Dynamic Spatial Predicted Background

Yaniv Tocker, Rami R. Hagege, and Joseph M. Francos

***Abstract*— We present a novel method for online background modeling for static video cameras - Dynamic Spatial Predicted Background (DSPB). Our unique method employs a small subset of image pixels to predict the whole scene by exploiting pixel correlations (distant and close). DSPB acts as a hybrid model combining successful elements taken from two major approaches: local-adaptive that propose to fit a distribution pixelwise, and global-linear that reconstruct the background by finding a low-rank version of the scene. To our knowledge, this is the first attempt to combine these approaches in a unified system. DSPB models the scene as a superposition of illumination effects and predicts each pixel's value by a linear estimator comprised of only 5 pixels of the scene and can initialize the background starting from the 5th frame. By doing so, we keep the computational load low, allowing our method to be used in many real-time applications using simple hardware. The suggested prediction model of scene appearance is novel, and the scheme is very accurate and efficient computationally. We show the method merits on an application for video FG-BG separation, and how some of the main existing approaches may be challenged and how their drawbacks are less dominant in our model. Experimental results validate our findings, by computation speed and mean F-measure values on several public datasets. We also examine how results may improve by analyzing each video individually according to its content. DSPB can be successfully incorporated in other image processing tasks like change detection, video compression and video inpainting.**

***Index Terms*— Background modeling, foreground-background separation, motion detection, spatial prediction, video analysis.**

I. Introduction

THE use of cameras has become increasingly common in a variety of fields in recent years. As a result, the need for video analysis algorithms has risen to automate tedious manual tasks, such as: intelligent surveillance cameras, autonomous driving, gesture recognition, *etc*. All these applications have in common the need to initially detect the object of interest in a video, noted as "Foreground" (FG) and to separate them from the irrelevant static information, the background (BG). Therefore, background modeling received a lot of attention in past decades and many approaches were suggested.

In spite of the huge effort invested in this direction, there currently isn't a single algorithm to solve the FG-BG separation problem and this is mainly due to the significant challenges arising from the richness of natural scenes. Illumination changes (gradual or sudden) and dynamic background (*e.g.* waving trees) are some of the major concerns caused by real-life scenarios. Implications also arise from the cameras themselves, such as automatic camera adjustments, camera jitter and sensor noise. Some requirements of such a system are determined by the user as to how they should be dealt with; Should a parking car be incorporated into the BG or not? Are shadows part of the object or irrelevant? It is worth mentioning that since most surveillance cameras hardware is basic, there is a need to keep the computational load as low as possible, as in all video analysis applications, in order to deliver real-time performance. A foreground detection system is normally comprised of 3 components: (a) The background model initialization, typically reconstructing the background image somehow (b) creating the foreground mask, that is simply a binary image marking the foreground pixels, and (c) background maintenance - a necessary step to keep the background model relevant to the current scene content as the video advances.

In this paper, we focus our attention on handling illumination changes (gradual and sudden), as they are considered to be the main source of variation in videos that does not arise from foreground objects [1], [2]. We present a novel method for online background estimation that models the scene as a superposition of the light sources. The model was briefly described in [3]. Section 2 discusses various background subtraction methods and their relative advantages. In Section 3 we present our method and discuss its innovations - handling illumination changes, creating the BG model as early as the 5th frame, while requiring a minimal computational load. Section 4 describes comparisons to state-of-the-art methods, presenting both quantitative and qualitative results on public benchmark datasets. We conclude our findings in Section 5.

II. Related Work

Background modeling has been an active field of research during the past decades, yielding a plethora of algorithms and systems that can be found in surveys like [4]–[7]. A basic assumption which derives from the nature of the FG-BG separation problem is that the vast majority of a scene's content is composed by the background, while the foreground part exhibits objects that pass through the scene. Most methods reconstruct the BG image in some way and compare the obtained image to raw frames in order to mark FG pixels. Others treat the video as anomaly detection system and refer to the FG-BG separation as a change detection problem. The created BG model needs to be initialized and for this purpose many methods are reviewed in [8]. Afterwards, the scene

Manuscript received August 27, 2018; revised November 12, 2019 and February 5, 2020; accepted March 5, 2020. Date of publication April 1, 2020; date of current version April 17, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jocelyn Chanussot. *(Corresponding author: Yaniv Tocker.)*

Yaniv Tocker and Joseph M. Francos are with the Department of Electrical and Computer Engineering, Ben-Gurion University, Beersheba 8410501, Israel (e-mail: ytocker@gmail.com; francos@ee.bgu.ac.il).

Rami R. Hagege is with Lifetobot (e-mail: rami.r.hagege@gmail.com).

Digital Object Identifier 10.1109/TIP.2020.2983598

undergoes some changes (*e.g.* illumination changes, state change of objects *etc.*). Thus, the BG model is updated. This phase can be very resource-consuming in naive approaches that keep conducting the same procedure as in initializing the BG over and over again. Hence, implementing a learning rate to incrementally absorb the new frame's information is popular in many methods [2], [9], [10]. We divide the rich literature coarsely into local-adaptive, global-linear, hybrid models and learning based. Local-adaptive methods form a pixelwise statistical model of the scene. Hence, these methods do not explicitly construct a BG image but try to find the main trend pixelwise. Local-adaptive methods consider a given video stream as a bundle of individual 1D signals. By doing so, the spatial information remains untreated, except for some modern methods that consider neighbor pixels recent history. Typically, the first frames of each video are used to train a model for each pixel by empirically estimating statistic measures that suite the chosen model. The number of frames used for training impacts the model's ability to explain the variability among each pixel. Ideally, the first frames shouldn't contain any foreground objects, but this assumption is far-reaching and almost never is obtained in real scenarios. After the model is constructed, next frames are classified and the model parameters are updated constantly, mostly using a learning rate. Simple operations calculated on a temporal sliding window include average [11], median [12] or histogram analysis [13]. Applying popular filters (Kalman [14], Chebyshev [15] and Wiener [16]) express the same idea by obtaining the low frequencies and reconstructing the BG image from them. Parametric methods model pixel distributions and calculate the probability of a new sample belonging to it and classify the pixel according to the outcome. The simplest form of this approach assumes each pixel is derived from a unimodal Gaussian distribution [17], where the mean and variance are calculated empirically using the training set. The probability density function (pdf) parameters are continuously updated as the video stream keeps delivering frames for classification. These simple methods are very easy to implement and barely have any computational requirements in compare to more sophisticated methods. However, they cannot describe enough the variations arises from the richness of most natural scenes (*e.g.* moving background or illumination changes). Thus, employing a more elaborated pdf, like a Gaussian Mixture Model (GMM) is more popular. The Gaussians number directly affects computational demands, as for each its parameters need to be empirically evaluated and constantly updated. The first GMM was introduced by Stauffer and Grimmson [9] in the late 90's and since then many improvements have been suggested to make GMM more durable and accurate. In [10] for each pixel color and texture features are added and used to comprise the GMM. A popular extension [18] estimates for each pixel how many Gaussians should be in the model. By doing so it keeps the method flexible and avoids overfitting where a simple model can be used, while allowing more complexity to other pixels that tend to fluctuate. GMM-based methods are capable of explaining the majority of BG variations that appear in most outdoor scenes, particularly where minor gradual illumination changes are exhibited along with repetitive motions (*e.g.* waving branches and sea waves). GMM in all its versions along with the other pixelwise methods, struggle with handling the case of when in the training phase the data appears differently in compare to the scene content later on. For instance, if FG objects are present during the training phase, then the objects will get partially absorbed into the BG model (this is better known as "ghost-like" effect). Moreover, in indoor scenarios sudden illumination changes appear often (*e.g.* screens and light switches) and severely alters the scene's appearance leading to many false detections. To handle this issue, GMM can use an aggressive learning rate but this also results in blending FG object pixels to the BG model. Using non-parametric methods has the advantage of not forcing a specific distribution to pixels distribution, whereas the question if a Gaussian distribution or another fits the data in a statistically significant manner is hardly ever asked. In [19] the histogram is smoothed using Kernel Density Estimation (KDE). However, applying this technique by a sliding window is very time consuming. More non-parametric methods are based on preserving each pixel's recent BG values. Sample consensus is used in [20] to maintain and update each pixel's model. The visual background subtractor (ViBe) [21] is more robust to gradual illumination changes since it has a stochastic strategy to maintain the BG model by randomly neglecting a sample from a cached memory with recent BG values when adding a more recent one to it. Another interesting point the authors make is that neighboring pixels tend to express the same intensity distributions. They exploit this observation if a pixel is set to be FG then the BG model at its location is set by a value from neighboring pixels. The downside of these methods is that they need to store past values for each frame (20 is a reasonable number), which is a luxury that is not plausible when considering a method to be embedded on simple hardware like surveillance cameras. More methods that use the surrounding pixels around each to better estimate the correct BG value suggest to use Principal Component Analysis (PCA) on spatial blocks [22], [23]. This way, the computational needs are reduced as the classification of each block is homogeneous. Codebooks methods are used by establishing codewords for each pixel that describes the BG. The codewords are actually features extracted from the training frames. Kim *et al.* [24] use a unique color distortion metric based on RGB values and their acceptable limits, while in [25] the temporal and spatial context are represented in the codewords. The created codebooks serve as a compressed version of the BG. The performance speed of these methods relies on the size of the codebook which trades-off between the models accuracy.

