7717	0.5290
SWCD (Isik et al., 2018)	0.7583	0.5807
WisenetMD (Lee et al., 2018)	0.7535	0.5701
SharedModel (Chen et al., 2015)	0.7474	0.5419
WeSamBE (Jiang and Lu, 2018)	0.7446	0.5929
SuBSENSE (St-Charles et al., 2015)	0.7408	0.5599
C-EFIC (Allebosch et al., 2016)	0.7307	0.6677
EFIC (Allebosch et al., 2015)	0.7088	0.6548

[a]This method uses a pre-trained semantic network on 150 object classes of ADE20K dataset that include 12 CDnet relevant foreground classes. In this sense this is a borderline approach between unsupervised and supervised methods.

2.2. Motivation for our work

In spite of the number of change detection algorithms proposed in the last decades, with many of them performing very well on some types of videos, no single algorithm has proved to be able to deal with all the challenges in a robust way. Many approaches perform well under specific conditions, but the performance decreases significantly whenever one of the underlying assumptions is violated. This fact is not only well documented in the literature (Bouwmans, 2014; Sobral and Vacavant, 2014; Bouwmans, 2011; Brutzer et al., 2011; Benezeth et al., 2010; Bouwmans and Zahzah, 2014; Elhabian et al., 2008; Wang et al., 2014b) but can be easily verified in the ChangeDetection.net (CDnet) benchmark (ChangeDetection.NET, 2014), where change detection algorithms are evaluated on a common dataset composed of different types of videos and classified according to their performance. In this dataset, the video sequences are grouped into categories, and each category poses different challenges to the change detection algorithm (e.g., camera motion, nighttime lighting, dynamic background, etc.). No single algorithm that is able to manage all the challenges successfully. Instead, different algorithms are best suited to different problems. The Night Videos (NV) category in the CDnet 2014 dataset has proven to be one of the most difficult categories (Wang et al., 2014b), with most methods showing poor performance. As shown in Table 1, the learning of the background and foreground detection by the top-ranked state-of-the-art BS methods, critically fails in these scenes.

To cope with the variability of real-world videos, algorithms are becoming increasingly complex and thus computationally expensive. An alternative to improve performance without penalizing the complexity could be to combine state-of-the-art algorithms properly. The problem is how to choose the suitable algorithms to combine and what combination strategy to apply, preferably, in an automatic way. This approach has already proven to be successful (Wang et al., 2014b; Bianco et al., 2017).

In Wang et al. (2014b) the authors reported the overall performance of 14 methods evaluated for the IEEE Change Detection Workshop 2014 using the CDnet 2014 dataset. They also report the results obtained after combining all the methods (Majority Vote-all) and the top 3 methods (Majority Vote-3). Results were combined using a pixel-based majority voting. As reported, even by combining basic methods, Majority Vote-all outperforms every method except the top 2 methods, while Majority Vote-3 outperforms every other method. The same conclusion is reported in Goyette et al. (2014) for the methods evaluated using the CDnet 2012 dataset.

Unlike other fusion-based algorithms, which are not able to perform automatic algorithm selection, in Bianco et al. (2017) the authors exploit Genetic Programming to automatically select the best algorithms, combine them in different ways and execute the most suitable post-processing operations. The proposed solutions, termed IUTIS (In Unity There Is Strength), were compared against the methods evaluated for the IEEE Change Detection Workshop 2014 (Wang et al., 2014b). Results demonstrate that the proposed solutions outperform all the considered single algorithms. This is a strong indication that no single method decisively outperforms all other ones for all the possible scenarios. Moreover, it shows that these different methods seem to be complementary.

Therefore, instead of investing in the development of very flexible but complex algorithms that aim to cope with all the possible scenarios, an alternative is to develop focussed and robust methods to deal with specific scenarios. Thus, COLBMOG is proposed as a candidate method to be included in a system in combination with other state-of-the-art methods and to be chosen as the preferred method for nighttime outdoor scenarios.

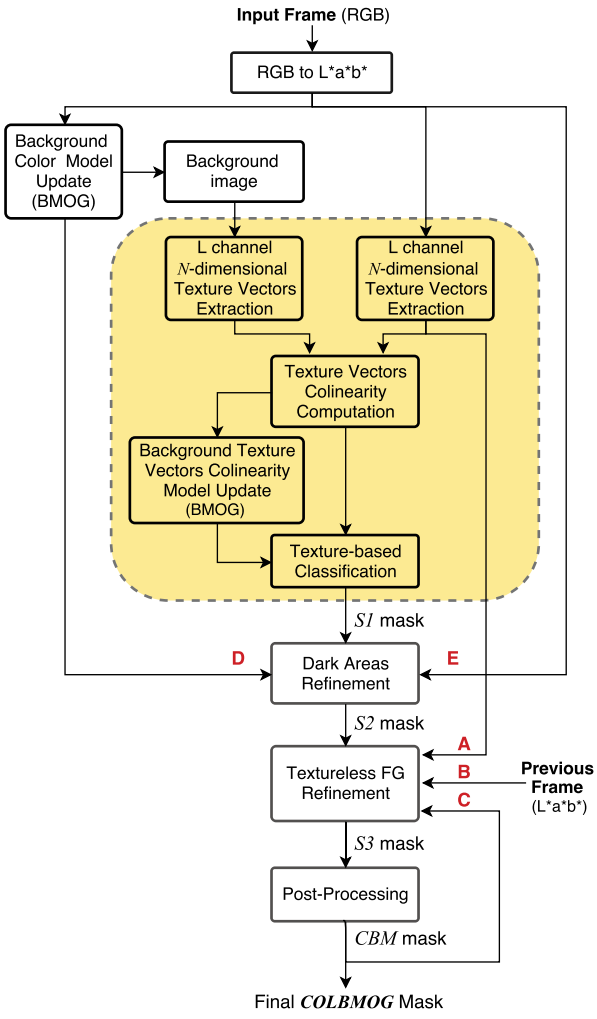

Fig. 1. COLBMOG block diagram.

3. The COLBMOG method

3.1. Overview

The proposed method is based on a local texture feature integrated with a parametric background model. The block diagram presented in Fig. 1 gives an overview of the method. Given an input frame, the algorithm returns a binary segmentation mask, $S1$, where the pixels whose associated texture vectors match the background image texture are classified as background (BG), and those that do not match are considered foreground (FG). The background color model adopted to generate the reference background image uses separate channels for luminance and chrominance, namely CIE L*a*b*. The local texture

features are extracted from the luminance channel and the chrominance channels are later used for dark-areas refinement. The main component of the method is the texture-based segmentation, whose central modules are shaded in Fig. 1 and are described in the following sections. The algorithm uses an N-dimensional vector, described in Section 3.2, as the feature representative of the local texture associated with each pixel. The collinearity between the corresponding texture vectors in the input frame and the background image is taken as the measure of similarity between both textures and is used to decide on the pixel classification as BG/FG. A model of the collinearity between texture vectors of background pixels in successive frames based on a Mixture of Gaussians (MoG) is created and updated at every frame using a fast algorithm. When the computed collinearity between the corresponding texture vectors in the input frame and the background image does not fit this model, the pixel is considered as belonging to a foreground object and is classified as FG. The binary segmentation mask obtained is named $S1$. Section 3.3 details this process.

The background frame used as a reference for the texture matching process is provided by a background color model that is updated every frame. Although other models could be used, the color model used in this approach is a robust and computationally efficient "Boosted MOG" algorithm, abbreviated as BMOG, proposed in Martins et al. (2017) and described in detail in Martins et al. (2018). The results presented in Martins et al. (2017, 2018) show that, for nighttime videos, it achieves very competitive results when compared to other MOG-based algorithms (Zivkovic, 2004; Varadarajan et al., 2015; ChangeDetection.NET, 2014).

The collinearity between texture vectors when the luminance is very low becomes very unstable. Hence, in really very dark areas of the background image, the segmentation mask needs to be improved. To this end, the algorithm disregards the texture-based classification and relies upon color information for the classification. Thus, in this situation, a classification based on color is favored, as described in Section 3.4. The resulting mask is named $S2$.

The mask obtained so far, is further refined to deal with large textureless foreground objects. The algorithm may fail to detect large textureless foreground objects when they move in front of textureless background regions. Therefore, a complementary mechanism has been introduced to tackle this problem, relying upon the measure of collinearity between the corresponding texture vectors in the current input frame and the previous input frame, as described in Section 3.5. This refined mask is referred to as $S3$.

Finally, a post-processing step consisting of median filtering and morphological operations allows the elimination of very small blobs and filling of closed contours. The output of this module, *CBM* mask, is the final COLBMOG mask.

3.2. Local texture modeling

The presence of moving objects in an image causes local intensity changes. Thus, we propose a representation of the local texture at the pixel neighborhood by looking at the values of the luminance of the pixel and its N-1 neighbors as a vector in an N-dimensional space. The N-1 neighboring pixels are chosen according to a predefined pattern in the surrounding area. This surrounding area must be small to be discriminative at every location. On the other hand, the number of pixels and the pattern must be chosen including a number of pixels large enough to capture the texture but without penalizing too much computational efficiency. As a compromise between these requirements, the algorithm uses a 9-dimensional vector to represent the local texture associated with each pixel, with the 8 neighboring pixels selected from a 5×5 surrounding area and chosen according to the pattern specified in Fig. 2. Different patterns with the same number of pixels chosen within the same area were tested, but leading to inferior results. The 9 vector elements are the values of the luminance of each of these pixels. Other texture representations can be plugged in here, but we have chosen a simple, effective and computationally fast one.

