Video Segmentation with Just a Few Strokes

Naveen Shankar Nagaraja Frank R. Schmidt Thomas Brox
Computer Vision Group
University of Freiburg, Germany
{nagaraja,schmidt,brox}@cs.uni-freiburg.de

Abstract

As the use of videos is becoming more popular in computer vision, the need for annotated video datasets increases. Such datasets are required either as training data or simply as ground truth for benchmark datasets. A particular challenge in video segmentation is due to disocclusions, which hamper frame-to-frame propagation, in conjunction with non-moving objects. We show that a combination of motion from point trajectories, as known from motion segmentation, along with minimal supervision can largely help solve this problem. Moreover, we integrate a new constraint that enforces consistency of the color distribution in successive frames. We quantify user interaction effort with respect to segmentation quality on challenging ego motion videos. We compare our approach to a diverse set of algorithms in terms of user effort and in terms of performance on common video segmentation benchmarks.

1. Introduction

An annotated video carries rich information that can be used in many tasks, such as visual learning or action recognition. Especially an object segmentation is very valuable as many simpler annotations, such as bounding boxes, can be derived from a segmentation. However, already manual segmentation of image sets is very tedious. With video datasets, the large number of frames in each video makes manual segmentation in all frames intractable. For this reason, all larger benchmark datasets have an annotation only every few frames, e.g. [9].

Annotation in all frames requires support by an automated process that propagates annotations over time. Since the content usually changes little from frame to frame, this propagation looks straightforward, and many techniques have been proposed in this context, e.g. [29, 1]. Most of them are based on optical flow or temporal connections modeled by Markov chains. Apart from drift, the main challenge in label propagation is due to disocclusion that comes with viewpoint changes. An object undergoing viewpoint changes in a video is highly challenging for annotation. New parts appear that were not visible in the annotated frame (cf. Figure 2). Consequently, there is nothing to propagate and a decision must be made which label to assign to this area. Existing approaches use color to assign labels in disocclusion areas, but this fails as new colors appear.

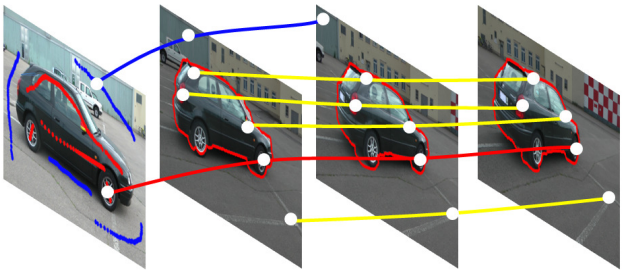

Figure 1. We propose to combine the motion cues of point trajectories with appearance and volume information. As a result, only little user annotation is needed to segment a whole video.

In this paper, we emphasize the importance of motion cues to deal with disocclusions. In contrast to color, the motion of an object is locally consistent. Motion becomes even more reliable if it is integrated over time. This has been demonstrated by Brox and Malik [4] in the scope of motion segmentation.

However, motion segmentation fails if there is no independent object motion. An important practical case that cannot be handled by motion segmentation, but by the framework presented in this paper, is that of a mostly static object but a moving camera. In such a case, there are significant motion differences between the object and its background *except* for the base point, where the object is touching the ground. At the base point, the motion on the object is identical to that of the ground, and a motion segmentation approach will leak object labels to the background. Often, appearance cues are unreliable in this region, too, due to shadows. Sometimes user input is the only reliable source of information to disambiguate at the base point. Since we combine many automated concepts into our framework, i.e., temporal propagation, long term motion, color distributions

Figure 2. Changing viewpoints lead to disocclusion areas (marked) with content that was not visible in the original frame. It must be inferred if these regions belong to object or background.

and volume consistencies, we require only very little user input to avoid segmentation errors.

We introduce several technical innovations to integrate all these cues in a principled way by optimizing a single cost function. This includes a new appearance consistency cost as well as a volume consistency cost, which can be optimized with the trust region framework of Gorelick et al. [11].

We quantify the amount of time the user has to spend with our tool to achieve a certain segmentation quality and compare this too alternative annotation strategies. For completeness, we also report results on standard benchmark datasets, where we also achieve state-of-the-art performance.

2. Related work

There are many related approaches to video segmentation each with its own set of advantages and disadvantages. *Interactive video segmentation*, e.g. [21, 16], relies on frequent user intervention to prevent errors from being propagated. It is very accurate and appreciated in video editing, yet the frequent user input prevents its application to large datasets.

Owing to the tediousness in interactive video segmentation, *bounding box tracking* offers a scalable solution. Boosting based classifiers [24] and random forest classifiers [23] are popular choices for learning and updating the template model. Godec et al. [10] used the bounding boxes to initialize a Grabcut segmentation for each video frame. Chockalingam et al. [7] proposed to use a classifier based data term for a level-set driven segmentation.