Global-linear methods main advantage is by acquiring the BG model at the frame level. These methods apply dimension reduction techniques to the FG-BG separation problem as they estimate a low rank version of the scene. In the most basic form, PCA is used for the BG reconstruction and the FG mask is created by thresholding the difference between the video frame and the BG image [26]. PCA creates an eigen background - a linear model that spans over the eigenimages of the data is was trained on. The eigenimages usually describe the possible illumination conditions, as they are the main

source of variation in the data. Hence, these methods tend to deal well with illumination changes (gradual or sudden). Applying simple PCA results in having many outlier values in the FG mask and also "ghost-like" effects, as a result of FG objects that do not pass fast enough through the scene and are partially absorbed into the BG model. Vosters *et al.* [2] suggest using the simple PCA model to create the BG, and further utilize a statistical illumination model [27] that models the distribution of the ratio of intensities using a GMM in order to better classify FG pixels. Robust PCA (RPCA) [28] handles outliers by forcing a division of the original frame into a BG image (low rank version of the scene) and another sparse matrix that represents FG objects. In their work they use principal component pursuit (PCP) as their solution to the RPCA problem. Some other methods treat the problem as a matrix [29] or tensor [30] completion task with respect to energy loss minimization criteria. In [31] graph-cut is used on the low-rank component of the video stream to better handle background variations. RPCA-based solutions have gained much popularity and became a thriving sub-criterion among the global-linear methods. Some surveys review solely RPCA-based method [32], [33] and demonstrate the various metrics and optimization methods used. Subspace learning yields a very good distinction quality by obtaining a linear subspace of the original video. It can deal with scenarios that express sudden and gradual illumination changes which are popular in indoor videos (*e.g.* light switches and screens). In spite of the mentioned benefits, applying such methods requires substantial computational resources due to the need to batch-process many frames and compute complex operations (*e.g.* inverting each batch's cross-correlation matrix). Moreover, each batch must be acquired in advance which dramatically effects the ability to perform in real-time. If a scene undergoes many lighting variations, which is typically the case for long videos as in surveillance cameras, in order to model illumination changes correctly there is a need to reconstruct the BG with many eigenimages or else the model's capacity to describe the illumination changes drastically diminishes.

Learning-based methods apply traditional machine learning techniques for FG-BG separation. Support Vector Machine (SVM) is used to classify pixels after the training set batch while constructing a density function based on varying features. In [34] inter-frame difference and optical flow are used, while [35] use Haar, color and gradient features. Neural Networks (NN) tackle the issue as a supervised learning problem, in which weights of the network are learned and serve as self-learned features. In a significant work, Maddalena *et. al* present the Self-Organizing Background Subtraction network (SOBS) [36] that learns motion patterns in the HSV color space on the image sequences. An updated version adds a spatial coherence mechanism to maintain the BG model (SC-SOBS) [37] that rejects many false detections. The idea to apply Deep Neural Networks (DNN) and Convolutional NN (CNN) architectures on FG-BG separation problem has intrigued researchers since they gained much popularity due to their success in object detection tasks on the ImageNet dataset classification challenge [38] by Krizhevsky *et. al* [39]. CNN/DNN differ from traditional NN by having more hidden layers in the network's architecture and by using images spatial information. Due to the many layers in DNN frameworks, a large portion of the BG variability can be explained. In [40] the authors suggest a ground-truth generation method where some objects need to be marked by the user and then the network classifies each pixel by feeding it a 31×31 patch around it. Braham and Droogenbroeck [41] apply CNN in a pioneering work by using some scenario-specific training frames in order to teach the network the BG variations. The authors use a 27×27 patch to calculate the FG probability of each pixel. CNN methods main virtue is by not requiring a complex modeling procedure - the network learns themselves how to model the scene. The downside of the mentioned DNN methods is that they are scene-specific, so the networks need to be retrained on new scenarios. Also, patch processing is very resource consuming. DNN architectures for FG-BG separation continues to be an active field of research and is constantly evolving. We refer the reader to the recent work of Bouwmans *et. al* [42] that reviews and compares recent advancements.

Another research topic with tight relations to the presented one addresses the BG-FG separation problem while allowing cameras to be freely moving. In this way, there is a need to estimate the camera's motion along with local motion patterns that differ from the predicted global one in order to classify pixels as FG or BG. Feature points trajectories, geometric camera movement using homographic transformations and obtaining the camera's fundamental matrix transformations, are some of the techniques used to infer the motion patterns in the scene. Reference [43] adapts a multi-label process by assigning a unique foreground layer for each foreground object. Next, a probability is set using a Bayesian filtering framework and labeling is done using a graph-cut on a Markov Random Field (MRF). Sugimura *et al.* [44] suggest using a subset of the scene pixels and estimate their motion using optical flow to create "motion seeds". These seeds are used to establish motion boundaries that help differentiating FG objects motion in compare to the global motion of the camera. An interesting point the authors make is that when the scene is partially stationary for a long period, as in surveillance cameras, due to the lack of motion it is hard to differentiate the FG from the BG, therefore a different scene appearance modeling should be applied.

III. DSPB: A Dynamic Spatially Predicted Background Model

A. Theoretical Foundations

Cameras acquire images according to the amount of light perceived on each detector in a sensor-matrix array that represents each pixel's intensity value. Therefore, handling gradual and sudden illumination changes remains key challenges in any image and video processing algorithm. In this section we aim to model a given scene as a function of its light sources and show that it can be expressed as a linear combination of a subset of the scene's pixels. To do so, we need to understand the function of light sources for the incoming light to the camera. This is better known as Bidirectional Reflectance

Distribution Function (BRDF), which is merely the relation between the light emitted by a source partially absorbed in an object and the amount of light reflected by it that the camera acquires.

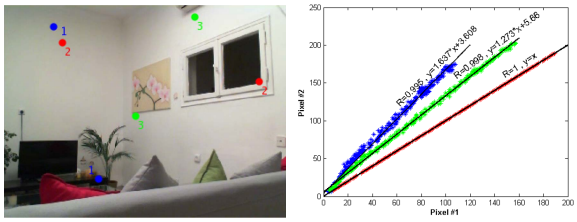
Appearance modeling under different illuminations for object recognition and analysis was investigated by acquiring images with an object in the same pose under different lighting conditions, and then reconstructing the scene using a projection matrix, such as PCA [45] and harmonic spheres [46]. In both works the authors use a linear approximation to minimize the energy loss to some extent. In other cases, appearance models were based on a-priori knowledge of the content of the scene. This a-priori knowledge can be: the 3D structure of the observed object [46], the reflectance properties of the observed surface and the type of illumination in shape from shading [47] and photometric stereo [48]. Linear models for scene appearance modeling for separation of foreground objects is used extensively [26], [28], [33]. Theoretically, the use of linear models can be explained as a superposition of light sources. The measured image can be expressed as a linear combination of the different individual light sources. Basri and Jacobs [46] provide a theoretical explanation why we can assume that a 9D model should suffice for modeling a scene appearance in the vast majority of scenarios. In practice we see that even fewer dimensional models are enough in many cases. In [49], Hagege discusses extensively how the background can be modeled as a linear function of the scene's light sources. We elaborate on the notions presented there to form a FG-BG separation method. In our work, a key assumption is that the illumination in the scene is influenced by a finite number of independently varying static light sources. In such a case, the background model can be analyzed using a linear model [49]. In fig.1 we show how strong correlations among pixels are, as 3 randomly chosen pixel pairs values are shown during a short video clip in a static scene as illumination changes are caused by a single varying light source. A linear fit on each pair almost fits perfectly to each pair. This demonstration lays the foundations for predicting a pixel's value by knowing another's history and modeling the illumination effects of the given scene. A predictor for the measurements of each pixel is built based on a subset of other pixels in the scene. We call the pixels used for prediction "control pixels". Each predictor is constructed by exploiting the correlations between measurements of the observed and control pixels. To our knowledge, our work is the first to exploit pixel correlations (far or near) to predict the scene. The correlations are calculated empirically and therefore not based on any assumptions about the type of illumination sources, potential locations, or the type of material (Lambertian or not) in front of the camera.

B. Model Formation

The measurements of a single pixel p in image I can be written as

$$I(p) = M^p A \tag{1}$$

where M^p is a row vector describing the contributions of each light source power to the total outcome. Each element of M^p



(a) (b)

Fig. 1. (a) 3 randomly taken pixels pairs and their values with a linear fit in (b).

stands for the spatial distributions of the light sources and the effects of their BRDF's on the scene. We assume that the changes in the illumination powers are the only ones in the scene, beside foreground objects. Given the light sources powers A, we assume that the measurements of two pixels are independent. We argue that given a set of N pixels $\{p_k\}_{k=1}^N$ measurements, any other pixel in the scene can be estimated. the set $\{p_k\}_{k=1}^N$ when used in eq.1 forms N equations in the form:

$$I(p_k) = M^{p_k} A \tag{2}$$

we set $I_{\{p_k\}_{k=1}^N}$ as a column vector, where $[I_{\{p_k\}_{k=1}^N}]_k = I(p_k)$ and $M \in \mathbb{R}^{NxN}$ where $[M]_k = M^{p_k}$, to get the matrix form of the previous equation.

$$I_{\{p_k\}_{k=1}^N} = MA \tag{3}$$

We assume the matrix M is of full rank (which is empirically evaluated in [49]), therefore matrix M is invertible, yielding:

$$A = M^{-1} I_{\{p_k\}_{k=1}^N} \tag{4}$$

Since the measure of any pixel is given by Eq. 1, then using the previous equation, we get

$$I(p) = M^p A = M^p M^{-1} I_{\{p_k\}_{k=1}^N} \tag{5}$$

Let us denote $T = M^p M^{-1}, T \in \mathbb{R}^{1xN}$ and we get

$$I(p) = T I_{\{p_k\}_{k=1}^N} \tag{6}$$

Eq. 6 indicates that any pixel of an image is linearly dependent and well approximated by any other N pixels. This implies that correlations between pixels have valuable information (fig. 1) that can be used to predict the scene. The last equation can be rearranged to establish direct relations of the chosen control pixels $\{p_k\}_{k=1}^N$ to all the pixels in a given image I at time t:

$$I_t = \mathbb{T} I_{\{p_k\}_{k=1}^N} \tag{7}$$

where $\mathbb{T} \in \mathbb{R}^{MxN}$ denotes the weight of each control pixel in each of the M pixels of the scene.