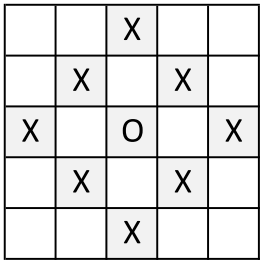

Fig. 2. Pattern of 8 neighboring pixels chosen as representative of the local texture at pixel o.

As a pre-processing step for the texture vectors extraction, a 3×3 Gaussian Low Pass Filter is applied to each of the images before taking the texture vectors.

3.3. Classification based on texture vectors collinearity

3.3.1. Texture vectors collinearity computation

The uniform illumination variations result in a texture vector that is collinear with the reference background texture vector and can be discarded with a simple collinearity test. Thus, the similarity between the texture at pixel j in the input frame and the texture at pixel j in the background image is defined as the collinearity between the corresponding texture vectors, $\vec{x}_j$ and $\vec{b}_j$, respectively. The measure of the collinearity is defined as the angle between the texture vectors, $\theta(\vec{x}_j, \vec{b}_j)$, computed by applying Eq. (1),

$$\theta(\vec{x}_j, \vec{b}_j) = \cos^{-1}\left(\frac{\vec{x}_j \cdot \vec{b}_j}{\left\|\vec{x}_j\right\|\left\|\vec{b}_j\right\|}\right) \tag{1}$$

where $\vec{x}_j \cdot \vec{b}_j$ denotes the dot product between vectors $\vec{x}_j$ and $\vec{b}_j$, $\left\|\vec{x}_j\right\|$ and $\left\|\vec{b}_j\right\|$ are the magnitudes of vectors $\vec{x}_j$ and $\vec{b}_j$, N is the number of pixels included in the texture pattern and x_{ji} and b_{ji} are each of the N elements of vectors $\vec{x}_j$ and $\vec{b}_j$. As the dissimilarity between the local textures increases, the value of θ also increases. Note that due to non-negativity of vector elements, angles lie in the positive quadrant, so, the texture angle goes from 0, when textures are identical, to $\pi/2$ when textures are entirely different. For each pixel j in the input frame, the angle $\theta(\vec{x}_j, \vec{b}_j)$ is used to decide if the input pixel texture matches the corresponding background pixel texture or not, so, the classification method is more robust to illumination-induced variations because it uses only the direction of the vectors, and not their lengths.

This approach to compare the local texture in the current frame and the reference frame can be related to methods using correlation analysis since the Normalized Cross-Correlation (NCC) between vectors $\vec{x}_j$ and $\vec{b}_j$ is defined as the cosine of $\theta(\vec{x}_j, \vec{b}_j)$. Recently Boulmerka and Allili (2018) used multiscale correlation analysis between color image square blocks, using multiple window sizes, to compare the local structure between the current and the reference frames. This approach, combined with local color histogram matching, is used to model the spatial information. However, the method performs poorly for night videos. Our approach uses a simple and computationally fast feature extracted only from the luminance channel that proves to capture the local texture efficiently and to be robust to sudden illumination changes, providing a better model for the estimation of the foreground/background probabilities in nighttime videos.

3.3.2. Background texture vectors collinearity model

The collinearity between texture vectors of background pixels in successive frames exhibits a regular behavior, making it possible to be described by a statistical model. Thus, a multimodal temporal model

Fig. 3. Examples of original frame (left), reference background image using the weighted average of the mean of each Gaussian (center) and reference background image using the mean of the matched mode (right).

based on a Mixture of Gaussians is created and updated at every frame. The BMOG algorithm (Martins et al., 2017, 2018) is used to update this model and to determine those pixels whose texture most closely matches the background texture. This model can also accommodate noise. BMOG uses a conditional update mechanism where a dynamic learning rate is adapted independently for each pixel and depends on the change of classification decision. In this case, the model has only one channel, corresponding to the texture angle.

3.3.3. Texture-based classification

Global illumination variations result in a texture vector that is collinear with the reference background texture vector. Input texture vectors that significantly differ from the corresponding background texture vectors originate large values of θ. Therefore, when the computed collinearity between the corresponding texture vectors in the input frame and the reference background frame does not fit the collinearity background model, the pixel is considered as belonging to a foreground object.

Hence, for each Gaussian m in the mixture, if the squared Mahalanobis distance between the angle computed, $\theta(\vec{x}_j, \vec{b}_j)$, and the Gaussian mean is larger than an acceptable similarity threshold, the match is rejected. Thus, for each pixel j, the texture mask $S1$ is set according to (2)

$$S1_j = \begin{cases} \text{FG}, & \text{if } \dfrac{\left(\theta(\vec{x}_j,\vec{b}_j)-\mu_{j,m}\right)^2}{\sigma^2_{j,m}} > (T_{sim} \pm \delta_{T_{sim}}) \\ \text{BG}, & \text{if } \dfrac{\left(\theta(\vec{x}_j,\vec{b}_j)-\mu_{j,m}\right)^2}{\sigma^2_{j,m}} \leq (T_{sim} \pm \delta_{T_{sim}}) \end{cases} \tag{2}$$

where $\mu_{j,m}$ is the estimated Gaussian mean and $\sigma^2_{j,m}$ is the estimated Gaussian variance. As in BMOG, a hysteresis mechanism has been implemented to prevent noisy pixels whose angle distance is very close to the decision threshold, T_{sim}, from incorrectly changing the classification. Hence, depending on the classification of the same pixel in the previous frame, the threshold values in (2) are increased or decreased by $\delta_{T_{sim}}$, to make the change of classification more difficult. The values of T_{sim} and $\delta_{T_{sim}}$ had to be set according to the characteristics of the texture angles as they are clearly different from the default values in Martins et al. (2017, 2018) that were defined for color. These new values were set experimentally.

The texture-based approach improves the detection of camouflaged objects when their texture differs from the background texture and increases the robustness to illumination changes when all local luminance values suffer the same variation over time. This local texture feature has shown to be particularly discriminative when applied to the segmentation of videos captured at night that present difficult challenges like low-visibility of objects and very strong lights that cause very hard reflections, a problem that has to be dealt with in common nighttime surveillance applications.

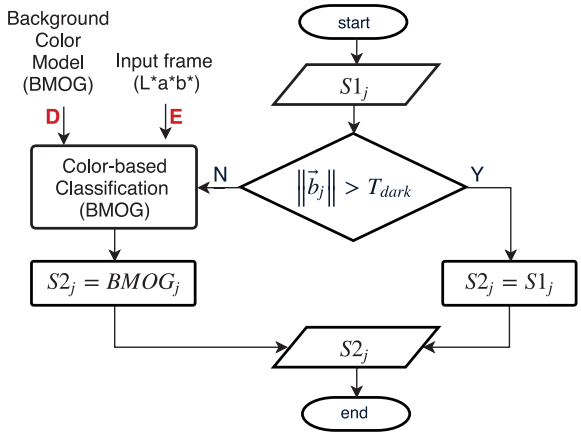

Fig. 4. Mechanism of classification of very dark areas of the image. D and E inputs are marked in Fig. 1.

3.3.4. Background image

An up-to-date reference background frame needs to be generated and updated to be used in the texture matching process. To that end, a color model of the background based on a Mixture of Gaussians is created and maintained using the BMOG algorithm (Martins et al., 2017, 2018).

The background image is obtained from the average background statistics. For each pixel, the weighted average of the mean of each Gaussian in the mixture is computed. This averaging results in an image that, sometimes, looks blurry but is very stable over time. Another option, which might seem more obvious, would be to choose the mean of the matched mode. However, the averaging operation has proved to provide a more reliable background image, particularly when pixels are persistently misclassified as background and become quickly incorporated into the background model. In this case, false textures are induced in the background image, as illustrated in Fig. 3, leading to an overall lower performance of the system.

Fig. 3 shows the original frame on the left, the reference background image using the weighted average of the mean of each Gaussian on the center and the reference background image using the mean of the matched mode on the right, for frame 860 of video *fluidHighway* on the top row and frame 1305 of video *winterStreet* on the bottom row. Both videos belong to the CDnet dataset.

3.4. Very dark areas refinement

The collinearity between texture vectors in really very dark areas of the images, where the elements of the vectors have very low values,

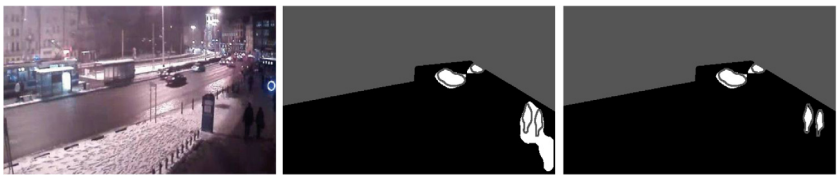

Fig. 5. From left to right: original frame, FG mask with $T_{dark} = 0.0$ and FG mask with $T_{dark} = 45$, for frame 1732 of video *tramStation*.

becomes very unstable since small differences in the luminance may lead to large values of computed angles. As image noise is much more noticeable in very dark areas, relying on the angle between the texture vectors is more prone to errors. In this case, the color-based classification becomes more reliable. As we maintain a model of the background color using the BMOG algorithm, the color-based classification rule implemented is the same proposed in Martins et al. (2017, 2018). The color-based segmentation mask provided by the BMOG algorithm is named $BMOG$ mask.

