Temporally Consistent Superpixels

Matthias Reso[†], Jörn Jachalsky[‡], Bodo Rosenhahn[†], Jörn Ostermann[†]
[†]Leibniz Universität Hannover, Germany
reso@tnt.uni-hannover.de
[‡]Technicolor Research & Innovation, Germany
joern.jachalsky@technicolor.com

Abstract

Superpixel algorithms represent a very useful and increasingly popular preprocessing step for a wide range of computer vision applications, as they offer the potential to boost efficiency and effectiveness. In this regards, this paper presents a highly competitive approach for temporally consistent superpixels for video content. The approach is based on energy-minimizing clustering utilizing a novel hybrid clustering strategy for a multi-dimensional feature space working in a global color subspace and local spatial subspaces. Moreover, a new contour evolution based strategy is introduced to ensure spatial coherency of the generated superpixels. For a thorough evaluation the proposed approach is compared to state of the art supervoxel algorithms using established benchmarks and shows a superior performance.

1. Introduction

The idea to utilize superpixels as primitives for image analysis and processing was introduced by Ren and Malik in [14]. In the following years, several authors proposed different approaches to generate superpixels with special properties from still images [12, 23, 9, 1, 19, 13]. They all follow the common principle to group spatially coherent pixels sharing similar low-level features like color or texture into so called superpixels. This grouping leads to a major reduction of the image primitives, which results in an increased computational efficiency for subsequent processing steps and allows for more complex algorithms computationally infeasible on pixel level [14]. Another benefit is the creation of a spatial support for region-based features [6]. There are a wide variety of applications utilizing superpixels including tracking [20], image parsing [16], depth-map enhancement [24], 3D geometry reconstruction [6] and video segmentation [18].

Especially for video applications, the usage of superpixels instead of raw pixel data is beneficial, as otherwise a vast amount of data has to be handled. But until recently, superpixel algorithms were mainly targeting still images. When applied to video sequences, this leads to volatile and flickering superpixel contours even if there are only slight changes between consecutive frames. Moreover, by design they omit the temporal connection between superpixels in successive images. Consequently, the same image regions in consecutive frames are not consistently labeled. As an example, Figure 1 shows the benefits of consistent labels, which are considered to be valuable for a wide range of video applications.

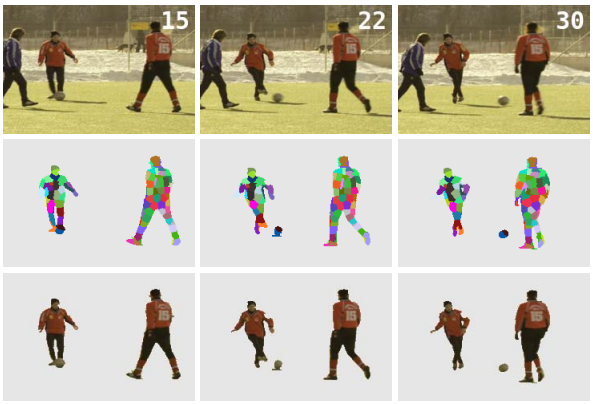

Figure 1. Top row: Original sequence with frame numbers. Mid row: Subset of superpixels manually selected in frame 15 and shown as color-coded labels. The superpixels in the frames 22 and 30 are generated with our approach and are displayed using the same label colors to indicate temporal consistency.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

Hence, in this work we propose a new approach to generate superpixels that ensures temporal consistency and provides a consistent labeling. It can be seen as one way for spatio-temporal over-segmentation. We call our approach *temporally consistent superpixels* (TCS).
The **key contributions** of our paper are:

- an approach for temporally consistent superpixels based on energy-minimizing clustering utilizing a novel hybrid clustering strategy working in a global color subspace and local spatial subspaces
- a new contour evolution based strategy to ensure spa-

tially coherent superpixels generated with clustering based approaches

The remainder of the paper is organized as follows: In Section 2, we shortly summarize the previous work on spatio-temporal over-segmentation. Subsequently, in Section 3, we briefly explain the generation of superpixels using energy-minimizing clustering that is extended in Section 4, where we present our approach for temporally consistent superpixels. In Section 5 our approach is thoroughly evaluated and compared to other state of the art approaches using established benchmarks before concluding the paper in Section 6.

2. Related Work

In [19, 5, 8, 1] the superpixel idea is extended from the still image to the video domain starting to take the issue of temporal consistency into focus. One proposal was to generate so called supervoxels by grouping adjacent voxels in the video volume, which are similar *e.g.* in terms of color. These supervoxels connect coherent image regions or segments over multiple frames.