In real scenarios, eq 7 isn't plausible directly, since there isn't a way to obtain T or $\mathbb{T}$, and the fact that we omitted the statistical correlations between pixels of the observation set $\{p_k\}_{k=1}^N$. However, this framework serves as a theoretical explanation that predicting pixel measurements linearly from other pixels of the scene should behave well. In practice, the solution can be found by solving a minimization problem,

of finding a linear subspace that is similar to a given frame with respect to the L_2 norm.

$$\min||I_t - AI_{\{p_k\}_{k=1}^N}||_2 \tag{8}$$

where $I_t \in \mathbb{R}^{M\times 1}$ is a given image in column stack fashion at time t, $I_{\{p_k\}_{k=1}^N} \in \mathbb{R}^{N\times 1}$ are the observations at control pixels locations, and $A \in \mathbb{R}^{M\times N}$ is the coefficients matrix. The least-squares solution of Eq.8 can easily be retrieved. However, such a solution is computationally demanding and requires storing all the observed images. An online version can be achieved by constructing the optimal linear estimator [50]:

$$I_t = \mathbb{E}_{I_t} + Cov_{I_t P} \cdot (Cov_P)^{-1} \cdot (P_t - \mathbb{E}_P) \tag{9}$$

- $P \in \mathbb{R}^{N\times 1}, [P]_k = p_k$ is the vector of control pixels and P_t is their value at time t.
- $\mathbb{E}_{I_t}$: The empirical Expectation of all pixels.
- $\mathbb{E}_P$: The empirical Expectation of $\{p_k\}_{k=1}^N$ control pixels.
- Cov_P: The empirical Covariance matrix of P
- $Cov_{I_t P}$: The empirical Cross-Covariance matrix of P with all other pixels.

All the values mentioned above are calculated empirically on the training set which is taken as the first few frames of the video stream. By using the linear estimator for each pixel, the complete frame is estimated and models the background image, since it describes how the scene should appear based on the lighting conditions exhibited in the scene and has no relation to modeling foreground objects.

An example of the process using a control set of 3 pixels (red, blue & green) to estimate another (in pink) is shown in fig 2. The upper sub images show cases in which the control pixels estimate a pixel, once as BG (a) and as FG (b). Lower subfigures show the control pixel's measurements during the video stream (c) and the observations of the pink pixel (d) are compared to the estimated ones (in black). Points where there is a large deviation between black and pink pixels indicate that outlier values are measured and are suspected to derive from the presence of a FG object.

C. Relations to Other Image Representation Models

The suggested appearance model is strongly related to other linear subspace reconstruction methods that also take place in video compression. A common way to do so is by coding the data using a linear subspace:

$$||I - Ba||_2 = ||I - \sum_{i=1}^{m} \alpha_i B_i||_2 \tag{10}$$

where $\{\alpha_i\}_{i=1}^m \in \mathbb{R}$ and $\{B_i\}_{i=1}^m \in \mathbb{R}^{N\times 1}$ are the weights and basis images set that span the image space. $I \in \mathbb{R}^{N\times 1}$ is the current frame in column stack fashion and N is the number of pixels in each frame.

m turns out to be the number of control pixels used to form the frame prediction, as shown further on in this section. For convenience we denote: $a = [\alpha_1 \alpha_2 \dots \alpha_m]^T$ and $B = [B_1\ B_2..B_m]$

Rearranging eq. 9 yields:

$$I_t - \mathbb{E}_{I_t} \approx Cov_{I_t P} \cdot (Cov_P)^{-1} \cdot (P_t - \mathbb{E}_P) \tag{11}$$

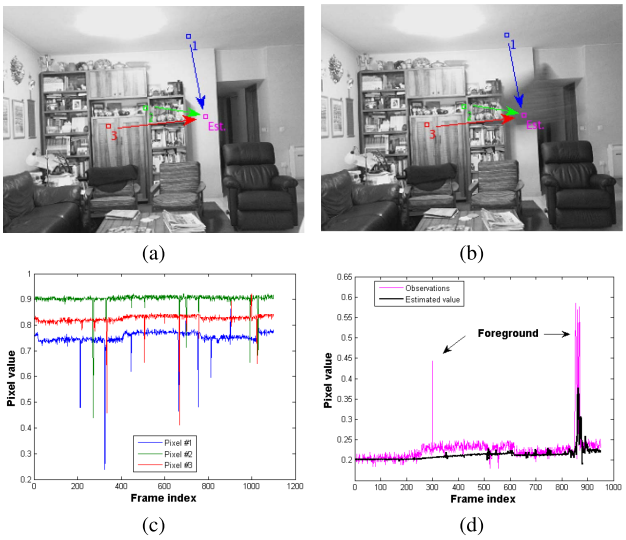


Fig. 2. An example of 3 control pixels (red, green & blue) predicting another (in pink). BG value (a) and FG (b) examples from the video. (c) control pixels measurements during the video sequence, and (d) shows the predicted values compared to measured ones.

and set $I_t^* = I_t - \mathbb{E}_{I_t}$ and $P^* = (P_t - \mathbb{E}_P)$ to get

$$I_t^* \approx Cov_{I_t P} \cdot inv(Cov_P) \cdot P^* \tag{12}$$

while the dimensions of I_t^* and P^* match I_t and P correspondingly, since these are merely their values without the empirical expectation. $C = Cov_{I_t P} \cdot inv(Cov_P)$ yields:

$$I_t^* \approx C \cdot P^* \tag{13}$$

Note that $C \in \mathbb{R}^{N\times m}$ and $P^* \in \mathbb{R}^{m\times 1}$ have the same dimensions as a and B in Eq 10.

From here we conclude that matrix C can be considered as a set of basis images (as columns) and their weights for each frame that form the linear combination is vector P^* which is the control pixel set values in the current frame without their empirical expectation. Mathematically, the suggested predictor acts as a projection of the observed image into the subset of the potential scene appearance, as given by the scene appearance model. It is important to mention that the basis images as vectors are highly correlated. This isn't surprising, since each of them can be thought of as the estimation of the current frame using a single control pixel. The estimated image is considered to be the background image, so we expect that each estimation should have similar behavior, but under different illumination conditions. Thus, the set of basis images does not exactly span the image space, since its components are not orthogonal. To achieve this, further processing is needed, like applying PCA or the Gramm-Schmidt process on the basis images set.

D. Model Limitations

The estimator quality depends on the following:

(a) *Illumination changes among the video stream*- To construct an estimator of I_t based on the control pixels set, we need to have enough observations under various light

conditions. In natural scenes there are many light variations (gradual or sudden), like screens (phones, TV etc.), windows and sunlight (which is often partially occluded by clouds). Thus, most surveillance camera's typical scenarios produce enough variability to enable the usage of the proposed method.

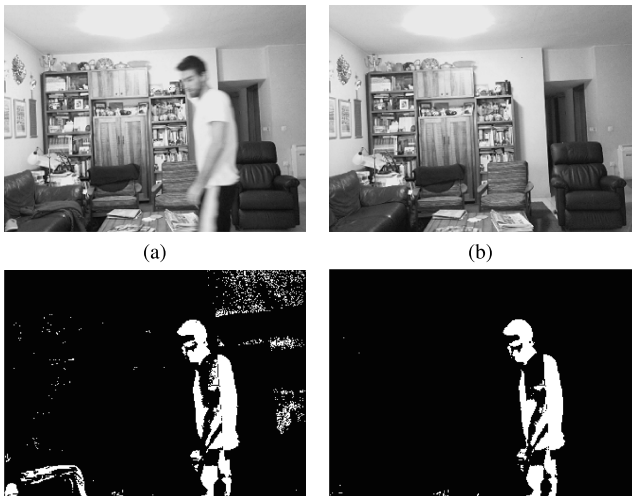
(b) *Number of control pixels*- if we have a prediction scheme based on N pixels, we need to have N images such that their linear combination with coefficients $I_{\{p_k\}_{k=1}^N}$ constitutes the prediction of the scene appearance. The estimation of these N basis images is the statistical approximation of $I_{\{p_k\}_{k=1}^N}$ in all image locations. Using more control pixels leads to more accurate results. However, enlarging their amount also increases the computational demand, as the number of parameters to be empirically estimated also rises. In Section 4-D we examined a large range of values for the control pixels set size and empirically found that 5 control pixels are sufficient to correctly estimate the BG in a satisfying manner.

(c) *Specific control pixel locations*- A different choice of control pixels produces a different scene appearance model; yet, different scene appearance models should span approximately the same linear space. The scene appearance predictors, however, behave differently in the presence of errors and outliers; For example, some predictors may completely fail as a result of occlusions. The accuracy of the predictions as a function of location using any specific predictor is not uniform, namely, there are pixels with smaller and larger errors. The use of more than one scene appearance predictor creates regions in which one predictor is statistically better than another scene appearance predictor. This information can be used to create an overall better predictor. In light of these observations, we improve our model's robustness by taking a few independent sets of control pixels. De facto, we take 3 control sets: $\left(P_1 = \{p_k^1\}_{k=1}^N, P_2 = \{p_k^2\}_{k=1}^N, P_3 = \{p_k^3\}_{k=1}^N\right)$ to form 3 linear predictors. Each control pixel is unique and is used in one control pixel set solely-$\left((P_1 \cap P_2 \cap P_3) = \emptyset\right)$. The scene's prediction is done independently for each set, yielding 3 candidates at the pixel level for the background model. In the presence of a disturbance (*e.g.* noise or occluded control pixels) it is unlikely that all 3 different estimators will predict the same value at a specific pixel. To avoid picking an unreliable value, we use the 3 candidates median value. By doing so, durability to occlusions by outlier rejection is achieved which enhances our model performance.

