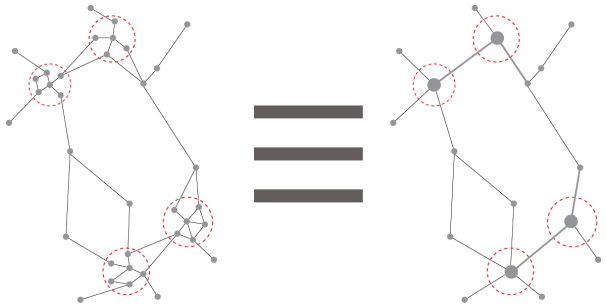
Updated paper (3 July 2014)
The condition of Theorem 2 differs.
Please also see the supplementary material.

Spectral Graph Reduction for Efficient Image and Streaming Video Segmentation

Fabio Galasso[1], Margret Keuper[2], Thomas Brox[2], Bernt Schiele[1]
[1] Max Planck Institute for Informatics, Germany
[2] University of Freiburg, Germany

Abstract

Computational and memory costs restrict spectral techniques to rather small graphs, which is a serious limitation especially in video segmentation. In this paper, we propose the use of a reduced graph based on superpixels. In contrast to previous work, the reduced graph is reweighted such that the resulting segmentation is equivalent, under certain assumptions, to that of the full graph. We consider equivalence in terms of the normalized cut and of its spectral clustering relaxation. The proposed method reduces runtime and memory consumption and yields on par results in image and video segmentation. Further, it enables an efficient data representation and update for a new streaming video segmentation approach that also achieves state-of-the-art performance.



(a) Original graph (b) Equivalent graph

Figure 1. In graph-based segmentation, certain associations among pixels may be assumed in the form of superpixels. These are *must-link* constraints. In spectral clustering, balancing plays a key role to exploit these constraints. We provide two graph reductions to preserve the normalized cut or its spectral clustering relaxation.

1. Introduction

Spectral clustering methods build the basis of many state-of-the-art image [2] and video [12, 3, 7, 11, 24] segmentation techniques. Spectral methods convince by their ability to naturally include long-range affinities. Their *globalization* effect (cf. [10]) helps robustness and is an important part of the hierarchical image segmentation of [2]. While segmentation techniques based on minimum cut formulations are suited for problems with strong unary terms, the balancing property of spectral clustering techniques is favorable when the focus is on pairwise terms.

The resource requirements of spectral clustering are still manageable for image segmentation up to a certain image resolution, but the limits are easily reached with video data. [24] presented a heavily parallelized extension of the pixel-based image segmentation approach of [2] to whole videos running on a GPU cluster. Since such computational resources are often not available, video segmentation is usually implemented as a two-step approach. The graph is either constructed on pre-computed superpixels/supervoxels [27, 12, 16, 7] which are then clustered, or the clustering is applied to sparse trajectories and the result converted into a dense segmentation in a postprocessing step [3, 20].

In this paper, we focus on the methods of the first group and notice that, in conjunction with spectral clustering, they are limited by the same pre-groupings that they leverage. Consider the case of a superpixel which perfectly covers an entire visual object. In theory, such a superpixel is desirable, as it provides the maximum computational reduction with minimum increase in model error. However, in a spectral method [12, 11, 7] balanced solutions are preferred. Thus, single-node clusters, such as the single superpixel covering the entire object, are almost impossible. The standard graph reduction by superpixels of [12, 11, 7] is not compatible with the association balancing of spectral clustering. The volume of the superpixels must be taken into account.

To this end, we investigate two graph reduction approaches which guarantee equivalence with the original graph under certain assumptions (Figure 1). Hereby, it is assumed that all pixels within a superpixel are connected by *must-link constraints*. The first approach is drawn from [21] and reduces the original graph such that the normalized cut (NCut) is preserved. However, finding the NCut is NP-hard and all spectral techniques recur to relaxations to solve

the problem. The most common 2-norm relaxation [23, 19] has proved successful in segmentation, but it may yield solutions far from the NCut. Consequently, the (theoretical) equivalence to the NCut practically still leads to different solutions. We introduce therefore a graph reduction that preserves the relaxed solution under stronger assumptions. If the assumptions are not satisfied, the proposed reduction leads to optimizing a new *density-normalized-cut*, which can be favorable in some applications.

The proposed reduction models (Section 3) can be applied readily in image and video segmentation. Runtime and memory consumption are reduced in all cases. Image segmentation (Section 4) performance stays on par with the state-of-the-art. Video segmentation (Section 5) outperforms all but one recent methods. Additionally, the reduction techniques allow a new streaming video segmentation approach (Section 6), based on graph reduction over time, which goes beyond state-of-the-art performance.

2. Related work

The two main objectives of graph reduction have been: **i**. reducing the computational complexity and **ii**. constraining (biasing) the solutions. Along the first line of work, this relates to techniques reducing the number of data points in the graph. This is highly desired in spectral clustering [23, 19]. In image segmentation, [9] achieves the objective by employing the Nyström method, i.e. randomly sampling the points and extrapolating the approximate sampled solution to the whole system. Elsewhere, [30, 5] approximate the graph by representative points (*aka* landmarks). By contrast, we approximate the graph with superpixels.

Biasing a solution has also motivated much research. Biasing allows inclusion of additional information, e.g. provided in an interactive fashion, into the solution. In image segmentation, [32, 8, 18] propose models to include constraints into the spectral clustering solutions. In contrast to ours, these methods process the whole graph, i.e. the expensive eigendecomposition underlying spectral clustering is computed on the whole affinity matrix.

