# Online Video SEEDS for Temporal Window Objectness

Michael Van den Bergh[1] Gemma Roig[1] Xavier Boix[1] Santiago Manen[1] Luc Van Gool[1,2]
[1]ETH Zürich, Switzerland [2]KU Leuven, Belgium
{vamichae,boxavier,gemmar,vangool}@vision.ee.ethz.ch *

## Abstract

*Superpixel and objectness algorithms are broadly used as a pre-processing step to generate support regions and to speed-up further computations. Recently, many algorithms have been extended to video in order to exploit the temporal consistency between frames. However, most methods are computationally too expensive for real-time applications. We introduce an online, real-time video superpixel algorithm based on the recently proposed SEEDS superpixels. A new capability is incorporated which delivers multiple diverse samples (hypotheses) of superpixels in the same image or video sequence. The multiple samples are shown to provide a strong cue to efficiently measure the objectness of image windows, and we introduce the novel concept of objectness in temporal windows. Experiments show that the video superpixels achieve comparable performance to state-of-the-art offline methods while running at 30 fps on a single 2.8 GHz i7 CPU. State-of-the-art performance on objectness is also demonstrated, yet orders of magnitude faster and extended to temporal windows in video.*

## 1. Introduction

Many algorithms use superpixels or objectness scores to efficiently select areas which to analyze further. With an increasing number of papers on the analysis of videos, the interest in having similar concepts extracted from time sequences is increasing as well. The exploitation of temporal continuity can indeed help boost several types of applications. Yet, most current solutions are computationally expensive and non-causal (*i.e.* need to see the whole video first). We propose a novel method for the online extraction of video superpixels. In terms of its still counterparts, it comes closest to the recently introduced SEEDS superpixels [15].

Similar to SEEDS, we define an objective function that prefers video superpixels to have a homogeneous color, and our video superpixels can be extracted efficiently. Their optimization is based on iteratively refining the partition, by exchanging blocks of pixels between superpixels. When starting off the partition of a new video frame, we exploit the hierarchical superpixel organization of the previous frame, the coarser levels of which serve as initialization.

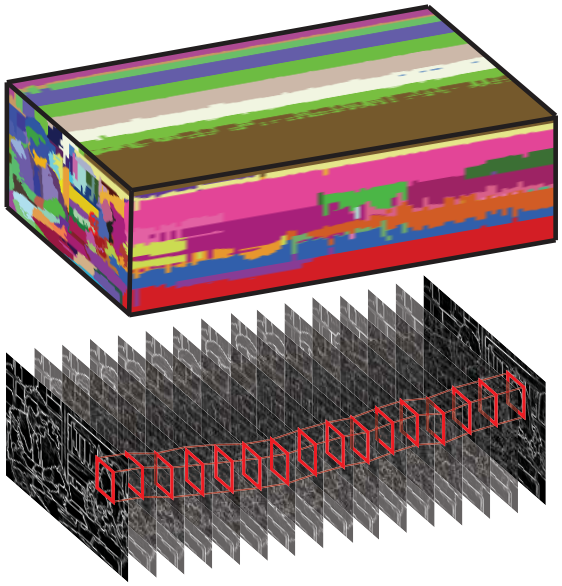


Figure 1. Top: *Video SEEDS* provide temporal superpixel tubes. Bottom: *Randomized SEEDS* efficiently produce multiple label hypotheses per frame. Based on these, a *Video Objectness* measure is introduced to propose temporal windows (tubes of bounding boxes) that are likely to contain objects.

Moreover, we propose a method to extract multiple superpixel partitions with a value of the objective function close to that of the optimum. Typically the overlapping superpixels differ in non-essential parts of their contours, but those segments that correspond to a genuine object contour are shared. This allows us to introduce a new and highly efficient objectness measure, together with its natural extension to videos (a tube of bounding boxes spanning a time interval). Fig. 1 depicts a summary of the contributions of the paper.

We experimentally validate the video superpixel and objectness algorithms, where we use standard benchmarks where possible. Both methods achieve state-of-the-art re-

*This work has been supported by the European Commission project RADHAR (FP7 ICT 248873).

sults but at much higher speeds than available methods.

## 2. Related Work

In this section, we review previous work related to superpixels and objectness in videos, the two tasks tackled in this paper.

**Video Superpixels.** Most methods are approaches for still images that have been extended to video. They either progressively add cuts or grow superpixels from centers. Adding cuts are the graph-based method [5] and its hierarchical extensions [8, 17], segmentation by weighted aggregation (SWA) [12], and normalized cuts with Nystrom optimization [7]. Methods that grow centers are based on mean shift [10, 9]. Our method also starts from a still-oriented method, *i.e.* the recently introduced SEEDS approach [15]. Thus, our approach can be seen to add a third strand to video superpixel extraction, namely one that that moves the boundaries in an initial superpixel partition.

Recently, Xu *et al.* [16, 17] proposed a benchmark to evaluate video superpixels and a framework for streaming video segmentation using the graph-based superpixel approach of [5]. They achieved state-of-the-art results, but only at 4 seconds/frame, *i.e.* 2 orders of magnitude from real-time.

