This CVPR2015 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Superpixel-based Video Object Segmentation using Perceptual Organization and Location Prior

Daniela Giordano Francesca Murabito Simone Palazzo Concetto Spampinato
University of Catania
Department of Electrical, Electronic and Computer Engineering
{dgiordan, palazzosim, cspampin}@dieei.unict.it, francescamurabito@gmail.com

Abstract

In this paper we present an approach for segmenting objects in videos taken in complex scenes with multiple and different targets. The method does not make any specific assumptions about the videos and relies on how objects are perceived by humans according to Gestalt laws. Initially, we rapidly generate a coarse foreground segmentation, which provides predictions about motion regions by analyzing how superpixel segmentation changes in consecutive frames. We then exploit these location priors to refine the initial segmentation by optimizing an energy function based on appearance and perceptual organization, only on regions where motion is observed. We evaluated our method on complex and challenging video sequences and it showed significant performance improvements over recent state-of-the-art methods, being also fast enough to be used for "on-the-fly" processing.

1. Introduction

Object segmentation in videos is a fundamental task for many computer vision applications, ranging from object tracking to behaviour understanding to event detection. One of the most common approaches for object segmentation is through background modeling/subtraction techniques[5], which aims at estimating a model of the scene without objects of interest; this model is then compared to each video frame in order to extract *foreground* objects. Motion analysis [15, 2], object ranking [24, 9] and clustering point tracks [17, 4] methods have also been proposed, but they impose strict assumptions on objects' appearance and motion patterns and on the target videos. Traditionally, background modeling has been addressed by density-based methods, where the distribution of each background pixel in time is modeled by a probability density function (e.g. Gaussian) [20, 18]. A recent trend in background modeling is not to provide a pixelwise model but to exploit superpixels [16, 12], thus increasing the spatial coherency and the overall quality of segmented objects. Superpixel-based modeling not only improves segmentation performance but also boosts efficiency, in terms of both required memory and computation time, since fewer models have to be kept in memory and updated at every frame.

In this paper we describe an approach for object segmentation in videos which is able to work with fast moving and multimodal backgrounds, highly deformable and/or articulated objects and with different video qualities. At the same time, it does not make any assumptions on how objects look like or move but, instead, it adopts general properties of real-world objects.

A key distinctive element of our method is the capability to quickly identify candidate motion regions, as those where significant variations on superpixel segmentation in consecutive frames have been observed. This initial background/foreground segmentation does not rely on optical flow as in [16], which is known to be computationally expensive, but is based on assessing similarity between spatio-temporal neighbor superpixels. The initial coarse foreground segmentation proposes a set of location priors, which are used as a basis for small segmentation problems (one for each motion region) solved by minimizing an energy function designed to take into account how combinations of superpixels resemble both foreground/background models and "real-world" objects, according to perceptual organization principles.

The performance evaluation carried out on three standard datasets shows that our approach 1) is able to deal with complex scenes, with several non-rigid objects undergoing sudden appearance changes, and with fast varying and multimodal backgrounds; 2) is able to generalize over different object classes since no offline training or a-priori knowledge is required; 3) outperforms existing and powerful video object segmentation approaches, e.g., [3, 16]; 4) achieves encouraging results on challenging datasets such as SegTrack [21], Underwater Dataset [19], I2R [10]; and 5) is able to work "on-the-fly".

2. Related Work

The goal of background modeling and subtraction algorithms is to build a model of the background in an offline phase and then extract objects of interest by comparing frames with the estimated model, which must be robust enough to cope with background changes, both fast and slow. The most popular background modeling methods are the *density-based* ones, where background pixel appearance is modeled by a probability density function (*pdf*; e.g. Gaussian [25]). The main shortcoming of these methods is the extremely low performance in dynamic natural scenes which, instead, involve the use of multimodal density functions as Gaussian mixture models [20]. However, methods based on mixtures of *pdf*s require effective strategies to update adaptively the components of the mixtures. Methods based on kernel density estimation [18] or not using a *pdf* such as [3], which classify background and foreground pixels according to their historical color values, have demonstrated superior performance also in complex scenes [19]. Nevertheless, exploiting only pixel colors imposes several limitations to the video object segmentation task: from the erroneous identification of luminosity changes and shadows to missing objects with colors similar to the background. Indeed, recent research [7, 11, 23] has proved that a proper combination of visual features (color, texture and/or motion) modeling temporal and spatial pixel variations improves performance sensibly. Explicitly modeling the foreground, instead of only the background, seems to enhance performance as well [18].

In the last years, a trend towards modeling spatio-temporal uniform (in terms of either appearance or motion) regions instead of single pixels has been observed [16, 12]. These works are closely related to ours, as they rely on superpixels for object segmentation in videos. The core idea is that in superpixels appearance and motion are more or less uniform, thus estimated density functions are likely to be quite accurate. However, these methods 1) need to compute the motion field through optical flow, which is computationally expensive; 2) group superpixels together according to pure spatio-temporal similarity (in terms of appearance) without exploiting real-world object features; and 3) produce segmentation through global minimization of an energy function, thus considering video object segmentation as a single-objective optimization problem, while, in fact, it is intrinsically multi-objective.

