Proceedings, The Fourth AAAI Conference on
Human Computation and Crowdsourcing

# Click Carving: Segmenting Objects in Video with Point Clicks

**Suyog Dutt Jain and Kristen Grauman**
University of Texas at Austin
{suyog | grauman}@cs.utexas.edu
http://vision.cs.utexas.edu/projects/clickcarving/

## Abstract

We present a novel form of interactive video object segmentation where a few clicks by the user helps the system produce a full spatio-temporal segmentation of the object of interest. Whereas conventional interactive pipelines take the user's initialization as a starting point, we show the value in the system taking the lead even in initialization. In particular, for a given video frame, the system precomputes a ranked list of thousands of possible segmentation hypotheses (also referred to as object region proposals) using image and motion cues. Then, the user looks at the top ranked proposals, and clicks on the object boundary to carve away erroneous ones. This process iterates (typically 2-3 times), and each time the system revises the top ranked proposal set, until the user is satisfied with a resulting segmentation mask. Finally, the mask is propagated across the video to produce a spatio-temporal object tube. On three challenging datasets, we provide extensive comparisons with both existing work and simpler alternative methods. In all, the proposed Click Carving approach strikes an excellent balance of accuracy and human effort. It outperforms all similarly fast methods, and is competitive or better than those requiring 2 to 12 times the effort.

## Introduction

Video object segmentation entails computing a pixel-level mask for an object(s) across the frames of video, regardless of that object's category. Analogous to image segmentation, which produces a 2D map delineating the object's spatial region ("blob"), video segmentation produces a 3D map delineating the object's spatio-temporal extent ("tube"). The problem has received substantial attention in recent years, with methods ranging from wholly unsupervised bottom-up approaches (Grundmann et al. 2010; Xu, Xiong, and Corso 2012), to propagation methods that exploit user input on the first frame (Ren and Malik 2007; Tsai, Flagg, and Rehg 2010; Fathi et al. 2011; Vijayanarasimhan and Grauman 2012; Jain and Grauman 2014; Wen et al. 2015), to human-in-the-loop methods where a user closely guides the system towards a good segmentation output (Wang et al. 2005; Li, Sun, and Shum 2005; Bai et al. 2009; Shankar Nagaraja, Schmidt, and Brox 2015). Successful video segmentation algorithms have potential for significant impact on tasks like activity and object recognition, video editing, and abnormal event detection.

Despite very good progress in the field, it remains challenging to collect quality video segmentations at a large scale. The system is expected to segment objects for which it may have no prior model, and the objects may move quickly and change shape or appearance over time—or (often even worse) never move with respect to the background. To scale up the ability to generate well-segmented data, *human-in-the-loop* methods that leverage minimal human input are appealing (Ren and Malik 2007; Tsai, Flagg, and Rehg 2010; Badrinarayanan, Galasso, and Cipolla 2010; Fathi et al. 2011; Vondrick and Ramanan 2011; Vijayanarasimhan and Grauman 2012; Jain and Grauman 2014; Wen et al. 2015; Wang et al. 2005; Li, Sun, and Shum 2005; Bai et al. 2009; Shankar Nagaraja, Schmidt, and Brox 2015). These methods benefit greatly by combining the respective strengths of humans and machines. They can reserve for the human the more difficult high-level job of identifying a true object, while reserving for the algorithm the more tedious low-level job of propagating that object's boundary over time. This synergistic interaction between humans and computers results in accurate segmentations with minimal human effort.

Critical to the success of an interactive video segmentation algorithm is how the user interacts with the system. In particular, how should the user indicate the spatial extent of an object of interest in video? Existing methods largely rely on the tried-and-true interaction modes used for image labeling; namely, the user draws a bounding box or an outline around the object of interest on a given frame, and that region is propagated through the video either indefinitely or until it drifts (Ren and Malik 2007; Tsai, Flagg, and Rehg 2010; Fathi et al. 2011; Vijayanarasimhan and Grauman 2012; Jain and Grauman 2014; Wen et al. 2015). Furthermore, regardless of the exact input modality, the common assumption is to get the user's input *first*, and then generate a segmentation hypothesis thereafter. In this sense, in video segmentation propagation, information flows first from the user to the system.

We propose to reverse this standard flow of information. Our idea is for the system itself to first hypothesize plausible object segmentations in the given frame, and then allow the human user to efficiently and interactively prioritize those hypotheses. Such an approach stands to reduce human an-

notation effort, since the user can use very simple feedback to guide the system to its best hypotheses.

To this end, we introduce *Click Carving*, a novel method that uses point clicks to obtain a foreground object mask for a video frame. Clicks, largely unexplored for video segmentation, are an attractive input modality due to their ease, speed, and intuitive nature (e.g., with a touch screen the user may simply point a finger). Our method works as follows. First, the system precomputes thousands of mask hypotheses based on object proposal regions. Importantly, those object proposal regions exploit both image coherence cues as well as motion boundaries computed in the video. Then, the user efficiently navigates to the best hypotheses by clicking on the boundary of the true object and observing the refined top hypotheses.

Essentially, the user's clicks "carve" away erroneous hypotheses whose boundaries disagree with the clicks. By continually revising its top rated hypotheses, the system implicitly guides the user where input is most needed next. After the user is satisfied, or the maximal budget of clicks is exhausted, the system propagates the best mask hypothesis through the video with an existing propagation algorithm. For videos in the three datasets we tested, only 2-4 clicks are typically required to accurately segment the entire clip. Note that our novel idea is not so much about the "clicking" interface itself; rather our new ideas center around the idea of simple point supervision as a sufficient cue to perform semi-automatic segmentation and the carving backend that efficiently discerns the most reliable proposals.

