$\mathbb{J}$Superpixels for Video Content Using a Contour-based EM Optimization

Matthias Reso[†], Jörn Jachalsky[‡], Bodo Rosenhahn[†], and Jörn Ostermann[†]

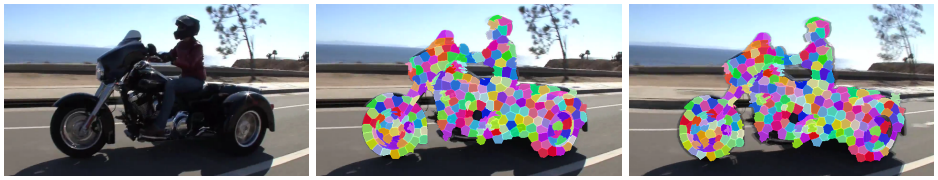
[†]Leibniz Universität Hannover, Germany
reso@tnt.uni-hannover.de
[‡]Technicolor Research & Innovation, Germany
joern.jachalsky@technicolor.com

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 algorithms have gained in the recent years. However, most methods show a lack of temporal consistency or fail in producing temporally stable segmentations. In this paper, we propose a novel, contour-based approach that generates temporally consistent superpixels for video content. It can be expressed in an expectation-maximization framework and utilizes an efficient label propagation built on backward optical flow in order to encourage the preservation of superpixel shapes and their spatial constellation over time. Using established benchmark suites, we show the superior performance of our approach compared to state of the art supervoxel and superpixel algorithms for video content.

1 Introduction

In [1] superpixels were introduced as new image primitives grouping spatially coherent pixels that share the same low-level features as e.g. color or texture into small segments of approximately same size and shape. Over the last decade, superpixel algorithms have become a common preprocessing step for a variety of computer vision applications. These applications include e.g. video segmentation [2,3], tracking [4], multi-view object segmentation [5], scene flow [6], 3D layout estimation of indoor scenes [7], interactive scene modeling [8], image parsing [9], and semantic segmentation [10,11]. Using such an over-segmentation has two major benefits. First, the number of image primitives is significantly reduced. Second, superpixels provide a spatial support for the extraction of region-based features [12].

More recently, the idea of superpixels was extended from the domain of still images to the domain of video sequences. In general, all related approaches can be classified as generating either supervoxels (e.g. [13,14,15]) or superpixels that are temporally consistent (e.g. [16,17,18,19]). As noted in [18], superpixels with temporal consistency and supervoxels can be converted into the other class if certain constraints are met. While over-segmentation algorithms for still images



(a) (b) (c)

Fig. 1. Example of a superpixel segmentation providing temporal consistency and a steady spatial constellation of the superpixels over time. Despite the movement and the jiggling camera shot the superpixels stay at their initial positions on the motorcycle over several frames. (a) Original frame; (b) A full superpixel segmentation of the video was performed and a subset of superpixels was manually selected in one frame and colored for visualization; (c) Subset with the same superpixels after several frames. Same color as in (b) means temporal connectedness.

should capture main object boundaries, the methods for video content should additionally capture the temporal connections between regions in successive frames. In order to achieve a consistent labeling, that can be leveraged for applications like tracking or video segmentation, it is also important for the segments to reflect the motion of the image regions they represent. Thus, a segment should not change its shape if the corresponding image region does not change its shape and the spatial constellation of the segments should stay constant over time as long as the corresponding regions do not change positions (See Fig. 1 for an example). In the following, we briefly describe oversegmentation approaches for video content that aim at compact and spatially coherent regions of approximately the same size and shape that are also consistent over time. An early example, which is not explicitly labeled as superpixel or supervoxel approach but shares a similar idea, can be found in [20].

In [14] a first supervoxel approach was published that covers the video volume with overlapping cubes, whereas each cube corresponds to one label. The volume of the cubes determines the maximum volume of the supervoxels to be generated. The longer the cubes are, the higher the temporal consistency can be. The assignment of each voxel to one label is done using energy minimization techniques. In [15] not only the SLIC superpixel approach is described but also its extension to supervoxels. Thereby, it introduces a temporal distance term penalizing supervoxels with a long duration.

Other approaches like [13,21,16,17,18] aim at supervoxel and superpixel representations with extended temporal duration. In [13] an approach for hierarchical video segmentation is proposed that is based on the graph-based image segmentation approach introduced in [22], which is first applied on pixel-level and then iteratively on region-level in order to create a hierarchical segmentation. Streaming capabilities were added to [13] in [21] by applying a Markov assumption on the video stream. Both, [13] and [21] generate supervoxel seg-

mentations, which —if converted to a superpixel representation— show a lack of temporal stability as the shape of the segments changes extensively from frame to frame.

A first approach towards temporal superpixels was introduced in [19] using optical flow information to initialize the seeds for the superpixels in each new frame. Using these seeds, the superpixels are grown only on frame level. While achieving a more temporally stable superpixel segmentation it fails to explicitly handle structural changes in the video sequences. A strategy for creation or splitting as well as termination of superpixels, which provides the capability to handle structural changes in the video scene, was first introduced by [16]. The approach utilizes a generative probabilistic model for superpixels in video sequences. Moreover, the flow is explicitly modeled between the frames in order to propagate the superpixels. In [17] an online video superpixel algorithm based on [23] was introduced. It uses hill climbing for the optimization and considers a hierarchy of blocks at different sizes. The results of the intermediate block level are used to initialize new frames. The superpixel approach presented in [18] uses a global color subspace and multiple spatial subspaces to cluster the pixels in an observation window that comprises multiple frames and is shifted along the video volume. In order to initialize new frames the spatial centers of the superpixels are propagated into a new frame similarly to [19].