3. Our approach

The proposed method takes inspiration from [16] but extends it and outperforms it sensibly since 1) it is able to segment objects also in crowded scenes; 2) it accurately segments complex articulated objects (e.g., "girl" in the SegTrack dataset); and 3) it is fast enough to be used for on-the-fly video processing.

The basic principles which led the design of the algorithm are the following:

- *Superpixels as segmentation units*: Working with pixels is very susceptible to noise and fuzzy region boundaries, besides being in general more computationally expensive as the number of elements to analyze becomes very large.
- *Objects as superpixel aggregations*: Since superpixels typically tend to largely oversegment an image, we can assume that object boundaries always correspond to superpixel boundaries (i.e., no superpixels span over two objects). Therefore, as superpixel segmentation already guarantees a fairly robust boundary detection, we can formulate the segmentation task as the identification of connected foreground and background superpixels.
- *Motion superpixels as location priors*: Assuming that static video regions produce no *motion superpixels*, defined as superpixels on which we detect motion activity in two consecutive frames (see Sect. 3.1), we can limit our analysis to areas where motion superpixels aggregate, and process them independently as several subtasks, which is more efficient than performing a global segmentation on all superpixels and yields better results.
- *Appearance similarity*: By managing foreground and background models, we are able to know what objects look like in terms of color. Therefore, the segmentation algorithm should try and keep similar superpixels together.
- *Perceptual organization*: Objects in the real world have a generally regular and compact geometrical structure, according to the Gestalt principles of *attachment*, *similarity*, *continuity* and *symmetry*, which are believed to encode the capability of humans to capture the whole from the parts [6] without a-priori knowledge. Enforcing such principles in the way superpixels are combined together can help obtain segments which are more likely to match the actual objects in the scene.

Based on these criteria, our algorithm consists of the following steps:

Initial foreground estimation. In this phase, *motion regions*, defined as the bounding boxes (suitably expanded by $D_{\text{pad}} = 3$ pixels) around connected groups of *motion superpixels*, are identified. Unlike previous methods [16], this preliminary segmentation is carried out without computing optical flow, but, instead, analyzing superpixel segmentation changes in consecutive frames (see Fig. 1). Ideally, in two consecutive frames, superpixel segmentation

changes only in areas with moving objects. This gives a straightforward condition to rapidly identify foreground but, practically, background object movements and light changes may generate false positives that need to be removed. Each motion region is then treated as a single optimization problem for the subsequent accurate object segmentation step.

Background/foreground models estimation. Usually background subtraction approaches maintain a model for each background pixel, which is initialized in an off-line phase where only background frames are taken into account and then updated using the classification map. In our approach, we do not build a background model for each pixel; instead, we have an on-line model for each background region and each foreground object.

Accurate object segmentation. The goal of this step is to accurately identify object boundaries by grouping/removing superpixels starting from motion regions identified previously (see Fig. 1) in order to encourage spatial smoothness. To obtain an accurate object segmentation, we group superpixels by optimizing an energy function, which includes appearance similarity to the background/foreground models and perceptual organization principles. This energy minimization process is done for each detected motion region, as opposed to global minimization approaches [16] (see Fig. 3). We do not impose any constraints on motion smoothness (unlike [16, 12]) since it makes the entire process too dependent on the frame rate of the analyzed videos.

3.1. Initial Foreground Segmentation

Our approach starts with superpixel segmentation carried out by means of SLIC [1], which is an efficient adaptation of k-means in the *labxy* image space for robust superpixel generation. This step operates on pairs of consecutive frames $(t, t+1)$ and identifies motion regions based on the consideration that superpixel segmentation, in two subsequent frames, remains more or less stable in background regions, while it changes substantially in the case of moving objects.

Let S_t and S_{t+1} be the sets of the superpixels computed, respectively, at frame t and $t+1$. For each superpixel $s^i_{t+1} \in S_{t+1}$ we compute the Jaccard distances (d_J) between its backprojection at time t ($s^i_{t+1\to t}$) and all the superpixels in S_t. If the minimum of such distances is above a threshold, we mark the superpixel as "motion superpixel" (see Fig. 2). Therefore, the initial foreground mask M^{t+1} at time $t+1$ is given by:

$$M^{t+1}_{s^i_{t+1}} = \begin{cases} 1 & \text{if } \min_{s\in S_t} d_J(s^i_{t+1\to t}, s) > T \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The threshold T is adaptively computed as the average of the minimum Jaccard distance between all superpixels in frame $t+1$, which allows to handle, even in this early stage, slow object motion (T will be low, enabling the detection of fine superpixel variations) and camera motion (T will be high, and many superpixels with apparent motion will be filtered). To further remove false positives (red-colored superpixels in Fig. 1), isolated motion superpixels or small groups of connected superpixels are discarded. Moreover, as soon as the background/foreground models become reliable (after three frames; see Sect. 3.2) they are used to remove background superpixels misclassified as motion ones, by fitting a Mixture of Gaussians (MoG) for each superpixel and computing the Kullback-Leibler divergence from the background/foreground models.