Aside from testing our approach with real users, we also develop several simulated user clicking models in order to systematically analyze the relative merits of different clicking strategies. E.g., is it more effective to click in the object center, or around its perimeter? How should multiple clicks be spaced? Is it advantageous to place clicks in reaction to where the system currently has the greatest errors? One interesting outcome of our study is that the behavior one might assume as a default—clicking in the object's interior (Bearman et al. 2015; Wang, Han, and Collomosse 2014)—is much less effective than clicking on its boundaries. We show that boundary clicks are better able to discriminate between good and bad object proposal regions.

The results show that Click Carving strikes an excellent balance of accuracy and human effort. It is faster (requires less annotation interaction) than most existing interactive methods, yet produces better results. In extensive comparisons with state-of-the-art methods on three challenging datasets we show that Click Carving outperforms all similarly fast methods, and is competitive or better than those requiring 2 to 12 times the effort. This large reduction in annotion time by effective use of human interaction naturally leads to large savings in annotation costs. Because of the ease with which our framework can assist even non-experts in making high quality annotations, it has great promise for scaling up video segmentation. Ultimately such tools are critical for accelerating data collection in several research communities (e.g., computer vision, graphics, medical imaging), where large-scale spatio-temporal annotations are lacking and/or often left to experts.

## Related Work

Unsupervised video segmentation methods use no human input, and typically produce an over-segmentation of the video that is useful for mid-level grouping. Supervoxel methods find space-time blobs cohesive in color and/or motion (Grundmann et al. 2010; Xu, Xiong, and Corso 2012), while point trajectory approaches find consistent motion threads beyond optical flow (Brox and Malik 2010; Lezama et al. 2011). Unlike unsupervised work, we consider interactive video segmentation and our method produces spatio-temporal tubes covering the extent of the complete object.

Other methods extract "object-like" segments in video (Lee, Kim, and Grauman 2011; Papazoglou and Ferrari 2013), typically by learning the category-independent properties of good regions, and employing some form of tracking. Related are the methods that generate a large number of bounding box or region *proposals* (Li et al. 2013; Oneata et al. 2014) for videos, an idea originating in image segmentation (Carreira and Sminchisescu 2012; Arbeláez et al. 2014). The idea is to maintain high recall for the sake of downstream processing. As such, these methods typically produce many segmentation hypotheses, 100s to 1000s for today's popular datasets. To adapt them for the object segmentation problem would require human inspection to select the best one, which is non-trivial once the video contains more than a handful. Our approach makes use of object proposals, but our idea to prioritize them with Click Carving is entirely new. We are the first to propose using proposals for the sake of speeding up interactive segmentation, whether for images or videos.

Semi-automatic video segmentation methods accept manually labeled frame(s) as input, and propagate the annotation to the remaining video clip (Ren and Malik 2007; Tsai, Flagg, and Rehg 2010; Badrinarayanan, Galasso, and Cipolla 2010; Fathi et al. 2011; Vijayanarasimhan and Grauman 2012; Jain and Grauman 2014; Wen et al. 2015). Often the model consists of an MRF over (super)pixels or supervoxels, with both spatial and temporal connections. Some systems actively guide the user how to annotate (Fathi et al. 2011; Vondrick and Ramanan 2011; Vijayanarasimhan and Grauman 2012). All the prior methods assume that initialization starts with the user, and all use traditional modes of user input (bounding boxes, scribbles, or outlines). In contrast, we explore the utility of clicks for video segmentation, and we propose a novel, interactive way to perform system-initiated initialization. We show our approach achieves comparable performance to drawing complete outlines, but with much less annotation effort. Following our novel user interaction stage, we leverage our existing method (Jain and Grauman 2014) to do the propagation.

Human-in-the-loop interactive video segmentation methods also leverage user input, but unlike the above propagation methods, the user engages in a back and forth until the video is adequately segmented (Wang et al. 2005; Bai et al. 2009; Shankar Nagaraja, Schmidt, and Brox 2015). In all existing methods, the user guides the system to generate a segmentation hypothesis, and then iteratively corrects

(a) Image (b) Static Boundaries (c) Static Proposals (d) Motion Boundaries (e) Motion Proposals

Figure 1: Generation of object region proposals using both static and dynamic cues. Best viewed on pdf.

mistakes by providing more guidance. In contrast, Click Carving pre-generates thousands of possible hypotheses and then employs user guidance to efficiently filter high quality segmentations from them. Our approach could potentially be used in conjunction with many of these systems as well, to reduce the interaction effort. Compared to propagation methods, the interactive methods usually have the advantage of greater precision, but at the disadvantages of greater human effort and less amenability to crowdsourcing.

Only limited work explores click supervision for image and video annotation. Clicks on objects in images can remove ambiguity to help train a CNN for semantic segmentation from weakly labeled images (Bearman et al. 2015), or to spot object instances in images for dataset collection (Lin et al. 2014). Clicks on patches are used to obtain ground truth material types in (Bell et al. 2015).