**Temporal Window Objectness.** The objectness measure was introduced by Alexe *et al.* [1] for still images, whereafter [11] and [6] introduced new cues to boost performance. To the best of our knowledge, objectness throughout video shots has not been introduced before. It should not be confused with the recently introduced dynamic objectness [13], which extracts objectness within a frame by including instantaneous motion. In contrast, we deliver tubes of bounding boxes throughout extended time intervals.

## 3. Video SEEDS

In this section, we first review the SEEDS algorithm [15] for the extraction of superpixels in stills. Subsequently, we discuss the extension of this concept for videos, the corresponding energy function, and how to optimize it.

### 3.1. SEEDS for stills

Let $s$ represent the superpixel partition of an image, such that $s : \{1, \dots, N\} \rightarrow \{1, \dots, K\}$, in which $N$ represents the number of pixels in the image, and $K$ the number of superpixels. Superpixels are constrained to be contiguous blobs, which is indicated by $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of valid superpixel partitions. The SEEDS approach [15] for extracting superpixels in stills serves as starting point for our video extension. Yet, we propose important refinements on which the algorithm's efficiency critically depends.

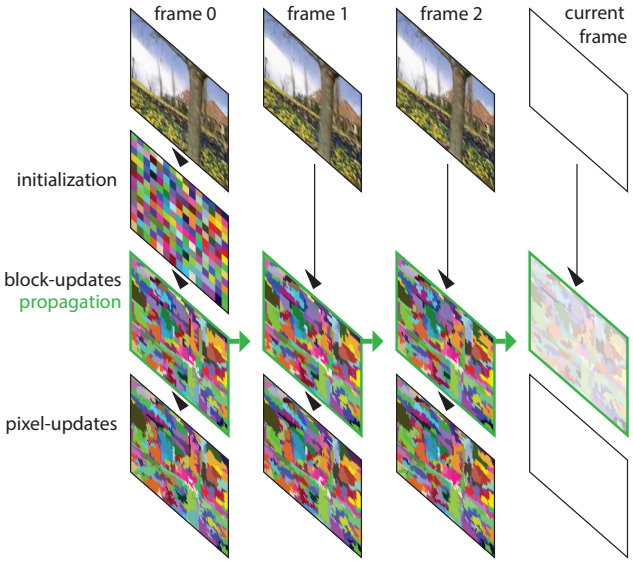



Figure 2. Overview of the Video SEEDS algorithm: The superpixel labels are propagated at an intermediary step of block-level updates. The result is fine-tuned for each frame individually.

SEEDS extracts superpixels by maximizing an objective function, thus enforcing the color histograms of superpixels to be each concentrated in a single bin. The hill climbing optimization starts from a grid of square superpixels, which it iteratively refines by swapping blocks of pixels at their boundaries. We chose SEEDS as they are extracted in real-time on a single CPU.

### 3.2. SEEDS for videos

Our video approach propagates superpixels over multiple frames to build 3D spatio-temporal constructs. As time goes on, new video superpixels can appear and others may terminate. In the literature, this is controlled by constraining the number of superpixel tubes in the sequence. For online applications this is not possible however, since the upcoming length and content of the sequence are unknown. Thus, we use alternative constraints defined through 2 parameters:

- *Superpixels per frame:* number of superpixels in which each single frame is partitioned.
- *Superpixel rate*: the rate of creating/terminating superpixels over time.

In order to fulfill both constraints, the termination of a superpixel implies the creation of a new one in the same frame. In the experiments, we discuss how we select these parameters.

Let $\mathcal{S}$ be the set of valid partitions of a video. These are the partitions for which the superpixels are contiguous blobs in all frames and that exhibit the correct superpixel-per-frame and superpixel-rate behavior. Let $\mathcal{A}_k^t$ denote the

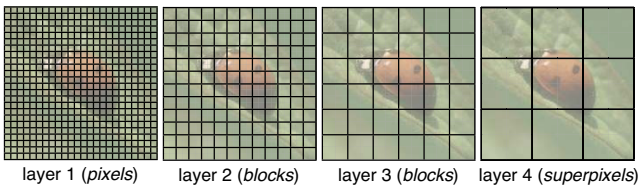



Figure 3. Hierarchy of blocks of pixels of 4 layers.

set of pixels that belong to superpixel $k$, at frame $t$. To indicate all pixels of the video superpixel up to frame $t$, we use $\mathcal{A}_k^{t:0}$.

Similarly to [15], the energy function encourages color homogeneity within the 3D superpixels. We use a color histogram of each superpixel to evaluate this. The color histogram of $\mathcal{A}_k^{t:0}$ is written as $c_{\mathcal{A}_k^{t:0}}$. Let $\mathcal{H}_j$ be a subset of the color space which determines the colors in a bin of the histogram. Then the energy function is

$$H(s) = \sum_k \sum_{\{\mathcal{H}_j\}} (c_{\mathcal{A}_k^{t:0}}(j))^2, \tag{1}$$

which is maximal when the histograms have only one non-zero bin for each video superpixel.