Although [16] provides a mostly temporally stable segmentation it falls behind the more recent approaches of [17] and [18] with respect to the duration of the spatio-temporal segments. However, the latter two algorithms —each to some extent— fail to produce segmentations with a steady spatial constellation of the superpixels over time. Hence, in this work we introduce a novel method for superpixels on video content. We utilize the main ideas of [18] to maximize the length of the spatio-temporal segments and introduce new techniques to generate a temporally more stable segmentation. The key contributions of this paper are the following: *(i)* we propose a fully contour-based approach for superpixels on video sequences, which is expressed in an expectation-maximization (EM) framework, and generates superpixels that are spatially coherent and temporally consistent. *(ii)* We utilize an efficient label propagation using backward optical flow in order to encourage the preservation of superpixel shapes when appropriate. Finally, *(iii)* we present superior results comparing our approach against the state of the art using the established benchmark suites [24,25].

The remainder of the work is organized as follows: In Section 2, we discuss the details of our approach and present the experimental results comparing it to the state of the art using the established benchmark suites in Section 3. Section 4 concludes this paper.

2 Superpixels for Video Content

Our algorithm is based on an analysis of the approach proposed in [18], entitled *Temporally Consistent Superpixel* (TCS). Thus, before we discuss our algorithm in Section 2.2, we will shortly summarize the main ideas of TCS.

2.1 Temporally Consistent Superpixels in a Nutshell

In general, TCS performs an energy-minimizing clustering using a multi-dimensional feature space. For the clustering, the feature-space is separated into a global color subspace and multiple local spatial subspaces.

More specifically, the energy-minimizing framework used in TCS clusters pixels based on their five dimensional feature vector $\begin{bmatrix} l & a & b & x & y \end{bmatrix}$. Each vector contains the three color values $\begin{bmatrix} l & a & b \end{bmatrix}$ in CIELAB-color space and the pixels coordinates $\begin{bmatrix} x & y \end{bmatrix}$. In order to capture the temporal connections between superpixels in successive frames, the clustering is performed over an observation window spanning K frames. The separated feature space is realized in the following way. Each cluster center represents one temporal superpixel. A cluster center consists of one color center for the complete observation window and multiple spatial centers with one for each observed frame.[1]

While processing the video volume the observation window is shifted in steps of one frame along the timeline. After each step an optimal set of cluster centers Θ_{opt} is obtained. The mapping of the pixels inside the observation window to these cluster centers is denoted as σ_{opt}. An energy function (1) is defined, which sums up the energies necessary to assign a pixel at position x, y in frame k to a cluster center $\theta \in \Theta_{opt}$. This assignment or mapping is here denoted by $\sigma_{x,y,k}$.

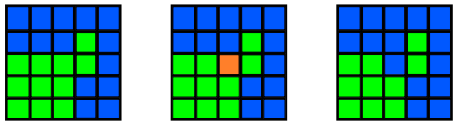
$$E_{total} = \sum_{k} \sum_{x,y} (1-\alpha) E_c(x,y,k,\sigma_{x,y,k}) + \alpha E_s(x,y,k,\sigma_{x,y,k}) \tag{1}$$

The energy needed for an assignment is the weighted sum of a color dependent energy $E_c(x,y,k,\sigma_{x,y,k})$ and a spatial energy $E_s(x,y,k,\sigma_{x,y,k})$. Both energy terms are proportional to the Euclidean distance in color space and image plane, respectively. The trade-off between color-sensitivity and spatial compactness is controlled by a weighting factor α, which has a range between 0 (fully color-sensitive) and 1 (fully compact). Thereby, $\alpha = 1$ results in Voronoi cells. The energy function is minimized using an iterative optimization scheme, which can be viewed as an EM approach.

In the expectation-step (E-step) of iteration $l+1$ a new estimation of the optimal mapping, here denoted as $\hat{\sigma}^{l+1}_{x,y,k}$, is determined, which minimizes (1) based on the estimation of the optimal set of cluster centers $\hat{\Theta}^{l}_{opt}$ calculated in the maximization-step (M-step) of iteration l.

After that, the estimation of the optimal set of cluster centers $\hat{\Theta}^{l+1}_{opt}$ is updated in the M-step of iteration $l+1$ given the updated mapping by calculating the mean color and mean spatial values of the assigned pixels. The alternation of the two steps continues until the energy (1) drops below a specific bound or a fixed number of iterations is performed. In the hybrid clustering proposed for TCS, only the $K_F < K$ most future frames in the observation window are reassigned during the optimization. For the remaining $K - K_F$ frames the determined mapping is kept in order to preserve the color clustering found.

[1] The underlying assumption is that a temporal superpixel should share the same color in successive frames but not necessarily the same position.