We are aware of only two prior efforts in video segmentation using clicks, and their usage is quite different than ours. In one, a click and drag user interaction is used to segment objects (Pont-Tuset, Farré, and Smolic 2015). A small region is first selected with a click, then dragged to traverse up in the hierarchy until the segmentation does not bleed out of the object of interest. Our user-interaction is much simpler (just a few mouse clicks or taps on the touchscreen) and our boundary clicks are discriminative enough to quickly filter good segmentations. In the other, the TouchCut system uses a single touch to segment the object using level-set techniques (Wang, Han, and Collomosse 2014). However, the evaluation is focused on image segmentation, with only limited results on video; we show our approach outperforms it.

## Approach

We now present our approach.

### Generating video foreground proposals

Existing propagation-based video segmentation methods rely on human input (a bounding box, contour, or scribble) at the onset to generate results. The key idea behind our Click Carving approach is to flip this process. Instead of the human annotator providing a foreground region from scratch, the system generates many plausible segmentation mask hypotheses and the annotator efficiently navigates to the best ones with point clicks.

Specifically, we use state-of-the-art region proposal generation algorithms to generate 1000s of possible foreground segmentations for the first video frame.[1] Region proposal methods aim to obtain high recall at the cost of low precision. Even though this guarantees that at least a few of these segmentations will be of good quality, it is difficult to filter out the best ones automatically with existing techniques.

To generate accurate region proposals in videos, we use the multiscale combinatorial grouping (MCG) algorithm (Arbeláez et al. 2014) with both static and motion boundaries. The original algorithm uses image boundaries to obtain a hierarchical segmentation, followed by a grouping procedure to obtain region-based foreground object proposals. The video datasets that we use in this work have both static and moving objects. We observed that due to factors like motion blur etc., static image boundaries are not very reliable in many cases. On the other hand, optical flow provides a strong cue about the object contours while the object is in motion. Hence we also use motion boundaries (Weinzaepfel et al. 2015) to generate per-frame motion region proposals using MCG. The two sources are complementary in nature: for static objects, the per-frame region proposals obtained using static boundaries will be more accurate, and vice versa.

Figure 1 illustrates this with an example. Both the person and bike (Figure 1a) are in motion. As a result, we get weaker static boundaries (Figure 1b). Figure 1c shows the best static proposal for each object; the proposal quality for the bike is very poor. On the other hand, the motion boundaries (Figure 1d) are much stronger and result in very accurate proposals for both the person and the bike (Figure 1e).

In summary, given a video frame, we generate the set of foreground region proposals ($\mathcal{M}$) for it by taking the union between the static region proposals ($\mathcal{M}_{static}$) and motion region proposals ($\mathcal{M}_{motion}$), i.e., $\mathcal{M} = \{\mathcal{M}_{static} \cup \mathcal{M}_{motion}\}$. On average we generate a total of about 2000 proposals per frame, resulting in a very high overall recall. (The Mean Average Best Overlap score (MABO) is 78.3 on the three datasets that we use. This is computed by selecting the proposal with highest overlap score in each frame and taking a dataset-wide average.) In what follows, we explain how Click Carving allows a user to efficiently navigate to the best proposal among these thousands of candidates.

### Click Carving for discovering an object mask

The region proposal step yields a large set of segmentation hypotheses (1000s), out of which only a few are very accurate object segmentations. A naive approach that asks an annotator to manually scan through all proposals is both tedious and inefficient. We now explain how our Click Carving algorithm effectively and very quickly identifies the

[1] For clarity of presentation, we describe the process as always propagating from the first annotated frame. However, the system can be initialized from arbitrary frames.

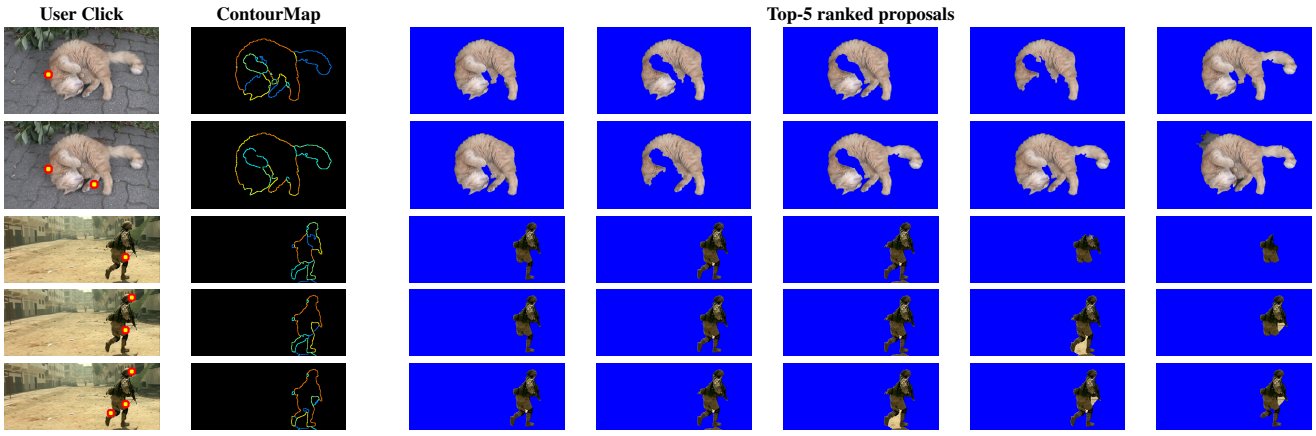



Figure 2: Click Carving based foreground segmentation. Best viewed on pdf and in video on project page. See text for details.

quality segmentations. We show that within a few clicks, it is possible to obtain a very high quality segmentation of the desired object of interest. We stress that while Click Carving assists in getting the mask for a single frame, it is closely tied to the video source due to the motion-based proposals.