### 3.3. Online Optimization via Hill Climbing

The optimization algorithm is designed to maximize the energy function in an online fashion (*i.e.* only using past frames and at video rate). It computes the partition of the current frame, starting from an approximation of the last partition. Once the partition of the current frame is delivered, it remains fixed. We introduce a hill climbing algorithm that runs in real-time. It maximizes the energy by exchanging pixels between superpixels at their boundaries. This section describes the optimization in more detail. See Fig. 2 for an overview of the algorithm.

**Hierarchy of blocks of pixels.** Both the pixel exchange between superpixels and their temporal propagation are regulated through blocks of pixels. The SEEDS algorithm [15] started by dividing a still image into a regular grid of blocks. An important difference with our algorithm is that we consider a hierarchy of blocks at different sizes. Starting from pixels as the most detailed scale, $2 \times 2$ or $3 \times 3$ pixel blocks are formed (how that choice is made is to be clarified soon) for the second layer. Further layers each time combine $2 \times 2$ blocks of the previous one. The block size at the second layer ($2 \times 2$ or $3 \times 3$) and the number of layers are chosen such that the image subdivision at the highest layer approximately yields the prescribed number of superpixels per frame. In Fig. 3 we illustrate an example of the hierarchy of 4 layers of block sizes.

**Pixel and block-level updates.** An initial partition of the current frame is provided by the previous frame. This propagation process will be described shortly. In case of the first

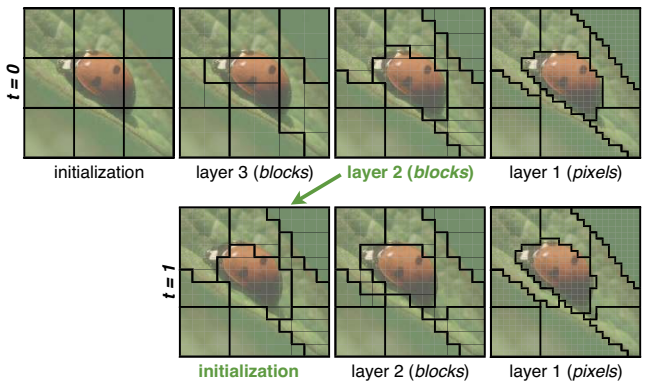



Figure 4. Efficient updating at different block sizes.

frame, the initial partition corresponds to the highest block layer as just described, *i.e.* a regular grid. The hill climbing optimization starts from the initialization to then iteratively propose local changes in the partition. Multiple pixel block exchanges between superpixels are considered, one after the other. If such an exchange increases the objective function, it is accepted and the partition is updated; else, the exchange is discarded. The exchanged pixel blocks are adjacent to the superpixel boundaries. The algorithm starts by exchanging bigger blocks, and then it descends in the block hierarchy until it reaches the pixel level. Thus, in the first iterations larger blocks are exchanged to quickly arrive at a coarse partition that captures the global structure. Later, the partition is refined through smaller blocks and pixels that capture more details. This process is shown in Fig. 4.

Let $\mathcal{B}_n^t$ be a block of pixels of the current frame that belongs to the superpixel $n$, *i.e.* $\mathcal{B}_n^t \subset \mathcal{A}_n^t \subset \mathcal{A}_n^{t:0}$. To evaluate whether exchanging the block $\mathcal{B}_n^t$ from superpixel $n$ to $m$ increases the objective function, we can use one histogram intersection computation, rather than evaluating the complete energy function. This is

$$\mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_m^{t:0}}) \geq \mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_n^{t:0} \setminus \mathcal{B}_n^t}), \tag{2}$$

in which $\mathbf{int}(\cdot,\cdot)$ denotes the intersection between two histograms, and $\setminus$ the exclusion of a set. Thus, if the intersection of $\mathcal{B}_n^t$ to the video superpixel $\mathcal{A}_m^{t:0}$ is higher than the intersection to the superpixel it currently belongs to, the exchange is accepted, otherwise it is discarded. The speed of the hill climbing optimization stems from Eq. (2), since it can evaluate a block exchange with a single intersection distance computation.

In the supplementary material we show that using Eq. (2) maximizes the energy under the assumptions that $|\mathcal{A}_m^{t:0}| \approx |\mathcal{A}_n^{t:0}|$, $|\mathcal{B}_n^t| \ll |\mathcal{A}_n^{t:0}|$, where $|\cdot|$ is the cardinality of the set. Also, it assumes that the histogram of $\mathcal{B}_n^t$ is concentrated in a single bin. The first one is that video superpixels are of similar size and that the blocks are much smaller than the video superpixels. This holds most of the time, since superpixels indeed tend to be of the same size, and the blocks are

defined to be at most one fourth of a superpixel in a frame, and hence, are much smaller than superpixels extending on multiple frames in the video. The second assumption is that the block of pixels have a homogeneous color histograms. This was empirically shown to hold in practice by [15] (in more than 90% of the cases), and we observed the same.

**Creating and terminating video superpixels.** According to the superpixel rate, some frames are selected to terminate and create superpixels. When a frame is selected, we first terminate a superpixel, and then we create a new one. To this aim, we introduce similar inequalities as in Eq. (2). They allow to evaluate which termination and creation of superpixels yield higher energy using efficient intersection distances, as well.

