Video Processing
and Understanding
Lab

UNIVERSIDAD AUTONOMA
DE MADRID

***TEC2011-25995 EventVideo (2012-2014)***

*Strategies for Object Segmentation, Detection and Tracking in Complex Environments for Event Detection in Video Surveillance and Monitoring*

# D2.1

# SEGMENTATION FOR STATIC CAMERAS

Video Processing and Understanding Lab
Escuela Politécnica Superior
Universidad Autónoma de Madrid

*Supported by* GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

## AUTHOR LIST

| | |
|---|---|
| *Marcos Escudero Viñolo* | marcos.escudero@uam.es |
| *Diego Ortego Hernández* | diego.ortego@uam.es |

## CHANGE LOG

| Version | Data | Editor | Description |
|---|---|---|---|
| 0.0 | 04-07-2014 | Marcos Escudero Viñolo | Initial version. Chapter 1 |
| 0.1 | 07-07-2014 | Diego Ortego Hernández | Chapters 2.1 and 3.1 |
| 0.2 | 10-07-2014 | Marcos Escudero Viñolo | Chapters 2 3 |
| 0.3 | 11-07-2014 | Diego Ortego Hernández<br>Marcos Escudero Viñolo | Chapter 4 |
| 1.0 | 15-07-2014 | José M. Martínez | Final edition |

# CONTENTS

# 1. Introduction

In scenarios recorded by a static camera, the problem of automatically segment in relevant-and-moving and irrelevant objects—as a two-class problem, it is also sometimes known as object segregation—has been recursively studied. Although Background Subtraction (BS) is not the only technique available for this task—alternatives include motion-compensation **Error! Reference source not found.** and image-scanning approaches [2], it has been, by far, the most used and referenced.

There is a significant quantity of scientific studies that use BS as a primary tool to feed higher-level tasks, including: object tracking, object/people recognition or scene understanding. This multi-task nature leads to two major implications: i) BS has been widely used in many computer-vision applications such as video surveillance, traffic monitoring and human computer interfaces and ii) BS has been exhaustively studied—with up to 160.000 enters in Google Scholar—.

The principle of BS algorithms is to build a model of the *empty* scene (commonly named as *background*) and then detect—and segregate—objects of interest as elements (usually called *foreground*) that do not fit into the background model. According to [3], a BS algorithm can be described by its solutions to the following key-tasks:

**Background initialization:** defines the strategies to initialize the model with a true background image free of foreground objects thus determining an appropriate point of departure for the background modeling stage.
**Background modelling**: describes the nature of the model and associated statistics used to store the *empty* scene—this task is also known as background representation—.
**Background maintenance:** devoted to adapt the model to the changes occurred in the scene over time.
**Foreground detection:** measure the difference between new samples and the model according to a set of features.

This document compiles the contributions to BS developed in the Video Processing and Understanding Lab within the scope of the EventVideo project. We start by briefly describing the remaining challenges in BS as well as the relevant state-of-the-art for each aforementioned key-task. Then, we organize our contributions in a per-stage basis and, finally, we arise a set of conclusions that result in the definition of the future work.

## 1.1. Document structure

This document is composed of the following chapters:

Chapter 1: Introduction to this document.

Chapter 2: State-of-the-art in background subtraction.

Chapter 3: Proposed contributions

Chapter 4: Conclusions and future work.

# 2. State-of-the-Art in background subtraction

This chapter briefly summarized existing techniques for the task of BS. We propose to organize them in a per-stage basis remarking the challenges the aim to resolve. To this aim, we first review the common challenges that should be faced when designing a BS approach.

## 2.1. Challenges

According to the remaining challenges in BS, those identified by Toyama [4] are still the reference. Furthermore, in [3] three new camera-related challenges are included. We propose to organize them in three categories, according to the challenge's source: camera, background and foreground—mixed-originated challenges, as camouflage, are here assigned to foreground—. These can be listed, slightly modifying the nomenclature in [3], as:

***Camera-related challenges***

- *Noisy image*: includes the acquisition-noise in the recording process, the interpolation-noise of resized frames and the block-noise of decompressed videos.
- *Camera jitte*r: when static cameras are placed in non-stable supports—as highway's cameras placed on bridges or poles—wind can make the camera vibrate, which results in nominal motion and—if unconsidered, misdetections—.
- *Camera automatic adjustments*: automatic processes included in some cameras to adapt to scene changes—including refocus, automatic control gain, white balance and brightness control—completely change the background colors respect to those modelled.

***Background-related challenges***

- *Illumination changes*: these are divided in global, which is further subdivided in gradual—daylight in outdoors scenes—and abrupt—switch on and off of lights in indoors scenarios, and local—self-shadows and highlights.
- *Removed background objects*: inanimate background objects can be taken—e.g. stolen— by animated foreground objects—e.g. a person—, leaving a wake—also known as a *ghost*—in the original position.
- *Inserted background objects*: the opposite of *removed background objects*; inanimate objects may be placed in the background. Both situations are especially common in surveillance scenarios.
- *Dynamic backgrounds*: especially in outdoor scenarios some parts of the background may be moving. This motion results in different values—multimodality—to those stored in the model. Common examples of dynamic backgrounds include moving water and waving trees.

***Foreground-related challenges***

- *Bootstrapping*: In crowded scenes part of the background can be occluded for a long time, then hindering the availability of enough samples to model its evolution or even its appearance.
- *Shadows*: whereas background shadows—self-shadows—can be considered an illumination issue, foreground or moving shadows—commonly named cast-shadows—represent a problem as they move as foreground while being represented by lower-intense modes than those in the model.

- *Beginning moving object*: It is sometimes considered the equivalent to *removed background object* for foreground objects. The main difference relies in the object nature, here we usually refer to people, but other examples include cars or animals.
- *Sleeping foreground object*: The parallelism continues with this human-driven version of *inserted background objects*. Even though the decision of incorporate these objects to the background—and then potentially leading to *beginning moving object* situations—or not is task-dependent, people is usually expected to move again. If the foreground object is there since the initialization—and no management of this situation is performed—the challenge is also known as a *hot-start*.
- *Camouflage*: Probably—together with *bootstrapping*—the least studied challenge. Background and foreground objects may share equal—or even similar—appearances, then leading to an inaccurate discrimination process. Obviously this challenge is feature-dependent.
- *Foreground aperture*: This challenge only applies for homogeneous foreground objects that were incorporated to the background. Partial movement of these objects is only detected at the boundaries whereas the interior remain equal to the stored appearance in the model. In our opinion, it is a special sequence of three challenges: *sleeping foreground object*, *beginning moving object* and *camouflage*. However, this also applies to the sequence: *removed background object* and *camouflage*. For both sequences the consequence is the detection of incomplete object regions.

Despite the enormous amount of efforts and studies devoted to solve them, research community agrees [3][4][5][6] that it does not yet exist a system able to solve all of these challenges at the same time. This is mainly due to a tug-of-war between generalist background-modelling and accurate foreground detection; i.e. enhancing approach's flexibility to learn the different background appearances usually harms its ability to adequately discriminate the foreground.

Furthermore, they cannot be solved at the same stage; those related to *Illumination changes* need to be addressed at the modelling and updating stages, and those associated with the foreground density, e.g. *bootstrapping* usually require also specific solutions in the initialization stage, whereas, in our opinion, *camouflage* should be addressed at the foreground detection stage—explicitly by exploring new features and metrics—which is rarely the main topic of BS solutions—which usually rely in color and luminance features—.

## 2.2. Background initialization

Background initialization (BI) is an important stage of the BS algorithms that has been weakly investigated in comparison with the remainder stages [3]. It consists in initializing the background model computing a background image free of foreground objects—True Background: TB—from a training sequence. Background initialization [3] [7-9] is usually also referred as Bootstrapping [3][10], Background estimation [11][12], Background generation [13][14] or Background reconstruction [15]. Several BS approaches in the State-of-the-art use an unreliable scheme to initialize the model, based on the assumption that the TB could be easily captured from the first frames of the sequence. This assumption is incorrect in many video-surveillance scenarios where there may be many foreground objects due to crowds and stationary objects. Therefore, capturing the TB in these situations is not an easy task and BI is a suitable way to tackle it. Furthermore, BI could be very useful to deal with high illumination changes due to the implicit capability to estimate a new background image and re-initialize the background model with it.

First of all, some premises related with BI and the results expected by the algorithms must be defined:

a. Every background pixel should be visible in at least one frame to be considered as background, i.e. if a person is static along the training sequence it should be considered as background.
b. Initial static regions that leave their spatial location during the training sequence should not be considered as background.
c. Moving regions that become stationary during the training sequence should not be considered as background.

