IEEE Transactions on Circuits and Systems for Video Technology 29.6 (2019): 1645 - 1658

DOI: https://doi.org/10.1109/TCSVT.2018.2851440

Hierarchical improvement of foreground segmentation masks in background subtraction

Diego Ortego, Juan C. SanMiguel, José M. Martínez

***Abstract*—A plethora of algorithms have been defined for foreground segmentation, a fundamental stage for many computer vision applications. In this work, we propose a post-processing framework to improve foreground segmentation performance of background subtraction algorithms. We define a hierarchical framework for extending segmented foreground pixels to undetected foreground object areas and for removing erroneously segmented foreground. Firstly, we create a motion-aware hierarchical image segmentation of each frame that prevents merging foreground and background image regions. Then, we estimate the quality of the foreground mask through the fitness of the binary regions in the mask and the hierarchy of segmented regions. Finally, the improved foreground mask is obtained as an optimal labeling by jointly exploiting foreground quality and spatial color relations in a pixel-wise fully-connected Conditional Random Field. Experiments are conducted over four large and heterogeneous datasets with varied challenges (CDNET2014, LASIESTA, SABS and BMC) demonstrating the capability of the proposed framework to improve background subtraction results.**

***Index Terms*—Foreground segmentation improvement, background subtraction, foreground quality, post-processing**

I. Introduction

BENCHMARKING computer vision algorithms has recently garnered remarkable attention as a methodological performance assessment [1][2][3][4][5] driving the development of better algorithms. Alternatively, one may focus on improving the results of algorithms by post-processing techniques. This scheme may be of interest when the details of algorithms are not available and, therefore, making further changes or adjusting parameters is not possible.

In this context, foreground segmentation is a popular low-level task in computer vision to detect the objects of interest or foreground in images or videos [3][5][6][7][8] where such "interest" depends on the application domain. For example, foreground in images can be defined as salient or co-salient objects [3][9][10] or as generic objects [11][12]. In videos, foreground may correspond to all moving objects [6] or specific objects relying on saliency [13] or co-saliency [14], spatio-temporal patterns [15] or weak labels [16]. Moreover, unconstrained video object segmentation addresses challenges related to camera motion, shape deformations of objects or motion blur [17]. Existing approaches are unsupervised (e.g. detect spatio-temporal relevant objects [18][19]), semi-supervised (e.g. propagate initially segmented objects [20]) or supervised (e.g. frame-by-frame human intervention [21]). In this paper, we focus on video sequences with a relative control of camera motion, where video object segmentation is tackled through background subtraction (BS) [6][22] which compares each frame with a background model of the sequence.

Boosting BS performance has been mainly addressed by making use of three strategies. Firstly, selecting appropriate background models is akin to the ability of simultaneously dealing with several challenges [6] while accurately adapting the background model to sequence variations. For example, Gaussian and support vector models [23][24] deal effectively with dynamic background; subspace learning models [25][26] handle better illumination changes; neural networks [27][28] offer a good computation-accuracy trade-off; and RPCA (Robust Principal Component Analysis) and sparse models [29][30][31][32] provide suitable frameworks to integrate constraints for foreground segmentation under different challenges. Secondly, properly choosing BS features [33][34][35] is key as each feature type (e.g. color, gradient, texture, motion) exhibits robustness against different BS challenges, thus combining them may overcome single-feature shortcomings. Moreover, deep learning models [36][37][38][39] have recently emerged as promising frameworks to unify modeling and feature selection. However, current models [36][37][38][39][40] are limited to employ train and test data from the same video sequence. Thirdly, post-processing techniques may improve foreground segmentation masks by either removing false positives or recovering false negatives [41]. For instance, there are techniques independent of the BS algorithm such as morphological operations [42][43] to fill holes or remove small regions; and inspection foreground mask properties [44][45] to filter false positives and expand to undetected areas. Moreover, specific post-processing may tackle errors due to illumination changes [46][47], shadows [48][49] or dynamic backgrounds [42][50]; but the designed features depend on the employed background model, thus limiting their applicability.

For BS post-processing, the use of generic properties from foreground masks is desired to provide independence of specific phenomena (e.g. illumination or shadows) and, unlike morphological operations, to exploit complementary features to the ones extracted from the mask only. A recent analysis of these properties to estimate performance without ground-truth data (i.e. quality) [51] identified the best property as

Diego Ortego, Juan C. SanMiguel and José M. Martínez are with the Video Processing and Understanding Lab, Universidad Autónoma the Madrid, Madrid, Spain, e-mail: diego.ortego@uam.es, juancarlos.sanmiguel@uam.es, josem.martinez@uam.es.

the fitness between connected components of the foreground mask (i.e. blobs) and the regions of the segmented image (fitness-to-regions). Therefore, in this paper we propose to improve foreground segmentation masks in BS through the fitness to several segmented image regions partitions, which enables extending foreground masks to undetected areas while removing poorly fitted and isolated foreground regions.

The contribution of this paper is five-fold. Firstly, we introduce motion constraints to build an image segmentation hierarchy without merging moving foreground and background regions. Secondly, unlike related state-of-the-art [44][45], we apply the fitness-to-regions property to estimate the quality of the foreground mask using each image in the segmentation hierarchy. We obtain a hierarchy of foreground quality images leading to better improvement scores as compared to [44]. Thirdly, a motion-based combination of the foreground quality images hierarchy is proposed to prevent foreground-background merging in absence of motion, while promoting the extension of foreground regions in presence of motion. Fourthly, we improve foreground mask by fusing the foreground quality images into a unique foreground quality that is later converted into a foreground probability map by applying a pixel-wise fully-connected Conditional Random Field (CRF). Fifthly, we demonstrate the utility of the proposed approach to improve BS results of both top and low performing algorithms as presented in the experimental comparisons conducted using fourteen algorithms over four heterogeneous datasets with varied challenges (CDNET2014 [1], LASIESTA [52], SABS [53] and BMC [54]). Moreover, we also show the potential application of foreground quality images for algorithm combination.

The reminder of this paper is organized as follows: Section II overviews existing post-processing techniques for BS. Section III-B details the proposed framework for BS post-processing. Subsequently, Section IV presents the experimental methodology and the experimental results. Finally, Section V summarizes the main conclusions.

II. Related Work

Post-processing techniques for BS can be classified into model-dependent and model-independent. The former employs the background model, such as shadows detectors to compare image and background features in foreground areas [48], whereas the latter only uses image and foreground properties [44][45], thus being independent of a particular algorithm.

Model-dependent techniques target challenging situations that produce erroneous foreground such as illumination changes, shadows or dynamic background. Removing erroneously detected foreground due to illumination changes has been addressed through color relations between the image and the background model in foreground areas [46][47]. Furthermore, chromatic, physical, geometric or texture relations between images and its related background model can be exploited to detect cast shadows in foreground masks [48][49][55]. Additionally, detecting dynamic background motion [50] has not directly been tackled to post-process the result but to guide parameter tuning [42][56]. However, the joint analysis of blinking pixels and background to image differences performed in [42] could be directly applied to remove false positives rather than influence the background modeling. Similarly, one can find that contour based techniques for abandoned object detection [57][58], based on both image and background information in foreground areas, can be applied to remove foreground errors associated to ghosts.

Model-independent techniques are based on the analysis of foreground mask properties to improve results. A common strategy is to post-process foreground masks through morphological operations [42][43]. This strategy only relies on the foreground mask, thus obviating useful information that can be extracted from a joint analysis of the foreground and the color image. In this sense, there are techniques that analyze generic foreground mask properties [44][45] to filter erroneous foreground or to expand it to undetected foreground areas. In [45], region or blob mask properties associated to the internal uniformity, contrast in contours, shape complexity and fitness-to-regions are used to remove false positives blobs. Furthermore, [44] employs fitness-to-regions embedded into a Markov Random Field framework where high (low) fitness is associated to good (poor) foreground probability. In [59], the coherence of optical flow directions in each individual frame and frame-by-frame coherence of optical flow are used to remove erroneous blobs, split blobs that contain different objects and merge blobs belonging to the same object, thus improving foreground segmentation performance in background subtraction. Moreover, in [60] image boundaries are used to remove erroneously detected blobs caused by the effect of illumination. Also, ghosts can be post-processed using optical flow [41], as foreground objects often moves. However, absence of motion is not only characteristic in ghosts, but also in static foreground objects.

As a conclusion, *Model-independent* techniques stand out as very interesting alternatives due to their independence of BS algorithms. The fitness-to-regions property has demonstrated a great potential to both estimate foreground quality [51] and improve results [44]. However, the use of over-segmented images (i.e. superpixels) in [44] highly limits the improvement capabilities, as superpixels normally do not extend over complete objects. In fact, such mapping between superpixels and objects remains an open issue in the object proposal literature [61][62][63], where superpixel merging to cover large or complete object regions is inspected.

III. Foreground mask improvement

A. Overview

We propose a framework to improve foreground masks $\mathcal{M}_t$ obtained by BS algorithms from an image $\mathcal{I}_t$ in the temporal instant t (see Figure 1). Firstly, we compute a motion-aware segmentation hierarchy $\mathbb{H}_t = \left\{\mathcal{R}_t^l\right\}_{l=1}^{L}$, where $\mathcal{R}_t^l = \left\{R_{t,i}^l\right\}_{i=1}^{k^l}$ is the image segmentation partition at hierarchy level l that is composed by k^l individual image regions $R_{t,i}^l$ and L is the number of hierarchy levels. This hierarchy contains several image segmentation partitions, each describing a degree of detail of the image $\mathcal{I}_t$ (from fine to coarse levels). The coarser the level the higher the merging

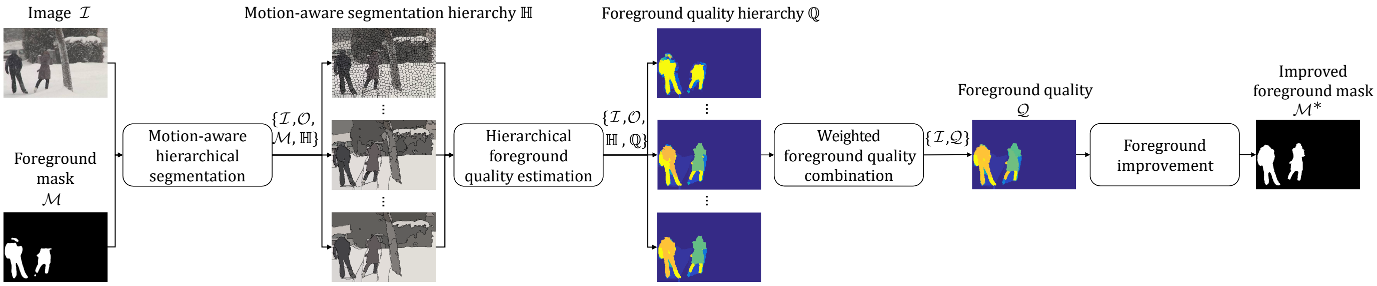


Image $\mathcal{I}$
Motion-aware segmentation hierarchy $\mathbb{H}$
Foreground quality hierarchy $\mathbb{Q}$
Foreground mask $\mathcal{M}$
Motion-aware hierarchical segmentation
$\{\mathcal{I},\mathcal{O},\mathcal{M},\mathbb{H}\}$
Hierarchical foreground quality estimation
$\{\mathcal{I},\mathcal{O},\mathbb{H},\mathbb{Q}\}$
Weighted foreground quality combination
$\{\mathcal{I},\mathcal{Q}\}$
Foreground quality $\mathcal{Q}$
Foreground improvement
Improved foreground mask $\mathcal{M}^*$