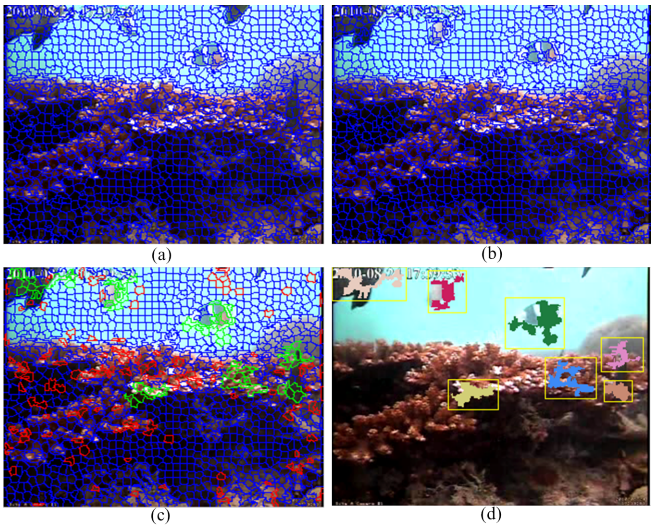

Figure 1. (a) and (b) two input frames. (c) Set of motion superpixels: false positives — filtered out at this stage — are shown in red, while correct ones are shown in green. (d) Motion regions built according to the filtered motion superpixels. In each motion region, we then perform accurate segmentation by energy minimization.

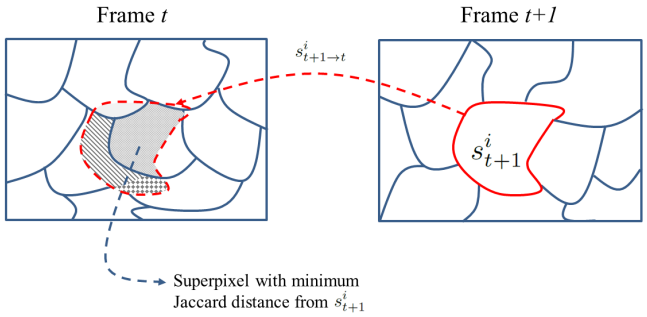

Figure 2. **Example of motion superpixel identification**. Superpixel s^i_{t+1} at frame $t+1$ is backprojected on frame t, overlapping four superpixels. s^i_{t+1} is marked as "motion superpixel" if the Jaccard distance between it and the superpixel with the highest overlap (i.e. minimum Jaccard distance) is above threshold T.

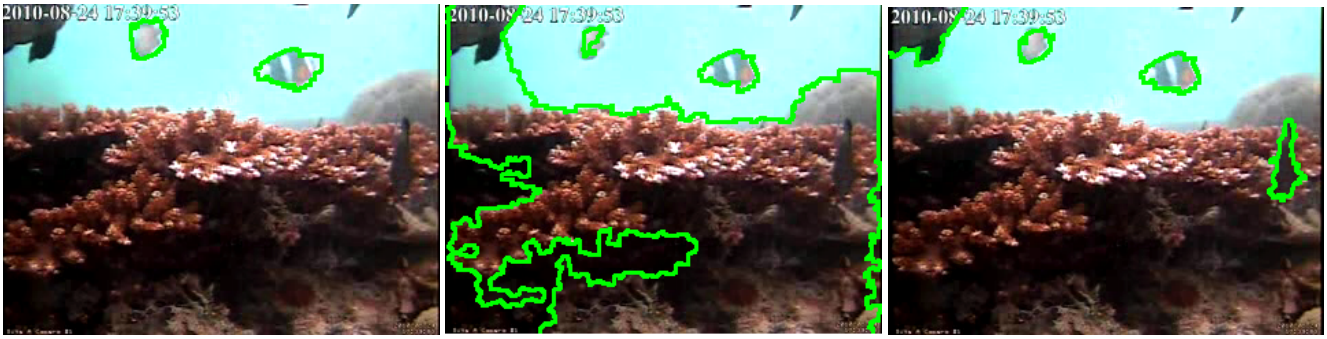

Figure 3. (a) Output mask obtained by [16] performing energy minimization of the whole image, (b) Output mask of our method when excluding location priors, thus performing energy minimization taking into account all image superpixels and (c) Output mask of our approach with location priors.

3.2. Background/foreground model estimation

In order to include constraints on the visual appearance of the objects in the scene, we maintain a set of background and foreground color models. Using several models for background and foreground, instead of only one each (as in [16]), allows to handle appearance multimodality: this is especially important when several moving objects are present in the scene, so that each can be modeled and matched independently. It is important to understand that, although in this section we will describe the construction of the models from a pixel-based point of view (and necessarily so due to the nature of color features), all other parts of the proposed method (from the identification of motion superpixels to segmentation as a minimization problem, in the next section) deal with superpixels as a basic and atomic unit.

Model initialization is performed at the first processable frame (i.e., the second video frame, when the first motion superpixel segmentation is available) by fitting a set of mixtures of Gaussians (MoGs)[1] to each *background region* and *foreground region*: background regions are obtained using adaptive k-means clustering on the whole image excluding motion superpixels and very small clusters, whereas foreground regions simply consist in connected sets of motion superpixels, ideally associated to each moving object in the scene. After model initialization is performed, we have a set of background models $\{\psi_{b,1}, \psi_{b,2}, \ldots, \psi_{b,N_b}\}$ and one of foreground models $\{\psi_{f,1}, \psi_{f,2}, \ldots, \psi_{f,N_f}\}$, where N_b is the number of clusters obtained from the adaptive k-means on the background pixels and N_f is the number of foreground regions from the initial motion superpixel segmentation.