The relation between supervoxels and temporally consistent superpixels can be described in the following way: Temporally consistent 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 non-contiguous segments. Nevertheless, these approaches are the most akin methods and therefore will serve as comparison in Section 5.

In [5] an approach for hierarchical video segmentation was proposed, which is referred to as GBH in [21]. It is based on a twofold application of the graph based image segmentation approach presented in [4], which tends to generate rather small segments in the vicinity of edges and in highly structured areas. In [22] a solution based on GBH was presented that provides streaming capabilities by using a Markov assumption (sGBH). Moreover, [21] presents an overview of available supervoxel methods and proposed corresponding benchmark metrics that are extensions of the established superpixel metrics.

The SLIC supervoxel approach [1] as well as the approach presented in [19] enforce a rather short temporal duration of the generated supervoxels, either implicitly or explicitly. Therefore, the superpixels are temporally consistent but only over a short range of frames.

A different approach towards spatio-temporal superpixels, which utilizes optical flow information, was published in [8]. This reduces to some extent the noisy flickering of the superpixels from one frame to the next. Still the superpixels are only generated on a per frame basis and there is no explicit strategy to handle disocclusions and new objects entering the scene.

3. Superpixels based on Energy-minimizing Clustering

As our approach for temporally consistent superpixels is based on energy-minimizing clustering (*c.f.* [24, 1, 23]), we will briefly outline the principles of clustering for the generation of superpixels.

For the clustering, pixels of an image are seen as data points in a multi-dimensional feature space, in which each dimension corresponds to a color channel or image coordinate of the pixels. Superpixels are represented by clusters in this multi-dimensional feature space and each data point can only be assigned to one cluster. This assignment finally determines the over-segmentation and thus the superpixel generation.

In order to find an optimal solution for this assignment problem, an energy function E_{total} is defined, which sums up the energy $E(n,k)$ that is needed to assign a data point $n \in \mathcal{N}$ to a cluster $k \in \mathcal{K}$:

$$E_{total} = \sum_{n \in \mathcal{N}} E(n,k)\,, \tag{1}$$

where $\mathcal{N}$ is the set of pixels in the image and $\mathcal{K}$ is the set of clusters representing the superpixels. The energy $E(n,k)$ can be further refined as the weighted sum of a color-difference related energy $E_c(n,k)$ and a spatial-distance-related energy $E_s(n,k)$:

$$E(n,k) = (1-\alpha)E_c(n,k) + \alpha E_s(n,k) \tag{2}$$

The energy $E_c(n,k)$ is directly proportional to the Euclidean distance between a data point n and the color center of cluster k in the CIELAB color space. Likewise $E_s(n,k)$ is proportional to the Euclidean distance of the spatial position of n and the spatial position of the center of cluster k. In order to make the results independent from the image size, the spatial distance is scaled with the factor $\sqrt{|\mathcal{K}|/|\mathcal{N}|}$ where $|\cdot|$ is the number of elements in a set. With the parameter α that was introduced in [15] the user can steer the segmentation results to be more compact or more sensitive to fine-grained image structures. For a given number of clusters $|\mathcal{K}|$ and a user-defined α, an optimal over-segmentation in terms of energy can be determined by finding a constellation of clusters that minimizes (1).

The assignment problem is solved by applying the iterative Lloyd's algorithm [10], which converges to a locally optimal solution. The initial spatial position of the cluster centers is grid-like including a perturbing of the spatial centers towards the lowest gradient in a 3×3 neighborhood (see [9, 1]). To minimize the energy term (1), the algorithm iterates two alternating steps: the *assignment*-step and the

update-step. In the *assignment*-step, each data point n is assigned to the cluster, for which the energy term (2) has its minimum given the set of clusters $\mathcal{K}$. Based on these assignments, the parameters of the cluster centers are re-estimated in the *update*-step by calculating the mean color and mean position of their assigned pixels. The iteration stops when no changes in the *assignment*-step are detected or a maximum number of iterations have been performed.

As the spatial extent of the superpixels is known to be limited a priori, it is sufficient in the *assignment*-step to search for pixels only in a limited search window around each cluster center. This leads to a significant reduction of the computational complexity [1]. In order to enforce the spatial connectivity of the resulting segments, a post-processing step assigns split-off fractions, which are not connected to the main mass of the corresponding superpixel, to its nearest directly connected superpixel.