(a) (b) (c)

Fig. 2. The three subfigures exemplarily show pixels between two superpixels (green and blue). If the centered pixel (colored orange in (b)) changes its assignment, the two pixels on its right lose connection to the green superpixel and thus they would be split-off from the main mass (as shown exemplarily in (c)). Therefore no assignment change is performed in situations like these.

While shifting the observation window new frames entering the window need to be initialized. In TCS this is done by projecting each spatial center of the most future frame into the new frame in the direction of the weighted average of the dense optical flow calculated over the corresponding superpixel.

2.2 Superpixels for Video Content Using a Contour-based EM Optimization

Revisiting the ideas of TCS, we made the following two observations: *(a)* In order to achieve a higher run-time performance the initial energy-minimizing clustering and the contour-based post processing are separated steps. Thereby, the shape of the superpixels can change completely in each iteration. *(b)* New frames added to the observation window are initialized by propagating only the spatial centers of the preceding frame into the new frame. As a consequence, the shape information obtained in the frames before is discarded. These observations lead to our two proposals.

Firstly, we employ the optimization scheme, proposed by [26] for still images, to optimize the energy function (1). This means that only pixels at a contour of a superpixel, so called contour pixels, can change their assignment to a cluster. A contour pixel at position x, y has at least one pixel in its 4-connected neighborhood $\mathcal{N}^4_{x,y}$, which is assigned to a different cluster, i.e. a temporal superpixel, or is unassigned. The occurrence of unassigned pixels and their handling is described in detail below. Moreover, the assignment of a contour pixel can only be changed to one of the clusters of the pixels in $\mathcal{N}^4_{x,y}$ as proposed by [26]. The E-step of the optimization can be expressed as

$$\hat{\sigma}^{l+1}_{x,y,k} = \underset{\hat{\sigma}^l_{\bar{x},\bar{y},k}:\bar{x},\bar{y}\in(\mathcal{N}^4_{x,y}\cup x,y)}{\operatorname{argmin}} (1\text{–}\alpha)E_c(x,y,k,\hat{\sigma}^l_{\bar{x},\bar{y},k})+\alpha E_s(x,y,k,\hat{\sigma}^l_{\bar{x},\bar{y},k}) \;\; \forall x,y \in \mathcal{C}^l_k \tag{2}$$

where $\mathcal{C}^l_k$ is the set of contour pixels after iteration step l in frame k. The optimization is done for the K_F most future frames in the observation window. The M-step remains unmodified. The optimization can be terminated if there are no further assignment changes for the contour pixels or if a maximum number of iterations has been reached.

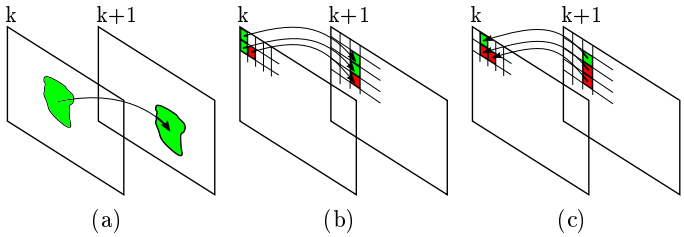


Fig. 3. Possible variations of superpixel label propagation to new frames: (a) The whole superpixel is shifted by the mean optical forward flow, (b) each label is propagated using a dense optical forward flow and (c) for each pixel in the new frame the label is looked up at a position determined by the optical backward flow. For case (c) no collisions can occur as for each pixel position only one label is looked up in the previous frame. Gaps can not occur as the optical backward flow vector are cropped when pointing outside the valid image area.

In addition to the description above, there are two constraints. (a) An assignment change is only done if the spatial coherency of the superpixels is guaranteed. This constraint prevents that fragments of a temporal superpixel are split-off during the optimization, as shown in Fig. 2 (See [27] for more details on this constraint). (b) A newly proposed constraint affects unassigned contour pixels. These are assigned to the cluster of one of its adjacent pixels based on (2). As a consequence, the additional post-processing step required in TCS [18] to ensure the spatial coherency is not necessary and can be omitted.

Secondly, we propose to transfer the whole shape of the superpixels to the new frame to be initialized leveraging optical flow information unlike the approach described in [17]. This helps to preserve the shape information as well as the superpixel constellation obtained in previous frames. There are several ways to realize such an initialization of the new frames. One could be the shift of the complete superpixel label using the mean optical flow as depicted in Fig. 3a. An alternative would be the usage of a dense optical flow predicted for each pixel of the superpixel. Thus, the superpixel label is propagated into the new frame as shown in Fig. 3b.

These two options have the following drawback: If two superpixels propagated into the new frame overlap, it is necessary to detect collisions. In addition, it is possible that there are unassigned parts in the frame that need to be initialized if e.g. adjacent superpixels are moved away from each other, resulting in a gap between the superpixels. Both cases are illustrated in Fig. 4 and apply in the same manner to the shifting of pixels by a dense optical forward flow.

