Occlusion-Aware Method for Temporally Consistent Superpixels

Matthias Reso, Jörn Jachalsky, *Member, IEEE*, Bodo Rosenhahn, and Jörn Ostermann, *Fellow, IEEE*

Abstract—A wide variety of computer vision applications rely on superpixel or supervoxel algorithms as a preprocessing step. This underlines the overall importance that these approaches have gained in recent years. However, most methods show a lack of temporal consistency or fail in producing temporally stable superpixels. In this paper, we present an approach to generate temporally consistent superpixels for video content. Our method is formulated as a contour-evolving expectation-maximization framework, which utilizes an efficient label propagation scheme to encourage the preservation of superpixel shapes and their relative positioning over time. By explicitly detecting the occlusion of superpixels and the disocclusion of new image regions, our framework is able to terminate and create superpixels whose corresponding image region becomes hidden or newly appears. Additionally, the occluded parts of superpixels are incorporated in the further optimization. This increases the compliance of the superpixel flow with the optical flow present in the scene. Using established benchmark suites, we show that our approach produces highly competitive results in comparison to state-of-the-art streaming-capable supervoxel and superpixel algorithms for video content. This is further shown by comparing the streaming-capable approaches as basis for the task of interactive video segmentation where the proposed approach provides the lowest overall misclassification rate.

Index Terms—Video segmentation, oversegmentation, supervoxels, superpixels

1 Introduction

The idea to group spatially coherent pixels sharing similar low-level features like color or texture into so called superpixels and utilize them as primitives for image analysis and processing was introduced by Ren and Malik in [1]. The pixel grouping leads to a major reduction of image primitives, which results in an increased computational efficiency for subsequent processing steps and allows for more complex algorithms computationally infeasible on pixel level [1]. Another benefit is the creation of a spatial support for region-based features [2]. The applications of superpixels are widely spread and include e.g., tracking [3], scene flow [4], 3D layout estimation of indoor scenes [5], image parsing [6], video coding [7], and semantic segmentation [8].

Especially for video applications, the usage of superpixels instead of raw pixel data is beneficial. This has e.g., been shown in [9], [10] for the case of unsupervised video segmentation by partitioning a superpixel graph using spectral clustering. The usage of superpixels boosts the runtime performance as well as the segmentation quality because a richer (region-based) feature set can be utilized. It has been shown in [11] that the benefits can be further amplified by learning the graph weights between the superpixels. But quite often the superpixel algorithms used for video applications like [12], [13], [14], [15], [16], [17], [18] only target single images. When applied to video sequences, the results show volatile and flickering superpixel contours even if there are only slight changes between consecutive frames. Moreover, by design the temporal connections between superpixels in successive video frames are not determined. Consequently, the same image regions in consecutive frames are not consistently labeled. The benefits of a consistent labeling has e.g., been shown in [19] for the special case of interactive video segmentation. Similarly, it was observed in [11] that the selection of the graph structure is crucial for a good segmentation result. This can either be accomplished by merging the independently calculated superpixels in the temporal dimension, as e.g., done in [20], or by directly creating a temporally consistent oversegmentation.

In a temporally consistent oversegmentation, each segment follows the underlying image patch when it moves over time as can be seen in Fig. 1. The example also illustrates that natural scenes, which involve moving objects or camera motion, in general include some form of occlusion or disocclusion of image regions. For a segmentation that can handle structural scene changes, its segments should disappear as soon as the corresponding image patch becomes occluded and new segments should be created where disocclusion happens. Previous approaches like [21], [22], [23] and [24] produce a temporally consistent superpixel segmentation but due to a lack of an explicit awareness of occlusion and disocclusion boundaries, segments are deleted and created in a rather randomized fashion.

This work is an extended version of two previously published articles ([23] and [24]) that introduce a hybrid

- *M. Reso, B. Rosenhahn, and J. Ostermann are with the Insitut für Informationsverarbeitung, Gottfried Wilhelm Leibniz Universität Hannover, Hannover 30167, Germany.*
  E-mail: {reso, rosenhahn, ostermann}@tnt.uni-hannover.de.
- *J. Jachalsky is with Technicolor Research & Innovation, Hannover 30657, Germany. E-mail: jachalsky@tnt.uni-hannover.de.*

Manuscript received 6 Aug. 2016; revised 24 Feb. 2018; accepted 3 Apr. 2018.
Date of publication 2 May 2018; date of current version 14 May 2019.
(Corresponding author: Matthias Reso.)
Recommended for acceptance by J. J Corso.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2832628

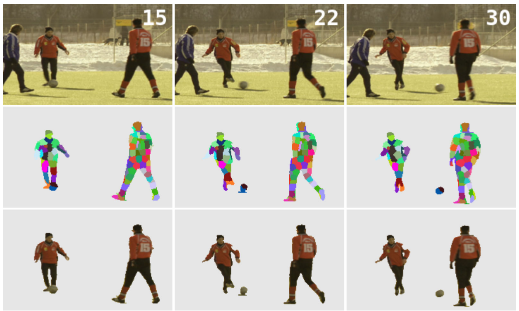

Fig. 1. Top row: Original sequence with frame numbers. Mid row: Subset of superpixels manually selected in frame 15 and shown as color-coded labels. The superpixels in the frames 22 and 30 are generated with our approach and are displayed using the same label colors to indicate temporal consistency. Bottom row: The soccer players are cut out based on the selected superpixels. Best viewed in color.

clustering scheme for temporally consistent superpixels for video content. Besides a more detailed description of our approach, the novel *key contributions* of this paper are:

- We propose a new segmentation propagation method to initialize frames encouraging consistent superpixel shape and relative positioning.
- We introduce an approach to detect occluded superpixel parts as well as disoccluded image regions while propagating the superpixels onto new frames.
- The dis-/occlusion information is used to explicitly handle structural changes in the video volume which are induced by object- and self-occlusion.
- Additionally, we consider the hidden superpixel parts during the optimization in order to increase the consistency of the superpixel flow.
- Finally, we evaluate our newly proposed approach and compare it against state-of-the-art streaming-capable methods for video oversegmentation (as well as our previous work) by using well established benchmarks.
- In addition, we compare the approaches by utilizing them for an interactive video segmentation task in order to show the superiority of our proposed approach in terms of segmentation quality in a common application.

The remainder of this work is organized as follows: In Section 2, we shortly summarize previous works on spatio-temporal oversegmentation and segmentation propagation. Subsequently, in Section 3, we revisit the generation of superpixels using energy-minimizing contour-evolution which is extended in Section 4 to the generation of temporally consistent superpixels. In the Sections 4.1, 4.2, and 4.3, we introduce the basic ideas of our framework for temporally consistent superpixel. Section 4.4 introduces the new method to propagate superpixel contours onto new frames for initialization and additionally shows how structural changes in the video volume are handled. Section 5 contains the detailed evaluation of our approach and a comparison to other state-of-the-art video oversegmentation approaches. Finally, we conclude our paper in Section 6.

2 Related Work

In general, all related approaches can be classified as either generating superpixels with temporal consistency (e.g., [21], [22], [25]) or supervoxels (e.g., [14], [15], [26]). The relation between supervoxels and temporal superpixels can be described in the following way: Temporal superpixels can be stacked up to build supervoxels. Similarly, a superpixel representation with temporal consistency can be obtained by slicing a supervoxel representation at frame instances. It should be noted that this does not hold in the case where the cross section of a supervoxel at a frame instance splits up into spatially non-contiguous segments. In the following, we will give a brief overview of available supervoxel and temporal superpixel algorithms. An early example of this kind of algorithms, which is not explicitly labeled as a superpixel or supervoxel approach but shares a similar idea, can be found in [27]. A more extensive survey of a number of temporally superpixel and supervoxel approaches as well as benchmark metrics for their comparison can be found in [28].

In [15], a first supervoxel approach was published that covers the video volume with overlapping cuboids, whereas each cuboid corresponds to one label in the final segmentation. The volume of each cuboid determines the maximum volume of the supervoxel to be generated. Thus, longer cuboids encourage higher temporal consistency. The assignment of each voxel to one label is done by formulating an energy function incorporating image gradients and minimizing this energy function using graph cut.

The authors of [26] proposed an approach for hierarchical video segmentation that is based on the graph-based image segmentation method introduced in [29]. To leverage the information of color histograms, the optimization procedure is applied twice. In an initial run neighboring voxels are merged into small voxel groups from which color histograms can be computed. Based on the chi-square distances between their color histograms the voxel groups are further merged into larger spatio-temporal regions. By keeping track of the mergers, a hierarchical video segmentation is created. As the original approach of [26] requires access to the whole video during the computation, it was extended in [30] to provide streaming capabilities. By applying the Markovian assumption to the segmentation of overlapping chunks of the video stream, only a subset of frames is needed during the segmentation process.

The clustering-based approach for superpixels of [14] can be extended to a supervoxel algorithm by extending the data points with a temporal dimension. Thereby, each voxel is viewed as a data point in a six-dimensional feature space consisting of three color, two spatial and one temporal dimension. The supervoxels form clusters of data points in the feature space and they are represented by the mean vector of the assigned voxels. To estimate the parameters of the clusters, an iterative expectation-maximization framework is used. The distance of a voxel to a cluster center is expressed by using a weighted norm to encounter the different scales of the original dimensions[14].

A first approach towards temporal superpixels was introduced in [25]. The approach is based on the TurboPixel algorithm proposed in [13] which uses level-set techniques to grow equally distributed seed points to non-overlapping superpixels. To derive a temporally consistent segmentation, it was proposed in [25] to propagate the central point of each superpixel using optical flow information in order to initialize the seeds for the superpixels in each new frame.

