

Object Detection, Tracking, and Motion Segmentation for Object-level Video Segmentation

Benjamin Drayer and Thomas Brox

Department of Computer Science,
University of Freiburg, Germany
{drayer, brox}@cs.uni-freiburg.de

Abstract. We present an approach for object segmentation in videos that combines frame-level object detection with concepts from object tracking and motion segmentation. The approach extracts temporally consistent object tubes based on an off-the-shelf detector. Besides the class label for each tube, this provides a location prior that is independent of motion. For the final video segmentation, we combine this information with motion cues. The method overcomes the typical problems of weakly supervised/unsupervised video segmentation, such as scenes with no motion, dominant camera motion, and objects that move as a unit. In contrast to most tracking methods, it provides an accurate, temporally consistent segmentation of each object. We report results on four video segmentation datasets: YouTube Objects, SegTrackv2, egoMotion, and FBMS.

Keywords: Video Segmentation, Motion Segmentation, Object Tracking

1 Introduction

Video object segmentation plays a role in many high level computer vision tasks, such as action and event recognition. In contrast to single images, videos provide motion as a very strong bottom-up cue that can be exploited to support the high level tasks.

For this reason, video segmentation is often approached with unsupervised, purely bottom-up methods [11, 13, 14, 17, 18, 25, 27–29]. Especially motion segmentation can work quite well in a bottom-up fashion, if the objects of interest show some independent motion in the video. However, this is not the case in all videos. Very often, objects of interest are mostly static and almost all motion is due to camera motion. In such cases, motion segmentation fails. Also in cases where objects are moving jointly, such as a horse and its rider, a separation of the objects is often not possible with just bottom-up cues.

These limitations are avoided by adding user input that decides in these cases [3, 9, 16, 24]. However, this is not an option for a system that is supposed to automatically interpret video material.



Fig. 1. This example from the YouTube Objects dataset highlights common challenges in video segmentation: strong camera motion, multiple object instances, and appearance of a new object. Unsupervised methods as [18] (**top row**) and [11] (**middle row**), fail to recognize the rider as an individual object as well as the static person on the left. Our weakly supervised method (**bottom row**) deals well with these issues and correctly identifies these objects. Besides the segmentation, we also retrieve the class label of each object.

In this paper, we propose a weakly supervised method. The weak supervision is due to the use of an off-the-shelf detector which was trained in a supervised manner on annotated images. However, running the video segmentation on new videos does not require any user input anymore. Our technical contribution is an effective way to combine the concept of tracking-by-detection with concepts from motion segmentation and local appearance cues from the object detector. Typically failure cases in video segmentation, such as constant motion, jointly moving objects, as well as objects that move into the field of view, are well handled. We provide ablation studies for the proposed tube extraction and the segmentation, as well as a detailed runtime analysis.

We report results on four common video segmentation datasets: YouTube Objects [9], SegTrackv2 [14], egoMotion [16], and FBMS [17]. Regarding the largest and, thus, most relevant YouTube Objects dataset, we perform 3% better than the current state of the art on video segmentation methods without user interaction. The robustness of our method is further demonstrated by the good results on the other datasets with average performance gains of up to 16%.

2 Related Work

Video segmentation has quite some overlap with tracking, especially in the case here, where object detections are propagated over time. The typical tracking scenario is based on bounding boxes and usually does not provide accurate object contours. Tracking-by-detection is a popular approach to find consistent tracks with only little supervision [2,4,10]. Since this is a field of its own, we only review the most related works in the context of video segmentation. Prest *et al.* [19] generate detections in each frame, which are subsequently tracked in a greedy fashion. Dong *et al.* [29] use the appearance and motion based proposals from

Lee *et al.* [13] to build a graph and extract the longest tube. In the recent work of Weinzaepfel *et al.* [26], convolutional neuronal networks generate the features for tracking. The work of Hua *et al.* [8] uses some intermediate motion segmentation to model occlusion in bounding box tracking.

Supervised video segmentation methods achieve good results at the expense of user interaction for each video to be segmented [3,9,16,24]. The most popular procedure here is to annotate a single frame and the algorithm propagates the information and segments accordingly. In general this works well, but as new objects enter the scene, these methods fail or additional user input is required.

Unsupervised video segmentation is usually based on motion to a certain degree. In motion segmentation, motion is the only feature for localizing objects. Ochs *et al.* [17] and Keuper [11] cluster long term point trajectories. Papazoglou and Ferrari [18] use optical flow to compute so-called inside-outside maps, partitioning the frames into foreground and background. Yang *et al.* [28] use motion to detect disoccluded areas and assign them to the correct object. The common drawback of pure motion segmentation is the need for distinct motion of the objects and the background. Lee *et al.* [13] employ object proposals [5] to enhance motion cues with a set of static features. A sequence of min-cuts generates the figure ground segments in the work of Li *et al.* [14]. Multiple paths connecting the segments are extracted and post-processed, resulting in a set of multiple possible segmentations. Dong *et al.* [29] enforce the temporal consistency of object proposals via optical flow. Wang&Wang [25] discover reoccurring objects in the video, from which they estimate a holistic model. Yang *et al.* [27] estimate the appearance and the segmentation simultaneously by adding auxiliary nodes to the Markov random field model.

The work of Prest *et al.* [20] uses point trajectories like Ochs *et al.* [17] and Keuper [11] to identify objects. To assign class labels to the object regions, they jointly optimize over videos with the same class label. Hartmann *et al.* [7] and Tang *et al.* [23] also use the video tag to train a classifier for frame wise segments.

In Zhang *et al.* [30], frame-by-frame detections and segmentation proposals are combined to a temporally consistent semantic segmentation. The combination of a detector and a video segmentation approach is similar in spirit to our work, but technically, the approach is very different. We directly compare to Zhang *et al.* on the YouTube dataset. Also in the recent work of Seguin *et al.* [22], object tracking (either manual or via detection) guides a multiple instance segmentation. However, they do not make use of motion information, thus ignoring a powerful bottom-up cue in videos.