Background model re-initialization is performed at the second processable frame (i.e., the third video frame, when the first object segmentation is available) and after every T_{init} frames, since as time passes scene conditions may change and the models may become outdated: this happens, for example, if foreground regions become stable and are absorbed into the background, or if new moving objects appear. Moreover, the need for re-initializing the background model at the second processable frame comes from the fact that the initial models are based on the inaccurate segmentation provided when using motion superpixels only, whereas at this point we can use the accurate object segmentation map for the previous frame to separate background and foreground regions.

Model update is performed at every frame (except when the model is re-initialized) after segmentation is completed. The update process for the background models at frame t consists of the following steps:

1. Initialize sets $P_{b,1} = \emptyset, \ldots, P_{b,N_b} = \emptyset$, representing the sets of pixel values (as RGB triplets) which will be used to update the corresponding background model.
2. Put into each $P_{b,i}$ all pixel values from frame $t-1$ which had been associated to background model $\psi_{b,i}$.
3. Compute background model priors $\pi_1, \ldots, \pi_{N_b}$ based on cluster sizes up to the previous frame.
4. For each pixel p belonging to superpixels labeled as background, add it to set $P_{b,i}$ according to a maximum-a-posteriori criterion, i.e. such that: $i = \arg\max_j P(\psi_{b,j}|p)$.
5. Fit a MoG ψ_{b_i} from each set $P_{b,i}$, using the current models as initial conditions for the expectation-maximization algorithm.
6. Remove models $\psi_{b,i}$ from the model set if $P_{b,i} = \emptyset$.

Using also pixels from the previous frame to fit the models (item 2) helps to prevent problems with the fitting algorithm when the initial conditions are too different from the target data set.

Foreground models are updated on a per-object basis, as follows:

1. Initialize sets $P_{f,1}, \ldots, P_{f,N_f}$, similarly as above.

[1] The number of components is adaptively set by minimizing the Akaike information criterion.

2. For each foreground object O_i segmented at frame t which contains at least one motion superpixel (see section 3.3), fit a MoG Γ_i on the object's pixels.
3. Identify the foreground model $\psi_{f,j}$ which best matches O_i using the Kullback-Leibler (KL) divergence: $j = \arg\min_k d_{\mathrm{KL}}(\psi_{f,k}, \Gamma_i)$
4. If the KL divergence between $\psi_{f,j}$ and Γ_i is smaller than a threshold T_{fg}, add O_i's pixels to $P_{f,j}$. Otherwise, create a new set P_{f,N_f+1} containing O_i's pixels, and increase N_f by 1.
5. Fit a MoG ψ_{f_i} from each set $P_{f,i}$, similarly as above.
6. Remove models $\psi_{f,i}$ from the model set if it has matched no objects for the past T_{f} frames.

3.3. Accurate object segmentation

The initial segmentation based on motion superpixels is not accurate enough, as it does not take into account any information on visual appearance or on how well a set of superpixels geometrically fit together, as shown in Fig. 1. Nevertheless, motion regions provide initial location priors for accurate segmentation based on appearance similarity and perceptual organization. These location priors are combined with the previous foreground map to allow segmenting objects which become temporarily stationary. However, in order to avoid a self-feeding effect on background regions incorrectly identified as foreground, and to let the algorithm "forget" foreground regions which are absorbed into the background, foreground models for appearance (see section 3.2) are updated only from superpixels belonging to regions which originally contained motion superpixels.

Then, for each motion region, a local segmentation subtask is defined by taking into account also non-motion superpixels intersecting the region's bounding box. Depending on the size of the object, the number of superpixels involved in each subtask is relatively small (in the order of the tens), which allows to solve the problem efficiently. If several motion regions intersect, we join them into a unique region. After that, considering each subtask independently, we pose the segmentation task as an energy minimization problem, where higher segmentation costs are due when the algorithm assigns different labels to similar contiguous superpixels or to contiguous superpixels which perceptually fit to each other. Formally, given the set of superpixels $S = \{s_1, \ldots, s_N\}$ and a set of corresponding labels $\mathcal{L} = \{l_1, \ldots, l_N\}$, where each $l_i \in \{0 : \text{background}, 1 : \text{foreground}\}$, the overall energy function is as follows:

$$E(\mathcal{L}) = A(\mathcal{L}) + P(\mathcal{L}) \tag{2}$$

$$A(\mathcal{L}) = \sum_{l_i \in \mathcal{L}} a_1(l_i) + \sum_{(l_i,l_j)\in\mathcal{N}(\mathcal{L},S)} a_2(l_i, l_j) \tag{3}$$

$$P(\mathcal{L}) = \sum_{(l_i,l_j)\in\mathcal{N}(\mathcal{L},S)} p(l_i, l_j) \tag{4}$$

where $A(\mathcal{L})$ and $P(\mathcal{L})$ respectively represent the overall appearance and perceptual organization energies, $\mathcal{N}(\mathcal{L}, S)$ is the set of all pairs of neighbor superpixels (i.e., with part of boundary in common), and the potentials $a_1(\cdot)$, $a_2(\cdot,\cdot)$ and $p(\cdot,\cdot)$ enforce our design principles on visual similarity and perceptual organization. As shown below, these potentials are defined so that $E(\mathcal{L})$ is a binary pairwise function with sub-modular pairwise potentials, thus efficiently minimizable using graph cuts in order to obtain the final segmentation:

$$\overline{\mathcal{L}} = \arg\min_{\mathcal{L}} E(\mathcal{L}) \tag{5}$$

In the following, each potential function is described in detail.

