Interactive Segmentation of High-Resolution Video Content using Temporally Coherent Superpixels and Graph Cut

Matthias Reso[1], Björn Scheuermann[1], Jörn Jachalsky[2],
Bodo Rosenhahn[1] and Jörn Ostermann[1]

[1]Leibniz Universität Hannover, Germany
{reso,scheuerm,rosenhahn,ostermann}@tnt.uni-hannover.de
[2]Technicolor Research & Innovation Hannover, Germany
joern.jachalsky@technicolor.com

Abstract. Interactive video segmentation has become a popular topic in computer vision and computer graphics. Discrete optimization using maximum flow algorithms is one of the preferred techniques to perform interactive video segmentation. This paper extends pixel based graph cut approaches to overcome the problem of high memory requirements. The basic idea is to use a graph cut optimization framework on top of temporally coherent superpixels. While grouping spatially coherent pixels sharing similar color, these algorithms additionally exploit the temporal connections between those image regions. Thereby the number of variables in the optimization framework is severely reduced. The effectiveness of the proposed algorithm is shown quantitatively, qualitatively and through timing comparisons of different temporally coherent superpixel approaches. Experiments on video sequences show that temporally coherent superpixels lead to significant speed-up and reduced memory consumption. Thus, video sequences can be interactively segmented in a more efficient manner while producing better segmentation quality when compared to other approaches.

1 Introduction

Segmenting an image into foreground and background is a basic step in many computer vision and computer graphics applications. Likewise the segmentation of videos can be fundamental for applications like object tracking, video editing or video content analysis [1–3]. Algorithms for image and video segmentation can be divided into two major categories. First, there are unsupervised techniques in which the algorithm decides which portion of an image or video is the region of interest that should be segmented. Second, the object or region of interest can be specified in a supervised manner e.g. by a human operator who roughly marks the region of interest as can be seen in Fig. 1 and 2. A special form of supervised segmentation is the interactive segmentation were the user helps the algorithm to segment a region of choice. This is done by repeatedly correcting the system's segmentation result using additional strokes on foreground or background. As

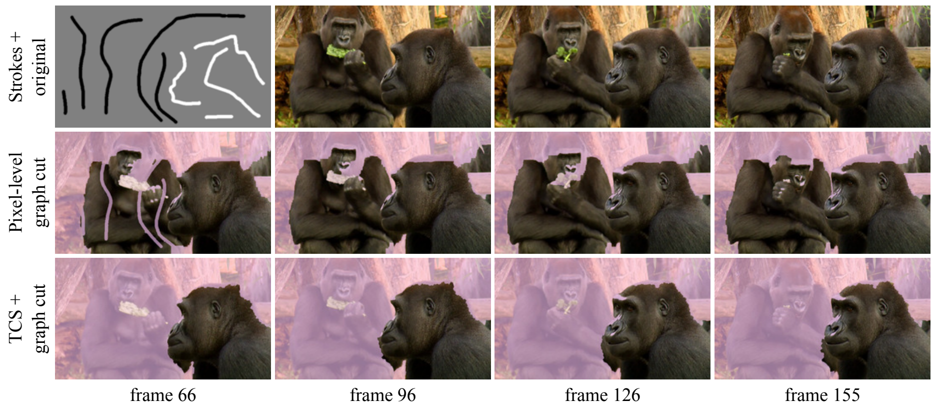

Fig. 1. Example of an interactive video segmentation. First row: user strokes and original images of the *zoo* sequence. The foreground strokes are painted white the background strokes black. Only the right gorilla should be segmented as foreground. Second row: graph cut applied on a pixel-level graph. Third row: segmentation result of graph cut applied on a graph built from temporally coherent superpixels created with [4].

the user has to wait for the system's response to do his or her additional strokes it is crucial for these systems to provide a minimal response time.