4. Temporally Consistent Superpixels

4.1. General Idea

To be able to generate temporally consistent superpixels, we separate the original five-dimensional feature space for the superpixels into a global color subspace comprising multiple frames and multiple local spatial subspaces on frame level following the idea that the color clustering is done globally and the spatial clustering locally. As a consequence, each temporally consistent superpixel has a single color center 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. The motivation for this approach is the observation that the color of matching image regions occupied by a temporally consistent superpixel over multiple frames does not change rapidly in most cases. Therefore, the mean colors of the associated superpixels are –in a first approximation– almost constant over multiple frames. In contrast, the positions can vary significantly depending on the motion in the scene.

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 and F so called *future* frames and one *current* frame with $W = F+P+1$. An example with $W = 5$ and $P = F = 2$ is depicted in Figure 2. In this example, the frame t 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 it influences the superpixel generation in the *current* frame and *future* frames. The segmentation of *current* and

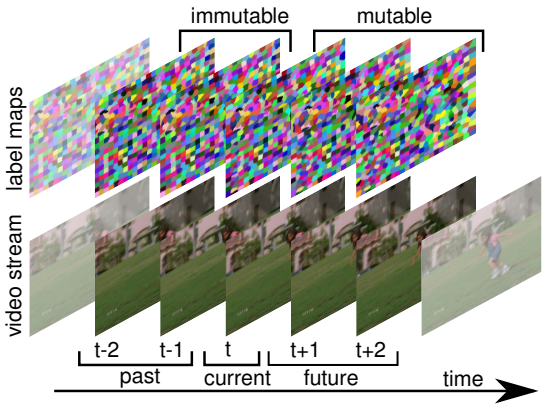

Figure 2. Sliding window approach. Bottom row: Frames in sliding window (non-transparent) are divided into three groups. Top row: Corresponding label maps.

future frames is still mutable and thus can change during the optimization. The *future* frames help to adapt to changes in the scene, whereas the *past* frames are conservative and try to preserve the superpixel color clustering found. 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.

4.2. Hybrid Clustering Approach

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 general idea of global color and local spatial centers. First, we extend the energy term (2) with the frame index τ as the energy E_s is now proportional to the distance to the spatial centers in the local frame:

$$E(n,k,\tau) = (1-\alpha)E_c(n,k) + \alpha E_s(n,k,\tau)\,. \qquad (3)$$

Second, we need to sum over all the frames in the sliding window to calculate the total energy with regard to the *current* frame t:

$$E_{total}(t) = \sum_{\tau=t-P}^{t+F} \sum_{n \in N(\tau)} E(n,k,\tau)\,, \qquad (4)$$

where $\mathcal{N}(\tau)$ is the set of pixels in the frame τ. Third, the iterative optimization scheme is adopted to the hybrid approach as explained below. Algorithm 1 shows the principal approach for the hybrid clustering for I iterations.

After each shift of the sliding window, a number of I iterations of the hybrid clustering algorithm is performed. In the *assignment*-step, each pixel of the mutable frames, *i.e.* the *current* and the *future* frames, is assigned to one cluster (and thus to one temporally consistent superpixel), for

input : W frames in sliding window; $\mathcal{K}$
output: Assignment of pixels to clusters; updated $\mathcal{K}$

for $i \in I$ **do**
  foreach *mutable frame in sliding window* **do**
    assign pixels to clusters;
  end
  forall the *frames in sliding window* **do**
    if *mutable frame* **then**
      update local spatial centers;
    end
    accumulate global color information;
  end
  update global color centers;
end

Algorithm 1: Hybrid clustering

which the energy term (3) has its minimum. The color-difference related energy E_c is proportional to the Euclidean distance to the global color center and the spatial-distance-related energy E_s is proportional to the Euclidean distance to the local spatial center on frame level.

In the *update*-step, for each cluster a new global color center is calculated using the accumulated color information of those pixels in all frames in the sliding window, which are assigned to this cluster. The spatial centers are updated locally per frame using only the image coordinates of the pixels that are assigned to this cluster in the corresponding frame. For our experiments we use $I = 5$ 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. Contour Evolution

As already stated in Section 3, the clustering does not necessarily lead to spatially coherent superpixels. Thus, a post-processing step is required to ensure the spatial connectivity of the pixels. Contour evolution approaches like [11, 15] can overcome this drawback.