Fig. 1: Foreground improvement framework overview. For clarity, we avoid the temporal index t (common to all notation). The motion-aware hierarchy $\mathbb{H}$ computed from the motion-aware color-based UCM (Eq. 1) is explained in Subsection III-B1, while the foreground quality hierarchy $\mathbb{Q}$ is computed using a fitness-to-regions property (Eq. 3) defined in Subsection III-B2. Then, a unique foreground quality $\mathcal{Q}$ is estimated using the weighted combination (Eq. 4) from Subsection III-B3. Finally, the improved foreground mask $\mathcal{M}^*$ is obtained via optimal labeling (Eq. 10) as presented in Subsection III-B4.

of regions, thus covering larger object areas. We consider spatial similarities based on color and introduce motion constraints through the optical flow $\mathcal{O}_t$ in order to avoid merging foreground and background regions in each partition of the hierarchy $\mathbb{H}_t$. Then, we estimate a foreground quality image for each level of the hierarchy $\mathcal{Q}_t^l$ using a fitness-to-region property, thus obtaining a foreground quality hierarchy $\mathbb{Q}_t = \left\{\mathcal{Q}_t^l\right\}_{l=1}^{L}$. The quality image Q_t^l of each level has the same size as $\mathcal{I}_t$ where each pixel is a score denoting its foreground quality. Subsequently, all levels of foreground qualities are combined to estimate a unique foreground quality image $\mathcal{Q}_t$ using a weighted average scheme based on the optical flow magnitude. This weighted average increases the importance of coarse levels in $\mathbb{H}_t$ for high optical flow magnitudes, as the presence of strong motion boundaries prevents an undesired foreground-background merging. Finally, we use a Conditional Random Field (CRF) to obtain an improved foreground mask $\mathcal{M}_t^*$ through an optimal labeling process that combines both foreground quality and spatial information. For simplifying notation, the temporal index is omitted in Figure 1 and in the following subsection.

B. Description

1) Motion-aware hierarchical segmentation: Merging superpixels to estimate semantically meaningful image regions containing objects is a common practice in the object proposal literature [61][62][63]. Building on such idea, we compute a motion-aware hierarchical image segmentation that extends over different degrees of details through each level partition into regions while preventing foreground-background merging.

A complete hierarchy of partitions can be defined as the set of all image segmentation results $\mathbb{H}' = \{\mathcal{R}^n\}_{n=1}^N$ where the level index n goes from the finest segmentation $\mathcal{R}^1$ (i.e. superpixels) to the coarsest segmentation $\mathcal{R}^N$ (i.e. complete image domain). The complete hierarchy can be understood as a dendrogram (tree) of regions where coarse levels are built merging regions from finer ones according to adjacent regions similarities. Such complete hierarchy can be computed through an ultrametric contour map (UCM) [64], which is a boundary map that can be thresholded to obtain a set of closed boundaries containing segmented image regions. The lowest threshold leads to $\mathcal{R}^1$, while the highest threshold produces $\mathcal{R}^N$. Monotonically increasing the threshold merges the superpixels whose dissimilarity is under the threshold. Therefore, superpixels and their dissimilarities are required to compute the UCM by applying a greedy graph-based region merging algorithm [64]. In particular, we have used the Piotr Dollar's proposal[1] which employs the mean boundary value [65] as dissimilarity between SLIC based superpixels [66]. Figure 2 presents an image (a), whose UCM [64] (d) is extracted from superpixels (c) and dissimilarities defined by image boundaries [65] (b). Therefore, thresholding the UCM with increasing values provides coarser partitions as presented in Figure 2 (c) and (e). We name this UCM based on color image properties as color-based UCM $\mathcal{U}^{col}$. While merging regions to fit foreground objects, merging between adjacent foreground regions is expected to occur before foreground-background merging. However, computing the hierarchy relying on appearance similarities as done by the color-based UCM $\mathcal{U}^{col}$ does not necessarily lead to the desired result (i.e. foreground and background not merged in the same regions). For example, in Figure 2 the color-based UCM $\mathcal{U}^{col}$ (d) of an image (a) lacks of boundaries in the top front part of a car due to color similarities with background regions. Therefore, we address such problem by including motion constraints to prevent foreground-background merging. We first create a motion-based UCM $\mathcal{U}^{mot}$ (see Figure 2(h)) based on per-pixel optical flow magnitude [67] (see Figure 2(f)) which defines moving object boundaries (see Figure 2(g)). To obtain $\mathcal{U}^{mot}$, we extract boundaries and superpixels over the optical flow magnitude (replicated to 3 channels). Similarly to [68], we do not re-train the boundary detector [65] (trained for static image boundaries) as it effectively detects motion boundaries (see Figure 2(g)) and re-training may confuse the detector due to the misalignment of optical flow boundaries with the true image boundaries. Then, $\mathcal{U}^{mot}$ and $\mathcal{U}^{col}$ are combined into the motion-aware color-based UCM $\mathcal{U}$ (see Figure 2(i)):

$$\mathcal{U} = f_{ucm}\left(\mathcal{U}^{col},\mathcal{U}^{mot}\right), \tag{1}$$

where $f_{ucm}(\cdot,\cdot)$ is the combination function applied to $\mathcal{U}^{mot}$ and $\mathcal{U}^{col}$. We propose a combination to keep only strong

[1] https://github.com/pdollar/edges

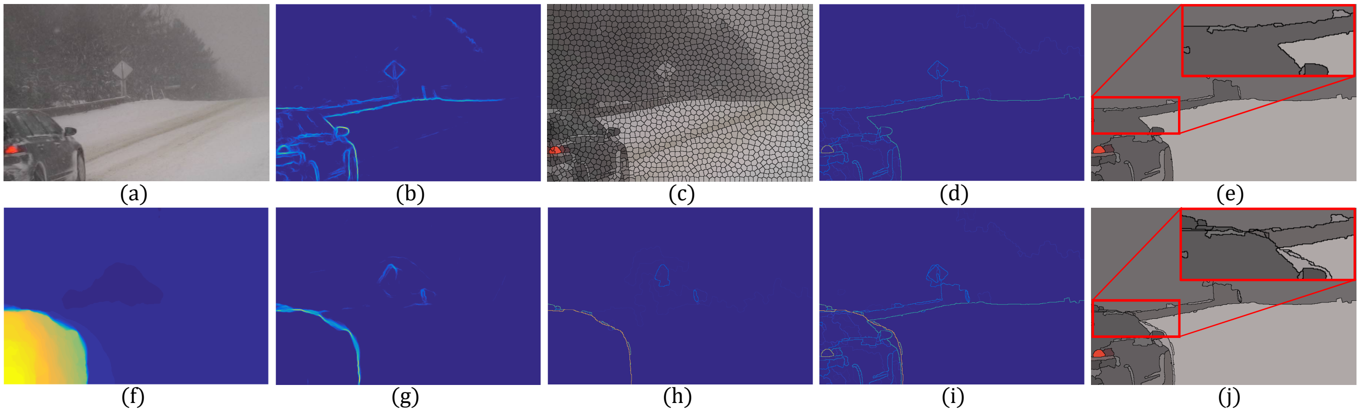


Fig. 2: Examples of ultrametric contour map (UCM) [64] and motion-aware image segmentation. The UCM (d) of an image (a) is obtained through superpixels (c) and their similarities (b), whereas thresholding the UCM leads to different image segmentation partitions (c)-(e). Furthermore, given the optical flow magnitude (f) for image (a), we name the UCM from (d) as color-based UCM $\mathcal{U}^{col}$ and compute a motion-based UCM $\mathcal{U}^{mot}$ (h). This $\mathcal{U}^{mot}$ is obtained from motion boundaries (g) computed from the optical flow magnitude (f). Combining both UCMs we obtain a motion-aware color-based UCM $\mathcal{U}$ (i) that produces an image segmentation (j) with no foreground-background merging, unlike the direct use of $\mathcal{U}^{col}$ (e). The top-right rectangle of (e)(j) zooms an area to observe differences between merged regions.

boundaries of the motion UCM $\mathcal{U}^{mot}$, thus obtaining the motion-aware color-based UCM $\mathcal{U}$ as:

$$\mathcal{U}^{\mathbf{p}} = \begin{cases} \max\left(\mathcal{U}^{\mathbf{p},col}, \mathcal{U}^{\mathbf{p},mot}\right) & if \quad \mathcal{U}^{\mathbf{p},mot} > \lambda^{L} \\ 0 & \quad otherwise \end{cases}, \quad (2)$$

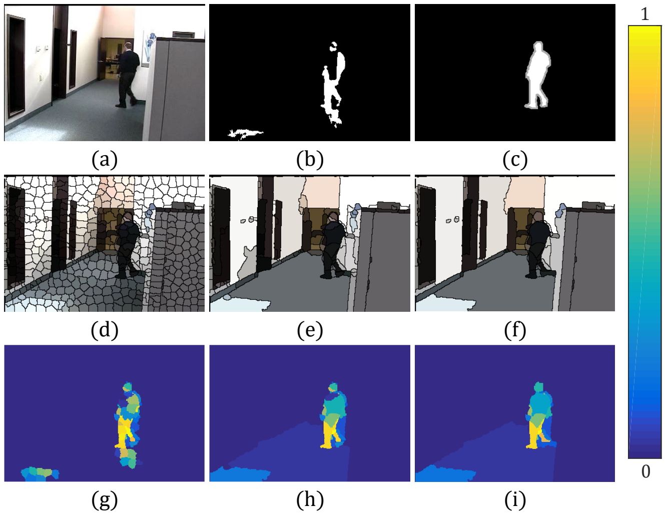
where $\mathbf{p}$ is the 2D pixel location in the UCM maps and λ^L is a threshold large enough to assure that only strong motion boundaries are added. This combination employs color merging while introducing only strong motion boundaries, thus preventing from over-segmentation due to weak motion boundaries that may appear. Therefore, the motion-aware color-based UCM $\mathcal{U}$ allows the computation of a complete hierarchy $\mathbb{H}'$ that prevents foreground-background merging.

Foreground segmentation requires foreground-background separation, thus we need each image region to contain foreground or background without merging both classes. This desired result does not occur for partitions close to $\mathcal{R}^N$ (i.e. partitions close to the complete image domain that tend to contain foreground and background merged), thus we sample the complete hierarchy to get a hierarchy $\mathbb{H} \subset \mathbb{H}'$ conformed by a subset of L levels (as introduced in Subsection III-A) starting from the finest one. To that end, we threshold $\mathcal{U}$ to produce an image segmentation where foreground and background are not merged (see Figure 2(j)), whereas directly thresholding $\mathcal{U}^{col}$ merges both classes (see Figure 2(e)). We uniformly threshold $\mathcal{U}$ with L thresholds or levels ranging from the finest one (i.e. superpixels) to a maximum value. The result after applying the multiple thresholds is a motion-aware color segmentation hierarchy $\mathbb{H} = \left\{\mathcal{R}^l\right\}_{l=1}^{L}$ (see Figure 1), where each level l is composed by an image segmentation partition $\mathcal{R}^l$ obtained applying a threshold $\lambda^l = s(l-1)$ over $\mathcal{U}$ and s is the step between levels. We avoid using a single threshold λ generating a unique image segmentation that may have errors. Instead we consider selecting a number of levels L (i.e. $\left\{\lambda^l\right\}_{l=1}^{L}$) and defining the step between levels s to obtain each threshold λ^l (note that λ^L from Eq. 2 corresponds to the coarsest level threshold). Therefore, using a high (low) value of s means that there are less (more) λ^l possible values from the finest to the coarsest segmentation. Then, fixing the step between consecutive levels s and varying L reveals the effect of including more levels as analyzed in Subsection IV-B1. This hierarchy $\mathbb{H}$ serves as the basis of the hierarchical foreground quality estimation, presented in Subsection III-B2, to extend foreground for different image partitions.