(d) *The scene's content*- Neighboring pixels tend to express the same spatial and temporal behaviors [21]. Thus, using near pixels to estimate their surrounding area may deliver better predictions. Segmenting an image into clusters, blocks or super-pixels exists in some prior works. [22] performs PCA on image fragments and classifies them homogeneously in order to reduce computational time, while [51] segments the scene into super-pixels and then uses a density-based spatial clustering (DBSCAN) to form megapixels. A probability is set to each megapixel of belonging to the BG. We suggest dividing the image content into areas that express the same characteristics and to construct a predictor from control pixels in a specific area to estimate pixels that solely belong to the same region. Due to the fact that the vast majority of the scene does not change during a video stream, it is sufficient to divide the image at the start of the video without any update to the cluster borders. To give respect to the surroundings of each pixel, we use k-means clustering on the median image of the first 10 frames, while the used features are pixel cartesian coordinates and color levels. The number of clusters and pixels that assemble the control set are parameters that require tuning and are further discussed in the Section 4-D. At the end of the process, the predicted background image is built as a mosaic by a superposition of the different individually predicted image clusters. The clustering process is demonstrated in Fig. 3 as an image frame is divided to 6 parts and from each part 2 control pixels are randomly chosen as the control set for estimation of the matching cluster.

(a) (b) (c)

Fig. 3. (a) Original image, clustered to 6 parts in (b). (c) shows 2 random pixels taken from each cluster as control set.



(a) (b) (c) (d)

Fig. 4. FG-BG Separation results: (a) original frame, (b)- BG, FG results before using heuristics (c) and after (d).

To conclude, the results of applying the mentioned heuristics are shown in Fig 4, where appears an image (a), its estimated BG (b) and FG before adding the mentioned heuristics (c) and after (d). Notice how many misclassified pixels as FG, mainly in the upper-right image section (c) are rejected and a more accurate version of the FG is obtained (d).

E. Background Initialization & Maintenance

Typically, the first frames of a video stream serve as a training set for initializing the background model, as many popular techniques need at least dozens of frames [19], [24] to do so. There seems to be a trade-off between the desire to

have a BG model as soon as possible and the need of gathering enough samples in order to produce a more statistically accurate parameters prediction. Furthermore, the training set doesn't necessarily avoid being contaminated by noise or FG objects. Thus, some FG pixels that appear in the training set may be partially absorbed into the BG model creating ghost-like effects. Since we are using N pixels in the control set, N measurements would yield a linear dependency, so we need only N past frames to train our background model. The size of the control set is a parameter we tuned in our experiments and found that $N = 5$ is suffice. A common frame rate for video is 30 fps (frames per second), so we can initialize our BG model after 0.167 seconds. Setting up a BG model is an important feat [8] and being able to do it so early is one of the suggested method's main advantages in compare to other leading methods.

A scene's BG is expected to change during a video, caused by local state change (*e.g.* a parking car) or illumination changes. In the case where dynamic visual information suddenly turns into static (and vice versa), if the model was calculated before the BG has changed, then there is a need to update the BG model to avoid misclassifications. Modifying the predictors adaptively improves considerably the accuracy of the constructed models. A robust system needs to detect these situations and absorb them into the BG model. We achieve this by defining a learning rate variable $\alpha \in [0, 1]$ which updates our model parameters in each consecutive frame.

$$M_{i+1} = \alpha M_i + (1 - \alpha) D_{i+1} \tag{14}$$

where M_i is the model until frame i and D_i represents the parameters from the i^{th} frame. In our approach, we have two adaptive elements: (a) The linear predictor itself that uses the current control pixels values to estimate the scene. Doing so enables the method to discern the changing illumination effects by estimating how an image should look like by evaluating the changes among the control set pixels and their relations with any other pixel in the scene. (b) Learning rate update rules adjusts the parameters in our model in an incremental fashion to absorb changes. In fig. 5 we show an example on "Light Switch" video taken from LIMU dataset [52]. The video exhibits varying and sudden illumination changes. In such a case, which is quite common for indoor surveillance cameras, it is necessary to adapt the BG model by considering the current illumination status in order to keep the BG model relevant to the current scene's condition. In the upper row are some frames with diverse lighting conditions and the corresponding BG estimation our method yielded beneath them. Note how similar the BG images are to the original frames. Also, where there is an FG object (2 right columns), the BG remains intact.

F. Foreground Detection

Foreground Detection is typically referred as the "answer" of a FG-BG separation algorithm. It creates a mask - a binary video where 0's mark BG locations and 1's indicate FG pixels. Traditionally this is obtained by hard thresholding the absolute difference of a frame and its corresponding BG reconstruction using a fixed value which is chosen empirically (Eq. 15).

$$FG_i = |I_i - BG_i| > \tau \tag{15}$$

Driven by the fact that each pixel's distribution isn't stationary, we assume that its pdf can be approximated by a Gaussian with time varying statistic moments. We calculate for every pixel its first and second empirical moments of previous frames (up to 150) and update the moments according to new observations as the video advances. The decision to classify a new pixel observation is done *locally* for each pixel by:

$$FG(x, y, t) = |I(x, y, t) - BG(x, y, t)| > 3 \cdot \sigma(x, y, t) \tag{16}$$

where $\sigma(x, y, t)$ is the standard deviation of pixel (x, y) at time t. This unique thresholding resembles outlier detection mechanisms used in local-adaptive methods [9], [18], as for every pixel in each frame the classification is done individually according to its distribution. In the bottom row of Fig. 5 we show the FG mask created for each frame.

G. Algorithm

Our model has the following parameters:

1) $\#_{cs}$ - Number of control sets.
2) $\#_k$ - Number of clusters to divide the initial background.
3) $\#_{cp}$ - Number of control pixels.
4) α - Learning rate for model update.

Algorithm 1 DSPB

1: Choose model parameters {default: $\#_{cs} = 3, \#_k = 1, \#_{cp} = 5, \alpha = 0.99$}.
2: Given a video use first N frames for initialization.
3: Construct DSPB's $\#_{cs}$ linear predictors using $\#_{cp}$ randomly chosen pixels in $\#_k$ image clusters.
4: Obtain DSPB's model components (expectations and covariance matrices) empirically using the training set.
5: **while** $N < i < frames$ **do**
6: BG_i = model estimation
7: FG_i = local outliers of $|Frame_i - BG_i|$
8: FG_i = post processing of (FG_i)
9: update DSPB's model parameters with learning rate α
10: **end while**

IV. Experiments & Results

In the following section, we examine scenarios in which global-linear and local-adaptive methods tend to struggle. Later, we tune our method by optimizing the model parameters, establish a post-processing technique for it and to other compared algorithms. We end with an evaluation on a few public datasets with other state-of-the-art methods and discuss the outcomes. For all the experiments described in this chapter, we used a computer with 4-cores, i7 processor with 16GB RAM memory. The coding was done with MATLAB 2017a.

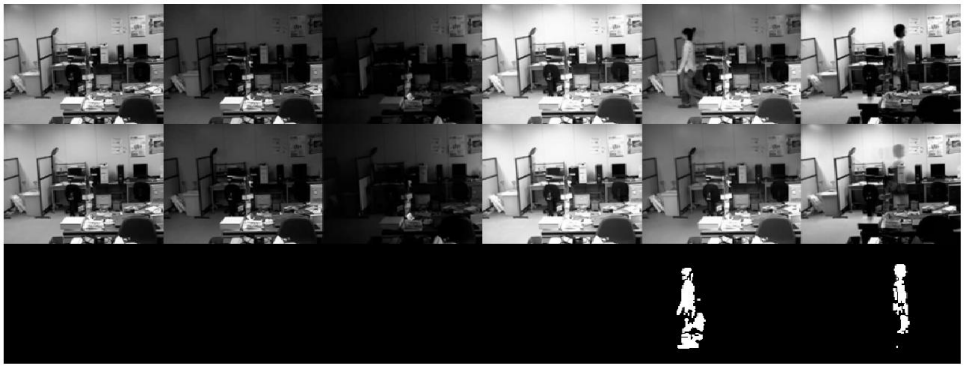


Fig. 5. Some of "Light Switch" frames with diverse illumination conditions, their background estimation and foreground mask.

A. Global-Linear Methods Analysis

One of global methods main merit is their ability to extract the BG at the frame level, usually by processing a batch of frames and minimizing some energy criteria. Hence, global methods prefer the background to be static with minor gradual changes. In long videos (*e.g.* surveillance cameras, which is the main use for FG-BG separation) the illumination varies continuously which results in a dynamic scene appearance at different timestamps. A popular and representative global method is to apply a version of PCA to distinguish the BG by taking a large chunk of a video and extracting a low rank image while requiring heavy computational resources, since it is necessary invert an $N \times M$ matrix (assuming there are N pixels in each image and M images). The computational burden, in addition to the need to acquire a significant number of frames in advance, make global methods unable to reach real-time performance. Since frames are projected onto the low rank version of the scene, global methods tend to handle well with instant illumination changes, but gradual ones do not match the linear components of the model. To demonstrate the ability of global methods to hold a firm background representation, we sampled an hour-long video of a beach during sunset. In the video the sun changes its location consistently and is occluded occasionally by clouds, resulting in a video with both rapid and gradual illumination changes. We took 7200 consecutive video frames, cropped a (100×100) patch and computed its eigenvalues from 50 frames sampled uniformly in different time intervals. The scenery is shown in two frames in Fig. 6 (a) and (b). Notice how the sun is occluded in different ways that result in changing illumination. At the left-bottom corner a yellow rectangle marks the patch we cropped from the images to analyze, as a few patch frames demonstrate the different lighting conditions in Fig. 6 (c-e). If the patch size was solely one pixel or had only lambertic materials exhibited in it, then the dimension would have been simply one. But as the patch gets bigger and holds more objects with varying reflectivity properties, then there is a need for a more complicated model to properly describe the changes the scene undergoes.