Recently, [21] has addressed both the computational complexity and the constraints and suggested a reduction that preserves the (normalized) cut. We study its reduction in the scope of image and video segmentation. [26] leverages a reduction close in spirit to [21] for image segmentation. We complement both approaches with a formulation that considers the spectral relaxation.

Eigensolvers based on multigrid methods temporarily reduce the number of nodes and speed up computation while keeping the solution of the original problem [15], yet the coarser grids do not fit well to image structures. In [6] an approximate algebraic multigrid has been proposed, which focuses on the use of different features at different scales rather than a fast exact solution of the single-grid problem. Our section on video segmentation is close in spirit to this approach. The recent work of Maire et al. [17] combines traditional multigrid ideas with the option to use different affinities at different scales.

3. Graph reduction model

This section provides the mathematical motivation for the proposed reduction models. We briefly review the normalized cut (NCut) and its relation to spectral clustering and refer to [28] for more details. After stating the model of equivalence from [21] in terms of the NCut, we introduce a new re-weighting method for the reduced graph that is associated with the relaxed NCut objective.

3.1. NCut and spectral clustering

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and undirected edges $e \in \mathcal{E}$. We cast the segmentation problem as a graph partitioning. The vertices in the graph represent the pixels or superpixels of an image or video to segment. The edges e_{ij} take weights from the pairwise affinities w_{ij} between the corresponding pixels or superpixels i and j.
NCut provides a clustering objective where the cluster sizes are balanced by their volumes. For the case of partitioning into two disjoint sets A and B, $A \cup B = \mathcal{V}$, $A \cap B = \emptyset$, it is

$$\text{NCut}(A,B) = \frac{\text{cut}(A,B)}{\text{vol}(A)} + \frac{\text{cut}(A,B)}{\text{vol}(B)}, \qquad (1)$$

with $\text{cut}(A,B) = \sum_{i\in A, j\in B} w_{ij}$ and $\text{vol}(A) = \sum_{i\in A, j\in\mathcal{V}} w_{ij}$.
Determining the optimal sets A and B requires solving the following minimization problem:

$$\min f^\top L f \quad \text{subject to} \quad Df \perp \vec{1}, \quad f^\top D f = \text{vol}(\mathcal{V}),$$
$$\text{and } f_i = \begin{cases} \sqrt{\frac{\text{vol}(B)}{\text{vol}(A)}} & \text{if } i \in A \\ -\sqrt{\frac{\text{vol}(A)}{\text{vol}(B)}} & \text{if } i \in B \end{cases}, \qquad (2)$$

where $L = D - W$ is the (unnormalized) graph Laplacian, W is the matrix containing the pairwise affinities w_{ij} and D is the diagonal degree matrix with $d_{ii} = \sum_{j\in\mathcal{V}} w_{ij}$.
Solving for NCut is NP-Hard. Its relaxation (allowing f to take arbitrary real values) leads to *spectral clustering*:

$$\min g^\top L_{\text{sym}} g = \min \sum_{i,j=1} w_{ij} \left(\frac{g_i}{\sqrt{d_{ii}}} - \frac{g_j}{\sqrt{d_{jj}}} \right)^2$$
$$\text{subject to} \quad g \perp D^{1/2}\vec{1},\ \|g\|^2 = \text{vol}(\mathcal{V}), \qquad (3)$$

where g replaces f by a change of coordinates, *i.e.* $g := D^{1/2} f$, and $L_{\text{sym}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ is the normalized graph Laplacian. The Rayleigh-Ritz theorem provides the solution to (3) as the eigenvector corresponding to the second smallest eigenvalue λ_2 of L_{sym}.

In what follows, we consider the case of two partitions, as the mathematical formulation is simpler and more intuitive. See [28] for a generalization to multi-partitionings in the case of spectral clustering.

3.2. Equivalence of graphs

Given a point grouping $\{I_1, \ldots, I_m\}$ that separates the image into disjoint superpixels and connects pixels in the graph via must-link constraints, we want to determine a reduced graph $\mathcal{G}^Q = (\mathcal{V}^Q, \mathcal{E}^Q)$, where the vertex set $\mathcal{V}^Q = \{I_1, \ldots, I_m\}$ is reduced to the set of superpixels, and the edges $\mathcal{E}^Q$ describe pairwise affinities w^Q_{IJ} between these vertices. The new graph $\mathcal{G}^Q$ should reflect the original volumes in a meaningful way, allowing for a balanced segmentation. This can be achieved by providing equivalence to the original $\mathcal{G}$ in terms of the NCut in (2). A second approach is motivated by finding equivalence in terms of (3) (the spectral clustering relaxation), yielding a graph re-weighting that leads to a different balancing term.

Equivalence of normalized cuts ($\mathcal{G}^Q$ NCut)

Theorem 1. *The NCut of partitions determined on the graph $\mathcal{G}^Q$ is the same as the NCut of the corresponding partitions on the original graph $\mathcal{G}$ if the affinity matrix W^Q is defined as*

$$w^Q{}_{IJ} = \sum_{i \in I} \sum_{j \in J} w_{ij} \tag{4}$$

While theoretically appealing from a graph perspective, the previous equivalence focusses on the cut objective. Given the fact that the optimal solution of (2) is usually not found, we introduce a graph re-weighting that, under certain conditions, provides equivalence to the spectral clustering problem. We further show that, if these conditions are not fulfilled, the proposed re-weighting optimizes a graph partitioning objective slightly different from NCut. Later we will show empirically that optimizing this objective often leads to favorable results.