3 Video Object Segmentation

The key input to our video object segmentation are so-called tubes. Individual tubes are generated by tracking the detections of an off-the-shelf R-CNN detector [6]. The subsequent spatio-temporal segmentation is guided by this initial localization and the corresponding appearance cues.

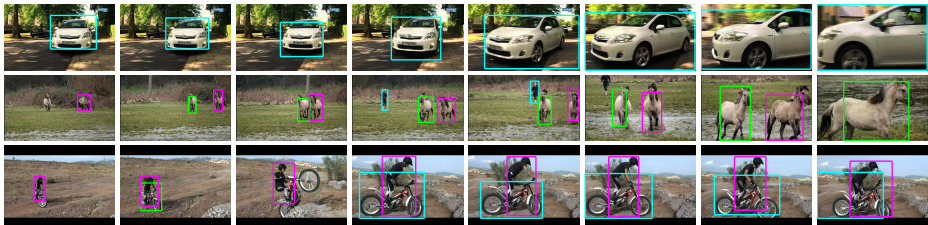


Fig. 2. Tube extraction on three shots from the Youtube dataset. Neither strong changes in viewpoint, as in the first example, nor multiple instances of the same object class, as in the second example, are a problem for our approach. The third example shows a wrong classification of the motorcycle as a bicycle in the second image (green box). The method recovers later from this failure case.

3.1 Tube Extraction

Good object tubes relief us from misleading motion cues, which typically occur when the camera motion dominates the object motion, the object does not move at all, or multiple objects move as a unit.

The initial set of detections is generated by classifying the fast edge boxes from Zitnick and Dollar [31] with the R-CNN from Girshick *et al.* [6]. We denote the set of detected boxes with \mathcal{B} and the i th detection in frame t with B_t^i . Extracting a consistent tube over time translates to finding the longest path in a graph that connects all bounding boxes of a frame with all bounding boxes in successive frames:

$$P^* = \operatorname{argmax}_{P \subseteq \mathcal{B}} \sum_{t_1 < t_2} S(B_{t_1}^i, B_{t_2}^j), \quad (1)$$

where $S(\cdot, \cdot)$ measures the similarity between two detection $B_{t_1}^i$ and $B_{t_2}^j$; see Figure 4 for an illustration of the graph. Note that there can be multiple frames between t_1 and t_2 . Consistent detections with only little or no change in appearance, position and shape reflect in high similarity values. The exact formulation of the proposed similarity metric involves several terms, for which we refer to Section 4.1, where we also provide associated ablation studies. An exemplary tube extraction is shown in Figure 4 and some qualitative results are shown in Figure 2.

In general this problem is NP-hard, but in this setting where we have a directed acyclic graph, we find the global optimum in linear time using a dynamic programming approach comparable to [29]. The used topology in [29] might look similar to our graph, but we encode the node weights already in the edges and the similarity metric is a different one.

Apart from increasing robustness, the detector labels the objects, which can be crucial for many high-level vision tasks.

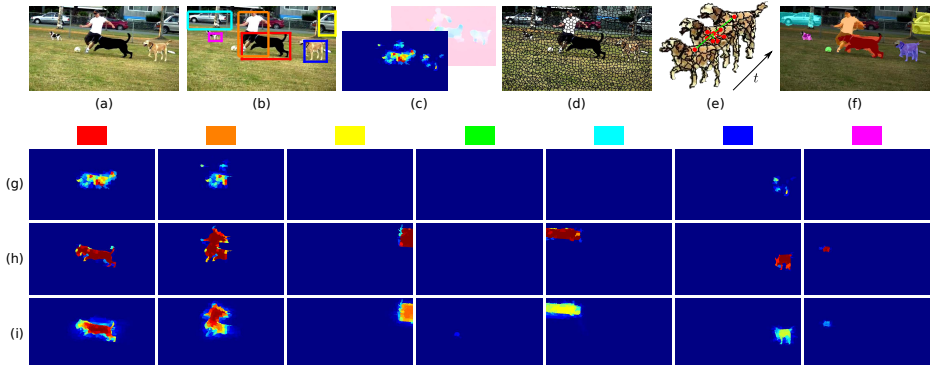


Fig. 3. Top row: Overview of the segmentation pipeline. **(a)** One frame of the video. **(b)** In the video, we detect spatio-temporal object tubes. **(c)** The optical flow and the tubes together yield inside-outside maps. **(d)** Superpixels. **(e)** Spatio-temporal graph based on these superpixels. **(f)** Final segmentation. **Bottom row:** Location priors for the objects detected in this example (columns correspond to the bounding boxes in the respective color). **(g)** Inside-outside maps based on optical flow. **(h)** Grab cut segmentations within the detected bounding boxes. **(i)** Combination of both and propagated over time.

3.2 Segmentation

We formulate the segmentation as an energy minimization problem with a unary and a pairwise term:

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} E_U(\mathbf{u}) + E_P(\mathbf{u}), \quad (2)$$

where \mathbf{u} assigns to each node in the video a label $\{1, \dots, K\}$. The scaffold of the most supervised (e.g. [9, 16]) and unsupervised (e.g. [18, 27]) video segmentation algorithm is a spatio-temporal graph $G = (\mathcal{V}, \mathcal{E})$ that primarily enforces consistency within the frame and over time. Long and higher resolution video shots make pixel-level segmentation computationally too demanding. Therefore the nodes $v \in \mathcal{V}$ and the final segmentation are on super-pixel level. We use superpixels from Achanta *et al.* [1].

Regarding the edges, we distinguish between spatial and temporal connections. Adjacent super-pixels build a spatial edge and super-pixels connected by optical flow build a temporal edge. For an example see Figure 3 (e).

Similar to [18, 27], the weighting $\lambda_{(v_1, v_2)}$ of the spatial edges is proportional to the color similarity, whereas it depends on the number of matched pixels for the temporal domain.