Attending to the main issues found in the literature, some key challenges could be described, dividing them in two categories:

**Background visibility:** when a background pixel or region is seen in few frames during the training sequence there are some situations derived:

a) Low background visibility due to motion: Crowded environments involve many background occlusions, however if the background is occluded by moving objects the main representation of the background in that location is not going to be a foreground object.
b) Stationary objects: This issue involves either removed background and inserted background challenges. The target is to detect TB in spatial locations where there are static objects during the training sequence but the TB is visible sometimes. The stationary object could be in the same spatial location during most of the sequence.

**Photometric factors:**

a) Shadows and highlights: As mentioned above, shadows and highlights involve illumination changes in the scene. In the case of cast shadows, they do not represent the TB whereas there could be some shadows or highlights inherent to the TB.
b) Camouflages: This problem makes harder to distinguish between TB and camouflaged foreground.

There are different approaches in the state-of-the-art for BI based on pixel-level or block-level analysis, however we perform a classification of the most recent and relevant approaches attending to the time strategy followed to build the true background from the training sequence and the capability of operation provided:

**On-line:**

This category groups approaches which build the background relying on the temporal evolution of the frames [10][13][14][15] and are able to build a background image in each temporal instant. This category also includes background modeling approaches [3] operating in an on-line way. For this group conditions ‘b’ and ‘c’ are challenging.

**Batch:**

This category groups algorithms that analyze the whole training sequence requiring to be executed completely at each temporal instant to deliver a background image. The basic approach in this category is the median, however there are more recent and accurate algorithms as [7], [16] and proposed. Furthermore it exist an approach based on graph cuts and *image inpainting* [14]. In this category background subspace learning models are also included **Error! Reference source not found.**.

**Hybrid:**

This category groups approaches [8][11] which, as Batch, build the background analyzing the whole training sequence, however they are able to provide a background image at each temporal instant due to an on-line clustering instead of the batch clustering made by some algorithms included in the Batch category.

## 2.3. Background modelling

The background modelling stage has classically been the main criteria to organize BS approaches. In fact, for years, BS approaches were divided in parametric—evolutions of the well-known Mixture-of-Gaussians MoG [18]—and non-parametric—a successful alternative [19] [20] started by the top-referenced Kernel-Density-Estimation KDE [21], with cluster or codebook models [22][23] and Principal Component Analysis (PCA)-based subspace-learning [25] models being the rare alternative. However, recently we can identify new trends, including modelling based on self-organized neural networks [26], uncertainty-based fuzzy models [27] and evolutions of sub-space methods [17]. Basically, all these methods aim to provide robustness to *dynamic backgrounds*.

## 2.4. Background maintenance

Usually closely linked with the background modelling stage, maintenance mechanisms are fully included in the model definition. However, we can distinguish two main strategies for model maintenance: blind and selective. Blind maintenance equally considered every incoming sample—both background and foreground samples—for updating the model whereas selective maintenance uses different strategies for background—usually some sort of running average scheme—and foreground—most of the times no updating at all—samples.

Evolutions of these strategies include multiclass selective updating [28] and confidence-driven updating [29]. Multiclass updating enhances the segregation process (foreground-background) by introducing gradual classifications (e.g. foreground-shadows-background) and designing *ad-hoc* maintenance strategies for each class. On the other hand, confidence-driven updating combines the likelihood between new background and model samples with model history to adapt the learning rate.

Robust maintenance mechanisms used in flexible models aim to overcome camera—*noisy image*, *camera automatic adjustments*—and background—*illumination changes*—related challenges. Furthermore, is in this stage where the maintenance mechanism that defines whether inserted objects are incorporated to the model—*inserted background objects*, *sleeping foreground object*—and whether *ghosts* are updated—*removed background objects*, *beginning moving object*—.

## 2.5. Foreground detection

Foreground is detected as unobserved—or not modeled—samples. However, whereas this is sometimes understood as a simple classification task [3], in our opinion is determinant for obtaining accurate results whereas it is mainly driven by the features used for characterization.

Several features [30] have been proposed in the literature: color and luminance—spectral—which operate well in most scenarios but suffer from *camouflage*, *shadows*, *foreground aperture* and *illumination changes*, edge and texture—spatial—which can be used to remove *ghosts* (edges) or are assumed to operate better where color fail (texture) and disparity and depth—stereo—which are the best for handling *camouflage* but require the use of at least two cameras.

These features can be used to compute second-order features as motion—which inherits the advantages and disadvantages of the feature(s) used to obtain it—or combined by different schemes [31] [32].

# 3. Contributions

This chapter compiles the contributions developed in the VPU in the scope of this project. Although some of them—those in sections 3.2, 3.3 and **Error! Reference source not found.**—are integrated in a single system, we respect the per-stage organization.

## 3.1. Background initialization

We are currently developing a BI algorithm that will be published in the next months, however we include the current state of the approach as it is part of the project. The algorithm is currently providing similar results as the best state-of-the-art approaches.

Once we analyzed the state-of-the-art, we decided to build a block-based BI algorithm due to the higher amount of information that a block-level approach can provide, in comparison with a pixel-level, for the spatial continuity stage that was going to have our algorithm.

The proposed algorithm is divided in three stages: Temporal clustering, seed selection and spatial continuity analysis. Subsequently, stages are roughly explained:

**Temporal clustering:** First of all, each image under analysis is divided with a grid of N×N (N=16, as [11]) blocks (see Figure 1).

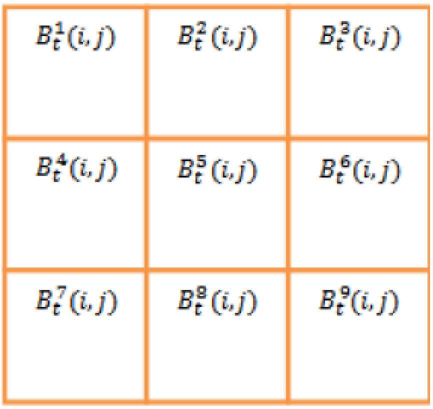


**Figure 1.** Block division performed.

The purpose of this stage is to reduce the amount of information (i.e. blocks in each spatial location) to analyze in the spatial continuity stage (see Figure 2). To this end, first a motion filtered is performed and subsequently an agglomerative clustering approach is applied together with a dimensionality reduction via Principal Component Analysis (PCA).

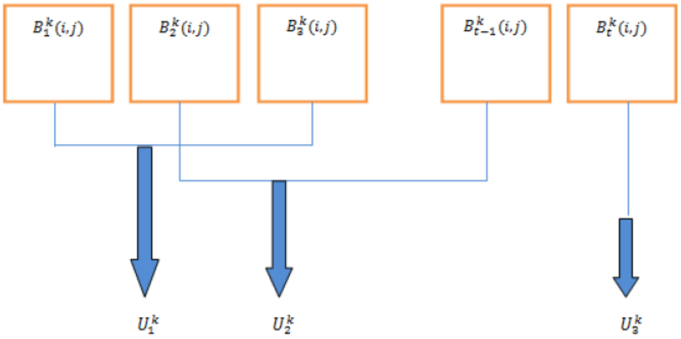


**Figure 2.** Clustering stage target. For example, in the location k similar blocks are grouped as a unique candidate ($U_i$ – Mean of $\mathrm{B}_j$ candidates), reducing the options available from *t* to 3.

Motion filter consists in removing those blocks of the training sequence belonging to moving objects due to their uselessness to reconstruct the TB. To extract motion information a classical pixel-level frame-difference (FD) is performed, with the slight variation from FD with distance 1 of computing differences among q-separated frames. The reason is to extract higher motion than 1-separated FD thus filtering more undesirable blocks. A block is considered to be moving if any of the pixels included suffer motion.

Once motion information is computed, a clustering technique is applied. The target is to group in each spatial location (block-level) the different candidates along time to form the TB, thus grouping similar candidates in one cluster and reducing the future computations (i.e. if the training sequence has *t* frames, there are *t* representations in each spatial location and with the clustering stage that number is decreased).

To increase the computational efficiency without losing accuracy the dimensionality of the data (blocks) is reduced applying PCA before the clustering is performed. PCA is applied to the $t$×(16*16*3) data matrix formed by *t* rows representing the block in each frame and 768 columns representing the rasterized block with each color channel (RGB) concatenated. The base to apply PCA is its property to take into account the dimensions which experiment a higher variance under the assumption that variance and information goes together, fact that in this problem is true because dimensions (pixels) that experience variation are the ones needed to perform the clustering.