At a high level, our Click Carving algorithm converts the user clicks into votes cast for the underlying region proposals. The user initiates the algorithm by clicking somewhere on the boundary of the object of interest. This click casts a vote for all the proposals whose boundaries also (nearly) intersect with the user click. Using these votes, the underlying region proposals are re-ranked and the user is presented with the top-$k$ proposals having the highest votes.

This process of clicking and re-ranking iterates. At any time, the user can choose any of the top-$k$ as the final segmentation if he/she is satisfied, or he/she can continue to re-rank by clicking and casting more votes.

More specifically, we characterize each proposal, $\mathcal{M}_j \in \mathcal{M}$ with the following components ($\mathcal{M}_j^m, \mathcal{M}_j^e, \mathcal{M}_j^s, \mathcal{M}_j^v$):

- Segmentation mask ($\mathcal{M}_j^m$): This quantity represents the actual region segmentation mask obtained from the MCG region proposal algorithm (static or dynamic).
- Contour mask ($\mathcal{M}_j^e$): Our algorithm requires the user to click on the object boundaries, which as we will show later is much more discriminative than clicking on interior points and results in a much faster filtering of good segmentations. To infer the votes on the boundaries, we convert the segmentation mask $\mathcal{M}_j^m$ into a contour mask. This contour mask only contains the boundary pixels from $\mathcal{M}_j^m$. For error tolerance, we dilate the boundary mask by 5 pixels on either side. This reduces the sensitivity of the exact user click location, which need not coincide exactly with the mask boundary.
- Objectness score ($\mathcal{M}_j^s$): We use the objectness score from the MCG algorithm (Arbeláez et al. 2014) to break ties if multiple region proposals get the same number of votes. This score reflects the likelihood of a given region to be an accurate object segmentation.
- User votes ($\mathcal{M}_j^v$): This quantity represents the total number of user votes received by a particular proposal at any given time. It is initialized to 0.

As a first step, we begin by computing a lookup table which allows us to efficiently account for the votes cast for each proposal by the user. Let $n$ be the total number of pixels in a given image and $m$ be the total number of region proposals generated for that image. We define and precompute a lookup table $\mathcal{T} \in \{0,1\}^{n \times m}$ as follows:

$$\mathcal{T}(i,j) = \begin{cases} 1 & \text{if } \mathcal{M}_j^e(i) = 1 \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $i$ denotes a particular pixel and $j$ denotes a particular region proposal.

When the user clicks at a particular pixel location $c$, the weights for each of the region proposal are updated as follows:

$$\mathcal{M}_j^v = \mathcal{M}_j^v + \mathcal{T}(c,j). \quad (2)$$

The updated set of votes is used to re-rank all the region proposals. The proposals with equal votes are ranked in the order of their objectness scores. This interactive re-ranking procedure continues until the user is satisfied with any of the top-$k$ proposals and chooses that as the final segmentation. In our implementation, $k$ is set such that $k$ copies of the image, one proposal on each, fit easily on one screen ($k = 9$).

Figure 2 illustrates our user interface and explains this process with two examples. We show the user interaction on the leftmost column. Red circles denote clicks. The "ContourMap" column shows the average contour map of the top-5 ranked proposals after the user click. Here the colors are a heat-map coding of the number of votes for a boundary fragment. Remaining columns show the top-5 ranked proposals.

The top two rows show a frame from a "cat" video in the iVideoSeg (Shankar Nagaraja, Schmidt, and Brox 2015) dataset. The user[2] places the first click on the left side of the object (top left image). We see that the resulting top ranked proposals (5 foreground images in top row) align well to the current user click, meaning they all contain a boundary

[2]We discuss our user study in the next section.

near the click point. The average contour map of these top ranked proposals informs the user about areas that have been carved well already (red lines) and which areas may need more attention (blue lines, or contours on the true object that remain uncolored). The user observes that most current top-$k$ segmentations are missing the cat's right leg and decides to click there next (second row, leftmost image).

In the next example, we take a frame from the "soldier" video in the Segtrack-v2 dataset (Li et al. 2013). The user decides to place a click on the right side of the object (third row, leftmost image). This click itself retrieves a very good segmentation for the soldier. However, to explore further, the user continues by making more clicks. Each new constraint eliminates the bad proposals from the previous step, and after just 3 clicks, all the top-ranked proposals are of good quality. See video on project webpage for more examples.

## User clicking strategies

To quantitatively evaluate Click Carving, we employ both real human annotators and simulated users with different clicking strategies. We design a series of clicking strategies to simulate, each of which represents a hypothesis for how a user might efficiently convey which object boundaries remain missing in the top proposals. While real users are arguably the best way to judge final impact of our system (and so we use them), the simulated user models are complementary. They allow us to run extensive trials and to see at scale which strategies are most effective. Simulated human users have also been studied in interactive segmentation for brush stroke placement (Kohli et al. 2012).

We categorize the user models into three groups: human annotators, boundary clickers, and interior clickers.