2) *Hierarchical foreground quality estimation:* Based on the potential of image regions to estimate blob-level foreground quality [51], we employ the property of fitness-to-regions to extend detected foreground blobs over foreground objects while removing erroneous foreground pixels. For each hierarchy level l, we compute a foreground quality q_i^l for each region R_i^l as:

$$q_i^l = \frac{\sum\limits_{\mathbf{p}\in R_i^l} \mathcal{M}^{\mathbf{p}}}{\left|R_i^l\right|}, \quad (3)$$

where $|\cdot|$ denotes cardinality (i.e. $|R_i|$ is the number of pixels in region R_i) and $\mathcal{M}^{\mathbf{p}}$ is the pixel location $\mathbf{p}$ in the foreground segmentation mask $\mathcal{M}$ with values of 1(0) for foreground (background). This per-region quality q_i^l measures the fitness of the foreground mask to the region R_i through its percentage of foreground pixels. Therefore, the per-level foreground quality image is defined as an image $\mathcal{Q}^l = \left\{q_i^l\right\}_{i=1}^{k^l}$ with the same size of the image $\mathcal{I}$, where q_i^l is the quality per-region R_i^l and k^l is the number of regions in level l. Furthermore, the set of quality images per-level form a foreground quality hierarchy $\mathbb{Q} = \left\{\mathcal{Q}^l\right\}_{l=1}^{L}$ that is combined to obtain a unique foreground quality image as depicted in Subsection III-B3. Figure 3 shows examples of foreground qualities $\mathcal{Q}^l$ (g)-(i) extracted from fitness of the foreground segmentation mask $\mathcal{M}$ (b) of image $\mathcal{I}$ (a) to different segmentation partitions (d)-(f) of the hierarchy $\mathbb{H}$, having in fine (detailed) levels a weak



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 3: Hierarchical quality estimation. The image under analysis (a) has an associated foreground mask (b) that can be improved to accurately detect foreground as done by the ground-truth (c). The fitness between the foreground mask and the several image segmentation partitions (d)-(f) is the per-level quality $\mathcal{Q}^l$ shown in (g)-(i).

spatial extension of the quality scores and high fitness to false positives of the foreground mask, while coarse levels enlarge regions covering foreground objects and diffusing foreground errors over background regions.

3) Weighted foreground quality combination: Given all the foreground quality images $\mathbb{Q} = \left\{\mathcal{Q}^l\right\}_{l=1}^{L}$, we obtain a unique foreground quality $\mathcal{Q}$ by combining all levels instead of selecting the best one, as such selection is not trivial. When no foreground-background merging is guaranteed, the coarsest level would be the best choice. However, stationary or slowly moving objects have, respectively, no motion boundaries or weak ones, thus easing foreground-background merging in coarse levels. Therefore, we perform a per-pixel weighted average to combine all levels by assigning different weights to each level based on the pixel optical flow magnitude.

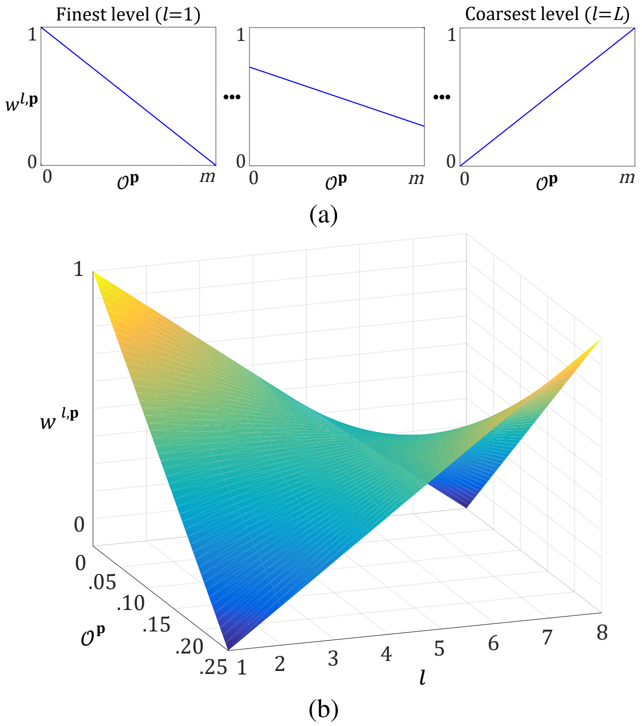
In video sequences, we can distinguish between stationary objects or background and moving foreground objects through motion data. This premise has already been introduced in the hierarchy through strong motion boundaries provided by $\mathcal{U}^{\mathbf{p},mot}$ and it can also be used to estimate $\mathcal{Q}$ through a weighted average as:

$$\mathcal{Q}^{\mathbf{p}} = \frac{\sum_{l} w^{l,\mathbf{p}} \left(\|\mathcal{O}^{\mathbf{p}}\|\right) \mathcal{Q}^{l,\mathbf{p}}}{\sum_{l} w^{l,\mathbf{p}} \left(\|\mathcal{O}^{\mathbf{p}}\|\right)}, \qquad (4)$$

where $w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right)$ is the level l weighting function for pixel location $\mathbf{p}$ based on the optical flow magnitude $\|\mathcal{O}^{\mathbf{p}}\|$ associated to $\mathbf{p}$. We propose a weighting function linear with the level indexes and the motion values:

$$w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right) = \left[\frac{2d\left(l-1\right)-1}{m}\right]\|\mathcal{O}^{\mathbf{p}}\| + \left[1-d\left(l-1\right)\right], \qquad (5)$$

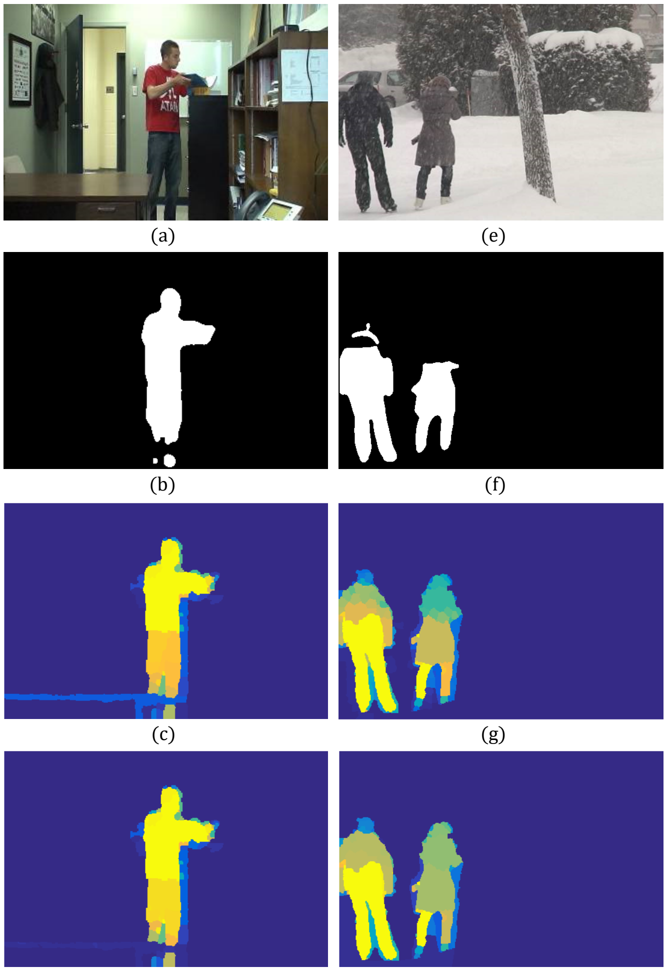
where $d = \frac{1}{L-1}$ and m is an upper bound for $\|\mathcal{O}^{\mathbf{p}}\|$ that assures maximum confidence in the coarsest level when there is enough motion (see Subsection IV-B1 for an analysis of the effect of m in the performance). The higher the motion the higher the weight value for coarse levels (see the right subfigure in Figure 4(a)) where foreground is highly merged and the motion-aware UCM $\mathcal{U}$ has strong motion boundaries preventing foreground-background merging. However, for low $\|\mathcal{O}^{\mathbf{p}}\|$ values the combined UCM does not guarantee avoiding foreground-background merging, thus the coarser the level the lower the weight (see the left subfigure in Figure 4(a)) to reduce the contribution of coarse levels that may merge foreground and background. Therefore, the intermediate levels weights (see the middle subfigure in Figure 4(a)) range between the aforementioned finest and coarsest level weights. Additionally, the weighting function $w^{l,\mathbf{p}}\left(\|\mathcal{O}^{\mathbf{p}}\|\right)$ can be represented in 3D as depicted in Figure 4(b). In Figure 5 we present examples comparing the proposal and an equally weighted average (i.e. mean). In the first column, an image (a) and its foreground segmentation mask (b) contain a stationary person. The absence of motion induces a foreground-background merging that leads to the extension of scores out of the foreground area when equally weighting the per-



(a)

(b)

Fig. 4: Weighting function to combine all hierarchy levels. Weights for each level are shown in (a), where the finest level (a)-left is weighted with maximum (minimum) weight in stationary (moving) pixels and the coarsest level (a)-right is weighted exactly in an opposite fashion. This assures that in cases of moving regions, where motion boundaries prevents from foreground-background merging, higher confidence is assigned to the coarsest level. The intermediate levels weights (a)-middle are defined to progressively move from the finest to the coarsest weight. In (b), the complete weighting function is presented with the parameters used, $L = 8$ and $m = 0.25$.



(a) (e)

(b) (f)

(c) (g)

(d) (h)

Fig. 5: Example for the effect of the proposed weighted average. For each row, from top to bottom: images under analyses (a)(e), segmented foreground masks (b)(f) and foreground quality $\mathcal{Q}$ applying, respectively, an equally weighted average (c)(g) and the proposed weighting (d)(h).

level foreground qualities (c), whereas the proposed weighting palliates such merging by assigning a higher weight value to fine levels in absence of motion. Conversely, in the second column an image (e) contains moving people that are not fully segmented in its foreground segmentation mask (f). The presence of motion allows improving the foreground quality obtained by applying equal weights (g) through the proposed weighting that assigns higher scores in the unsegmented top parts of the people due to the higher importance of coarse levels in presence of motion (h).

For scenarios with camera jitter, our assumption for the optical flow magnitude is not satisfied, leading to $\|\mathcal{O}^{\mathbf{p}}\|$ values exceeding m and therefore promoting coarse levels. In these cases we simply average all hierarchy qualities to compute $\mathcal{Q}$. The detection of frames affected by camera jitter is conducted using the average value of the temporal median of the optical flow magnitudes over large temporal windows.