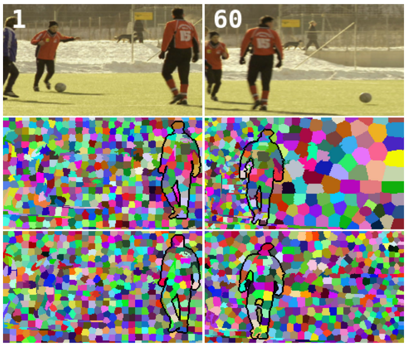

Fig. 2. Top row: Frame 1 and 60 of the soccer sequence. Second row: Label maps with temporal consistency but without a method to cope with structural changes in the video volume. The superpixels in the later part of the sequence are squeezed together on the left side of the frame, while on the right side the size of the superpixels has to grow to fully occupy the newly uncovered image regions. Third row: Label maps created with our approach. The superpixels are temporally consistent and have an equal size over the whole sequence. The silhouette of the player was marked for visualization purposes. Best viewed in color.

Using these seeds, the superpixels are then grown on frame level. A similar propagation approach is applied in [31] in which the superpixels on each frame are created using partially absorbing random walks. To improve the robustness of the propagation, only reliable optical flow vectors are utilized. While achieving a more temporally stable superpixel segmentation and establishing temporal connections between superpixels in adjacent frames, both approaches do not handle structural changes like occlusion and disocclusions. This leads to superpixels of in-homogeneous size in long video sequences similar to the effect shown in Fig. 2.

The problem of structural changes in the video volume was addressed first in the works of [21], [22], and [23] by providing a strategy for the creation or splitting as well as the termination of superpixels.

In [21], a generative probabilistic framework is proposed to model the segmentation of each frame. The inference is done on frame level by proposing label changes and accepting them only if they increase the log-likelihood function of the new segmentation given the observed pixel data. The superpixel movement from frame to frame is modeled by a Gaussian process initialized using optical flow. To address the problem of structural changes the authors propose split, merge and switch moves in which superpixels can be split up into two, merged together or take the label of a superpixel that was previously merged into them, respectively. A set of proposed moves is only accepted if the new resulting segmentation increases the joint log-likelihood function.

The work of [22] is an extension of the superpixel approach from [32] to video segmentation. It uses color histograms to represent superpixels and sets up an objective function which is maximized if the number of populated bins per histogram is minimized. The proposed hill-climbing algorithm optimizes the segmentation by proposing the reallocation of single pixels or pixel blocks from one superpixel to an adjacent one. Changes are accepted if they maximize the objective function. Influenced by a parameter called superpixel rate some frames are selected for termination and splitting of superpixels[22]. To keep the number of superpixels constant over time, a new superpixel is created for every terminated superpixel by splitting off a part from another superpixel. The decision for termination and splitting is based on the lowest impact on the objective function.

Our previous works published in [23] and [24] introduce a hybrid clustering approach which separates the five-dimensional cluster centers of [14] into local spatial centers and global color centers. The details of this approach will be further described in the first part of Section 4. In [23] the segmentation propagation was done by propagating the spatial centers using forward optical flow. In [24] it was proposed to look-up the superpixel label in the previous frame using pixel-wise, backward-directed optical flow. The latter approach produces a more stable segmentation result as it propagates the superpixels' relative positioning (in the following also described as their constellation) as well as their shape. To handle structural changes in the video volume both approaches rely on the number of pixels each superpixel comprises. While the former predicts the positive and negative growth using a linear assumption, the latter sets minimal and maximal thresholds to identify the superpixels that need to be terminated or split.

While the approaches proposed in [21], [22], [23], and [24] avoid effects as seen in Fig. 2, the decision to terminate a superpixel is solely based on the objective function or the superpixel size. As a result, terminations of superpixels often do not coincide with the actual occlusion boundaries present in the video scene. But instead, they happen at rather random spots in the scene. This misalignment comes from the fact that none of the termination conditions utilized are specific to the area surrounding an occlusion boundary.

We therefore propose a new approach to handle structural changes which explicitly detects occlusion and disocclusion boundaries during the superpixel propagation onto new frames. By classifying the overlapping parts of the propagated superpixels as either occluded or occluding we gain knowledge of where the the actual occlusion boundaries lie. This enables the termination of occluded superpixels. Additionally, it is revealed which superpixel is partially occluded. This knowledge is used during further optimization of the segmentation to improve the consistency of the superpixel flow with the underlying video scene. Our new method integrates seamlessly into our previously published approaches for temporally consistent superpixels [23] and [24].

3 Superpixels Based on Energy-Minimizing Contour-Evolution

Our method for temporally consistent superpixels is based on the superpixel approach described in [18]. In contrast to the popular clustering-based method of [14], the contour-evolving approach of [18] does not need a post-processing step to ensure the spatial coherency of the resulting superpixels. In this Section, we will briefly revisit the basics of [18].

The problem of a superpixel segmentation can be formulated as a label assignment problem where each pixel n of an image I is assigned a label l. The labels come from the discrete set of superpixel labels $\mathcal{L}$. We can evaluate a particular labeling L by computing its total energy cost $E_{total}(L)$ as follows:

$$E_{total}(L) = \sum_{n \in \mathcal{N}_I} E_n(l_n)\,. \tag{1}$$

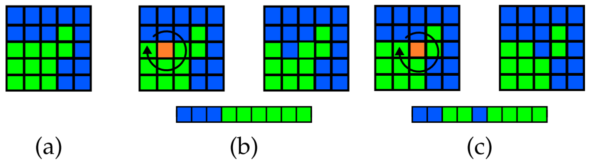

Fig. 3. The three subfigures exemplarily show pixels at the border between two superpixels (green and blue). In (b) and (c), a pixel is marked (orange) and it is examined if a label change (to blue) would violate the spatial coherence constraint. The eight neighboring pixels (marked by the circling arrow) are plotted as an array on the bottom of each subfigure. While the label in the array of (b) only changes once and thus a change is valid, the label in (c) changes a second time from blue to green which indicates an invalid change. Best viewed in color.

Here, $\mathcal{N}_I$ is the set of pixels in image I and l_n is the label currently assigned to pixel n. $E_n(l_n)$ denotes the energy needed to assign the label l_n to pixel n and it is defined by [18] as

$$E_n(l_n) = (1-\alpha)E_{c,n}(l_n) + \alpha E_{s,n}(l_n)\,. \tag{2}$$

The energies $E_{c,n}(l_n)$ and $E_{s,n}(l_n)$ are related to the likeliness of the pixel n belonging to the superpixel with label l_n. The weighting parameter α is a user-selected trade-off factor steering the superpixel compactness opposed to the sensitivity to fine-grained image structures. The authors of [14] and [18] chose to model each superpixel by its mean color value and spatial center. The energy $E_{c,n}(l_n)$ is chosen to be proportional to the euclidean distance between the pixel's color value and the superpixel's average color $\bar{\mu}_{c,l}$. Equally, $E_{s,n}(l_n)$ is selected to be proportional to the pixel's spatial position and the spatial center of the superpixel $\bar{\mu}_{s,l}$. In contrast to [18], this work uses the CIELAB color space to perform the color distance calculations as it was proposed in [14]. In order to make the results independent from the image resolution as well as the selected number of superpixels, the spatial distance is scaled with the factor $\sqrt{|\mathcal{L}|/|\mathcal{N}_I|}$ where $|\cdot|$ is the number of elements in a set.

To find an optimal superpixel segmentation given this model, one has to find the labeling and the corresponding superpixel parameter $(\bar{\mu}_{c,l}, \bar{\mu}_{s,l})$ which minimize the total energy (1). This can be done using an expectation-maximization approach where the superpixel labeling and their parameters are estimated in two separate steps. These steps are then alternately run in an iterative matter to approach a local minimum. In the *expectation*-step of iteration j the optimal assignment $\hat{L}^j$ of the pixels to the superpixels given the parameters from iteration $j-1$ is determined. This is done by assigning each pixel to the label which minimizes the energy term (2)

$$\hat{L}_n^j = \operatorname*{arg\,min}_{l_m,\, m\in(\mathcal{N}_n^4 \cup n)} E_m(l_m)\ \forall\, n \in \mathcal{N}_{\mathcal{C}}^j\,. \tag{3}$$

Here, as proposed in [18] only the pixels which reside on a boundary between superpixels (denoted as $\mathcal{N}_{\mathcal{C}}^j$) are considered for a label change. A label change is only allowed to a label of a pixel which is part of the 4-connected neighborhood around the pixel n. These are denoted with $\mathcal{N}_n^4$.

After assigning each contour pixel to the best matching neighboring superpixel, their parameters are recomputed from their assigned pixels in the *maximization*-step. The iteration stops when no assignments to a different label take place or a maximum number of iterations has been performed.

To ensure that the pixels of each superpixel are still spatially connected through a 4-connected neighborhood, a simple check is performed before a label change is executed. As illustrated in Fig. 3, the check looks at the labels in the 8-connected neighborhood around the pixel to be examined as if they are lined up in an array. While traversing the array, label changes are detected. After each label change, the check looks up if the label has been seen before. In this case, the check can only exit successfully if the array can be further traversed without an additional label change. Otherwise, assigning a different label to this pixel would result in a split of the superpixel. Therefore, a label change is not permitted in this case. Examples for both cases are shown in Figs. 3b and 3c. A similar approach to verify if a label change breaks the spatial coherency is described in [33].