Once the dimensions are reduced, the clustering is performed through an agglomerative hierarchical approach where a dendrogram is build based on the minimum Euclidean distance among clusters. Then the dendrogram is cut at different levels doing a scanning of different cluster hypotheses. These hypotheses are validated via clustering validation measures that check compactness and separation among clusters (Silhouette index and Davies-Bouldin index). The cut with best score is selected as the optimum clustering for that spatial location.

**Seed selection:** The purpose of this stage is to decide the starting point to begin the reconstruction, namely fix at least one block of the TB (see Figure 3). In this stage we develop a novel strategy to perform this task including motion information. Two admissible assumptions are taken into account:

a) The background is not completely occluded neither in the first frame nor in the last frame.

b) Static objects are not going to occupy the same spatial location at the beginning and at the end of the training sequence, excluding the case in which the object is stopped during the whole sequence, i.e. the case in which the object is part of the TB.

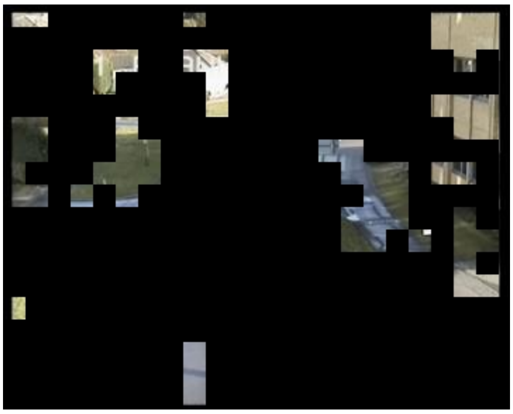


**Figure 2.** Example of seed selection. Blocks already selected are going to be used as point of departure for the final reconstruction of the spatial continuity stage.

The approach computes the FD between the first frame of the training sequence and the rest of the frames, thus obtaining motion information in comparison with the first instant. Once this information is computed, a confidence measure of each pixel belonging to the TB is computed averaging the motion score at each temporal instant. The same procedure is applied between the last frame and the rest. Subsequently both confidence measures are combined via mean operation. As we are operating at block-level we give a block confidence score taking into account the minimum pixel confidence score included in each block location. Then, the measure is normalized with the maximum block level confidence, thus obtaining the final measure to select the TB seeds. A seed is going to be fixed with the cluster grouping more blocks in those locations where the confidence is maxim. This technique for initializing the TB has been compared with approaches for the same task in the state-of-the-art reaching a higher TB initial reconstruction without errors. With this measure we obtain low confidence in those areas with high motion and stationary objects (no matter the instant where the stationary object is due to assumption *b)*).

**Spatial continuity analysis:** The purpose of this stage is to build the complete TB measuring continuities between fixed background blocks and their neighbor candidates. This stage is the only one that is not finished yet. Subsequently, we explained the current operation.
The reconstruction is performed with an iterative scheme, identifying in each step the best location to reconstruct (the one with more information). The reconstruction scheme consist on fixing in each iteration the 4 neighbors (up, left, down, right) of the selected seed, i.e. the one with more fixed blocks in the 3×3 neighborhood (external neighborhood blocks already fixed are taken into account in case of draw). To decide the best block in the neighbors each cluster candidate is check it measuring the edge continuity with all available neighbors via color differences. The best candidate is the one with lower difference. This reconstruction is made 8 times via multi-path reconstruction scheme. The different reconstructions are performed beginning in each 4 neighbors and moving along the two possible paths in the 3×3

neighborhood that begin and finish in the same point. Then, having 8 possible reconstructions the neighbors are selected among them as the ones with best spatial continuity. This procedure is repeated until the TB is obtained.

## 3.2. Background modelling

In the scope of the project, we have designed a Background Model at pixel level that follows a multi-layer philosophy under a non-parametric modelling strategy. To this aim, we have integrated several solutions from the state-of-the-art in a modular framework that ease module replacement while respecting the general idea of the design. The system is a real-time general-purpose solution. However, it was explicitly designed to operate in outdoors environments and then specific mechanisms to account for *dynamic backgrounds* are proposed.

In particular, the model is composed of $L+1$ layers, with $L$—the number of layers modelling the background—being a configurable parameter that *grows* with the background dynamics. Furthermore, an extra layer is included to account for foreground statistics. Each model stores through tensor-representation three statistics per pixel: an *RGB*-color representation of the pixel mode or appearance, a confidence value that measures the reliability of the stored representation, and a permissiveness scalar that adapts to the mode variability. From this point we explain the mechanisms associated to one of the layers, being the operation in the others equivalent. Section 3.4 describes the processes for layer initialization and inter-layer replacements.

**Appearance**: defined as subspace in the *RGB*-color Cartesian space $\mathbb{R}^3$ in order to account for local illumination changes as *shadows* and highlights. This subspace is designed to account for medium-intense local illumination changes under the assumption that an appearance affected by these changes would be similar to a scaled version of the non-affected appearance. This in general not true for strong illumination changes as the green (*G*) channel contains much more *illumination information* than the blue (*B*) channel. However, shadowing and reflects are not usually very intense in video sequences and there are several schemes in the literature that achieve excellent results by following this premise [33][34][35].

In particular, we designed a similar approach to the one proposed in [35], where a model's pixel $\vec{x}=\{x,y\}$ at instant $t$ is described through its *RGB*-mode $\vec{\mu}(\vec{x};t)$. The mode defines a color vector in $\mathbb{R}^3$ starting at the space origin $(0,0,0)$ and ending at the point coordinate: $\vec{\mu}(\vec{x};t)=\{\mu_R(\vec{x};t),\mu_G(\vec{x};t),\mu_B(\vec{x};t)\}$.

The magnitude of such vector is strongly related with the pixel's luminance and its expected that a shadowed version of the model-pixel would be represented by a similar directional vector—described by polar coordinates azimuth and elevation—but with smaller magnitude. Similarly, a highlighted version would be close in direction while presenting a higher magnitude. In order to account for the non-linear resolution of the *RGB*-color space—as it is designed in consonance with the human visual system, darker colors are assigned a smaller representation space—in [] it is proposed to place a cone aligned with the color vector with the vertex in the origin. Differently than their solution, we here defined the cone base radius $r(\vec{x};t)$ as a function of the model pixel permissiveness $K(\vec{x};t)$ as described in chapter 3.3. Finally the allowed strength of shadows ($\upsilon$) and highlights ($\nu$) defines a cone-truncated-shaped subspace. Only appearances of samples $\vec{I}(\vec{x};t)$ falling inside the subspace would be considered new samples of the model-pixel, i.e. a correspondence is found. Additionally, for all the new samples—with independence of its belonging to the subspace—the L2-norm to the stored model: $d(\vec{x};t)$ is computed. For further details see Figure 1 and read [35].

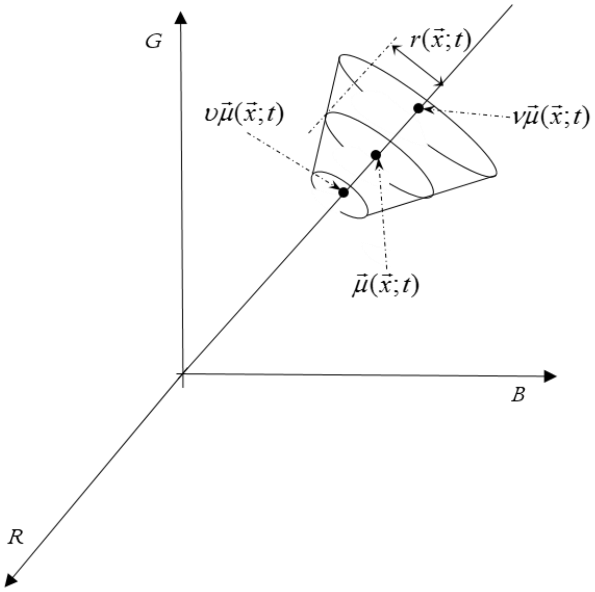


Figure 1. Cone-truncated subspace that defines a model-pixel appearance

**Confidence**: classical BS solutions do not include a statistic to measure the goodness of the model-pixel. A repetitive and stable mode can result in over-fitted distributions whereas a highly dynamic pixel can derive to a representation composed of several not-remarkable modes. A measure of the confidence of each model-pixel provide indications to drive the updating and foreground detection stages [28][29] whereas can be used as a tool to avoid over-fitting. In our solution the confidence of each model-pixel $C(\vec{x};t)$ is computed as a function of the number of pixel samples with *the same* appearance—falling in the cone-truncated subspace—, the distance of these samples to the stored mode and the permissiveness. The process is further explained in chapter 3.3.