To prevent these problems, we propose to use a dense optical backward flow, which is computed from the frame entering the observation window $k+1$ to the preceding frame k in the window. The initial mapping of pixels to cluster centers of the new frame $k+1$ denoted as $\hat{\sigma}^{init}_{x,y,k+1}$ can be deduced from the mapping

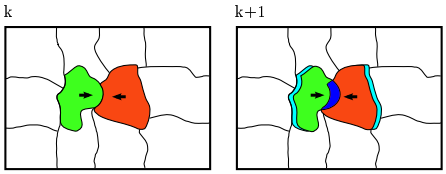


Fig. 4. Problems that can occur when propagating whole superpixels by mean optical flow from frame k to frame $k+1$: Moving adjacent superpixels in opposite directions produces gaps (cyan) while a movement toward each other leads to overlapping areas (dark blue). This also applies in the same manner to the propagation of pixels by a dense optical forward flow.

for frame k (after L iteration steps) as follows:

$$\hat{\sigma}^{init}_{x,y,k+1} = \hat{\sigma}^{L}_{x+u,y+v,k}\,, \tag{3}$$

where u and v are the optical backward flow components, which are rounded to the nearest integer for the horizontal and vertical direction. Additionally, the components are clipped if pointing outside of the valid image area

By using this approach no collisions have to be detected as for each pixel position only one label is determined eliminating the possibility of collisions. Gaps do not occur as the optical backward vectors are cropped if pointing outside the valid image area. The only issue left, which also exists for the both cases using optical forward flow, is that the propagated superpixels can be fragmented, i.e. they are not spatially coherent. In that case, the largest fragment is determined and the others are set to unassigned. These are handled in the E-step of the optimization, as they are part of the contour pixels. The first frame to be segmented is initialized with non-overlapping rectangles of equal size.

In [18] a heuristic was introduced to encounter structural changes in the video volume, which are e.g. occlusions, disocclusions, and objects approaching the camera as well as zooming. The decision to split or terminate a temporal superpixel was made based on a linear growth assumption of the superpixel size. Additionally, a separate balancing step was performed to keep the number of superpixels per frame constant. We replaced these two steps with a single one by introducing an upper and lower bound for the superpixel size. Superpixels that are larger than the upper bound are split. The ones that are smaller than the lower bound are terminated. These bounds are coupled to the number of superpixels initially specified by the user. Thus, the user defines a minimum and maximum number of superpixel per frame N_{min} and N_{max}, respectively. Based on that, the upper and lower bound A_{low} and A_{up} are set as follows

$$A_{low} = \frac{|P|}{N_{max}} \quad \text{and} \quad A_{up} = \frac{|P|}{N_{min}} \tag{4}$$

where $|P|$ is the number of pixels per frame. In our implementation we specified a number of superpixels N and set N_{min} and N_{max} to $\frac{1}{2}N$ and $2N$.

3 Experimental Results

In this section, we evaluate the quantitative performance of the proposed approach using standard benchmark metrics. Additionally, we present qualitative results and compare the approach to state of the art supervoxel and superpixel approaches for video content. For the experiments we set a fixed α of 0.96, a window size of $K = 15$ and performed $L = 5$ EM-iterations.[2] We used the datasets provided by [28] and [29]. The first dataset provides 40 training and 60 test sequences of up to 121 frames. A multi-label ground truth segmentation is made available by [30] including four segmentations for every twentieth frame. We use the half-HD version of the dataset and show the results for the test sequences. The second dataset provides 8 video sequences of around 80 frames including a single multi-label ground truth segmentation for every frame. The results are shown as mean values calculated over each dataset separately. To create them we use version 3.0 of LIBSVX originally published in [24] as well as the code provided by [25]. Our current MATLAB implementation processes 3 to 4 video frames (in an HD-ready resolution) per minute including the optical flow calculation and $N = 3000$ superpixels. Thereby, it should be noted that the current version is only moderately optimized with respect to the runtime-performance.

3.1 Metrics and Baseline

As the quality of the spatio-temporal segmentation is as important as the quality of the segmentation on frame level, we considered the following set of supervoxel and superpixel benchmark metrics that we will review briefly below. For a more thorough explanation please refer to [24,21,31,32]. The first four metrics are tailored to the evaluation of supervoxel and video segmentation algorithms and indicate the quality of the spatio-temporal segmentation. The last two metrics are suitable for evaluating the image segmentation quality on frame level.

3D Undersegmentation Error (*UE*): This metric proposed by [24] counts the number of voxels *bleeding out* of the ground truth segmentation volume. For a given segmentation with non-overlapping segments $s_1, s_2, ..., s_M$ and a ground truth segment g_n the 3D undersegmentation error is calculated as follows

$$UE(g_n) = \frac{\left[\sum_{(s_m | s_m \cap g_n \neq \emptyset)} |s_m|\right] - |g_n|}{|g_n|}. \tag{5}$$

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

3D Segmentation Accuracy (*SA*): Also proposed in [24] the 3D segmentation accuracy denotes the fraction of the video volume that can be reproduced by the segmentation's overlap with the ground truth segments if for each segment only the overlap with a single ground truth segment is counted. Therefore, the

[2] The changes after 5 iterations are only marginal. It should be noted that the boundary can move more than 1 pixel per iteration.

segments are assigned to the ground truth segment, for which it has the maximum overlap with, and then just the overlap of the segments with its assigned ground truth segment is counted.

$$SA = \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{o \in O_n} (|s_o \cap g_n|)}{|g_n|} \tag{6}$$

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

Average Temporal Length: 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. This metric always has to be evaluated in conjunction with a metric like 3D segmentation accuracy or undersegmentation error as a long temporal segment duration is only valuable together with a high quality spatio-temporal segmentation.