Before the first assignment step, the labeling and thus the superpixel parameters have to be initialized. This can be done using an initial grid-like or honeycomb-like superpixel configuration.

4 Temporally Consistent Superpixels

The Sections 4.1, 4.2, and 4.3 revisit and explain in more detail our approach for temporally consistent superpixels previously published in [23] and [24]. Our new approach for segmentation propagation and the handling of structural changes in the video volume is then described in Section 4.4.

4.1 General Idea

Our approach is motivated by the observation that the color of matching image regions in consecutive frames do not change rapidly in most cases. In a temporal consistent superpixel segmentation these matching regions would be occupied by a single superpixel over multiple frames. Therefore, the mean color of the associated superpixel is –in a first approximation-almost constant over time. In contrast to that, the positions can vary significantly, depending on the motion which is present in the scene.

To enable the generation of temporally consistent superpixels, the color and spatial models of the superpixels are separated into a global color model, comprising multiple frames, and multiple local spatial models on frame level. Thus, following the idea that the color is globally valid while the spatial position is only locally valid. As a consequence, each temporally consistent superpixel is modeled by using a single color mean value for all frames and a separate spatial center for each frame. The latter preserves the spatial locality on frame level and the former ensures temporal consistency.

In order to allow for a certain degree of scene changes, e.g., gradual changes of illumination or color over time, we introduce a sliding window approach. For this, a window comprising W consecutive frames is shifted along the video volume frame by frame. This sliding window contains P so called *past* frames, F *future* frames as well as one *current* frame with $W=F+P+1$. An example with $W=5$ and $P=F=2$ is depicted in Fig. 4. In this example, the frame k is the *current* frame and it is in the center of the sliding window.

For the *current* frame, the resulting, final superpixel segmentation is generated. The segmentation of the *past* frames is immutable and thus will not be altered anymore. But through the global color model it influences the segmentation in the

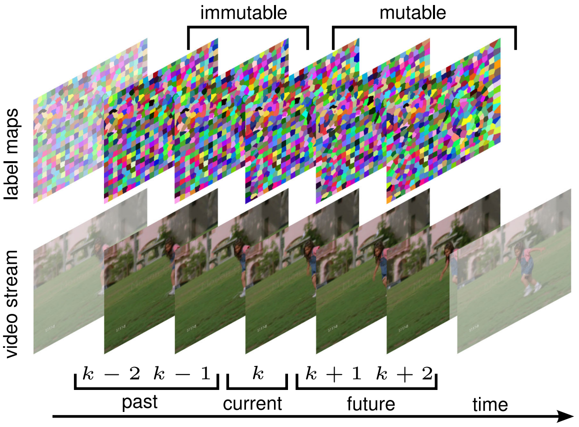

Fig. 4. Sliding window approach. Bottom row: Frames inside the sliding window (non-transparent) are divided into three groups. Top row: Corresponding label maps.

current and *future* frames as these are still mutable and thus can change during the optimization. The *future* frames help to adapt to changes in the scene. The *past* frames are conservative and try to preserve the superpixels color value over time. If more *past* than *future* frames are used, the update of the color centers is more conservative. If more *future* than *past* frames are used, the update is more adaptive. As the optimization procedure utilizes global color models for the whole sliding window and local spatial models for each frame we call it a hybrid optimization which will be explained in detail below.

4.2 Hybrid Optimization

The energy function (1) and the energy term (2) as well as the iterative optimization algorithm explained in Section 3 have to be extended to the idea of the global color and local spatial models. First, we extend the energy term (2) with the frame index k as the energy $E_{s,n}$ is now proportional to the distance to the spatial centers in the specific frame

$$E_n(l_n, k) = (1-\alpha) E_{c,n}(l_n) + \alpha E_{s,n}(l_n, k) \,. \tag{4}$$

Second, we need to sum up the energies of all pixels in all frames inside the sliding window to calculate the total energy with regard to the *current* frame k

$$E_{total}(L_k^W) = \sum_{\kappa=k-P}^{k+F} \sum_{n \in \mathcal{N}_\kappa} E_n(l_n, \kappa) \,, \tag{5}$$

where $\mathcal{N}_\kappa$ is the set of pixels in the frame κ and L_κ^W denotes the labeling of all pixels in all frames inside the sliding window around the current frame k.

Third, the iterative optimization scheme is adopted to the hybrid approach as explained below and is summarized in Algorithm 1.

After each shift of the sliding window, a number of J iterations of the hybrid optimization algorithm is performed. In the *expectation*-step, the contour pixels of the mutable frames, i.e., the *current* and the *future* frames, are reassigned to the best matching neighboring superpixel to minimize the energy term (4). The color-energy $E_{c,n}$ is proportional to the euclidean distance to the global color mean value of the superpixel $\bar{\mu}_{c,l}$. The spatial-energy $E_{s,n}$ is proportional to the euclidean distance to the spatial center $\bar{\mu}_{s,l,\kappa}$ in frame κ.

Algorithm 1. Hybrid Optimization of the Segmentation Inside a Sliding Window Positioned Around the Current Frame with Index k. $L_k^{W,j}$ Denote the Labeling of all Pixels Currently Inside the Sliding Window at Iteration j

```
Input: $W$ frames in sliding window around $k$; initial labeling $L_k^{W,0}$
Output: updated labeling $L_k^{W,J}$
determine parameters of color and spatial models for given $L_k^{W,0}$;
for $j \in [1, J]$ do
  foreach mutable frame $\kappa$ in sliding window do
    reassign contour pixels $\mathcal{N}_{\mathcal{C},\kappa}^j$ according to Eq. (3)
    given the model parameters of $j-1$;
  for all frames $\kappa$ in sliding window do
    if $\kappa$ is mutable frame then
      update local spatial models in $\kappa$;
    end
    accumulat e global color information;
  end
  update global color models from accumulated information;
end
```

In the *maximization*-step, the parameters of the global color model for each superpixel are updated using the accumulated color information of all pixels in all frames inside the sliding window

$$\bar{\mu}_{c,l} = \frac{1}{\sum_{\kappa \in W} |\mathcal{N}_{l,\kappa}|} \sum_{\kappa \in W} \sum_{n \in \mathcal{N}_{l,\kappa}} [l,a,b]_{n,\kappa}^T \,. \tag{6}$$

Where $\mathcal{N}_{l,\kappa}$ is the set of pixels assigned to the superpixel with label l in frame κ of the sliding window. $[l,a,b]_{n,\kappa}^T$ is the transposed color vector of the pixel n in frame κ.

The spatial models are updated locally per frame using only the image coordinates of the pixels that are assigned to this superpixel in the corresponding frame

$$\bar{\mu}_{s,l,\kappa} = \frac{1}{|\mathcal{N}_{l,\kappa}|} \sum_{n \in \mathcal{N}_{l,\kappa}} [x,y]_n^T \,. \tag{7}$$

Here, $[x,y]_n^T$ is the transposed spatial position vector of pixel n. For our experiments we use $J=5$ iterations after each shift of the sliding window. During our evaluation it turned out that the gain using a higher number of iterations is negligible.

4.3 Initialization of Sliding Window

Because the position of corresponding image regions and thus the superpixel position can differ in consecutive frames, a concurrent initialization of all frames of the sliding window is not practicable. Therefore, we propose a successive filling of the sliding window according to the scheme visualized in Fig. 5. During the initialization of the sliding window as well as afterwards, frames are added to the sliding window. The segmentation of these added frames have to be initialized as well. To better distinguish between the two types of initialization, we refer to the frame initialization as segmentation propagation. The details to this propagation step can be found in the Section 4.4.

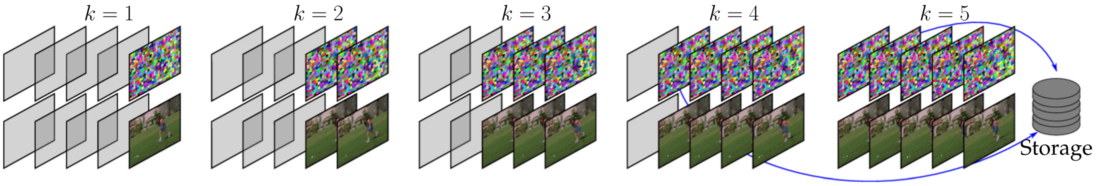

Fig. 5. The sliding window is initialized from the front to the back positions. After J iterations (cf. Algorithm 1), the sliding window is shifted and the segmentation of the *current* frame moves into the area of the past frames. As these segmentations are not altered anymore, they can be stored to disk which reduces the delay of the algorithm to $F+1$ frames.

At the start, the sliding window is empty. The segmentation of the first frame of a video sequence to enter the sliding window cannot be propagated from former frames. Therefore, it is initialized by a regular grid- or a honey-comb-like arrangement of superpixels as proposed in [18]. This frame is positioned at index $k+F$ in the sliding window. As a *future* frame its segmentation is mutable. Therefore, an energy-minimization with regard to Eq. (4) is performed. Then, the sliding window is shifted, whereby a new frame enters the window at position $k+F$. The old frame is moved to $k+F-1$. The initial segmentation of frame $k+F$ is created by propagating the current segmentation of frame $k+F-1$. The propagation procedure only roughly adapts the segmentation to the new frame. Therefore, after each propagation step several optimization iterations as described in Section 4.2 are performed to fit the superpixel boundaries to the frame content.

This procedure is repeated until all positions in the sliding window are occupied. Then the generation of the temporally consistent superpixels can further proceed by repeatedly shifting the sliding window by one frame as described above until the video sequence is completely processed. After each shift the superpixel segmentation of frame $k-1$ of the sliding window is stored, which is the first *past* frame and thus immutable.