**Permissiveness**: almost every BS approach defines strategies to control the permissiveness for obtaining correspondences, e.g. in MoG solutions the standard deviations of the Gaussians modelling each mode and in KDE approaches the width of the kernel used to estimate the distribution. As aforementioned, in our system the permissiveness parameter drives the comparison and confidence updating processes. Furthermore, we propose a mechanism to also update this parameter under an expectation premise (see chapter 3.3).

## 3.3. Background maintenance

In the maintenance mechanisms designed, the permissiveness plays a key-role. The whole updating process is illustrated in Figure 2. The permissiveness is first used to define the cone-truncated subspace radius—i—. New samples are compared—ii—, according to such radius, with model samples, then obtaining or not a correspondence but always a distance. Such distance is combined with the permissiveness to compute the confidence updating factor—iii—. The so-computed updating factor and the update confidence are used to decide whether or not replace—***R***—the model mode with the new sample—iv—. Finally the distance is used to update the permissiveness—v—.

The whole process is proposed as the solution to an expectation problem: according to the observed distances, how big need to be the permissiveness to increase the mode confidence an expected quantity $\Theta_e$?

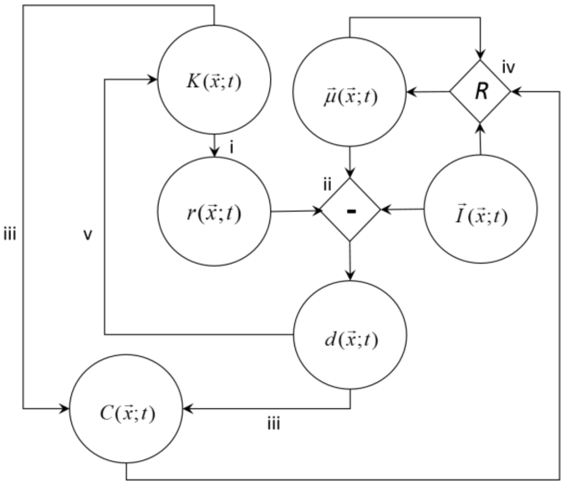


Figure 2. Sequence of updating process

**Confidence updating from permissiveness and distance**: We propose to use an exponential function on the distance to define the confidence learning rate. The cut-off of this function is function of the permissiveness. Explicitly, the updating factor $\Theta(\vec{x};t)$ is obtained through:

$$\Theta(\vec{x};t) = e^{\tilde{d}(\vec{x};t)^{K(\vec{x};t)}} - \delta\ , \qquad \tilde{d}(\vec{x};t) = 1 - \frac{d(\vec{x};t)}{d_{\max}}$$

, where $d_{\max}$ is the maximum reachable distance with the selected comparison scheme and $\delta = e - 1$ is a normalization constant. Figure 3 illustrated the exponential function for several permissiveness values. Note that the bigger is $K(\vec{x};t)$ the more restrictive is the system, i.e. the lower the distances are needed to positively update the confidence.

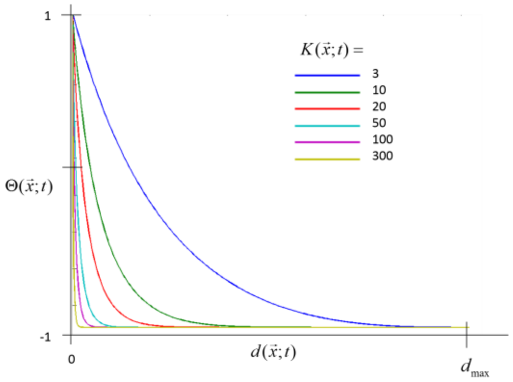


Figure 3. Sequence of updating process

**Radius from permissiveness:** The radius is obtained by solving the updating factor equation such that distances equal to the radius result in confidence increments $\Theta(\vec{x};t)=\Theta_e$:

$$r(\vec{x};t)=d_{\max}\left(\sqrt[K(\vec{x};t)]{\ln(\Theta_e)+\delta}-1\right)$$

**Updating the permissiveness:** We aim to smoothly update the permissiveness according to the distances among samples in the subspace. To this aim, we follow a classical running-average scheme:

$$K(\vec{x};t)=\alpha K(\vec{x};t)+(1-\alpha)K'(\vec{x};t), \qquad K'(\vec{x};t)=\frac{\ln(\ln(\Theta_e+\delta))}{\ln(\tilde{d}(\vec{x};t))}$$

, where $K'(\vec{x};t)$ is obtained by relating the obtained distance with the expected updating. Through this process the permissiveness controls the learning rate. Big distances would made the model more permissive—increasing $K(\vec{x};t)$ and then $r(\vec{x};t)$ whereas distances smaller than the radius—then resulting in bigger than $\Theta_e$ updating factors—would slowly narrow the subspace.

**Confidence-driven mode replacing:** Both evolved confidence and immediate updating factor are used as decision-makers to update the mode. Modes *under-construction* i.e. with associated low confidence are replaced with matching modes. Additionally, small—even positive—updating factors may also involve replacement. In particular, modes are replaced if:

$$\frac{C(\vec{x};t)}{\Theta(\vec{x};t)-\Lambda}\leq C_{\min}$$

, where $\Lambda \setminus \Lambda>\Theta_e$ threshold the updating factor and $C_{\min}$ the confidence.

**Initialization of new layers:** Samples resulting in updating factors smaller than $\Theta_e$ are used to initialize new layers in the model.

## 3.4. Foreground detection

In this section we first described and present results for a multiclass scheme that uses the model statistics described in section 3.2 and the updating process described in section 3.3. Then we further explore alternative features explicitly devoted to solve *camouflage* by presenting and on-going idea.

### 3.4.1. Class-driven foreground detection

We propose to classify a pixel according to its feasibility of being part of the temporal evolution of the model. First, we aim to distinguish between pixels belonging to moving objects—dynamic—and pixels which representation appears invariant—static—. Then, pixels in each class are further classify in pixels to which there are previous evidences in the background model—background—and pixels which previous evidences are in the foreground model or have not been yet observed—foreground—. Following these premises four classes are defined: static background, dynamic background, static foreground and dynamic foreground.

- Static / dynamic: we simply establish a threshold over each mode confidence, such threshold is a configurable parameter, but must be proportional to—and bigger than—

the minimum confidence that inhibits mode replacement: $C_{static} = \gamma C_{\min}, \gamma > 1$. Only new samples corresponding with a layer in the model—inside the cone-truncated subspace—are evaluated to be static. The rest of the samples are tagged as dynamic.

- Static background / static foreground: to perform this classification we made use of the foreground model. Only if the correspondence was obtained with the layer devoted to model the foreground the sample is classified as static foreground. In the eventuality of multi-layer correspondence, the layer with the lowest distance assigned prevails.
- Dynamic background / dynamic foreground: there is no temporal information—history— available for dynamic samples. Alternative processes are then required. We propose to use a regional comparison process which quantifies the difference between a dynamic region and its corresponding pixels in the best background model. Three processes are required; first, dynamic pixels are grouped in regions by a connected-component analysis. The best background model result from the selection of the modes along the model that maximize per-pixel confidence, this process result in an image where the most confidence modes are combined. Finally, for each connected component we study the cross-correlation between the normalized histograms of the *RGB*-modes of the pixels inside the component and that built by selecting in the best background model those modes that fall inside the connected component.
- Dynamic foreground—the whole connected component is classified—is detected at low values of the cross-correlation and used to initialize new areas in the foreground model. Dynamic background is used to initialize new areas in the background model.

**Updating inhibition:** classification process is performed immediately after the computation of the updating factor and is used to inhibit the updating of models with samples assigned to a different class, i.e. the statistics of a mode in the background model are not updated if a correspondence with a dynamic foreground pixel is obtained—such a situation can be possible due to the analysis of connected components—. This inhibition scheme reduces the likelihood of model's perturbation.

**Management of static objects:** the use of two models—one for the background and one for the foreground—along with the information provided by the confidence can be used to identify—and correct if desired— the challenges associated with removal or insertion of objects (see section 2.1).

**System performance:** Proposed system—described in sections 3.2, 3.3 and **Error! Reference source not found.**—is evaluated with all the videos in the *Change Detection* dataset—2012 version—[1]. Videos in the dataset are varied in size, nature, scenario and complexity and organized in six categories: *baseline*, *camera jitter*, *dynamic background*, *intermittent object motion*, *shadows* and *thermal*. A comparison in terms of per-category and overall precision, recall and f-score is carried out. Results are compiled in Figure 4, Figure 5 and Figure 6.