Unary Potential We extract the location prior L and the appearance model A from motion features and the tubes. The sum builds the unary term:

$$E_U(\mathbf{u}) = \sum_{v \in \mathcal{V}} \left(L_{t(v)}^{\mathbf{u}(v)}(v) + A_{t(v)}^{\mathbf{u}(v)}(v) \right), \quad (3)$$

where $t(v)$ is the time of super-pixel v and $\mathbf{u}(v)$ the corresponding label.

For the location prior L , we first partition the inside-outside maps M_t [18]. These maps classify a pixel as object, if it lies within an area surrounded by motion boundaries. We use the object tubes to restrict M_t :

$$M_t^i = B_t^i \cap M_t, \quad (4)$$

with the box B_t^i , so that we can reliably distinguish the motion between different objects. Additionally, motion of the background or the camera is suppressed.

We complement the motion features with foreground features F_t^i by segmenting the individual boxes of the tubes with GrabCuts [21]. This feature becomes valuable, especially when the motion is constant or unreliable.

From the union of the two sets $F_t^i \cup M_t^i$, we directly compute the respective appearance models A^i as Gaussian mixture models, where the background is modeled as complement of all tubes.

Temporal smoothing of the combined motion and foreground features gives us the location prior L . We use an optical flow propagation in a similar fashion as [18] to remove single bad motion or foreground estimates and carry the information beyond the endings of the tube.

A comprehensive example of this process, including the different location and foreground maps as well as the location prior L is given in Figure 3.

Pairwise Potential The pairwise term, enforcing spatial and temporal smoothness is a weighted Potts model:

$$E_p(\mathbf{u}) = \sum_{(v_1, v_2) \in \mathcal{E}} \delta(\mathbf{u}(v_1), \mathbf{u}(v_2)) \cdot \lambda_{(v_1, v_2)}, \quad (5)$$

where δ is the Kronecker delta and $\lambda_{(v_1, v_2)}$ is the edge weight.

We efficiently minimize the submodular energy with the Fast_PD solver from Komodakis and Tziritas [12].

4 Implementation Details

4.1 Similarity Measure

Favoring consistency in position, shape and appearance, reflects in the following similarity measure for two detections B_{t_1}, B_{t_2} :

$$\begin{aligned} S(B_{t_1}, B_{t_2}) = & score(B_{t_2}) \cdot S_{category}(B_{t_1}, B_{t_2}) \cdot S_{app}(B_{t_1}, B_{t_2}) \\ & \cdot S_{vol}(B_{t_1}, B_{t_2}) \cdot S_{side}(B_{t_1}, B_{t_2}) \cdot S_{match}(B_{t_1}, B_{t_2}) \\ & \cdot S_{center}(B_{t_1}, B_{t_2}). \end{aligned} \quad (6)$$

With the category label, the appearance and center term, we favor a consistent appearance of the object. The side and volume constraints enforce the tube to

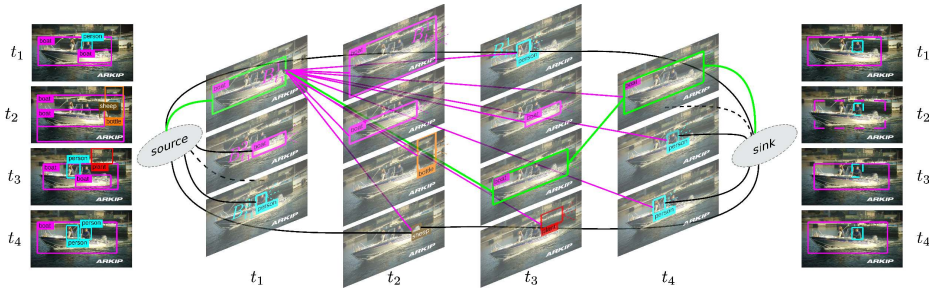


Fig. 4. Illustration of the graph structure for a sample video. The left column shows all initial detections, the right column the two final high scoring tubes that have been extracted: the boat (magenta) and the person (cyan). Dashed boxes indicate parts of the tube that have been interpolated between frames. The graph structure is shown in the middle, where we show exemplified all edges for $B_{t_1}^1$. The corresponding longest path is shown in green.

change its shape smoothly. Temporal consistency is encoded in the matching term, and the score rewards confident detections.

The category label is a very powerful indicator and has to be consistent through time. Therefore we set

$$S_{category} = \begin{cases} 1 & \text{if } category(B_{t_1}) = category(B_{t_2}) \\ -\infty & \text{else .} \end{cases} \quad (7)$$

Due to movement of the camera and/or the object itself, the bounding box can change over time. This is supposed to be a rather slow process, therefore we favor small changes in both, the volume and the sides.

$$S_{vol} = \min \left(\frac{Vol(B_{t_1})}{Vol(B_{t_2})}, \frac{Vol(B_{t_2})}{Vol(B_{t_1})} \right) \quad (8)$$

The cost for the side change is computed in the same way, where we take the minimum of the height and the width change. The matching term gives the ratio on how many points of B_{t_2} are matched by the optical flow F originating from B_{t_1} .

$$S_{match} = \frac{|Matches|}{Vol(B_{t_2})} \quad (9)$$

$$Matches = \{\mathbf{p} \in B_{t_2} | \exists \mathbf{q} \in B_{t_1} : F(\mathbf{q}) = \mathbf{p}\}$$

Although the optical flow is an indicator on how similar the two boxes are, the volume of B_{t_2} and the possible distinct motion of object and background weaken this term.

We compensate for that by additionally penalizing the deviation of the propagated center c_p of box B_{t_1} with the actual center c of B_{t_2} .

$$S_{center} = \frac{1}{1 + 0.1 \cdot \|c_p - c\|} \quad (10)$$

Correlating B_{t_1} with frame t_2 gives us the propagated center c_p . On a finer level, this is less accurate than optical flow, but it is more robust.