A common problem with tracking is the continuous drift and error propagation. Hence, it is quite common to consider video segmentation as a *spatio-temporal MRF* optimization problem [1, 26, 5]. In order to reduce the running time, some methods create superpixels per frame and connect them temporally in order to generate temporally consistent object regions [22, 3, 27, 13]. This leads to temporally consistent superpixels that must be connected spatially by other cues. In Jain and Grauman [14] this is a manual segmentation provided for the first frame.

There are also some works that try to segment objects in videos in an unsupervised manner not using any user input [18, 19]. These works rely on the independent motion of the object. The approach breaks in case of static objects due to the aforementioned base point problem.

Contributions. The main contribution of this work is the smart combination of many of the good ideas presented in previous works to provide a convenient tool for annotation of large video datasets without restrictions on the video content. The final segmentation is the result of an interaction between the user input, motion cues, spatio-temporal consistency, and per-frame driven appearance cues.

Moreover, we present a new color constraint that encourages consistency of the color distribution between frames rather than just assigning pixels to the most likely distribution model. This is technically related to the proportion priors proposed by Nieuwenhuis et al. [17], where the size of regions is constrained between successive frames. In contrast, we apply constraints to the color distributions.

3. Constrained motion segmentation

In the following, an image $I\colon \Omega \to \mathbb{R}^3$ is a mapping from the image domain $\Omega \subset \mathbb{R}^2$ into the color space $\mathbb{R}^3$. We consider a video $\mathcal{I}\colon \Omega \times [1,T] \to \mathbb{R}^3$ as a temporal sequence of images $I_t := \mathcal{I}(\cdot, t)$ to which we refer to as *frames*. A video is more than just a collection of images, since it includes motion, which becomes available by computing the optical flow between successive frames I_{t-1} and I_t.

3.1. Optical flow and point trajectories

In this paper, the optical flow is used in two ways: (1) It allows us to propagate user annotation to neighboring frames, as used in most tracking and label propagation works. (2) Similarities and dissimilarities in the motion indicate points belonging to the same or different objects, respectively; this is the way motion is typically used in motion segmentation approaches.

By concatenating the optical flow vectors to point trajectories [25], both ways to use optical flow for segmentation can be combined in an elegant way. If the video is regarded as a single graph, a point trajectory connects nodes (pixels) in this graph over time via must-link constraints. This way, it propagates information available at one node on the trajectory to all other nodes along the trajectory. This propagation of information over time gives the unsupervised motion segmentation approach [18] a clear boost over motion segmentation based on just two-frame optical flow. We combine this idea of point trajectories with user scribbles (cf. Figure 3). Technically this comes down to replacing the hard decision of spectral clustering in [18] into soft data terms derived from a random walker computation [12].

We compute point trajectories based on [25] and define affinities between trajectories in the same way as described in [18]. A trajectory is encoded as a triple (t_1, t_2, c),

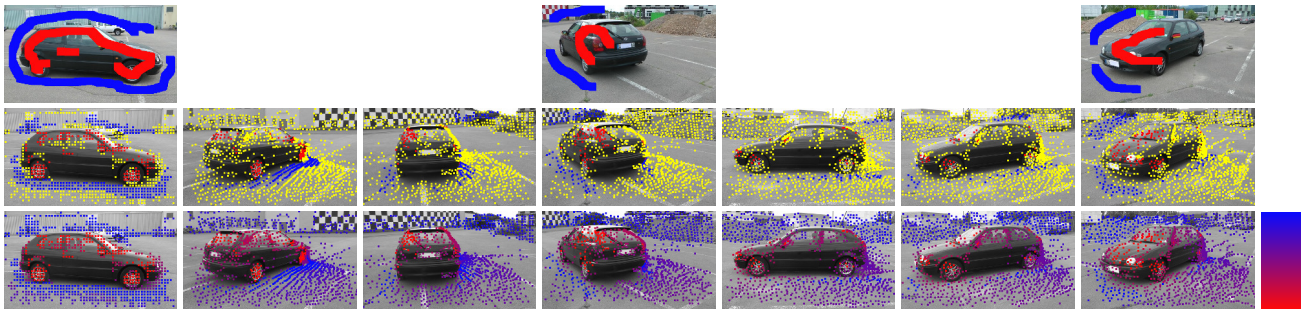

Figure 3. If we sparsely annotate a video over time and space (top row), the pre-computed trajectories propagate this information to other frames (middle row). Unlabeled trajectories (yellow) get a likelihood of object or background from a random walk computation [12] (bottom row). The likelihood is shown by a color that is a linear interpolation between *red* and *blue* as shown in the color bar. The strokes in just these three frames were the only user input for this video.

where $[t_1, t_2] \subset [1, T]$ describes the time where the path $c : [t_1, t_2] \to \Omega$ of the trajectory is visible. We denote the set of all trajectories by $\mathcal{C}$. The approach of [25] allows us to choose the spatial sampling density of the trajectories. We sample a trajectory every 8×8 pixels.