For the proposed system two configurations are designed. In the first configuration the cone-truncated subspace is disabled and replaced by a sphere of radius $r(\vec{x};t)$, and a maximum number of $L = 3$layers were available. In the second configuration the cone-truncated is activated and the number of layers was increased$L = 5$. Additionally, we included a neighborhood-wise module in the comparison, such that the distance is obtained—and then correspondence is searched—as the minimum when compared with a *3x3* neighborhood around the corresponding model-pixel.

[1] http://www.changedetection.net/

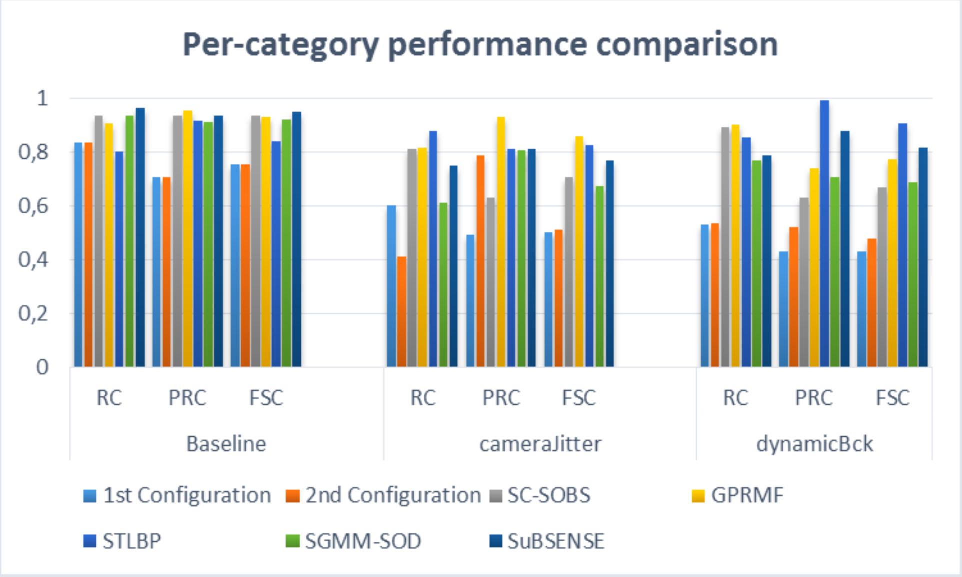


Figure 4. Recall (RC), precision (PRC) and f-score (FSC) results for three categories of the Change detection dataset. For comparison, results of leading approaches in the state-of-the-art [26][?][36][37][38]—from left to right—are also included.

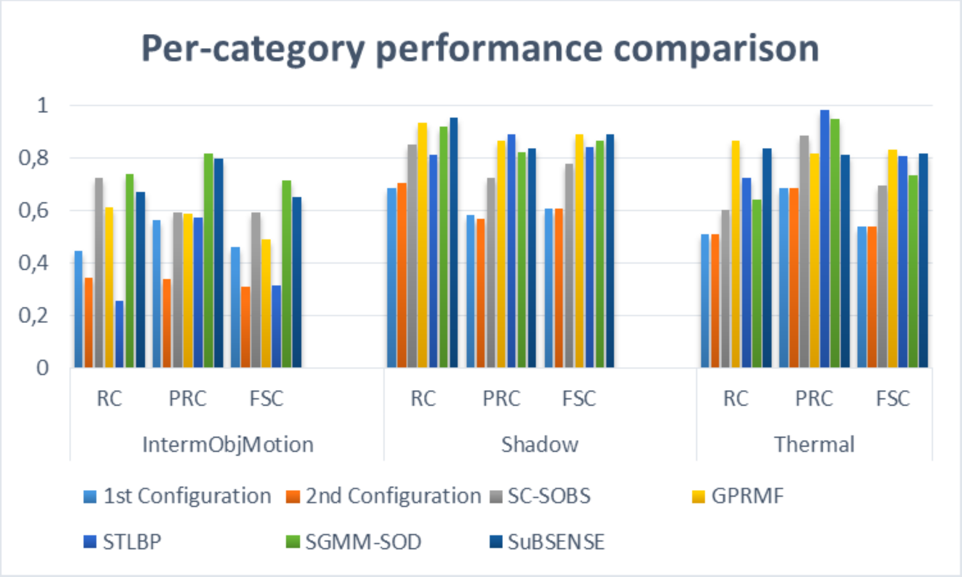


Figure 5. Recall (RC), precision (PRC) and f-score (FSC) results for the other three categories of the Change detection dataset. For comparison, results of leading approaches in the state-of-the-art [26][?][36][37][38]—from left to right—are also included.

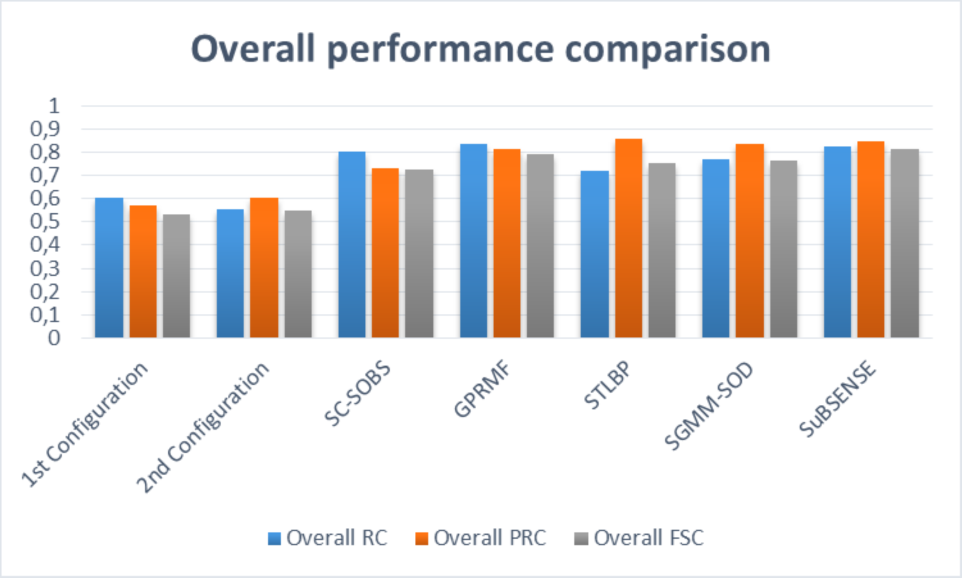


Figure 6. Average recall (RC), precision (PRC) and f-score (FSC) results for the Change detection dataset. For comparison, results of leading approaches in the state-of-the-art [26][?][36][37][38]—from left to right—are also included.

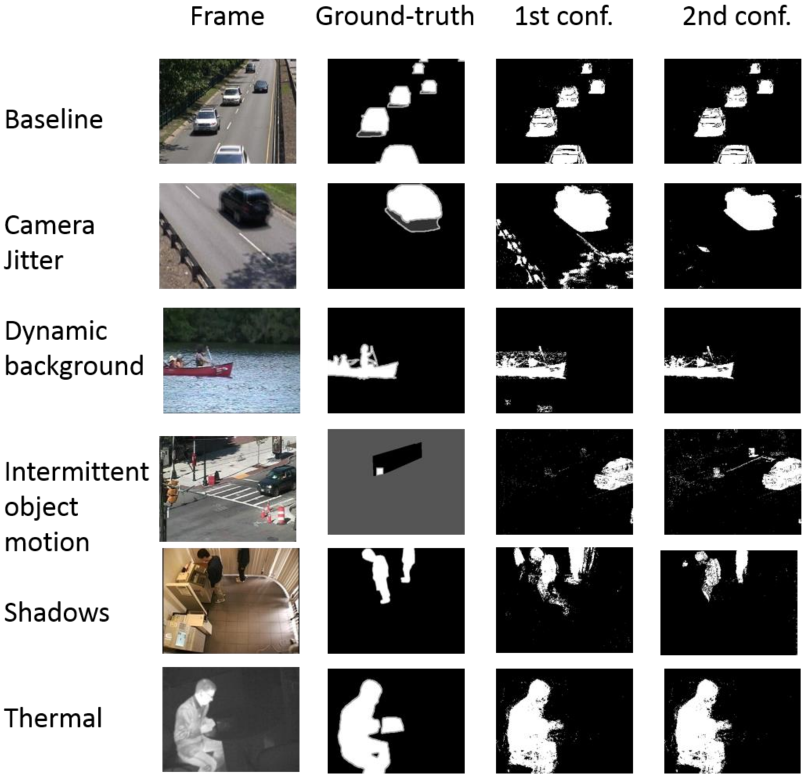


Figure 7. Examples of qualitative performance of the proposed system

**Results discussion:** Obtained results show that our system's performance is still far from that of the best approaches available in the state-of-the-art. Carefully analyzing the results, the main problem was identified as the management of *camouflage*—see qualitative results in Figure 7—.