**Human annotators:** We conduct a user study to analyze the performance of our method by recruiting 3 human annotators to work on each image. The 3 annotators included a computer vision student and 2 non-expert users. The human annotators were encouraged to click on object boundaries, while observing the current best segmentations. They were also given some time to familiarize themselves with the interface, before starting the actual experiments. They had a choice to stop by choosing one of the segmentations among the top ranked ones or continue clicking to explore further. A maximum budget of 10 clicks was used to limit the total annotation time, after which the annotation process stops and a final object mask selection has to be made. The target object was indicated to them before starting the experiment. In the case of multiple objects, each object was chosen as the target object in a sequential manner. We recorded the number of clicks, time spent, and the best object mask chosen by the user during each segmentation. The user corresponding to the median number of clicks is used for our quantitative evaluation. The total recorded time includes the time to both place the clicks and to select the best segmentation mask.

**Boundary clickers:** We design three simulated users which operate by clicking on object boundaries. To simulate these artificial users, we make use of the ground-truth segmentation mask of the target object. Equidistant points are sampled from the ground truth object contour to define object boundaries. Each simulated boundary clicker starts from the same initial point. We use principal component analysis (PCA) on the ground truth shape to find the axis of maximum shape variation. We consider a ray from the centroid of the object mask along the direction of this principal axis. The furthest point on the object boundary where this ray intersects is chosen as the starting point. The three boundary clickers that we design differ in how they make subsequent clicks from this starting point. They are:

(a) ***Uniform clicker:*** To obtain uniformly spaced clicks, we divide the total number of boundary points by the maximum click budget to obtain a fixed distance interval $d$. Starting from the initial point and walking along the boundary, a click is made every $d$ points apart from the previous click location.

(b) ***Submod clicker:*** The uniform user has a high level of redundancy, since it clicks at locations which are still close to the previous clicks; hence the gain in information between two consecutive clicks might be small. Next we design a boundary clicker that tries to impact the maximum boundary region with each subsequent click. This is done by placing the click at a boundary point which is furthest away from its nearest user click among all boundary points. This resembles the sub-modular subset selection problem (Krause and Guestrin 2007), where one tries to maximize the set coverage while choosing a subset. We employ a greedy algorithm to find the next best point.

(c) ***Active clicker:*** The previous two methods only looked at the ground truth segmentation to devise a click strategy, without taking into account the segmentation performance after each click is added. Our active clicking strategy takes into account the current best segmentation among the top-$k$ (vs. the ground truth) and uses that to make the next click decision. It is similar in design to the Submod user, except that it skips those boundary points which have already been labeled correctly by the top-ranked proposal. We find that this active simulated user comes the closest in mimicking the actual human annotators (see results for details).

**Interior clickers:** A novel insight of our method is the discriminative nature of boundary clicks. In contrast, default behavior and previous user models (Wang, Han, and Collomosse 2014; Bearman et al. 2015) assume a click in the interior of the object is well-suited. To examine this contrast empirically, our final simulated user clicks on interior object points. To simulate interior clicks, we uniformly sample object pixel locations from the entire ground truth segmentation mask (up to the maximum click budget) and then sequentially place clicks on the object of interest.

We analyze the impact of the click strategies in our results section.

## Propagating the mask through the video

Having discovered a good object mask using Click Carving in the initial frame, the next step is to propagate this segmentation to all other frames in the video. We use the foreground propagation method of (Jain and Grauman 2014) as our segmentation method primarily due to good performance and

efficiency. We also tried other methods like (Wen et al. 2015) but found (Jain and Grauman 2014) to be most scalable for large experiments. In its original form the method requires a human drawn object outline in the initial frame. We instead initialize the method using the region proposal which was selected using Click Carving. This initial mask is then propagated to the entire video to obtain the final segmentation. We compute the supervoxels required by (Jain and Grauman 2014) using (Grundmann et al. 2010) and use the default parameter settings.

# Results

## Datasets and metrics

We evaluate on 3 publicly available datasets: Segtrack-v2 (Li et al. 2013), VSB100 (Sundberg et al. 2011; Galasso et al. 2013) and iVideoSeg (Shankar Nagaraja, Schmidt, and Brox 2015). For evaluating segmentation accuracy we use the standard intersection-over-union (IoU) overlap metric between the predicted and ground-truth segmentations. A brief overview of the datasets:

- **SegTrack v2**: This dataset is the most common benchmark to evaluate video object segmentation. It consists of 14 videos with a total of 24 objects and 976 frames. Challenges include appearance changes, large deformation, motion blur etc. Pixel-wise ground truth (GT) masks are provided for every object in all frames.
- **Berkeley Video Segmentation Benchmark (VSB100)**: This dataset consists of 100 HD sequences with multiple objects in each video. We use the "train" subset of this dataset in our experiments, for a total of 39 videos and 4397 frames. This is a very challenging dataset; interacting objects and small object sizes make it difficult to segment and propagate. We use the GT annotations of multiple foreground objects provided by (Pont-Tuset, Farré, and Smolic 2015) on every 20th frame.
- **iVideoSeg**: This new dataset consists of 24 videos from 4 different categories (car, chair, cat, dog). Some videos have viewpoint changes and others have large object motions. GT masks are available for 137 of all 11,882 frames.

