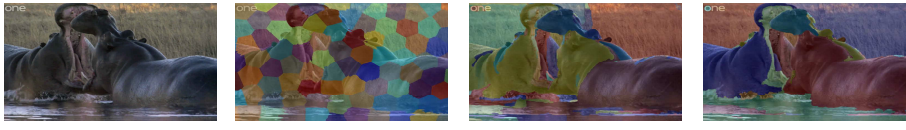
Improved Image Boundaries for Better Video Segmentation

Anna Khoreva[1] Rodrigo Benenson[1] Fabio Galasso[2] Matthias Hein[3]
Bernt Schiele[1]

[1]Max Planck Institute for Informatics, Saarbrücken, Germany
[2]OSRAM Corporate Technology, Germany
[3]Saarland University, Saarbrücken, Germany

Abstract Graph-based video segmentation methods rely on superpixels as starting point. While most previous work has focused on the construction of the graph edges and weights as well as solving the graph partitioning problem, this paper focuses on better superpixels for video segmentation. We demonstrate by a comparative analysis that superpixels extracted from boundaries perform best, and show that boundary estimation can be significantly improved via image and time domain cues. With superpixels generated from our better boundaries we observe consistent improvement for two video segmentation methods in two different datasets.



Video | TSP superpixels [7] (117 spx) | $gPb_{\mathcal{I}}$ superpixels [2] (101 spx) | Our superpixels (101 spx)

Figure 1: Graph based video segmentation relies on having high quality superpixels/voxels as starting point (graph nodes). We explore diverse techniques to improve boundary estimates, which result in better superpixels, which in turn has a significant impact on final video segmentation.

1 Introduction

Class-agnostic image and video segmentation have shown to be helpful in diverse computer vision tasks such as object detection (via object proposals) [25,32,18,19], semantic video segmentation (as pre-segmentation) [9], activity recognition (by computing features on voxels) [37], or scene understanding [21].

Both image and video segmentation have seen steady progress recently leveraging advanced machine learning techniques. A popular and successful approach consists of modelling segmentation as a graph partitioning problem [12,29,22], where the nodes represent pixels or superpixels, and the edges encode the spatio-temporal structure. Previous work focused on solving the partitioning problem [6,16,30,41], on the unary and pairwise terms of the graph [14] and on the graph construction itself [33,38,24].

The aim of this paper is to improve video segmentation by focusing on the graph nodes themselves, the video superpixels. These nodes are the starting point for unary and pairwise terms, and thus directly impact the final segmentation quality. Good superpixels for video segmentation should both be temporally consistent and give high boundary recall, and, in the case of graph-based video segmentation, for efficient runtime should enable to use a few superpixels per frame which is related to high boundary precision.

Our experiments show that existing classical superpixel/voxel methods [7,1,4] underperform for graph-based video segmentation and superpixels built from per-frame boundary estimates are more effective for the task (see §5). We show that boundary estimates can be improved when using image cues combined with object-level cues, and by merging with temporal cues. By fusing image and time domain cues, we can significantly enhance boundary estimation in video frames, improve per-frame superpixels, and thus improve video segmentation.

In particular we contribute:

- a comparative evaluation of the importance of the initial superpixels/voxels for graph-based video segmentations (§5).
- significantly improved boundary estimates (and thus per-frame superpixels) by the careful fusion of image (§6.1) and time (§6.2) domain cues.
- the integration of high-level object-related cues into the local image segmentation processing (§6.1).
- state-of-the-art video segmentation results on the VSB100 [15] and BMDS [6] datasets.

2 Related work

Video segmentation Video segmentation can be seen as a clustering problem in the 3D spatial-temporal volume. Considering superpixels/voxels as nodes, graphs are a natural way to address video segmentation and there are plenty of approaches to process the graphs. Most recent and successful techniques include hybrid generative and discriminative approaches with mixtures of trees [3], agglomerative methods constructing video segment hierarchies [16,30], techniques based on tracking/propagation of image-initialized solutions [4,7] and optimization methods based on Conditional Random Fields [8]. We leverage spectral clustering [35,28], one of the most successful approaches to video segmentation [12,29,2,24,22] and consider in our experiments the methods of [15,14].

The above approaches cover various aspects related to graph based video segmentation. Several papers have addressed the features for video segmentation [6,16,30] and some work has addressed the graph construction [33,38]. While these methods are based on superpixels none of them examines the quality of the respective superpixels for graph-based video segmentation. To the best of our knowledge, this work is the first to thoroughly analyse and advance superpixel methods in the context of video segmentation.

Superpixels/voxels We distinguish two groups of superpixel methods. The first one is the classical superpixel/voxel methods [7,1,4,26]. These methods are designed to extract superpixels of homogeneous shape and size, in order for them to have a regular topology. Having a regular superpixel topology has shown a good basis for image and video segmentation [16,31,3,33].

The second group are based on boundary estimation and focus on the image content. They extract superpixels by building a hierarchical image segmentation [2,20,10,32] and selecting one level in the hierarchy. These methods generate superpixels of heterogeneous size, that are typically fairly accurate on each frame but may jitter over time. Superpixels based on per-frame boundary estimation are employed in many state-of-the-art video segmentation methods [14,39,21,41].