Popular representatives for these kinds of approaches have been proposed by Boykov et al. [5] and Rother et al. [6]. The segmentation problem is formulated using conditional random fields and the maximum a posteriori solution can be computed by optimizing a discrete energy function. For low-resolution images (e.g. 1Mpixels) these algorithms are able to segment an image on pixel-level in a run-time less than one second [6]. The approach of Boykov et al. [5] was also applied on a video sequence of 21 frames and a resolution of 255×189 pixels which is quite small regarding HD content that is ubiquitous even in home entertainment systems nowadays. With the emerging market of 4K content the amount of data that needs to be processed will rise accordingly. The amount of memory that is needed to segment a 120 frames sequence of HD-ready content (1280×720) with the approach of [5] can easily rise up to 30 GB which is near the upper bound of memory today's desktop computer can have built-in. As shown by Delong et al. [7] the run-time of the max-flow algorithm rises rapidly if the data of the problem does not fit into the physical memory. In [8] the authors presented a variable grouping based on the energy function to reduce the problem size and showed that the grouping helps to increase the segmentation quality while decreasing the run-time.

But this variable grouping can only be applied to some extent. To overcome this problem it has become popular to create an over-segmented representation of the input image which reduces the size of the graph that needs to be processed. These still image techniques are often called superpixels [9] and have also been applied to the task of video segmentation as publications like [10, 11] show.

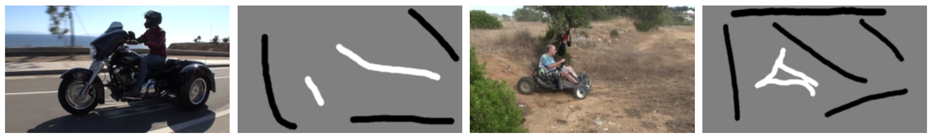

Fig. 2. Example user strokes for the sequences *harley* and *gokart* with the original video frames. Foreground is marked white and background black. Only for one frame per sequence user strokes were generated.

Beside, there has also been much effort to modify the still image approaches to directly process video content. The output if these techniques are small, compact image regions connected over the image plane and over time [12, 13, 4, 14] (we call them temporally coherent superpixels). These approaches have already shown promising results in unsupervised video segmentation frameworks as shown by Xu et al. [15].

For this reason the contribution of this paper is to examine the benefits temporally coherent superpixels can have in an interactive video segmentation framework. We show that, besides severely reducing the time needed to find an optimal cut of the graph, the temporally coherent superpixels based framework can additionally increase the segmentation quality. The rest of the paper is structured as follows. In Sec. 2 we give a short overview of the related work. Our framework is described in Sec. 3 while Sec. 4 gives experimental results and Sec. 5 concludes the paper.

2 Related work

Although unsupervised video segmentation frameworks can benefit from the usage of temporally coherent superpixels as well, this work focuses on their usage in an interactive segmentation framework. Therefore, this chapter will leave out the broadly populated sector of unsupervised video segmentation literature and will concentrate on the recent works on interactive video segmentation. One of the first authors addressing the task of interactive video segmentation were Wang et al. [16] with their framework called interactive video cutout. To boost the system's performance they used a two-staged hierarchical meanshift clustering as a preprocessing step. The graph built from the over-segmented spatio-temporal video volume was then cut by applying the popular graph cut framework of [5]. Their system allowed the user to paint strokes on single frames as well as along the time axis of the video volume and contained a postprocessing step for creating a spatio-temporally coherent alpha matte to visually optimize the blending of segmented objects onto a new background. A more recent framework named LIVEcut of Price et al. [17] individually over-segmented the video frames using Watershed and incorporated appearance, motion and shape features just like the framework proposed by Bai et al. [18] did. While the former used graph cut like [16] to find a global optimal cut through the graph the latter used local overlapping classifiers which were propagated to new frames using optical flow information.

In [19] Dondera et al. adopted the framework of [20] to produce superpixels on every video frame independently. Afterwards, a spatio-temporal superpixel graph is built-up using optical forward and backward flow information. Their main contribution is to use the information of an occlusion boundary detector to modify the superpixel graph on occlusion boundaries. Subsequently they partition the spatio-temporal superpixel graph into foreground and background using graph cut similar to [16] and [17]. To reduce the size of the problem, [21] as well as [22] proposed algorithms to reduce the graph without changing the optimal solution. These techniques can be combined with [16, 17, 19] as well as our proposed framework.