3.2. User input

The user can put disjoint scribbles $P_0, P_1 \subset \Omega \times [1, T]$ somewhere in the video to enforce the marked areas to be assigned to background or object, respectively. The sparse user input serves to resolve motion ambiguities at the base point of objects and can correct for errors or limited accuracy of the optical flow. Every trajectory that passes through a point marked by the user gets assigned a hard constraint. In the rare case of trajectories that coincide with both, P_0 and P_1 (due to tracking errors or mistakes by the user), we remove the respective trajectory to avoid contradicting constraints. Contradicting constraints can be created also by overlapping trajectories due to missed occlusions, erroneous optical flow, or significant scaling. We stop the two overlapping trajectories at the frame of overlap and restart a new trajectory.

3.3. Object-background bias

After seeding some of the trajectories to the object resp. background, these seeds are propagated with the random walker to the remaining trajectories using the motion based affinities [25]. The random walker only yields a soft labeling $u_{\text{Motion}} : \mathcal{C} \to [0, 1]$, which is usually rounded.

In this paper, we do not need rounding. The soft solution rather serves as a spatio-temporal skeleton for the dense segmentation described in the next section. The distance of a label from binary values is used as a fidelity measure of the label. To this end, we transform u_{Motion} into the weighted segmentation bias $D_{\text{Motion}} : \mathcal{C} \to [-\infty, \infty]$:

$$D_{\text{Motion}}(q) := \frac{1}{u_{\text{Motion}}(q)} - \frac{1}{1 - u_{\text{Motion}}(q)} \qquad (1)$$

For trajectories q that have passed through a user scribble, the random walk ensures $u_{\text{Motion}} = 0$ or $u_{\text{Motion}} = 1$, and D_{Motion} provides a hard constraint for background ($D_{\text{Motion}} = \infty$) or object ($D_{\text{Motion}} = -\infty$), respectively. If the random walker's result is undecided, i.e. $u_{\text{Motion}} = 0.5$, the bias $D_{\text{Motion}} = 0$. Even little user input (strokes in just 3 frames; Figure 3, 1st row) yields a strong bias thanks to the motion based affinities (Figure 3, 3rd row).

4. User guided dense video segmentation

If we want to track an object in a video $\mathcal{I}$, it is not enough to segment each frame I_t of the video independently. Instead, we assume no rapid changes from one frame to the other. Incorporating this temporal consistency is important in video segmentation. In Section 3, we derived a temporally consistent bias D_{Motion} from point trajectories. This bias only takes motion and user scribbles into account, it is sparse, and many points are still undecided. In the present section, we are interested in a dense segmentation that incorporates this bias, but also enforces temporal consistency of appearance and shape. To this end we minimize

$$E(\mathcal{S}) := \sum_{t=1}^{T} E^t_{\text{Motion}}(S_t) + E^t_{\text{Appear}}(S_t) + E^t_{\text{Reg}}(S_t), \quad (2)$$

where $\mathcal{S} : \Omega \times [1, T] \to \{0, 1\}$ denotes the whole video segmentation and $S_t := \mathcal{S}(\cdot, t)$ the segmentation of frame I_t. E^t_{Motion} incorporates the motion bias of Section 3, and E^t_{Appear} enforces temporal consistency of the appearance models (cf. Section 4.2). The energy E_{Reg} combines a volume and a length driven regularization (cf. Section 4.3).

4.1. Motion and user input

The energy E^t_{Motion} is based on the hard constraints P_0, P_1 provided by the user (cf. Section 3.2) and the bias D_{Motion} derived from the optical flow (cf. Section 3.3). Us-


I_{t-1}


$A^{t-1,1}_{t-1,i}$


I_t


$A^{t-1,1}_{t,i}$

Figure 4. Computing a Gaussian mixture model (GMM) of frame I_{t-1}, leads to partitions $A^{t-1,\ell}_{t-1,i}, \ell \in \{0,1\}$ that represent the densities of the foreground and background (only foreground is shown here). Applying the same GMM to the following frame I_t leads to a different partition $A^{t,\ell}_{t,i}$. We enforce similarity between the densities modeled by $A^{t,\ell}_{\tau,i}$.

ing the notation $\langle f, g\rangle := \int_\Omega f(x)\cdot g(x)\mathrm{d}x$, we set

$$\tilde{E}^t_{\text{Motion}}(S_t) := \alpha_{\text{Motion}} \cdot \left\langle f^t_{\text{Motion}}, S_t \right\rangle \tag{3}$$

with a weighting parameter $\alpha_{\text{Motion}} \in \mathbb{R}^+$ and

$$f^t_{\text{Motion}}(x) := \begin{cases} +\infty & \text{if } x \in P_0 \\ -\infty & \text{if } x \in P_1 \\ D_{\text{Motion}}(q) & \text{if } \exists q = (t_1, t_2, c) \in \mathcal{C} : \\ & t \in [t_1, t_2] \text{ and } x = c(t) \\ 0 & \text{otherwise.} \end{cases}$$

One can easily verify that f^t_{Motion} is well defined and does not have contradictive constraints. In contrast to Section 3, the user scribbles are now also exploited for positions $x \in \Omega$ without a trajectory (cf. 1st and 2nd row of Figure 3).