Explained Variation (EV): This metric was proposed in [31] and indicates how well the original information can be represented with a given oversegmentation as a representation of lower detail.

2D Boundary Recall (BR): The 2D boundary recall measures the fraction of the boundary annotated in the ground truth that is covered by a superpixel boundary. A ground truth boundary pixel is counted as covered if a superpixel boundary is within the pixel-distance ϵ, which is set to 1 for our experiments.

Variance of Area (VoA): In [32] the variance of area was proposed as a metric for the homogeneity of superpixels sizes and is calculated for a frame k as follows

$$VoA(k) = \text{var}\left(\frac{A_{m,k}}{\bar{A}_k}\right). \tag{7}$$

$A_{m,k}$ 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.

In [16] the 3D benchmark metrics like UE and SA are plotted over the average number of superpixels per frame arguing that different video length and content require in general a different number of supervoxels. We will plot only the BR and VoA over the average number of superpixels per frame as otherwise the temporal consistency of the spatio-temporal segmentation is not taken into consideration at all in the 3D metrics. We think it is reasonable to plot the metrics over the number of supervoxels as long as the number of frames in the sequences used is not deviating too much.

3.2 Quantitative Evaluation

The baseline (BL) for the following experiments is [18] with the slight modification that for the optical flow [33] is used instead of [34]. First we successively show the impact of our contributions that were added to the baseline version: the contour-based optimization scheme in an EM framework (CE), label propagation

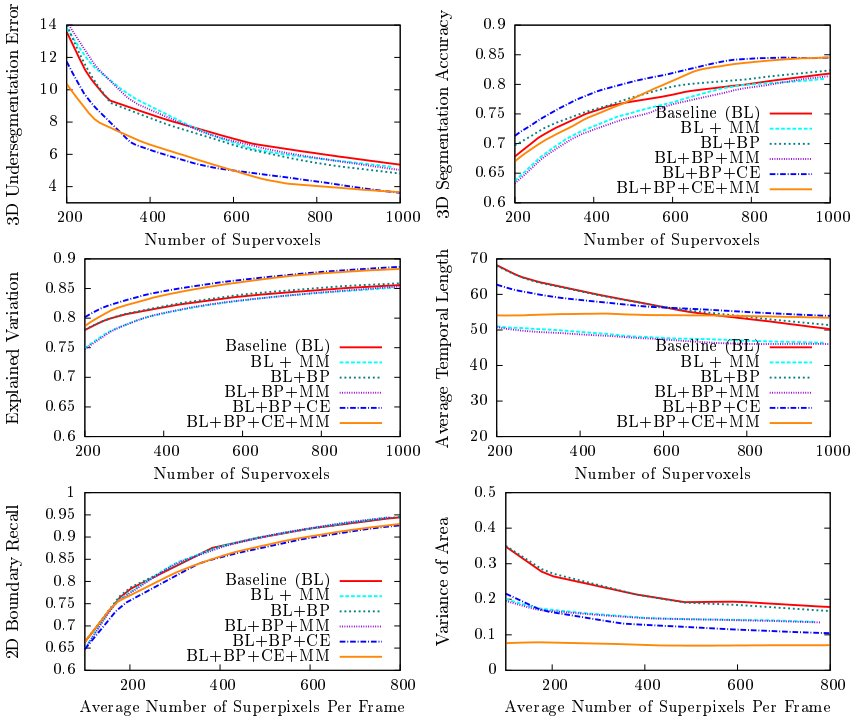


Fig. 5. Benchmark results on the dataset from [29] for the baseline implementation and our contributions. Please note that the 2D boundary recall and the variance of area are plotted over the average number of superpixels per frame and not over the number of supervoxels as in the other diagrams.

for initialization using the optical backward flow (BP) as well as the simplified handling of structural changes, i.e. the splitting and termination of superpixels based on a minimal and maximal number of superpixels (MM). As the contour-based optimization requires label propagation, the results for CE alone cannot be presented, only in combination with the optical flow backward propagation (BP). For each contribution, we performed several segmentations of the video sequences provided by [29] using a range of desired superpixels per frame (resulting in different numbers of supervoxels) and used the aforementioned metrics on the segmentations. The results are shown in Fig. 5.

It is evident that MM and BP alone and their combination (MM+BP) have virtually no effect on the *UE*, *SA*, *EV* and *BR* compared to the baseline (BL). Only a slight degradation for small numbers of supervoxels for *UE*, *SA* and *EV* can be noticed. In addition, MM reduces the average temporal length while at the same time *VoA* is improved. This can be explained with the fact that the baseline (BL) allows for smaller superpixels. Again, BP alone achieves nearly identical results as the baseline for the average temporal length as well as *VoA*. The improvements are achieved with the introduction of CE. Nearly for all metrics