4) Foreground improvement: Foreground mask improvement can be performed by thresholding the quality image $\mathcal{Q}$, as it expands over detected an undetected foreground regions. However, as motion boundaries used to restrict foreground-background merging are often not fitted to foreground object contours, a simple thresholding may add erroneous foreground pixels to the improved mask. Therefore, we introduce additional constraints to reduce such misclassifications near foreground object contours using a pixel-wise Conditional Random Field (CRF), which provides a robust framework to incorporate such constraints via spatial information potentials.

Using a CRF casts foreground segmentation into a binary pixel labeling problem, where a labeled image $\mathcal{C}$ has either foreground $\mathcal{C}^{\mathbf{p}} = 1$ or background $\mathcal{C}^{\mathbf{p}} = 0$ pixels. We use the fully-connected CRF model of [69] to compute the optimal labeling $\mathcal{C}^*$ after an energy minimization process. The energy is defined over pixels and their labels as:

$$E\left(\mathcal{C}\right) = \sum_{\mathbf{p}\in\mathcal{I}} f_u\left(\mathcal{C}^{\mathbf{p}}\right) + \sum_{\mathbf{p}\in\mathcal{I}} \sum_{\mathbf{q}\in\mathbb{N}_{\mathbf{p}}} f_p\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right), \qquad (6)$$

where f_u is a unary potential function to define the foreground probability, f_p is a pairwise potential function for labeling smoothness by penalizing neighboring pixels taking different labels and $\mathbb{N}_{\mathbf{p}}$ is the set of neighbors of pixel location $\mathbf{p}$.

For the pairwise potential f_p we use the model from [69]:

$$f_p\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right) = \mu\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right) \left[w_1 \exp\left(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma_\alpha^2} - \frac{\|\mathcal{I}^{\mathbf{p}} - \mathcal{I}^{\mathbf{q}}\|^2}{2\sigma_\beta^2} \right) + w_2 \exp\left(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma_\gamma^2} \right) \right], \qquad (7)$$

where each term is multiplied by $\mu\left(\mathcal{C}^{\mathbf{p}}, \mathcal{C}^{\mathbf{q}}\right) = 1$ if $\mathcal{C}^{\mathbf{p}} \neq \mathcal{C}^{\mathbf{q}}$ and zero otherwise to penalize locations with distinct labels; the first term is an appearance Gaussian kernel based on RGB and pixel location euclidean distances that aims to assign the same label to pixels with similar color and near positions; the second term is a Gaussian kernel dependent on pixel location euclidean distance to smooth the label assignment by removing isolated labels; the parameters σ_α, σ_β and σ_γ control the scale of the kernels; and w_1 and w_2 weight the contribution of each kernel to the pairwise potential. We set $\sigma_\alpha = 10$, $\sigma_\beta = 5$, $\sigma_\gamma = 3$, $w_1 = 1$ and $w_2 = 1$ that are all default parameters[2] in the implementation used, except σ_α and σ_β that have been set to a smaller value in order to limit long range spatial connections that may decrease foreground segmentation performance due to similarities between foreground and background colors in the scene (we refer the reader to the additional material in http://www-vpu.eps.uam.es/publications/HFI/ for an experiment on these parameters). The pairwise potential in [69] was originally used for semantic segmentation in scenarios where foreground and background colors better define foreground and background classes, thus higher σ_α and σ_β values lead to extremely accurate foreground segmentation.

Moreover, we define the unary potential function f_u as:

$$f_u\left(\mathcal{C}^{\mathbf{p}}\right) = -\ln\left(\mathcal{F}^{\mathbf{p}}\right), \qquad (8)$$

where $\mathcal{F}$ is a foreground probability estimated from the foreground quality image $\mathcal{Q}$. Such estimation is performed in order

[2] https://github.com/johannesu/meanfield-matlab

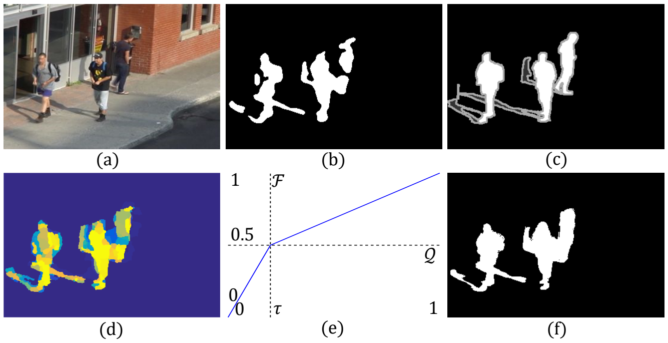


Fig. 6: Example of the foreground segmentation process. An image $\mathcal{I}$ (a) with foreground segmentation mask $\mathcal{M}$ (b) has a ground-truth shown in (c). The foreground quality $\mathcal{Q}$ (d) is linearly transformed to resemble a probability (optimal foreground-background separation threshold τ for quality $\mathcal{Q}$ is mapped into 0.5 in $\mathcal{F}$) and compute an improved foreground mask $\mathcal{M}^*$(f) through a CRF.

to transform $\mathcal{Q}$ into an information resembling a probability as needed by the CRF to correctly perform the foreground segmentation through maximum a posteriori inference. To that end, we perform a linear mapping between $\mathcal{Q}$ and $\mathcal{F}$ (see Figure 6(e)) as:

$$\mathcal{F}^{\mathbf{P}} = \begin{cases} \frac{0.5}{\tau}\,\mathcal{Q}^{\mathbf{P}}, & if \quad \mathcal{Q}^{\mathbf{P}} \leq \tau \\ \frac{0.5}{1-\tau}\,\mathcal{Q}^{\mathbf{P}} + \frac{0.5-\tau}{1-\tau}, & if \quad \mathcal{Q}^{\mathbf{P}} > \tau \end{cases}, \tag{9}$$

where τ is the foreground-background separation threshold associated to 0.5 foreground probability after the mapping (we analyze the effect of τ in the performance in Subsection IV-B2). Note that linearly mapping fitness between superpixels and probability scores has been successfully performed in the literature [44].

Finally, we obtain the improved foreground mask $\mathcal{M}^*$ as the optimal labeling:

$$\mathcal{M}^* = \arg\min_{\mathcal{C}} E\left(\mathcal{C}\right). \tag{10}$$

Figure 6 depicts the foreground segmentation process of the image (a) given the foreground quality $\mathcal{Q}$ (d) of the foreground segmentation mask $\mathcal{M}$ (b) with associated ground-truth (c). The foreground quality $\mathcal{Q}$ is transformed into a foreground probability $\mathcal{F}$ using the linear transformation shown in Figure 6(e) to compute the improved foreground mask $\mathcal{M}^*$ in Figure 6(f). Note that in the example presented in Figure 6 τ is set to 0.2. Furthermore, in Figure 7 an example image (a) is segmented with errors (b) compared to ground-truth (c) and has an estimated foreground probability $\mathcal{F}$ (d) that leads to different improved foreground masks (e)-(f) depending on the technique applied. The foreground mask presented in (e) is obtained by directly applying maximum a posteriori inference over $\mathcal{F}$ (i.e. thresholding over 0.5 without considering the pairwise potential), thus leading to errors in the object contours that are mostly solved in $\mathcal{M}^*$ (f) as it jointly considers the unary and pairwise potentials via the CRF framework.

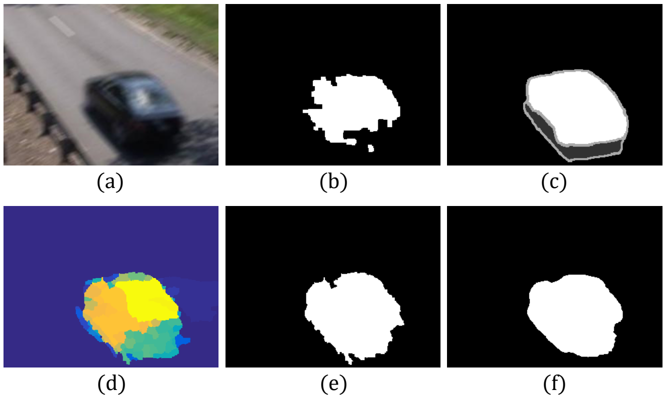


Fig. 7: Example of foreground segmentation improvement when using or not the pairwise potential. An image $\mathcal{I}$ (a) with foreground segmentation mask $\mathcal{M}$ (b) has a ground-truth shown in (c). Maximum a posteriori inference over the foreground probability $\mathcal{F}$ (d) leads to the foreground segmentation shown in (e) when only $\mathcal{F}$ is used, whereas $\mathcal{M}^*$ (f), obtained through the CRF that considers spatial information, produces a better foreground mask.

IV. Experimental work

A. Experimental methodology

We use real and synthetic sequences from four datasets: the well-known CDNET2014 dataset [1], the recent LASIESTA dataset [52] and the synthetic datasets SABS [53] and BMC [54]. These datasets contain common BS challenges with their corresponding ground-truth data. For CDNET2014, we select eight of the eleven categories (*PTZ*, *Thermal* and *Turbulence* are excluded) as the proposed framework has been designed for color images in stationary camera scenarios, thus using 40 sequences (113848 frames). For LASIESTA, we select both indoor and outdoor sequences discarding those involving moving cameras (*MC Moving Camera* and the first three sequences of *SM Simulated Motion*), thus using 38 sequences (16250 frames). For the SABS synthetic dataset, we select 8 of the 12 sequences (6400 frames) and discard 4 out of 5 sequences with different compression qualities. For the BMC synthetic dataset, we use 10 sequences from the learning category (14990 frames). We do not use the rest due to the extremely low availability of ground-truth for long sequences. Note that we do not use unconstrained video object segmentation datasets [5][70] as they consider that moving objects may not be part of the foreground.

To apply the proposed post-processing framework, we analyze the datasets with several algorithms (see Table I for a brief summary) to demonstrate that the improvement achieved is generalizable: CwisarDH [28], SuBSENSE [42], AMBER [71], MBS [72], PAWCS [73], SharedModel [74], WeSamBE [75], Spectral-360 [76], FTSG [77], LOBSTER [78], SC-SOBS [79], FuzzySOM [80], MLAYER [81], GMM [82] and KDE [83]. We have selected this set of algorithms to demonstrate the framework capability to improve results from low to top performance algorithms. We use the results provided in CDNET2014, whereas we employ the BGSlibrary [84] to

TABLE I: Background subtraction algorithms selected to validate the improvement obtained by the proposed post-processing framework. Key: C: Color. T: Texture. M: Motion.