In addition, in [15] it was stated that the post-processing method proposed in [1] assigns the isolated superpixel fragments to arbitrary neighboring segments without considering any similarity measure between the isolated fragments and the neighboring segments. In our approach, we also utilize contour evolution. But in contrast to [11, 15] we use it as a post-processing step. In this way, we combine the fast initial convergence of a clustering approach and the connectivity-preserving properties of the contour evolution. Therefore, we can avoid the high number of iterations to find a locally optimal solution required by conventional contour evolution approaches.

In our approach, the contour evolution step is applied for those frames transitioning from the *current* to the first *past* frame, *i.e.* changing the position from t to $t-1$ in the sliding window. Thereby, we determine for each cluster the largest spatially coherent part and set the unconnected fragments of the cluster to unassigned and mark them as mutable. Figure 3 shows a small example. The contours of those clusters adjacent to a region marked as mutable can evolve into this region during the contour evolution iterations. In the right image of Figure 3 this can be the contours of the red and the yellow cluster. Only those pixels that are in a region marked as mutable are processed, the other pixels are unaffected. In each iteration of the contour evolution the cluster assignment for those pixels at a boundary within a region marked as mutable can be changed. The assignment of a pixel is changed if the pixel has no assignment yet. Then, it is assigned to the cluster of one of its adjacent pixels, which minimizes the energy term (3). In addition, an assignment of a pixel is changed to the cluster of one of its adjacent pixels if the energy term (3) is smaller for this cluster than for the one it was previously assigned to. The iterations are stopped if all pixels in the marked regions are assigned to a cluster and no further changes at the boundaries occur. The resulting spatially coherent clusters are the final superpixels.

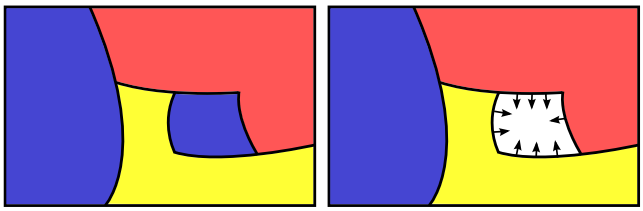

Figure 3. Contour evolution. Left: Clusters after hybrid clustering. The blue cluster is not completely spatially coherent. Right: Small split-off fragment is set to unassigned and marked as mutable. The contours of the red and yellow cluster can evolve into the unassigned region (Best viewed in color).

4.4. Initialization

As the position of matching image regions and thus the superpixel position can differ in consecutive frames, a concurrent initialization of all frames in the sliding window is not practicable. Therefore, we propose a successive filling of the sliding window according to the following scheme.

At the start, the sliding window is empty. The first frame of a video sequence to enter the sliding window is initialized by distributing $|\mathcal{K}|$ spatial cluster centers in a grid-like structure on the frame similar to [9, 1] including a perturbation of the spatial centers towards the lowest gradient in a 3×3 neighborhood. This frame is positioned at index $t+F$ in the sliding window. As a *future* frame its segmentation is mutable. For this *future* frame the energy-minimizing clustering with regard to (3) is performed.

Then, the sliding window is shifted, whereby a new

frame enters the window at position $t+F$ and the old frame is moved to $t+F-1$. The spatial centers of frame $t+F$ are initialized by projecting the spatial centers of frame $t+F-1$ into frame $t+F$ using optical flow. As in [8], a weighted average of the dense optical flow computed over all pixels assigned to the center is used. For our experiments we calculate the dense optical flow using the Horn-Schunck method [7]. During our evaluations, we found out that the results of our approach are almost independent from the optical flow algorithm utilized. After the projection of the centers is done, the energy-minimizing clustering is performed again.

This procedure is repeated until the sliding window is completely filled. Then the generation of temporally consistent superpixel can further proceed. Thereby, the sliding window is repeatedly shifted as described above until the video sequence is completely processed. The superpixel segmentations of frame $t-1$ of the sliding window are stored, which is the first *past* frame and thus immutable.

4.5. Structural Changes in the Video Volume

In general, the generated superpixels should capture the temporal consistency inherent in the video volume as completely as possible. But the continuous adaptation of the superpixels to the video content can lead to steadily growing or shrinking superpixels that tend to violate the constraint of a rather homogeneous size. This effect can be observed in Figure 4 that depicts the temporally consistent label maps of two segmented frames from the soccer sequence that were generated without utilizing any method to ensure a homogeneous size of the superpixels over time. One can see that the superpixels in the right image are squeezed together on the left side of the soccer player while they are huge on the right side. Similar effects can be found using the method described in [8].