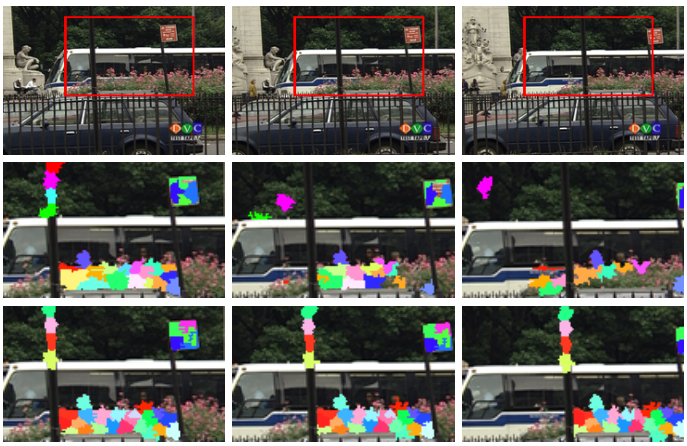


Fig. 6. Example for a challenging segmentation task with low contrast and high motion. Top row shows the original sequence with a marked area magnified in the rows below. For the two lower rows a full segmentation was performed with the baseline approach (BL, middle) and the proposed approach (BL+BP+CE+MM, bottom). Only a subset of superpixels is shown, which was manually selected and colored. Same color means temporal connectedness. In the middle row the superpixels are torn away by the motion introduced by the camera panning, while they keep their position and constellation for the proposed approach.

improved results are obtained, especially for higher numbers of supervoxels. Only the *BR* is slightly impaired. This highlights the positive impact of the contour-based optimization approach that it can achieve in combination with the efficient label propagation using optical backward flow.

3.3 Qualitative Evaluation

Temporal superpixels should cover corresponding image regions over time. Hence, their spatial constellation should not change if they cover a nearly rigidly moving object. Although this seems to be an easy task to accomplish, Fig. 6 shows an example of a scene likely to be found in natural video sequences were previous approaches fail. The top row shows the original sequence. The region of interest including low contrast (street light, trees) and high motion (bus, flowers) is marked with a red rectangle and cropped in the rows below. The marked superpixels in the middle row are taken from the segmentation produced by the baseline approach (BL) from Sec. 3.2. The superpixels in the lower row were taken from a segmentation with our proposed approach (BL+BP+CE+MM). It can be seen that, while both approaches use the same optical flow algorithm, the baseline approach has an issue with keeping the superpixels to their initial positions. Even in the area with higher contrast (flowers) the superpixels change their position from frame to frame in an unpredictable way. In the bottom row,

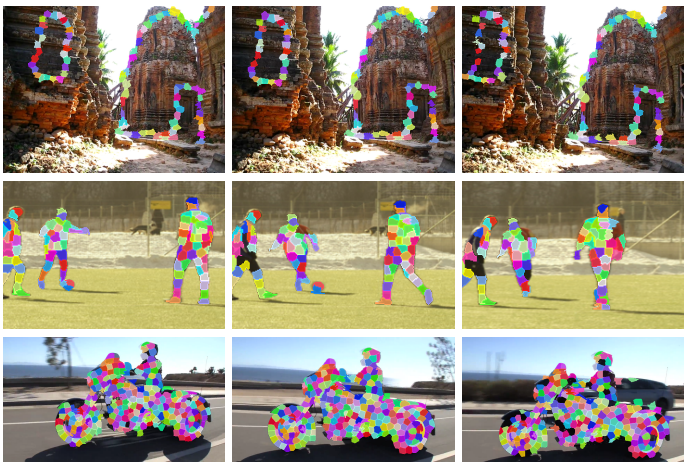


Fig. 7. Results generated by our algorithm showing the segmentation quality and temporal stability. For each sequence a full superpixel segmentation was performed and a subset of superpixels was manually selected in one frame (**not** necessarily in the first frame). Same color means temporal connectedness.

showing the result of our proposed approach, the superpixels stick to their original positions. While this is only a single example picked out for illustration, this behavior can be observed in other sequences and is also common for supervoxel methods. It should also be noted that present established benchmark metrics do not capture these kinds of errors as the jumping superpixels often stay within the same ground truth label and therefore do not have a negative impact on metrics like undersegmentation error or segmentation accuracy.

In Fig. 7 additional qualitative results are shown. The upper row of images shows equally spaced frames from a subsequence spanning 40 frames of an unsteady hand camera shot. Despite the shaking camera, the superpixel formation sticks close to their initial positions. The second row illustrates the performance of our approach for non-rigid motion. The sequence shown in the third row spans 69 frames of a sequence with high motion blur noticeable e.g. in the second image. It should be noted that the last two images reveal the limits of the approach as some superpixels are misled and switch to the car that is overtaken by the motorcycle.