For each pixel, a validity test is performed based on the magnitude of the texture vectors in the reference background frame. For pixel j, if this magnitude is below a pre-defined threshold, T_{dark}, the classification based on the texture, $S1_j$, is discarded and replaced by the classification based on color, $BMOG_j$, as depicted in Fig. 4.

Hence, a refined $S2$ mask is obtained by setting each pixel j according to (3)

$$S2_j = \begin{cases} S1_j, & \text{if } \left\|\vec{b}_j\right\| > T_{dark} \\ BMOG_j, & \text{if } \left\|\vec{b}_j\right\| \leq T_{dark} \end{cases} \tag{3}$$

where $\vec{b}_j$ is the texture vector associated with pixel j in the reference background frame. The threshold T_{dark} was set experimentally to 45. It must be highlighted that these pixels have a low impact on the overall result because, in general, the regions of interest do not include a large number of these pixels. Fig. 5, from left to right, shows the original frame, the FG mask without very dark areas refinement ($T_{dark} = 0$) and the FG mask with very dark areas refinement ($T_{dark} = 45$), for frame 1732 of video *tramStation* from the CDnet dataset.

3.5. Textureless foreground objects refinement

It is well known that texture features are not useful on textureless regions. Thus, problems may occur when large textureless foreground objects move in front of textureless background regions. In this situation, the texture vector associated with the foreground pixel and the texture vector associated with the background pixel may have similar directions, even if they have very different magnitudes, and the algorithm misclassifies the FG pixel as BG. However, the texture vectors associated with the pixels on the object contour will include pixels both from the foreground object and from the background resulting in a vector that is not collinear with the texture vector of the corresponding pixel in the background image and so is detected as a FG pixel. If all the pixels in the object contour are detected as FG, even if the inside pixels are not, the closed contour generated will later be filled in a post-processing step. However, if the algorithm fails to detect all the pixels in the object contour as FG, large missing regions may appear in the foreground object, leading to a highly fragmented FG mask. A texture-driven filling mechanism relying upon the measure of collinearity between the texture vectors in the current input frame and the previous input frame was introduced to overcome this problem, as depicted in Fig. 6.

If a pixel is classified as BG and the same pixel was classified as FG in the previous frame, the collinearity between the associated texture vectors from the current input frame and the previous input frame is computed. If it is below a pre-defined threshold, T_{tl}, the pixel is assumed to still belong to the foreground object and the classification

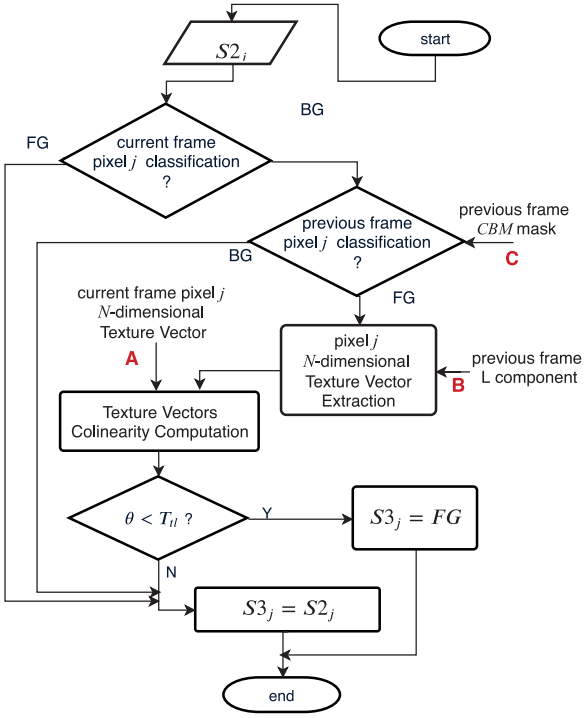

Fig. 6. Mechanism of classification of large textureless foreground objects. A, B and C inputs are marked in Fig. 1.

from the previous frame, FG, is retained. Therefore, for each pixel j, the mask $S3$ is set according to (4)

$$S3_j = \begin{cases} \text{FG}, & \text{if } S2_j = \text{BG} \wedge CBM_j^{t-1} = \text{FG} \\ & \quad \wedge\, \theta(\vec{x}_j^t, \vec{x}_j^{t-1}) < T_{tl} \\ S2_j, & \text{otherwise} \end{cases} \tag{4}$$

where $\vec{x}_j^t$ and $\vec{x}_j^{t-1}$ are the texture vectors at pixel j in the current and previous frames, respectively, and CBM^{t-1} is the final mask for the previous frame. The threshold T_{tl} was set experimentally to 0.65. Fig. 7, from left to right, shows the original frame, the FG mask without textureless foreground objects refinement ($T_{tl} = 0.0$) and the FG mask with textureless foreground objects refinement ($T_{tl} = 0.65$), for frame 1427 of video *winterStreet* from the CDnet dataset.

3.6. Post-processing

The refined $S3$ mask obtained in the previous step is finally post-processed with a median filter, followed by morphological close, filling of closed contours, and morphological erosion, thus eliminating irrelevant blobs/holes from the mask. At the end of this process, the final COLBMOG mask, CBM, is obtained.

Fig. 7. From left to right: original frame, FG mask with $T_{tl} = 0.0$ and FG mask with $T_{tl} = 0.65$, for frame 1427 of video *winterStreet*.

4. Experimental setup

4.1. Dataset

The scarcity of publicly available datasets including real videos captured by night, with the corresponding ground truth segmentation of foreground objects, makes it difficult to perform exhaustive experiments, with the possibility of being replicated, to assess the performance of algorithms in nighttime scenarios. Although a number of datasets dedicated to the evaluation of moving objects detection have been recently proposed (Cuevas et al., 2016), the absence of nighttime videos is a common issue. The SABS (Stuttgart Artificial Background Subtraction) dataset (Brutzer et al., 2011) is an artificial dataset for the evaluation of background models that includes a single nighttime video. The video simulates a nighttime urban surveillance context but it is not very realistic, and its interest is decaying as a number of real videos with real situations have appeared. The BMC (Background Models Challenge 2012) dataset (Vacavant et al., 2012) dataset includes nine real videos but only one of them, Video 004 — *Rabbit in the night*, is an outdoor nighttime video acquired in a video-surveillance context, with a lot of noise on top of cast shadows and sudden light changes in the scene. However, ground truth data exists only for some frames and is encrypted. Consequently, the performance of the algorithms has to be evaluated using the "BMC Wizard" software (BMC, 2012) to compute the average quality measures making it difficult to perform a detailed analysis of the results. The CDnet 2014 dataset (ChangeDetection.NET, 2014; Wang et al., 2014b) is the first public dataset that includes a "Night Videos" category, made up of six videos consisting of real outdoor scenes. These videos consist of nighttime urban traffic surveillance videos, suffering from photon shot noise, compression artifacts, camouflaged objects, shadows and glare effects from car headlights that must all be handled simultaneously. The main challenge is really to deal with low-visibility of vehicles and their very strong headlights that cause halos and reflections on the street. The poor illumination causes numerous false negatives while strong light reflections cause systematic false positives. This category has proven to be one of the most difficult categories (Wang et al., 2014b), as already shown in Table 1.

The results reported here were mainly conducted using the CDnet videos. Testing has been performed using the ground truth (GT) segmentation provided along with the videos at the CDnet (ChangeDetection.NET, 2014) site. Pixels in the mask may have one of 5 labels: *Moving*, corresponding to foreground pixels; *Static*, corresponding to background pixels; *Shadow* corresponding to moving shadows; *Non-ROI* corresponding to regions outside the ROI; *Unknown* corresponding to pixels whose status is unclear. For evaluation, pixels classified as *Shadow* in the GT masks are considered as *Static* and pixels classified as *Non-ROI* and *Unknown* are discarded. To facilitate the visual comparison, *Non-ROI* or *Unknown* pixels were artificially marked gray in the computed masks presented.

These experiments involved the generation of all masks for our method and its submission to the CDnet site to be evaluated and ranked. In a second step, we compared COLBMOG with the top-ranked unsupervised methods in this category, C-EFIC (Allebosch et al., 2016) and EFIC (Allebosch et al., 2015), using the results publicly reported in the CDnet site (ChangeDetection.NET, 2014) for the 2014 dataset for all the three methods. Methods published more recently, e.g., WeSamBE (Jiang and Lu, 2018) and SWCD (Isik et al., 2018), underperform in this specific category and therefore are not considered. As required by CDnet, only one set of parameters was used for all the videos. The default values of BMOG (Martins et al., 2017, 2018) were used for the background color model. These values were determined using the complete CDnet dataset that includes a wide variety of camera-captured videos. For the background texture vectors collinearity model, a one-channel model, we set T_{sim}=1.4 and δ_{Tsim}=0.95. These default values were determined empirically and worked well for different scenarios as demonstrated by the results obtained with videos that incorporate a wide range of challenges.

Table 2
Average metrics for each of the videos and across the overall set of videos for COLBMOG.