In order to exploit the optical flow also for locations that are not captured by trajectories we are interested in the pixelwise consistency measure $\phi^t_{\mathcal{S}} : \Omega \to \mathbb{R}$

$$\begin{aligned}\phi^t_{\mathcal{S}}(x) := &|S_t(x) - S_{t+1}(x + w_t)| + \\ &|S_t(x) - S_{t-1}(x + \hat{w}_t)|\end{aligned}$$

where w_t and $\hat{w}_t$ refer to the forward and backward flow of frame I_t, respectively. For $\phi^1_{\mathcal{S}}$ and $\phi^T_{\mathcal{S}}$ we only consider the forward or the backward flow respectively. Together with $\tilde{E}^t_{\text{Motion}}$ we obtain

$$E^t_{\text{Motion}}(S_t) := \tilde{E}^t_{\text{Motion}}(S_t) + \alpha_{\text{Flow}} \cdot \left\langle \phi^t_{\mathcal{S}}, c^t_w \right\rangle$$

where the binary function $c^t_w(x)$ indicates whether the flow is reliable according to the forward-backward consistency check and $\alpha_{\text{Flow}} \in \mathbb{R}^+$ is a weighting factor.

4.2. Color distribution consistency

While long term motion is a valuable cue, it is unreliable in homogeneous areas of the image and near object boundaries. Color, although often ambiguous, is much more precise, which is why it is a central cue in most video segmentation approaches. Most common is the definition of a unary appearance cost that assigns a pixel to the most likely region according to a color distribution model for each region. The distribution model can be, for example, modeled by a Gaussian mixture model (GMM), and it is estimated based on user input or a segmentation in the previous frame.

This approach is suboptimal in video sequences. Consider an example where a blueish pixel should be assigned a label in the new frame. Assume there are many more blue pixels in the background (due to sky or water) than in the object region. If the blueish pixel actually belongs to the object, it would strongly tend to be assigned to the background if we used a GMM approach.
In fact, we rather want to propagate the share of blue pixels in the two regions to the new frame, or, more generally, pixels in the new frame should be assigned in such a way that the color distributions of the previous frame are preserved. In contrast to the unary cost in a GMM approach, the assignment of a pixel has a global effect, since the assignment changes the preferences of all other pixels: as more and more blue pixels are assigned to the foreground, it becomes less likely that other blue pixels should be assigned there as well. Interestingly, this problem can be integrated into our global energy formulation.

We model the distributions of both regions by two independent Gaussian mixture models with k components each. The $2k$ mixture components $(\mu^{t,\ell}_i, \Sigma^{t,\ell}_i)$ are described by the mean $\mu^{t,\ell}_i$ and the inverted covariance matrix $\Sigma^{t,\ell}_i$, where $1 \le i \le k$ denotes the model and $\ell \in \{0,1\}$ indicates whether the model belongs to the object ($\ell = 1$) or the background ($\ell = 0$). Using the Mahalanobis distance

$$d^{t,\ell}_i(y) := \left\langle y - \mu^{t,\ell}_i, \Sigma^{t,\ell}_i \cdot \left(y - \mu^{t,\ell}_i\right) \right\rangle,$$

we can divide Ω into $2k$ regions depending on the color information and the color model of frames I_{t-1} and I_t. This leads to sets

$$A^{t,\ell}_{\tau,i} := \left\{ x \in \Omega \,\middle|\, d^{\tau,\ell}_i(I_t(x)) = \min_{\substack{m\in\{0,1\} \\ j\in[1,k]}} d^{\tau,m}_j(I_t(x)) \right\}$$

that are based on the Mahalanobis-distance driven nearest neighbor approach (cf. Figure 4). Note that $A^{t,\ell}_{t,i}$ and $A^{t,\ell}_{t-1,i}$ are both using the *color information* of frame I_t, but $A^{t,\ell}_{t-1,i}$ uses the *color model* of the previous frame. This is necessary to compare the same color models for consecutive frames. Let $f^{t,\ell}_{\tau,i} : \Omega \to \{0,1\}$ be the indicator function of $A^{t,\ell}_{\tau,i}$ ($f^{t,\ell}_{\tau,i}(x) = 1 \Leftrightarrow x \in A^{t,\ell}_{\tau,i}$). Then, the distributions that we obtain for S_t are given by

$$p^{t,\ell}_i := \frac{\left\langle f^{t,\ell}_{t,i}, S_t \right\rangle}{\langle 1, S_t \rangle} \qquad q^{t,\ell}_i := \frac{\left\langle f^{t,\ell}_{t-1,i}, S_t \right\rangle}{\langle 1, S_t \rangle}.$$

In order to use p^{t-1} as prior for q^t, we define E^t_{Appear} as the KL-divergence of these two distributions

$$E^t_{\text{Appear}}(S_t) := \alpha_{\text{Appear}} \cdot \sum_{\ell=0}^{1} \sum_{i=1}^{k} p^{t-1,\ell}_i \log \left(\frac{p^{t-1,\ell}_i}{q^{t,\ell}_i} \right) \quad (4)$$

with a weighting parameter $\alpha_{\text{Appear}} \in \mathbb{R}^+$.