In Fig. 5 there is an illustration of the creation and termination of superpixels with the notation used. When a superpixel is terminated, its pixels at frame $t$ are incorporated to a neighbor superpixel. Let $\mathcal{A}_n^t \subset \mathcal{A}_n^{t:0}$ and $\mathcal{A}_m^t \subset \mathcal{A}_m^{t:0}$ be two candidates of superpixels to terminate at frame $t$. Let $\mathcal{A}_p^{t:0}$ and $\mathcal{A}_q^{t:0}$ be the superpixel candidate to incorporate $\mathcal{A}_n^t$ and $\mathcal{A}_m^t$, respectively. The superpixel with larger intersection with its neighbor is the one selected to terminate, *i.e.*

$$\mathbf{int}(c_{\mathcal{A}_n^t}, c_{\mathcal{A}_p^{t:0}}) \geq \mathbf{int}(c_{\mathcal{A}_m^t}, c_{\mathcal{A}_q^{t:0}}). \tag{3}$$

We terminate the superpixel with higher intersection to its neighbor among all superpixels in the frame. In the supplementary material, we show that Eq. (3) leads to the highest energy state, under the assumptions that $|\mathcal{A}_p^{t:0}| \approx |\mathcal{A}_q^{t:0}|$, $|\mathcal{A}_n^t| \ll |\mathcal{A}_p^{t:0}|, |\mathcal{A}_m^t| \ll |\mathcal{A}_q^{t:0}|$, and that both $\mathcal{A}_n^t$ and $\mathcal{A}_m^t$ have histograms concentrated into one bin. These are similar to the assumptions for Eq. (2). Additionally, it is also assumed that $c_{\mathcal{A}_n^{t:0}} \approx c_{\mathcal{A}_n^{(t-1):0}}$ and $c_{\mathcal{A}_m^{t:0}} \approx c_{\mathcal{A}_m^{(t-1):0}}$. This is that the color histogram of the temporal superpixel remains approximately the same including and excluding the pixels at the current frame. This holds most of the time, given the fact that $|\mathcal{A}_n^t| \ll |\mathcal{A}_n^{t:0}|$.

If a superpixel is terminated, a new one should be created to fulfill the constraint of number of superpixels per frame (Sec. 3.2). The candidates to form a new superpixel are blocks of pixels that belong to an existing video superpixel. Let $\mathcal{B}_n^t \subset \mathcal{A}_n^{t:0}$ and $\mathcal{B}_m^t \subset \mathcal{A}_m^{t:0}$ be blocks of superpixels candidates to create a new superpixel. We select the block of pixels which histogram minimally intersects with its current superpixel. This is,

$$\mathbf{int}(c_{\mathcal{B}_m^t}, c_{\mathcal{A}_m^{t:0} \setminus \mathcal{B}_m^t}) \leq \mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_n^{t:0} \setminus \mathcal{B}_n^t}). \tag{4}$$

We select the block of pixels with minimum intersection in the frame. We show in the supplementary material, that this yields the highest energy, assuming that $|\mathcal{A}_m^{t:0}| \approx |\mathcal{A}_n^{t:0}|$, $|\mathcal{B}_n^t| \ll |\mathcal{A}_n^{t:0}|, |\mathcal{B}_m^t| \ll |\mathcal{A}_m^{t:0}|$, and that both $\mathcal{B}_n^t$ and $\mathcal{B}_m^t$ have histograms concentrated into one bin. These assumptions are similar to the ones of Eq. (3).

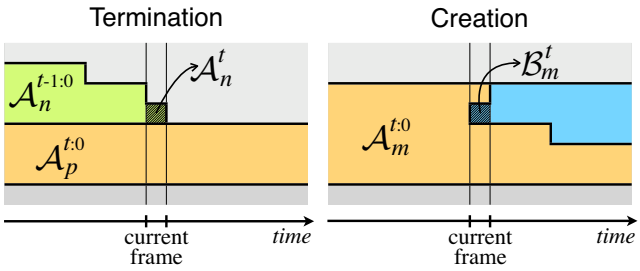


Figure 5. Termination and creation of superpixels.

**Iterations.** We can stop the optimization for a frame at any time and obtain a valid partition. We expect a higher value of the energy function if we let the hill-climbing do more iterations, until convergence. We can fix the allowed time to run per frame, or set it *on-the-fly*, depending on the application. In principle, the algorithm can run for an infinitely long video, since it generates the partition online, and in memory we only need the histograms of the video superpixels that propagate to the current frame.

**Initialization and Propagation.** In the first frame of the video, the superpixels are initialized along a grid using the hierarchy of blocks. In the subsequent frames, the block hierarchy is exploited to initialize the superpixels. Rather than re-initializing along a grid, the new frame is initialized by taking an intermediary block-level result from the previous frame (Fig. 2). Like this, the superpixel structure can be propagated from the previous frame while discarding small details. In practice, we use 4 block layers and propagate at the 2nd layer, as shown in Fig. 4.