Algorithm	Model type description	Features	Dataset			
			CDNET	LASIESTA	SABS	BMC
CwisarDH	Weightless neural network	C	✓			
SuBSENSE	Non-parametric sample-based	C, T	✓	✓	✓	✓
AMBER	Multi-resolution temporal templates	C,T	✓			
MBS	Single Gaussian of multiple features	C	✓			
PAWCS	Non-parametric sample-based	C, T	✓			
SharedModel	Mixture of Gaussians	C	✓			
WeSamBE	Non-parametric sample-based	C	✓			
Spectral-360	Dichromatic reflection model	C	✓			
FTSG	Flux tensor and mixture of Gaussians	C, M	✓			
LOBSTER	Non-parametric sample-based	C, T		✓	✓	✓
SC-SOBS	Self-organized neural network	C	✓			
FuzzySOM	Self-organized neural network	C		✓	✓	✓
MLAYER	Layer-based	C, T		✓	✓	✓
GMM	Mixture of Gaussians	C	✓	✓	✓	✓
KDE	Non-parametric kernel	C		✓	✓	✓

TABLE II: Example of the effect of L and m in the F-score. The higher *L*, the better the performance until too coarse levels are used and foreground-background merging occurs (see *L*=32 and *L*=64). The selection of the parameter m has low impact in the F-score. $\%\Delta Fs = \frac{Fs^{new} - Fs^{old}}{Fs^{old}}$ denotes the improvement percentage achieved in terms of average F-score. Note that $\tau = 0.25$ is used for the experiment.

		m				**Mean**	$\%\Delta Fs$
		0.25	**0.5**	**0.75**	**1**		
L	**1**	.7852	.7852	.7852	.7852	.7852	-
	2	.7911	.7909	.7908	.7906	.7908	0.71
	4	.7958	.7954	.7952	.7950	.7953	0.57
	8	.8011	.8006	.8003	.8001	.8005	0.65
	12	.8044	.8040	.8038	.8035	.8039	0.42
	16	.8069	.8066	.8066	.8065	.8067	0.34
	32	.8005	.8015	.8027	.80394	.8021	-0.57
	64	.5062	.5331	.5555	.5705	.5413	-32.51

run selected algorithms in the remaining datasets. We do not consider recently emerged deep learning models [37][40] as they currently rely on the ground-truth data for training from the same sequences in which tests are performed. Also, we have selected a top (SuBSENSE) and a low (GMM) performing algorithm across all datasets to compare performance among databases.

To assess the algorithms performance, we use standard Precision (Pe), Recall (Re) and F-score (Fs) based on pixel-level comparisons between foreground segmentation masks and ground-truth. These measures are computed as:

$$Pe = TP/\left(TP + FP\right), \tag{11}$$

$$Re = TP/\left(TP + FN\right), \tag{12}$$

$$Fs = 2 \cdot Pe \cdot Re/\left(Pe + Re\right), \tag{13}$$

where *TP*, *FP* and *FN* are, respectively, correct, false and missed detection pixels (as compared to ground-truth ones).

B. Effect of parameters in performance improvement

1) Number of levels and optical flow bounding: The use of a hierarchy to extend over foreground objects is one of the main contributions of this paper. This hierarchy has a predefined number of levels that are combined using a weighted average dependent on the upper bound m for $\|\mathcal{O}^{\mathbf{p}}\|$. We present in Table II the impact of these parameters values in the average F-score of six sequences from CDNET2014 dataset (*skating, highway, canoe, winterDriveway, tramCrossroad_1fps and cubicle*) segmented with SuBSENSE and GMM. Firstly, a higher number of hierarchy levels *L* leads to higher performance due to larger extensions of uncompleted foreground objects and the removal of more erroneous foreground pixels through low fitness-to-regions values. Secondly, the value of m is related to the optical flow magnitude and the importance given to coarse levels. The lower the value the better, but its value has little impact in the performance. Attending to Table II, we have used *L*=8 due to its pick of performance increment ($\%\Delta L$) and $m = 0.25$ as it provides slightly better results than the rest of the values analyzed. Additionally, we have heuristically set the step s to 0.015, thus leading to the coarsest level *L*=8 using a threshold $\lambda^L = 0.105$. Note that heuristically setting the number of levels and using a step to threshold an UCM are common practices in the literature [85][86].

2) Linear mapping: The transformation of foreground quality to foreground probability is done through a linear mapping guided by parameter τ (see Eq. 9). Therefore, we sweep the value of $\tau \in [0, 1]$ to find out how its value affects the improvement capabilities (using $L = 8$, $m = 0.25$). In particular, we compare the original algorithm performance against the performance obtained by the proposed improvement framework when only a unary or both a unary and a pairwise potential are used in the CRF energy function.

We have performed this experiment in CDNET2014 (using CwisarDH, SuBSENSE, AMBER, MBS, FTSG, SC-SOBS and GMM) and in LASIESTA (using the six algorithms evaluated) datasets. For space constraints, we have selected SuBSENSE and AMBER and SuBSENSE and FuzzySOM as top and medium performance algorithms, respectively, in CDNET2014 and LASIESTA datasets. The remaining algorithm results are available online (http://www-vpu.eps.uam.es/publications/HFI/). Figures 8 and 9 present the average performance achieved in terms of Pe, Re and Fe (columns) for each pair of algorithms (rows) in CDNET2014 and LASIESTA datasets, respectively. In general terms, using a unary potential alone (superscript *1 in the figures) improves recall for low values of τ (approximately between 0.1 and 0.5), thus supporting the capability to extend over foreground objects. However, this recall improvement comes with the reduction of the precision due to contour-inaccurate partitions in the motion-aware hierarchy that lead to an extension of foreground masks not fitted to objects contours. This precision reduction is overcome by including the pairwise potential in the CRF energy function (superscript *2 in the figures), which is able to fit foreground masks to object contours while keeping and improved recall (see Figure 7). Therefore, as shown in Figures 8 and 9, we can conclude that a good value of τ is approximately between 0.2 and 0.3 as both precision and recall are improved and the CRF with both unary and pairwise potentials outperforms the use of the unary potential alone, thus we select the unary and pairwise based CRF to present the results in the following subsections.

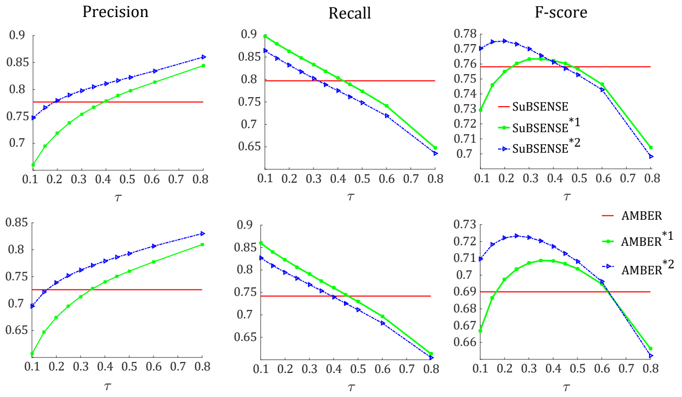


Fig. 8: Examples of the effect of τ parameter in the performance of SuBSENSE [42] and AMBER [71] algorithms. Each row denotes an algorithm, whereas each column presents, respectively, the average Precision (Pe), Recall (Re) and F-score (Fs) in CDNET2014 dataset. In each figure, the red line denotes the performance of the algorithm in the dataset, the green line with dots is the performance achieved by applying maximum a posteriori inference only using the foreground probability $\mathcal{F}$ (*1) and the blue line with triangles is the performance using both $\mathcal{F}$ and the pairwise potential (*2).

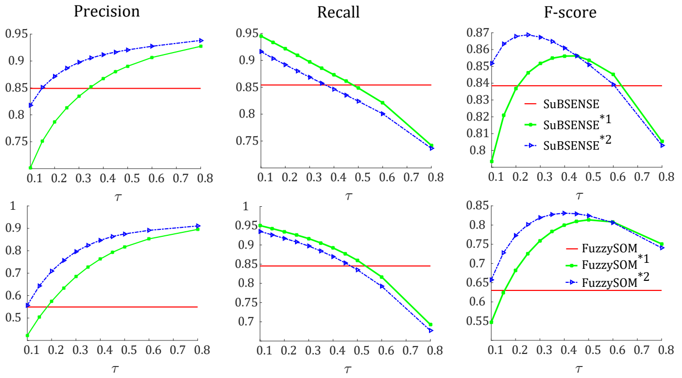


Fig. 9: Examples of the effect of τ parameter in the performance of SuBSENSE [42] and FuzzySOM [80] algorithms. Each row denotes an algorithm, whereas each column presents, respectively, the average Precision (Pe), Recall (Re) and F-score (Fs) in LASIESTA dataset. In each figure, the red line denotes the performance of the algorithm in the dataset, the green line with dots is the performance achieved by applying maximum a posteriori inference only using the foreground probability $\mathcal{F}$ (*1) and the blue line with triangles is the performance using both $\mathcal{F}$ and the pairwise potential (*2).

C. Improvement over the original algorithms in CDNET2014, LASIESTA, SABS and BMC datasets

We present the improvement in all datasets results for a fixed configuration of $L = 8$, $m = 0.25$ and $\tau = 0.25$. In Table III, we show the average performance results in terms of Pe, Re and Fs, together with the percentage increases of Fs for LASIESTA, SABS and BMC datasets. In these datasets, improvements are obtained for all algorithms on average and we present some examples of these improvements in Figures 10 and 11 for, respectively, LASIESTA and SABS and BMC datasets. Moreover, we present per-category and overall performance results for the CDNET2014 dataset in Table IV. Note that an improvement of around 2% for top algorithms in CDNET2014 (SuBSENSE, FTSG, WeSamBE or SharedModel) is a significant one as the percentage between the first and fifth performing unsupervised algorithms in CDNET2014[3] is 2.6%. From the tables it can be observed that results are better than the original performance for almost all algorithms and categories in CDNET2014 (see Table IV) with the exception of the *Intermittent Object Motion* category for top-performing algorithms (SuBSENSE, FTSG and WeSamBE), where there are weak decreases in performance due to the static nature of most of the foreground objects in this category. This stationarity leads to no foreground-background merging prevention when performing the hierarchical image segmentation, thus foreground probabilities are easily expanded over background regions and foreground regions are less extended to undetected areas due to the lower importance of high levels in the hierarchy when combining the quality images. Additionally, Figures 12 presents some examples of improvements achieved in CDNET2014. Please, see online (http://www-vpu.eps.uam.es/publications/HFI/) all the foreground masks and complete performance results.

D. Comparison against the state-of-the-art

We compare our improvement capabilities against available similar approaches in the literature, i.e. approaches aiming to improve foreground masks from a model-independent perspective. In particular, we present in Table V the improvements over SOBS algorithm [87] (a previous version of the already evaluated SC-SOBS algorithm) for the algorithm in [44] and the proposed framework using $\tau = 0.25$ (marked with *). We use SOBS algorithm in 5 categories of CDNET2014 as that are the categories and algorithm with available results. Despite the use of fitness-to-regions by [44], we achieve a superior improvement as we introduce a hierarchical approach that enables the extension of the segmented foreground masks to undetected foreground areas while fitting to object contours. We have a higher improvement in all categories attending to Pe, Re and Fs and we only perform worse for Pe in Shadow category, where we decrease in a 0.3% the original Pe performance. Note that we only compare our post-processing improvement capabilities against [44] as other model-independent post-processing works, such as [45] and [59], do not provide code to reproduce the complete post-processing technique.

E. Applying foreground quality to algorithm combination

Recently, combining BS algorithms results demonstrated to obtain substantially better results [88]. Adopting this idea, we present here a potential use of the foreground quality image as the information to guide the algorithm combination. For each frame we average the foreground quality images from a set of algorithms and we use that image as the quality $\mathcal{Q}$ to feed to foreground improvement from Subsection

[3] http://changedetection.net/

TABLE III: Overall average performance for each analyzed algorithm and the proposed improvement in LASIESTA, SABS and BMC datasets. $\%\Delta Fs = \frac{Fs^{new} - Fs^{old}}{Fs^{old}}$ denotes the improvement percentage achieved for F-score.