In this work we argue that boundaries based superpixels are more suitable for graph-based video segmentation, and propose to improve the extracted superpixels by exploring temporal information such as optical flow and temporal smoothing.

Image boundaries After decades of research on image features and filter banks [2], most recent methods use machine learning, e.g. decision forests [10,17], mutual information [20], or convolutional neural networks [5,40]. We leverage the latest trends and further improve them, especially in relation to video data.

3 Video segmentation methods

For our experiments we consider two open source state-of-the-art graph-based video segmentation methods [15,14]. Both of them rely on superpixels extracted from hierarchical image segmentation [2], which we aim to improve.

Spectral graph reduction [14] Our first baseline is composed of three main parts.
1. *Extraction of superpixels.* Superpixels are image-based pixel groupings which are similar in terms of colour and texture, extracted by using the state-of-the-art image segmentation of [2]. These superpixels are accurate but not temporally consistent, as only extracted per frame.
2. *Feature computation.* Superpixels are compared to their (spatio-temporal) neighbours and affinities are computed between pairs of them based on appearance, motion and long term point trajectories [29], depending on the type of neighbourhood (e.g. within a frame, across frames, etc.).
3. *Graph partitioning.* Video segmentation is cast as the grouping of superpixels into video volumes. [14] employs either a spectral clustering or normalised cut formulation for incorporating a reweighing scheme to improve the performance.

In our paper we focus on the first part. We show that superpixels extracted from stronger boundary estimation help to achieve better segmentation performance without altering the underlying features or the graph partitioning method.

Segmentation propagation [15] As the second video segmentation method we consider the baseline proposed in [15]. This method does greedy matching of superpixels by propagating them over time via optical flow. This "simple" method

obtains state-of-the-art performance on VSB100. We therefore also report how superpixels extracted via hierarchical image segmentation based on our proposed boundary estimation improve this baseline.

4 Video segmentation evaluation

VSB100 We consider for learning and for evaluation the challenging video segmentation benchmark VSB100 [15] based on the HD quality video sequences of [36], containing natural scenes as well as motion pictures, with heterogeneous appearance and motion. The dataset is arranged into train (40 videos) and test (60) set. Additionally we split the training set into a training (24) and validation set (16).

The evaluation in VSB100 is mainly given by:
Precision-recall plots (BPR, VPR): VSB100 distinguishes a boundary precision-recall metric (BPR), measuring the per-frame boundary alignment between a video segmentation solution and the human annotations, and a volume precision-recall metric (VPR), reflecting the temporal consistency of the video segmentation result.
Aggregate performance measures (AP, ODS, OSS): for both BPR and VPR, VSB100 reports average precision (AP), the area under the precision-recall curves, and two F-measures where one is measured at an optimal dataset scale (ODS) and the other at an optimal segmentation scale (OSS) (where "optimal" stands for oracle provided).

BMDS To show the generalization of the proposed method we further consider the Berkeley Motion Segmentation Dataset (BMDS) [6], which consists of 26 VGA-quality videos, representing mainly humans and cars. Following prior work [23] we use 10 videos for training and 16 as a test set, and restrict all video sequences to the first 30 frames.

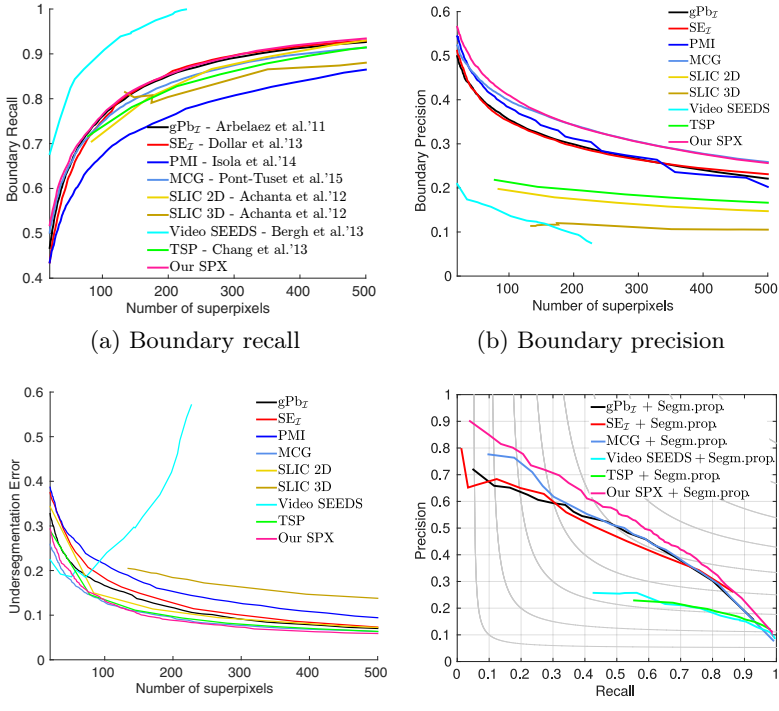
5 Superpixels and supervoxels

Graph-based video segmentation methods rely on superpixels to compute features and affinities. Employing superpixels as pre-processing stage for video segmentation provides a desirable computational reduction and a powerful per-frame representation.