4.4 Segmentation Propagation and Handling of Structural Changes

After a new frame has entered the sliding window the latest segmentation of the previous frame needs to be propagated onto the new frame. As image regions can move significantly from frame to frame, a simple copy of the previous segmentation as described in [22] can be error-prone in many situations. This is especially the case in videos with large object motion or camera movement. Therefore, the segmentation needs to be warped to roughly fit the content of the new frame.

In [25] and [23], the weighted average optical flow is used to propagate superpixel seed-points or their spatial centers onto new frames. The weighting function gives optical flow vectors near the superpixel center more weight than vectors at the superpixel boundary. These boundary vectors tend to be more noisy and inaccurate when the superpixel boundary coincides with an object boundary, as the smoothness assumption, which is part of most optical flow algorithms, does not hold at these locations when the motion direction of the object differs from the motion direction of the background.

While using the averaged optical flow to project seed points or spatial centers gives a certain degree of robustness against noise and inaccuracies, it results in a complete loss of the superpixel shape information. This can lead to a higher volatility in the superpixels boundaries and less stable spatial constellations over time as it has been shown in [24]. Therefore, [24] proposed to use a pixel-wise, backward-directed optical flow to propagate the segmentation. By looking-up the label for each pixel in the previously segmented frame, using the backward-directed flow vectors, shape and constellation information are preserved.

But as the reliance on a pixel-wise optical flow can be error-prone when the noise level increases, we propose to use the weighted averaged optical flow to propagate the complete superpixel shapes in a forward directed manner. This approach leverages the robustness of the averaged optical flow while it concurrently exploits the constellation and shape preserving behavior of a pixel-wise propagation. To propagate the segmentation, we shift each superpixel by the weighted average optical flow vector. The weighted average optical flow vectors are gained by applying a symmetric, two-dimensional Gaussian function on the flow vectors of each superpixel and averaging over the weighted vectors. Each weighting kernel is centered around the spatial center of the superpixel. We select the standard deviation σ_w of the weighting function to be $\sqrt{|\mathcal{N}_I|/4 \cdot |\mathcal{L}|}$ which corresponds to the radius of the average sized superpixel in the frame.

If the superpixels follow their underlying image regions over time, the structural changes inherent in the video volume lead to the squeezing and expanding effects shown in Fig. 2. To avoid this effect and to satisfy the homogeneous superpixel size constraint, the superpixels whose corresponding image regions become occluded have to be terminated. Simultaneously, new superpixels have to be created when new image regions become disoccluded.

Due to the nature of the proposed forward propagation, the locations of occluded and disoccluded image regions can be directly extracted from the propagated superpixel labeling. It can be seen in Fig. 6 that the detection of gaps and thus disocclusion in the propagated superpixel labeling is trivial. For the occlusion case we need to determine which of the superpixels is the occluding one.

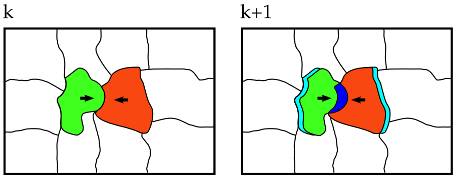

Fig. 6. Schematic example of two adjacent superpixels being propagated towards each other by optical flow. The overlapping areas (dark blue) can yield essential information about the structure of the video as it indicates the occlusion of an image region. Similarly, a gap in the propagated superpixel labeling indicates a disoccluded region (light blue). Best viewed in color.

We therefore propagate the superpixels successively in an arbitrary order onto the new frame. Simultaneously, the positions of any overlapping regions and the involved superpixels are recorded. Afterwards, we determine for the individual pixels which is the topmost superpixel at this location. This is done by finding the optimal labeling of the pixels given the image data and the superpixel color models. To avoid any interference by the compactness and homogeneous size constraints of the superpixels, we do not employ the same optimization strategy as described above. Instead, we represent the overlapping regions as a graph structure and apply the graph-cut optimization algorithm [34] to solve the multi-label assignment problem. This procedure is inspired by [15] where superpixels are created by laying out overlapping patches on the image.

In the graph, each vertex represents a pixel and can be assigned a label from the discrete set of overlapping superpixel labels $\mathcal{L}_{OL}$. The edges between vertices indicate neighboring pixels. As the optimal labeling of the overlapping areas is determined for each propagated frame independently, we will skip the frame index in this passage. The quality of every possible labeling of the graph $L_{OL} = \{l_n | l_n \in \mathcal{L}_{OL}, n \in \mathcal{N}_{OL}\}$ can then be assessed by an energy function we define as follows:

$$E_{OL}(L_{OL}) = \sum_{n \in \mathcal{N}_{OL}} D_n(l_n) + \gamma \sum_{\substack{n \in \mathcal{N}_{OL} \\ m \in \mathcal{N}_n^4}} V_{n,m}(l_n, l_m)\,. \tag{8}$$

Here, $\mathcal{N}_{OL}$ is the set of pixels which are part of the overlapping area. The decision about the topmost superpixel should depend on the similarity of the image data to the appearance models of the possible superpixels. We therefore define the unary term as

$$D_n(l) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_{C,l}^2}} \exp\left(-\frac{E_{c,n}^2}{2\sigma_{C,l}^2}\right), & \text{if } l \in \mathcal{L}_{OL,n} \\ \infty, & \text{else.} \end{cases} \tag{9}$$

With $\mathcal{L}_{OL,n}$ denoting the set of superpixel labels which overlap at the pixel n and $\sigma_{C,l}^2$ denoting the color variance of the superpixel l. As the unary term only includes the color depending energy of Eq. (4), there is no interference of the superpixel compactness constraint enforced through E_s. To still favor equally labeled neighbors and thus a spatially coherent labeling, we select the pairwise term $V_{n,m}$ to be

$$V_{n,m}(l_n, l_m) = \begin{cases} \exp\left(-\frac{\left|[l,a,b]_n^T - [l,a,b]_m^T\right|^2}{2\sigma_C^2}\right), & \text{if } l_n \neq l_m \\ 0, & \text{else} \end{cases} \tag{10}$$

where σ_C^2 is the variance of all color differences in the overlapping region. For our experiments we used the implementation of [35] and performed two alpha expansion iterations to find the optimal labeling.

Although the pairwise term (10) favors label consistency, it is not guaranteed that the resulting superpixels are spatially coherent. For those rather rare cases, we determine the largest fragment of each superpixel and set the other smaller fragments to an invalid label. These regions are then handled in the contour evolution procedure described in Section 4.2 by automatically assigning them to a directly connected valid neighbor.

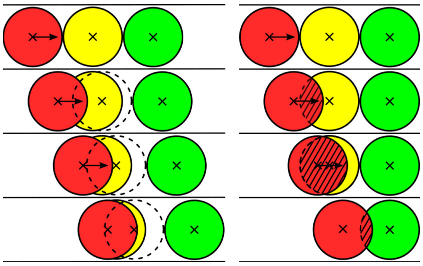

Fig. 7. The left superpixel (red) is propagated (indicated by the arrow) towards two stationary superpixels (yellow, green). Without the knowledge that the yellow superpixel becomes occluded by the red superpixel (left column) its newly calculated spatial center will have an offset compared to the propagated center. The offset of the spatial center will lead to a shift of the yellow superpixel as the spatial term of Eq. (2) enforces the homogeneous size constraint (indicated by the dotted circle). This effect is further propagated and results in a shift of the neighboring (green) superpixel. By utilizing the occlusion information to compute the "true" spatial center (right column) the drift can be avoided resulting in a more accurate superpixel movement.

Given the final labeling of the overlapping regions, we store for each superpixel the pixels which are not part of the final labeling. This results in a set of occluded pixels for each superpixel and frame denoted by $\mathcal{N}_{OC,l,\kappa}$. To make the decision on which superpixel to terminate, we observe the number of occluded pixels per superpixel over time and check after each propagation if the major part of the superpixel l got occluded

$$\sum_{\kappa \in W} |\mathcal{N}_{OC,l,\kappa}| > |\mathcal{N}_{l,P+F+1}|\,. \tag{11}$$

$|\mathcal{N}_{l,P+F+1}|$ denotes the number of pixels assigned to superpixel l in the last frame of the sliding window. To terminate a superpixel, it is removed from all future frames (compare Fig. 4) by assigning an invalid label to its pixels in these frames. This also excludes such a superpixel from further propagation steps.

To keep the number of superpixels constant over time, we split up as many superpixels as were terminated before. In order to increase the compliance of the superpixel flow with the optical flow present in the scene, the superpixels in the surrounding of the gaps in the propagated superpixel labeling are preferred. The splitting is done similar as described in [17].

In addition to the squeezed and expanded superpixels, Fig. 2 illustrates a second effect which can be described as superpixels being pushed aside by other superpixels. The effect and its cause are schematically depicted in the left column of Fig. 7, where the green and yellow superpixel are gradually pushed to the right although the magnitude of the optical flow inside these superpixels is virtually zero in all frames.

The reason for this effect lies in the homogeneous size constraint enforced during the optimization procedure which is performed after each propagation step. When the spatial centers (indicated by crosses) are recalculated, the center of the middle superpixel will be far more right than its initially predicted spatial center. During the optimization, the middle superpixel will therefore regain size because the spatial distance term in Eq. (2) favors equally sized superpixels (indicated by the dotted outline). This in return will lead to a right-shift of the right superpixel. This observation is similar to observations made in [21] about the superpixel flow at image boundaries.