The appearance term is the cosine-distance of color histograms $H(\cdot)$:

$$S_{app} = \frac{\langle H(B_{t_1}), H(B_{t_2}) \rangle}{\|H(B_{t_1})\| \cdot \|H(B_{t_2})\|}. \quad (11)$$

This cue is independent of the area and shape given by the corresponding bounding box. Boxes with $S_{app} \leq 0.8$ are considered as distinct and the term is set to $-\infty$. When objects of the same class interact (e.g. overtaking cars, Figure 5), the appearance is important to track them correctly.

4.2 Graph

We build the graph by connecting the detections in a temporal order. Since we cannot assume that the detections are present in every frame we interconnect each detection with the detections of the subsequent 20 frames.

Additionally, each node is connected to the source and sink, so that new objects can enter and leave the scene, while being correctly tracked without introducing additional knowledge about the object’s presence in the video. See Figure 4 for a visualization of such a graph.

4.3 Post-processing

We interpolate the possible sparse tube into a dense one. The missing box in frame t is interpolated by correlating the box found in frame $(t - 1)$ with frame t . Gaps of more than one frame are interpolated from both sides.

The set of tubes is cleaned by a volumetric non-maximum suppression, where overlapping tubes of the same class with an intersection over union > 0.5 are suppressed by the longer, respective higher scoring tube.

4.4 Tube Parameter Evaluation

We give a justification of the different terms in our proposed similarity measure in form of an ablation study in Table 1. For the evaluation, we selected a subset of the SegTrackv2 dataset (birds_of_paradise, bmx, drift, girl, monkey, penguin and soldier) that excludes the categories, for which we have either no detector or not sufficient detections. With no or little detections, tracking is almost independent of different similarity measures. Note that we favored the SegTrackv2 dataset as it provides annotation for every frame and allows for a more detailed analysis. Having only every 10th frame annotated (YouTube), flickering or swapping of labels between objects (as in the drift sequence, Figure 5) will be missed by the evaluation metric.

We report the quality of the tube as IoU (intersection over union) between the boxes of the tube and the boxes spanned by the GT-annotation. We judge the impact of the different components by two experiments, first we drop the

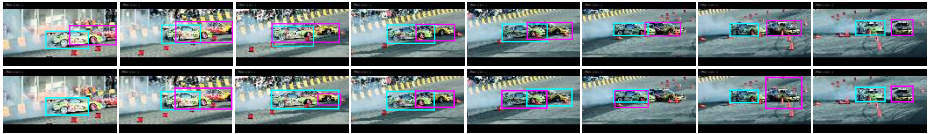


Fig. 5. Relevance of the appearance constraint S_{app} . Our tube extraction result (**top row**) gets significantly worse (**bottom row**) when we drop S_{app} , as the two cars get mixed up.

component (-) and measure how the performance decreases and second we use the specific component as sole similarity measure (+). While the best performance is achieved by using all components, the performance depends most on the appearance term, which is exemplified in Figure 5.

| | <i>score</i> | S_{side} | S_{vol} | S_{vol} | S_{side} | S_{match} | S_{center} | S_{match} | S_{center} | S_{app} | all |
|---|--------------|------------|-----------|-----------|------------|-------------|--------------|-------------|--------------|-----------|--------------|
| - | 58.75 | 59.7 | 59.5 | 59.0 | 57.8 | 59.13 | | 57.85 | | 55.73 | 59.83 |
| + | 53.47 | 55.07 | 54.33 | 53.99 | 54.75 | 54.81 | | 54.21 | | 57.39 | 59.83 |

Table 1. Ablation study of the tube extraction on a subset of SegTrackv2, reported as IoU. Removing components from the system makes results worse (-). Some components are more important than others. The performance of the individual components (+) confirm that the appearance term is the most significant

4.5 Resolving Oversegmentation

When objects suddenly deform or rapidly change their appearance, it is likely that they are tracked by multiple tubes. Lowering the restriction of the tube is not a solution since the tubes would lose consistency. Especially when tracking multiple objects, it is important that the tube does not jump between different objects (see Figure 5).

Multiple tubes lead to a later oversegmentation. We use the location prior L and the class label, to merge cohesive tubes. If the correlation between different locations priors with the same class label is ≥ 0.5 , we fuse the tubes. Figure 6 gives an typical example.

5 Results

With the evaluation on four video segmentation datasets (YouTube Objects [9], egoMotion [16], SegTrackv2 [14] and FBMS [17]), we prove the performance of the proposed method. The datasets impose different challenges and shortcomings.

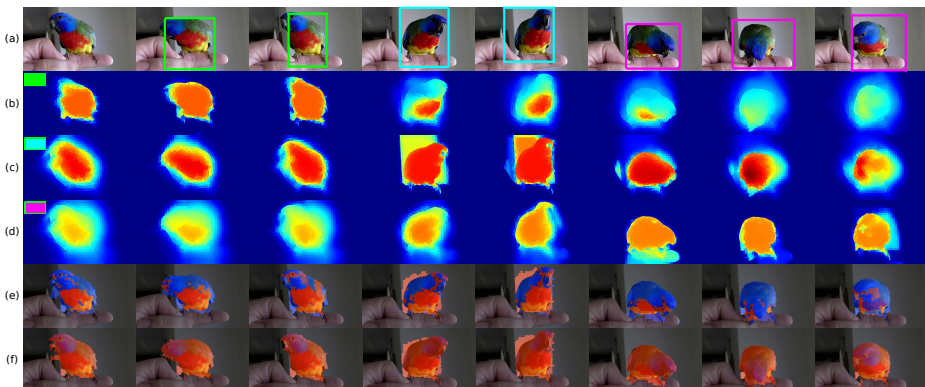


Fig. 6. Multiple tubes tracking the same object (a) are a result of quick changes in appearance, shape or position and lead to oversegmentation (e). Using the location cues (b-d), we merge the tubes and consistently segment the bird (f).