In Fig. 7 we show the percentage that cumulative eigenvalues hold at different time interval sizes (as number of frames). The trend is quite clear - as the time interval gets

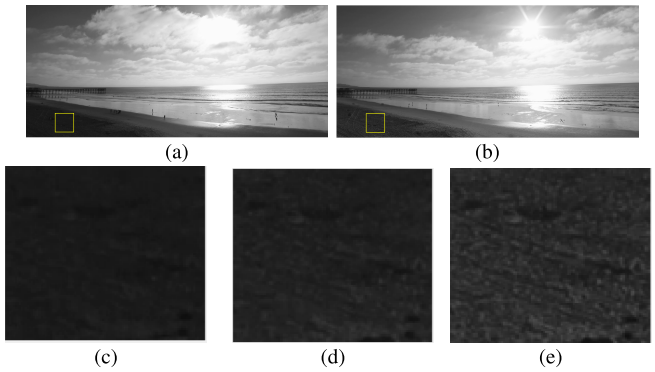


Fig. 6. (a-b) Scene under different settings with tested patch in yellow at bottom-left corner. (c-e) Patch examples with various lighting conditions.

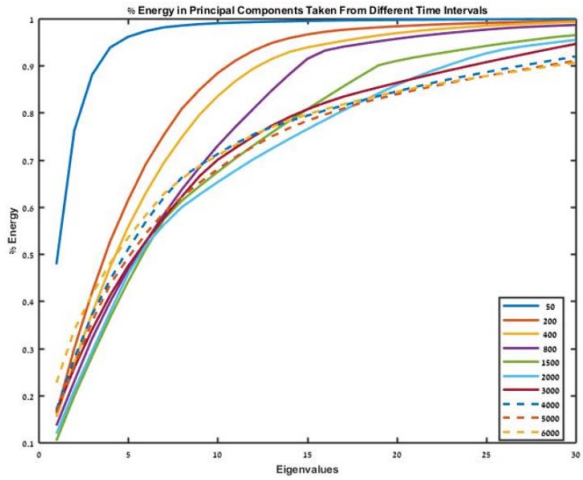


Fig. 7. Energy percentage in cumulative eigenvalues as taken from different interval sizes (in frames).

larger, more eigenvalues are needed in order to describe the same percentage of energy of the total scene that smaller intervals can explain using just a few. While the time interval increases, more variations in the scene occur (derived from different illumination conditions) and impairs the ability of top eigenvectors to successfully describe the scene. To conclude, the linear subspace created by global methods is usually able to handle sudden illumination changes, since the BG model

is trained on a large portion of the video which exhibited these illumination conditions. The frames containing sudden lighting changes lies on a component from the linear subspace as spanned by the low rank components. The problem arises while there are gradual illumination changes, because their projections do not naturally fit a component in the low rank representation. The solution to this situation might be reached by accumulating more eigenimages to compose a richer representation of the BG, but this results in more computational demands and absorbing FG objects in the BG.

B. Local-Adaptive Methods Analysis

Local methods treat the video sequence as an independent collection of 1D signals. This self-sufficient approach analyzes each pixel's temporal behavior separately, while neglecting any spatial information. State-of-the-art methods fit a GMM to each pixel that is able to explain most of the pixel's variability. Almost each local method has a user-defined learning rate $\alpha \in [0, 1]$ that adds adaptive qualities to the background model and controls how a new value is proportionally absorbed into the BG model to keep it relevant to the current scene conditions. Local methods excel in dealing with dynamic background (*e.g.* waving trees) and minor gradual illumination changes. The problems start when the BG changes globally in a manner that cannot be modeled by high frequencies. Forcing a more aggressive learning rate may help adjust to a new scene appearance but it also adds many false detections to the FG mask. The problem is more dominant for indoor scenarios which often express rapid light changes (*e.g.* screens and light switches) that impact the scene significantly. To illustrate this issue, we used "light switch" video from LIMU dataset [52] that is consist of people crossing an office with sudden light changes. We fit various Gaussians and learning rates based on the first 200 frames. In Fig. 8 (a-c) we show 3 frames that express the BG, the BG under sudden illumination change and a FG object and examine how GMM classifies the latter two frames. Fig.8 (d) shows the results for classifying Fig.8 (b) as columns represent number of Gaussians used in the model for each pixel and rows demonstrate the effect of different learning rate values. As can be seen in fig.8 (a,b), the tested frame is significantly different than the complete scene solely due to illumination changes. Hence, we expect a low number of Gaussians to fail in generalizing the light changes and consequently classify many BG pixels as FG. Fitting 5 and 7 Gaussians reduces the misclassification rate, but still there is a significant amount to even consider using a GMM in these scenarios. The only way GMM can handle better this situation is by applying a more aggressive learning rate to absorb the changes by updating the model parameters much faster by giving recent frames more weight in the total BG representation. Enlarging the Gaussians number does not improve enough the classification outcome. This is understandable since the frames on which the models were trained did not evidence such dramatic illumination changes so pixel values are detected as anomalies. In Fig.8 (e) the same setting was taken but with another frame with an FG object and novel illumination conditions. Enlarging the

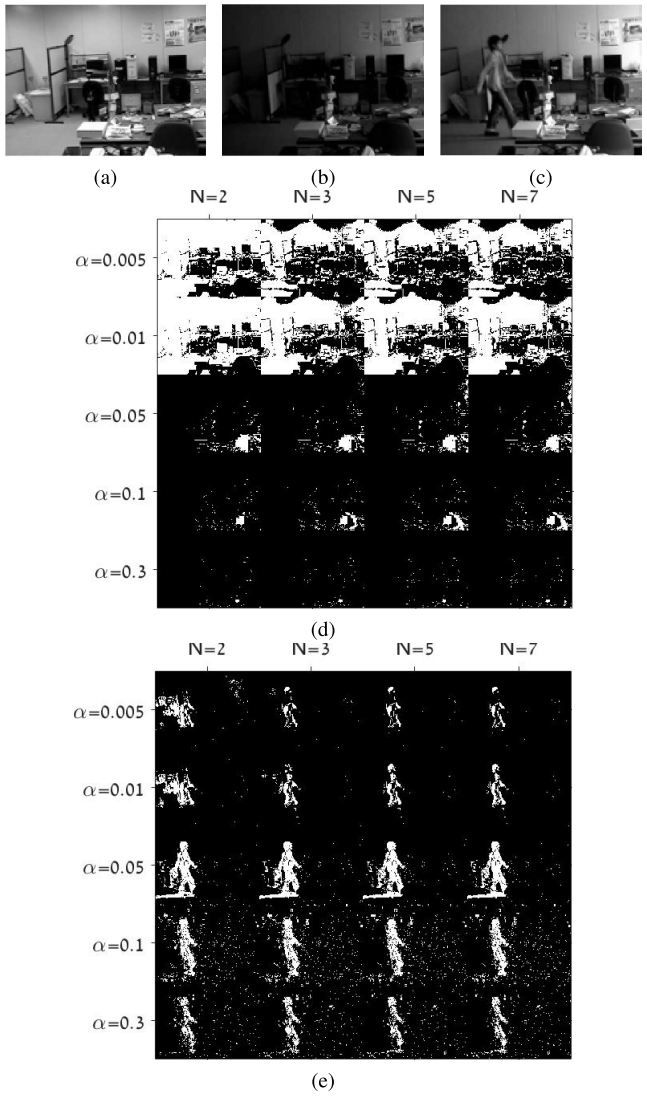


Fig. 8. Testing the effect of different learning rates (α) and no. of Gaussians (N) in the pixel model: (a) Original frame. The scene under different lighting conditions: (b) only BG, (c) with FG object, and their FG masks in (d) & (e) respectively.

Gaussians number reduces the number of anomalies detected, since the models are richer and can describe better the scene variability. The best result is achieved at $N = 5, 7$ and $\alpha = 0.01, 0.05$ but when using these values in the previous tested frame there were many misclassified pixels. Increasing the learning rate further helped handling the illumination changes but it also distorts the result by adding noise.

By observing these two test cases, we conclude that fitting a pdf pixelwise is capable of dealing well with many changes in the BG that can be modeled as repetitive interference (*e.g.* moving trees or waves), but more drastic changes in the scene, like illumination, causes the system to misclassify many pixels and using a larger learning rate trades-off between a better BG and a worse FG. The ideal learning rate value is hard to tune for each video individually and remains one of the major drawbacks in applying local methods for FG-BG separation.

C. Experiment Setup

1) Metrics for Evaluation: As stated by Goyette *et al.* [53], it is not a trivial task to find the right metric to accurately measure the ability of a method to distinguish between FG and BG. If we consider BG segmentation as a classification process, then we can recover the following four quantities: True Positive (TP): The algorithm marks correctly a pixel as FG; False Positive (FP): Pixels are set to be FG mistakenly; True Negative (TN): A BG pixel is classified correctly as BG; False Negative (FN): A pixel is classified to BG but it should have been FG. Many evaluation metrics combine all or some of the quantities. Specifically, we use:

- Precision: $Pr = \frac{TP}{TP+FP} \in [0, 1]$, from all the algorithm classifications as FG, the percentage it is correct.
- Recall: $Re = \frac{TP}{TP+FN} \in [0, 1]$, from all the pixels that should have been classified as FG, the percentage that the algorithm classified correctly.
- F measure: $F = \frac{2 \cdot Pr \cdot Re}{Pr+Re} \in [0, 1]$
- fps - Frames processed per second (real-time$> 20\ fps$)

2) Compared Methods: In our experiments we compare our method with a basic method and to more 8 popular state-of-the-art algorithms: (1) Thresholding frame difference; (2) Codebook algorithm [25]; (3) Kernel-Density Estimation (KDE) [19]; (4) Linear Binary Pattern (LBP) [54]; (5) GMM (zGMM) [18] with a flexible amount of Gaussians; (6) ViBe [21]; (7) SubSENSE [20]; (8) SC_SOBS [37]; (9) Augmented Lagrangian approach (ALM) as a representative of the RPCA family [55] that is considered to be a relative fast RPCA-based method according to [56] which reviews dozens of recent global methods. For ALM implementation we use the available version at [57]. All the rest methods implementations are taken from the BGS library [7] which is a C++ library embedded with OpenCV using a MATLAB wrapper.