In contrast to the work of [19] that created superpixels in every frame independently, we use for our work the recently introduced class of algorithms [13, 4, 14] producing temporally coherent superpixels inherently equipped with spatio-temporal connections to the superpixels in different frames. Showing in our experiments that, besides speeding-up the calculation of the optimal cut, the segmentation quality is increased.

3 Interactive Video Segmentation using Superpixels

Our framework consists of two main components. First an over-segmentation step is performed generating temporally coherent superpixels. In the second component a spatio-temporal graph is built from the superpixel segmentation which is cut using a maximum flow algorithm. The maximum flow algorithm is initialized by user defined strokes. In the following sections the individual components of our framework are described.

3.1 Temporal Coherent Superpixels

In this section we will give a short overview of the algorithms for generating temporally coherent superpixels utilized in the experimental section of this paper. For a more detailed explanation please refer to the work of the original authors we will abbreviate from here as TSP [13], TCS [4] and OVS [14].

Temporal Superpixels The approach proposed by Chang et al. in [13] transfers the iterative clustering approach based on still images of [23] into a probabilistic framework introducing a generative superpixel model. Instead of using an EM-fashioned approach they maximize the log-likelihood-function

$$\mathcal{L}(z) = \log \left[p(z) \prod_k \prod_d p(\mu_{k,d}) \prod_i p(x_{i,d} | z_i, \mu_d) \right] + C \qquad (1)$$

for a superpixel segmentation z while implicitly estimating the parameters of their generative model. Every pixel i is described using a feature vector $x_i =$

$\left[a_i\ l_i\right]$ where a_i are the pixels' color values in CIELAB-color space and l_i is the pixels' location. The mean color and mean spatial values are denoted with μ. The indices k and d select the superpixel and feature vector dimension, respectively. C is a constant term and the likelihood $p(z)$ of a superpixel segmentation z includes a geometric distribution controlling the number of superpixels and their compactness. For invalid superpixel topologies (i.e. a single pixel is split from the rest of the superpixel) the likelihood is zero. The observations' likelihood for a pixel i being assigned to superpixel k is expressed as

$$p(x_i|z_i = k, \mu) = \prod_d \mathcal{N}(x_{i,d}; \mu_{k,d}, \sigma_d^2). \tag{2}$$

The mean value distributions $p(\mu_{k,d})$ are assumed to be uniform.

The log-likelihood is maximized by proposing label swaps of single pixels and proposing splitting and merging of superpixels. A proposal is only accepted if the log-likelihood function (1) is increased. After acceptance the parameters of the generative model are recalculated. The generative model is extended to videos by modelling superpixel motion using a gaussian process and modifying $p(z)$ to enable the deletion and creating of superpixels.

Temporally Consistent Superpixels In [4] and [24] the problem of temporally coherent superpixels is considered as a cluster assignment problem which is solved by energy minimization. Pixels that are part of a temporally coherent superpixel are viewed as part of a cluster which is represented by a color vector and a position vector. As the position of a temporally coherent superpixel can change along the time axis of a video, distinguished position vectors exist for every frame. The color center is a three dimensional vector in CIELAB-color space. An energy E_{total} is defined to rate every possible assignment of pixels to the cluster centers where T is the set of video frames and $N(t)$ are all pixels of the frame t.

$$E_{total} = \sum_{t \in T} \sum_{n \in N(t)} E(n, \delta_n, t) \tag{3}$$

Table 1. Mean segmentation error of 9 sequences of the Berkeley Video Dataset for graph cut on the whole video volume compared to graph cut applied to the over-segmented video on three different temporally coherent superpixel approaches and a meanshift over-segmentation. While all combinations with the temporally coherent superpixels perform significantly better than graph cut applied on pixel-level there is only a slight difference between the different temporally coherent superpixel approaches.