3.4 Comparison to State of The Art Algorithms

In this section we compare our final approach to four state of the art algorithms producing supervoxels or temporally consistent superpixels. We compare our approach to *StreamGBH* published in [21], which is the only representative of the class of supervoxel algorithms. Furthermore, we compare it against the superpixel approaches *Temporally Consistent Superpixel* (TCS) [18], *Temporal*

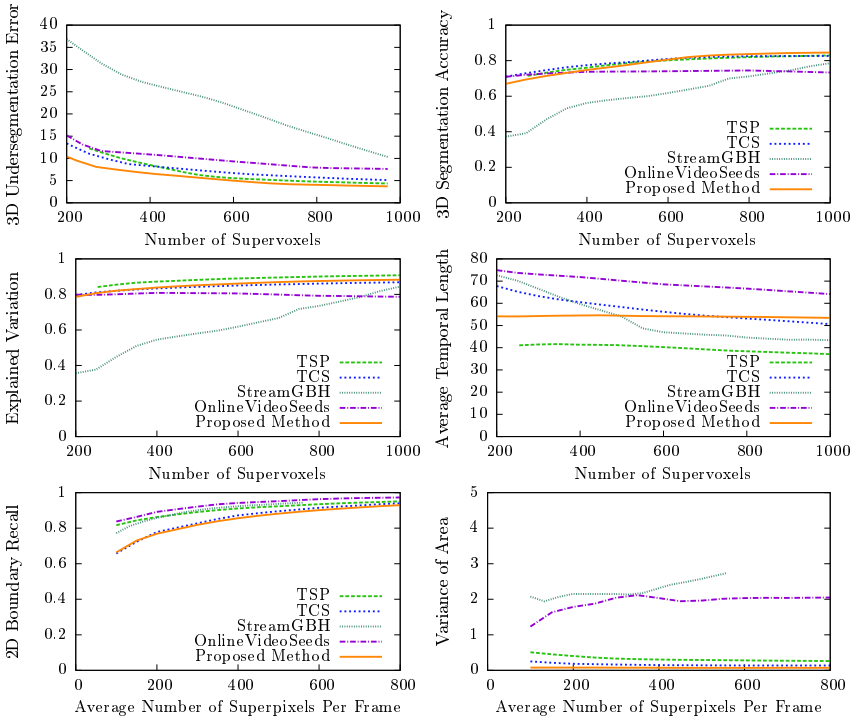


Fig. 8. Benchmark results for the dataset from [29] for state of the art supervoxel and superpixel algorithms for video content. Please note that the 2D boundary recall and the variance of area are plotted over the average number of superpixels per frame. Higher values are better except for the 3D undersegmentation error and variance of area.

Superpixels (TSP) [16] and *OnlineVideoSeeds* [17]. For StreamGBH, TSP, and OnlineVideoSeeds, the source code of the algorithms was publicly available. We used the sources from the authors' websites to produce the following results. For TCS, we used the original version that was the basis for [18]. Whenever possible, the parameters were set as mentioned in the authors' publications or documentation. For the dataset [29], the benchmarks results are depicted in Fig. 8 and for the dataset [28] with ground truth segmentations from [30] results are shown in Fig. 9.

For [29] our algorithm performs best in *UE* and *VoA* and is also slightly better in *SA* for higher numbers of supervoxels while performing worst in *BR* with comparable results to TCS (see Fig. 8). On the second dataset, which is much larger and more diverse, our method performs best in *SA*, *EV* as well as *VoA* and for higher numbers of supervoxels also in *UE* (see Fig. 9). The unusual behavior of OnlineVideoSeeds for *UE* may have its roots in the code provided by the authors. The number of histogram bins is hard coded for several fixed numbers of superpixels, which may not work well on the dataset provided by [28],

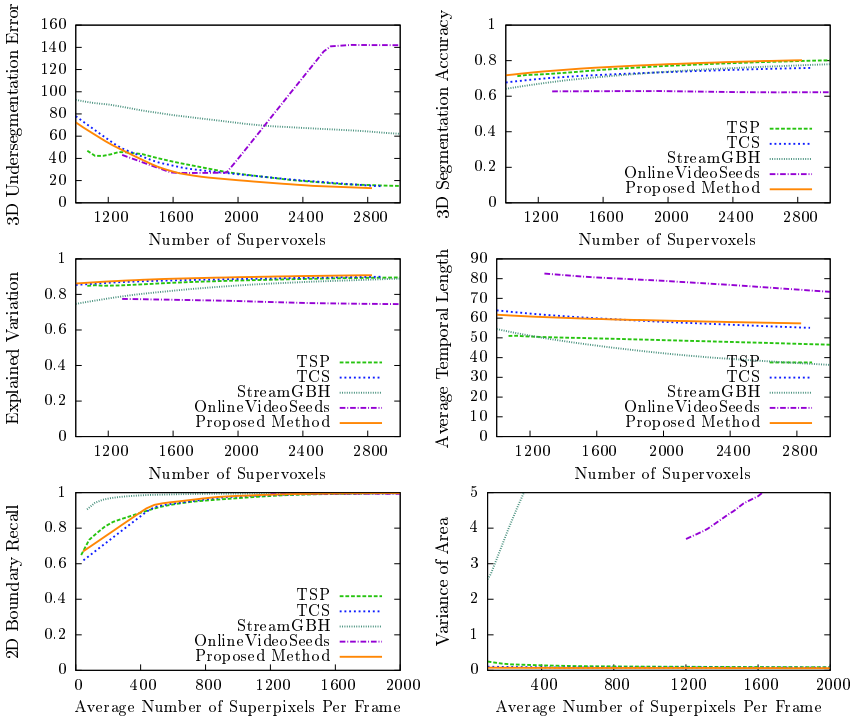


Fig. 9. Results generated for the dataset provided by [28] including scenes with diverse types of camera motion, motion blurring and non-rigid-motion. Please note the different abscissa. Higher values are better except for the 3D undersegmentation error and variance of area.

as it is has a higher resolution and is more complex than [29]. In the *VoA* diagram of Fig. 9 the graphs of StreamGBH and OnlineVideoSeeds rise approximately linearly to a *VoA* of 18.4 and 7.9, respectively, for 2000 superpixels per frame.