Ideally these superpixels have high boundary recall (since one cannot recover from missing recall), good temporal consistency (to make matching across time easier), and are as few as possible (in order to reduce the chances of segmentation errors; to accelerate overall computation and reduce memory needs).

In this section we explore which type of superpixels are most suitable for graph-based video segmentation.

Superpixel/voxel methods Many superpixel/voxel methods have been explored in the past. We consider the most promising ones in the experiments of Figure 2. SLIC 2D/3D [1] is a classic method to obtain superpixels via iterative clustering (in space and space-time domain). TSP [7] extends SLIC to explicitly



(a) Boundary recall (b) Boundary precision

(c) Under-segmentation error (d) BPR of video segmentation

Figure 2: Comparison of different superpixel/voxel methods, and their use for video segmentation. VSB100 validation set. SPX: superpixels. Segm. prop.: segmentation propagation [15] (see §3).

model temporal dynamics. Video SEEDS [4] is similar to SLIC 3D, but uses an alternative optimization strategy. Other than classic superpixel/voxel methods we also consider superpixels generated from per-frame hierarchical segmentation based on boundary detection (ultrametric contour maps [2]). We include gPb$_\mathcal{I}$ [2], SE$_\mathcal{I}$ [10], PMI [20] and MCG [32] as sources of boundary estimates.

Superpixel evaluation We compare superpixels by evaluating the recall and precision of boundaries and the under-segmentation error [27] as functions of the average number of superpixels per frame. We also use some of them directly for video segmentation (Figure 2d). We evaluate (use) all methods on a frame by frame basis; supervoxel methods are expected to provide more temporally consistent segmentations than superpixel methods.

Results Boundary recall (Figure 2a) is comparable for most methods. Video SEEDS is an outlier, showing very high recall, but low boundary precision (2b) and high under-segmentation error (2c). gPb$_\mathcal{I}$ and SE$_\mathcal{I}$ reach the highest boundary recall with fewer superpixels. Per-frame boundaries based superpixels perform better than classical superpixel methods on boundary precision (2b). From these figures one can see the conflicting goals of having high boundary recall,

high precision, and few superpixels.
We additionally evaluate the superpixel methods using a region-based metric: under-segmentation error [27]. Similar to the boundary results, the curves are clustered in two groups: TSP-like and $\text{gPb}_{\mathcal{I}}$-like quality methods, where the latter underperform due to the heterogeneous shape and size of superpixels (2c).

Figure 2d shows the impact of superpixels for video segmentation using the baseline method [15]. We pick TSP as a representative superpixel method (fair quality on all metrics), Video SEEDS as an interesting case (good boundary recall, bad precision), $\text{SE}_{\mathcal{I}}$ and MCG as good boundary estimation methods, and the baseline $\text{gPb}_{\mathcal{I}}$ (used in [15]). Albeit classical superpixel methods have lower under-segmentation error than boundaries based superpixels, when applied for video segmentation the former underperform (both on boundary and volume metrics), as seen in Figure 2d. Boundary quality measures seem to be a good proxy to predict the quality of superpixels for video segmentation. Both in boundary precision and recall metrics having stronger initial superpixels leads to better results.

Intuition Figure 1 shows a visual comparison of TSP superpixels versus $\text{gPb}_{\mathcal{I}}$ superpixels (both generated with a similar number of superpixels). By design, most classical superpixel methods have a tendency to generate superpixels of comparable size. When requested to generate fewer superpixels, they need to trade-off quality versus regular size. Methods based on hierarchical segmentation (such as $\text{gPb}_{\mathcal{I}}$) generate superpixels of heterogeneous sizes and more likely to form semantic regions. For a comparable number of superpixels techniques based on image segmentation have more freedom to provide better superpixels for graph-based video segmentation than classical superpixel methods.

Conclusion Based both on quality metrics and on their direct usage for graph-based video segmentation, boundary based superpixels extracted via hierarchical segmentation are more effective than the classical superpixel methods in the context of video segmentation. The hierarchical segmentation is fully defined by the estimated boundary probability, thus better boundaries lead to better superpixels, which in turn has a significant impact on final video segmentation. In the next sections we discuss how to improve boundary estimation for video.

6 Improving image boundaries

To improve the boundary based superpixels fed into video segmentation we seek to make best use of the information available on the videos. We first improve boundary estimates using each image frame separately (§6.1) and then consider the temporal dimension (§6.2).

6.1 Image domain cues

A classic boundary estimation method (often used in video segmentation) is $\text{gPb}_{\mathcal{I}}$ [2] ($\mathcal{I}$: image domain), we use it as a reference point for boundary quality

metrics. In our approach we propose to use $\text{SE}_{\mathcal{I}}$ ("structured edges") [10]. We also considered the convnet based boundary detector [40]. However employing boundaries of [40] to close the contours and construct per-frame hierarchical segmentation results in the performance similar to $\text{SE}_{\mathcal{I}}$ and significantly longer training time. Therefore in our system we employ $\text{SE}_{\mathcal{I}}$ due to its speed and good quality.