4.3. Volume and Length Regularization

We assume that the observed object is present in each frame. As a result, we want to exclude the trivial solutions $\emptyset, \Omega$ for S_t. Moreover, we want to favor segmentations S_t that cover a similar area as S_{t-1}. To incorporate this prior into our framework, we consider the following energy

$$V(S) := \frac{\beta}{\text{vol}(S)} + \frac{1-\beta}{\text{vol}(1-S)} \qquad \beta \in [0,1].$$

One can easily verify that if we choose β as

$$\beta_\theta := \frac{\theta^2}{\theta^2 + (1-\theta)^2},$$

V is globally minimized by segmentations that cover $\theta \in [0,1]$ of the image domain Ω. This leads to the volume- and length-based regularization term

$$E^t_{\text{Reg}}(S_t) := \alpha_{\text{Reg}} \left(\frac{\beta_{\theta_t}}{\text{vol}(S_t)} + \frac{1-\beta_{\theta_t}}{\text{vol}(1-S_t)} \right) + \text{len}_{g^t}(S_t)$$

with a weighting parameter $\alpha_{\text{Reg}} \in \mathbb{R}^+$. $\text{vol}(S_t) := \langle 1, S_t \rangle$ is the size of the segment S_t and $\text{len}_{g^t}(S_t) := \langle g^t, |\nabla S_t| \rangle$ is the weighted total variation corresponding to the geodesic contour length. The weighting function g^t is a positive, decreasing function of the image gradient. Combining the length term with the volume term excludes noisy solutions as well as trivial solutions. For $t > 1$ we choose $\theta_t := \frac{\text{vol}(S_{t-1})}{\text{vol}(\Omega)}$ to favor segmentations S_t that cover a similar area as S_{t-1}. For $t = 1$ we set $\theta_1 := 0.5$.

4.4. Energy minimization

The overall energy (2) reads in detail

$$E(\mathcal{S}) = \sum_{t=1}^{T} \alpha_{\text{Motion}} \left\langle f_{\text{Motion}}, S_t \right\rangle + \alpha_{\text{Flow}} \left\langle \phi^t_{\mathcal{S}}, c^t_w \right\rangle + \alpha_{\text{Appear}} \sum_{\substack{\ell \in \{0,1\} \\ i \in [1,k]}} p^{t-1,\ell}_i \log \left(\frac{p^{t-1,\ell}_i \langle 1, S_t \rangle}{\left\langle f^{t,l}_i, S_t \right\rangle} \right) + \alpha_{\text{Reg}} \left(\frac{\beta_{\theta_t}}{\langle 1, S_t \rangle} + \frac{1-\beta_{\theta_t}}{\langle 1, 1-S_t \rangle} \right) + \text{len}_{g^t}(S_t).$$

E is not convex due to the appearance and the volume terms, but it fits into the regional energies that can be optimized via the fast trust region framework [11], which provides a local minimum. In contrast to Gorelick et al. [11], we use the continuous primal-dual scheme of Pock and Chambolle [6] instead of the graph cut framework, because it can be run efficiently on a GPU.

In a first step, we compute for each $t \in [1,T]$ a segmentation S_t that only uses the energy terms involving previously computed S_τ with $\tau < t$. This provides us with a reasonable initialization for $\mathcal{S}$. Afterwards, we compute for each $t \in [1,T]$ a local minimum of the energy $E(\mathcal{S})$ by fixing all the other segmentations S_τ with $\tau \neq t$. This optimization is done until convergence, which provides us with a (local) minimum of E.

5. Experiments

5.1. Running times

For pre-processing, our method needs up to 12 seconds per frame for the optical flow on a GPU, and up to 0.5 seconds per frame for the random walker computation on a CPU. The minimization of (2) on a GPU takes between 1 to 3 seconds per frame (1.3 to 2.0 Megapixels).