Algorithm	Mean segmentation error
Pixel-level graph cut	14.27
Graph cut + meanshift	12.72
Graph cut + TSP [13]	9.60
Graph cut + TCS [24]	9.45
Graph cut + OVS [14]	9.54

With δ_n denoting the assigned cluster center of a pixel n, the energy term $E(n, \delta_n, t)$ is a weighted sum of the energies $E_c(n, \delta_n)$ and $E_s(n, \delta_n, t)$.

$$E(n, \delta_n, t) = (1-\alpha) E_c(n, \delta_n) + \alpha E_s(n, \delta_n, t) \tag{4}$$

The factor α controls the trade-off between color-sensitivity and spatial compactness as the energies are proportional to the Euclidean distances of the pixel n to the assigned superpixels color center and spatial center in frame t.

The minimization of (3) is done by utilizing an EM-framework. In the Expectation step the assignment of the pixels at the superpixel contours is updated by assigning the pixels to the neighboring superpixel for which the lowest energy $E(n, \delta_n, t)$ is generated. An assignment is only changed if the 4-connected topology of each superpixel is preserved. In the Maximization step the cluster centers are updated by calculating the mean color and spatial values of the assigned pixels. To allow gradual changes in illumination and to enable streaming capability the clustering is only updated inside an observation window spanning multiple frames. The observation window is shifted in overlapping intervals over the video volume. New frames entering the window are initialized by utilizing the optical backward flow. By fixing the cluster assignment on the older frames the temporally coherent superpixels will eventually stick on the image region on which they were initialized. The number of superpixels is balanced to a fixed number by splitting and terminating superpixels that are too big or too small.

Online Video Seeds The algorithm proposed by Bergh et al. [14] works on a per-frame basis like the one of [13]. It defines an energy function $H(z)$ as written in (5) where z is again the pixel assignment to the superpixels and $c_{\mathcal{A}_k^{t:0}}$ is the histogram of the set of pixels $\mathcal{A}_k^{t:0}$ belonging to the temporally coherent superpixel k in frames 0 to t.

$$H(z) = \sum_k \sum_{\{\mathcal{H}_j\}} (c_{\mathcal{A}_k^{t:0}}(j))^2 \tag{5}$$

The set of superpixels is denoted by k and $\{\mathcal{H}_j\}$ is the color subspace covered for collecting the values for the histogram bin j. An iterative hill climbing algorithm is used to maximize the energy term using a hierarchical approach. The pixels are grouped in blocks of 2×2 pixels which are merged again to form 4×4 pixel blocks. The merging is done several times and the optimization for one frame is done by first swapping blocks at the highest level between neighboring superpixels and then going down the hierarchy until pixel-level is reached.

To avoid the costly calculation of histograms to evaluate the complete energy term (5) an approximation is introduced to accelerate the optimization. A pixel or block $\mathcal{B}_n^t$ is only swapped from superpixel n to m if the following inequation is fulfilled

$$\mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_m^{t:0}}) \geq \mathbf{int}(c_{\mathcal{B}_n^t}, c_{\mathcal{A}_n^{t:0} \setminus \mathcal{B}_n^t}). \tag{6}$$

Here $\mathbf{int}(\cdot,\cdot)$ denotes the intersection between two histograms and $\mathcal{A}_n^{t:0} \setminus \mathcal{B}_n^t$ is the set of pixels of the temporally coherent superpixel without the pixel or

Table 2. Segmentation error for the individual sequences. Each sequence was initialized with user strokes on a single frame

	Sequence								
Algorithm	gokart	harley	baldwin	squirrel	kia	lion	turtle	dog	zoo
Pixel-level graph cut	5.58	15.27	9.02	16.78	10.99	18.90	5.06	13.47	33.38
Graph cut + meanshift	**4.40**	18.70	5.32	**15.90**	7.08	**18.39**	**4.56**	**11.38**	28.76
Graph cut + TSP [13]	4.51	12.79	**4.76**	16.11	**6.27**	18.64	5.23	12.32	5.76
Graph cut + TCS [24]	4.91	13.04	4.99	16.17	6.54	18.40	5.01	12.36	**3.68**
Graph cut + OVS [14]	4.55	**12.52**	5.29	16.68	7.22	18.48	4.90	12.47	3.78

block of pixels $\mathcal{B}_n^t$. The assumption only holds if the size of the superpixels is approximately the same and $\mathcal{B}_n^t \ll \mathcal{A}_n^{t:0}$. The algorithm terminates a temporally coherent superpixel if this maximizes the energy term (5). In order to hold the number of superpixels per frame constant for each terminated superpixel a new superpixel is created. After the optimization of (5) on one frame is finished the subsequent frame is initialized by copying an intermediate block level of the block-hierarchy on the new frame. The optimization then starts again from the highest hierarchy level.