EgoMotion and FBMS are complementary. In egoMotion, there is always a single object that is largely static, whereas FBMS contains multiple moving objects. The downside of FBMS is that there is no ground-truth annotation for static objects, because it is a benchmark dataset designed for motion segmentation.

SegTrackv2 and YouTube Objects are the most relevant datasets, since they are composed of a variety of different settings. For video level segmentation YouTube Objects have become the dataset on which state-of-the-art methods report their results.

5.1 YouTube

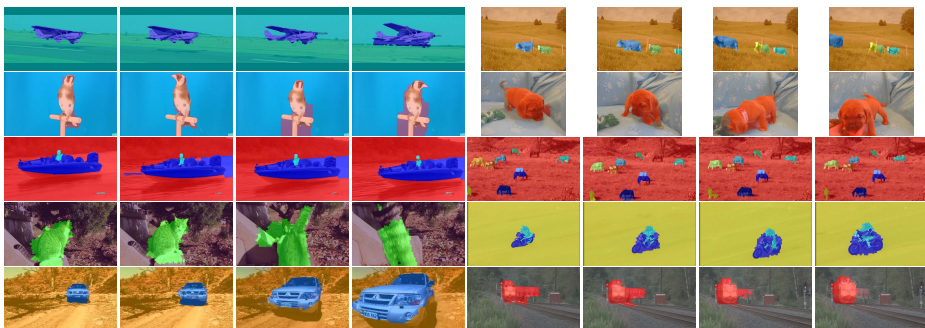


Fig. 7. Results for the 10 different object categories of the YouTube dataset. The proposed method distinguishes connected objects (boat, motorbike). Rapidly moving, non-rigid objects (cat) cause some problems.

126 Videos from 10 different classes make up the YouTube Objects dataset [20]. Jain *et al.* [9] provide ground-truth annotation on a super-pixel level for approximately every 10th frame. The evaluation metric is the average intersection over union

$$IOU = \frac{|S \cap GT|}{|S \cup GT|}, \quad (12)$$

where S is the segmentation and GT the ground truth.

With the proposed tubes, an implicit segmentation is already given by the foreground features F (tube cut). Besides our full segmentation, we evaluate two further approaches. First we use only the partitioned motion features M_t^i as prior for the segmentation (OURS-M). Analogously, we evaluate the segmentation based on pure appearance features F_t^i (OURS-F).

The results in Table 2 show that the foreground features without further processing perform similar to pure motion based segmentation. The best result is achieved by the combination of motion and appearance cues, which beats current state of the art methods by at least 3%.

| | | | | | | | | | | | | |
|----------|------|------|------|------|------|------|------|-------|-----------|-------|-------------|-------------|
| | aero | bird | boat | car | cat | cow | dog | horse | motorbike | train | avg cat | avg vid |
| [17] | 13.7 | 12.2 | 10.8 | 23.7 | 18.6 | 16.3 | 18.0 | 11.5 | 10.6 | 19.6 | 15.5 | – |
| [23] | 17.8 | 19.8 | 22.5 | 38.3 | 23.6 | 26.8 | 23.7 | 14.0 | 12.5 | 40.4 | 23.9 | 22.8 |
| [18] | 65.5 | 69.2 | 43.9 | 66.1 | 49.3 | 38.0 | 50.0 | 31.6 | 31.9 | 34.0 | 47.9 | 45.7 |
| [30] | 75.8 | 60.8 | 43.7 | 71.1 | 46.5 | 54.6 | 55.5 | 54.9 | 42.4 | 35.8 | 54.1 | 52.6 |
| [25] | 63.0 | 69.0 | 40.0 | 61.0 | 48.0 | 46.0 | 67.0 | 53.0 | 47.0 | 38.0 | 53.0 | – |
| [15] | 59.3 | 67.6 | 32.6 | 50.5 | 33.1 | 27.4 | 35.6 | 46.0 | 18.4 | 47.3 | 41.8 | 37.3 |
| tube cut | 72.3 | 46.4 | 56.6 | 47.0 | 30.1 | 55.7 | 39.4 | 47.9 | 35.1 | 36.7 | 46.7 | 45.8 |
| OURS-M | 65.0 | 72.7 | 49.1 | 68.9 | 49.9 | 49.6 | 54.4 | 39.0 | 37.2 | 37.0 | 52.3 | 51.2 |
| OURS-F | 73.3 | 67.0 | 60.0 | 57.3 | 34.5 | 62.4 | 54.7 | 54.6 | 42.1 | 38.1 | 54.4 | 54.0 |
| OURS | 74.4 | 72.1 | 58.5 | 60.0 | 45.7 | 61.2 | 55.2 | 56.6 | 42.1 | 36.7 | 56.2 | 55.8 |

Table 2. Results for the YouTube dataset, reported as IoU. The proposed method performs 3% better than the current state-of-the-art method [30]. The combination of motion and appearance features lead to the best performance.

5.2 SegTrackv2

The SegTrackv2 dataset [14] consists of 14 videos with frame-wise ground-truth annotation. Single and multiple objects, slow and fast motion, as well as occluding and interacting objects are present. Note that only in sequences, where known objects are present, our method can perform well. When processing sequences with unknown objects such as parachute or worm, we can only rely on motion features and we fall back to the approach in [18]. Regarding cases, in which we can extract tubes, e.g. bmx or drift sequence, we clearly outperform the other methods. On average, we are 4.3% better than the other methods; see Table 3. Qualitative results and comparisons are shown in Figure 8.