Video	Re	Sp	FPR	FNR	PWC	Pr	F-measure
bridgeEntry	0.6293	0.9970	0.0030	0.3707	1.0351	0.8098	0.7082
busyBoulvard	0.7586	0.9878	0.0122	0.2414	3.1627	0.8517	0.8024
fluidHighway	0.7836	0.9917	0.0083	0.2164	1.1904	0.6277	0.6970
streetCornerAtNight	0.8942	0.9976	0.0024	0.1058	0.3043	0.7001	0.7853
tramStation	0.9434	0.9833	0.0167	0.0566	1.8031	0.6652	0.7802
winterStreet	0.8190	0.9762	0.0238	0.1810	3.4657	0.7177	0.7650
Average	0.8047	0.9889	0.0111	0.1953	1.8269	0.7287	0.7564
St. Dev.	0.1103	0.0083	0.0083	0.1103	1.2507	0.0859	0.0435

Table 3
Average metrics across the overall set of videos for COLBMOG, C-EFIC and EFIC.

Metric	EFIC	C-EFIC	COLBMOG
Re	0.6704	0.7223	**0.8047**
Sp	**0.9893**	0.9866	0.9889
FPR	**0.0107**	0.0134	0.0111
FNR	0.3296	0.2777	**0.1953**
PWC	2.5739	2.5899	**1.8269**
Pr	0.6869	0.6636	**0.7287**
F-measure	0.6548	0.6677	**0.7564**

4.2. Evaluation metrics

Several methods have been proposed for the objective evaluation of segmentation quality with different metrics typically conveying different types of information. In this work, we used two approaches to evaluate the contribution of the proposed segmentation method.

Considering background/foreground segmentation as a classification process, the following well-known seven metrics based on the number of correctly and incorrectly classified pixels are often used to rank background subtraction methods (Wang et al., 2014b; Goyette et al., 2012): *Recall* (Re), *Specificity* (Sp), *False Positive Rate* (FPR), *False Negative Rate* (FNR), *Percentage of Wrong Classifications* (PWC), *Precision* (Pr) and *F-measure*. With these metrics we can evaluate and compare the average performance of the algorithms. We assessed the proposed method over each video by computing these metrics, followed by a category-average metric. In our comparisons, the *F-measure* was used as the main indicator of performance since, as reported in Wang et al. (2014b) and Goyette et al. (2012), it is considered a well-balanced metric in this context. In the CDnet benchmark, besides the ranking

Table 4
Average *F-measure* for each of the videos and across the overall set of videos for COLBMOG, C-EFIC and EFIC.

Video	EFIC	C-EFIC	COLBMOG
bridgeEntry	0.6240	0.6500	**0.7082**
busyBoulvard	0.4772	0.5261	**0.8024**
fluidHighway	0.5614	0.5880	**0.6970**
streetCornerAtNight	0.7596	0.7138	**0.7853**
tramStation	0.8077	**0.8190**	0.7802
winterStreet	0.6990	0.7095	**0.7650**
Average	0.6548	0.6677	**0.7564**
St. Dev.	0.1245	0.1034	**0.0435**

Table 5
Average *F-Measure* using different variants of the GMM as background model for the collinearity of texture vectors, namely with/without hysteresis in the classification process and with dynamic/constant learning rate for the model's update.

GMM used as background model for the collinearity of texture vectors	Avg. *F-Measure*
dynamic learning rate + hysteresis	0.7193
constant learning rate + hysteresis	0.7115
dynamic learning rate + no hysteresis	0.7041
constant learning rate + no hysteresis	0.7028

by each of these metrics, an average ranking for the category is also computed.

Considering the image segmentation as a partition, a metric based on the normalized symmetric distance between partitions, $dsym$, was proposed in Cardoso and Corte-Real (2005) and Cardoso et al. (2009) to evaluate the quality of an image segmentation by computing the error measures for each frame, thus revealing the temporal evolution of the dissimilarity over the video sequence. This metric has shown to be consistent with the subjective evaluation that a human observer would make by direct visualization of the segmentation partitions (Cardoso and Corte-Real, 2005). For each video, the distance between the ground truth segmentation (available from CDnet only for the first half of the videos) and the segmentation produced by each method was calculated for each of the frames. This allowed us to assess the performance of the different methods along each video timeline in an objective way, showing how the error is distributed over time, and identifying the points of failure.

5. Analysis of results and discussion

Table 2 shows the average values for the first set of metrics for each video and across the overall set of videos for COLBMOG. Table 3 shows the average values of the same metrics across the overall set of videos for COLBMOG and the two top-rank methods C-EFIC and EFIC. Best scores are in bold. COLBMOG ranks first in the CDnet "Night Videos" benchmark for the unsupervised methods. It ranks first not only in the *F-measure* ranking but also in the average ranking for the category.

The top rank methods, "FgSegNet v2" (Lim and Keles, 2018c), "FgSegNet S" (Lim and Keles, 2018b), "FgSegNet" (Lim and Keles, 2018a), "BSPVGAN" (Zheng et al., 2019), "Cascade CNN" (Wang et al., 2017) and "BSGAN" (Zhen et al., 2018), are all supervised methods. These methods are based on deep neural networks that not only determine the best values for the parameters but also the best features to use given the training data. Besides that, for the task of foreground segmentation, and due to the limited number of labeled datasets available, these algorithms are trained and tested on 2 partitions of the same sequences, which leads to poor generalization of the learnt deep architecture over unseen scenes. Nevertheless, Bayesian GAN, BSPVGAN have shown a good generalization performance when trained over the CDNet videos and tested over 3 different datasets. It should be highlighted that, despite being supervised methods, COLBMOG outperforms two of them, namely "BMN-BSN" (Mondéjar-Guerra et al., 2019) and "DeepBS" (Babaee et al., 2018).

In general, we can say that the COLBMOG approach is inherently more sensitive to relevant changes in the scene than the EFIC and C-EFIC approaches, as visible through the *Recall* and *Precision* scores.

Table 4 shows the average *F-measure* for each video and across the overall set of videos, along with the standard deviation across the overall set of videos. Using *F-measure* as an indicator of performance we can conclude that COLBMOG consistently outperforms C-EFIC and EFIC, with a 13.3% relative overall *F-Measure* improvement over the previous best method, C-EFIC, and 15.5% relative overall *F-Measure* improvement over EFIC. Only for the *tramStation* video, COLBMOG slightly underperforms the other methods due to lower performance in the segmentation of pedestrians in the very dark region at the bottom-right area of the images, where COLBMOG relies on the classification provided by the low complex color-based algorithm, BMOG, as explained in Section 3.4. It should be noted that our worst results were obtained for the *fluidHighway* video, a very low-quality video with very noticeable compression noise which induces false textures and has a strong impact in the representation of texture. However, even for this difficult scenario, COLBMOG still outperforms the other methods by a large margin. Both the *Recall* and the *Precision* scores of the proposed method contribute to an *F-Measure* score well above the other methods. The standard deviation of the *F-measure* across the overall set of videos is also significantly lower for COLBMOG, meaning a more consistent performance across different challenges. The results presented may be confirmed in the CDnet site (ChangeDetection.NET, 2014).

Although COLBMOG uses the BMOG model, the relevance of the value-added introduced by COLBMOG to the overall performance of the system is clear when we compare the *F-Measure* obtained for the Night Videos category by BMOG, 0.4982, with the one obtained by COLBMOG, 0.7564. An improvement in *F-Measure* of approximately 52%.

COLBMOG uses BMOG as a color background model and also as a background model for the collinearity of texture vectors. Another model, like the family of MOG methods, could have been used for these purposes. However, BMOG has already proved to outperform other methods, such as MOG (Stauffer and Grimson, 1999), MOG2 (Zivkovic, 2004) or RMOG (Varadarajan et al., 2015), as a color background model, particularly for nighttime videos (Martins et al., 2017, 2018). When using BMOG to provide a background model for the collinearity of texture vectors, only the dynamic learning rate mechanism and the classification with hysteresis are exploited, and not the chrominance-luminance separation because the model has only one channel, corresponding to the angle between texture vectors. The results obtained (for the first half of each video, for which the ground truth is publicly available) and presented in Table 5 show that the hysteresis mechanism has a higher impact on the segmentation results.

The evaluation of the results using the symmetric partition distance metric, $dsym$, provides an error value for each frame, thus showing how the different methods behave along each of the videos and where the new method improves the results. This analysis provides valuable information regarding the performance of the segmentation algorithm. As an example, Fig. 8 shows the evolution of the $dsym$ for the first half of the video *winterStreet*, for the three algorithms. Unlike the *F-measure*, this is an error measure and, therefore, a lower value means higher quality. As illustrated, all the algorithms tend to fail in the same frames, corresponding to the most difficult situations, and it is in these frames that COLBMOG achieves a significant improvement in the quality of the segmentation. This is evident looking at the shaded area in Fig. 8 where the difference in $dsym$ reaches the highest values between frames 1200 and 1290.

Fig. 9 shows the foreground masks generated by EFIC, C-EFIC and COLBMOG for frames 1056 and 1220 of video *winterStreet* with the misclassified pixels marked red. It is clear that the quality of the segmentation masks of the three methods is very similar for frame 1056, an "easy" frame, whereas for frame 1220, an "hard" frame,

Table 6
F-measure resulting from the one-scenery-out tests.