3.2 Graph Cut Framework

The problem of binary segmentation is modeled using a discrete energy $E : \mathcal{L}^n \to \mathbb{R}$. As shown by Boykov et al. [5, 25], the energy can be represented by a graph and minimized using the max-flow/min-cut theorem. The energy function is represented as the sum of unary potentials φ_i, for each individual pixel, and pairwise potentials $\varphi_{i,j}$ for neighboring pixels:

$$E(y) = \sum_{i \in \mathcal{V}} \varphi_i(y_i) + \sum_{(i,j) \in \mathcal{E}} \varphi_{i,j}(y_i, y_j)\,, \tag{7}$$

where y is a labeling, $\mathcal{V}$ is the set of pixels and $\mathcal{E}$ corresponds to the set of neighboring pixels. In our work, we use the neighborhood system $\mathcal{N}_6$ where each pixel is connected to four pixels in its own frame and one adjacent pixel in the frames immediately preceding and following it. For our problem of binary video segmentation, the label set $\mathcal{L}$ consists of a foreground (*fg*) and a background (*bg*) label.

The unary potential φ_i is defined as the negative log-likelihood using a *gaussian mixture model* (GMM) [6] with five kernels:

$$\varphi_i(y_i) = -\log p(\bar{\mathbf{c}}_i | y_i = l)\,, \tag{8}$$

where here $\bar{\mathbf{c}}_i$ is the color value of pixel i in RGB color space and l is either (*fg*) or (*bg*). In our work, the likelihoods are directly learned from user labeled pixels, so-called seeds. The pairwise potential $\varphi_{i,j}$ is defined as follows:

$$\varphi_{i,j}(y_i, y_j) = \gamma \cdot \exp(-\beta |\bar{\mathbf{c}}_i - \bar{\mathbf{c}}_j|)\,, \tag{9}$$