Equivalence of spectral clustering ($\mathcal{G}^Q$ SC)

Theorem 2. *The graph $\mathcal{G}^Q$ is equivalent to the original graph $\mathcal{G}$ with corresponding partitions with respect to spectral clustering, i.e. $\mathcal{G}^Q$ preserves the relaxed spectral clustering optimization problem* (3)*, if*

$$w^Q_{IJ} = \begin{cases} \sum_{i \in I} \sum_{j \in J} w_{ij} & \text{if } I \neq J \\ \frac{1}{|I|} \sum_{i \in I} \sum_{j \in J} w_{ij} - \frac{(|I|-1)}{|I|} \sum_{i \in I} \sum_{j \in \mathcal{V} \setminus I} w_{ij} & \text{if } I = J \end{cases} \tag{5}$$

provided equal affinities of nodes of $\mathcal{G}$ constrained in $\mathcal{G}^Q$:

$$w_{ik} = w_{jk} \quad \forall i, j, k \in \mathcal{G} \;\; : \;\; (i,j) \in I \;, \;\; I \in \mathcal{V}^Q \tag{6}$$

When the assumptions of Theorem 2 do not apply, (5) raises a new balanced cut problem, which we denote *density normalized cut* (DNCut):

Theorem 3. *A^Q and B^Q are solutions of* (3) *in graph $\mathcal{G}^Q$, reduction of $\mathcal{G}$ according to* (5)*,* ***iff*** *A and B are solutions in $\mathcal{G}$ of the DNCut problem:*

$$DNCut(A,B) = \frac{cut(A,B)}{\sum_{I \in A^Q} \frac{vol(I)}{|I|}} + \frac{cut(A,B)}{\sum_{I \in B^Q} \frac{vol(I)}{|I|}} \tag{7}$$

given A and B which respect the must-link constraints of $\mathcal{G}^Q$: $A = \{i \in \mathcal{V} | i \in I, I \in A^Q\}$ and similarly B.

Both graph reductions of Theorems 1 and 2 yield results that are superior to using superpixels without the proposed reduction equations. Note that, in our application on video segmentation in Section 5, the condition of equal affinities in Theorem 2 always applies by construction.

While NCut and DNCut coincide for equally sized superpixels, DNCut provides a different balancing term if this is not the case. Since the volume of a vertex set tends to increase with its cardinality, NCut yields clusters that are to a certain degree also balanced by their size. In DNCut, the volumes of every superpixel are normalized by their sizes such that the balancing is done purely based on the point association. Thus, even large segments have to be justified by their affinities. As shown in the following sections, this leads to favorable results in image and video segmentation.

4. Image segmentation

The state-of-the-art hierarchical image segmentation algorithm of [2] consists of three main components (illustrated in Figure 2). Local edge information is captured by **mPb**, which is a linear combination of color, brightness, and texture gradients at multiple scales and orientations. More global information is provided by **sPb**, which is the spectral decomposition of an affinity matrix of intervening mPb contours among pixels and the computation of eigenvector gradients. This spectral part is mainly responsible for the large consumption of memory and computational resources. Finally a hierarchical boundary map with closed boundaries is computed from the linear combination of mPb and sPb (called global Pb, or **gPb**), via an oriented watershed transform (owt) and ultrametric contour map (ucm).

Our proposed graph reduction addresses the expensive sPb step by grouping *most certain* pixel associations into superpixels based on mPb and reducing the per-pixel affinity matrix of [2] to a smaller matrix. The original matrix corresponds to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the pixels represent the nodes $\mathcal{V}$. They are connected to their neighbors within a certain radius r ($r = 5$ in [2]), by edges $e_{ij} \in \mathcal{E}$ that are weighted by the maximal intervening mPb bound-

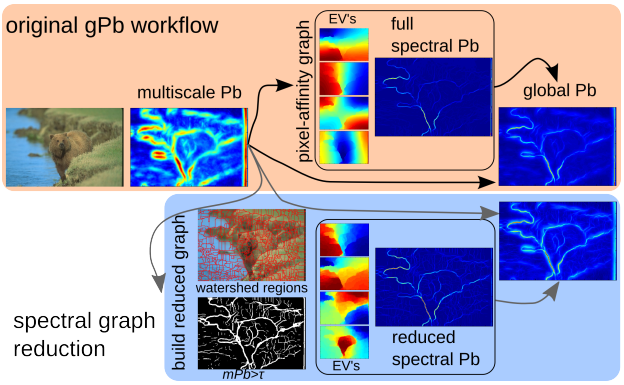


Figure 2. Workflow of [2] up to gPb. The computation of sPb on the pixel graph $\mathcal{G}$ is replaced by computation in the reduced pixel-and-superpixel graph $\mathcal{G}^Q$. Superpixels are provided by watershed regions; in areas where (mPb) $> \tau$ the grouping is considered uncertain and the single pixels are preserved as nodes. ($\tau = 0$ means using pixels just, i.e. the original problem, $\tau = 1$ means using only superpixels.)

ary along the line $\bar{ij}$:

$$w_{ij} = \exp\left(-\max_{p\in\bar{ij}} \max_{\theta} \frac{mPb(p,\theta)}{\rho}\right) \qquad (8)$$

with constant ρ. To reduce the complexity of the spectral analysis step, we pre-group the pixels into superpixels by computing watershed segmentation with the mPb values; cf. Figure 2. This pre-grouping is reliable in terms of boundary recall, i.e. important boundaries are not missed. However, mPb does not allow for an exact boundary localization. Therefore, in areas with mPb beyond a threshold τ, we keep a node for every single pixel in the reduced graph. Finally we compute sPb from the reduced graph, where the affinities are re-weighted according to either Theorem 1 or Theorem 2. We denote the two graphs as $\mathcal{G}^Q$ NCut and $\mathcal{G}^Q$ SC respectively.