## 4. Randomized SEEDS

Some superpixel methods offer extra capabilities, such as the extraction of a hierarchy of superpixels [17]. In this section, we introduce a new capability of superpixels that, to the best of our knowledge, has never been explored so far. In the next section we exploit it to design an objectness measure of temporal windows, though we expect that applications may not be limited to that one.

Superpixels are over-segmentations with many more regions than objects in the image. A region that is uniform in color can be over-segmented in many different correct ways, and thus, more than one partition can be valid. In Fig. 6, we give an example of different partitions with the same number of superpixels, with similar energy value and which solutions have very similar accuracy according to the superpixel benchmarks. This shows that we can extract multiple samples of superpixel partitions from the same video, all of them of comparable quality.

Since there may be a considerable amount of those partitions, we aim at extracting samples that differ as much as possible between themselves. We found a heuristic way, yet effective and fast to compute, that consists on injecting

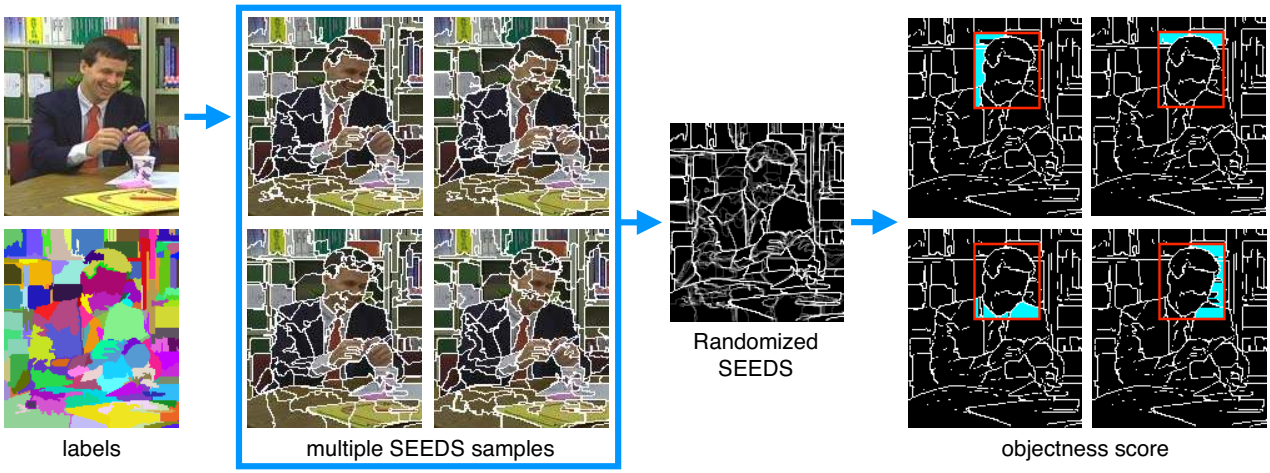



Figure 6. Different samples of randomized SEEDS segmentations of the same frame and with the same accuracy are combined. In the randomized SEEDS, we show the average of the different samples. The objectness score is computed as the sum of the distances to the common superpixel boundaries.

noise to the evaluation of the exchanges of pixels in the hill-climbing, *i.e.* in Eq. (2). This is,

$$\mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_m^{t:0}}) + a\xi \geq \mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_n^{t:0} \setminus \mathcal{B}_n^t}), \qquad (5)$$

where $\xi$ is the variable for the uniform random noise in the interval $[-1, 1]$ and $a$ is a scale factor. Note that if $a$ is small, the noise only affects the block exchanges which do not produce a large change in the energy value. In the experiments section, we analyze the effect of injecting noise by changing its scale $a$ and show that up to a certain level, the performance is not degraded compared to the sample obtained without adding noise, *i.e.* $a = 0$. This corroborates that there exists a diversity of over-segmentations with energy very close to the maximum that are equally valid.

Injecting noise may not be the only way for extracting samples, but is by far the most efficient to compute that we found. For example, changing the order in which we propose the exchanges of blocks of pixels in the hill-climbing, turned to be successful but slower in our implementation.

## 5. Video Objectness

In this section, we introduce an application of randomized SEEDS to video objectness. It is based on the observation that the coincidences among multiple superpixel partitions, reveal the true boundaries of objects. Fig. 6 shows that when superimposing a diverse set of superpixel samples obtained with randomized SEEDS, the boundaries of the objects are preserved, and the boundaries due to over-segmentation fade away. This is because the over-segmentation coincides where there are true region boundaries, and does not in regions with a similar uniform color.

In the following, we first define the measure of the objectness in a still image, and then we introduce how to extend it to temporal windows (tubes of bounding boxes).

**Objectness Measure for Still Images.** We use $O$ to represent the intersection of several superpixel samples of randomized SEEDS. $O(i)$ takes value 1 if all samples have a superpixel boundary at pixel $i$, and 0 otherwise. Thus, $O$ is an image that indicates in which pixels the samples of randomized SEEDS agree that there is a superpixel boundary.