A trivial solution to minimize this effect is to enforce a rather short temporal duration of the generated superpixels (see Section 2). But in order to meet the size constraints without enforcing a short temporal consistency, we try to solve the following constrained energy minimization:

$$\text{Minimize } E_{total}(t) \text{ s.t.} \quad (5)$$
$$\forall k \in \mathcal{K}, \tau \in [t-P;t+F] : A_{min} < A(k,\tau) < A_{max},$$

where $A(k,\tau)$ is the number of pixels assigned to cluster k in frame τ. We implemented this constrained energy minimization in a first simple but effective approach in our sliding window framework. To meet the constraints the number of pixels assigned to a cluster is traced in two consecutive *future* frames. The growth and decrease in size of the clusters is predicted for the next frames using a linear growth assumption. If the predicted number of pixels assigned to a cluster is greater than A_{max} in frame $\tau = t+F+2$ (outside the sliding window) the cluster is split in two. Thereby, each spatial center of the cluster is replaced by two new spatial centers in all *future* frames and its color center is duplicated. The new spatial centers are shifted in opposite directions towards the biggest eigenvector of the spatial distribution of the cluster similar to the superpixel splitting in [23]. In case that –based on this prediction– the number of pixels assigned to a cluster would be lower than A_{min} in frame $\tau = t+F+2$, the cluster is terminated by removing its spatial centers from the *future* frames. To keep the number of clusters (thus superpixels) constant over time, the number of splits and terminations should be equal. If this is not the case the initial number of superpixels is restored by splitting or terminating the biggest or smallest clusters, respectively. For our experiments in Section 5 we chose A_{min} to be 0 and A_{max} to be $1.5 \cdot \bar{A}$ where $\bar{A}$ is the targeted average cluster size.

Figure 4. Label maps of the frames 1 and 60 of the soccer sequence segmented with temporal consistency but without a method to cope with structural changes in the video volume.

5. Experiments

5.1. Experimental Setup and Performance Metrics

We implemented our approach for temporally consistent superpixels (TCS) in MATLAB and compared it with state of the art methods for spatio-temporal over-segmentation. For the experiments, we use the SegTrack data set [17] and the Chen data set [3]. The SegTrack data set contains six video clips with 21 to 71 frames and binary ground truth segmentation data. The Chen data set is a collection of eight video clips with around 80 frames per clip and a multi-label ground truth segmentation.

We compared our approach (TCS) against two state of the art supervoxel methods: the SLIC approach for supervoxels (SLIC) [1] and the streaming hierarchical video segmentation (sGBH) [22]. sGBH was selected as a top-performing candidate for the class of streaming capable supervoxel approaches and SLIC was selected as a top-performing candidate for the class of clustering based supervoxel approaches.

For both sGBH and SLIC, the implementations provided on the authors' websites were used to generate multiple spatio-temporal over-segmentations with different levels of detail, *i.e.* different numbers of supervoxels. Other parameters for sGBH and SLIC were left unchanged from the de-