4.1. Experimental evaluation

The effect of the graph reductions is evaluated on the BSDS 500 benchmark [2]. In Figure 3 (*left*), we present a study on the boundary precision-recall (BPR) and memory footprint of the sPb affinity matrix versus the mPb threshold τ, which determines superpixel aggregation within the watershed regions. We choose $\tau = 0.3$, which does not alter performance but reduces memory consumption by 50% and sPb processing time from 71.5 to 33.5 secs. (This includes the reduction cost, negligible in practise.)

Figure 3 (*right*) shows that the reduced graph $\mathcal{G}^Q$ leads to the same performance as the original algorithm of [2] using the full pixel graph. The reduced graph $\mathcal{G}^Q$ SC even yields better results in the high-precision BPR range and, consequently, achieves a higher average precision (AP) than the original; cf. Table 1. Moreover, the eigenvectors (EV's)

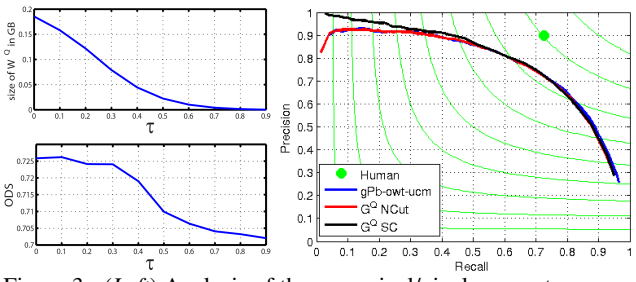


Figure 3. (*Left*) Analysis of the superpixel/pixel parameter τ versus boundary precision-recall (BPR) performance (*top*) and memory footprint of the affinity matrix in sPb (*bottom*) on all train images in BSDS500 [2]. (*Right*) Comparison of BPR performance of [2] against the proposed graph reduction methods on BSDS500.

	ODS	OIS	AP
[2] gPb-owt-ucm	**0.73**	**0.76**	0.73
$\mathcal{G}^Q$NCut	0.72	0.75	0.74
$\mathcal{G}^Q$SC	**0.73**	0.75	**0.75**

Table 1. Aggregated boundary performance evaluation (optimal dataset scale [ODS], optimal image scale [OIS], average precision [AP]) of [2] and our proposed methods on BSDS 500.

extracted by this reduction technique are a more compact representation than those of the full graph: in Figure 2, the first EV's of the original graph are mainly influenced by the global image coordinates (they emphasize groupings along the horizontal, vertical or diagonal direction). By contrast, the $4th$ eigenvector of the reduced graph already highlights the *bear* in the image. Note that, both for the SC and NCut reductions, the eigenvectors of the grouped pixels become piece-wise constant and thus closer, as we conjecture, to their respective non-relaxed problems, i.e. DNCut and NCut. This may further explain the slight improvement in performance.

5. Video segmentation

We use the graph reduction and re-weighting to develop a video segmentation method that combines more discriminative features computed at a coarse superpixel level with the balanced graph structure of the pixel or superpixel level. Moreover, the method benefits from the lower resource requirements by computing the solution at the coarse level; see Figure 4 for a sketch of the overall concept.

The method builds upon Galasso et al. [12], who construct a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ for the entire video sequence, where nodes $\mathcal{V}' = \{i_1, i_2, \ldots, i_n\}$ are pre-computed superpixels. Edges $\mathcal{E}'$ connect each superpixel to its direct spatial and temporal neighbors. Edge weights w'_{ij} are pairwise affinities between superpixels i and j, computed from motion, appearance and superpixel-shape features. In [12], superpixels consist of the lowest level (level 1) of the hierarchical image segmentation from [2], and a re-weighting

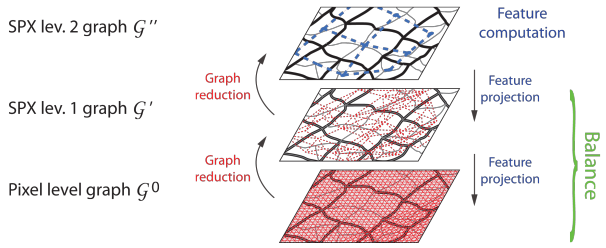


Figure 4. In video segmentation, discriminant affinities are computed from the coarse graph $\mathcal{G}'' = (\mathcal{V}'', \mathcal{E}'')$ among superpixels at level 2. Projection to finer graphs of superpixels at level 1 $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ or of pixels $\mathcal{G}^0 = (\mathcal{V}^0, \mathcal{E}^0)$ and further reduction are necessary to re-balance the volumes. We propose a reweighting which avoids the expensive expansions.

to account for the volume of superpixels is missing.

We use a second, coarser graph $\mathcal{G}'' = (\mathcal{V}'', \mathcal{E}'')$, where nodes $\mathcal{V}'' = \{I_1, I_2, \ldots, I_m\}$ represent coarser superpixels (level 2) from [2]; see an example in Figure 5. Affinities ${w''}_{IJ}$ between superpixels at level 2 are determined based on the features from [12], yet they are re-weighted to take into account the better balance of the finer graph structure of $\mathcal{G}'$. (cf. Introduction, superpixels at level 2 show large variations in size which impinge the computation of balanced cut solutions.) In this case, the reduced equivalent graph $\mathcal{G}^Q$ is identical to $\mathcal{G}''$ up to the re-weighted affinities w^Q_{IJ}.