D. Model Optimization

In our setup, we downsample the image resolution to half of the original size (to 120×160) due to hardware limitations. We are aware that this might leave small FG objects undetected. However, most of real scenes passing objects aren't affected. To tune our method we use LIMU dataset [52].

In Table I we show the best parameter formation that yielded the top F-measure value for each video separately, by scenario (indoor/outdoor) and in total. It is interesting to see that each video used a different parameter setup with similarity between the values used in indoor scenes (videos 3,5) and a different set of values for outdoor videos (1,2,4). The number of pixels required for prediction varies from 5 to 15 and the learning rate value is lower in indoor settings. To summarize, tuning model parameters is a crucial step in developing a system. Due to the richness of natural scenes, there is a unique parameter set for each video. As default values, we recommend using: $\{\#_{cs} = 3, \#_k = 1, \#_{cp} = 5, \alpha = 0.99\}$, and these values are used in further experiments in this chapter.

E. Post-Processing Technique

Post-processing the obtained FG frames is typically recommended [58], mainly by morphological operations, due to the many outliers that contaminate the FG mask. We apply and compare the results of the following post-processing strategies: raw FG, spatial medians ($3 \times 3, 5 \times 5, 7 \times 7$), morphological closing (using a square structure element with 4 pixels), morphological opening, filling holes (FG object with BG "blobs" inside are "filled"), remove connected components of 3 pixels or less, closing + filling holes and opening + filling holes. In Table II we compare the methods mean F-measure on LIMU dataset [52] using the mentioned post-processing techniques and show the best strategy for each method in bold. Applying morphological closing followed by filling holes along with 5×5 median yields the best results on the methods majority. This is not surprising, since in FG objects not all the pixels inside are always classified as FG, so filling the gaps enhance the amount of TP. Median filtering using a small spatial window removes scattered false alarms that helps methods with many outliers. All in all, for all methods applying a post-processing technique improved the mean F-value for about 0.05, which is a significant improvement obtained almost effortlessly. From this point on while reporting results we apply the best post processing strategy for each method.

TABLE I

TOP PARAMETER FORMATIONS RESULTS UNDER DIFFERENT SCENE SETTINGS

	Model Parameters				Evaluation Metrics			
Scene	$\#_{cs}$	$\#_k$	$\#_{cp}$	α	Pr	Re	**F**	fps
Video#1	1	3	7	0.9999	0.455	0.275	**0.343**	749
Video#2	5	1	15	0.999	0.866	0.361	**0.509**	1264
Video#3	3	1	5	0.99	0.888	0.371	**0.524**	347.8
Video#4	5	10	5	0.999	0.91	0.417	**0.571**	256.5
Video#5	3	1	7	0.99	0.749	0.731	**0.744**	173
Indoor	3	1	5	0.99	0.796	0.559	**0.624**	217.96
Outdoor	5	3	3	0.9995	0.684	0.349	**0.461**	91.77
Total	**3**	**1**	**5**	**0.99**	0.836	0.396	**0.503**	132.7

F. LIMU Dataset

LIMU dataset [52] contains 5 videos with (240×320) resolution, while 4 videos have 5000 frames and another 2800. The dataset contains 3 videos from traffic surveillance cameras and two indoor scenes with varying illumination. Ground truth is available every 15 frames starting from the 500th frame. Table II summarizes the metric results for each method and the mean F-measure with relation to the computational speed (frames processed per second) are exhibited in Fig. 9, as they are the two main considerations while choosing a FG-BG method. From all the tested methods, RPCA had the largest F-measure, but also was the less performing method by its runtime - 0.48 fps. The tested RPCA method, ALM, is relatively fast in its domain, but the fact that it mainly serves as an offline analysis (or the very least acquires a large batch before processing it) makes it an unappealing choice for embedding on surveillance cameras and could fit other usages where online analysis is not crucial. Most other methods achieved F-measure result between 0.55-0.6. As we examine the computation speed, we can notice that all except RPCA perform in real time pace (above 30 fps). DSPB turns out to be the fastest method which is even more impressive

TABLE II

F VALUES FOR FG-BG METHODS ON LIMU DATASET WHILE TRYING VARIOUS POST-PROCESSING TECHNIQUES, THE BEST STRATEGY MEAN VALUES ON THE WHOLE DATASET ARE BELOW THE THICK LINE

Method	FrameDiff	Codebook	KDE	LBP	zGMM	ViBe	SuBSENSE	SC_SOBS	RPCA	DSPB
Raw	0.3981	0.4738	0.5403	0.5879	0.5161	0.4121	0.5685	0.5665	0.6586	0.5356
3x3Med	0.3829	0.5502	0.5687	0.5919	0.5097	0.3775	0.5678	0.5917	**0.7226**	0.5138
5x5Med	0.3593	**0.5649**	**0.5721**	**0.5942**	0.4951	0.3485	0.5650	**0.6012**	0.7200	0.4809
7x7Med	0.3384	0.5635	0.5681	0.5935	0.4757	0.3215	0.5615	0.5997	0.7048	0.4491
Closing	0.4478	0.4853	0.5543	0.5879	0.5478	0.4393	0.5686	0.5778	0.6865	0.5728
Opening	0.3286	0.5350	0.5496	0.5893	0.4754	0.3529	0.5682	0.5778	0.6825	0.4786
Holes	0.4051	0.4795	0.5451	0.5879	0.5182	0.4128	0.5685	0.5707	0.6700	0.5378
LowCC	0.3902	0.5047	0.5539	0.5882	0.5119	0.4026	0.5685	0.5692	0.6917	0.5292
Close+Holes	**0.4790**	0.4895	0.5662	0.5879	**0.5743**	**0.4522**	**0.5686**	0.5879	0.6892	**0.5952**
Open+Holes	0.3412	0.5505	0.5644	0.5893	0.4831	0.3554	0.5682	0.5912	0.6875	0.4844
mean-F	0.4790	0.5649	0.5721	0.5942	0.5743	0.4522	0.5686	0.6012	**0.7226**	0.5952
Pr	0.6681	0.4589	0.5713	0.6898	0.5496	0.5856	0.5725	0.5094	0.7663	**0.8475**
Re	0.3734	**0.7347**	0.5729	0.5219	0.6014	0.3683	0.5648	0.7332	0.6836	0.4587
fps	217.2349	184.6500	162.8817	81.6114	190.5797	207.7631	35.1585	141.0229	0.4827	**252.1294**

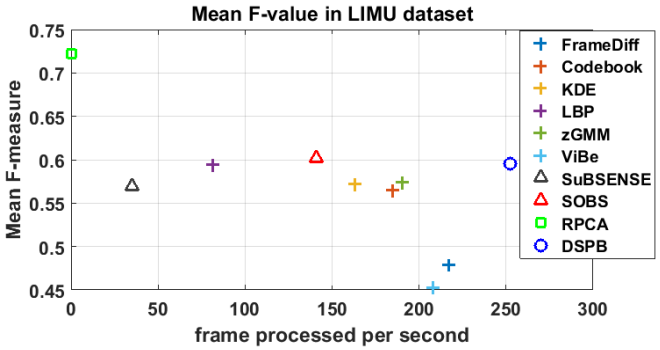


Fig. 9. LIMU mean F-value and frames processed per second.

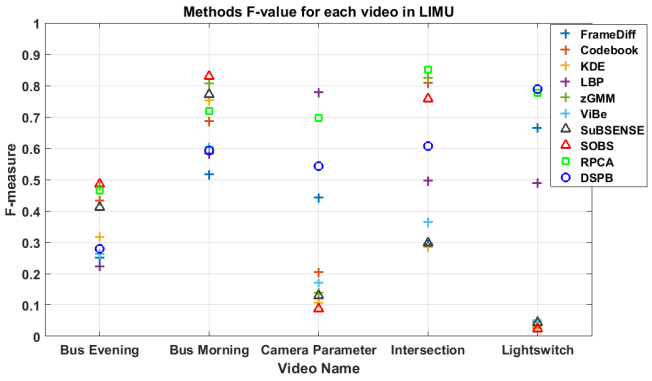


Fig. 10. LIMU mean F-value for FG-BG methods on each video separately.

considering it was implemented in MATLAB while the other methods implementations are in C++. It is a significant feat since low resource demanding methods allow further video analysis algorithms to be embedded on surveillance cameras, as FG-BG separation is typically just the first phase in an image processing pipeline.

Due to the rich and various scenarios in LIMU dataset, it is intriguing to investigate results on each video individually. Fig 10 shows the F-measure of each video separately and sheds more light on the strengths and weaknesses of each method. The videos 'Bus Evening', 'Bus Morning' and 'Intersection' (in the figure columns 1, 2 & 4 from the left) are outdoor scenes, while 'Camera Parameter' and 'Light switch' are indoor (3 & 5 from the left). RPCA remains in the top 5 methods in all videos, which shows the superior performance of global methods but as discussed before, they lack in computation speed. Among the local methods it is very notable that their results are much higher on outdoor scenes in compare to indoor ones. This observation matches the descriptive experiments done in section 4-B. LBP and simple frame difference performed relatively well on the indoor scenes and surpassed many local methods. This is understandable since LBP is a popular texture feature based method, that considers the surrounding area around each pixel, so illumination changes are handled well. Frame difference literally subtracts consecutive frames that eliminate illumination changes in a very simple way. DSPB is in the top 3 methods at indoor scenes as it utilizes pixel correlations to predict BG in varying light situations. When dealing with outdoor scenes, DSPB reached a decent place in the middle as illumination changes are less dominant. By reviewing DSPB's general performance, it verifies our claim that it acts as a hybrid model of global and local methods.