In order to prevent this false superpixel flow, we propose to utilize the knowledge about the occluded superpixel fractions. To stop the centers from shifting back, the hidden parts of the superpixels are integrated during the recalculation of the spatial center in Eq. (7) as follows:

$$\bar{\mu}_{s,l,\kappa} = \frac{1}{|\mathcal{N}_A|}\left(\sum_{n\in\mathcal{N}_{l,\kappa}} [x,y]_n^T + \sum_{n\in\mathcal{N}_{OC,l,\kappa}} [x,y]_n^T\right). \tag{12}$$

Here, $|\mathcal{N}_A|$ is a substitute for $|\mathcal{N}_{l,\kappa}| + |\mathcal{N}_{OC,l,\kappa}|$. The principle is illustrated in the right column of Fig. 7.

As the complete occlusion of a superpixel in general occurs over the term of multiple frames, the hidden part of a superpixel also needs to be propagated when a new frame is initialized. We therefore propagate the occluded fraction of a superpixel, by shifting it with the same displacement vector $[u,v]_l^T$ used for the visible part. Subsequently, we merge the propagated, hidden fraction of the superpixel with any pixels that get newly occluded in the current propagation step. Thus, the set of occluded pixels of a superpixel l in a frames κ becomes

$$\mathcal{N}'_{OC,l,\kappa} = \mathcal{N}_{OC,l,\kappa} \cup \left\{[x,y]_n^T + [u,v]_l^T \mid \forall\, n \in \mathcal{N}_{OC,l,\kappa-1}\right\}. \tag{13}$$

5 Experiments

In this Section, we will compare our method to the state-of-the-art, streaming-capable temporally consistent superpixel and supervoxel methods. First, we will describe the benchmark metrics used during the evaluation. Subsequently, the experimental setup and the derivation of the parameters for the proposed method will be described. Finally, we will present and discuss the results.

5.1 Benchmark Metrics

Recently, the computer vision community actively contributed to the field of video segmentation benchmarks as e.g., in [36]. While these benchmarks are often especially targeted at video object segmentation, there has been a lot of work on metrics especially tailored to evaluate temporal superpixel and supervoxel segmentations. In order to account for the special requirements for the evaluation of a video oversegmentation, this paper closely follows the protocol and metrics utilized in [21] and [28] to evaluate temporally consistent superpixels and supervoxel methods. To assess the video segmentation quality, five metrics are used which are tailored to the evaluation of supervoxel and video segmentation algorithms and indicate the quality of the spatio-temporal segmentation. As the quality of the spatio-temporal segmentation is as important as the quality of the segmentation on frame level, an additional set of three benchmark metrics suitable for evaluating the image segmentation quality on frame level are included. All benchmark metrics will be revisited briefly in the following. For a more thorough explanation please refer to [12], [16], [21], and [28].

3D Undersegmentation Error (3D UE). This metric was first proposed by [37]. It counts the number of voxels *bleeding out* of the ground truth segmentation volume. Given a segmentation with non-overlapping segments $s_1, s_2, \ldots, s_M$ and a ground truth segment g_n, the 3D undersegmentation error is calculated as follows:

$$\mathrm{UE}(g_n) = \frac{\left[\sum_{(s_m|s_m\cap g_n\neq\emptyset)} |s_m|\right] - |g_n|}{|g_n|}. \tag{14}$$

Here, $|s_m|$ denotes the number of voxels of the segment. The error is then averaged over all ground truth segments.

3D Segmentation Accuracy (3D SA). Also proposed [37], this metric denotes the fraction of the video volume that can be correctly reproduced by a given segmentation. To calculate the metrics, each segment is first assigned to the ground truth segment with which it has the maximum overlap. Subsequently, the overlap of the segments with the assigned ground truth segments is counted and divided by the size of the whole video volume

$$\mathrm{SA} = \frac{1}{N}\sum_{n=1}^{N} \frac{\sum_{o\in O_n}(|s_o \cap g_n|)}{|g_n|}, \tag{15}$$

where N is the number of ground truth segments and O_n is the set of segments s_o assigned to g_n.

Temporal Extent (TEX): This metric was introduced in [21] for measuring the ability to track regions over time by calculating the mean duration of the spatio-temporal segments. By evaluating this metric in conjunction with the 3D segmentation accuracy or the 3D undersegmentation error, it provides a suitable measure to judge how much of the temporal consistency inherent in the video volume has been revealed by a supervoxel segmentation. The combination with another metrics, penalizing the erroneous crossing of object boundaries, is necessary as a long temporal segment duration is only valuable together with a high quality spatio-temporal segmentation. It should be noted that the metrics described above only indicate if a superpixel crossed any object boundaries, as defined by the ground truth segmentation. However, the consistency of their shape or their relative positioning inside the boundaries of an object are completely ignored. In order to measure this type of temporal consistency, the label consistency metric as proposed in [21] is utilized during the evaluation as well.

Label Consistency: This metric measures the consistency of the superpixel flow with the underlying image movements and penalizes any temporal inconsistency in the shape as well in the constellation of the superpixels. It utilizes ground truth optical flow information to propagate the superpixel labeling of a segmented frame onto the next frame and determines the number of pixels that agree between the propagated labeling and the segmentation generated by the algorithms. The label consistency is given as the ratio between the number of pixels which agree and the total number of pixels per frame averaged over all frames.

Explained Variation (EV). This metric was proposed in [12] for the evaluation of superpixel segmentations. It indicates how well the original image content can be represented with a given oversegmentation as a representation of lower detail. Its extension to the video domain, first proposed in [37], can be calculated as follows:

$$\mathrm{EV} = \frac{\sum_n (\bar{\mu}_{c,n} - \bar{\mu}_c)^\top (\bar{\mu}_{c,n} - \bar{\mu}_c)}{\sum_n (\bar{x}_{c,n} - \bar{\mu}_c)^\top (\bar{x}_{c,n} - \bar{\mu}_c)}. \tag{16}$$

Here, $\vec{\mu}_c$ denotes the global mean color vector and $\vec{x}_{c,n}$ is the color vector at the voxel position n. The vector $\vec{\mu}_{c,n}$ is the mean color vector inside the segment that the voxel n is assigned to.

Boundary Recall Distance (BRD). The boundary recall distance was proposed in [21] and measures the average distance to the next boundary present in the ground truth segmentation. In contrast to the popular 2D boundary recall, the boundary recall distance does not require the selection of a fixed threshold by the user. For each frame k it can be independently calculated as follows:

$$\mathrm{BRD}(k) = \frac{1}{|\mathcal{N}_{\mathcal{C},\mathrm{gt},k}|} \sum_{i \in \mathcal{N}_{\mathcal{C},\mathrm{gt},k}} \min_{j \in \mathcal{N}_{\mathcal{C},\mathrm{seg},k}} d(i,j). \tag{17}$$

Here, $\mathcal{N}_{\mathcal{C},\mathrm{gt},k}$ and $\mathcal{N}_{\mathcal{C},\mathrm{seg},k}$ are the sets of boundary pixels of the ground truth segmentation and the superpixel segmentation, respectively. $d(\cdot,\cdot)$ denotes the euclidean distance between two boundary pixels.

Variance of Area (VoA). It was e.g., stated in [21] that a representation should be local to be meaningful. Therefore, the size of superpixels should be approximately equal in all areas of the frame. To measure this property, the *variance of area* (VoA) metric was proposed in [16]. It can be calculated for a frame k as follows:

$$\mathrm{VoA}(k) = \mathrm{E}\left[\left(\frac{A_{m,k}}{\bar{A}_k}\right)^2\right]. \tag{18}$$

Here, $\mathrm{E}[\cdot]$ is the expectation value, $A_{m,k}$ is the area of a superpixel in frame k belonging to a supervoxel m and $\bar{A}_k$ is the mean superpixel area in frame k. The metric is closely related to the superpixel size variation proposed in [21], but due to the normalization by the average superpixel size its value is independent of the image and superpixel resolution.

Superpixel Compactness (CO). As some applications favor more compact superpixels (e.g., to efficiently encode the contours of a superpixel segmentation), it was proposed in [18] to use the superpixel compactness as a benchmark metric. The compactness is calculated by weighting the isoperimetric quotient Q_m of a superpixel m (as it was also defined in [16]) with the relative superpixel size as follows:

$$\mathrm{CO}(k) = \sum_m Q_m \frac{A_{m,k}}{|\mathcal{N}_k|}. \tag{19}$$

In [21] the 3D benchmark metrics like undersegmentation error and segmentation accuracy are plotted over the average number of superpixels per frame. It was argued that different video lengths and contents require in general a different number of supervoxels. As thereby the temporal consistency of the spatio-temporal segmentation is not taken into consideration in the 3D metrics, we only plot the boundary recall distance, variance of area and the superpixel compactness over the average number of superpixels per frame. The remaining metrics are plotted over the number of supervoxels.

5.2 Experimental Setup

To evaluate the segmentation performance of our approach, we perform a series of experiments on the data set provided by [39] (BuffaloXiph) as well as the more diverse data set of [40] (BVDS). While the former is a collection of eight video clips with around 80 frames per clip the latter provides 60 clips with up to 121 frames. The data set of [39] provides one ground truth segmentation for every frame. For the data of [40] ground truth segmentation is provided by [41] for every 20th frame by four different human subjects. All the ground truth data is multi-label. To measure the label consistency, we use the data set of artificial scenes provided by [42].