We define the objectness score for a still image using $O$. It measures the closed boundary characteristic of objects. A bounding box is more likely to contain an object when there is a closed line in $O$ that fits tightly the bounding box. Specifically, we compute the distance from each pixel on the perimeter of the bounding box to the nearest pixel that fulfills $O(i) = 1$. Thus, in case we are in the bottom or the top of the bounding box, the distance is computed to the closest pixel in the same column, and in case we are in one of the sides, in the same row. See Fig. 6 for an illustration. Let $\mathcal{X}$ be the set of pixels inside the bounding box, Per($\mathcal{X}$) the set of pixels in the perimeter of the bounding box, and $\mathcal{X}_{\text{R,C(p)}}$ the pixels that are inside the bounding box and in the same row or column as pixel $p$. Thus, the objectness score is:

$$\frac{1}{A} \sum_{p \in \text{Per}(\mathcal{X})} \min_{\substack{i \in \mathcal{X}_{\text{R,C(p)}} \\ O(i)=1}} d(p, i), \qquad (6)$$

where $d(\cdot, \cdot)$ is the Euclidean distance, and $A$ normalizes the score using the area of the bounding box. In the supplementary material, we show that the score can be computed very efficiently using two levels of integral images, with only 8 additions, allowing for the evaluation of over 100 million bounding boxes per second. To the best of our knowledge, no earlier work has used multiple superpixel hypotheses to build an objectness score. In the experiments, we show that using multiple hypothesis has an important impact on the performance.

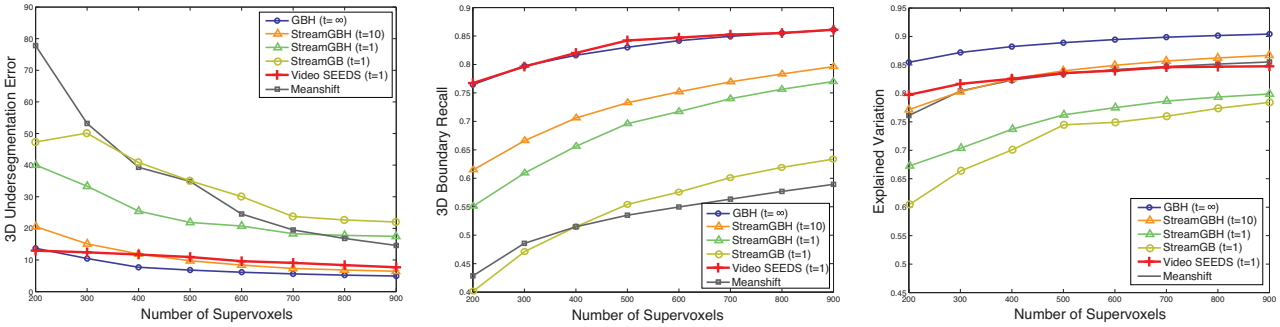



Figure 7. Comparison of our online video superpixels method to the state-of-the-art (s-o-a). For the first plot, lower is better, and for the second and third, higher is better.

**Objectness Measure for Temporal Windows.** We define a temporal window as a sequence of temporally connected bounding boxes, one per frame, which aim to surround an object in video. It can be thought as a rectangular-shaped tube in the time axis (illustrated in Fig. 1 bottom). The video is divided into overlapping shots of a predefined length, and for each shot all temporal windows are considered inside. They do not aim at replacing object tracking systems, but to assist them. The temporal windows in shots allow for incorporating features and classifiers that exploit the spatio-temporal regions, and can easily be incorporated in any video application that uses bounding boxes.

Note that there are many more temporal windows than bounding boxes in a still image. Say that in each frame there are $10^6$ possible bounding boxes. If each bounding box could move to 100 nearby positions in each subsequent frame, it leaves around $10^{50}$ possible temporal windows in a 25-frame sequence. The aim of video objectness is to reduce these $10^{50}$ temporal windows to the 100-1000 most likely to contain an object.

The video objectness score is proposed as a volumetric extension of Eq. (6) in the time dimension, normalized by the tube volume (we denoted as *3D edge score* in the experiments). In the first frame, all possible bounding boxes are extracted densely and ranked based on the objectness score for still images. In the subsequent frames, each bounding box is propagated in time by propagating the video superpixels that are completely inside the bounding box in the first frame. The score is updated online as each new frame is added until the shot is finished, and accordingly, the ranking of the temporal windows is updated online as well.

# 6. Experiments

In this section we report experimental evaluation of the introduced online video superpixel method. We also report results for the new application of video objectness. For all experiments we use a single 2.8 GHz i7 CPU. The source code of our methods will be made available online[1].

[1]http://www.vision.ee.ethz.ch/software/

## 6.1. Evaluation of Online Video SEEDS

We report results of the online video superpixels on the Chen Xiph.org benchmark [3] using the metrics proposed by [16]. The videos contain moving objects and are recorded with an uncontrolled camera. We use the standard metrics for evaluating temporal superpixels.[2] The 3D Under-segmentation Error penalizes temporal superpixels that contain more than one object, the 3D Boundary Recall is the standard recall for temporal object boundaries, and the Explained Variation is a human-independent metric that considers how well the superpixel means represent the information in the video. The benchmark evaluates these metrics varying the number of temporal superpixels. To achieve the desired amount of temporal superpixels, we select the number of superpixels per frame from a range between 200 and 600, and the superpixel rate from a range between 0 and 6. This results in a total number of video superpixels between 200 and 1086. For a detailed explanation of the metrics and these 2 parameters we refer to the supplementary material.