Object proposals Methods such as $\text{gPb}_{\mathcal{I}}$ and $\text{SE}_{\mathcal{I}}$ use bottom-up information even though boundaries annotated by humans in benchmarks such as BSDS500 or VSB100 often follow object boundaries. In other words, an oracle having access to ground truth semantic object boundaries should allow to improve boundary estimation (in particular on the low recall region of the BPR curves). Based on this intuition we consider using segment-level object proposal (OP) methods to improve initial boundary estimates ($\text{SE}_{\mathcal{I}}$). Object proposal methods [25,32,18,19] aim at generating a set of candidate segments likely to have high overlap with true objects. Typically such methods reach $\sim 80\%$ object recall with 10^3 proposals per image.

Based on initial experiments we found that the following simple approach obtains good boundary estimation results in practice. Given a set of object proposal segments generated from an initial boundary estimate, we average the contours of each segment. Pixels that are boundaries to many object proposals will have high probability of boundary; pixels rarely members of a proposal boundary will have low probability. With this approach, the better the object proposals, the closer we are to the mentioned oracle case.

We evaluated multiple proposals methods [25,32,18] and found RIGOR [18] to be most effective for this use (§6.1). To the best of our knowledge this is the first time an object proposal method is used to improve boundary estimation. We name the resulting boundary map $\text{OP}\,(\text{SE}_{\mathcal{I}})$.

Globalized probability of boundary A key ingredient of the classic $\text{gPb}_{\mathcal{I}}$ [2] method consists on "globalizing boundaries". The most salient boundaries are highlighted by computing a weighted sum of the spatial derivatives of the first few eigenvectors of an affinity matrix built based on an input probability of boundary. The affinity matrix can be built either at the pixel or superpixel level. The resulting boundaries are named "spectral" probability of boundary, $\text{sPb}\,(\cdot)$. We employ the fast implementation from [32].

Albeit well known, such a globalization step is not considered by the latest work on boundary estimation (e.g. [10,5]). Since we compute boundaries at a single-scale, $\text{sPb}\,(\text{SE}_{\mathcal{I}})$ is comparable to the SCG results in [32].

Re-training Methods such as $\text{SE}_{\mathcal{I}}$ are trained and tuned for the BSDS500 image segmentation dataset [2]. Given that VSB100 [15] is larger and arguably more relevant to the video segmentation task than BSDS500, we retrain $\text{SE}_{\mathcal{I}}$ (and RIGOR) for this task. In the following sections we report results of our system trained over BSDS500, or with VSB100. We will also consider using input data other than an RGB image (§6.2).

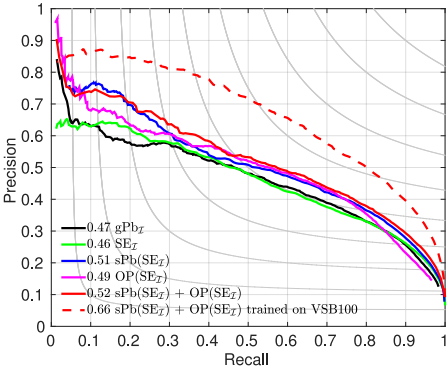


Figure 3: Progress when integrating various image domain cues (§6.1) in terms of BPR on VSB100 validation set.

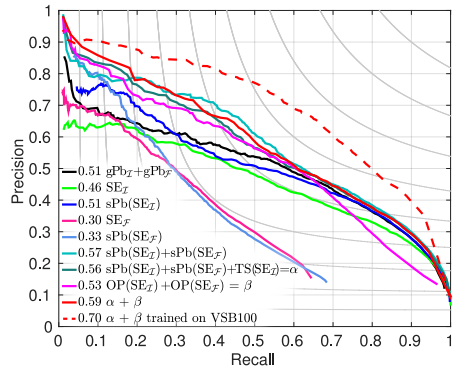


Figure 4: Progress when integrating image and time domain cues (§6.2) in terms of BPR on VSB100 validation set.

Merging cues After obtaining complementary probabilities of boundary maps (e.g. OP (SE$_\mathcal{I}$), sPb (SE$_\mathcal{I}$), etc.), we want to combine them effectively. Naive averaging is inadequate because boundaries estimated by different methods do not have pixel-perfect alignment amongst each other. Pixel-wise averaging or maxing leads to undesirable double edges (negatively affecting boundary precision).

To solve this issue we use the grouping technique from [32] which proposes to first convert the boundary estimate into a hierarchical segmentation, and then to align the segments from different methods. Note that we do not use the multi-scale part of [32]. Unless otherwise specified all cues are averaged with equal weight. We use the sign "+" to indicate such merges.

Boundary results when using image domain cues Figure 3 reports results when using the different image domain cues, evaluated over the VSB100 validation set. The gPb$_\mathcal{I}$ baseline obtains 47% AP, while SE$_\mathcal{I}$ (trained on BSDS500) obtains 46%. Interestingly, boundaries based on object proposals OP (SE$_\mathcal{I}$) from RIGOR obtain a competitive 49%, and, as expected, provide most gain in the high precision region of BPR. Globalization sPb (SE$_\mathcal{I}$) improves results to 51% providing a homogeneous gain across the full recall range. Combining sPb (SE$_\mathcal{I}$) and OP (SE$_\mathcal{I}$) obtains 52%. After retraining SE$_\mathcal{I}$ on VSB100 we obtain our best result of 66% AP (note that all cues are affected by re-training SE$_\mathcal{I}$).