G. I2R Dataset

I2R dataset is comprised of 9 videos with resolution of (120 × 160) and a varying number of frames between 523-3584 as ground truth is available for 20 frames in each video. The scenes described are mainly outdoor with repetitive motions (*e.g.* waving trees, water ripples fountains and escalators). A few other scenes are taken from surveillance cameras indoors while video 6 exhibits illumination changes [59]. Thus, we expected in this dataset that local-adaptive method will surpass global-linear and DSPB in their performance, as the scenes are more oriented to their strengths. During the experiment we neglected "bootstrap" video as it appears also in Wallflower set as tested in the next section. In Table III we show the F-measure of each video separately, followed by the whole dataset mean performance by F-measure, Precision, Recall and fps. As anticipated, local-methods show their advantage while handling

TABLE III
METHODS F-MEASURE ON EACH VIDEO ON I2R DATASET. BELOW THE THICK LINE ARE MEAN METRICS ON THE DATASET

Method	FrameDiff	Codebook	KDE	LBP	zGMM	ViBe	SuBSENSE	SC_SOBS	RPCA	DSPB
Video 1	0.2971	0.2027	0.1457	**0.7988**	0.3496	0.4967	0.7876	0.1990	0.4816	0.3772
Video 2	0.4393	0.8021	0.4637	0.7049	0.7500	0.7102	**0.9037**	0.8896	0.7909	0.4466
Video 3	0.2850	0.4860	0.1955	0.5449	0.6262	0.6214	0.6960	0.3397	**0.7691**	0.3008
Video 4	0.3415	0.4233	0.4315	0.7678	0.5305	0.6271	**0.8255**	0.3869	0.5534	0.6462
Video 5	0.5348	0.5824	0.2881	0.5187	0.6737	0.6717	**0.8091**	0.6878	0.6570	0.5266
Video 6	0.3880	0.4080	0.2007	0.3624	0.1941	0.1499	0.3126	0.2303	**0.7238**	0.5655
Video 7	0.6263	0.5601	0.4061	0.5949	0.6677	0.5382	0.7376	0.6074	**0.7648**	0.6210
Video 8	0.1868	0.8471	0.8684	0.3897	0.8862	0.8617	**0.9049**	0.8468	0.4870	0.3988
mean-F	0.4729	0.5672	0.4138	0.5997	0.6348	0.5958	**0.7662**	0.5755	0.6787	0.5221
Pr	0.4563	0.4251	0.2649	0.5072	0.6198	0.7191	**0.7714**	0.4217	0.6266	0.6316
Re	0.4907	0.8522	0.9455	0.7334	0.6505	0.5086	0.7610	**0.9058**	0.7403	0.4450
fps	258.7768	268.2934	221.9644	84.4726	295.3513	**348.4463**	34.4911	176.3129	0.9291	222.7974

TABLE IV
MEAN METRIC VALUES ON WALLFLOWER DATASET (WITHOUT VIDEO #5) BY METHODS

Method	FrameDiff	Codebook	KDE	LBP	zGMM	ViBe	SuBSENSE	SC_SOBS	RPCA	DSPB
mean-F	0.4211	0.5084	0.6539	**0.7883**	0.5827	0.5198	0.7170	0.6012	0.5949	0.5034
Pr	0.6196	0.4134	0.5155	**0.8099**	0.6130	0.7039	0.7264	0.5400	0.6647	0.6969
Re	0.3190	0.6601	**0.8937**	0.7677	0.5552	0.4120	0.7078	0.6781	0.5383	0.3940
fps	73.2652	**183.8477**	109.0932	52.7992	187.7175	219.5518	19.3017	111.9749	0.2938	86.2614

disturbances with a repetitive manner, due to their ability to model them as high frequencies in the model fit phase. SuBSENSE and RPCA seem to yield the highest results as they also lead in consuming computational resources. RPCA's frame rate leaves it to serve as on offline analysis, while SubSENSE is on the brink of performing in real time. Other methods including DSPB are capable in processing hundreds of frame per second in the current resolution which shows their strength.

H. Wallflower Dataset

Wallflower dataset [16] has 7 short videos with a resolution of (120×160) and varying number of frames (from 250-5000). Each video demonstrates a different challenge for FG-BG separation. The dataset has only 1 frame of ground truth, therefore, we use this set to show a visual result. The videos are designed to challenge FG-BG methods in extreme conditions as they exhibit problematic situations:

1) Bootstrap - No clean frames for training (FG objects are in the scene from the start)
2) Camouflage - Static FG covers dynamic BG.
3) Foreground Aperture - State change: BG to FG.
4) Light switch - Sudden illumination changes.
5) Moved Object - State change: FG to BG.
6) Time of Day - Gradual illumination changes.
7) Waving Trees - Dynamic BG.

Since video #5 ("Moved object") ground-truth frame has only BG pixels, it isn't possible to calculate Pr, Re and F-measure. Thus, the mean measures in Table IV and individually examined video graph (Fig 12) are shown without it.

In Table IV the mean F-measure is presented. We added the mean results table mainly for the sake of completeness of our work, but the fact that it is based on a single ground-truth image per video makes the numbers less meaningful, hence, we focus more on the visual results. To demonstrate a better how methods perform, we show in Fig. 11 the obtained FG masks in comparison to the original and ground-truth. The original frame from each video is in the left column, followed by the ground-truth (hand segmented) and then the tested algorithms FG masks. From a general point of view, it is clear why BG-FG separation is not considered to be a solved problem - each method has its merits and drawbacks, while performing well in some scenarios and lacking in others. Videos 4 and 6 exhibit illumination changes, sudden and gradual correspondingly. Note how local methods deal with the sudden light change: zGMM classifies most of the scene as FP, while ViBe absorbs the values very aggressively and incorporates the FG into the BG model. Again, LBP and simple frame difference show their strength in handling varying illumination conditions, as well as RPCA and DSPB. Video 1 doesn't have clean frames for training, resulting in the BG model being mixed with FG objects. This makes it hard for classifying as all methods express this difficulty. Dynamic BG covered by a static FG is demonstrated in Video 2. In the video, a person enters the scene and covers a flickering screen. Since the FG object stays for a decent amount of time, RPCA, DSPB and frame difference already absorbed a large portion of it (besides the edges) into the BG. Local methods seem to handle better this disturbance as their absorption strategy is slower. If the FG object stays longer it will get eventually absorbed into the BG model due to the usage of learning rates. On the other hand, Video 5 demonstrates a situation where a BG object moves to a new location and should be merged back to the BG after the change. Here learning rates achieve this goal, but it is interesting to witness how successful methods on Video 2 lacked performance on Video 5 (except for LBP). In Video 7 there are repetitive BG disturbances as waving trees grasp a large part of the video. Here local methods have their opportunity to shine and ViBe, zGMM, SuBSENSE and LBP seem to work well. The mean F-measure value for each

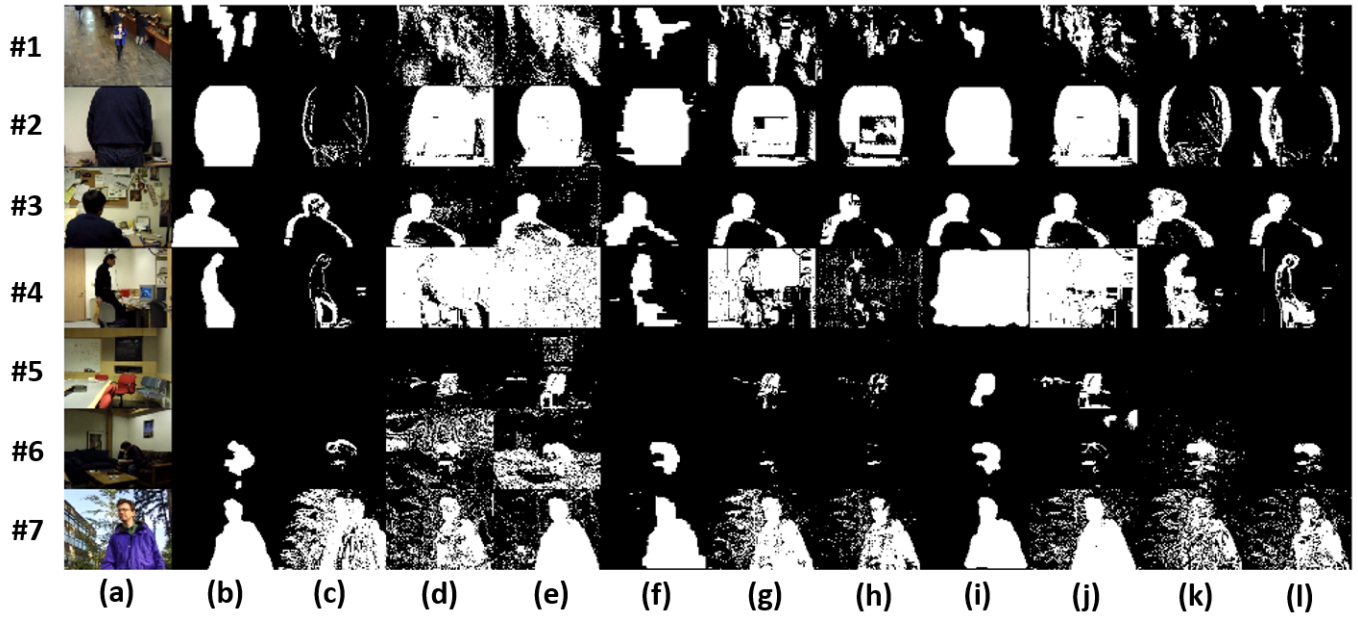


Fig. 11. Wallflower videos visual results. Rows indicate videos. Columns: (a) Original frame, (b) Ground Truth, (c) Frame difference, (d) Codebook, (e) KDE, (f) LBP, (g) zGMM, (h) ViBe, (i) SuBSENSE, (j) SC_SOBS, (k) RPCA, (l) DSPB.