Video (first half)	COLBMOG[a]	One-scenery-out		One-scenery-out		One-scenery-out		EFIC	C-EFIC
		T_{sim}	*F-measure*	δ_{Tsim}	*F-measure*	T_{dark}	*F-measure*		
bridgeEntry	0.7241	1.4	0.7241	0.95	0.7241	45	0.7241	0.598	0.6183
busyBoulvard	0.7068	1.5	0.6889	0.9	0.7033	46	0.7055	0.4182	0.4729
fluidHighway	0.6707	1.3	0.6611	1.15	0.6560	45	0.6707	0.5775	0.5910
streetCornerAtNight	0.7352	1.3	0.7230	0.9	0.7343	45	0.7352	0.6705	0.6450
tramStation	0.7519	1.3	0.7394	1.15	0.7276	44	0.7511	0.7922	0.7937
winterStreet	0.7268	1.4	0.7268	0.9	0.7258	45	0.7268	0.6077	0.6348
Average	**0.7193**		**0.7106**		**0.7119**		**0.7189**	0.6107	0.6260

[a]For COLBMOG: $T_{sim} = 1.4$, $\delta_{Tsim} = 0.95$, $T_{dark} = 45$.

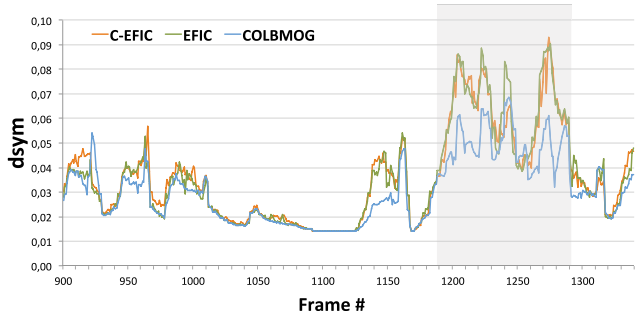

Fig. 8. Evolution of the $dsym$ from frame 900 to frame 1340 of video *winterStreet*.

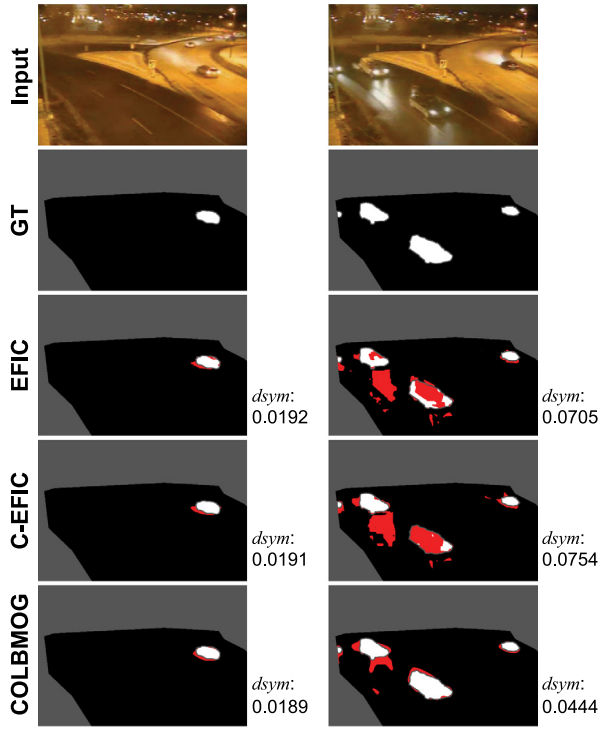

Fig. 9. Comparison of the foreground segmentation masks for COLBMOG, C-EFIC and EFIC, for frame 1056 (left) and frame 1220 (right) of video *winterStreet*. Misclassified pixels are marked red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

COLBMOG performs significantly better. This example reveals the superior performance of COLBMOG in the presence of camouflaged objects and strong reflections from the headlights on the street. From the $dsym$ plots this could be expected as it can be seen that all the three methods present similar values at frame 1056 but COLBMOG presents a much lower value at frame 1220.

A comparative example of foreground masks obtained with COLBMOG, C-EFIC and EFIC is illustrated in Fig. 10. These pictures show, from top to bottom, the original frame (Input), the ground truth (GT), and the foreground segmentation masks for EFIC, C-EFIC and COLBMOG. From left to right, frame 1662 of video *bridgeEntry*, frame 820 of video *busyBoulvard*, frame 443 of video *fluidHighway*, frame 2665 of video *streetCornerAtNight*, frame 1636 of video *tramStation* and frame 1278 of video *winterStreet*. These masks are all available at the CDnet site (ChangeDetection.NET, 2014). The pixels that are not labeled *Static* (BG) or *Moving* (FG) are not evaluated and are marked in gray. Misclassified pixels are marked red.

Although the CDnet category for Night Videos comprises a wide variety of challenges, it is also interesting to test the performance of COLBMOG in some of the other challenging scenarios, like shadows and dynamic background. In the first case, the average *F-measure* obtained for the first half of the daytime videos of the Shadow (SW) category is 0.8775, which is similar to C-EFIC (0.8778) and better than EFIC (0.8202). Dynamic backgrounds, an important challenge to overcome, are not present in the Night Videos and Shadow categories. For that reason, we have set the maximum number of gaussians to a low value to increase computational efficiency. In order to deal with a dynamic background, it is well known that a more flexible model would be needed, so we have set the maximum number of gaussians to 5, so we were able to reach higher *F-measure* averages for daytime videos of the Dynamic Background (DB) category, achieving 0.6532 — doing better than both C-EFIC (0.5627) and EFIC (0.5779).

The limitations of the BMC dataset did not allow an analysis as detailed as done for the CDnet dataset. Nevertheless, we evaluated our method using the "BMC Wizard" tool for Video 004 and obtained an *F-measure* of 0.9105. It cannot be compared against C-EFIC or EFIC, as there are no results published for this dataset. Anyway, the best *F-measure* obtained with this video, among the methods presented at the BMC workshop, is 0.904 (Yoshinaga et al., 2013) and a more recent work (Maddalena and Petrosino, 2014) achieves 0.8934 — however, we do not have results from these methods for the CDnet videos. Fig. 11 shows the foreground masks obtained with COLBMOG for frames 292 and 613.

The fact that these tests were performed on only seven videos (six from CDnet dataset and one from BMC dataset) may be considered a limitation of this research. However, the generation of a new dataset of nighttime videos with corresponding ground truth masks (labeled manually) would not allow the comparison with other methods for which we would not have the results. Thus, the validation of our results would not be possible.

The code implemented was not optimized for real-time performance. Nevertheless, the raw code[2] works at, approximately, 8.5 fps for 320 × 240 videos using an Intel Core i7 2 GHz processor with 16 GB 1333 MHz DDR3 and OS X Yosemite 10.10.5. An efficient implementation of the algorithm running in a faster processor would allow real-time processing.

[2] The code is available at url: https://github.com/mmartinspf/COLBMOG.

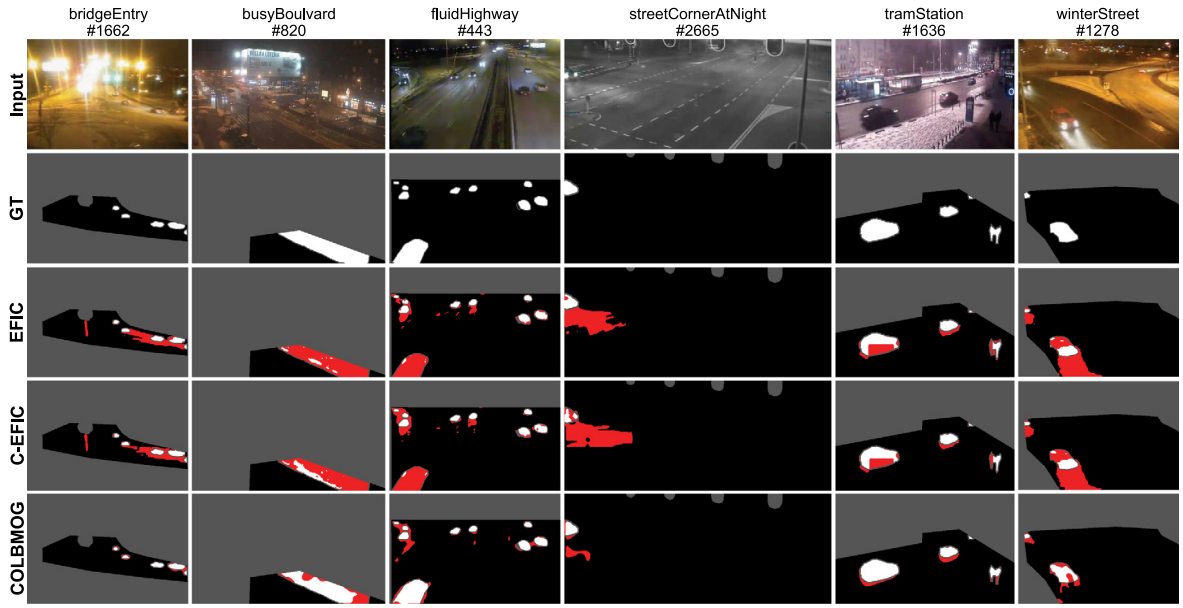

Fig. 10. Example of foreground masks for COLBMOG, C-EFIC and EFIC, and the corresponding ground truth (GT), for the night videos from the CDnet 2014 dataset. Misclassified pixels are marked red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

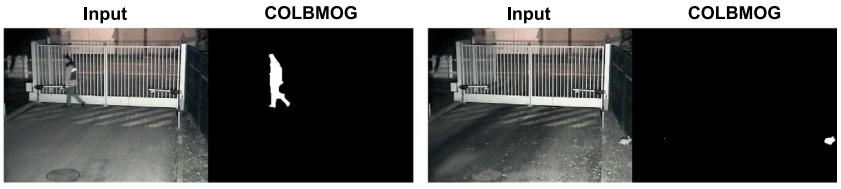

Fig. 11. Example of foreground masks for COLBMOG for frame 292 (left) and frame 613 (right) of video 004, *Rabbit in the night*, of BMC dataset.