Background/foreground similarity. The unary potential $a_1(\cdot)$ indicates whether a superpixel is best associated to the foreground or the background. Given superpixel $s_i = \{p_1, \ldots, p_n\}$, let us assume we want to compute the cost of assigning label 0 (i.e., background) to s_i. For each pixel $p_j \in s_i$ and for each background model $\psi_{b,k}$, we compute the posterior probability $P(\psi_{b,k}|p_j)$. We then average these probabilities for each background model and choose the maximum among the averages as the overall background probability P_b for s_i; the negative log-posterior is then used as value for $a_1(0)$ (since we are considering the background case).

If l_i is 1 (foreground), the prior for model $\psi_{f,k}$ is $c \cdot \frac{1}{t_k + N_f}$, where t_k denotes how many frames ago the model was last updated and c is a normalization factor.

Mathematically, the overall formula can be written as:

$$a_1(l_i) = -\log \max \left\{ \frac{1}{|s_i|} \sum_{p_j \in s_i} P(\psi_{x,k}|p_j) \right\}_{k=1\ldots N_x} \tag{6}$$

where $|s_i|$ is the number of pixels in s_i and the pair $(\psi_{x,k}, N_x)$ depends on l_i:

$$(\psi_{x,k}, N_x) = \begin{cases} (\psi_{b,k}, N_b) & \text{if } l_i = 0 \\ (\psi_{f,k}, N_f) & \text{if } l_i = 1 \end{cases} \tag{7}$$

Local similarity. The binary potential $a_2(\cdot,\cdot)$ defines the cost of assigning different labels to two neighbor superpixels, based on their color similarity. Our approach on estimating this quantity is based on the following consideration: the similarity of two superpixels can be seen as the probability that their union is generated by the same color distribution, be it a background or a foreground one; if they are not similar, their union will be unlikely to be generated by any background/foreground model. Thus, given superpixels s_i and s_j, we fit a MoG Γ_{ij} from the pixels belonging to $s_i \cup s_j$, then compute the minimum KL divergence between Γ_{ij} and all background and foreground models, and use it as a dissimilarity measure between s_i and s_j; in order

to guarantee submodularity [8], the final value of potential $a_2(l_i, l_j)$ is non-zero only if $l_i \neq l_j$.

Formally, the potential function is:

$$a_2(l_i, l_j) = [l_i \neq l_j] \left[1 - \min \left\{ d_{\text{KL}}(\Gamma_{ij}, \psi) \right\}_{\psi \in \Psi}\right] \quad (8)$$

where $[l_i \neq l_j]$ is 1 if the labels are different and 0 otherwise, $d_{\text{KL}}(\cdot)$ is the KL divergence function, and $\Psi = \left\{\psi_{b,1}, \ldots, \psi_{b,N_b}, \psi_{f,1}, \ldots, \psi_{f,N_f}\right\}$ is the set of all background and foreground models. Comparing the superpixels' union to the background/foreground models helps to prevent problems when fitting the pixel distribution to a MoG, since superpixels, by construction, are small and internally homogenous. Sometimes, when pixels from both superpixels are almost identical and some color channels are practically constant, it is impossible to compute Γ_{ij}: in such cases, we set $a_2(l_i, l_j) = [l_i \neq l_j]$, which reflects the high similarity between the two superpixels. The reduced number of neighbor pairs (due to the small number of superpixels in each segmentation subtask) and the small number of pixels in each superpixel makes the evaluation of $a_2(\cdot, \cdot)$ very fast, in spite of the number of models to build.

Perceptual organization. The binary potential $p(\cdot, \cdot)$ defines the cost of assigning different labels to two neighbor superpixels, based on how well they fit together from a perceptual and geometrical point of view. To estimate this quantity, we employ a variant of the approach proposed by [6]. The potential function is computed as:

$$p(l_i, l_j) = [l_i \neq l_j] e^{-\theta \cdot [B(s_i, s_j), C(s_i, s_j)]} \quad (9)$$

where $\theta = [18, 3.5]$ is a weighing vector (suggested in [6]), $B(s_i, s_j)$ is the *boundary complexity* of region $s_i \cup s_j$, and $C(s_i, s_j)$ is the *cohesiveness* between superpixels s_i and s_j.

Boundary complexity measures the regularity of the contour obtained by joining two superpixels: intuitively, if they belong to the same object, the contour of their union should be ideally as smooth as if it were a single object in the first place; similarly, if the contour of the union is not regular, it is less likely that they belong to the same image segment. In order to numerically encode this principle, an analysis of convexity and of the number of notches (non-convex angles) on the contour is performed: as we compute boundary complexity the same way as in [6], we refer the reader to that paper for details.

Cohesiveness also measures how well two superpixels fit to each other, but is defined according to principles of *symmetry*, *continuity*, and *attachment strength*. If the superpixels' sizes are similar (i.e., their sizes' ratio is smaller than 3), it is computed as:

$$C(s_i, s_j) = \lambda_{ij}(\phi_{ij} + \varphi_{ij}) \quad (10)$$

The symmetry score, ϕ_{ij}, evaluates whether the centers of mass of s_i and s_j are aligned vertically (same x coordinates) or horizontally (same y coordinates). If we define (x_i, y_i) and (x_j, y_j) to be the centers of mass of superpixels s_i and s_j respectively, the symmetry score is computed as:

$$\phi_{ij} = \min\{|x_i - x_j|, 1\} \cdot \min\{|y_i - y_j|, 1\} \quad (11)$$

which will return a small value if the differences between either pair of corresponding coordinates is close to zero.