	LASIESTA			
	Pe	Re	Fs	%ΔFs
SuBSENSE	.8491	.8542	.8385	3.60
SuBSENSE*	.8867	.8801	.8687	
LOBSTER	.6899	.8204	.7159	6.90
LOBSTER*	.7416	.8534	.7650	
FuzzySOM	.5491	.8452	.6299	27.20
FuzzySOM*	.7572	.9077	.8011	
MLAYER	.6514	.8237	.6749	6.13
MLAYER*	.6916	.8541	.7163	
GMM	.3227	.9234	.4134	17.30
GMM*	.4001	.9784	.4849	
KDE	.3792	.9493	.5013	36.7
KDE*	.5947	.9626	.6853	

	SABS			
	Pe	Re	Fs	%ΔFs
SuBSENSE	.7138	.7786	.6740	0.77
SuBSENSE*	.7125	.7961	.6792	
LOBSTER	.7429	.6834	.6555	5.00
LOBSTER*	.7483	.7511	.6883	
FuzzySOM	.4375	.5716	.4861	17.63
FuzzySOM*	.5813	.6618	.5718	
MLAYER	.5392	.7549	.6237	11.45
MLAYER*	.6012	.8315	.6951	
GMM	.5532	.6481	.5685	11.72
GMM*	.6690	.7020	.6351	
KDE	.3369	.7388	.4536	24.21
KDE*	.4658	.8000	.5634	

	BMC			
	Pe	Re	Fs	%ΔFs
SuBSENSE	.8420	.8706	.8494	0.21
SuBSENSE*	.8568	.8597	.8512	
LOBSTER	.8024	.7757	.7359	2.51
LOBSTER*	.8222	.7835	.7544	
FuzzySOM	.7099	.8083	.7166	7.31
FuzzySOM*	.7544	.8434	.7690	
MLAYER	.7686	.8222	.7500	1.96
MLAYER*	.8013	.8171	.7647	
GMM	.6789	.8833	.7448	7.73
GMM*	.7518	.9005	.8024	
KDE	.4934	.7848	.5395	46.45
KDE*	.8123	.8222	.7901	

TABLE IV: Per-category average foreground segmentation performance achieved by the proposed framework in CDNET2014 dataset. Bold denotes better performance of the proposed improvement (*). The Average column denotes average performance across all categories, being $\%\Delta Fs = \frac{Fs^{new} - Fs^{old}}{Fs^{old}}$ the improvement percentage in terms of average F-score.

	Baseline			Bad Weather			Camera Jitter			Dynamic Background			Intermittent Object Motion		
	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs
PAWCS	.9394	.9408	.9397	.9379	.7091	.8059	.8660	.7840	.8137	.9038	.8868	.8938	.8392	**.7487**	.7764
PAWCS*	**.9397**	**.9525**	**.9420**	.9370	**.7950**	**.8576**	**.8732**	**.8078**	**.8213**	**.9194**	**.9018**	**.9074**	**.9302**	.7200	**.8021**
FTSG	.9170	.9513	.9330	.9192	.7393	.8184	.7645	.7717	.7513	.9129	.8691	.8792	.8512	.7813	.7891
FTSG*	.9125	**.9606**	**.9352**	**.9413**	**.8244**	**.8769**	.7753	**.8204**	**.7664**	**.9303**	**.8824**	**.8974**	.8432	**.7873**	.7850
SuBSENSE	.9495	.9520	.9503	.9168	.8121	.8594	.8115	.8243	.8152	.8915	.7768	.8177	.7957	.6578	.6569
SuBSENSE*	.9430	**.9610**	**.9514**	**.9267**	**.8672**	**.8944**	**.8247**	**.8794**	**.8498**	**.9371**	**.7982**	**.8539**	**.8156**	.6270	.6414
SharedModel	.9502	.9545	.9522	.8559	.8387	.8439	.8377	.7960	.8141	.9198	.7597	.8222	.7587	.7182	.6727
SharedModel*	.9419	**.9669**	**.9541**	**.8649**	**.8840**	**.8706**	**.8474**	**.8376**	**.8377**	**.9400**	**.7768**	**.8384**	**.8002**	**.7252**	**.6930**
WeSamBE	.9422	.9422	.9413	.9184	.8017	.8531	.8395	.7777	.7976	.8933	.6796	.7440	.7888	.7472	.7392
WeSamBE*	.9356	**.9589**	**.9466**	**.9281**	**.8598**	**.8908**	**.8660**	**.8354**	**.8417**	**.9283**	**.6819**	**.7656**	**.8093**	.7254	.7381
Spectral-360	.9065	.9616	.9330	.8621	.7175	.7769	.8387	.6696	.7142	.8456	.7819	.7766	.7374	.5878	.5609
Spectral-360*	**.9105**	**.9709**	**.9395**	**.8756**	**.7878**	**.8242**	.8341	**.7306**	**.7471**	**.8906**	**.8085**	**.8317**	**.7804**	.5783	.5518
MBS	.9431	.9158	.9287	.7652	.8312	.7802	.8443	.8321	.8367	.8606	.7637	.7904	.8201	.6386	.7092
MBS*	.9389	**.9330**	**.9356**	**.8354**	**.8780**	**.8483**	**.8727**	**.8857**	**.8788**	**.8950**	**.8045**	**.8169**	**.9403**	**.6069**	**.7132**
AMBER	.8980	.8784	.8813	.9010	.6782	.7698	.8493	.6505	.7107	.7990	.9177	.8436	.7530	.7617	.7211
AMBER*	**.9067**	**.8913**	**.8925**	**.9297**	**.7854**	**.8460**	**.8636**	**.7230**	**.7579**	**.8373**	**.9358**	**.8740**	**.7891**	**.7706**	**.7366**
CwisarDH	.9337	.8972	.9145	.9173	.6697	.7477	.8516	.7437	.7886	.8499	.8144	.8274	.7417	.5549	.5753
CwisarDH*	.9322	**.9577**	**.9446**	**.9412**	**.7391**	**.8004**	**.8809**	**.832**	**.8513**	**.9248**	**.8878**	**.9019**	**.7923**	**.5905**	**.6008**
SC-SOBS	.9341	.9327	.9333	.8412	.5655	.6605	.6286	.8113	.7051	.6283	.8918	.6686	.5896	.7237	.5918
SC-SOBS*	**.9384**	**.9524**	**.9452**	**.8735**	**.6850**	**.7589**	**.7085**	**.8463**	**.7647**	**.6805**	**.9255**	**.7241**	**.8039**	.7065	**.6660**
GMM	.8461	.8180	.8245	.8285	.7152	.7662	.5126	.7334	.5969	.5989	.8344	.6330	.6688	.5142	.5207
GMM*	**.8670**	**.8581**	**.8569**	**.8951**	**.8245**	**.8572**	**.6188**	**.7972**	**.6759**	**.7180**	**.9020**	**.7301**	**.7345**	**.5488**	**.5503**

	Low Framerate			Night Videos			Shadows			Average			
	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	%ΔFs
PAWCS	.6285	.7555	.6433	.5559	.3929	.4171	.8710	.9172	.8913	.8179	.7669	.7726	1.7
PAWCS*	.6285	**.7702**	**.6512**	**.5570**	**.3984**	.4044	.8628	**.9470**	**.9000**	**.8310**	**.7866**	**.7857**	
FTSG	.6996	.7547	.6563	.4179	.6873	.5043	.8535	.9214	.8832	.7920	.8095	.7768	2.0
FTSG*	**.7087**	**.7669**	**.6673**	**.4268**	**.7196**	**.5158**	.8503	**.9561**	**.8973**	**.7985**	**.8397**	**.7926**	
SuBSENSE	.6276	.8435	.6594	.4224	.6494	.4918	.8646	.9419	.8986	.7849	.8072	.7687	2.1
SuBSENSE*	**.6353**	**.8600**	**.6763**	**.4317**	**.6832**	**.5083**	.8602	**.9596**	**.9041**	**.7968**	**.8294**	**.7849**	
SharedModel	.7362	.8342	.7696	.4030	.5810	.4663	.8455	.9445	.8898	.7884	.8033	.7788	2.3
SharedModel*	**.7614**	**.8517**	**.7950**	**.4159**	**.6181**	**.4864**	.8442	**.9651**	**.8981**	**.8020**	**.8282**	**.7967**	
WeSamBE	.6459	.8768	.6884	.4683	.6429	.5335	.8686	.9401	.8999	.7956	.8010	.7746	2.4
WeSamBE*	**.6535**	**.8966**	**.7072**	**.4797**	**.6724**	**.5520**	.8596	**.9560**	**.9017**	**.8075**	**.8233**	**.7930**	
Spectral-360	.6666	.7349	.6977	.3605	.7113	.4553	.8187	.8898	.8519	.7545	.7568	.7208	3.1
Spectral-360*	**.7351**	**.7616**	**.7425**	.3485	**.7367**	.4491	**.8247**	**.9046**	**.8620**	**.7749**	**.7849**	**.7435**	
MBS	.8864	.6727	.6754	.4716	.5049	.4834	.8063	.7762	.7784	.7997	.7419	.7478	3.6
MBS*	**.9192**	**.6853**	**.6810**	**.5049**	**.5373**	**.5137**	.8015	**.8431**	**.8111**	**.8391**	**.7717**	**.7748**	
AMBER	.5943	.4727	.4338	.3149	.6498	.3593	.8098	.8297	.8128	.7399	.7298	.6916	5.1
AMBER*	**.6534**	**.4805**	**.4859**	**.3303**	**.7004**	**.3799**	**.8199**	**.8818**	**.8431**	**.7662**	**.7711**	**.7270**	
CwisarDH	.7421	.6659	.6986	.4442	.4511	.3753	.8476	.8786	.8581	.7910	.7094	.7232	6.6
CwisarDH*	**.8407**	**.7783**	**.7962**	**.5132**	**.4948**	**.3801**	**.8569**	**.9395**	**.8927**	**.8353**	**.7775**	**.7710**	
SC-SOBS	.5451	.7844	.5565	.3303	.6225	.3841	.7230	.8502	.7786	.6506	.7727	.6586	8.3
SC-SOBS*	**.6290**	**.8688**	**.6325**	**.3585**	**.6826**	**.4142**	**.7376**	**.9188**	**.8149**	**.7162**	**.8232**	**.7151**	
GMM	.6997	.5643	.5284	.3300	.5531	.3793	.7156	.7960	.7370	.6500	.6911	.6232	10.4
GMM*	**.7824**	**.6226**	**.6539**	**.3417**	**.6170**	**.4001**	**.7340**	**.8678**	**.7785**	**.7114**	**.7548**	**.6879**	

TABLE V: Performance comparison of the proposed framework against [44] in categories with data available for [44].