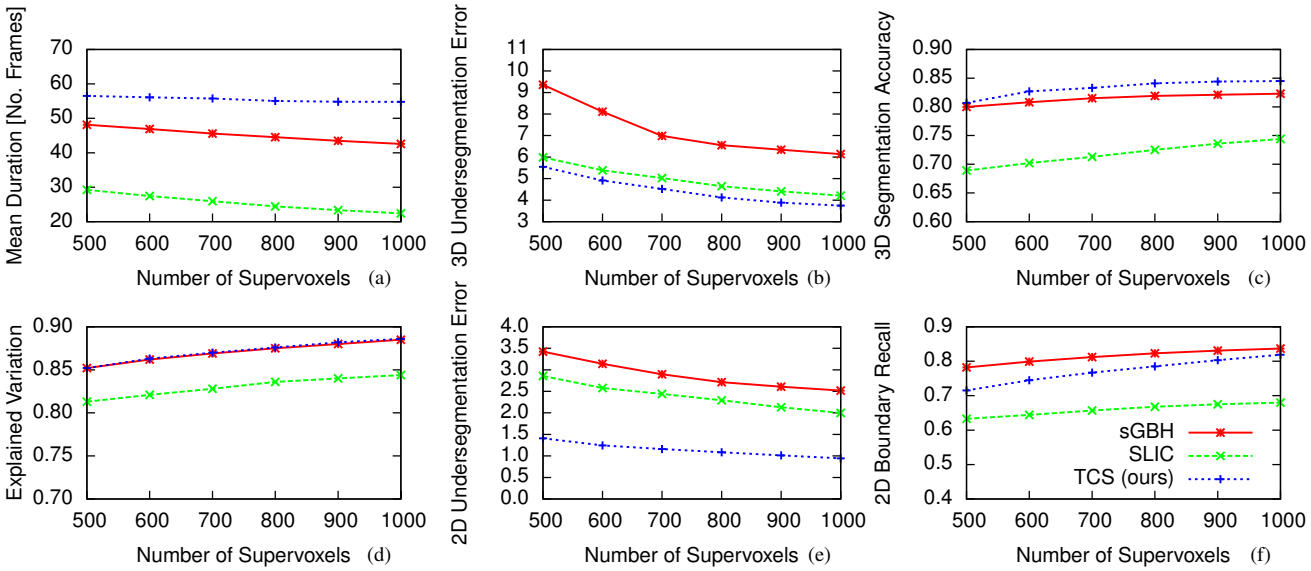

Figure 5. Results for Chen data set. All diagrams are plotted over the number of supervoxels (Best viewed in color).

fault values provided by the authors. As described in Section 2, temporally consistent superpixels can be stacked to obtain supervoxels. For a comparison, we selected the values for $|\mathcal{K}|$ in such way that the number of generated supervoxels is approximately identical with those of sGBH and SLIC and set α in (3) to 0.96. To evaluate the performance of our method we used the following performance metrics for supervoxels and superpixels. The benchmark results were produced using the code provided by [2] and [21].

Mean Duration measures the duration of the generated supervoxels or temporally consistent superpixels in terms of number of frames. *2D* and *3D Undersegmentation error* were introduced in [9] and [21], respectively. The undersegmentation error is a metric to evaluate how precisely the ground truth segments can be reproduced using the provided segmentation. Thereby, the number of pixels (or voxels) is determined that exceed the boundary of the ground truth. It should be noted, that a large mean duration of a spatio-temporal segment is only valuable in combination with a low 3D undersegmentation error. *3D Segmentation Accuracy* was introduced by [21]. It measures how well a segmentation fills out the ground truth. *2D Boundary Recall* quantifies how precise the alignment of the segmentation with the ground truth boundaries is. *Explained Variation* was introduced by [12]. It measures how well the data of the original pixels can be reproduced with the over-segmentation. *Variance of Area* (VoA) and *Mean Isoperimetric Quotient* (Q) were proposed in [13] to quantify the homogeneity of the size and the compactness of the generated superpixels. The iso-perimetric quotient was adopted in [15] to calculate the *Superpixel Compactness*.

5.2. Benchmark Results

The Figures 5 and 6 show the results for the performance metrics over the number of supervoxels as common parameter for the three compared approaches and the two benchmark data sets Chen (see Figure 5) and SegTrack (see Figure 6). In Figures 5a and 5b as well as Figures 6a and 6b it can be seen that our approach (TCS) generates the spatio-temporal segments with the longest temporal duration while producing the best over-segmentation with respect to the 3D undersegmentation error for the Chen data set and a comparable over-segmentation for the Segtrack data set. It should be added that the number of *past* frames in the sliding window has a negligible effect on the mean duration while the undersegmentation error, up to some extent, decreases with an increasing number of *past* frames. This behavior is inline with the description in Section 4.1 that the *past* frames preserve the color of superpixels and thus prevent them from *e.g.* overlapping object boundaries. For the sake of clarity, we show in the diagrams the results for TCS only with the number of *past* frames $P=12$. The number of *future* frames was fixed to $F=2$.

Figures 5c and 5d as well as Figures 6c and 6d also show that our approach (TCS) provides better or competitive segmentation results with respect to the 3D segmentation accuracy and the explained variation when compared to SLIC and sGBH. Finally, in Figures 5e and 6e it can be seen that TCS produces the lowest 2D undersegmentation error among the three approaches while producing only a slightly less precise but still highly competitive 2D boundary recall when compared to sGBH as depicted in Figures 5f and 6f. This is remarkable as sGBH produces considerably less