Table 7
Results of a sensitivity analysis on the average *F-measure*. Values shown are percent deviations.

Parameter	$T = T_{sim}$	$T = \delta_{Tsim}$	$T = T_{dark}$	$T = T_{tl}$
0.8*T	−0.72	−0.38	−1.05	0.04
0.9*T	−0.25	−0.13	−0.29	0.03
1.1*T	−0.53	−0.10	−0.05	−0.02
1.2*T	−1.17	−0.22	−0.62	−0.04

5.1. Analysis of the robustness to the parameters' setting

The method has some parameters that have to be set in advance. As stated in Section 4, the settings for the BMOG color background model are the default values specified in (Martins et al. 2017; Martins et al. 2018), that were determined using the complete CDnet dataset that includes a wide variety of camera-captured videos, not only the NightVideos, so their nominal values are theoretically more robust than using just the NightVideos. The parameters for the texture vectors collinearity background model were set experimentally. In this section, we analyze how the performance of the proposed method is influenced by the setting of these parameters, namely, the similarity threshold, T_{sim}, the hysteresis parameter for the decision threshold, δ_{Tsim}, the very dark areas threshold, T_{dark}, and the textureless foreground objects threshold, T_{tl}. The robustness of the COLBMOG method to each of these thresholds was assessed in two different ways: (1) performing a one-scenery-out test and, (2) a sensitivity analysis. This study was conducted using the first half of each video from the CDnet NV category, for which the ground truth masks are publicly available.

The value of the similarity threshold between the collinearity of texture vectors from the input frame and the background model image, and the background model (a one-channel model), T_{sim}, was set as the rounded average of the best value for each video. The decision threshold for the classification is determined as $T_{sim} \pm \delta_{Tsim}$. The threshold in the "Very Dark Areas Refinement" module, T_{dark}, leads to the decision of using the segmentation mask generated by the color background model or the mask generated by the texture vectors collinearity model. The influence of each of these parameters was analyzed by a one-scenery-out test. In each experiment, the best value for five videos is set, and that value is used for testing over the remaining video. The process was repeated six times for each parameter tested. Table 6 shows the *F-Measure* obtained for each video in each scenario tested. The average *F-Measure* from the one-scenery-out tests is only slightly lower than the one obtained with the chosen value for all the videos. In the worst case, it decreases by 1.2%, meaning that their setting is not critical. And, more importantly, a 20% of variation of threshold values around the default ones, still produce systems better than the state-of-the-art for a large margin except for one video. This is particularly true for T_{dark} because "very dark areas" occur only in small areas of some videos.

The threshold in the "Textureless Foreground Objects Refinement" module was experimentally set, by conducting a subjective evaluation of the resulting masks, frame by frame, for the complete videos. The presence of large textureless objects is significant only in the second part of the videos, particularly in the *winterStreet* video, as depicted in Fig. 7. This new challenge does not affect the first half of the videos. However, the GT masks for the second part of the videos are not publicly available. Therefore, the contribution of this module to the overall performance of the method, and the determination of the value of the threshold T_{tl}, was based on the subjective evaluation, frame by frame, of the foreground masks produced. The negative impact in the results for the first part of the videos is negligible compared to the improvement provided for the second part. And its effectiveness

Table 8
F-measure obtained for different values of T_{sim}.

Video (first half)	T_{sim} 1.0	T_{sim} 1.1	T_{sim} 1.2	T_{sim} 1.3	T_{sim} 1.4	T_{sim} 1.5	T_{sim} 1.6	T_{sim} 1.7	T_{sim} 1.8
bridgeEntry	0.7019	0.7188	0.731	0.7288	0.7241	0.7156	0.7064	0.6962	0.6863
busyBoulvard	0.7036	0.7514	0.7479	0.7345	0.7068	0.6889	0.6754	0.6561	0.6424
fluidHighway	0.4889	0.6129	0.6479	0.6611	0.6707	0.6720	0.6729	0.6731	0.6727
streetCornerAtNight	0.6431	0.6960	0.7094	0.7230	0.7352	0.7444	0.7501	0.7556	0.7602
tramStation	0.5818	0.6858	0.7200	0.7394	0.7519	0.7602	0.7672	0.7725	0.7765
winterStreet	0.6465	0.7157	0.7283	0.7311	0.7268	0.7190	0.7111	0.7024	0.6955
Average	**0.6276**	**0.6968**	**0.7141**	**0.7197**	**0.7193**	**0.7167**	**0.7139**	**0.7093**	**0.7056**

Table 9
F-measure obtained for different values of δ_{Tsim}.

Video (first half)	δ_{Tsim} 0.5	δ_{Tsim} 0.75	δ_{Tsim} 0.85	δ_{Tsim} 0.90	δ_{Tsim} 0.95	δ_{Tsim} 1.0	δ_{Tsim} 1.05	δ_{Tsim} 1.15	δ_{Tsim} 1.25
bridgeEntry	0.7338	0.727	0.7256	0.7244	0.7241	0.7234	0.7207	0.7173	0.7166
busyBoulvard	0.6814	0.6932	0.6982	0.7033	0.7068	0.7097	0.7173	0.7388	0.7510
fluidHighway	0.6687	0.6736	0.6733	0.6720	0.6707	0.6685	0.6652	0.6560	0.6335
streetCornerAtNight	0.7237	0.7309	0.7332	0.7343	0.7352	0.7356	0.7356	0.7332	0.7322
tramStation	0.7681	0.7622	0.7577	0.7553	0.7519	0.7476	0.7429	0.7276	0.6992
winterStreet	0.6994	0.7096	0.7206	0.7258	0.7268	0.7295	0.7294	0.7335	0.7296
Average	**0.7125**	**0.7161**	**0.7181**	**0.7192**	**0.7193**	**0.7191**	**0.7185**	**0.7177**	**0.7104**

Table 10
F-measure obtained for different values of T_{dark}.

Video (first half)	T_{dark} 30	T_{dark} 36	T_{dark} 40	T_{dark} 43	T_{dark} 45	T_{dark} 48	T_{dark} 50	T_{dark} 55	T_{dark} 60
bridgeEntry	0.7241	0.7241	0.7241	0.7241	0.7241	0.7241	0.7241	0.7241	0.7241
busyBoulvard	0.7072	0.7072	0.7072	0.7073	0.7068	0.7051	0.7048	0.6774	0.6180
fluidHighway	0.6479	0.6626	0.6674	0.6702	0.6707	0.6707	0.6694	0.6651	0.6562
streetCornerAtNight	0.7352	0.7352	0.7352	0.7352	0.7352	0.7352	0.7352	0.7352	0.7349
tramStation	0.6808	0.7141	0.7407	0.7496	0.7519	0.7524	0.7523	0.7518	0.7516
winterStreet	0.7268	0.7268	0.7268	0.7268	0.7268	0.7268	0.7268	0.7268	0.7271
Average	**0.7037**	**0.7117**	**0.7169**	**0.7189**	**0.7193**	**0.7191**	**0.7188**	**0.7134**	**0.7020**

was validated by the results computed by CDnet for the overall videos, presented in Table 4. The *F-Measure* for the complete videos, computed by CDnet is 0.7564, while for the first half of the videos is 0.7193. For this reason, the one-scenery-out test for this threshold did not make sense. For the first half of the videos, only videos *busyBoulvard* and *winterStreet* benefit from this mechanism. In the case of video *winterStreet*, it improves the results for some textureless foreground objects. In the case of video *busyBoulvard*, this mechanism also revealed to be useful when dealing with stopped foreground objects, preventing FG pixels from being classified as BG.

Sensitivity analysis is often used to explore the influence of varying model parameters on the outputs of a simulation model, helping to identify those parameters that have a strong influence on the output, indicating which ones are most important. To perform the sensitivity analysis, we varied the values of the thresholds being analyzed by a specified percentage around their default value. We selected a variation of ±10% and ±20%. Table 7 shows the results of the sensitivity analysis. Values shown are percent deviations of average *F-measure*. When changing one threshold, all other thresholds were kept constant. Negative values indicate a decrease in performance. Results show that the obtained *F-measure* is relatively insensitive to small variations in these thresholds, and the model settings are "controlled".

Tables 8 to 10 present the results obtained from the sensitivity analysis of these thresholds.

6. Conclusion

This paper proposes a new method for the unsupervised segmentation of moving objects in nighttime video sequences. The proposed method outperforms the best state-of-the-art algorithms in the most complex situations faced in this kind of videos. The information obtained from a texture-based change detection method, using local texture modeling, complemented by a color-based change detection method, that explores the characteristics of color spaces that separate luminance from chrominance, greatly increases the overall detection accuracy in these scenarios. A detailed analysis of the experimental results has revealed that these improvements are more significant when the more difficult scenarios are faced, like challenging illumination conditions, and all the algorithms tend to fail. Results, obtained using the same set of parameters for all the videos, show that COLBMOG outperforms the state-of-the-art methods, ranking first in the CDnet "Night Videos" benchmark for the unsupervised methods and behaves well for other daytime scenarios. This makes it a serious candidate for the background subtraction task in nighttime applications.