	Baseline			Camera jitter			Dynamic Background			Intermittent Object Motion			Shadows		
	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs	Pe	Re	Fs
SOBS	.9313	.9193	.9251	.6399	.8007	.7086	.5856	.8798	.6439	.5531	.7057	.5628	.7219	.8355	.7717
SOBS + [44]	.9261	.9319	.9289	.7009	.8211	.7502	.6576	.8955	.6960	.5727	.7010	.5645	**.7281**	.8736	.7907
SOBS*	**.9382**	**.9527**	**.9453**	**.7474**	**.8446**	**.7834**	**.6983**	**.9303**	**.7463**	**.7146**	**.7147**	**.6128**	.7198	**.9202**	**.8022**

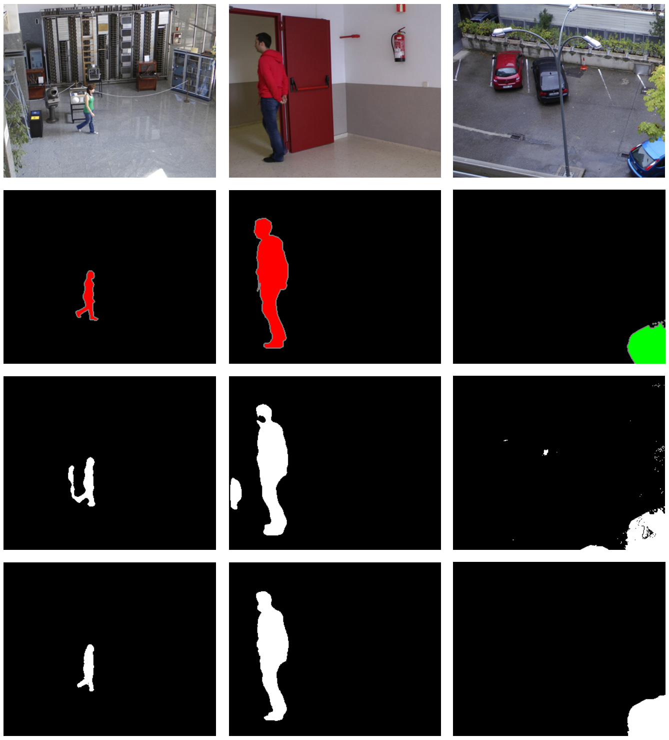


Fig. 10: Example of foreground improvements in LASIESTA dataset. For each row, from top to bottom: image, ground-truth, originally segmented foreground mask and improved foreground mask. Form left to right, example for: SuBSENSE in frame 72 of *I_BGS_02* sequence (*Bootstrap* category), LOBSTER in frame 301 of *I_CA_01* sequence (*Camouflage* category) and FuzzySOM in frame 984 of *O_RA_01* sequence (*Rainy* category).

TABLE VI: Per-category average performance in CDNET2014 achieved by the proposed combination (FusedQ) compared to the combination strategy IUTIS-5 [88].

	IUTIS-5			FusedQ		
	Pe	Re	Fs	Pe	Re	Fs
Baseline	.9464	.9680	.9567	.9197	.9793	.9484
Bad Weather	.9349	.7503	.8289	.9384	.8422	.8857
Camera Jitter	.8511	.8220	.8332	.8552	.8216	.8209
Dynamic Background	.9324	.8636	.8902	.9357	.9257	.9297
Intermittent Object Motion	.8501	.7047	.7296	.8304	.7590	.7532
Low Framerate	.7724	.8376	.7911	.7571	.8489	.7951
Night Videos	.4578	.6333	.5132	.3995	.7691	.4922
Shadows	.8766	.9492	.9084	.8545	.9651	.9039
Average	**.8277**	.8161	.8064	.8113	**.8639**	**.8161**

III-B4. Despite being a simple combination, we outperform the IUTIS-5 algorithm [88] as presented in Table VI. Note that IUTIS-5 combines the algorithms SuBSENSE, FTSG, CwisarDH, Spectral-360 and AMBER so we have used these five algorithms to average their foreground quality images.

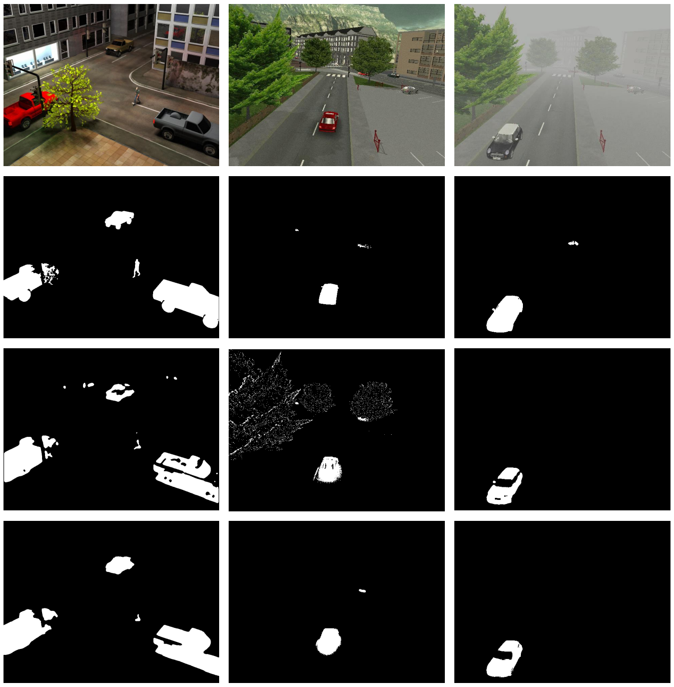


Fig. 11: Example of foreground improvements in SABS and BMC datasets. For each row, from top to bottom: image, ground-truth, originally segmented foreground mask and improved foreground mask. Form left to right, example for: LOBSTER in frame 544 of *Bootstrap* sequence (SABS), FuzzySOM in frame 484 of *511* sequence (BMC) and LOBSTER in frame 918 of *411* sequence (BMC).

F. Discussion

The results obtained in this section confirms that the proposed hierarchical fitness-to-regions strategy is effective for algorithm improvement. This capacity comes from its robustness to different challenges or distortions that typically affect background subtraction. Basically, regarding the distortions that produce false positives (e.g. dynamic backgrounds, camera jitter, shadows, illumination changes or ghost artifacts), a corresponding foreground quality image tends to produce low scores due to a low percentage of foreground pixels compared to the size of the corresponding segmented image regions in which that foreground is (i.e. low fitness). Furthermore, false negatives are typically induced by camouflages, challenge that the proposed framework overcomes using motion constraints to allow extending partially detected objects without merging with background regions. Moreover, a small image degradation like noise or compression should not substantially affect the proposed framework as modern optical flow is robust to these issues [89] and the key ingredient to keep the performance is

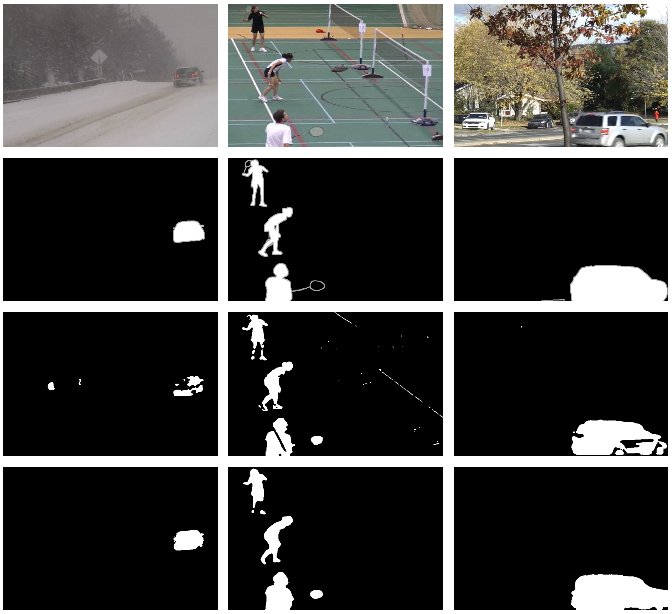


Fig. 12: Example of foreground improvements in CDNET2014 dataset. For each row, from top to bottom: image, ground-truth, originally segmented foreground mask and improved foreground mask. From left to right, example for: FTSG in frame 956 of *snowFall* sequence (*Bad Weather* category), SC-SOBS in frame 1071 of *badminton* sequence (*Camera Jitter* category) and SuBSENSE in frame 2063 of *fall* sequence (*Dynamic Background* category).

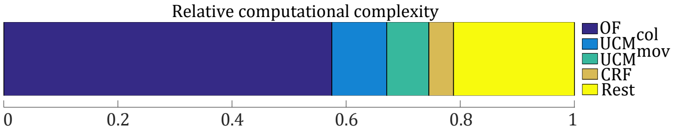


Fig. 13: Relative computational complexity for the proposed approach. From left to right: optical flow (OF), ultrametric contour maps (UCMs), CRF and the rest of the operations.

to be able to delimit objects in the motion-aware color-based UCM $\mathcal{U}$, task supported by the strong boundaries extracted from the optical flow magnitude.

However, despite the aforementioned good results, fitness-to-regions has two main limitations. Firstly, foreground objects with weak foreground qualities are removed by our approach, which means that we cannot deal with extremely uncertain cases for reconstructing an entire object from few pixels. Secondly, fitness-to-regions may lead to errors when a complete background object is almost detected as foreground (i.e. a false positive), as high qualities may be obtained.

Moreover, the computational cost of the proposed approach is mainly due to the optical flow, the UCMs and the CRF optimization that require approximately 80% of processing time (see Figure 13, where relative computational cost is presented). Our un-optimized MATLAB implementation of the proposed approach has an average running time of 0.43 fps for color images of 320×240 in a standard laptop (i7-4600U @ 2.1GHz 2.7GHz and 8GB RAM).

V. Conclusions

In this paper, we propose a framework for the improvement of foreground segmentation masks obtained by background subtraction algorithms that is independent of each algorithm characteristics. In particular, we use the foreground masks and the analyzed images to compute a foreground quality that is used to improve results through an optimization process. We obtain such foreground quality in a hierarchical manner by combining the fitness between the foreground mask and image segmentation partitions obtained at different degrees of detail that prevent foreground-background merging due to motion constraints. Experiments using fifteen algorithms and four large background subtraction datasets show that algorithms results can be improved analyzing the quality of their results. Current framework limitations are mainly related to a bad foreground probability estimation either when the original foreground segmentation is too bad for a segmented object or when complete foreground objects are not detected and, therefore, no fitness between foreground and segmented image regions can be estimated. Future work will explore the capabilities of semantic segmentation to improve foreground quality and the effects of temporal information in the energy function for foreground refinement.

Acknowledgment

This work was partially supported by the Spanish Government (HAVideo, TEC2014-53176-R).

References

[1] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “CDnet 2014: An expanded change detection benchmark dataset,” in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops (CVPRW)*, pp. 393–400, 2014.

[2] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 3061–3070, 2015.

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, “Salient object detection: A benchmark,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, 2015.

[4] M. Kristan, A. Leonardis, J. Matas, and M. Felsberg, “The visual object tracking VOT2016 challenge results,” in *Proc. Eu. Conf. Comp. Vis. Workshops*, pp. 777–823, 2016.

[5] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 724–732, 2016.

[6] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Comp. Sci. Rev.*, vol. 11-12, pp. 31–66, 2014.

[7] S. Minaee and Y. Wang, “Screen content image segmentation using robust regression and sparse decomposition,” *IEEE J. Emerg. Sel. Topic Circuits Syst*, vol. 6, no. 4, pp. 573–584, 2016.