We compare the results of the online video SEEDS to the state-of-the-art (s-o-a) methods. We compare to the Graph-based method (GB) [5], when processing the entire videos offline, denoted as GB $t = \infty$, its streaming version with 10 frames in the stream, (StreamGB $t = 10$), and its online version (StreamGB $t = 1$). We report the Hierarchical Graph-based method (GBH) [17], also when processing the entire video offline ($t = \infty$), with 10 frames in the stream (StreamGBH $t = 10$), and the online version (StreamGBH $t = 1$). We also compare it to the streaming meanshift [9] with 10 frames in the stream. To reproduce the results we use the code and parameters provided by the authors of [16].

In Fig. 7 we show that our method obtains comparable performance to s-o-a methods, even to GBH when processing the entire videos offline. Our algorithm obtains higher performance than the online ($t = 1$) version of GBH and GB. It is also orders of magnitude faster than all previous

[2]Evaluation code and dataset available at http://www.cse.buffalo.edu/∼jcorso/r/supervoxels/

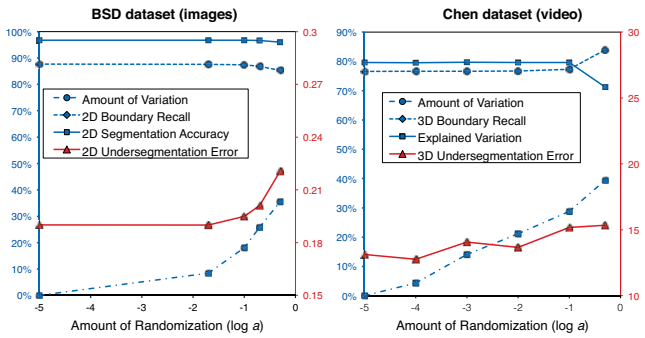



Figure 8. Evolution of superpixel metrics as a function of the amount of randomization introduced in Eq. 5.

methods, being able to run at 30 frames per second on a single CPU, in contrast to StreamGBH $t = 10$, which takes 4 seconds per image.

### 6.2. Evaluation of Randomized SEEDS

We evaluate the accuracy of the randomized superpixel samples by analyzing the effect of different levels of randomization added in Eq. (5). We use the BSD benchmark [2] for evaluating Randomized SEEDS on still images, and the Chen benchmark [3] on video. We use the standard metrics, which are 2D Undersegmentation Error, 2D Boundary Recall, and 2D Segmentation Accuracy for still images, and 3D Undersegmentation Error, 3D Boundary Recall and Explained Variation for video. For details about these metrics we refer to the supplementary material. We evaluate how the accuracy changes with different levels of randomization. We evaluate the amount of variation from a SEEDS sample without randomization. The amount of variation is computed by matching each superpixel with its closest counterpart, and summing the areas that do not overlap, normalized by the image size. The result of this experiment is shown in Fig. 8.

In both cases (still images and video), a variation between samples of about 20-30% per frame can be induced without significantly affecting the accuracy of the superpixels. Note that the amount of variation grows faster in videos because it is propagated from the first frame of the video until the end.

### 6.3. Evaluation of Video Objectness

We report results of the video objectness measure on temporal windows to showcase the advantages of randomized SEEDS on video. We also report results of objectness on still images using our objectness measure without temporal propagation. This allows for comparison to the s-o-a objectness methods in still images. In all the experiments, we use 5 samples of the randomized SEEDS, and a 4-out-of-5 criterion to define a valid intersection of the boundaries to make $O(i) = 1$, which we observed it is a good compromise between accuracy and efficiency.

**Objectness in still images.** We report results of the objectness measure on PASCAL VOC07 [4]. We use all the 4952 images of the test split, including all the objects (also the ones considered 'difficult'). We report the detection rate versus number of windows, using the PASCAL criteria of 50% overlap. We use NMS sampling to select the best 1000 bounding boxes according to the score, following the same procedure as in [1]. We use our score with the randomized SEEDS to measure the objectness in still images, without temporal propagation. In this way, we are able to compare it to s-o-a objectness measures [1, 11, 6, 14].