CRediT authorship contribution statement

Isabel Martins: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing - original draft, Writing - review & editing. **Pedro Carvalho:** Conceptualization, Methodology, Validation, Writing - review & editing. **Luís Corte-Real:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision. **José Luis Alba-Castro:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by Xunta de Galicia and the European Union (European Regional Development Fund — ERDF) through the Consolidated Strategic Group atlanTTic (2016-2019) and by Xunta de Galicia through the Potential Growth Group 2018/60, and partially supported by project "Cooperative Holistic view on Internet and Content CHIC" POCI-01-0247-FEDER-024498, financed by COMPETE 2020, under Portugal 2020, and through the European Regional Development Fund (ERDF).

References

Allebosch, G., Deboeverie, F., Veelaert, P., Philips, W., 2015. EFIC: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints. In: Proc. 16th Int. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS 2015). pp. 130–141. http://dx.doi.org/10.1007/978-3-319-25903-1_12.

Allebosch, G., Van Hamme, D., Deboeverie, F., Veelaert, P., Philips, W., 2016. C-EFIC: Color and edge based foreground background segmentation with interior classification. In: Braz, J., Pettré, J., Richard, P., Linsen, L., Battiato, S., Imai, F. (Eds.), 10th Int. Joint Conf. Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2015), Revised Selected Papers. pp. 433–454. http://dx.doi.org/10.1007/978-3-319-29971-6_23.

Azab, M.M., Shedeed, H.A., Hussein, A.S., 2010. A new technique for background modeling and subtraction for motion detection in real-time videos. In: 2010 IEEE Int. Conf. on Image Processing (ICIP). pp. 3453–3456. http://dx.doi.org/10.1109/ICIP.2010.5653748.

Babaee, M., Dinh, D.T., Rigoll, G., 2018. A deep convolutional neural network for video sequence background subtraction. Pattern Recognit. 76, 635–649. http://dx.doi.org/10.1016/j.patcog.2017.09.040.

Balcilar, M., Amasyali, M.F., Sonmez, A.C., 2014. Moving object detection using Lab2000HL color space with spatial and temporal smoothing. Appl. Math. Inf. Sci. 8, 1755–1766.

Bao, L., Yang, Q., Jin, H., 2014. Fast edge-preserving patchmatch for large displacement optical flow. IEEE Trans. Image Process. 23, 4996–5006. http://dx.doi.org/10.1109/TIP.2014.2359374.

Barnich, O., Droogenbroeck, M.V., 2011. Vibe: A universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. 20, 1709–1724.

Benezeth, Y., Jodoin, P.M., Emile, B., Laurent, H., Rosenberger, C., 2010. Comparative study of background subtraction algorithms. J. Electron. Imaging 19, http://dx.doi.org/10.1117/1.3456695, 033003–0330031–12.

Bianco, S., Ciocca, G., Schettini, R., 2017. Combination of video change detection algorithms by genetic programming. IEEE Trans. Evol. Comput. 21, 914–928. http://dx.doi.org/10.1109/TEVC.2017.2694160.

Bilodeau, G.A., Jodoin, J.P., Saunier, N., 2013. Change detection in feature space using local binary similarity patterns, In: Proc. Int. Conf. Comput. Robot Vis. pp. 106–112.

BMC, 2012. Background models challenge. http://bmc.iut-auvergne.com (accessed Jan. 2017).

Boulmerka, A., Allili, M.S., 2018. Foreground segmentation in videos combining general gaussian mixture modeling and spatial information. IEEE Trans. Circuits Syst. Video Technol. 28, 1330–1345. http://dx.doi.org/10.1109/TCSVT.2017.2665970.

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

Bouwmans, T., 2014. Traditional and recent approaches in background modeling for foreground detection: An overview. Comp. Sci. Rev. 11, 31–66. http://dx.doi.org/10.1016/j.cosrev.2014.04.001.

Bouwmans, T., Javed, S., Sultana, M., Jung, S.K., 2019. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. Neural Netw. 117, 8–66.

Bouwmans, T., Silva, C., Marghes, C., Zitouni, M.S., Bhaskar, H., Frelicot, C., 2018. On the role and the importance of features for background modeling and foreground detection. Comp. Sci. Rev. 28, 26–91. http://dx.doi.org/10.1016/j.cosrev.2018.01.004.

Bouwmans, T., Zahzah, E.H., 2014. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. Comput. Vis. Image Underst. 122, 22–34. http://dx.doi.org/10.1016/j.cviu.2013.11.009.

Braham, M., Droogenbroeck, M.V., 2016. Deep background subtraction with scene-specific convolutional neural networks. In: 2016 International Conference on Systems, Signals and Image Processing (IWSSIP). pp. 1–4. http://dx.doi.org/10.1109/IWSSIP.2016.7502717.

Braham, M., Piérard, S., Droogenbroeck, M.V., 2017. Semantic background subtraction. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 4552–4556..

Brutzer, S., Höferlin, B., Heidemann, G., 2011. Evaluation of background subtraction techniques for video surveillance. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1937–1944.

Cardoso, J.S., Carvalho, P., Teixeira, L.F., Corte-Real, L., 2009. Partition-distance methods for assessing spatial segmentations of images and videos. Comput. Vis. Image Underst. 113, 811–823. http://dx.doi.org/10.1016/j.cviu.2009.02.001.

Cardoso, J.S., Corte-Real, L., 2005. Toward a generic evaluation of image segmentation. IEEE Trans. Image Process. 14, 1773–1782. http://dx.doi.org/10.1109/TIP.2005.854491.

ChangeDetection.NET, 2014. http://www.changedetection.net (accessed September 2, 2019).

Chen, Y., Wang, J., Lu, H., 2015. Learning sharable models for robust background subtraction. In: 2015 IEEE Int. Conf. on Multimedia and Expo (ICME). pp. 1–6. http://dx.doi.org/10.1109/ICME.2015.7177419.

Chua, T.W., Wang, Y., Leman, K., 2012. Adaptive texture-color based background subtraction for video surveillance. In: 2012 19th IEEE Int. Conf. on Image Processing (ICIP). pp. 49–52. http://dx.doi.org/10.1109/ICIP.2012.6466792.

Cuevas, C., Yáñez, E.M., García, N., 2016. Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. Comput. Vis. Image Underst. 152, 103–117. http://dx.doi.org/10.1016/j.cviu.2016.08.005.

Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B., 2007. Neural network approach to background modeling for video object segmentation. IEEE Trans. Neural Netw. 18, 1614–1627. http://dx.doi.org/10.1109/TNN.2007.896861.

Davarpanah, S.H., Khalid, F., Nurliyan. Abdullah, L., Golchin, M., 2016. A texture descriptor: Background local binary pattern (BGLBP). Multimedia Tools Appl. 75, 6549–6568. http://dx.doi.org/10.1007/s11042-015-2588-3.

Elgammal, A., Duraiswami, R., Harwood, D., Davis, L., 2002. Background and foreground modeling using nonparametric kernel density for visual surveillance. Proc. IEEE 90, 1151–1163.

Elhabian, S.Y., El-Sayed, K.M., Ahmed, S.H., 2008. Moving object detection in spatial domain using background removal techniques–state-of-art. Recent Patents Comput. Sci. 1, 32–54.

Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P., 2012. Changedetection.net: A new change detection benchmark dataset. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–8. http://dx.doi.org/10.1109/CVPRW.2012.6238919.

Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P., 2014. A novel video dataset for change detection benchmarking. IEEE Trans. Image Process. 23, 4663–4679. http://dx.doi.org/10.1109/TIP.2014.2346013.

Gregorio, M.D., Giordano, M., 2014. Change detection with weightless neural networks. In: 2014 IEEE Conf. on Computer Vision and Pattern Recognition Workshops. pp. 409–413. http://dx.doi.org/10.1109/CVPRW.2014.66.

Han, B., Davis, L.S., 2012. Density-based multifeature background subtraction with support vector machine. IEEE Trans. Pattern Anal. Mach. Intell. 34, 1017–1023. http://dx.doi.org/10.1109/TPAMI.2011.243.

Heikkilä, M., Pietikäinen, M., 2006. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Mach. Intell. 28, 657–662.

Holtzhausen, P.J., Crnojevic, V., Herbst, B.M., 2015. An illumination invariant framework for real-time foreground detection. J. Real-Time Image Process. 10, 423–433. http://dx.doi.org/10.1007/s11554-012-0287-0.

Isik, S., Özkan, K., Günal, S., Gerek, Ö.N., 2018. SWCD: a sliding window and self-regulated learning-based background updating method for change detection in videos. J. Electron. Imaging 27. http://dx.doi.org/10.1117/1.JEI.27.2.023002.

Jain, V., Kimia, B.B., Mundy, J.L., 2007. Background modeling based on subpixel edges. In: 2007 IEEE Int. Conf. on Image Processing (ICIP). pp. 321–324. http://dx.doi.org/10.1109/ICIP.2007.4379586.

Jiang, S., Lu, X., 2018. Wesambe: A weight-sample-based method for background subtraction. IEEE Trans. Circuits Syst. Video Technol. 28, 2105–2115. http://dx.doi.org/10.1109/TCSVT.2017.2711659.

Jodoin, P.M., Maddalena, L., Petrosino, A., Wang, Y., 2017. Extensive benchmark and survey of modeling methods for scene background initialization. IEEE Trans. Image Process. 26, 5244–5256. http://dx.doi.org/10.1109/TIP.2017.2728181.

Kim, W., Jung, C., 2017. Illumination-invariant background subtraction: Comparative review, models, and prospects. IEEE Access 5, 8369–8384. http://dx.doi.org/10.1109/ACCESS.2017.2699227.