For a thorough evaluation, we implemented our approach in C++ and use the optical flow method provided by [43] for the segmentation propagation. The weighting term γ of Eq. (8) is set to the common value of 50 following [44]. Besides the user selected number of superpixels per frame $|\mathcal{L}|$, the proposed method contains parameters to control certain aspects of the resulting segmentation. Namely, this is the configuration of the sliding window (F and P) and the spatial weight α of Eq. (4).

The following passage describes the set of experiments which were conducted to select the optimal parameters. From preliminary analyses it is known that the segmentation quality can be increased, up to some extent, by selecting a larger number of *past* frames. The influence of the number of *future* frames, on the other hand, is compared to the influence of the *past* frames neglectable. Hence, we set $F=2$ for all experiments. Fixing the number of *future* frames leaves two remaining parameters to be selected, namely α and P. The value of these parameters will be selected by optimizing for the three main benchmark metrics for spatio-temporal segmentation quality. Namely, the 3D segmentation accuracy (SA), 3D undersegmentation error (UE) and temporal extent. Although very important, the label consistency was not regarded for these experiments as they require ground truth optical flow. To avoid any overfitting on the test data, the optimization is performed on a separate data set, provided by [38]. The Segtrack data set consists of six video clips with up to 73 frames and a binary ground truth segmentation for each frame. The parameter optimization is achieved by performing a grid search in the intervals $\alpha = [0.86, 0.98]$ and $P = [2, 20]$ of the parameter space. For each combination of parameters a segmentation of the whole Segtrack data set in seven levels of coarseness is performed. The levels were selected to be equidistantly distributed in the range of 50 to 950 superpixels per frame. The final benchmark value for each of the three metrics is yielded by averaging over all sequences and levels of coarseness. As a result, a single value $\bar{\zeta}_p$ for each parameter combination (α, P) and benchmark metric $p \in \{\mathrm{SA}, \mathrm{TEX}, \mathrm{UE}\}$ is gained. The sets of values for the different metrics are separately normalized to lie in the interval of zero and one. Further, the scale of the 3D undersegmentation error is inverted and thus higher denoted a better result for all $\bar{\zeta}_p$. The results for the selected range of parameter combinations are depicted as color coded maps in Fig. 8.

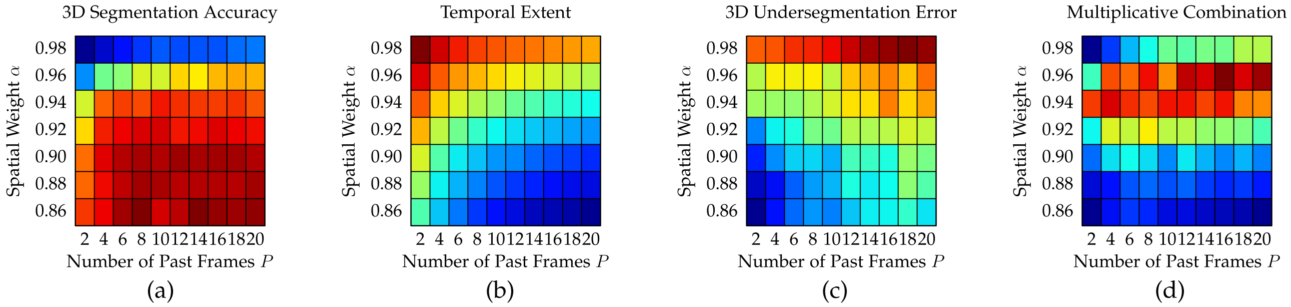

Fig. 8. Color coded plots for the 3D segmentation accuracy $\bar{\zeta}_{\mathrm{SA}}(\alpha, P)$, temporal extent $\bar{\zeta}_{\mathrm{TEX}}(\alpha, P)$, and 3D undersegmentation error $\bar{\zeta}_{\mathrm{UE}}(\alpha, P)$. For each rectangle in (a) to (c) the metrics were averaged over all sequences of the Segtrack [38] data set and seven levels of coarseness. The color maps are adjusted to the minimum and maximum values of the normalized metrics. Here, the scale of the 3D undersegmentation error is inverted, so red means better in all plots. In (d) the multiplicative combination is shown.

The plot of the 3D segmentation accuracy in Fig. 8a shows only a slight increase in 3D segmentation accuracy with a rising number of *past* frames. If, on the other hand, the spatial weight is selected to be too high the segmentation accuracy is decreased severely. This is different for the temporal extent and the 3D undersegmentation error, as it can be seen in (b) and (c) of Fig. 8. Here, both metrics show an improvement with increasing compactness of the superpixels. In contrast to this behavior, the influence of the number of *past* frames has an inverse nature. While the temporal extent decreases with a raising number of *past* frames, the segmentation error simultaneously improves.

In order to choose the optimal set of parameters, all three metrics have to be regarded simultaneously. We therefore propose, to combine the individual terms $\bar{\zeta}_p(\alpha, P)$ in a multiplicative way, to form the combined metric $\bar{\zeta}_{\mathrm{total}}$ as follows:

$$\bar{\zeta}_{\mathrm{total}}(\alpha, P) = \bar{\zeta}_{\mathrm{SA}}(\alpha, P) \cdot \bar{\zeta}_{\mathrm{TEX}}(\alpha, P) \cdot \bar{\zeta}_{\mathrm{UE}}(\alpha, P). \tag{20}$$

It should be noted that the scaling of the terms as well as their combination could have been chosen differently, e.g., a weighted sum where the weights reflect application specific preferences. But in the absence of any specific requirement the multiplicative approach was chosen, to avoid the need for an explicit weighting of the terms. This selection provides a rather general optimal choice of the parameters. The result of the combination is plotted in Fig. 8d. It can be seen that for the chosen scaling and combination the optimal band of parameters lies around $\alpha = 0.94$ to $\alpha = 0.96$ with a tendency to higher numbers of *past* frames. For the following evaluation, the parameters are selected to be $\alpha = 0.96$ and $P = 16$, as they form the maximum value in the chosen representation.

To evaluate the proposed approach, we compare it against the three state-of-the-art methods for spatio-temporal oversegmentation offering streaming capabilities. Namely, the streaming hierarchical video segmentation method (sGBH) of [30], temporal superpixels (TSP) [21] and online video seeds (OVS) [22]. Additionally, we included our previously published method [24] in all experiments (here abbreviated by TCS). The methods were chosen because they only process a subset of frames at once and thus are in principle capable of a streaming processing mode where no simultaneous access to the whole video clip is required.

For the comparison, the implementations and parameters provided on the authors' websites were used to generate multiple spatio-temporal oversegmentations with different levels of detail, i.e., different numbers of supervoxels. For a comparison, we selected the number of superpixels $|\mathcal{L}|$ (where applicable) in such a way that the number of generated supervoxels is approximately identical for all approaches. Some of the benchmark results were produced using the code provided by [28].

5.3 Experimental Results

The benchmark results for data sets with available ground truth segmentation are depicted in Figs. 9 and 10. The former contains the spatio-temporal benchmark metrics, while the latter contains the 2D benchmark metrics. In each figure the left column shows the results on BuffaloXiph and the right column the results on BVDS.

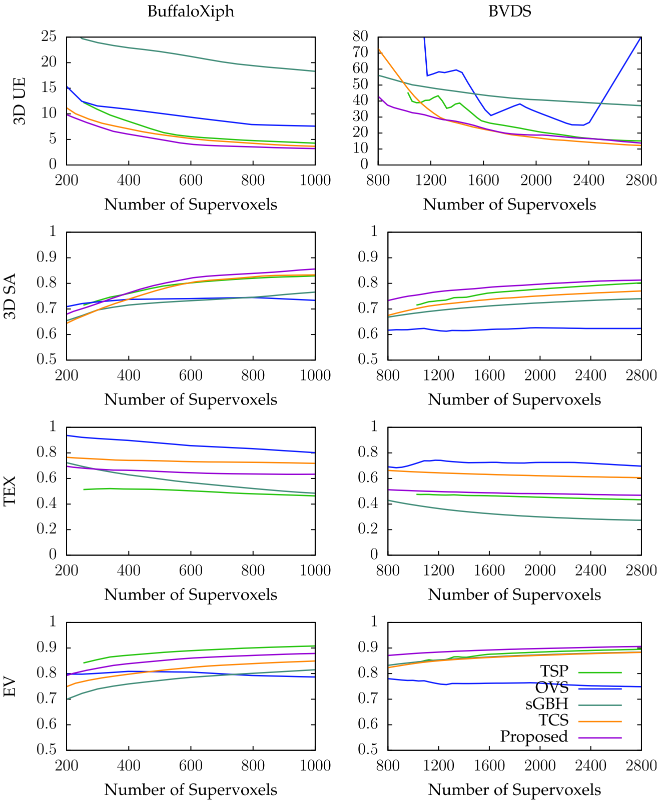

Fig. 9. Results for spatio-temporal benchmark metrics. The left column shows the results for the BuffaloXiph [39] data set, the right column for the BVDS [40] data set. Note that the metrics are plotted over the number of supervoxels. Higher values are better except for the 3D undersegmentation error (3D UE).