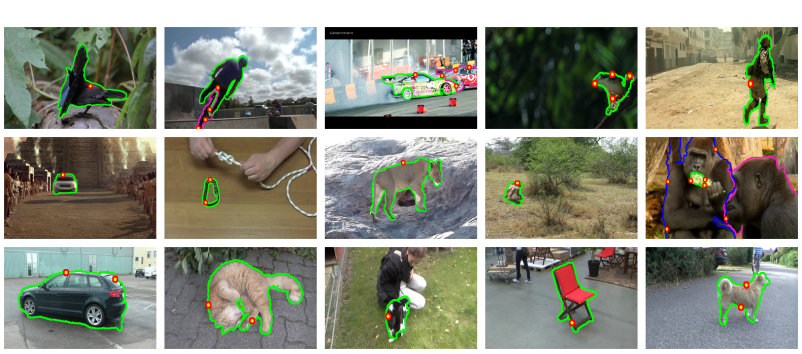
## Methods for comparison

We compare with several state-of-the art methods and our own baselines. Below we group them into 6 groups based on the amount of human annotation effort, i.e., the interaction time between the human and algorithm. In some cases, a human simply initializes the algorithm, while in others the human is constantly in the loop.

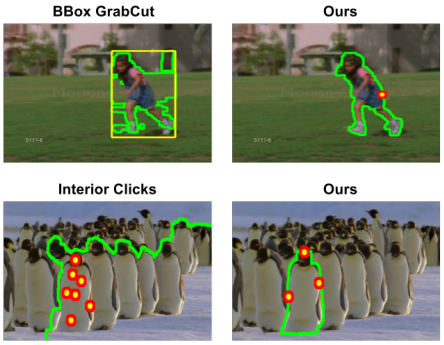
- **Unsupervised:** We use the state-of-the-art method of (Papazoglou and Ferrari 2013), which produces a single region segmentation per video with no human involvement.
- **Multiple segmentation:** Most existing unsupervised methods produce multiple segmentations to achieve high recall. We consider both **1) Static object proposals (BestStaticProp):** where the best per frame region proposal (out of approx 2000 proposals per frame) is chosen as the final segmentation for that frame **2) Spatio-temporal proposals** (Li et al. 2013; Lee, Kim, and Grauman 2011; Grundmann et al. 2010): These methods produce multiple spatio-temporal region tracks as segmentation hypotheses. To simulate a human picking the desired segmentation from the hypotheses, we use the dataset ground truth to select the most overlapping hypothesis. We use the duration of the video to estimate interaction time. This is a lower bound on cost, since the annotator has to at least watch the clip once to select the best segmentation. For the static proposals, we multiply the number of frames by 2.4 seconds, the time required to provide one click (Bearman et al. 2015).
- **Scribble-based:** We consider two existing methods: **1) JOTS** (Wen et al. 2015): the first frame is interactively segmented using scribbles and GrabCut. The segmentation result is than propagated to the entire video. We use the timing data from the detailed study by (McGuinness and OConnor 2010), who find it takes a human on average 66.43 seconds per image to obtain a good segmentation with scribbles. **2) iVideoSeg** (Shankar Nagaraja, Schmidt, and Brox 2015): This is a recently proposed state-of-the-art technique that uses scribbles to interactively label point trajectories. These labels are then used to segment the object of interest. We use the timing data kindly shared by the authors.
- **Object outline propagation:** the user outlines the object completely to initialize the algorithm (typically in the first frame), which then propagates to the entire video. Here we use the same method for propagation (Jain and Grauman 2014) as in our approach. Timing data from (Jain and Grauman 2013; Lin et al. 2014) indicate it typically takes 54-79 seconds to manually outline an object; we use the more optimistic 54 seconds for this baseline.
- **Bounding box:** Rather than segment the object, the annotator draws a tight bounding box around it. The baseline **BBox-VidGrabCut** uses that box to obtain a segmentation for the video as follows. We learn a Gaussian Mixture Model (GMM) based appearance model for foreground and background pixels using the box, then apply them in a standard spatio-temporal MRF defined over pixels. The unaries come from the learnt GMM model and contrast-sensitive spatial and temporal potentials are used for smoothness.
- **1-Click based:** We also consider baselines which perform video segmentation with a single click. **1) TouchCut** (Wang, Han, and Collomosse 2014) the only prior work using clicks for video segmentation. **2) Click-VidGrabCut:** This is similar to BBox-VidGrabCut except that we take a small region around the click to learn the foreground model. The background model is learnt from a small area around image boundaries. **3) Click-STProp:** To propagate the impact of a user click to the entire video volume, we use the spatio-temporal proposals from (Oneata et al. 2014). We do this by selecting all proposals which enclose the click inside them. Fg and bg appearance models are learnt using the selected proposals and refined using a spatio-temporal MRF. We again use the timing data from (Bearman et al. 2015), which reports

| | | Objectness | Interior | BBox-GrabCut | BBox-Prop | Uniform | Submod | Active | Human | BestProp |
|---|---|---|---|---|---|---|---|---|---|---|
| **Segtrack-v2** | Clicks | 0 | 6.29 | 2 | 2 | 4.46 | 3.83 | 3.34 | **2.46** | - |
| | Time (sec) | 0 | 15.09 | 7 | 7 | 16.98 | 14.58 | 12.72 | **9.37** | - |
| | IoU | 42.36 | 52.79 | 59.55 | 67.51 | 75.8 | 76.76 | 76.24 | **78.77** | 80.74 |
| **VSB100** | Clicks | 0 | 7.05 | 2 | 2 | 5.34 | 5.28 | 5.23 | **4.35** | - |
| | Time (sec) | 0 | **16.92** | 7 | 7 | 22.81 | 22.55 | 22.33 | 18.58 | - |
| | IoU | 28.45 | 46.98 | 57.81 | 58.98 | 64.2 | 65.67 | 66.91 | **69.63** | 72.82 |
| **iVideoSeg** | Clicks | 0 | 5.02 | 2 | 2 | 3.84 | 3.29 | 3.15 | **2.84** | - |
| | Time (sec) | 0 | 12.05 | 7 | 7 | 15.20 | 13.02 | 12.47 | **11.24** | - |
| | IoU | 50.69 | 72.54 | 65.43 | 68.04 | 77.57 | 77.84 | **78.65** | 78.24 | 81.34 |