4 Conclusion

We presented a novel, contour-based approach to generate temporally consistent superpixels for video content. It is based on an EM framework performing the optimization only on the pixels at the superpixel boundaries and leverages the optical backward flow for the propagation of superpixel labels for the initialization of new frames. In combination both contributions help to preserve superpixel shapes over multiple frames leading to a steady and accurate superpixel constellation. The evaluation on standard benchmarks shows that our approach outperforms or produces comparable results to state of the art supervoxel and temporal superpixel approaches, even on datasets with different kinds of camera movement, non-rigid motion, and motion blur.

References

1. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV. (2003) 10–17
2. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: CVPR. (2011) 3369–3376
3. Galasso, F., Cipolla, R., Schiele, B.: Video segmentation with superpixels. In: ACCV. (2012) 760–774
4. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: ICCV. (2011) 1323–1330
5. Djelouah, A., Franco, J.S., Boyer, E., Le Clerc, F., Pérez, P.: Multi-view object segmentation in space and time. In: ICCV. (2013) 2640–2647
6. Vogel, C., Schindler, K., Roth, S.: Piecewise Rigid Scene Flow. In: ICCV. (2013) 1377–1384
7. Zhang, J., Kan, C., Schwing, A.G., Urtasun, R.: Estimating the 3D Layout of Indoor Scenes and Its Clutter from Depth Sensors. In: ICCV. (2013) 1273–1280
8. van den Hengel, A., Dick, A., Thormählen, T., Ward, B., Torr, P.H.S.: VideoTrace. ACM TOG **26** (2007) 86
9. Tighe, J., Lazebnik, S.: Superparsing. IJCV **101** (2012) 329–349
10. Roig, G., Boix, X., Nijs, R.D., Ramos, S., Kuhnlenz, K., Gool, L.V.: Active MAP Inference in CRFs for Efficient Semantic Segmentation. In: ICCV. (2013) 2312–2319
11. Jain, A., Chatterjee, S., Vidal, R.: Coarse-to-Fine Semantic Video Segmentation Using Supervoxel Trees. In: ICCV. (2013) 1865–1872
12. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV. (2005) 654–661
13. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR. (2010) 2141–2148
14. Veksler, O., Boykov, Y., Mehrani, P.: Superpixels and supervoxels in an energy optimization framework. In: ECCV. (2010) 211–224
15. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. TPAMI **34** (2012) 2274–2282
16. Chang, J., Wei, D., Fisher, J.W.: A video representation using temporal superpixels. In: CVPR. (2013) 2051–2058
17. Van den Bergh, M., Roig, G., Boix, X., Manen, S., Van Gool, L.: Online video seeds for temporal window objectness. In: ICCV. (2013) 377–384
18. Reso, M., Jachalsky, J., Rosenhahn, B., Ostermann, J.: Temporally consistent superpixels. In: ICCV. (2013) 385–392
19. Levinshtein, A., Sminchisescu, C., Dickinson, S.: Spatiotemporal closure. In: ACCV. (2011) 369–382
20. Zitnick, C.L., Jojic, N., Kang, S.B.: Consistent segmentation for optical flow estimation. In: (ICCV). (2005) 1308–1315
21. Xu, C., Xiong, C., Corso, J.J.: Streaming hierarchical video segmentation. In: ECCV. (2012) 626–639
22. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV **59** (2004) 167–181
23. Bergh, M., Boix, X., Roig, G., Capitani, B., Gool, L.: SEEDS: Superpixels Extracted via Energy-Driven Sampling. In: ECCV. (2012) 13–26

24. Xu, C., Corso, J.J.: Evaluation of super-voxel methods for early video processing. In: CVPR. (2012) 1202–1209
25. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. TPAMI **33** (2011) 898–916
26. Schick, A., Fischer, M., Stiefelhagen, R.: Measuring and evaluating the compactness of superpixels. In: ICPR. (2012) 930–934
27. Schick, A., Fischer, M., Stiefelhagen, R.: An evaluation of the compactness of superpixels. Pattern Recognition Letters **43** (2014) 71–80
28. Sundberg, P., Brox, T., Maire, M., Arbelaez, P., Malik, J.: Occlusion boundary detection and figure/ground assignment from optical flow. In: CVPR. (2011) 2233–2240
29. Chen, A., Corso, J.J.: Propagating multi-class pixel labels throughout video frames. In: WNYIPW. (2010) 14–17
30. Galasso, F., Nagaraja, N.S., Cárdenas, T.J., Brox, T., Schiele, B.: A unified video segmentation benchmark: Annotation, metrics and analysis. In: ICCV. (2013) 3527–3534
31. Moore, A.P., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: CVPR. (2008) 1–8
32. Perbet, F., Maki, A.: Homogeneous superpixels from random walks. In: MVA. (2011) 26–30
33. Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. PhD thesis, Massachusetts Institute of Technology (2009)
34. Horn, B.K.P., Schunck, B.G.: Determining optical flow. AI **17** (1981) 185–203