[8] S. Minaee and Y. Wang, “Masked signal decomposition using subspace representation and its applications,” *CoRR*, vol. abs/1704.07711, 2017.

[9] Y.-H. Tsai, G. Zhong, and M.-H. Yang, “Semantic co-segmentation in videos,” in *Proc. Eu. Conf. Comp. Vis.*, pp. 760–775, 2016.

[10] D. Zhang, D. Meng, and J. Han, “Co-saliency detection via a self-paced multiple-instance learning framework,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, 2017.

[11] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the objectness of image windows,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, 2012.

[12] S. Jain, B. Xiong, and K. Grauman, “Pixel objectness,” *CoRR*, vol. abs/1701.05349, 2017.

[13] W. Wang, J. Shen, and L. Shao, “Consistent video saliency using local gradient flow optimization and global refinement,” *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, 2015.

[14] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3196–3209, 2017.

[15] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 1995–2002, 2011.

[16] D. Zhang, L. Yang, D. Meng, D. Xu, and J. Han, "SPFTN: A self-paced fine-tuning network for segmenting objects in weakly labelled videos," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 5340–5348, 2017.

[17] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *Proc. British Mach. Vis. Conf.*, 2014.

[18] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 3395–3402, 2015.

[19] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 1777–1784, 2013.

[20] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *Proc. of Eu. Conf. Comp. Vis.*, pp. 656–671, 2014.

[21] K. Maninis, S. Caelles, J. Pont-Tuset, and L. V. Gool, "Deep extreme cut: From extreme points to object segmentation," *CoRR*, vol. abs/1711.09081, 2017.

[22] M. H. Yang, C. R. Huang, W. C. Liu, S. Z. Lin, and K. T. Chuang, "Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 595–608, 2015.

[23] A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu, "Efficient background modeling through incremental support vector data description," in *Proc. Int. Conf. Pattern Recogn.*, pp. 1–4, 2008.

[24] H. H. Lin, T. L. Liu, and J. H. Chuang, "Learning a scene background model via classification," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1641–1654, 2009.

[25] D. M. Tsai and S. C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, 2009.

[26] Y. Tian, Y. Wang, Z. Hu, and T. Huang, "Selective eigenbackground for background modeling and subtraction in crowded scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 11, pp. 1849–1864, 2013.

[27] L. Maddalena and A. Petrosino, "The 3dSOBS+ algorithm for moving object detection," *Comp. Vis. Image Underst.*, vol. 122, pp. 65–73, 2014.

[28] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops*, pp. 409–413, 2014.

[29] S. Erfanian Ebadi and E. Izquierdo, "Foreground segmentation via dynamic tree-structured sparse RPCA," in *Proc. Eu. Conf. Comp. Vis.*, pp. 314–329, 2016.

[30] A. Sobral, C. Baker, T. Bouwmans, and E.-H. Zahzah, "Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction," in *Proc. Int. Conf. Image Anal. Recogn.*, pp. 94–103, 2014.

[31] T. Bouwmans, A. Sobral, S. Javed, S. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comp. Sci. Rev.*, vol. 23, pp. 1–71, 2016.

[32] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[33] F. López-Rubio and E. López-Rubio, "Features for stochastic approximation based foreground detection," *Comp. Vis. Image Underst.*, vol. 133, pp. 30–50, 2015.

[34] B. Dey and M. K. Kundu, "Enhanced macroblock features for dynamic background modeling in h.264/avc video encoded at low-bitrate," *IEEE Trans. Circuits Syst. Video Technol.*, 2016.

[35] T. Bouwmans, C. Silva, C. Marghes, M. S. Zitouni, H. Bhaskar, and C. Frélicot, "On the role and the importance of features for background modeling and foreground detection," *CoRR*, vol. abs/1611.09099, 2016.

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

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

[38] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural networks," *Multim. Tools App.*, 2017.

[39] M. Babaee, D. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recogn.*, vol. 76, pp. 635–649, 2018.

[40] L. Ang Lim and H. Yalim Keles, "Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding," *CoRR*, vol. abs/1801.02225, 2018.

[41] D. Parks and S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proc. IEEE Int. Conf. Adv. Vid. Signal Based Surv.*, pp. 192–199, 2008.

[42] 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, 2015.

[43] E. Dougherty and S. of Photo-optical Instrumentation Engineers, *An introduction to morphological image processing*. Bellingham, Washington USA : Spie Optical Engineering Press, 1992.

[44] A. Schick, M. Bauml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops*, pp. 27–31, 2012.

[45] D. Giordano, I. Kavasidis, S. Palazzo, and C. Spampinato, "Rejecting false positives in video object segmentation," in *Proc. Int. Conf. Comp. Anal. Images and Patterns*, pp. 100–112, 2015.

[46] F. Lopez-Rubio and E. López-Rubio, "Local color transformation analysis for sudden illumination change detection," *Image Vis. Comp.*, vol. 37, pp. 31–47, 2015.

[47] Z. Chen and T. Ellis, "A self-adaptive gaussian mixture model," *Comp. Vis. Image Underst.*, vol. 122, pp. 35–46, 2014.

[48] A. Sanin, C. Sanderson, and B. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods," *Pattern Recogn.*, vol. 45, pp. 1684–1695, 2012.

[49] I. Huerta, M. Holte, T. Moeslund, and J. Gonzalez, "Chromatic shadow detection and tracking for moving foreground segmentation," *Image Vis. Comp.*, vol. 41, pp. 42–53, 2015.

[50] D.-S. Pham, O. Arandjelovic, and S. Venkatesh, "Detection of dynamic background due to swaying movements from motion features," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 332–344, 2015.

[51] D. Ortego, J. C. SanMiguel, and J. M. Martínez, "Stand-alone quality estimation of background subtraction algorithms," *Comp. Vis. Image Underst.*, vol. 162, pp. 87–102, 2017.

[52] C. Cuevas, E. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA," *Comp. Vis. Image Underst.*, vol. 152, pp. 103–117, 2016.

[53] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 1937–1944, 2011.

[54] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre, "A benchmark dataset for outdoor foreground/background extraction," in *Proc. Asian Conf. Comp. Vis.*, pp. 291–300, 2013.

[55] N. Al-Najdawi, H. Bez, J. Singhai, and E. Edirisinghe, "A survey of cast shadow detection algorithms," *Pattern Recogn. Lett.*, vol. 33, no. 6, pp. 752–764, 2012.

[56] J. Ramirez-Quintana and M. Chacon-Murguia, "Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios," *Pattern Recogn.*, vol. 48, no. 4, pp. 1137–1149, 2015.

[57] L. Caro Campos, J. C. SanMiguel, and J. M. Martínez, "Discrimination of abandoned and stolen object based on active contours," in *Proc. IEEE Int. Conf. Adv. Vid. and Signal Based Surv.*, pp. 101–106, 2011.

[58] J. Kim, A. R. Rivera, B. Ryu, K. Ahn, and O. Chae, "Unattended object detection based on edge-segment distributions," in *Proc. IEEE Int. Conf. Adv. Vid. and Signal Based Surv.*, pp. 283–288, 2014.

[59] R. Raman, S. Choudhury, and S. Bakshi, "Spatiotemporal optical blob reconstruction for object detection in grayscale videos," *Multim. Tools App.*, 2017.

[60] M. Hassan, A. Malik, W. Nicolas, and I. Faye, "Adaptive foreground extraction for crowd analytics surveillance on unconstrained environments," in *Proc. Asian Conf. Comp. Vis.*, pp. 390–400, 2015.

[61] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comp. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[62] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 328–335, 2014.

[63] Y. Xiao, C. Lu, E. Tsougenis, Y. Lu, and C.-K. Tang, "Complexity-adaptive distance metric for object proposals generation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 778–786, 2015.

[64] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 2294–2301, 2009.

[65] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 1841–1848, 2013.

[66] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[67] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eu. Conf. Comp. Vis.*, pp. 25–36, 2004.

[68] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 4083–4090, 2015.

[69] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Proc. Adv. Neural Inform. Process. Systems*, pp. 109–117, 2011.

[70] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 3282–3289, 2012.

[71] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops*, pp. 401–404, 2014.

[72] H. Sajid and S.-C. Samson Cheung, "Background subtraction for static & moving camera," in *Proc. IEEE Int. Conf. Image Process.*, pp. 4530–4534, 2015.

[73] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "Universal background subtraction using word consensus models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4768–4781, 2016.

[74] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *Proc. IEEE Int. Conf. Multim. Expo*, pp. 1–6, 2015.

[75] S. Jiang and X. Lu, "WeSamBE: A weight-sample-based method for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, pp. 1–1, 2017.

[76] M. Sedky, M. Moniri, and C. C. Chibelushi, "Spectral-360: A physics-based technique for change detection," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops*, pp. 405–408, 2014.

[77] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split gaussian models," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops*, pp. 420–424, 2014.

[78] P. St-Charles and G. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *Proc. IEEE Wint. Conf. App. Comp. Vis.*, pp. 509–515, 2014.

[79] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn. Workshops*, pp. 21–26, 2012.

[80] L. Maddalena and A. Petrosino, "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection," *Neural Comp. and App.*, vol. 19, no. 2, pp. 179–186, 2010.

[81] J. Yao and J. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 1–8, 2007.

[82] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, vol. 2, pp. 246–252, 1999.

[83] D. Elgammal, A.and Harwood and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eu. Conf. Comp. Vis.*, pp. 751–767, 2000.

[84] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comp. Vis. Image Underst.*, vol. 122, pp. 4–21, 2014.

[85] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, pp. 1155–1162, 2013.

[86] Z. Liu, W. Zou, and O. L. Meur, "Saliency tree: A novel saliency detection framework," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1937–1952, 2014.

[87] 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, 2008.

[88] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 914–928, 2017.

[89] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comp. Vis.*, vol. 92, no. 1, pp. 1–31, 2011.

Diego Ortego received the Ph.D. degree in computer science and telecommunication from University Autonoma of Madrid, Spain, in 2018. Since 2012 he is a member of the Video Processing and Understanding Lab (VPU-Lab) at Universidad Autónoma of Madrid. His research interests are related to computer vision and deep learning, with a focus on segmentation, tracking and event detection tasks. He was awarded in 2013 with the AIRBUS prize to the best degree dissertation thesis in secure communications and cibersecurity (COIT and AEIT).

Juan C. SanMiguel received the Ph.D. degree in computer science and telecommunication from University Autonoma of Madrid, Madrid, Spain, in 2011. He was a Post-Doctoral Researcher with Queen Mary University of London, London, U.K., from 2013 to 2014, under a Marie Curie IAPP Fellowship. He is currently Associate Professor at University Autónoma of Madrid and Researcher with the Video Processing and Understanding Laboratory. His research interests include computer vision with a focus on online performance evaluation and multicamera activity understanding for video segmentation and tracking. He has authored over 40 journal and conference papers.

José M. Martínez received the Ph.D. degree in computer science and telecommunication from the Universidad Politécnica de Madrid, Madrid, Spain, in 1998. He is currently a Full Professor with the Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid. He has acted as an auditor and a reviewer for the EC for projects of the frameworks program for research in Information Society and Technology (IST). He is the author or coauthor of more than 100 papers in international journals and conferences and a coauthor of the first book about the MPEG-7 standard published in 2002. His professional interests cover different aspects of advanced video surveillance systems and multimedia information systems.