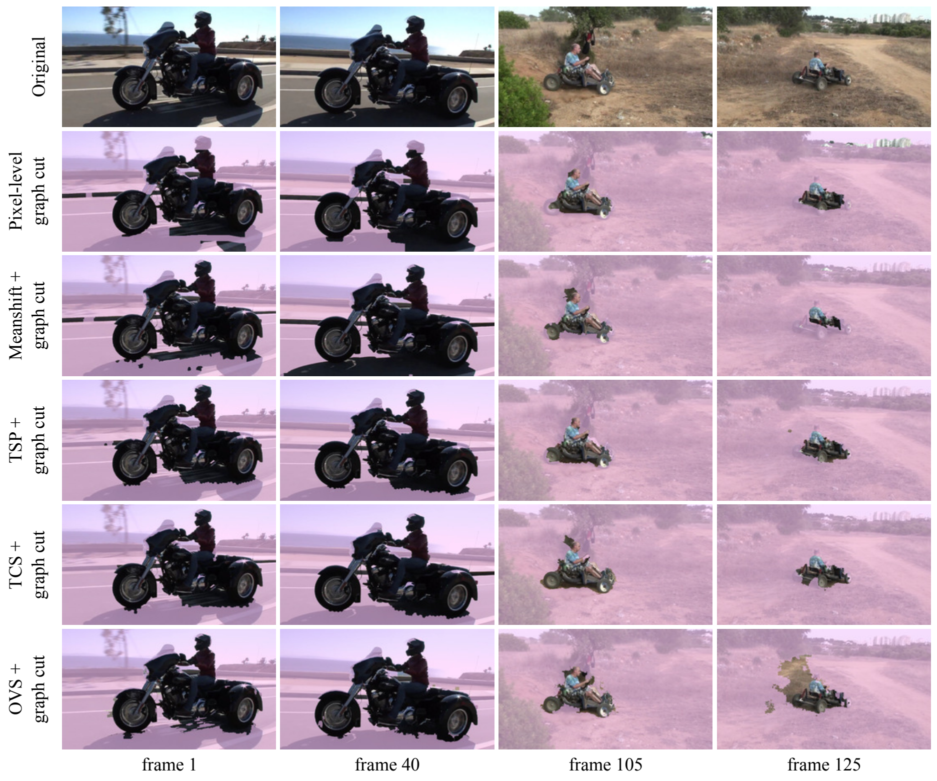

Fig. 3. Further example segmentations for the sequences *harley* and *gokart*. User strokes for the sequences can be found in Fig. 2. Note that the shadow of the motorcycle is segmented correctly by the temporally coherent superpixel approaches while the pixel-level and meanshift based approach fail.

where the parameter γ weights the influence of the pairwise term and β includes the feature variance of the image and is just as defined in [6].

The energy function E' based on temporally coherent superpixels is defined by a surjective map $m : \mathcal{V} \to \mathcal{V}'$ that maps each pixel to its corresponding superpixel. Thus, the energy function for superpixels read:

$$E'(y) = \sum_{i \in \mathcal{V}} \varphi_i(\hat{y}_{m(i)}) + \sum_{(i,j) \in \mathcal{E}} \varphi_{i,j}(\hat{y}_{m(i)}, \hat{y}_{m(j)}) \,, \tag{10}$$

where $\hat{y}$ is the labeling of the superpixels. Solving the energy minimization on the superpixels dramatically reduces the number of variables and thus the processing time.

Table 3. Average processing time to find the minimal cut of the graph.

Algorithm	Average timing
Graph cut [5]	26.45 min
Graph cut + meanshift	0.51 sec
Graph cut + TSP [13]	0.02 sec
Graph cut + TCS [24]	0.0071 sec
Graph cut + OVS [14]	0.0017 sec

4 Experimental Results

The usual way to benchmark interactive video segmentation frameworks as e.g. done in [19] is to let user segment video sequences and measure the time they need to reach a satisfying segmentation result. As the purpose of this paper is not to propose an all new interactive video segmentation framework we did not perform a user study for benchmarking. Instead we compare the segmentation results of video sequences which are produced when user strokes are placed on one frame only. This implicitly assumes that by generating a better segmentation quality with the first strokes, we can expect an equivalent segmentation quality as produced by previous approaches with less user effort in total.

To compare the performance we used nine video sequences from the Berkeley Video Dataset provided by [26]. The videos have a HD-ready resolution and a length of around 120 frames each. For the evaluation we utilized the multi-label ground-truth data of [27] which provides ground-truth segmentations produced by four different persons for every twentieth frame of the sequences. We converted the multi-label ground-truth into binary segmentations by manually selecting the labels representing the object selected for segmentation.

For comparison with previous techniques we segmented the video sequences concurrently with the same input data using a pixel-level graph like proposed in [5] using the energy function (7) and additionally on a superpixel graph built up from frame-wise generated superpixels using meanshift [28] using the energy function (10). The implementations of the temporally coherent superpixel algorithms were downloaded from the authors website. We kept the standard settings and chose 3000 superpixels per frame to be produced by TSP and TCS. Due to the implementation OVS produced only 1600 superpixels per frame.

The algorithms were initialized by user strokes which were manually created for one frame of each sequence without having insight into any resulting segmentation. The strokes were placed such as to select the main object (as seen by the user). Examples of the initializations can be seen in Fig. 1 and 2.

For a quantitative analysis we adopt the segmentation error ϵ as proposed in [29] and extend it to video by considering all pixels of frames with available ground-truth data

$$\epsilon = \frac{\text{no. missclassified pixels}}{\text{no. pixels in unclassified regions}}. \tag{11}$$

Note that in our setup user classified regions only occur on one frame per sequence. As we have four ground-truth segmentation per sequence we calculate the mean error rate for each sequence. By calculating the mean value over all sequences we show the overall performance of the temporally coherent superpixel approach in Table 1. It can be seen that the overall segmentation performance increases significantly by using the temporally coherent over-segmented video data as input for the graph cut algorithm. Among the temporally coherent superpixel approaches the algorithm proposed in [24] produced slightly better results then the other approaches. In Table 2 the mean segmentation error for each individual sequence is shown.