Next section describes an on-going work that aims to face this challenge at the feature description and comparison stages.

### 3.4.2. A DCT-based robust to camouflage metric for foreground detection

**Problem description:** Foreground-background camouflage in BS techniques is a critical problem, which has been usually underestimated or miss-solved via post-processing techniques. Recent approaches tackle it by modelling pixels neighborhood instead of isolated pixels, which, more generally, aims to enhance foreground-background discrimination. These approaches, while more robust to foreground-background similarity, may affect the accuracy of the background model.

**Proposed solution**: We design a new feature to model pixel local variability, which enhances background-foreground discrimination. It is derived from the Discrete Cosine Transform (DCT) and was explicitly motivated to handle *camouflage* and local *illumination changes*.

As generally known, the DCT coefficients, $c(u,v)$ of a $WxW$ -pixels square block *centered* at a pixel $\vec{x}_0 = (x_0, y_0)$ of a scalar image $I(x,y)$, are computed as:

$$c(u,v) = \alpha(u)\alpha(v) \sum_{x=x_0-\frac{W}{2}}^{x_0+\frac{W}{2}-1} \sum_{y=y_0-\frac{W}{2}}^{y_0+\frac{W}{2}-1} I(x,y) cos\left[\frac{\pi(2x+1)u}{2W}\right] cos\left[\frac{\pi(2y+1)v}{2W}\right]$$

, where $\alpha(0) = \sqrt{\frac{1}{W}}$, $\alpha(u,v \neq 0) = \sqrt{\frac{2}{W}}$ and $0 \leq u,v < W$ .

The DCT has several properties that make it a suitable tool for estimating the color distribution of a pixel block neighborhood.

- Each DCT coefficient conveys a measure of the similarity between the $I(x,y)$ values distribution inside the block centered at $\vec{x}_0 = (x_0, y_0)$ and a directional response determined by the 2D basis functions or images. The whole block distribution can be seen as a weighted combination of these directional responses.
- It leads to a set of low-correlated coefficients which are suitable to be modeled independently.
- Illumination changes that have effect on the whole block and are not so strong to occlude variability inside the block, mainly affect the DC coefficient, $c(0,0)$ . Then, a technique not considering this coefficient would be less sensitive to these changes.
- The transform is separable, symmetric and orthogonal. It is separable as each coefficient $c(u,v)$ can be computed in two steps by successive one dimensional operations on rows and columns of a block. Symmetry means that row and column operations are functionally

identical. Finally, as the inverse DCT transformation matrix is equal to its transpose, the DCT is othogonal. These properties allow a fast and efficient computation of the DCT.

From now on we will just focus on the AC coefficients, which are those describing local variability. Despite all the indicated properties, up to our knowledge, there are no BS works describing local variability with a block based DCT. In our opinion, this might be due to the problematic involved in computing differences between the DCTs of two image blocks. If either all the AC coefficients or a fixed subset are considered as a single feature vector, intensity to intensity comparison between vectors would be adequate, as each vector component represents the same coefficient. However, firstly, considering all the coefficients is both inefficient and noisy as most of the information is compacted in a few coefficients. This would result in a system very sensitive to foreground presence, but also to slight background changes; that is, a system with high recall in foreground detection, but poor results in foreground precision. Secondly, selecting only a fixed subset of AC coefficients would fail for blocks mainly characterized by the non-selected ones.

In practical situations, most of the information in an image block is represented by a few AC coefficients. In order to evaluate the dissimilarity between two blocks, it would be natural to compute differences just among the representative ACs, which are not generally the same for all blocks. Each AC coefficient, $c(u,v)$ represents a response to a pattern or basis image function, $B_{u,v}$—depicted in Figure 8 a) for $W=8$—. Hence, independently of the AC coefficient value, dissimilarity evaluation first requires a measure of the similarity between every pair of patterns.

We here propose a simple estimation of such subjective similarity, attending to spatial variability rhythm and direction, and weighting them in a well-balanced fashion. Considering the classical 2D representation of the DCT basis functions—see Figure 8 a) —, we obtain a measure of the distance between two of these functions following:

$$M[B_{u_1,v_1}, B_{u_2,v_2}] = k_1\left[| u_1 - u_2 | \vee | v_1 - v_2 |\right] + k_2\left[| \, atan(\frac{u_1}{v_1}) - atan(\frac{u_2}{v_2}) |\right]$$

, where $a \vee b$ means the maximum of $a$ and $b$, while $atan(a)$ stands for the arc tangent of $a$. The non-negative weight factors $k_1$ and $k_2$ are set to equally weight both terms of the equation, which intends to formalize that patterns with maximum difference in variability orientation—i.e., orthogonal orientation—are considered as different as those with equal orientation but maximum difference in variability rhythm. Observing that the first term takes values in the range $[0:W-1]$, and the second one varies in $[0:\pi/2]$, we can set $k_1=1$ and $k_2=(W-1)\pi/2$. Other combinations of $k_1$ and $k_2$ keeping balance between the two parts of the equation would be also valid. The proposed measure fulfills the properties of non-negativity, positive definition, symmetry and sub-additivity—proof of these properties is leaving out of this document for legibility—.

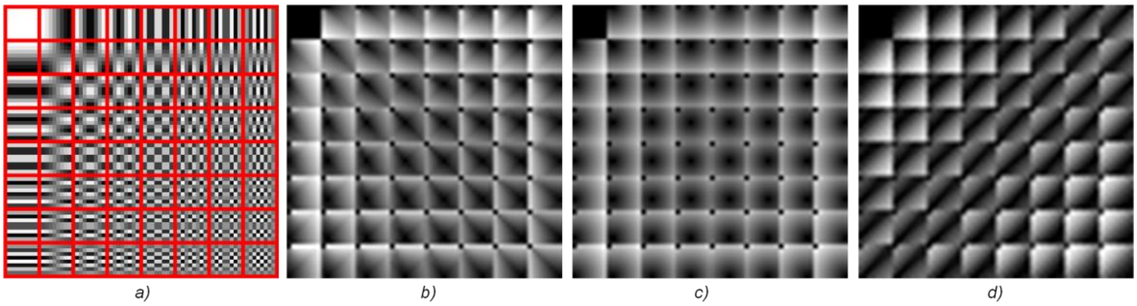


Figure 8. a) Representation of the DCT basis functions for $W=8$, b) Metric evaluation between every single AC basis function and all the other ones, c) 2D Euclidean distance between every single AC basis function and all the other ones, d) 1D Euclidean distance between every single zigzag ordered AC basis function and all the other ones.

The proposed metric can be visually inspected in Figure 8 b). The metric evaluated for every pair of AC basis functions is plotted block-wise, that is, it is organized in a $WxW$ -blocks gray-level image. Each block's pixel presents the distance (the higher the brighter) between the co-located basis function displayed in Figure 8 a) and all the other $WxW-1$ functions —including self-similarity and excluding similarity with the DC basis function, which is set to zero or black—. The block corresponding to the DC coefficient is set to zero as it is unused. For visualization purposes we have set $W=8$ and scaled the resulting images. Observe, for instance, that $B_{0,1}$ results as different from $B_{1,0}$—just due to variability direction—as from $B_{0,7}$—just due to variability rhythm—.

An intuitive but in this case senseless alternative is to use the Euclidean distance between the basis-functions positions, i.e. $(u,v)$, in a 2D vector space; this is illustrated in Figure 8 c). Observe that in this case $B_{0,1}$ results relatively similar to $B_{1,0}$, while representing orthogonal patterns. Finally, we also include in Figure 8 d) the 1D Euclidean distance between every single AC basis function and all the other ones, ordered following the classical zigzag technique; again, orthogonal patterns are very close in the distance space—observe the similarity between $B_{0,1}$ and $B_{1,0}$ which are separated by the minimum distance step—.