Re-weighting implies first a projection of edges e''_{IJ} to the finer graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of superpixels at level 1. This is a *higher-order* expansion problem, since an edge e''_{IJ} connects generally more than 2 superpixels at level 1. *Clique expansion* [1] yields a possible mathematical formulation:

$$\mathcal{E}' = \{{e'}_{ij} \,|\, \exists\, {e''}_{IJ} \in \mathcal{E}'',\, i \in I,\, j \in J\} \tag{9}$$

$${w'}_{ij} = \sum_{e'' \in \mathcal{E}'' : (I,J) \in e'', i \in I, j \in J} {w''}_{IJ}. \tag{10}$$

In fact, the costly computation of the clique expanded graph $\mathcal{G}'$ is not necessary, as the re-weighting can be done directly on the coarser graph $\mathcal{G}''$.

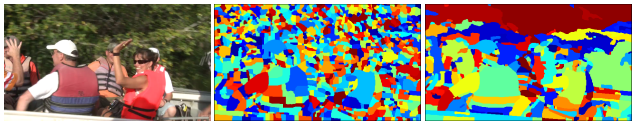
According to Theorem 1, the reduced graph $\mathcal{G}^Q$ preserving the normalized cut of $\mathcal{G}'$ gets the weights:

$${w^Q}_{IJ} = \begin{cases} |I||J|{w''}_{IJ} & \text{if } I \neq J \\ |I|(|I|-1) \sum_{K \neq I} {w''}_{IK} & \text{if } I = J \end{cases} \tag{11}$$

According to Theorem 2, the reduced graph $\mathcal{G}^Q$ preserving the spectral clustering solutions of $\mathcal{G}'$ gets the weights:

$${w^Q}_{IJ} = \begin{cases} |I||J|{w''}_{IJ} & \text{if } I \neq J \\ (|I|-1) \sum_{K \neq I} (1-|K|){w''}_{IK} & \text{if } I = J \end{cases} \tag{12}$$

Rather than re-balancing with regard to a finer superpixel resolution, it is also possible to efficiently recompute the weights according to the original pixel level graph $\mathcal{G}^0$; see Figure 4.



riverboat SPX Lev.1 SPX Lev.2

Figure 5. Sample superpixels at level 1 (lowest layer) and 2 (*ucm*=0.12) extracted from a hierarchical image segmentation [2].

5.1. Experimental evaluation

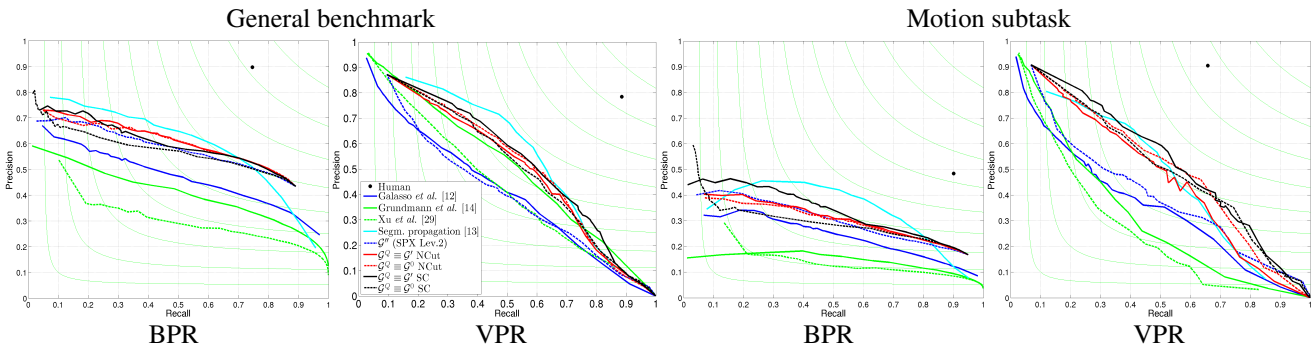
[13] has recently provided a challenging video segmentation benchmark based on the HD quality videos from [25], the boundary precision-recall (BPR) metric from [2] and a volume precision-recall metric (VPR) that reflects the properties of a good video segmentation, such as temporal consistency. The benchmark also considers various subtasks including a motion segmentation task, where only moving objects are evaluated. We consider both the overall task and the motion subtask.

In Figure 6 (*top*), we compare the different variants of graph reduction to the state-of-the-art in video segmentation. We consider reduced graphs $\mathcal{G}^Q$ equivalent to the graph $\mathcal{G}'$ at superpixel level 1 or the graph $\mathcal{G}^0$ over the original pixels. Each version is considered with regard to equivalence of the normalized cut (NCut), cf. Theorem 1, and of the spectral clustering eigenproblem (SC), cf. Theorem 2.

Besides the precision-recall graphs, we also report aggregate performance indicators (optimal dataset scale [ODS], optimal segmentation scale [OSS], average precision [AP]) of BPR and VPR, for the general and motion task. The existing methods in block A (with the exception of [13]) are outperformed by the graph reduction methods in block B. Among these variants, $\mathcal{G}^Q \equiv \mathcal{G}'$ SC, using the spectral clustering equivalence, performs best. The average improvement to the best of those video segmentation methods is 7.3% on the overall task and even 11% on the motion subtask. Qualitative examples are shown in Figure 7.

Comparing the results for the graph $\mathcal{G}''$ of superpixels at level 2 (without re-weighting) to results from [12], where superpixels at level 1 are used without re-weighting, shows that affinities estimated over larger superpixels are advantageous. This is probably because they are computed over larger homogeneous image areas where, e.g., motion can be computed more reliably. Additionally re-balancing $\mathcal{G}''$ further improves performance, especially with regard to VPR.