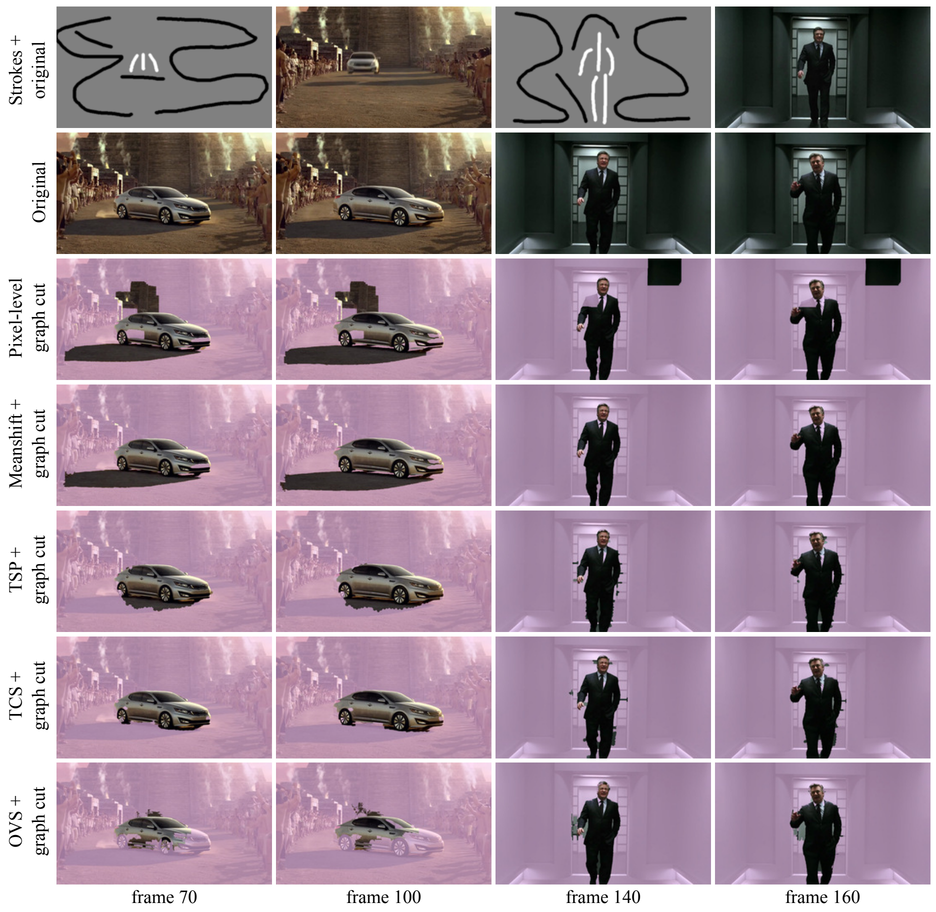

Fig. 4. Further example segmentations for the sequences *kia* and *baldwin*. First row: Strokes and original images of frame 50 and 120, respectively. Second and below: Further frames and their segmentation results. While meanshift performs visually well on the *baldwin* sequence it fails to assign the background label to the shadow of the car.

To show that the spatio-temporal over-segmentation not only increases the segmentation quality but also boosts the processing time required to find the minimal-cut in the graph we give the average processing time needed to find the optimal cut in Table 3. The benchmark was performed on an Intel Xeon E5-2690 with 132 GB of memory. Of course the processing time of the over-segmentation will need to be considered as well when being integrated into an application. But as stated by Wang et al. [16] the preprocessing can be done *overnight* as the critical phase is when the user has painted his or her stroke and waits for the new segmentation results. Additionally a fair comparison from our side would not be possible due to the diverse type of implementations (MATLAB/C/C++). The processing time of the combination of graph cut and OVS [14] is lower than the ones with TSP [13] and TCS [24]. This is due to the smaller amount of superpixels the implementation of [14] produced. The difference between TCS and TSP comes from the longer temporal segments produced by TCS resulting in a smaller graph.