The continuity property indicates whether the line along which s_i and s_j's common boundary is oriented does not intersect either object at any other points. When this condition is verified, the union of the two superpixels yields an object with a perceptual impression of "continuity", in the sense that it is not evident that it is made up of two distinct regions. The corresponding score, φ_{ij}, is defined as:

$$\varphi_{ij} = \begin{cases} 0 & \text{if } e(\partial ij) \cap \partial i = \emptyset \wedge e(\partial ij) \cap \partial j = \emptyset \\ 1 & \text{otherwise} \end{cases} \quad (12)$$

where ∂i is s_i's contour, ∂j is s_j's contour, ∂ij is the common boundary, and $e(\partial ij)$ is the portion of the line passing by the extrema of the common boundary, excluding the segment between the extrema.

Attachment strength depends on how large the common boundary is with respect to the superpixels' whole boundaries:

$$\lambda_{ij} = \beta e^{-\alpha \frac{L(\partial ij)}{L(\partial i) + L(\partial j)}} \quad (13)$$

where $\beta = 3$ and $\alpha = 20$ are two parameters (again, suggested in [6]), and $L(\cdot)$ returns the length in pixels of a boundary. In other words, if two objects are "well-attached" (imagine two halves of a disk), the length of the common boundary should be large; similarly, if the attachment is weak (imagine two tangent circles), the length of common boundary will be very small in comparison to the superpixels' contour lengths.

In the particular case when one superpixel (say, s_i) is markedly larger than the other (s_j), symmetry and continuity may not be meaningful. Therefore, our cohesiveness score for these situations becomes:

$$C'(s_i, s_j) = \lambda'_{ij} = \beta e^{-\alpha \frac{L(\partial ij)}{L(\partial j)}} \quad (14)$$

that is, we only evaluate attachment strength on the smallest superpixel only.

Once all potentials in the energy function are defined, we can perform graph cut-based minimization to find the optimal segmentation. However, since each segmentation subtask is performed locally on a small set of superpixels, it may happen that the region we are analyzing is included into a large object, only a part of which was initially detected by the motion superpixels[2]. Therefore, the above approach is iteratively applied (until no changes are detected

[2]Of course, the opposite case is not a problem: if a set of connected motion superpixels span a much larger area than the actual object, the excess part will be segmented out by the energy minimization phase.

in consecutive iterations) both to capture large objects and to refine the obtained masks. At each iteration, we perform motion region–based segmentation (as above) with the difference that the object blobs detected at previous iterations are now considered as single large superpixels (hard-constrained to be labeled as foreground), thus allowing to iteratively refine object segmentation at a low processing cost, since all superpixels merged into blobs in previous iterations do not need to be processed again.

4. Experimental Results

In this section we present qualitative and quantitative results of our approach on three datasets — the Underwater dataset [19], SegTrack [21], and I2R [10] — to show how our method performs in cases of slow motion, camera motion, small objects and cluttered scenes. The parameters T_{init}, T_{fg} and T_{f} are set, respectively, to 15, 0.8 and 10 for all the employed datasets. Superpixel size was set to 7×7, as a compromise between the risk of segmentation errors, sensitivity of threshold T to noise, and processing speed. As [16] is also based on superpixel segmentation, but employs optical flow, it was used as the main baseline in all evaluations, using the public source code with default parameters.

4.1. Underwater Dataset

The underwater dataset is a collection of 14 "real-life" underwater videos (10-minute videos with spatial resolution from 320×240 to 640×480, at 5 *fps*) taken with static cameras to monitor Taiwan coral reef, and is featured by small objects and cluttered scenes. The videos are classified into seven different classes: *Blurred* (low-contrast scenes with well-separated background and foreground), *Complex Background* (background featuring complex textures, thus suitable to test superpixel-based methods), *Crowded* (highly cluttered scenes with several occlusions), *Dynamic Background* (background movements, e.g., due to plants), *Luminosity Change* (abrupt light changes), *Hybrid* (plant movements together with luminosity changes), *Camouflage Foreground Object* (e.g., objects very similar to the background). The dataset provides also ground-truth, consisting of about 20 frames per video segmented at pixel level. We compared our method to some background modeling state-of-the-art approaches [18, 22, 3, 19] and also included the well-known Gaussian Mixture Model [20] as baseline. For these methods we report their performance as stated in [19] where the original implementations (provided by the respective authors) were used, thus avoiding implementation bias in the performance analysis. The evaluations in terms of F-measure scores (computed at pixel-level) are shown in Table 1: on average, our method outperformed the other approaches in all videos, achieving good results in handling light changes, deformable objects and cluttered scenes. Fig. 3 shows a qualitative comparison between our method and [16]; it is possible to notice how our method was able to identify objects hidden in background areas (see the fish on the right side in Fig. 3) while [16] missed them.