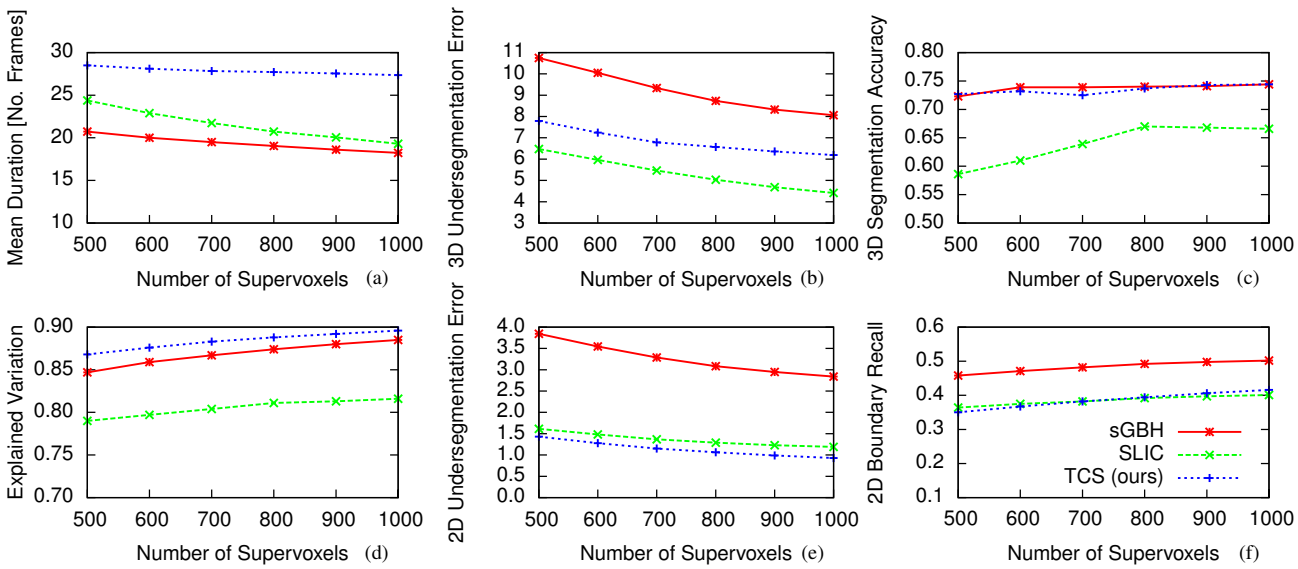

Figure 6. Results for SegTrack data set. All diagrams are plotted over the number of supervoxels (Best viewed in color).

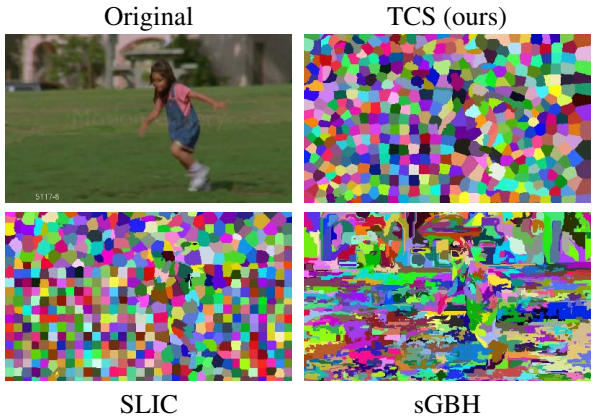

Figure 7. Comparison of color-coded label maps. All frames have approximately 700 superpixels. The label maps show that TCS and SLIC produce more compact superpixels than sGBH. (Best viewed in color)

compact superpixels, which –by intuition– makes it easier to capture fine-grained details compared to the more compact superpixels of SLIC and TCS. The lack of compactness can be seen in a qualitative manner in Figure 7 where the color-coded label maps of an example frame generated by the three approaches are shown. The visual impression gained from Figure 7 is confirmed by the variance of area, the iso-perimetric quotient and the superpixel compactness that are depicted in Table 1. For each approach a level of detail was selected that generates a comparable number of superpixels or sliced supervoxels with a mean area of approximately 100 pixels. It can be seen that TCS produces

Method	Mean Area	VoA	Q	Compactness
SLIC	98	0.44	0.67	0.67
sGBH	98	1.69	0.48	0.39
TCS (ours)	100	0.15	0.67	0.69

Table 1. Variance of area (VoA), average iso-perimetric quotient Q and superpixel compactness calculated for the entire data set of Chen for an approximately similar level of detail (100 pixel per superpixel).

the lowest variance of area while the iso-perimetric quotient and the superpixel compactness are comparable to SLIC. This indicates that the superpixels generated by TCS and SLIC are more homogeneous in size and more compact in shape than those of sGBH.