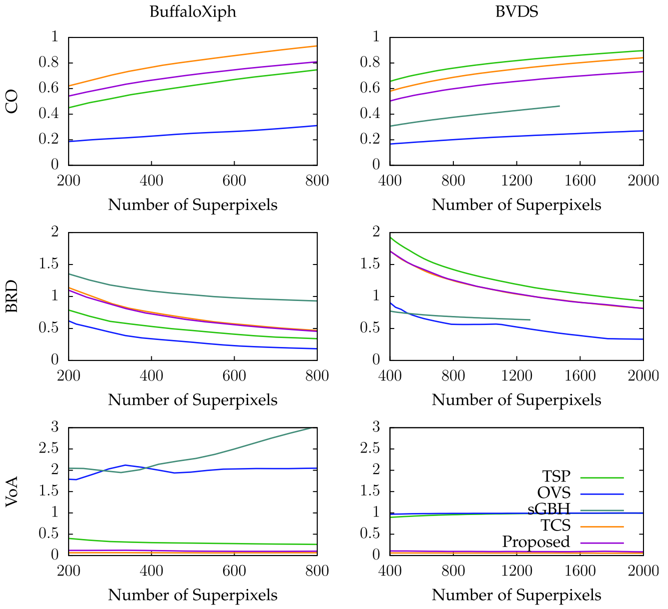

Fig. 10. Results for the 2D benchmark metrics. The left column shows the results for the BuffaloXiph [39] data set, the right column for the BVDS [40] data set. Note that the metrics are plotted over the number of superpixels per frame. Lower values are better except for the superpixel compactness (CO). For BVDS the curve of sGBH is out of the range in the variance of area diagram. It starts at about 7.5 and linearly increases up to 13.

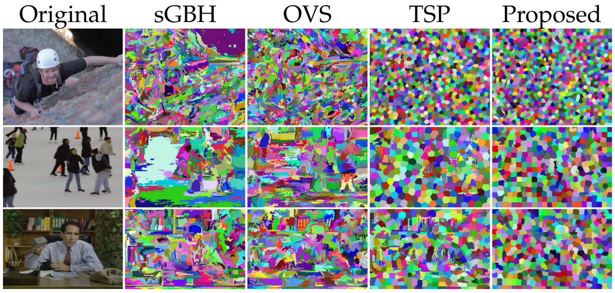

Fig. 11. Comparison of color-coded label maps. All frames have approximately 1,500 (first row) or 400 (second and third row) superpixels (frames are partially cropped for display purposes). The label maps show that our proposed method and TSP produce more compact superpixels than sGBH and OVS. Best viewed in color.

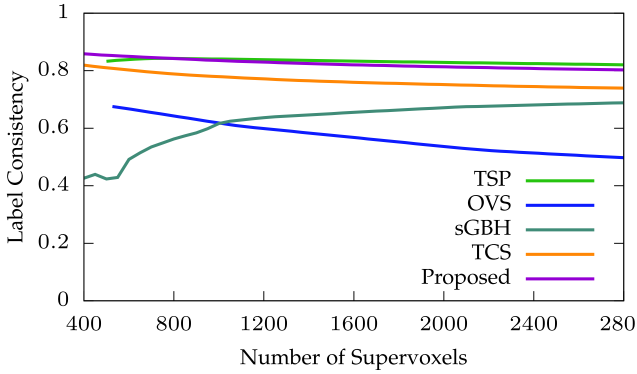

Fig. 12. Results for the label consistency benchmark showing that our approach significantly improves the consistency when compared to [24].

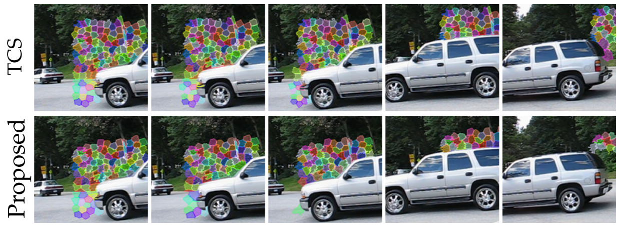

Fig. 13. Qualitative label consistency comparison with [24] on a sequence from [45]. In front of the car, the newly proposed approach correctly deletes the superpixels which should disappear due to the occlusion. Behind the car, new superpixels are created (not displayed) which prevent the superpixels from slipping down.

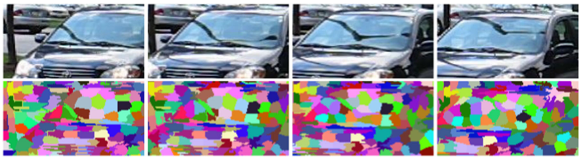

Fig. 14. Failure case in which the color consistency assumption is not fulfilled. Due to the reflection of a lamp post, the superpixels on the windshield of the car diverge from the underlying image flow. Best viewed in color.

For both data sets our algorithm performs best in 3D undersegmentation error and for most ranges of supervoxel numbers in 3D segmentation accuracy. The temporal extent is slightly decreased when compared to our previous version, while the variance of area is slightly increased. Though, the increase in variance of area is significant in some areas of the graph (nearly doubled in the lower numbers of superpixels per frame) the overall value is still quite low. The increase can be explained by the partially occluded superpixels being considered the first time in this work. Because the occluded superpixel parts are only considered during the optimization, they are not part of the final segmentation. As a result, the homogeneity constraint on the superpixel size is not fulfilled entirely in areas where occlusion happens which naturally leads to an increase in variance of area.

Fig. 11 shows color-coded label maps for all approaches highlighting the differences in compactness in a qualitative manner. For each approach a level of detail was chosen showing approximately 1,500 or 400 superpixels per frame. A frame of our previously published approach [24] is not shown as the difference in compactness is not noticeable. With the compactness parameter α in Eq. (4) our method and TCS could be made more sensitive to fine-grained details achieving a better boundary recall distance at the price of a lower compactness. But as stated in [1], [13], [18] it is beneficial to have compact superpixels. It e.g., allows for a better capturing of spatially coherent information. In addition, it simplifies the execution of subsequent processing steps, as e.g., compact superpixels tend to have a lower average number of neighbors which eases the evaluation of neighborhood relations. Additionally, further calculations like feature extraction can be performed on almost equally sized segments.

Fig. 12 shows the results for the label consistency benchmark where TSP performs best and our proposed method second best. Compared to [24] our new approach produces more consistent results for all number of supervoxels. A qualitative comparison of the label consistency between [24] and our newly proposed method can be seen in Fig. 13. Additionally, a failure case of the proposed method is shown in Fig. 14. Here, the reflection of a lamp post breaks the color consistency assumption of the image regions our method is relying on.

5.4 Complexity Considerations

In [14], the closely related SLIC superpixel approach is approximated to have a complexity of $\mathcal{O}(|\mathcal{N}_I|)$. Using this approximation, our approach for temporally consistent superpixels has a complexity of $\mathcal{O}(|\mathcal{N}_I||W|\mathcal{V})$, where $\mathcal{V}$ is the number of frames in the video sequence. As it holds that $|W| \ll \mathcal{V} < |\mathcal{N}_I|$ for reasonably long video sequences (e.g., full feature film length) and frames with mega-pixel resolution, the complexity of our approach is $\mathcal{O}(|\mathcal{N}_I|\mathcal{V})$. Compared to [30] that has a complexity of $\mathcal{O}(|\mathcal{N}_I|\mathcal{V}\log|\mathcal{N}_I|)$ it shows that our approach is more efficient with regard to the computational complexity.

5.5 Application to Interactive Video Segmentation

This section is dedicated to show the advantages of the proposed method for temporally consistent superpixels on the task of interactive video segmentation as it was described e.g., in [46]. In contrast to a fully automatic segmentation, the interactive video segmentation creates a binary segmentation of a video with the support of a human operator who roughly marks the region of interest with strokes. Given an initial input from the user, a binary segmentation is created by applying a graph cut on the voxel graph. Subsequently, the resulting segmentation can be interactively improved by the user through a refinement of the given strokes. The task of interactive video segmentation was chosen for this evaluation because it is crucial for a positive user experience to minimize the waiting time after an input has been given. It has been shown e.g., in [46] that by performing the graph cut on top of a video oversegmentation the waiting time for the user can be significantly decreased. Simultaneously, it is of high importance that the underlying oversegmentation is accurate enough to enable a final segmentation result of high quality. To evaluate the supervoxel and superpixel methods on the task of interactive video segmentation, we follow the evaluation protocol described in [19]. The described protocol avoids the need for an extensive user study by performing an offline evaluation. For the offline evaluation only the initial segmentation is considered which is obtained after a user has given an initial input. For a fair comparison, the user inputs are the same for each of the different oversegmentation algorithms whose outputs are processed by the graph cut algorithm. This implicitly assumes that by improving the quality of the initial segmentation a better or equivalent final video segmentation can be achieved with fewer user effort. Additionally, an improved initial segmentation should eventually result in a shorter overall time the user has to stay in the loop, as less user interactions are necessary. The evaluation was performed on the 40 training sequences of [40] at the highest available resolution. A segmentation level of about 3000 superpixels per frame was chosen for each approach. For each video sequence a pair of input strokes was utilized which was provided by [47]. The resulting segmentation for each oversegmentation approach was compared against the ground truth data by calculating the misclassification error rate. For this purpose the ground truth data was manually converted into a binary segmentation. A more thorough explanation of the task of interactive video segmentation and the benchmark process can be found in [19] and [47].

In addition to the four competitive oversegmentation approaches of the previous section, the proposed method is compared to a mean shift segmentation [48] (performed on frame level) and a graph cut applied on voxel-level. Table 1 shows the misclassification error rate for all five oversegmentation approaches and the voxel-wise graph. It can be seen that the usage of OVS increases the error rate when compared to the voxel-level approach. On the other hand, our newly proposed approach nearly cuts the error rate in half when compared to the voxel-level approach. A more detailed version of Table 1 as well as qualitative results can be found in the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2018.2832628.