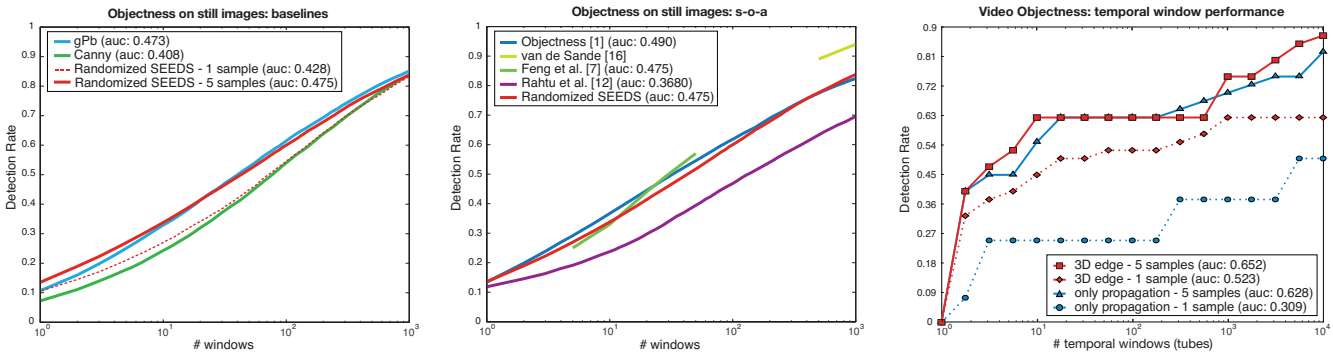
As baselines, we use the output of boundary detectors, instead of using randomized SEEDS, to compute our objectness score in still images. We use the Canny boundary detector, which is very fast to compute, and the gPb boundary detector [2], which is computationally very expensive, but is the s-o-a in boundary detection. We also show the result when only using only 1 sample of SEEDS.

The results for these baselines are shown in Fig. 9a. The objectness measure based on randomised SEEDS with 5 samples outperforms the one computed using only one sample, which emphasises the usefulness of using Randomized SEEDS. Also, it outperforms Canny edge detector in the same score. It has comparable performance to using gPb boundary detector, while being orders of magnitude faster.

In Fig. 9b there are the results compared to s-o-a objectness measures in still images. It shows that our objectness method is competitive with the s-o-a, while being an order of magnitude faster. Also note that the presented objectness measure only uses superpixels, while the others rely on additional cues (*e.g.* saliency). The presented method takes 0.39 seconds per image using a single CPU, while the s-o-a, *i.e.* [1], takes 4 seconds for a similar performance.

**Video Objectness.** We report results for our video objectness score using the Chen dataset [3] where we manually annotated object bounding boxes in the video sequences. In the video case, a stricter 50% criterion is used over the entire bounding box tube: the temporal window must overlap at least 50% with the ground truth over the entire shot of the video. In the experiments the shot length is set to 25 frames.

As these temporal objectness windows are presented as a novel concept, we compare our method to some baselines. To show the usefulness of Randomized SEEDS in video, we compare the result with using only 1 sample. Additionally, to show the usefulness of the video objectness score (noted as 3D edge in the figure), we compare with a method that uses only propagation. This baseline computes the objectness score in the first frame and then propagates the window using the video superpixels. This means the ranking of the windows is static through the entire sequence. This is equivalent to taking the method for static images followed by label propagation. In Fig. 9c we show that using the




(a) (b) (c)

Figure 9. Comparison of the objectness measure with sampling superpixels on PASCAL VOC07 to (a) baselines, (b) s-o-a, and (c) evaluation of video objectness on the Chen dataset.

Figure 10. Example of the highest ranked temporal window rendered at different frames in the video.

video objectness score (3D edge) there is an improvement in accuracy because the score is updated over time. Also, we can see that using multiple samples has a clear advantage, at more than double the performance. It is interesting to note that the 1-sample-version benefits much more from the video objectness score than the 5-sample-version. The reason why is that the video objectness score can be seen as a form of multiple samples as well: the score is the sum over 25 samples in time.

In the case of 5 samples and 1000 temporal windows, the presented method is able run at 0.17 seconds per frame: $5 \times 0.03$s for the superpixel samples, $10^{-5}$s for the score computation (0.01 in the first frame), and 0.02s for the bounding box propagation. Some example of temporal windows are shown in Fig. 10.

## 7. Conclusions

In this paper we have introduced a novel online video superpixel algorithm that is able to run in real-time, with accuracy comparable to offline methods. To achieve this, we have introduced novel concepts for temporal propagation, termination and creation of superpixels in time, using hierarchical block sizes and temporal histograms. We have demonstrated a new capability of our superpixel algorithm by efficiently extracting multiple diverse samples of superpixels. This allowed us to introduce a new, highly efficient objectness measure, together with its extension to video objectness. It enables an efficient online selection of temporal windows (tubes of bounding boxes) that contain object candidates. Finally, our experiments have shown that both the video superpixel and objectness algorithms match s-o-a offline methods in terms of accuracy, but at much higher speeds.

## References


[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 2012.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011.

[3] A. Chen and J. Corso. Propagating multi-class pixel labels throughout video frames. In *WNYIPW*, 2010.

[4] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (voc) challenge. *IJCV*, 2009.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 2004.

[6] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun. Salient object detection by composition. In *ICCV*, 2011.

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2004.

[8] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[9] S. Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. In *ECCV*, 2008.

[10] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *CVPR*, 2007.

[11] E. Rahtu, J. Kannala, and M. B. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011.

[12] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 2006.

[13] S. Stalder, H. Grabner, and L. V. Gool. Dynamic objectness for adaptive tracking. In *Asian Conference on Computer Vision*, 2012.

[14] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

[15] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. SEEDS: Superpixels extracted via energy-driven sampling. In *ECCV*, 2012.

[16] C. Xu and J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012.

[17] C. Xu, C. Xiong, and J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.