Table 1: Click-carving proposal selection quality for real users (Human), the different user click models (Interior, Uniform, Submod, Active), Objectness, and BBox baselines. With an average of 2-4 clicks to carve the proposal boundaries, users attain IoU accuracies very close to the upper bound (BestProp). Objectness, Interior clicks, and the BBox baselines are substantially weaker. IoU measures segmentation overlap with the ground truth; perfect overlap is 100.



Visual results for Click-Carving




Visual comparisons with baselines

Figure 3: **Left:** Qualitative results for Click Carving. The yellow-red dots show the clicks made by human annotators. The best selected segmentation boundaries are overlayed on the image (green). **Right:** Comparisons with baselines: The top example shows the segmentation we obtain with a single click as opposed to applying GrabCut segmentation with a tight bounding box. Bottom example shows the discriminative power of clicking on boundaries by comparing it with a baseline which clicks in the interior regions. Best viewed on pdf.

that a human takes about 2.4 seconds to place a single click on the object of interest.

## Experiments

We first test the accuracy/speed tradeoff in terms of locating the best available proposal, and compare the simulated user models. Then we present comparisons against all the existing methods and baselines.

**Click Carving for region proposal selection:** We first present the performance of Click Carving for interactively locating the best region proposal for the object of interest. We do this for the first frame in all videos. In all experiments, we set the total click budget to be a maximum of 10 clicks per object. For simulated users, clicks are placed sequentially depending on its design, until a proposal which is within 5% overlap of the best proposal is ranked in the top-$k$ or the click budget is exhausted. For the human user study, the user stops when they decide that they found a good segmentation within the top-$k$ ranked proposals or have exhausted the click budget.

Table 1 shows the results for all datasets and compares the performance with all simulated users. We compare both in terms of the number of clicks and time required[3] and also how close they get to the best proposal available in the pool of $\sim$2000 (BestProp). As expected, in all cases real users achieve the best segmentation performance and require far fewer clicks than all simulated users to achieve it. Our simulated Active user, which takes into account the current state of the segmentation, comes closest to matching the human's performance. Also, we see clicking uniformly on the object boundaries requires more clicks on average than the Active and Submodular users, which try to influence the largest object area with each subsequent click. The Objectness baseline, which first ranks all the proposals using objectness scores and picks the best proposal among top-$k$ ($k$=9), performs the worst. This shows that user interaction is key to picking good quality proposals among 1000s of candidates.

All users that operate by clicking on boundaries (Human, Uniform, Submod, and Active), come very close to choosing the best proposal in most cases. In contrast, clicking on the interior points requires substantially more clicks—often

[3]We use the average time per click from our human studies as an estimate for simulated boundary clickers. For interior clicks we use 2.4 seconds per click (Bearman et al. 2015).

| | Unsup. | Multiple Segmentations | | | Scribbles | Outline | Bounding Box | Click Based | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | (Papazoglou et al. 2013) | (Lee et al. 2011) | (Li et al. 2013) | BestStatic Prop | (Wen et al. 2015) | (Jain et al. 2014) | BBox - VidGrabCut | Click - VidGrabCut | Click-STProp | Ours |
| **Avg. Accuracy** | 35.24 | 45.26 | 65.92 | 78.48 | 71.91 | 67.86 | 23.04 | 16.81 | 46.18 | 63.65 |
| **Annot. Effort** | - | 10.6 tracks | 60 tracks | 120k proposals | 1 frame | 1 frame | 2 clicks | 1 click | 1 click | 2.46 clicks |
| **Annot. Time (sec)** | 0 | 21.2 | 120 | 142.5 | 66.43 | 54 | 7 | 2.4 | 2.4 | 9.37 |

Table 2: Video segmentation accuracy (IoU) on all 14 videos from Segtrack-v2. See project webpage for per-video results. The last column shows our result with real human users. The bottom two rows summarize the amount of human annotation effort required to obtain the corresponding segmentation performance, for all methods. Our approach leads to an excellent trade-off between video segmentation accuracy and human annotation effort.

| | Unsup. | Outline | Bounding Box | Click Based | | |
|---|---|---|---|---|---|---|
| | (Papazoglou et al. 2013) | (Jain et al. 2014) | BBox-VidGrabCut | Click-VidGrabCut | Click-STProp | Ours |
| **Avg. Accuracy** | 17.79 | 61.43 | 14.74 | 11.14 | 26.76 | 56.15 |
| **Annot. Effort** | - | 1 frame | 2 clicks | 1 click | 1 click | 4.35 clicks |
| **Annot. Time (sec)** | 0 | 54 | 7 | 2.4 | 2.4 | 18.58 |

Table 3: Video segmentation accuracy (IoU) on all 39 videos in VSB100; format as in Table 2. Our approach provides an excellent trade-off between video segmentation accuracy and human annotation effort.

double the number. More importantly, the best segmentation it obtains is much worse in quality than the best possible segmentation. This makes the use of interior clicks impractical here even after accounting for the fact that they may be faster to provide than boundary clicks. This supports our hypothesis that clicking on boundaries is much more discriminative in separating good proposals from the bad ones. Whereas a matching between an object proposal contour and a boundary click will rarely be accidental, several bad proposals may have the interior click point lie within them.