The algorithm of [13] (hierarchical image segmentation [2] propagated with optical flow) is only partially outperformed by the proposed methods. This may be explained by the more complex image features of [2], e.g. texture descriptors. There is much to improve on the low level features of [12] (mean color and motion), which we use here. This is further substantiated by our better results on the motion tasks, where static image features play a less prominent



BPR VPR BPR VPR

	General benchmark						Motion subtask						Volume statistics	
	BPR			VPR			BPR			VPR			Length	NCL
Algorithm	ODS	OSS	AP	ODS	OSS	AP	ODS	OSS	AP	ODS	OSS	AP	$\mu(\delta)$	μ
Human	0.81	0.81	0.67	0.83	0.83	0.70	0.63	0.63	0.44	0.76	0.76	0.59	83.24(40.04)	11.90
A. State-of-the-art video segmentation algorithms														
*Corso et al. [6]	0.47	0.48	0.32	0.51	0.52	0.38	0.21	0.21	0.11	0.37	0.35	0.23	70.67(48.39)	25.83
*Galasso et al. [12]	0.51	0.56	0.45	0.45	0.51	0.42	0.34	0.43	0.23	0.42	0.46	0.36	80.17(37.56)	8.00
*Grundmann et al. [14]	0.47	0.54	0.41	0.52	0.55	0.52	0.25	0.34	0.15	0.37	0.41	0.32	87.69(34.02)	18.83
*Ochs and Brox [20]	0.17	0.17	0.06	0.25	0.25	0.12	0.26	0.26	0.08	0.41	0.41	0.23	87.85(38.83)	3.73
Xu et al. [29]	0.38	0.46	0.32	0.45	0.48	0.44	0.22	0.30	0.15	0.32	0.36	0.27	59.27(47.76)	26.58
Segm. propagation [13]	**0.61**	**0.65**	**0.59**	**0.59**	**0.62**	**0.56**	**0.47**	**0.52**	**0.34**	**0.52**	**0.57**	**0.47**	25.50(36.48)	258.05
B. Proposed graph reduction methods: $\mathcal{G}^Q$ equiv. to SPX Lev.1 ($\mathcal{G}'$) or pixel level ($\mathcal{G}^0$) graphs, wrt NCut or spectral clustering (SC)														
*$\mathcal{G}^Q \equiv \mathcal{G}'$ NCut	**0.62**	**0.66**	**0.54**	0.54	0.57	0.52	0.39	**0.49**	0.28	0.51	0.54	0.48	69.08(40.91)	20.00
*$\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut	**0.62**	**0.66**	0.53	**0.55**	**0.59**	0.53	0.39	0.48	0.28	0.53	0.59	0.52	71.77(40.27)	20.00
*$\mathcal{G}^Q \equiv \mathcal{G}'$ SC	**0.62**	**0.66**	**0.54**	**0.55**	**0.59**	**0.55**	**0.41**	**0.49**	**0.32**	**0.55**	**0.60**	**0.54**	61.25(40.87)	80.00
*$\mathcal{G}^Q \equiv \mathcal{G}^0$ SC	0.60	0.65	0.52	0.53	0.57	0.52	0.38	0.45	0.28	0.52	0.56	0.52	52.81(46.25)	100.00
*$\mathcal{G}''$ (SPX Lev.2)	0.60	0.65	0.52	0.44	0.50	0.42	0.38	**0.49**	0.28	0.44	0.51	0.41	60.42(41.85)	18.00
C. Streaming algorithms: variants of methods ($\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut) and ($\mathcal{G}^Q \equiv \mathcal{G}'$ SC) above, wrt max number of nodes maintained														
*Stream^{5k} $\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut	**0.61**	**0.67**	**0.52**	**0.55**	**0.59**	**0.53**	0.39	**0.50**	0.28	**0.52**	**0.58**	0.50	73.31(40.33)	15.63
*Stream^{500} $\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut	**0.61**	0.66	**0.52**	**0.55**	**0.59**	0.52	**0.40**	0.48	0.27	0.51	0.56	0.48	68.61(41.21)	16.57
*Stream^{5k} $\mathcal{G}^Q \equiv \mathcal{G}'$ SC	0.59	0.64	**0.52**	0.53	0.56	0.52	**0.40**	0.47	**0.29**	**0.52**	0.55	**0.51**	33.55(36.11)	116.78
*Stream^{500} $\mathcal{G}^Q \equiv \mathcal{G}'$ SC	0.58	0.62	0.51	0.51	0.51	0.51	0.37	0.43	0.28	**0.52**	0.53	**0.51**	3.17(11.49)	586.47

Figure 6. Comparison of (*block A in table*) state-of-the-art video segmentation algorithms [6, 14, 12, 29, 13], (*block B*) the proposed spectral graph reduction methods and (*block C*) the streaming variants, on the benchmark of [13]. The plots show boundary precision-recall (BPR) and volume precision-recall (VPR) curves, for the *general* benchmark and the *motion* segmentation subtask. The table shows aggregate performance evaluations (optimal dataset scale [ODS], optimal segmentation scale [OSS], average precision [AP]) of BPR and VPR and includes length statistics (mean μ and std. dev. δ) and no. of clusters (NCL). (*) indicates video frames resized by 0.5. **Best performances highlighted in each block.** (*Plots and blocks A, B*) the proposed methods based on graph reductions ($\mathcal{G}^Q \equiv \mathcal{G}'$ NCut, $\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut, $\mathcal{G}^Q \equiv \mathcal{G}'$ SC, $\mathcal{G}^Q \equiv \mathcal{G}^0$ SC) outperform all recent methods but segmentation propagation [13] (see Section 5). (*Blocks A, B, C*) The proposed streaming variants *Stream*^{X} of two proposed reduction methods marginally decrease their performance and provide state-of-the-art results. (X in *Stream*^{X} indicates the maximum number of nodes maintained over the whole video.) (see Section 6.)

role.