| | [17] | [11] | [18] | [15] | OURS |
|-------------------|-------------|-------------|-------------|-------------|-------------|
| bird_of.paradise | 17.2 | 79.0 | 74.9 | 43.2 | 50.5 |
| birdfall | 0.5 | 0.5 | 4.5 | — | 4.5 |
| bmx-person | 4.8 | 70.4 | 47.8 | 0.9 | 90.7 |
| bmx-bike | 1.2 | 17.3 | 16.3 | 20.0 | 33.5 |
| cheetah-deer | 1.9 | 1.9 | 47.1 | — | 41.9 |
| cheetah-cheetah | 0.9 | 3.9 | 17.9 | — | 0 |
| drift-car1 | 35.1 | 50.2 | 48.4 | 36.0 | 70.1 |
| drift-car2 | 12.4 | 0.3 | 35.0 | 39.6 | 60.2 |
| frog | 41.5 | 43.1 | 57.3 | — | 68.4 |
| girl | 52.1 | 51.4 | 53.8 | 65.8 | 65.4 |
| monkey | 35.8 | 22.8 | 64.8 | — | 65.2 |
| monkeydog-dog | 1.4 | 6.8 | 0 | 0 | 0 |
| monkeydog-monkey | 54.9 | 54.9 | 77.7 | 0 | 77.7 |
| hummingbird-bird1 | 3.9 | 11.0 | 10.1 | 39.8 | 10.4 |
| hummingbird-bird2 | 55.4 | 32.4 | 51.6 | 30.2 | 9.4 |
| parachute | 90.3 | 89.8 | 68.7 | — | 68.7 |
| penguin-penguin1 | 8.5 | 8.6 | 4.7 | 20.0 | 43.0 |
| penguin-penguin2 | 3.8 | 4.1 | 1.9 | 0.7 | 0 |
| penguin-penguin3 | 0 | 3.8 | 1.7 | 14.9 | 0 |
| penguin-penguin4 | 0 | 0 | 2.2 | 0 | 0 |
| penguin-penguin5 | 0 | 0 | 8.9 | 0 | 0 |
| penguin-penguin6 | 0 | 5.3 | 18.3 | 0.61 | 73.6 |
| soldier | 63.0 | 50.1 | 36.9 | 0 | 64.0 |
| worm | 2.7 | 23.0 | 69.0 | — | 69.0 |
| avg obj | 27.9 | 34.5 | 43.7 | 24.8* | 48.0 |
| avg vid | 20.3 | 26.3 | 34.2 | 18.3* | 40.3 |

Table 3. Quantitative results for the SegTrackv2 dataset, reported as IoU. (*) Results are averaged over the videos containing objects from the 20 pascal classes.

| | car | cat | chair | dog | average |
|------|-------------|-------------|-------------|-------------|-------------|
| [17] | 33.6 | 13.5 | 16.2 | 41.7 | 26.3 |
| [11] | 37.9 | 45.3 | 19.8 | 53.4 | 39.1 |
| [18] | 47.6 | 56.6 | 59.5 | 64.2 | 57.0 |
| [15] | 86.1 | 16.6 | 39.0 | 47.1 | 47.2 |
| OURS | 78.0 | 65.7 | 73.5 | 75.2 | 73.1 |

Table 4. Results reported as IoU for the egoMotion dataset [16]. We have a 16% improvement compared to motion segmentation approaches.

5.3 egoMotion

The egoMotion dataset [16] consists of 24 videos from 4 categories (cars, cats, chairs, dogs), where each video features a single object. The main challenge is the dominant camera motion, which is hard to handle for pure motion based methods. The extraction of the tubes give us a vital prior, resulting in 73.1

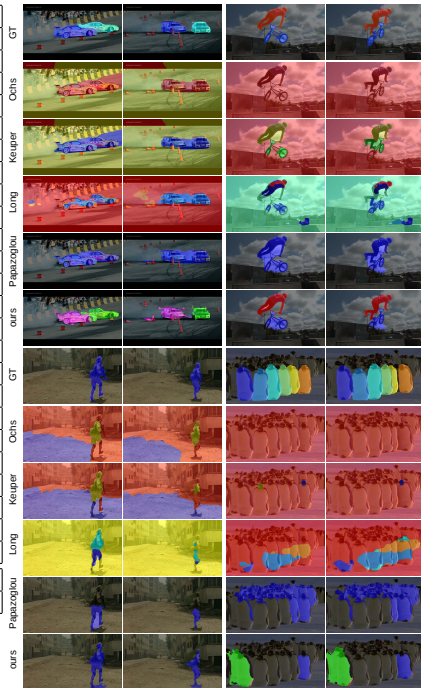


Fig. 8. Results for the SegTrackv2 dataset. From top to bottom: ground truth, Ochs *et al.* [17], Keuper *et al.* [11], Papazoglou and Ferrari [18] and ours. The example from the drift sequence shows that motion based methods fail when the objects are close to each other and move similarly. In the second frame the two pylons in the upper left corner are detected as traffic lights. For the penguin sequence, we detected only two tubes and thus did not segment the remaining penguins.

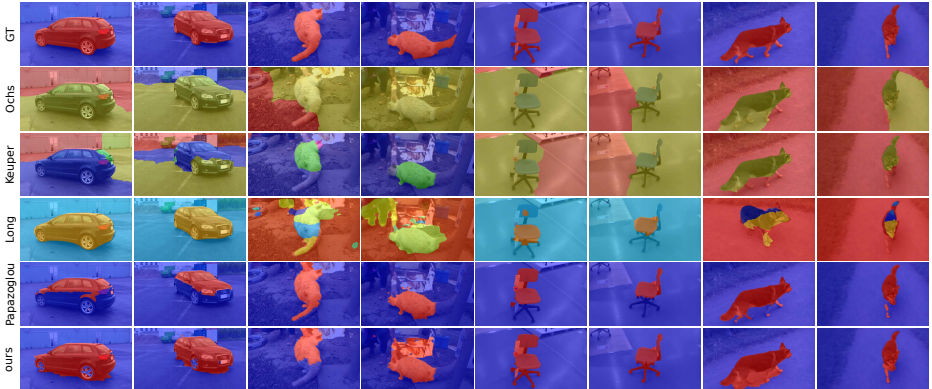


Fig. 9. Results for the egoMotion dataset [16]. **From top to bottom:** Ground truth, Ochs *et al.* [17], Keuper *et al.* [11], Long *et al.* [15], Papazoglou and Ferrari [18] and ours. For static objects, the image motion at the bottom of the object and the ground is the same. Thus, pure motion segmentation methods fail.