Conclusion Even when using only image domain cues, large gains can be obtained over the standard gPb$_\mathcal{I}$ baseline.

6.2 Temporal cues

The results of §6.1 ignore the fact that we are processing a video sequence. In the next sections we describe two different strategies to exploit the temporal dimension.

Optical flow We propose to improve boundaries for video by employing optical flow cues. We use the state-of-the-art EpicFlow [34] algorithm, which we feed with our $\mathrm{SE}_{\mathcal{I}}$ boundary estimates.

Since optical flow is expected to be smooth across time, if boundaries are influenced by flow, they will become more temporally consistent. Our strategy consists of computing boundaries directly over the forward and backward flow map, by applying SE over the optical flow magnitude (similar to one of the cues used in [11]). We name the resulting boundaries map $\mathrm{SE}_{\mathcal{F}}$ ($\mathcal{F}$: optical flow). Although the flow magnitude disregards the orientation information from the flow map, in practice discontinuities in magnitude are related to changes in flow direction.

We then treat $\mathrm{SE}_{\mathcal{F}}$ similarly to $\mathrm{SE}_{\mathcal{I}}$ and compute $\mathrm{OP}(\mathrm{SE}_{\mathcal{F}})$ and $\mathrm{sPb}(\mathrm{SE}_{\mathcal{F}})$ over it. All these cues are finally merged using the method described in §6.1.

Time smoothing The goal of our new boundaries based superpixels is not only high recall, but also good temporal consistency across frames. A naive way to improve temporal smoothness of boundaries consists of averaging boundary maps of different frames over a sliding window; differences across frames would be smoothed out, but at the same time double edge artefacts (due to motion) would appear (reduced precision).

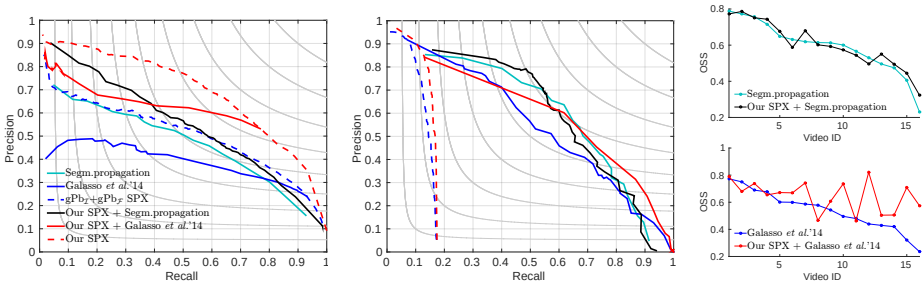
We propose to improve temporal consistency by doing a sliding window average across boundary maps of several adjacent frames. For each frame t, instead of naively transferring boundary estimates from one frame to the next, we warp frames $t_{\pm i}$ using optical flow with respect to frame t; thus reducing double edge artefacts. For each frame t we treat warped boundaries from frames $t_{\pm i}$ as additional cues, and merge them using the same mechanism as in §6.1. This merging mechanism is suitable to further reduce the double edges issue.

Boundary results when using temporal cues The curves of Figure 4 show the improvement gained from optical flow and temporal smoothing.

Optical flow Figure 4 shows that on its own flow boundaries are rather weak ($\mathrm{SE}_{\mathcal{F}}$, $\mathrm{sPb}(\mathrm{SE}_{\mathcal{F}})$), but they are quite complementary to image domain cues ($\mathrm{sPb}(\mathrm{SE}_{\mathcal{I}})$ versus $\mathrm{sPb}(\mathrm{SE}_{\mathcal{I}})$+$\mathrm{sPb}(\mathrm{SE}_{\mathcal{F}})$).

Temporal smoothing Using temporal smoothing ($\mathrm{sPb}(\mathrm{SE}_{\mathcal{I}})$+$\mathrm{sPb}(\mathrm{SE}_{\mathcal{F}})$+ $\mathrm{TS}(\mathrm{SE}_{\mathcal{I}})=\alpha$) leads to a minor drop in boundary precision, in comparison with $\mathrm{sPb}(\mathrm{SE}_{\mathcal{I}})$+$\mathrm{sPb}(\mathrm{SE}_{\mathcal{F}})$ in Figure 4. It should be noted that there is an inherent tension between improving temporal smoothness of the boundaries and having better accuracy on a frame by frame basis. Thus we aim for the smallest negative impact on BPR. In our preliminary experiments the key for temporal smoothing was to use the right merging strategy (§6.1). We expect temporal smoothing to improve temporal consistency.

Object proposals Adding $\mathrm{OP}(\mathrm{SE}_{\mathcal{F}})$ over $\mathrm{OP}(\mathrm{SE}_{\mathcal{I}})$ also improves BPR (see $\mathrm{OP}(\mathrm{SE}_{\mathcal{F}})$ +$\mathrm{OP}(\mathrm{SE}_{\mathcal{I}})=\beta$ in Figure 4), particularly in the high-precision area. Merging it with other cues helps to push BPR for our final frame-by-frame result.



(a) BPR on validation set (b) VPR on validation set (c) OSS per video

Figure 5: VSB100 validation set results of different video segmentation methods. Dashed lines indicate only frame-by-frame processing (see §7.1 for details).