Regarding runtime and memory usage, the algorithm of [12] peaks for sequences like *koala*, *panda* and *buffalos*, where the number of superpixels (nodes in graph $\mathcal{G}'$) is $\sim 200k$ ($k = 1000$). For the same sequences, the number of superpixels at level 2 (nodes of $\mathcal{G}^Q$) is $\sim 65k$. On average, the number of superpixels reduces by a factor of 3, which means that only 10% of the runtime and 20% of the memory is needed.

6. Streaming video segmentation

In a streaming scenario new data becomes available only sequentially and should be discarded after processing. In the case of video segmentation, this yields two constraints: **i**. future frames are not yet available and **ii**. only a limited fraction of information from previous frames can be kept in memory. The first constraint appears in all applications that must make decisions in real-time[1]. The second constraint appears as large amounts of data must be processed with limited memory resources.

Some video segmentation algorithms from literature are also available in *streaming mode* [14, 29, 4] (sometimes *online* is used as a synonym), yet they all recur to a strict Markov assumption (cf. [29]). This means that the segmentation at the current frame is computed from the cur-

[1]The definition of real-time depends on the application.

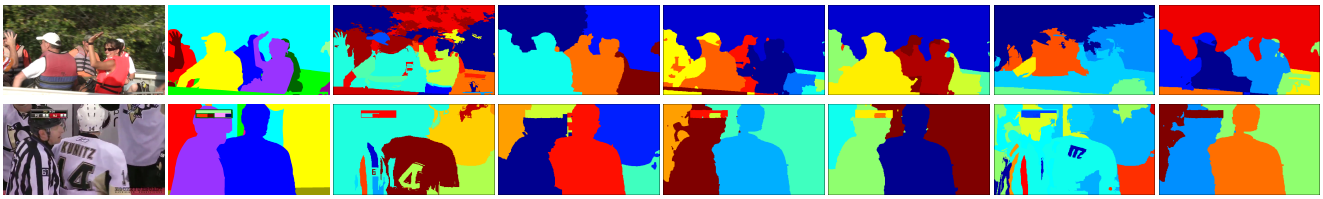


Figure 7. Video segmentation results comparing the algorithms of [14, 12, 29] and our proposed spectral graph reduction methods to one of the available ground truths [13] (we report for each algorithm the coarse-to-fine level with best performance in VPR). Both our best performing methods ($\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut, $\mathcal{G}^Q \equiv \mathcal{G}'$ SC) qualitatively improve on the algorithm of [12], better discerning the visual objects. The proposed streaming method (Stream^{500} $\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut) provides similar results as its batch version ($\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut).

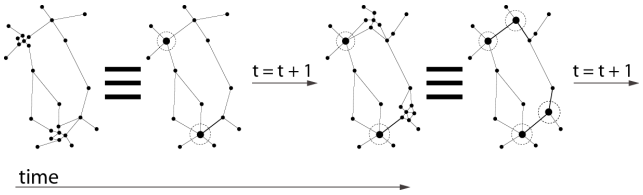
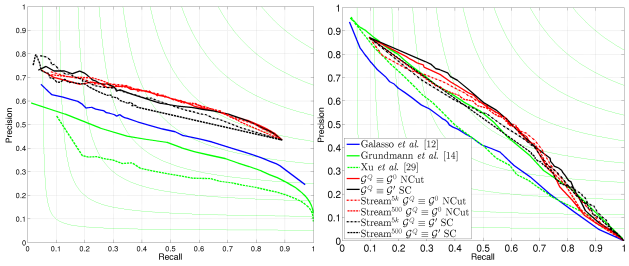


Figure 8. In streaming video segmentation, the spectral graph reduction allows a novel approach whereby, at each instant in time $t = i$, only those nodes for which a grouping is available with largest certainty are merged. According to the hypothesized grouping, an equivalent reduced graph $\mathcal{G}^Q_{1:i}$ is determined, which is carried over at the following instance in time $t = t+1$, iterating the process. At time $t = i$ the optimal (over frames $1 : i$) video segmentation is obtained by clustering the equivalent graph $\mathcal{G}^Q_{1:i}$.



BPR VPR

Figure 9. Evaluation of streaming algorithms with BPR and VPR on the general benchmark of [13]. The proposed streaming variants (dashed/dotted red/black curves) of our graph reduction algorithms maintain performance and outperform video segmentation [14, 12] and streaming algorithms [29].

rent image V_i and the segmentations $S_{i-1}, ..., S_{i-n}$ from a fixed number n of previous frames. Often multiple images $V_i, ..., V_m$ rather than a single image are considered in a temporal window of a fixed size m.

Obviously, this suffers from early, suboptimal segmentation decisions. Especially when motion is necessary to disambiguate the appearance between the visual objects, necessary information may only emerge at a later frame. While in a streaming setting there is no way to avoid this problem on the first frames, the lacking information at the beginning should not affect the quality of the segmentation in the current and future frames.

Rather than taking into account just information from a fixed number of frames, we propose a more flexible data reduction where certain decisions are made immediately and uncertain decisions are postponed to the time when the necessary information is available. This concept is known as *deferred inference* in tracking [22, 31].

The proposed graph reduction enables an implementation of this concept in video segmentation as illustrated in Figure 8. Given a graph $\mathcal{G}_{1:i}$ associated to the observed portion of a video sequence $V_{1:i} = \{V_1, \dots, V_i\}$, we propose to only merge those nodes for which a grouping is available with larger certainty, determining therefore a reduced graph $\mathcal{G}^Q_{1:i}$ to carry over. At each point in time, the optimal segmentation $S^*_{1:i}$ is provided by clustering the reduced graph. The provided optimal solution from $\mathcal{G}^Q_{1:i}$ is equivalent to $\mathcal{G}_{1:i}$ under the conditions stated in Section 3, if the optimal solution from $\mathcal{G}_{1:i}$ is the superset of the groupings in $\mathcal{G}^Q_{1:i}$. Therefore, we formulate the streaming algorithm from time $i-1$ to i as two basic steps: **i**. input new frames into the current graph, i.e. $\mathcal{G}^Q_{1:i-1} \cup \mathcal{G}_i$; **ii**. cluster the graph into a superset of hypotheses and reduce it accordingly, i.e. $\mathcal{G}^Q_{1:i}$.