To model each pixel, we simply store the $N$ highest energy—in absolute value—AC coefficients per pixel, weight their relevance according to their relative contribution—in energy percentage—and then compare new samples through the described metric. We named this description WRAC (i.e., based on intensity Weighted Ranked AC patterns):

**Separating foreground and background:** In order to evaluate the discriminative power of the proposed characterization and metric, four other features have been selected for comparison. Two of them aim to compare the proposed metric against two alternative ways of considering DCT coefficients. One, which we will refer as AC1, replicates the proposed characterization but using the 2D Euclidean distance to measure the similarity between two DCT basis-functions:

$$M'[B_{u_1,v_1},B_{u_2,v_2}]=\sqrt{(u_1-u_2)^2+(v_1-v_2)^2}$$

The other, which we will refer as AC2, replicates AC1 but using the first $N$ coefficients of the DCT (following the classical zigzag order), instead of the $N$ higher energy ones. The third selected feature is the original uniform LBP [39], designed, as the proposed feature, to measure local variability. The circular radio around the pixel region that defines how many neighbors are used to build the LBP descriptor has been set to $W/2$ in order to perform a faithful comparison in terms of quantity of neighbors accounted. Finally, the fourth feature is the pixel luminance Y, which has been, for years, one the most popular way of considering the pixel value.

For the experiment we have selected four videos from the data set described in [40], as ground truth segmentation is available for every frame and contain several of the complex situations that affect backgrounds in real scenarios, which supports the robustness of the obtained results. These raw videos, described in Figure 9, are 600 to 1200 frames long each, with 720x576 resolution. Apart from an example frame (a), we also include an average mask of foreground occurrence in the video (b), the average squared luminance difference between foreground and background for each pixel (uniform red areas correspond to frame areas not affected by the foreground) (c) and a frame showing the background pixels prone-to-camouflage (d). These refer to background pixels whose difference to the foreground is zero in at least one video frame, although larger differences might also cause camouflage. For the third experiment we use some more popular videos, but with ground truth segmentation just on some selected frames.

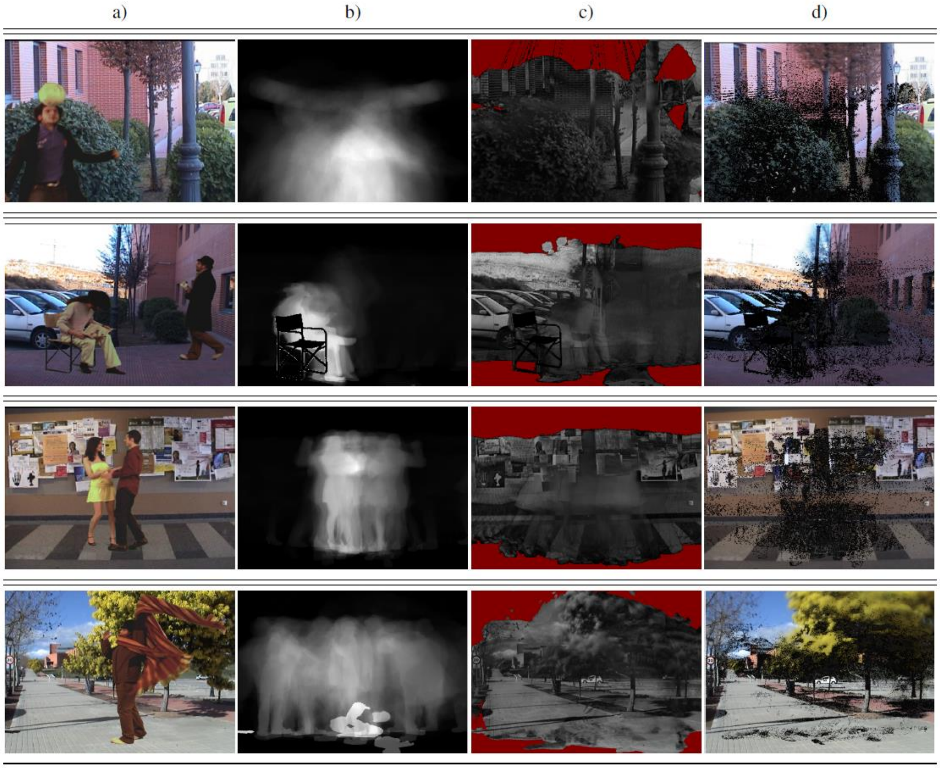
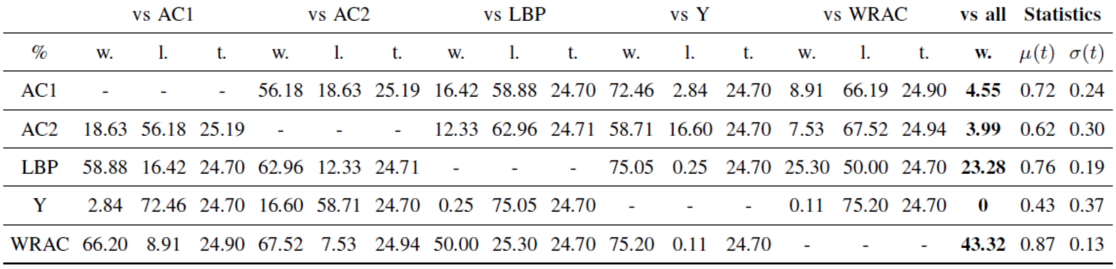


Figure 9. Videos extracted from [40]. a) Example frame, b) Foreground evolution, c) Average difference between foreground and background (red areas are never foreground), d) Prone to camouflage pixels (in black).

We use the ground-truth segmented videos to obtain, for background pixel instances and for foreground ones, for every pixel position and for all the data set frames, the histograms or distributions of the values of the five features. Then the overlap for each feature between both distributions is evaluated using the well-known Bhattacharyya distance.

The comparison is performed in terms of wins (w), losses (l) and ties (t): given two Bhattacharyya distances, $B_1$ and $B_2$, resulting from computing the overlap between foreground and background distributions for features $M_1$ and $M_2$ respectively, $M_1$ beats (wins) $M_2$ if $B_1$ is higher than $B_2$, $M_1$ ties with $M_2$ if $B_1$ equals $B_2$, and losses if $B_1$ is lower than $B_2$. A Kolmogorov-Smirnov test with a 5% significance level is previously performed over each pair of background-foreground distributions in order to avoid comparison of identical distributions, a situation which finally did not occur in the selected data set. Comparisons between every pair of considered features, as well as overall winning and mean and standard deviations of pixel-average Bhattacharyya distances are included in Table 1.



| | vs AC1 | | | vs AC2 | | | vs LBP | | | vs Y | | | vs WRAC | | | **vs all** | **Statistics** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | w. | l. | t. | w. | l. | t. | w. | l. | t. | w. | l. | t. | w. | l. | t. | **w.** | $\mu(t)$ | $\sigma(t)$ |
| AC1 | - | - | - | 56.18 | 18.63 | 25.19 | 16.42 | 58.88 | 24.70 | 72.46 | 2.84 | 24.70 | 8.91 | 66.19 | 24.90 | **4.55** | 0.72 | 0.24 |
| AC2 | 18.63 | 56.18 | 25.19 | - | - | - | 12.33 | 62.96 | 24.71 | 58.71 | 16.60 | 24.70 | 7.53 | 67.52 | 24.94 | **3.99** | 0.62 | 0.30 |
| LBP | 58.88 | 16.42 | 24.70 | 62.96 | 12.33 | 24.71 | - | - | - | 75.05 | 0.25 | 24.70 | 25.30 | 50.00 | 24.70 | **23.28** | 0.76 | 0.19 |
| Y | 2.84 | 72.46 | 24.70 | 16.60 | 58.71 | 24.70 | 0.25 | 75.05 | 24.70 | - | - | - | 0.11 | 75.20 | 24.70 | **0** | 0.43 | 0.37 |
| WRAC | 66.20 | 8.91 | 24.90 | 67.52 | 7.53 | 24.94 | 50.00 | 25.30 | 24.70 | 75.20 | 0.11 | 24.70 | - | - | - | **43.32** | 0.87 | 0.13 |

Table 1. Overall results for Foreground-Background separability of raw data for proposed feature (WRAC), *Ranked Euclidean (AC1)*, *ZigZag Euclidean (AC2)*, LBP and Luminance (*Y*) in terms of Bhattacharyya distance.

# 4. Conclusions and Future Work

Background subtraction is a complex task, in this document we have analyzed the challenges and propose several solutions to face them. The main objective is still shared with the state-of-the-art: design an approach able to face all the challenges at the same time. In this vein, there is a trade-off problem between efficiency and challenge covering, i.e. there are efficient solutions that succeed in the management of a subset of the challenges, and almost every challenge has been target of excellent research. However, the integration of all those solutions at the same time may result in time- consuming analysis which inhibit the use of the system at real-time-demanding scenarios.