Combination and re-training Combining all cues together improves the BPR metric with respect to only using appearance cues, we reach 59% AP versus 52% with appearance only (see §6.1). This results are better than the $\text{gPb}_{\mathcal{I}}$+$\text{gPb}_{\mathcal{F}}$ baseline (51% AP, used in [14]).
Similar to the appearance-only case, re-training over VSB100 gives an important boost (70% AP). In this case not only $\text{SE}_{\mathcal{I}}$ is re-trained but also $\text{SE}_{\mathcal{F}}$ (over EpicFlow).

Figure 2 compares superpixels extracted from the proposed method (α+β model without re-training for fair comparison) with other methods. Our method reaches top results on both boundary precision and recall. Unless otherwise specified, all following "Our SPX" results correspond to superpixels generated from the hierarchical image segmentation [2] based on the proposed boundary estimation α+β re-trained on VSB100.

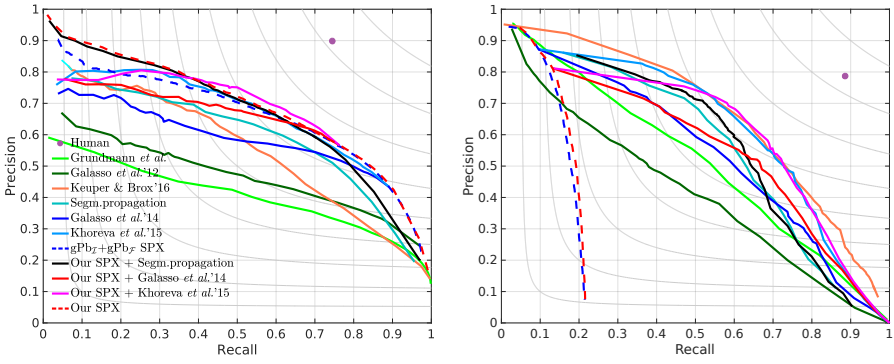
Conclusion Temporal cues are effective at improving the boundary detection for video sequences. Because we use multiple ingredients based on machine learning, training on VSB100 significantly improves quality of boundary estimates on a per-frame basis (BPR).

7 Video segmentation results

In this section we show results for the state-of-the-art video segmentation methods [15,14] with superpixels extracted from the proposed boundary estimation. So far we have only evaluated boundaries of frame-by-frame hierarchical segmentation. For all further experiments we will use the best performing model trained on VSB100, which uses image domain and temporal cues, proposed in §6 (we refer to (α + β) model, see Figure 4). Superpixels extracted from our boundaries help to improve video segmentation and generalizes across different datasets.

7.1 Validation set results

We use two baseline methods ([14,15], see §3) to show the advantage of using the proposed superpixels, although our approach is directly applicable to any graph-



(a) BPR on test set (b) VPR on test set

Figure 6: Comparison of state-of-the-art video segmentation algorithms with/without our improved superpixels, on the test set of VSB100 [15]. Dashed lines indicate only frame-by-frame processing. See table 1 and §7.2 for details.

	BPR			VPR			Length	NCL
Algorithm	ODS	OSS	AP	ODS	OSS	AP	$\mu\,(\delta)$	μ
Human	0.81	0.81	0.67	0.83	0.83	0.70	83.2(40.0)	11.9
Grundmann et al. [16]	0.47	0.54	0.41	0.52	0.55	0.52	87.7(34.0)	18.8
Galasso et al.'12 [13]	0.51	0.56	0.45	0.45	0.51	0.42	80.2(37.6)	8.0
Yi and Pavlovic [41]	0.63	0.67	0.60	**0.64**	**0.67**	0.65	35.83(38.9)	167.3
Keuper and Brox [22]	0.56	0.63	0.56	**0.64**	0.66	**0.67**	1.1(0.7)	962.6
Segm. propagation [15]	0.61	0.65	0.59	0.59	0.62	0.56	25.5(36.5)	258.1
Our SPX + [15]	**0.64**	**0.69**	**0.67**	**0.61**	**0.63**	**0.57**	22.2(34.4)	216.8
Spectral graph reduction[14]	0.62	0.66	**0.54**	0.55	0.59	**0.55**	61.3(40.9)	80.0
Our SPX + [14]	**0.66**	**0.68**	0.51	**0.58**	**0.61**	**0.55**	70.4(40.2)	15.0
Graph construction [24]	0.64	**0.70**	**0.61**	0.63	0.66	**0.63**	83.4(35.3)	50.0
Our SPX +[24]	**0.66**	**0.70**	0.55	**0.64**	**0.67**	0.61	79.4(35.6)	50.0

Table 1: Comparison of state-of-the-art video segmentation algorithms with our proposed method based on the improved superpixels, on the test set of VSB100 [15]. The table shows BPR and VPR and length statistics (mean μ, standard deviation δ, no. clusters NCL), see figure 6 and §7.2 for details.

based video segmentation technique. The baseline methods originally employ the superpixels proposed by [2,13], which use the boundary estimation gPb$_\mathcal{I}$+gPb$_\mathcal{F}$ to construct a segmentation.

For the baseline method of [14] we build a graph, where superpixels generated from the hierarchical image segmentation based on the proposed boundary estimation are taken as nodes. Following [14] we select the hierarchy level of image segmentation to extract superpixels (threshold over the ultrametric contour map) by a grid search on the validation set. We aim for the level which gives the best video segmentation performance, optimizing for both BPR and VPR.