5.2. User interaction

As this paper aims on providing a method for efficient annotation of large video datasets with reasonable quality, we first evaluated the tradeoff between the time it takes the user to interact with the software and the quality of the final segmentation. To this end we collected a dataset of 24 videos each showing one object out of 4 different categories - car, chair, cat, and dog - with changing viewpoint (cf. Figure 8). There was no object motion in the car and chair videos, whereas some cats and dogs show strong articulated motion. We manually provided accurate ground truth segmentation for 137 out of the 11882 frames of this dataset. The quality was measured with the Pascal Overlap Measure (POM),*i.e.*, intersection over union of ground truth GT and

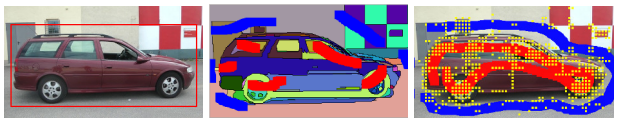

Figure 5. User input using the GUI tool. **left-** a bounding box for [10], **center-** superpixels at different hierarchies are displayed for [13], and **right-** tracks are displayed for our method. Note that the user scribbles are displayed in *red* and *blue*.

computed segmentation S:

$$\mathrm{POM}(\mathrm{GT}, S) := \frac{1}{T}\sum_{i=1}^{T}\frac{|\mathrm{GT}_i \cap S_i|}{|\mathrm{GT}_i \cup S_i|}. \qquad (5)$$

We compare our approach to three other obvious approaches that can be assembled in a straightforward manner from state-of-the-art software that is publicly available. Papazoglou and Ferrari [19] provide a fast software to segment large video datasets without the need of any user interaction. Godec et al. [10] provide a contour tracker that gets initialized by a bounding box set by the user in the first frame (cf. Figure 5, left hand side) and a GrabCut segmentation based on this bounding box. We allowed for putting bounding boxes every few frames to reinitialize the tracker. Finally, Grundmann et al. [13] provide a software to create a hierarchy of spatio-temporally consistent supervoxels based on color and optical flow. We integrated these precomputed supervoxels into a user interface that allows the user to assign supervoxels to the foreground by drawing scribbles in frames of their choice. The user immediately sees the resulting segmentation and can put more scribbles to improve the result (cf. Figure 5, center). The user can decide on a suitable hierarchy level of the supervoxels at any time.

Also the proposed method was integrated into a user interface where the user sees the video with the trajectories and can draw scribbles in frames[1] of his/her choice. When satisfied, the user presses a button to run the random walker (cf. Figure 5, right hand side). If the user is not yet satisfied with the outcome, he can put more scribbles and run the random walker again. The user can also see a per-frame instant preview of the expected dense segmentation during interaction.

We collected interaction data from 5 different users. Some of them took more time than others until they were satisfied with the result. We measured the accumulated time of each user and also took intermediate results (iterations) into account. In the approach by Godec et al., each new bounding box is a new iteration. In the tool based on Grundmann et al., the user could press a button to generate a new iteration, and in our method, an intermediate result was created each time the random walker was run.

[1]The software and data is available at http://lmb.informatik.uni-freiburg.de/resources/binaries/iVideoSeg/

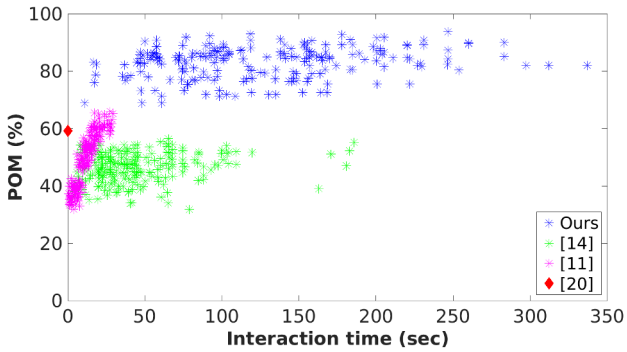

Figure 6. Interaction time vs. segmentation quality. The graph shows the amount of supervision needed in terms of user time against the achieved segmentation quality obtained with this input. With the proposed approach, one gets a reasonably good segmentation after approximately one minute. The contour tracking of Godec et al. [10] is much faster but does not reach the same quality even when putting bounding boxes frequently.

Both users achieved similar accuracies with the different annotation methods, but with the proposed tool the unexperienced user took longer. This suggests that for annotating a larger dataset, users should be trained to make them more efficient.

Figure 6 shows a dot plot, where each dot corresponds to one of the iterations of one user working with one method on one of the 24 videos. We also report the performance of the unsupervised algorithm [19], where the user time is zero. Clearly the proposed approach enables higher quality segmentations than the compared approaches with reasonable effort, which is also confirmed by the results in Figure 8. If user time is most valuable and quality is secondary, the unsupervised method [19] is an efficient alternative option, but it has problems especially on static objects due to the base point problem.

The curve for our method saturates below 100%, i.e., segmentation is not perfect even if the user spends more time. This is also due to the present user interface that only shows the random walk output but not the full segmentation. However, such high accuracy segmentations, where users spend up to multiple hours on one video, is already covered

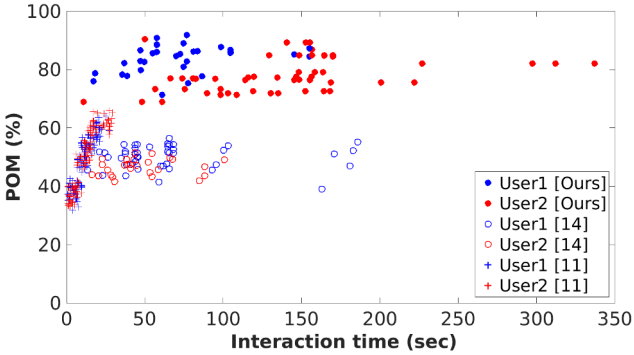

Figure 7. Interaction vs. segmentation quality for two different users for all the compared methods.