With the compactness parameter α in (3) TCS could be made more sensitive to fine-grained details achieving a better 2D boundary recall at the price of a lower compactness. But as stated in [14, 9, 15] it is beneficial to have compact superpixels. It *e.g.* allows for a better capturing of spatially coherent information. In addition, it simplifies the processing in subsequent processing steps, as *e.g.* compact superpixels tend to have a lower average number of neighbors which eases the evaluation of neighborhood relations, and further calculations, *e.g.* feature extraction, can be performed on almost equally sized segments.

5.3. Complexity Considerations

In [1], the SLIC superpixel approach for still images is approximated to have a complexity of $\mathcal{O}(|\mathcal{N}|)$, where $|\mathcal{N}|$ is the numbers of pixels per image. Using this approxima-

tion, our approach for temporally consistent superpixels has a complexity of $\mathcal{O}(|\mathcal{N}|WV)$, where W is the sliding window size in frames and V is the number of frames in the video sequence. As it holds that $W \ll V < |\mathcal{N}|$ for reasonably long video sequences (*e.g.* full feature film length) and frames with mega-pixel resolution, the complexity of our approach TCS is $\mathcal{O}(|\mathcal{N}|V)$ as it is for the SLIC supervoxel approach. Compared to [22] that has a complexity of $\mathcal{O}(|\mathcal{N}|V \log |\mathcal{N}|)$ it shows that our approach is more efficient with regard to the computational complexity.

6. Conclusion and Future Work

In this paper, we propose a novel approach for spatio-temporal over-segmentation for video content called *temporally consistent superpixels* (TCS). It performs an energy-minimizing clustering utilizing a hybrid clustering strategy for a multi-dimensional feature space that is separated into a global color subspace and multiple local spatial subspaces. Moreover, a new contour evolution based strategy is introduced that ensures spatial coherency of the generated superpixels. 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 color clustering found. An additional benefit of the sliding window approach is the resulting capability of short-delay streaming and processing arbitrarily long video sequences. In a thorough, in-depth evaluation based on established benchmarks, the proposed approach was compared to state of the art supervoxel methods and provided a superior performance. These results make the approach an excellent basis for all tasks requiring temporal consistency and a high segmentation accuracy as *e.g.* video segmentation and tracking applications.

In future work, we want to investigate the introduction of additional cues or features for the superpixel generation. Moreover, we would like to enhance the iterative energy-minimization scheme to intrinsically cope with structural changes in the video volume.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012. 1, 2, 3, 4, 5, 7

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 6

[3] A. Chen and J. Corso. Propagating multi-class pixel labels throughout video frames. In *WNYIPW*, pages 14–17, 2010. 5

[4] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 2

[5] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010. 2

[6] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, pages 654–661, 2005. 1

[7] B. K. P. Horn and B. G. Schunck. Determining optical flow. *AI*, 17(1-3):185–203, 1981. 5

[8] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Spatiotemporal closure. In *ACCV*, pages 369–382, 2011. 2, 5

[9] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *TPAMI*, 31(12):2290–2297, 2009. 1, 2, 4, 6, 7

[10] S. Lloyd. Least squares quantization in PCM. *TOIT*, 28(2):129–137, 1982. 2

[11] R. Mester, C. Conrad, and A. Guevara. Multichannel segmentation using contour relaxation: fast super-pixels and temporal propagation. In *SCIA*, pages 250–261, 2011. 4

[12] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *CVPR*, pages 1–8, 2008. 1, 6

[13] F. Perbet and A. Maki. Homogeneous superpixels from random walks. In *MVA*, pages 26–30, 2011. 1, 6

[14] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, pages 10–17, 2003. 1, 7

[15] A. Schick, M. Fischer, and R. Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *ICPR*, pages 930–934, 2012. 2, 4, 6, 7

[16] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365, 2010. 1

[17] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking with multi-label MRF optimization. In *BMVC*, pages 56.1–56.11, 2010. 5

[18] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, pages 268–281, 2010. 1

[19] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. In *ECCV*, pages 211–224, 2010. 1, 2

[20] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330, 2011. 1

[21] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, pages 1202–1209, 2012. 2, 6

[22] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, pages 626–639, 2012. 2, 5, 8

[23] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha. Structure-sensitive superpixels via geodesic distance. In *ICCV*, pages 447–454, 2011. 1, 2, 5

[24] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1):49–65, 2007. 1, 2