(IoU), whereas the best motion based method [18] achieves 57.0; compare Table 4.

5.4 FBMS

The Freiburg-Berkely-Motion-Segmentation dataset [17] shows a great variety of different moving objects over 59 videos. Regarding the evaluation we follow [17] and report the results as average precision and recall, where

$$P_{i,j} = \frac{|S_i \cap GT_j|}{|S_i|} \quad R_{i,j} = \frac{|S_i \cap GT_j|}{|GT_i|} \quad F_{i,j} = \frac{2P_{i,j}R_{i,j}}{P_{i,j} + R_{i,j}} \quad (13)$$

The assignments between segmentations and ground-truth are chosen so that the F-measure is maximized. An object is counted as successfully segmented, if $F \geq 0.75$, where the background does not count as an object, so the number is reduced by 1. Our method clearly performs better than [17] and [18], both in terms of the F-measure and the number of segmented objects. However, [11] and [28] achieve better results; compare Table 5. In Figure 10, we reveal some of the cases that reduce our performance. The missing annotation of static objects (parking cars) and the labeling of different objects as one moving unit (rider on horse) decrease our performance.

5.5 Runtime and Scalability

The average runtime is ~ 8 seconds per frame. Table 6 gives an detailed overview of the contribution of the different components. The main costs (in seconds) are caused by running the object detector (1.53), GrabCuts (2.1), Optical Flow (1.04) and the correlation of boxes (1.52).

| | Training | | | | Test | | | |
|------|----------|--------|---------------|---------------|--------|--------|---------------|---------------|
| | P | R | F | $F \geq 75\%$ | P | R | F | $F \geq 75\%$ |
| [17] | 81.50% | 63.23% | 71.21% | 16/65 | 74.91% | 60.14% | 66.72% | 20/69 |
| [11] | 85.31% | 68.70% | 76.11% | 24/65 | 85.95% | 65.07% | 74.07% | 23/69 |
| [18] | 85.86% | 61.85% | 71.90% | 13/65 | 88.72% | 54.84% | 67.78% | 14/69 |
| [28] | 89.53% | 70.74% | 79.03% | 26/65 | 91.47% | 64.75% | 75.82% | 27/69 |
| [15] | 79.03% | 63.66% | 70.52% | 13/65 | 75.36% | 59.66% | 66.60% | 19/69 |
| OURS | 84.27% | 66.48% | 74.32% | 20/65 | 86.76% | 63.28% | 73.18% | 23/69 |

Table 5. Quantitative evaluation on the FBMS dataset [17].



Fig. 10. Qualitative results for the FBMS dataset [17]. From top to bottom: ground truth, Ochs *et al.* [17], Keuper *et al.* [11], Long *et al.* [15], Papazoglou and Ferrari [18] and ours. Typical failure cases of our method on this dataset are: the segmentation of static objects, e.g. the white car (segmented in cyan) in the car sequence. Splitting moving objects, e.g. the rider and horse. For the giraffe sequence, the extracted tubes switch the object leading to an inconsistent segmentation.

The scalability is analyzed in Figure 11, where we observe a linear behavior for both, the tube extraction and the segmentation.

6 Conclusions

We have presented a video object segmentation algorithm that combines object detection with bottom-up motion and appearance cues. The detection makes the segmentation robust against a variety of challenges in pure bottom-up methods and provides a class label for each object instance. In cases where the detector is not available, because the object class is unknown, the method falls back to a bottom-up approach and can still perform very well. Our evaluation on four

| components | Tube extraction | | | | | | | Segmentation | | | | | | | |
|-------------|-----------------|-----------|------------|-------------|---------|-------|--------------|--------------|-------------|---------|----------|---------|------------------|---------------------|-------|
| | Optical Flow | Detection | Histograms | Correlation | File IO | Graph | Longest path | Build tube | Superpixels | File IO | Grabcuts | IO Maps | Unary potentials | Pairwise potentials | MRF |
| time in sec | 1.04 | 1.53 | 0.58 | 1.52 | 0.04 | 0.05 | 0.004 | 0.42 | 0.34 | 0.21 | 2.1 | 0.18 | 0.28 | 0.03 | 0.002 |
| | 4.14 | | | | | | | 3.14 | | | | | | | |
| | 8.33 | | | | | | | | | | | | | | |

Table 6. Runtime for the video segmentation and its components in seconds per frame.

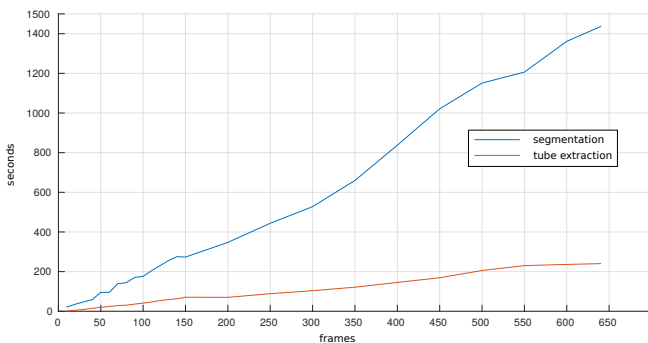


Fig. 11. The runtime of the segmentation (blue) and tube (red) scales linear with the number of frames. Detection, optical flow and super pixel computation take constant time per frame.

video segmentation datasets showed that we achieve state-of-the-art performance except for the FBMS dataset due to the missing annotation of static objects.

Acknowledgements

This work was funded by the ERC Starting Grant VideoLearn.