Implementation details. If looking for tens of segments, a superset of hypotheses may be given by thousands of nodes (we experiment on this parameter). The clustering step is an iteration of spectral clustering: compute eigenvectors and cluster with k-means. Two extensions are required for k-means: **i**. seed solutions from existing groupings (clusters are initialized with the random partition method, using labels from the previous iterations when available); **ii**. consider the multiplicity of the groupings in the update of cluster centers (the estimated cluster means are weighted by the node cardinalities).

6.1. Experimental evaluation

We evaluate the proposed streaming video segmentation algorithms in the same setup as described in Section 5.1. Figure 9 compares existing approaches [14, 12, 29] and selected graph reduction variants from the previous section ($\mathcal{G}^Q \equiv \mathcal{G}^0$ NCut, $\mathcal{G}^Q \equiv \mathcal{G}'$ SC) with the proposed stream-

ing variants *Stream*^{X}. X is the maximum number of nodes maintained over the whole video. We vary this number from 5000 to 500, out of a total of ~ 33000 nodes contained in the full graph at level 2. Note that 500 nodes are as much as the number of superpixels at level 2 within $\sim$1.5 frames. The table in Figure 6 shows the corresponding aggregate performance measures.

A good streaming algorithm should get as close as possible to the performance of its batch processing correspondent. All proposed streaming variants only decrease performance by a few percentage points, but still achieve state-of-the-art results. In particular, streaming reduction based on NCut preserves performance best. Its performance decrease of 1.9% for Stream^{500} is very small compared to the 12.8% decrease in the streaming extension [29] of Grundmann et al. [14]. Streaming methods based on spectral clustering maintain performance but provide less desirable segmentations (length statistics in the table in Figure 6). This is due to the changing balance among clusters over time, e.g. objects entering the scene or zooming.

Both streaming variants further reduce runtime and memory consumption. At each point in time, Stream^{500} only keeps in memory 500 nodes ($\sim$1.5 frames) plus the newly input f frames ($1 \leq f \leq n$ depending on the application). Given ~ 330 superpixels at level 2 per frame, the reduction stage runs spectral clustering on $|\mathcal{V}^Q| \approx 500 + 330f$. This reduces runtime and memory needs of [12] to 5% and 2% respectively (we set $f = 10$ but increment $\mathcal{G}^Q$ of 1 frame for increased robustness).

7. Conclusions

We have addressed the common practice in video segmentation to work with graphs based on aggregated superpixels rather than single pixels. We have shown that the corresponding graphs should be re-weighted and analyzed two formulations that maintain the solutions of the full graph under certain conditions. Our experiments encourage the use of superpixels in conjunction with one of the presented re-weighting schemes. We have achieved state-of-the-art performance in image and video segmentation. At the same time runtime and memory consumption have been reduced significantly. Further, we have introduced the concept of deferred inference in streaming video segmentation, which leads to top results also in this field.

Acknowledgements

We acknowledge partial funding by the ERC Starting Grant VideoLearn.

References

[1] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *ICML*, 2006.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011.

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

[4] J. Chang, D. Wei, and J. W. Fisher. A video representation using temporal superpixels. In *CVPR*, 2013.

[5] X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, 2011.

[6] J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille. Efficient multilevel brain tumor segmentation with integrated bayesian model classification. *TMI*, 2008.

[7] X. Di, H. Chang, and X. Chen. Multi-layer spectral clustering for video segmentation. In *ACCV*, 2012.

[8] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *ICCV*, 2007.

[9] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *TPAMI*, 26(2), 2004.

[10] C. Fowlkes and J. Malik. How much does globalization help segmentation? Technical report, EECS – UC Berkeley, 2004.

[11] K. Fragkiadaki and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.

[12] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012.

[13] F. Galasso, N. S. Nagaraja, T. J. Cárdenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013.

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

[15] D. Kushnir, M. Galun, and A. Brandt. Efficient multilevel eigensolvers with applications to data analysis tasks. *TPAMI*, 32(8), 2010.

[16] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Spatiotemporal closure. In *ACCV*, 2010.

[17] M. Maire and S. X. Yu. Progressive multigrid eigensolvers for multiscale spectral segmentation. In *ICCV*, 2013.

[18] S. Maji, N. Vishnoi, and J. Malik. Biased normalized cuts. In *CVPR*, 2011.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[20] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011.

[21] S. Rangapuram and M. Hein. Constrained 1-spectral clustering. In *AISTATS*, pages 1143–1151, 2012.

[22] D. B. Reid. An algorithm for tracking multiple targets. *Automatic Control*, 24(6), 1979.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000.

[24] N. Sundaram and K. Keutzer. Long term video segmentation through pixel level spectral clustering on gpus. In *ICCV Workshops*, 2011.

[25] P. Sundberg, T.Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.

[26] C. Taylor. Towards fast and accurate segmentation. In *CVPR*, 2013.

[27] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.

[28] U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4), 2007.

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

[30] D. Yan, L. Huang, and M. Jordan. Fast approximate spectral clustering. In *KDD*, pages 907–916, 2009.

[31] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.

[32] S. X. Yu and J. Shi. Grouping with bias. In *NIPS*, 2001.