5 Conclusions

We extended the pixel based graph cut approach for interactive video segmentation by using temporally coherent superpixels instead of single pixels. By grouping similar pixels to temporally coherent superpixels we overcome the problem of high memory requirements since the number of variables in the optimization framework is reduced dramatically. In experiments on high-resolution video sequences we analyze three different temporally coherent superpixel approaches and show that the spatio-temporal over-segmentation increases the segmentation quality and reduces the processing time required to optimize the energy function. Using TCS [24], the segmentation quality increases by 5.62% while the execution time of the graph cut framework is reduced from 26 minutes to 7.1 milliseconds. An even faster run-time can be achieved, while segmentation quality is a little decreased, by using OVS [14].

References

1. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: ICCV. (2011) 1323–1330
2. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. In: SIGGRAPH. (2004) 572–581
3. van den Hengel, A., Dick, A., Thormählen, T., Ward, B., Torr, P.H.: Videotrace: rapid interactive scene modelling from video. In: SIGGRAPH. Volume 26. (2007)
4. Reso, M., Jachalsky, J., Rosenhahn, B., Ostermann, J.: Temporally consistent superpixels. In: ICCV. (2013) 385–392
5. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: ICCV. Volume 1. (2001) 105–112
6. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In 3, ed.: SIGGRAPH. Volume 23. (2004) 309–314

7. Delong, A., Boykov, Y.: A scalable graph-cut algorithm for nd grids. In: CVPR. (2008) 1–8
8. Scheuermann, B., Schlosser, M., Rosenhahn, B.: Efficient pixel-grouping based on dempster's theory of evidence for image segmentation. In: ACCV. Volume 7724. (2012) 745–759
9. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV. (2003) 10–17
10. Galasso, F., Cipolla, R., Schiele, B.: Video segmentation with superpixels. In: ACCV. (2013) 760–774
11. Vazquez-Reina, A., Avidan, S., Pfister, H., Miller, E.: Multiple hypothesis video segmentation from superpixel flows. In: ECCV. (2010) 268–281
12. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR. (2010) 2141–2148
13. Chang, J., Wei, D., Fisher III, J.W.: A video representation using temporal superpixels. In: CVPR. (2013) 2051–2058
14. Bergh, M.V.D., Roig, G., Boix, X., Manen, S., Gool, L.V.: Online video seeds for temporal window objectness. In: ICCV. (2013) 377–384
15. Xu, C., Whitt, S., Corso, J.J.: Flattening supervoxel hierarchies by the uniform entropy slice. In: ICCV. (2013) 2240–2247
16. Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., Cohen, M.F.: Interactive video cutout. In: SIGGRAPH. Volume 24. (2005) 585–594
17. Price, B.L., Morse, B.S., Cohen, S.: Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: ICCV. (2009) 779–786
18. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapcut: robust video object cutout using localized classifiers. In: SIGGRAPH. Volume 28. (2009)
19. Dondera, R., Morariu, V., Wang, Y., Davis, L.: Interactive video segmentation using occlusion boundaries and temporally coherent superpixels. In: WACV. (2014) 784–91
20. Levinshtein, A., Sminchisescu, C., Dickinson, S.: Spatiotemporal closure. In: ACCV. (2011) 369–382
21. Lermé, N., Malgouyres, F., Létocart, L.: Reducing graphs in graph cut segmentation. In: ICIP. (2010) 3045–3048
22. Scheuermann, B., Rosenhahn, B.: Slimcuts: Graphcuts for high resolution images using graph reduction. In: EMMCVPR. Volume 6819. (2011) 219–232
23. 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
24. Reso, M., Jachalsky, J., Rosenhahn, B., Ostermann, J.: Superpixels for video content using a contour-based em optimization. In: ACCV. (2014)
25. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. TPAMI **26** (2004) 1124–1137
26. Sundberg, P., Brox, T., Maire, M., Arbelaez, P., Malik, J.: Occlusion boundary detection and figure/ground assignment from optical flow. In: CVPR. (2011)
27. 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
28. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. TPAMI **24** (2002) 603–619
29. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive gmmrf model. In: ECCV. (2004) 428–441