We have propose a flexible BS approach and realize that whereas its operation is functional for most of the applications, its statistics suffer from camouflage problems. With this in mind, we have inspected new information representation schemes, achieving promising—but not concluding—results.

Future work is mainly focused in the exhaustive evaluation of ongoing approaches—initialization and camouflage—and in its integration in the proposed analysis framework.

# References

[1] Jain, M.; Jegou, H.; Bouthemy, P., "Better Exploiting Motion for Better Action Recognition", *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2555-2562, June 2013.

[2] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D., "Cascade object detection with deformable part models", *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2241-2248, June 2010.

[3] Thierry Bouwmans, "Traditional and recent approaches in background modelling for foreground detection: An overview", *Computer Science Review*, vol. 11–12, pp. 31-66, May 2014.

[4] Toyama, K.; Krumm, J.; Brumitt, B.; Meyers, B., "Wallflower: principles and practice of background maintenance", *Computer Vision and Pattern Recognition (CVPR), 1999 IEEE Conference on*, vol. 1, pp. 255-261, September 1999.

[5] Marco Cristani, Michela Farenzena, Domenico Bloisi, and Vittorio Murino, "Background subtraction for automated multisensor surveillance: a comprehensive review", *EURASIP J. Adv. Signal Process* 2010, no. 43, pp. 1-24, 2010.

[6] Elhabian, S. Y., El-Sayed, K. M., & Ahmed, S. H., "Moving object detection in spatial domain using background removal techniques-state-of-art", *Recent patents on computer science*, vol. 1, pp. 32-54, 2008.

[7] Colombari, A.; Fusiello, A., "Patch-Based Background Initialization in Heavily Cluttered Video", *Image Processing, IEEE Transactions on*, vol. 19, no.4, pp. 926-933, April 2010.

[8] Baltieri, D.; Vezzani, R.; Cucchiara, R., "Fast Background Initialization with Recursive Hadamard Transform", *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 165-171, August-September 2010.

[9] D. Park, H. Byun, "A unified approach to background adaptation and initialization in public scenes", *Pattern Recognition*, vol. 46, no. 7, pp. 1985-1997, July 2013.

[10] Han-Hui Hsiao and Jin-Jang Leou, "Background initialization and foreground segmentation for bootstrapping video sequences", *EURASIP Journal on Image and Video Processing*, vol. 12, 19 pages, February 2013.

[11] Vikas Reddy, Conrad Sanderson, and Brian C. Lovell. 2011. "A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts", *J. Image Video Process. 2011*, no. 1, pp. 1-14, January 2011.

[12] Xida Chen, Yufeng Shen, and Yee Hong Yang. "Background estimation using graph cuts and inpainting", *In Proceedings of Graphics Interface 2010 (GI '10)*, pp. 97-103, June 2010.

[13] Rui Zhang; Weiguo Gong; Yaworski, A.; Greenspan, M., "Nonparametric on-line background generation for surveillance video", *2012 Pattern Recognition (ICPR) International Conference on*, pp. 1177-1180, November 2012.

[14] Colque, R.V.H.M.; Camara-Chavez, G., "Progressive Background Image Generation of Surveillance Traffic Videos Based on a Temporal Histogram Ruled by a Reward/Penalty Function", *2011 24th Graphics, Patterns and Images (SIBGRAPI), Conference on*, pp. 297-304, August 2011.

[15] Crivelli Tomás, Bouthemy Patrick, Cernuschi-Frías Bruno, Yao Jian-feng, "Simultaneous Motion Detection and Background Reconstruction with a Conditional Mixed-State Markov Random Field", *International Journal of Computer Vision*, vol. 94, no. 3, pp. 295-316, 2011.

[16] Wang, H.; Suter, D., "A Novel Robust Statistical Method for Background Initialization and Visual Surveillance", *Proceedings of the 7th Asian Conference on Computer Vision (ACCV)*, vol. 3851, part 1, pp. 328-337, January 2006.

[17] Thierry Bouwmans, El Hadi Zahzah, "Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance", *Computer Vision and Image Understanding*, vol. 122, pp. 22-34, May 2014.

[18] Stauffer, Chris; Grimson, W. E L, "Adaptive background mixture models for real-time tracking," *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, June 1999.

[19] Barnich, O.; Van Droogenbroeck, M., "ViBe: A Universal Background Subtraction Algorithm for Video Sequences", *Image Processing, IEEE Transactions on* , vol. 20, no. 6, pp. 1709-1724, June 2011.

[20] Hofmann, M.; Tiefenbacher, P.; Rigoll, G., "Background segmentation with feedback: The Pixel-Based Adaptive Segmenter," *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 38-43, June 2012.

[21] Elgammal, A; Duraiswami, R.; Harwood, D.; Davis, L.S., "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151-1163, July 2002.

[22] Butler, D. E., Bove, V. M., & Sridharan, S., "Real-time adaptive foreground/background segmentation", *EURASIP Journal on Advances in Signal Processing*, vol. 14, pp. 2292-2304, 2005.

[23] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, Larry Davis, "Real-time foreground–background segmentation using codebook model", *Real-Time Imaging*, vol. 11, no. 3, pp. 172-185, June 2005.

[24] N. Oliver, B. Rosario, A. Pentland, "A Bayesian computer vision system for modeling human interactions", *International Conference on Vision Systems (ICVS)*, January 1999.

[25] Bouwmans, T., "Subspace learning for background modeling: A survey", *Recent Patents on Computer Science*, vol. 2, no. 3, pp. 223-234, 2009.

[26] L. Maddalena, A. Petrosino, "The SOBS algorithm: What are the limits?", *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, June 2012.

[27] Wonjun Kim; Changick Kim, "Background Subtraction for Dynamic Texture Scenes Using Fuzzy Color Histograms", *Signal Processing Letters, IEEE*, vol. 19, no. 3, pp. 127-130, March 2012.

[28] Colmenarejo, A.; Escudero-Viñolo, M.; Bescós, J.: "Class-driven Bayesian background modelling for video object segmentation", *Electronics Letters*, vol. 47, no. 18, pp. 1023-1024.

[29] Porikli, F., Tuzel, O.: "Bayesian background modeling for foreground detection", *Proc. ACM Visual Surveillance and Sensor Networks*, vol. 1, pp. 55–58, 2005.

[30] Liyuan Li; Weimin Huang; Gu, IY.-H.; Qi Tian, "Statistical modeling of complex backgrounds for foreground object detection," *Image Processing, IEEE Transactions on* , vol. 13, no. 11, pp. 1459-1472, November 2004.

[31] H. Zhang, D. Xu, "Fusing color and texture features for background model", *Third International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 4223, pp. 887–893, September 2006.

[32] H. Bhaskar, L. Mihaylova, A. Achim, "Video foreground detection based on symmetric alpha-stable mixture models", *Circuits Syst. Video Technol. IEEE Transactions on*, vol. 20, no. 8, pp. 1133-1138, 2010.

[33] Horprasert, T.,Harwood, D., andDavis, L. S., "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection", *In Proceedings of the IEEE Conference ICCV*, vol. 99, pp. 1–19, September 1999.

[34] Jian Yao; Odobez, J., "Multi-Layer Background Subtraction Based on Color and Texture", *Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on*, pp. 1-8, June 2007.

[35] Ruben Heras Evangelio, Michael Pätzold, Ivo Keller, Thomas Sikora, "Adaptively Splitted GMM with Feedback Improvement for the Task of Background Subtraction", Information Forensics and Security IEEE Transactions on, vol. 9, no. 5, pp. 863-874, 2014.

[36] Zhang, S., Yao, H., & Liu, S., "Dynamic background modelling and subtraction using spatio-temporal local binary patterns", *Image Processing 15th IEEE International Conference on*, pp. 1556-1559, October 2008.

[37] Evangelio, R. H., & Sikora, T, "Complementary background models for the detection of static and moving objects in crowded environments", *Advanced Video and Signal Based Surveillance (AVSS), 2011 8th IEEE International Conference on* (pp. 71-76). IEEE, October 2011.

[38] St-Charles, P. L., Bilodeau, G. A., & Bergevin, R., "Flexible Background Subtraction With Self-Balanced Local Sensitivity".

[39] HEIKKILÄ, Marko; PIETIKÄINEN, Matti., "A Texture-Based Method for Modeling the Background and Detecting Moving Objects", Pattern Analysis and Machine Intelligence IEEE Transactions on, vol. 28, no. 4, pp. 657-662, 2006.

[40] Tiburzi, F.; Escudero, M.; Bescos, J.; Martinez, J.M., "A ground truth for motion-based video-object segmentation", *Image Processing 15th IEEE International Conference on*, pp. 17-20, October 2008.