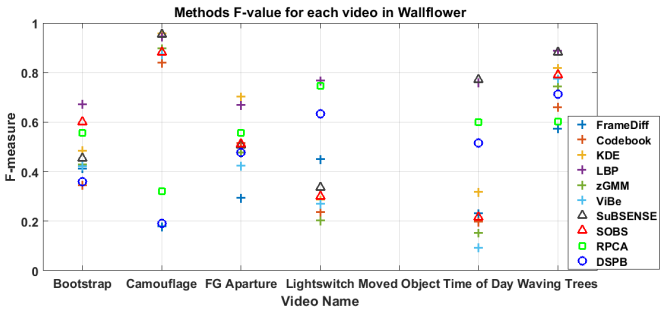


Fig. 12. F-value for each video individually on Wallflower dataset (besides #5).

video (except 5) is shown in Fig 12. The results reinforce the visual investigation in the previous paragraph and mostly agree with the insights about analyzing LIMU and I2R datasets but taken to the extreme, due to the design of Wallflower dataset. RPCA, DSPB and LBP are more successful in scenes with illumination changes, while local methods perform better on outdoor scenes, especially when dynamic background appears.

V. Conclusion

In this work we introduced a novel FG-BG separation method. DSPB is derived by a physical model of how illumination forms a scene. Hence, the suggested background estimation method can handle well with illumination changes (sudden or gradual), which are considered to be the main source of variation in a given scene that does not derive from foreground objects. Due to the inherent simplicity of DSPB, it can deliver real-time performance using basic hardware, which makes it tractable for usage on surveillance cameras that are the main use of such a system. The main novelties of our system are: (1) The use of pixel correlations - far or near, (2) Creating an initial BG model as soon as the 5th frame, (3) A computationally tractable method (4) Using spatial information combined with pixel temporal statistics, yielding a hybrid model that incorporates desired qualities from local and global methods. Modeling a scene appearance accurately despite large illumination variations has many potential applications. The suggested prediction model of scene appearance is novel, and the scheme is very accurate and efficient computationally. We show the method merits on an application for video FG-BG separation, but it may be successfully incorporated in other image processing tasks like change detection, video compression and video inpainting.

References

[1] W. Kim and C. Jung, "Illumination-invariant background subtraction: Comparative review, models, and prospects," *IEEE Access*, vol. 5, pp. 8369–8384, 2017.

[2] L. P. J. Vosters, C. Shan, and T. Gritti, "Background subtraction under sudden illumination changes," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2010, pp. 384–391.

[3] Y. Tocker, R. R. Hagege, and J. M. Francos, "Dynamic spatial predicted background for video surveillance," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 4005–4009.

[4] S.-C.-S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Proc. SPIE*, vol. 5308, pp. 881–892, Jan. 2004.

[5] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. CVPR*, Jun. 2011, pp. 1937–1944.

[6] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection—A systematic survey," *Recent Patents Comput. Sci.*, vol. 4, no. 3, pp. 147–176, Sep. 2011.

[7] A. Sobral, "BGSLibrary: An OpenCV C++ background subtraction library," in *Proc. 9th Workshop De Visão Comput. (WVC)*, Rio de Janeiro, Brazil, Jun. 2013, pp. 1–3.

[8] T. Bouwmans, L. Maddalena, and A. Petrosino, "Scene background initialization: A taxonomy," *Pattern Recognit. Lett.*, vol. 96, pp. 3–11, Sep. 2017.

[9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 1999, pp. 246–252.

[10] P. D. Z. Varcheie, M. Sills-Lavoie, and G.-A. Bilodeau, "A multiscale region-based motion detection and background subtraction algorithm," *Sensors*, vol. 10, no. 2, pp. 1041–1061, 2010.

[11] B. Lee and M. Hedley, "Background estimation for video surveillance," in *Proc. Image Vis. Comput. New Zealand (IVCNZ)*, 2002, pp. 315–320.

[12] D. M. Ha, J.-M. Lee, and Y.-D. Kim, "Neural-edge-based vehicle detection and traffic parameter extraction," *Image Vis. Comput.*, vol. 22, no. 11, pp. 899–907, Sep. 2004.

[13] J. Zheng, Y. Wang, N. L. Nihan, and M. E. Hallenbeck, "Extracting roadway background image: Mode-based approach," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1944, no. 1, pp. 82–88, Jan. 2006.

[14] K. Karmann and A. Von Brand, "Moving object recognition using an adaptive background memory," *Time-Varying Image Process. Moving Object Recognit.*, vol. 2, pp. 289–296, 1990.

[15] T. Chang, T. Ghandi, and M. Trivedi, "Vision modules for a multi sensory bridge monitoring approach," in *Proc. ITSC*, 2004, pp. 971–976.

[16] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 255–261.

[17] C. Wren and A. Azarbayejani, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[18] R. Chang, T. Gandhi, and M. M. Trivedi, "Vision modules for a multi-sensory bridge monitoring approach," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst. (ICPR*, Oct. 2004, pp. 1051–1054.

[19] A. Elgammal and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2000, pp. 751–767.

[20] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[21] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[22] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2003, pp. 65–72.

[23] P. W. Power and J. A. Schoonees, "Understanding background mixture models for foreground segmentation," in *Proc. Image Vis. Comput. New Zealand*, 2002, pp. 10–11.

[24] K. Kim, H. T. Chalidabhongse, D. Harwood, and S. L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-Time Imag.*, vol. 11, pp. 172–185, Jun. 2005.

[25] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU-Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, Aug. 2010.

[26] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," in *Computer Vision Systems* (Lecture Notes in Computer Science), vol. 1542. Berlin, Germany: 1999, pp. 255–272.

[27] J. Pilet, C. Strecha, and P. Fua, "Making background subtraction robust to sudden illumination changes," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 5305, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-540-88693-8_42.

[28] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[29] A. Sobral, T. Bouwmans, and E. H. Zahzah, "Comparison of matrix completion algorithms for background initialization in videos," in *Proc. Workshops New Trends Image Anal. Process. (ICIAP)*, 2015, pp. 510–518.

[30] A. Sobral and E.-H. Zahzah, "Matrix and tensor completion algorithms for background model initialization: A comparative evaluation," *Pattern Recognit. Lett.*, vol. 96, pp. 22–33, Sep. 2017, doi: 10.1016/j.patrec.2016.12.019.

[31] S. Javed, S. Ki Jung, A. Mahmood, and T. Bouwmans, "Motion-aware graph regularized RPCA for background modeling of complex scenes," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 120–125.

[32] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vols. 11–12, pp. 31–66, May 2014.

[33] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understand.*, vol. 122, pp. 22–34, May 2014.

[34] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, "A probabilistic SVM approach for background scene initialization," in *Proc. Int. Conf. Image Process.*, 2014, pp. 893–896.

[35] B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.

[36] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.

[37] L. Maddalena and A. Petrosino, "The sobs algorithm: What are the limits?" in *Proc. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 21–26.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[39] A. Krizhevsky, I. Sutskever, and E. Geoffrey Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[40] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognit. Lett.*, vol. 96, pp. 66–75, Sep. 2017.

[41] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2016, pp. 1–4.

[42] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction:A systematic review and comparative evaluation," *Neural Netw.*, vol. 117, pp. 8–66, Sep. 2019.

[43] Y. Zhu and A. Elgammal, "A multilayer-based framework for online background subtraction with freely moving cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Oct. 2017, pp. 5142–5151.

[44] D. Sugimura, F. Teshima, and T. Hamamoto, "Online background subtraction with freely moving cameras using different motion boundaries," *Image Vis. Comput.*, vol. 76, pp. 76–92, Aug. 2018.

[45] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible lighting conditions?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1996, pp. 270–277.

[46] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 218–233, Feb. 2003.

[47] E. Prados and O. Faugeras, "Shape from shading," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. Boston, MA, USA: Springer, 2006, doi: 10.1007/0-387-28831-7_23.

[48] B. K. P. Horn, *Robot Vision*, New York, NY, USA: McGraw-Hill, 1986.

[49] R. R. Hagege, "Scene appearance model based on spatial prediction," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1241–1256, Jul. 2014.

[50] D. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*. Reading, MA, USA: Addison-Wesley, 1998.

[51] H. Sajid and S.-C.-S. Cheung, "Universal multimode background subtraction," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3249–3260, Jul. 2017.

[52] S. Yoshinaga, A. Shimada, H. Nagahara, and R. Taniguchi. *LIMU Dataset*. Accessed: Jan. 14, 2018. [Online]. Available: http://limu.ait.kyushu-u.ac.jp/dataset/en/

[53] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A new change detection benchmark dataset," in *Proc. IEEE Workshop Change Detection (CVPR)*, 2012, pp. 16–21.

[54] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[55] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *Math. Program.*, vol. 141, nos. 1–2, pp. 349–382, Oct. 2013.

[56] T. Bouwmans, A. Sobral, S. Javed, S. Jung, and E. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Elsevier*, vol. 23, pp. 1–71, Feb. 2017.

[57] A. Sobral, T. Bouwmans, and E. Zahzah, "LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos," in *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. Boca Raton, FL, USA: CRC Press, 2016.

[58] D. H. Parks and S. S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proc. IEEE 5th Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2008, pp. 192–199.

[59] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. 11th ACM Int. Conf. Multimedia (MULTIMEDIA)*, Jan. 2003, pp. 2–10.