Ours | Godec et al. [10] | Papazoglou et al. [19] | Grundmann et al. [13]

Figure 8. Qualitative comparison on car,cat,chair and dog videos for the output of the first iteration of user interaction for our method, [10], [19], and [13]. The foreground is highlighted with an *yellow* contour. An image with no foreground contour has an empty segmentation.

by commercial software, for instance, Adobe Premiere [2]. For dataset annotation, especially when it is training data, pixel accurate results are not necessary. In this regime, the proposed approach is quite efficient.

Figure 7 shows a comparison between a more experienced user (one of the authors) and an untrained user, who was only given rough instructions.

5.3. Comparison on standard benchmarks

In order to link our results to video segmentation literature, we also ran our method on some common benchmark datasets, namely *SegTrack1*, *SegTrack2* [15] and *Youtube Objects* [20]. These datasets are not meant for evaluating interactive segmentation, thus, they do not provide scribbles but a ground truth segmentation in the first frame, which we take as user input P_0 and P_1 to our method and also to the other compared methods that rely on user input.

SegTrack1 dataset. SegTrack1 is a small dataset of 6 very low resolution videos, having rapid and articulated motion, motion blur and color similarity between object and background. Although this dataset is very small and a little outdated, most results are reported on this dataset. It is common practice to use a different set of parameters for each sequence. The only parameters that we optimized for the sequences are α_{Flow} and α_{Motion}. The remaining parameters are the same for all sequences. For slow sequences we have a higher emphasis on the *temporal smoothness constraint* and for fast sequences we rely more on the optical flow driven trajectories.

Method	[28]	[8]	[26]	[7]	[14]	Ours
Average	2169.5	745.8	866.8	1873	874.3	**694.4**

Table 1. Average number of mislabeled pixels in SegTrack1 for our method and other methods as reported by [14].

Instead of POM, Segtrack1 evaluates the segmentation by the average number of mislabeled pixels. Supervised methods typically use the ground truth in the first frame as user input. We did the same. The numbers reported in Table 1 are taken from the recent work by Jain and Grauman [14] and include also the state-of-the-art on this dataset from many previous papers. It shows that on average we outperform all methods reported in Jain and Grauman [14].

SegTrack2 dataset. SegTrack2 [15] is a more recent extension of SegTrack1 with 14 low resolution videos (0.08-0.23 Megapixels per frame). Apart from Godec et al. [10] and Papazoglou and Ferrari [19], we compared to Ochs et

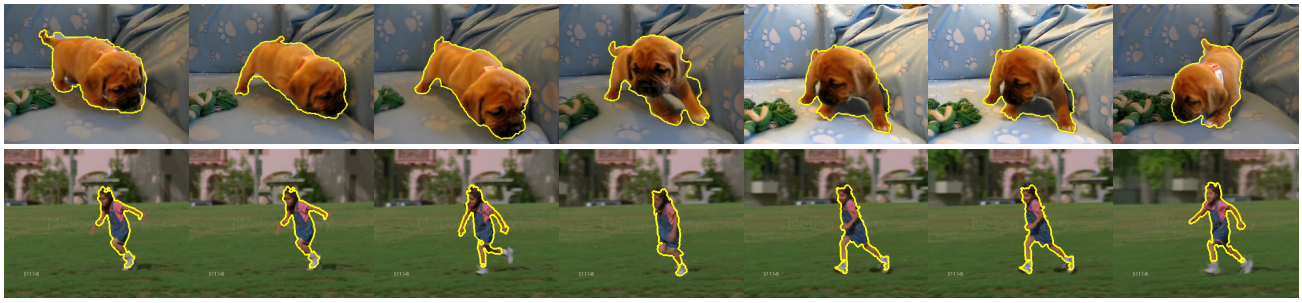

Figure 9. Sample results of the proposed method on Youtube Objects [20] (1st row) and Segtrack2 [15] (2nd row).

Method	[10]	[19]	[18]	OF	Ours
Average	41.3	53.5	8.0	40.1	**69.6**

Table 2. Average POM in percentage over all the sequences for the Segtrack2 dataset.

al. [18] and a baseline termed OF that propagates the given annotations using the optical flow [4]. The methods were chosen based on their ability to produce object-level segmentations, the availability of code, and their running time in order to run them also on the much larger *Youtube Objects* dataset. The proposed method on average outperformed the other methods (cf. Table 2) and obtained the best performance for 9 out of 14 sequences. Moreover, to assess the contribution of different terms in Equation (2), we evaluated our method by skipping each term in it (cf. Table 3).