Class	[16]	[20]	[22]	[3]	[19]	*Our method*
Blurred	35.1	83.3	70.3	85.1	93.3	89.8
Complex	36.1	67.0	83.7	74.2	81.8	86.3
Crowded	73.7	85.2	79.8	84.6	84.2	84.2
Dynamic	18.6	62.0	77.5	67.0	75.6	83.7
Hybrid	5.5	62.7	72.2	79.8	82.6	88.9
Luminosity	53.1	63.1	82.7	70.4	73.0	89.6
Camouflage	18.4	66.3	73.5	76.3	82.2	85.7
Avg	34.3	69.9	77.1	76.7	81.8	86.9
Std	23.2	9.2	4.9	6.4	6.0	2.4

Table 1. **Results on the Underwater dataset**. F-measure scores (in percentage) for different methods on the Underwater dataset. Our method is very robust to light changes and background movements (see rows 4 and 6).

	[16]	[9]	[24]	[17]	[3]	*Our method*
Birdfall	217	288	155	468	606	278
Cheetah	890	905	633	1968	11210	824
Girl	3859	1785	1488	7595	26409	1029
Monkey	284	521	365	1434	12662	192
Parachute	855	201	220	1113	40251	251

Table 2. **Results on SegTrack**. The penguin video was discarded since the annotations provided in the ground truth were not reliable as only one penguin in a group of penguins was segmented.

4.2. SegTrack Dataset

SegTrack [21], originally built for testing tracking algorithms, has been widely employed as a video object segmentation benchmark [9]. It contains six videos (*monkey-dog*, *girl*, *birdfall*, *parachute*, *cheetah*, *penguin*) and the ground truth provides pixel-level foreground object annotations for each video frame. The dataset is known for being very challenging due to camera motion, slow object motion, object-background similarity, non-rigid deformations and articulated objects. We compared our method to [9, 13, 24, 4, 17, 3] and reported their performance as stated in [16]. Table 2 shows the achieved performance as the average number of misclassified pixels per frame. Our method performed remarkably well when compared to the other methods, especially on the *girl* video where our method shows its ability to segment articulated objects. In fact, we were able to segment very well also legs and arms, which were missed by [16].

4.3. I2R Dataset

The last evaluation was carried out on the I2R Dataset [10], which contains nine videos (at 120×160 resolution) taken with static cameras showing people in different indoor and outdoor scenes. This dataset is commonly em-

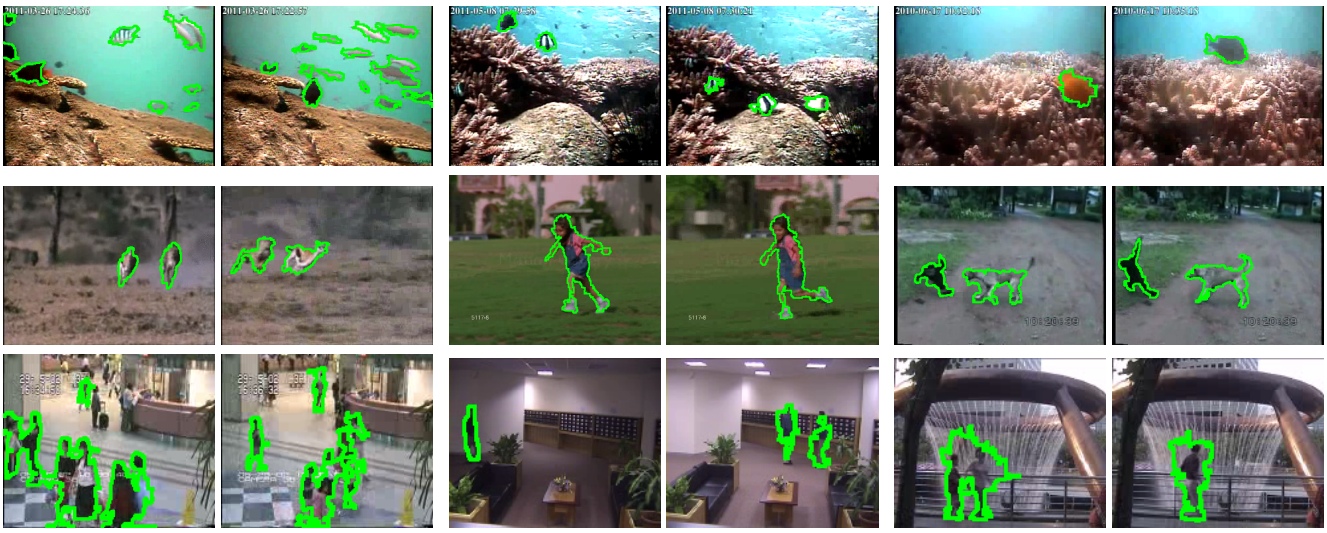

Figure 4. **Example results**. We show two images for each video with our final mask superimposed in green. (First row) **Underwater Dataset.** From left to right: Crowded, Dynamic, Luminosity. (Second row) **SegTrack.** Left to right: cheetah, girl, monkeydog. (Last row) **I2R.** Left to right: AirportHall, Lobby, Fountain. The irregularities in object boundaries are due to the superpixel-based classification and depends much on the employed superpixel segmentation approach. The effect seems to be more evident in some images because of the boundary thickness.

ployed for testing video object segmentation approaches and presents several challenges including slow motion, cluttered scenes, non-rigid deformations, articulated objects, camouflage. The ground truth consists of 20 labeled frames (at pixel-level) per video. Table 3 compares the F-measure scores of our method to the ones achieved by the recent background modeling approaches [11], [14], [19] that, similarly to our approach, model (at pixel-level) background and foreground and use combination of visual cues including texture. Our method outperformed all the other methods on the I2R dataset, especially on crowded scenes (e.g. *AirportHall*) and with articulated objects (e.g. *Escalator*). The high performance obtained on the *Escalator* class is remarkable given the presence of many occlusions.