Kim, J., Rivera, A.R., Ryu, B., Chae, O., 2015. Simultaneous foreground detection and classification with hybrid features. In: 2015 IEEE Int. Conf. on Computer Vision (ICCV). pp. 3307–3315. http://dx.doi.org/10.1109/ICCV.2015.378.

Kristensen, F., Nilsson, P., Owall, V., 2006. Background segmentation beyond RGB. In: others, P.J.N. (Ed.), Proc. of the 7th Asian Conf. on Computer Vision (ACCV 2006). Springer-Verlag, Berlin Heidelberg, pp. 602–612.

Lee, D.S., 2005. Effective gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. 27, 827–832. http://dx.doi.org/10.1109/TPAMI.2005.102.

Lee, S.H., cheol Lee, G., Yoo, J., Kwon, S.C., 2018. Wisenetmd: Motion detection using dynamic background region analysis. Symmetry 11, 621.

Li, L., Huang, W., Gu, I.Y.H., Tian, Q., 2004. Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. Image Process. 13, 1459–1472. http://dx.doi.org/10.1109/TIP.2004.836169.

Li, C., Lin, C.W., Yu, S.S., Chen, T., 2011. Joint optimization of background subtraction and object detection for night surveillance. In: 18th IEEE Int. Conf. on Image Processing (ICIP). pp. 1753–1756. http://dx.doi.org/10.1109/ICIP.2011.6115799.

Liao, S., Zhao, G., Kellokumpu, V., Pietikäinen, M., Li, S.Z., 2010. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1301–1306. http://dx.doi.org/10.1109/CVPR.2010.5539817.

Lim, L.A., Keles, H.Y., 2018a. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. arXiv preprint arXiv:1801.02225 (under consideration at Pattern Recognition Letters).

Lim, L.A., Keles, H.Y., 2018b. Foreground segmentation using convolutional neural networks for multiscale feature encoding. Pattern Recognit. Lett. 112, 256–262. http://dx.doi.org/10.1016/j.patrec.2018.08.002.

Lim, L.A., Keles, H.Y., 2018c. Learning multi-scale features for foreground segmentation. CoRR abs/1808.01477.

Lindström J. Lindgren, F., Åström, J., Holst, U., 2006. Background and foreground modelling using an online em algorithm. In: IEEE Int. Workshop on Visual Surveillance VS 2006 in Conjunction with ECCV 2006. pp. 9–16.

López-Rubio, F.J., López-Rubio, E., 2015. Features for stochastic approximation based foreground detection. Comput. Vis. Image Underst. 133, 30–50. http://dx.doi.org/10.1016/j.cviu.2014.12.007.

López-Rubio, E., Luque-Baena, R.M., Domínguez, E., 2011. Foreground detection in video sequences with probabilistic self-organizing maps. Int. J. Neural Syst. 21, 225–246. http://dx.doi.org/10.1142/S012906571100281X.

Maddalena, L., Petrosino, A., 2014. The 3dSOBS+ algorithm for moving object detection. Comput. Vis. Image Underst. 122, 65–73. http://dx.doi.org/10.1016/j.cviu.2013.11.006.

Maddalena, L., Petrosino, A., Ferone, A., 2008. Object motion detection and tracking by an artificial intelligence approach. Int. J. Pattern Recognit. Artif. Intell. 22, 915–928. http://dx.doi.org/10.1142/S0218001408006612.

Martins, I., Carvalho, P., Corte-Real, L., Alba-Castro, J.L., 2016. Bio-inspired boosting for moving objects segmentation. In: Int. Conf. Image Analysis and Recognition, ICIAR 2016. Springer International Publishing, pp. 397–406.

Martins, I., Carvalho, P., Corte-Real, L., Alba-Castro, J.L., 2017. BMOG: Boosted gaussian mixture model with controlled complexity. In: Iberian Conf. Pattern Recognition and Image Analysis, IbPRIA 2017. Springer International Publishing, pp. 50–57.

Martins, I., Carvalho, P., Corte-Real, L., Alba-Castro, J.L., 2018. BMOG: boosted Gaussian mixture model with controlled complexity for background subtraction. Pattern Anal. Appl. 21, 641–654. http://dx.doi.org/10.1007/s10044-018-0699-y.

Mondéjar-Guerra, V., Rouco, J., Novo, J., Ortega, M., 2019. An end-to-end deep learning approach for simultaneous background modeling and subtraction. In: 30th British Machine Vision Conference (BMVC 2019).

Oliver, N.M., Rosario, B., Pentland, A.P., 2000. A bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. 22, 831–843. http://dx.doi.org/10.1109/34.868684.

Porikli, F., Davis, L., Wang, Q., Li, Y., Regazzoni, C., 2016. Special issue on deep learning for video surveillance. IEEE Trans. Circuits Syst. Video Technol. 26, 2159–2160.

Sobral, A., Vacavant, A., 2014. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput. Vis. Image Underst. 122, 4–21. http://dx.doi.org/10.1016/j.cviu.2013.12.005.

St-Charles, P.L., Bilodeau, G.A., Bergevin, R., 2015. Subsense: A universal change detection method with local adaptive sensitivity. IEEE Trans. Image Process. 24, 359–373. http://dx.doi.org/10.1109/TIP.2014.2378053.

St-Charles, P.L., Bilodeau, G.A., Bergevin, R., 2016. Universal background subtraction using word consensus models. IEEE Trans. Image Process. 25, 4768–4781. http://dx.doi.org/10.1109/TIP.2016.2598691.

Stauffer, C., Grimson, W., 1999. Adaptive background mixture models for real-time tracking. In: 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2. pp. 246–252. http://dx.doi.org/10.1109/CVPR.1999.784637.

Tanaka, T., Shimada, A., Arita, D., ichiro Taniguchi, R., 2007. A fast algorithm for adaptive background model construction using parzen density estimation. In: 2007 IEEE Conf. on Advanced Video and Signal Based Surveillance. pp. 528–533. http://dx.doi.org/10.1109/AVSS.2007.4425366.

Vacavant, A., Chateau, T., Wilhelm, A., Lequièvre, L., 2012. A benchmark dataset for outdoor foreground/background extraction. In: Park, J.I., Kim, J. (Eds.), ACCV 2012, Workshop: Background Models Challenge. Springer Berlin Heidelberg, pp. 291–300. http://dx.doi.org/10.1007/978-3-642-37410-4_25.

Varadarajan, S., Miller, P., Zhou, H., 2015. Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. Pattern Recognit. 48, 3488–3503. http://dx.doi.org/10.1016/j.patcog.2015.04.016.

Wang, R., Bunyak, F., Seetharaman, G., Palaniappan, K., 2014a. Static and moving object detection using flux tensor with split gaussian models. In: 2014 IEEE Conf. on Computer Vision and Pattern Recognition Workshops. pp. 420–424. http://dx.doi.org/10.1109/CVPRW.2014.68.

Wang, Y., Jodoin, P.M., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P., 2014b. Cdnet 2014: An expanded change detection benchmark dataset. In: 2014 IEEE Conf. on Computer Vision and Pattern Recognition Workshops. pp. 393–400. http://dx.doi.org/10.1109/CVPRW.2014.126.

Wang, Y., Luo, Z., Jodoin, P.M., 2017. Interactive deep learning method for segmenting moving objects. Pattern Recognit. Lett. 96, 66–75. http://dx.doi.org/10.1016/j.patrec.2016.09.014.

Wang, H., Suter, D., 2007. A consensus-based method for tracking: Modelling background scenario and foreground appearance. Pattern Recognit. 40, 1091–1105. http://dx.doi.org/10.1016/j.patcog.2006.05.024.

Yang, L., Cheng, H., Su, J., Li, X., 2016. Pixel-to-model distance for robust background reconstruction. IEEE Trans. Circuits Syst. Video Technol. 26, 903–916. http://dx.doi.org/10.1109/TCSVT.2015.2424052.

Yoshinaga, S., Shimada, A., Nagahara, H., Taniguchi, 2013. Background model based on statistical local difference pattern. In: Park, J.I., Kim, J. (Eds.), Computer Vision - ACCV 2012 Workshops. Springer Berlin Heidelberg, pp. 327–332.

Zhang, B., Gao, Y., Zhao, S., Zhong, B., 2011. Kernel similarity modeling of texture pattern flow for motion detection in complex background. IEEE Trans. Circuits Syst. Video Technol. 21, 29–38. http://dx.doi.org/10.1109/TCSVT.2011.2105591.

Zhao, Y., Gong, H., Lin, L., Jia, Y., 2008. Spatio-temporal patches for night background modeling by subspace learning. In: 19th Int. Conf. on Pattern Recognition. pp. 1–4. http://dx.doi.org/10.1109/ICPR.2008.4761197.

Zhen, W.B., Wang, K.F., Wang, F.Y., 2018. Background subtraction algorithm with bayesian generative adversarial networks. Acta Automat. Sinica 44, 878–890. http://dx.doi.org/10.16383/j.aas.2018.c170562.

Zheng, W., Wang, K., Wang, F.Y., 2019. A novel background subtraction algorithm based on parallel vision and Bayesian GANs. Neurocomputing http://dx.doi.org/10.1016/j.neucom.2019.04.088.

Zivkovic, Z., 2004. Improved adaptive gausian mixture model for background subtraction. In: Proc. of the 17th Int. Conf. on Pattern Recognition, 2004. ICPR 2004. pp. 28–31.

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