Many sequences have more than one annotated object. In such a case we ran our method for each object separately by considering the rest as background and then averaged the result over all objects for that sequence. Since [19] automatically returns all moving objects as one foreground, we computed POM by taking a union of all annotated objects as ground truth. Note that this protocol is slightly biased towards [19], since it does not have to differentiate between the different annotated objects.

Youtube objects dataset. Youtube Objects [20] is a large dataset of 1407 videos (extracted from 155 web videos) of different resolutions (0.17-2.07 Megapixels per frame). The dataset provides a diversity in object classes, motions and appearances. Jain et al. [14] provide ground truth annotation for 126 of the 1407 videos. These annotations are provided on every 10th frame of downsampled videos. Therefore, we ran our method on the downsampled videos and the POM was computed with respect to the subset of annotated frames.

We used the same set of parameters for all sequences and outperform the other methods in all object classes, while reaching an average performance of 74.1% (cf. Table 4). In Figure 9 we show a few qualitative results.

6. Summary

We proposed a tool for video object segmentation that efficiently combines sparse user input with long term motion cues and color consistency constraints. We analyzed the impact of the amount of user interaction on segmentation quality. Our method fills a gap between professional tools that allow for pixel accurate segmentation spending up to multiple hours for one video and current tools that are based on bounding box annotation or supervoxels. Although the focus of the work was on setting up a convenient tool, the underlying technique also performs well on standard benchmark datasets. In particular, the approach deals well with large viewpoint changes, where other methods typically have problems. The data and software can be downloaded from our website.

Terms	M/A/V/-	M/A/-/L	M/-/V/L	-/A/V/L	All
POM	58.5	68.9	43.8	52.5	**69.6**

Table 3. POMs on SegTrack2 by skipping each term in Equation (2). Term legend - Motion, Appearance, Volume, Length. Clearly Motion, Appearance and Length are vital for segmentation. Volume is crucial in sequences where the motion cue is negligible. Combination of all cues, thus, yields the best result.

Category	[10]	[19]	[14]	[18]	OF	Ours
aeroplane	73.6	70.9	86.3	13.7	61.3	**89.0**
bird	56.1	70.6	81.0	12.2	76.8	**81.6**
boat	57.8	42.5	68.6	10.8	73.6	**74.2**
car	33.9	65.2	69.4	23.7	56.4	**70.9**
cat	30.5	52.1	58.9	18.6	58.4	**67.7**
cow	41.8	44.5	68.6	16.3	60.5	**79.1**
dog	36.8	65.3	61.8	18.0	47.5	**70.3**
horse	44.3	53.5	54.0	11.5	43.3	**67.8**
motorbike	48.9	44.2	60.9	10.6	51.6	**61.5**
train	39.2	29.6	66.3	19.6	74.5	**78.2**
Average	46.2	54.8	66.6	15.5	60.3	**74.1**

Table 4. POM in percentage for the Youtube-Objects dataset.

Acknowledgements

We acknowledge partial funding by the ERC Starting Grant - VIDEOLEARN. We thank Markus Naether for developing a GUI for our tool. This study was also supported by the Excellence Initiative of the German Federal and State Governments: BIOSS Centre for Biological Signalling Studies (EXC 294).

References

[1] V. Badrinarayanan, F. Galasso, and R. Cipolla. Label propagation in video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: Robust video object cutout using localized classifiers. *ACM Trans. Graph.*, 28(3):70:1–70:11, July 2009. 7

[3] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 2

[4] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011. 1, 8

[5] I. Budvytis, V. Badrinarayanan, and R. Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[6] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, may 2011. 5

[7] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 2, 7

[8] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *British Machine Vision Conference (BMVC)*, 2011. 7

[9] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B.Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1

[10] M. Godec, P. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 2013. 2, 6, 7, 8

[11] L. Gorelick, F. R. Schmidt, and Y. Boykov. Fast trust region for segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2, 5

[12] L. Grady. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28:1768–1783, 2006. 2, 3

[13] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2, 6, 7

[14] S. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision (ECCV)*, 2014. 2, 7, 8

[15] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 7, 8

[16] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, 2005. 2

[17] C. Nieuwenhuis, E. Strekalovskiy, and D. Cremers. Proportion priors for image sequence segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2

[18] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187 – 1200, Jun 2014. Preprint. 2, 8

[19] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2, 6, 7, 8

[20] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7, 8

[21] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 2

[22] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[23] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *ICCV'09 Workshop on On-line Learning for Computer Vision*, 2009. 2

[24] S. Stalder, H. Grabner, and L. V. Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *ICCV'09 Workshop on On-line Learning for Computer Vision*, 2009. 2

[25] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *European Conference on Computer Vision (ECCV)*, 2010. 2, 3

[26] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *British Machine Vision Conference (BMVC)*, 2010. 2, 7

[27] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *European Conference on Computer Vision (ECCV)*, 2010. 2

[28] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *European Conference on Computer Vision (ECCV)*, 2012. 7

[29] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 1