Fig. 4 shows some example results where it is possible to appreciate the capability of our approach to adapt to different complex scenes (e.g., with very sudden light changes, see first row in Fig. 4) and targets (from highly deformable ones, e.g., fish, to articulated ones, e.g., girl) without performance loss.

We believe that including perceptual organization constraints into the method has effectively boosted its performance: as further confirmation, the overall segmentation accuracy decreased by more than 30% when excluding the $P(\mathcal{L})$ term in Eq. 2.

4.4. Processing Times

Our method takes on average 0.2 sec/frame on the Underwater and SegTrack datasets (image resolution about 320×240) and 0.05 sec/frame on the I2R dataset (image resolution of 160×120), which is fast enough to be used for "on-the-fly" video processing. This is remarkable given that [16] takes 0.5 sec/frame on the SegTrack dataset without considering optical flow and superpixel processing times, that increase [16]'s processing time to about 3 sec/frame. All processing times of our method were measured on the same machine as [16] (Intel Core i7 2.0 Ghz, 8 GB RAM) to avoid bias in the comparison. The reason of the increased speed of our approach is mainly due to modeling/classifying superpixels instead on single pixels and to local energy minimization. Of course, faster methods exist, e.g. [19] achieving 0.05 sec/frame on the Underwater dataset (although it relied on a C++ implementation, while our method is currently written in Matlab 2013a), but we believe that our method shows a good speed/accuracy trade-off.

Class	[16]	[11]	[14]	[19]	*Our Method*
AirportHall	29.6	68.0	71.3	69.2	77.4
Bootstrap	17.9	72.9	76.9	76.5	81.0
Curtain	23.2	92.4	94.1	94.9	96.3
Escalator	26.1	68.7	49.4	72.0	84.8
Fountain	15.1	85.0	86.0	83.2	84.1
ShoppingMall	13.1	79.7	83.0	78.5	86.7
Lobby	5.0	79.2	60.8	66.3	82.5
Trees	21.6	67.8	87.9	81.9	89.0
WaterSurface	83.7	83.2	92.6	92.5	93.9
Avg	26.1	77.4	78.0	79.5	86.2
Std	24.3	8.2	14.2	9.3	5.7

Table 3. **Results on I2R**. F-measure scores (in percentage) for different methods on the I2R dataset. Our method outperforms all the reported methods, especially on the *Escalator* class.

References

[1] R. Achanta, A. Shaji, and K. Smith. SLIC superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence*, 6(1):1–8, 2012.

[2] X. Bai, J. Wang, and G. Sapiro. Dynamic color flow: A motion-adaptive color model for object segmentation in video. In *ECCV 2010*, pages 617–630. 2010.

[3] O. Barnich and M. Van Droogenbroeck. ViBe: a universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, June 2011.

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV 2010*, pages 282–295. 2010.

[5] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1937–1944, June 2011.

[6] C. Cheng, A. Koschan, C.-H. Chen, D. L. Page, and M. a. Abidi. Outdoor scene image segmentation based on background recognition and perceptual organization. *IEEE Transactions on Image Processing*, 21(3):1007–19, Mar. 2012.

[7] B. Han and L. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1017–1023, 2012.

[8] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[9] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1995–2002, 2011.

[10] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Foreground object detection from videos containing complex background. *Proceedings of the eleventh ACM international conference on Multimedia MULTIMEDIA 03*, 03:2, 2003.

[11] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1301–1306, 2010.

[12] J. Lim and B. Han. Generalized background subtraction using superpixels with label integrated motion estimation. In *ECCV 2014*, pages 173–187. 2014.

[13] T. Ma and L. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 670–677, June 2012.

[14] M. Narayana, A. Hanson, and E. Learned-Miller. Background modeling using adaptive pixelwise kernel variances in a hybrid feature space. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2104–2111, 2012.

[15] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, June 2014.

[16] A. Papazoglou and V. Ferrari. Fast Object Segmentation in Unconstrained Video. *2013 IEEE International Conference on Computer Vision*, pages 1777–1784, Dec. 2013.

[17] P.Ochs and T.Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[18] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[19] C. Spampinato, S. Palazzo, and I. Kavasidis. A texton-based kernel density estimation approach for background modeling under extreme conditions. *Computer Vision and Image Understanding*, 122(0):74 – 83, 2014.

[20] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:246–252, 1999.

[21] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *Proc. BMVC*, pages 56.1–11, 2010. doi:10.5244/C.24.56.

[22] J. Yao and J.-M. Odobez. Multi-layer background subtraction based on color and texture. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[23] B. Zhang, Y. Gao, S. Zhao, and B. Zhong. Kernel similarity modeling of texture pattern flow for motion detection in complex background. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(1):29–38, 2011.

[24] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:628–635, 2013.

[25] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.