Figure 5 presents results on the validation set of VSB100. The dashed curves indicate frame-by-frame segmentation and show (when touching the continuous

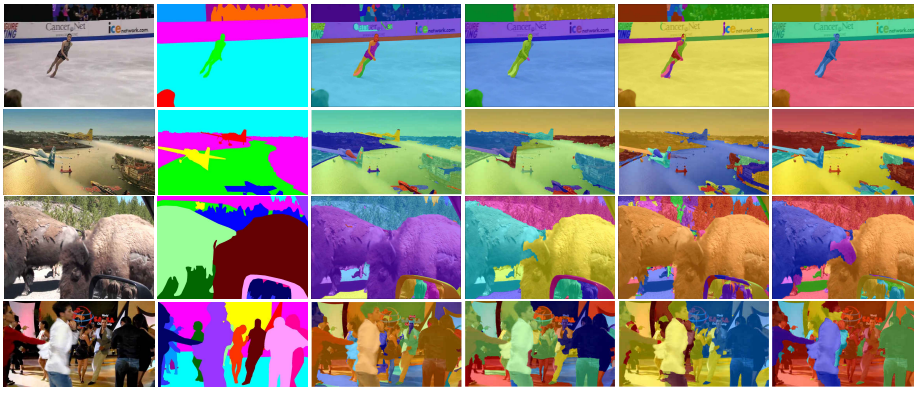


Figure 7: Comparison of video segmentation results of [15,14] with our proposed superpixels to one human ground truth. The last row shows a failure case for all methods.

curves) the chosen level of hierarchy to extract superpixels. As it appears in the plots, our superpixels help to improve video segmentation performance on BPR and VPR for both baseline methods [15,14]. Figure 5c shows the performance of video segmentation with the proposed superpixels per video sequence. Our method improves most on hard cases, where the performance of the original approach was quite low, OSS less than 0.5.

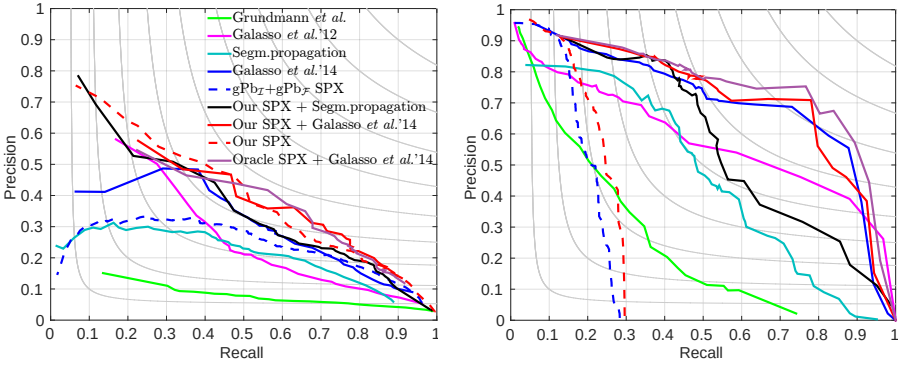
7.2 Test set results

VSB100 Figure 6 and Table 1 show the comparison of the baseline methods [15,14] with and without superpixels generated from the proposed boundaries, and with state-of-the-art video segmentation algorithms on the test set of VSB100. For extracting per-frame superpixels from the constructed hierarchical segmentation we use the level selected on the validation set.

As shown in the plots and the table, the proposed method improves the baselines considered. The segmentation propagation [15] method improves $\sim$5 percent points on the BPR metrics, and 1$\sim$2 points on the VPR metrics. This supports that employing temporal cues helps to improve temporal consistency across frames. Our superpixels also boosts the performance of the approach from [14].

Employing our method for graph-based video segmentation also benefits computational load, since it depends on the number of nodes in the graph (number of generated superpixels). On average the number of nodes is reduced by a factor of 2.6, 120 superpixels per frame versus 310 in [14]. This leads to $\sim$ 45% reduction in runtime and memory usage for video segmentation.

Given the videos and their optical flow, the superpixel computation takes 90% of the total time and video segmentation only 10% (for both [14] and our SPX+[14]). Our superpixels are computed 20% faster than $\text{gPb}_{\mathcal{I}}$+$\text{gPb}_{\mathcal{F}}$ (the



(a) BPR on BMDS test set (b) VPR on BMDS test set

Figure 8: Comparison of state-of-the-art video segmentation algorithms with the proposed superpixels, on BMDS [6]. Dashed lines indicate only frame-by-frame processing (see §7.2 for details).

bulk of the time is spent in OP $(\cdot)$). The overall time of our approach is 20% faster than [14].

Qualitative results are shown in Figure 7. Superpixels generated from the proposed boundaries allow the baseline methods [15,14] to better distinguish visual objects and to limit label leakage due to inherent temporal smoothness of the boundaries. Qualitatively the proposed superpixels improve video segmentation on easy (e.g. first row of Figure 7) as well as hard cases (e.g. second row of Figure 7).