TABLE 1
Misclassification Error Rate (in Percent) for the Interactive Video Segmentation Task

	Voxel	MS	TSP	OVS	sGBH	TCS	Proposed
Error	8.00	6.08	4.49	8.67	5.08	4.42	**4.18**

In addition to the four competitive oversegmentation algorithms, the results for a mean shift (MS) and a voxel-level (Voxel) based application of the graph cut algorithm are shown.

6 Conclusion

In this paper, we propose a novel approach for superpixels for video content. The proposed approach employs a sliding window comprising multiple consecutive frames, which are grouped into immutable *past* frames and mutable *current* and *future* frames. Whereas the *future* frames are intended to adapt to changes in the video volume, the *past* frames are conservative and try to preserve the superpixels color value over time. Our method is formulated as an efficient contour-evolving optimization scheme which adapts the superpixels only at their boundaries.

By propagating whole superpixel shapes using a weighted average optical flow, our method is able to preserve the shape and constellation of the superpixel segmentation over time while simultaneously detecting occluded superpixels and disoccluded image regions. This knowledge is used to adapt the superpixel segmentation to structural changes in the video volume and to improve the consistency of the superpixel flow with the movement of the underlying image patches.

In a thorough, in-depth evaluation based on established benchmarks, the proposed approach is compared to state-of-the-art, streaming-capable spatio-temporal oversegmentation methods. The evaluation shows that our approach produces highly competitive results making it an excellent basis for all tasks requiring temporal consistency and a high segmentation accuracy as e.g., video segmentation and tracking applications. This is further shown by comparing the streaming-capable approaches as basis for the task of interactive video segmentation where the proposed approach provides the lowest overall misclassification rate. Further details on the comparison can be found in the supplemental material, available online.

References

[1] X. Ren and J. Malik, “Learning a classification model for segmentation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 10–17.

[2] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 654–661.

[3] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1323–1330.

[4] C. Vogel, K. Schindler, and S. Roth, "Piecewise rigid scene flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1377–1384.

[5] J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun, "Estimating the 3D layout of indoor scenes and its clutter from depth sensors," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1273–1280.

[6] J. Tighe and S. Lazebnik, "SuperParsing: Scalable nonparametric image parsing with superpixels," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 352–365.

[7] H. Meuel, M. Munderloh, M. Reso, and J. Ostermann, "Mesh-based piecewise planar motion compensation and optical flow clustering for ROI coding," *APSIPA Trans. Signal Inf. Process.*, vol. 4, Oct. 2015, Art. no. e13.

[8] A. Jain, S. Chatterjee, and R. Vidal, "Coarse-to-fine semantic video segmentation using supervoxel trees," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1865–1872.

[9] F. Galasso, R. Cipolla, and B. Schiele, "Video segmentation with superpixels," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 760–774.

[10] F. Galasso, M. Keuper, T. Brox, and B. Schiele, "Spectral graph reduction for efficient image and streaming video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 49–56.

[11] A. Khoreva, F. Galasso, M. Hein, and B. Schiele, "Classifier based graph construction for video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 951–960.

[12] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[13] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "TurboPixels : Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

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

[15] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 211–224.

[16] F. Perbet, B. STENGER, and M. Atsuto, "Homogeneous superpixels from Markov random walks," *IEICE Trans. Inf. Syst.*, vol. 95, no. 7, pp. 1740–1748, 2012.

[17] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha, "Structure-sensitive superpixels via geodesic distance," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 1–21, May 2013.

[18] A. Schick, M. Fischer, and R. Stiefelhagen, "An evaluation of the compactness of superpixels," *Pattern Recognit. Lett.*, vol. 43, pp. 71–80, 2014.

[19] M. Reso, B. Scheuermann, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Interactive segmentation of high-resolution video content using temporally coherent superpixels and graph cut," in *Proc. Int Symp. Visual Comput.*, 2014, pp. 281–292.

[20] W.-D. Jang and C.-S. Kim, *Streaming Video Segmentation via Short-Term Hierarchical Segmentation and Frame-by-Frame Markov Random Field Optimization*. Cham, Switzerland: Springer, 2016, pp. 599–615.

[21] J. Chang, D. Wei, and J. W. Fisher III, "A video representation using temporal superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2051–2058.

[22] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Gool, "Online video seeds for temporal window objectness," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 377–384.

[23] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Temporally consistent superpixels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 385–392.

[24] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Superpixels for video content using a contour-based EM optimization," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 692–707.

[25] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Spatiotemporal closure," in *Proc. Asian Conf. Comput. Vis.*, 2010, pp. 369–382.

[26] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2141–2148.

[27] C. L. Zitnick, N. Jojic, and S. B. Kang, "Consistent segmentation for optical flow estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1308–1315.

[28] C. Xu and J. J. Corso, "LIBSVX: A supervoxel library and benchmark for early video processing," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 272–290, Sep. 2016.

[29] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[30] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 626–639.

[31] Y. Liang, J. Shen, X. Dong, H. Sun, and X. Li, "Video supervoxels using partially absorbing random walks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 928–938, May 2016.

[32] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," *Int. J. Comput. Vis.*, vol. 111, no. 3, pp. 298–314, 2015.

[33] O. Freifeld, Y. Li, and J. W. Fisher III, "A fast method for inferring high-quality simply-connected superpixels," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 2184–2188.

[34] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[35] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

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

[37] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1202–1209.

[38] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg, "Motion coherent tracking using multi-label MRF optimization," *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 190–202, 2012.

[39] A. Y. C. Chen and J. J. Corso, "Propagating multi-class pixel labels throughout video frames," in *Proc. Western New York Image Process. Workshop*, Nov. 2010, pp. 14–17.

[40] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik, "Occlusion boundary detection and figure/ground assignment from optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2233–2240.

[41] F. Galasso, N. S. Nagaraja, T. J. Cárdenas, T. Brox, and B. Schiele, "A unified video segmentation benchmark: Annotation, metrics and analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3527–3534.

[42] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 611–625.

[43] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[44] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Aug. 2004.

[45] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[46] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 585–594, 2005.

[47] M. Reso, "Temporally consistent superpixels," Institut für Informationsverarbeitung, Ph.D. dissertation, Leibniz Universität Hannover, Hannover, Germany, 2017.

[48] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 750–755.

Matthias Reso received the Dipl-Ing degree from the Universität Paderborn, in 2011 and the Dr-Ing degree from the Leibniz Universität Hannover, in 2017. He studied information technology with an emphasis on communication technology and microelectronics with the Universität Paderborn. From 2012 to 2016, he worked as a research assistant in the Institut für Informationsverarbeitung, Leibniz Universität Hannover. Since 2016 he is a research scientist with Fyusion Inc in San Francisco.

Jörn Jachalsky received the Dipl-Ing and Dr-Ing degrees in electrical engineering from the Leibniz Universität Hannover, Hannover, Germany, in 1998 and 2006, respectively. He is a principal scientist with Technicolor Research & Innovation. From 1998 to 2005, he worked as a research engineer in the Institute of Microelectronic Systems, Leibniz Universität Hannover. In 1999, he was a guest researcher at the C&C Media Lab, NEC Corporation, Kawasaki, Japan and since 2005 he has been with Technicolor Research & Innovation. Moreover, since October 2014 he is a visiting lecturer with the Leibniz Universität Hannover for the lecture “Digital Signal Processing”. His current research interests include algorithms and systems for computer vision and image/video analysis and processing as well as machine learning. He is a member of the IEEE.

Bodo Rosenhahn received the Dipl-Inf and Dr-Ing degrees from the University of Kiel, in 1999 and 2003, respectively. He studied computer science (minor subject medicine) with the University of Kiel. From 2003-2005 he was (DFG) postdoc with the University of Auckland, New Zealand. From 2005 to 2008, he worked as senior researcher in the Max-Planck Insitute for Computer Science in Saarbruecken, Germany. Since 2008, he is a full professor with the Leibniz-University of Hannover, heading a group on automated image interpretation. His research focus is on computer vision and machine learning, he has written more than 180 research papers, holds more than 10 patents and received several awards.

Jörn Ostermann received the Dipl-Ing and Dr-Ing degrees from the University of Hannover, in 1988 and 1994, respectively. He studied electrical engineering and communications engineering with the University of Hannover and Imperial College London, respectively. From 1988 till 1994, he worked as a research assistant in the Institut für Theoretische Nachrichtentechnik conducting research in low bit-rate and object-based analysis-synthesis video coding. In 1994 and 1995 he worked with the Visual Communications Research Department, AT&T BELL LABS on video coding. He was a member of the *Image Processing and Technology Research* within *AT&T Labs-Research* from 1996 to 2003. Since 2003, he is full professor and head of the Institut für Informationsverarbeitung, Leibniz Universität Hannover (LUH), Germany. From 1993 to 1994, he chaired the European COST 211 sim group coordinating research in low bitrate video coding. Since 2008, he is the chair of the Requirements Group of MPEG (ISO/IEC JTC1 SC29 WG11). He was a scholar of the German National Foundation. In 1998, he received the AT&T Standards Recognition Award and the ISO award. He is a fellow of the IEEE (class of 2005) and member of the *IEEE Technical Committee on Multimedia Signal Processing* and past chair of the *IEEE CAS Visual Signal Processing and Communications (VSPC) Technical Committee*. He served as a distinguished lecturer of the *IEEE CAS Society* (2002/2003). He published more than 100 research papers and book chapters. He is coauthor of a graduate level text book on video communications. He holds more than 30 patents. His current research interests include video coding and streaming, computer vision, 3D modeling, face animation, and computer-human interfaces.