References

1. ACHANTA, R. ; SHAJI, A. ; SMITH, K. ; LUCCHI, A. ; FUA, P. ; SUSSTRUNK, S. : SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (2012), Nov., Nr. 11
2. ANDRILUKA, M. ; ROTH, S. ; SCHIELE, B. : People-Tracking-by-Detection and People-Detection-by-Tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008
3. BADRINARAYANAN, V. ; BUDVYTIS, I. ; CIPOLLA, R. : Mixture of Trees Probabilistic Graphical Model for Video Segmentation. In: *International Journal of Computer Vision* (2013)
4. BREITENSTEIN, M. D. ; REICHLIN, F. ; LEIBE, B. ; KOLLER-MEIER, E. ; GOOL, L. V.: Robust Tracking-by-Detection using a Detector Confidence Particle Filter. In: *IEEE International Conference on Computer Vision (ICCV)*, 2009
5. ENDRES, I. ; HOIEM, D. : Category Independent Object Proposals. In: *European Conf. on Computer Vision (ECCV)*, 2010, S. 575–588
6. GIRSHICK, R. ; DONAHUE, J. ; DARRELL, T. ; MALIK, J. : Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014
7. HARTMANN, G. ; GRUNDMANN, M. ; HOFFMAN, J. ; TSAI, D. ; KWATRA, V. ; MADANI, O. ; VIJAYANARASIMHAN, S. ; ESSA, I. ; REHG, J. ; SUKTHANKAR, R. : Weakly Supervised Learning of Object Segmentations from Web-scale Video. In: *European Conf. on Computer Vision (ECCV)*, 2012
8. HUA, Y. ; ALAHARI, K. ; SCHMID, C. : Occlusion and motion reasoning for long-term tracking. In: *European Conf. on Computer Vision (ECCV)*, 2014
9. JAIN, S. D. ; GRAUMAN, K. : Supervoxel-Consistent Foreground Propagation in Video. In: *European Conf. on Computer Vision (ECCV)*, 2014, S. 656–671
10. KALAL, Z. ; MIKOLAJCZYK, K. ; MATAS, J. : Tracking-Learning-Detection. In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2012), Jul.
11. KEUPER, M. ; ANDRES, B. ; BROX, T. : Motion Trajectory Segmentation via Minimum Cost Multicuts. In: *IEEE International Conference on Computer Vision (ICCV)*, 2015
12. KOMODAKIS, N. ; TZIRITAS, G. : Approximate Labeling via Graph Cuts Based on Linear Programming. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2007), Aug., Nr. 8
13. LEE, Y. J. ; KIM, J. ; GRAUMAN, K. : Key-segments for video object segmentation. In: *IEEE International Conference on Computer Vision (ICCV)*
14. LI, F. ; KIM, T. ; HUMAYUN, A. ; TSAI, D. ; REHG, J. M.: Video Segmentation by Tracking Many Figure-Ground Segments. In: *IEEE International Conference on Computer Vision (ICCV)*, 2013
15. LONG, J. ; SHELHAMER, E. ; DARRELL, T. : Fully Convolutional Networks for Semantic Segmentation. In: *CoRR* abs/1411.4038 (2014)
16. NAGARAJA, N. ; SCHMIDT, F. ; BROX, T. : Video Segmentation with Just a Few Strokes. In: *IEEE International Conference on Computer Vision (ICCV)*, 2015
17. OCHS, P. ; MALIK, J. ; BROX, T. : Segmentation of moving objects by long term video analysis. In: *IEEE Trans. Pattern Anal. Mach. Intell.*
18. PAPAIOGLOU, A. ; FERRARI, V. : Fast Object Segmentation in Unconstrained Video. In: *IEEE International Conference on Computer Vision (ICCV)*
19. PREST, A. ; FERRARI, V. ; SCHMID, C. : In: *IEEE Trans. Pattern Anal. Mach. Intell.*

20. PREST, A. ; LEISTNER, C. ; CIVERA, J. ; SCHMID, C. ; FERRARI, V. : Learning object class detectors from weakly annotated video. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
21. ROTHER, C. ; KOLMOGOROV, V. ; BLAKE, A. : "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In: *ACM Trans. Graph.* 23 (2004), Aug., Nr. 3, S. 309–314. – ISSN 0730–0301
22. SEGUIN, G. ; BOJANOWSKI, P. ; LAJUGIE, R. ; LAPTEV, I. : *Instance-level video segmentation from object tracks*
23. TANG, K. ; SUKTHANKAR, R. ; YAGNIK, J. ; FEI-FEI, L. : Discriminative Segment Annotation in Weakly Labeled Video. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013
24. VIJAYANARASIMHAN, S. ; GRAUMAN, K. : Active Frame Selection for Label Propagation in Videos. In: *European Conf. on Computer Vision (ECCV)*, 2012
25. WANG, H. ; WANG, T. : Primary object discovery and segmentation in videos via graph-based transductive inference. In: *Computer Vision and Image Understanding* 143 (2016)
26. WEINZAEPFEL, P. ; HARCHAOUI, Z. ; SCHMID, C. : Learning to Track for Spatio-Temporal Action Localization. In: *IEEE International Conference on Computer Vision (ICCV)*, 2015
27. YANG, J. ; PRICE, B. L. ; SHEN, X. ; LIN, Z. L. ; YUAN, J. : Fast Appearance Modeling for Automatic Primary Video Object Segmentation. In: *IEEE Trans. on Image Processing* (2016), Feb
28. YANG, Y. ; SUNDARAMOORTHY, G. ; SOATTO, S. : Self-Occlusions and Disocclusions in Causal Video Object Segmentation. In: *IEEE International Conference on Computer Vision (ICCV)*, 2015
29. ZHANG, D. ; JAVED, O. ; SHAH, M. : Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 0 (2013), S. 628–635. – ISSN 1063–6919
30. ZHANG, Y. ; CHEN, X. ; LI, J. ; WANG, C. ; XIA, C. : Semantic Object Segmentation via Detection in Weakly Labeled Video. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
31. ZITNICK, C. L. ; DOLLÁR, P. : Edge Boxes: Locating Object Proposals from Edges. In: *European Conf. on Computer Vision (ECCV)*