As our approach is directly applicable to any graph-based video segmentation technique we additionaly evaluated our superpixels with the classifier-based graph construction method of [24]. The method learns the topology and edge weights of the graph using features of superpixels extracted from per-frame segmentations. We employed this approach without re-training the classifiers on the proposed superpixels. Using our superpixels alows to achieve on par performance (see Figure 6 and Table 1) while significantly reducing the runtime and memory load ($\sim$ 45%). Superpixels based on per-frame boundary estimation are also employed in [41]. However we could not evaluate its performance with our superpixels as the code is not available under open source.

BMDS Further we evaluate the proposed method on BMDS [6] to show the generalization of our superpixels across datasets. We use the same model trained on VSB100 for generating superpixels and the hierarchical level of boundary map as validated by a grid search on the training set of BMDS. The results are presented in Figure 8. Our boundaries based superpixels boost the performance of the baseline methods [15,14], particularly for the BPR metric (up to 4-12%).

Oracle Additionally we set up the oracle case for the baseline [14] (purple curve in Figure 8) by choosing the hierarchical level to extract superpixels from the boundary map for each video sequence individually based on its performance (we considered OSS measures for BPR and VPR of each video). The oracle result

indicates that the used fixed hierarchical level is quite close to an ideal video-per-video selection.

8 Conclusion

The presented experiments have shown that boundary based superpixels, extracted via hierarchical image segmentation, are a better starting point for graph-based video segmentation than classical superpixels. However, the segmentation quality depends directly on the quality of the initial boundary estimates.

Over the state-of-the-art methods such as $\text{SE}_{\mathcal{I}}$ [10], our results show that we can significantly improve boundary estimates when using cues from object proposals, globalization, and by merging with optical flow cues. When using superpixels built over these improved boundaries, we observe consistent improvement over two different video segmentation methods [15,14] and two different datasets (VSB100, BMDS). The results analysis indicates that we improve most in the cases where baseline methods degrade.

For future work we are encouraged by the promising results of object proposals. We believe that there is room for further improvement by integrating more semantic notions of objects into video segmentation.

References

1. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012.
2. P. Arbeláez, M. Maire, C. C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 2011.
3. V. Badrinarayanan, I. Budvytis, and R. Cipolla. Mixture of trees probabilistic graphical model for video segmentation. *IJCV*, 2013.
4. M. V. D. Bergh, G. Roig, X. Boix, S. Manen, and L. V. Gool. Online video seeds for temporal window objectness. In *ICCV*, 2013.
5. G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *CVPR*, 2015.
6. T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.
7. J. Chang, D. Wei, and J. W. Fisher. A video representation using temporal superpixels. In *CVPR*, 2013.
8. H.-T. Cheng and N. Ahuja. Exploiting nonlocal spatiotemporal structure for video segmentation. In *CVPR*, 2012.
9. J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv:1412.1283*, 2014.
10. P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *TPAMI*, 2015.
11. K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *CVPR*, 2015.
12. K. Fragkiadaki and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.
13. F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012.

14. F. Galasso, M. Keuper, T. Brox, and B. Schiele. Spectral graph reduction for efficient image and streaming video segmentation. In *CVPR*, 2014.
15. F. Galasso, N. S. Nagaraja, T. Z. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013.
16. M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.
17. S. Hallman and C. Fowlkes. Oriented edge forests for boundary detection. In *CVPR*, 2015.
18. A. Humayun, F. Li, and J. M. Rehg. Rigor: Recycling inference in graph cuts for generating object regions. In *CVPR*, 2014.
19. A. Humayun, F. Li, and J. M. Rehg. The middle child problem: Revisiting parametric min-cut and seeds for object proposals. In *ICCV*, 2015.
20. P. Isola, D. Zoran, D. Krishnan, , and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In *ECCV*, 2014.
21. A. Jain, S. Chatterjee, and R. Vidal. Coarse-to-fine semantic video segmentation using supervoxel trees. In *ICCV*, 2013.
22. M. Keuper and T. Brox. Point-wise mutual information-based video segmentation with high temporal consistency. *arXiv:1606.02467*, 2016.
23. A. Khoreva, F. Galasso, M. Hein, and B. Schiele. Learning must-link constraints for video segmentation based on spectral clustering. In *GCPR*, 2014.
24. A. Khoreva, F. Galasso, M. Hein, and B. Schiele. Classifier based graph construction for video segmentation. In *CVPR*, 2015.
25. P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014.
26. A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *TPAMI*, 2009.
27. P. Neubert and P. Protzel. Evaluating superpixels in video: Metrics beyond figure-ground segmentation. In *BMVC*, 2013.
28. A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.
29. P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 2014.
30. G. Palou and P. Salembier. Hierarchical video representation with trajectory binary partition tree. In *CVPR*, 2013.
31. A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.
32. J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *arXiv:1503.00848*, 2015.
33. X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.
34. J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *CVPR*, 2015.
35. J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 2000.
36. P. Sundberg, T.Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.
37. E. Taralova, F. D. la Torre, and M. Hebert. Motion words for videos. In *ECCV*, 2014.
38. S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung. Maximin affinity learning of image segmentation. In *NIPS*, 2009.
39. A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.
40. S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015.
41. S. Yi and V. Pavlovic. Multi-cue structure preserving mrf for unconstrained video segmentation. In *ICCV*, 2015.