In fact, selecting the best proposal using an enclosing bounding box around the true object (BBox-Prop, Table 1) is more effective than clicking on interior points. This is likely because a tight bounding box can eliminate a large number of proposals that extend outside its boundaries. On the other hand, an interior click cannot restrict the selected proposals to the ones which align well to the object boundaries. Our method outperforms the bounding box selection by a large margin, showing the efficacy of our approach. Our approach also significantly outperforms the standard GrabCut interactive image segmentation method, initialized with a tight bounding box around the object (BBox-GrabCut, Table 1).

On Segtrack-v2 and iVideoSeg, Click Carving requires less than 3 clicks on average to obtain a high quality segmentation. For the most challenging dataset, VSB100, we obtain good results with an average of 4.35 clicks. This shows the potential of our method to collect large amounts of segmentation data economically. The timing data reveals the efficiency and scalability of our method. Below we show how this translates to advantages for full video segmentation.

Figure 3 (left) shows qualitative results for Click Carving. In many cases (e.g., lions, soldier, cat), only a single click is sufficient to obtain a high quality segmentation. Several challenging instances like the cat (bottom row) and the lion (middle row), are segmented very accurately with a single click. These objects would otherwise require a large amount of human interaction to obtain good segmentation (say using a GrabCut like approach). More clicks are typically needed when multiple objects are close-by or interacting with each other. Still, we observe that in many cases only a small number of clicks on each object results in good segmentations. For example, in the car video (top row), only 5 clicks are required to obtain final segmentations for both objects.

Figure 3 (right) highlights the key strengths of our method over two baselines. In the top example, we see that GrabCut segmentation applied even with a very tight bounding box fails to segment the object. On the other hand, even with a single click, our proposed approach produces a very accurate segmentation. The example on the bottom shows the importance of clicking on boundaries. Clicking on the interior fails to retrieve a good proposal because several bad proposals also contain those interior clicks. Our boundary clicks, which are highly discriminative, retrieve the best proposal quickly.

Next we discuss the results for video segmentation, where we propagate the results of Click Carving to the remaining frames in the video.

**Video segmentation propagation on Segtrack-v2:** Table 2 shows the results on Segtrack-v2. We compare using the standard intersection-over-union (IoU) metric with a total of 10 methods which use varying amounts of human supervision. The unsupervised algorithm (Papazoglou and Ferrari 2013) that uses no human input results in the lowest accuracy. Among the approaches which produce multiple segmentations, BestStaticProp and (Li et al. 2013) have the best accuracy. This is expected because these methods are designed for having high recall, but it requires much more effort to sift through the multiple hypotheses to pick the best one. For example, it is prohibitively expensive to go through 2000 segmentations for each frame to get to the accuracy level of BestStaticProp. The method of (Li et al. 2013) produces much fewer segmentations, but still requires 12x more time than our method to achieve comparable performance.

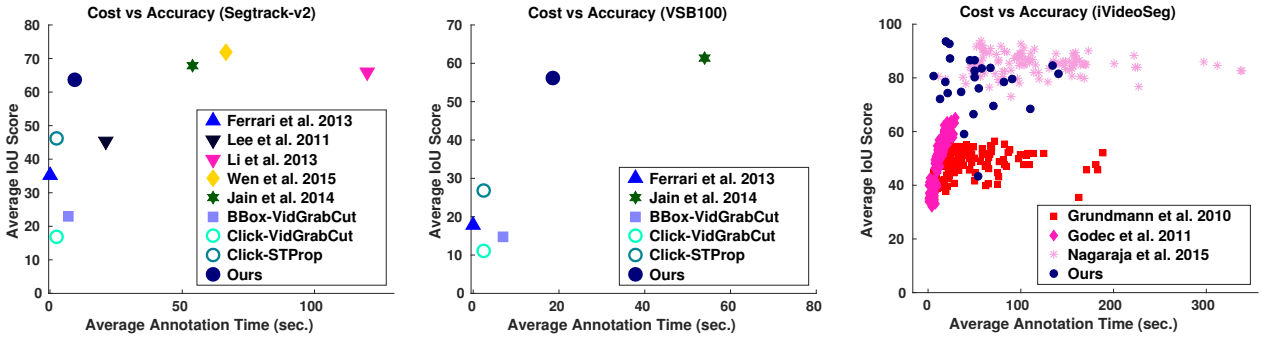



Figure 4: Cost vs accuracy on Segtrack (left), VSB100 (center), and iVideoSeg (right). Our Click Carving based video propagation results in similar accuracy as state-of-the-art metods, but it does so with much less human effort. Click Carving offers an excellent trade-off between cost and accuracy. Best viewed on pdf.

The scribble based method (Wen et al. 2015) achieves the best overall accuracy on this dataset, but is 6 times more expensive than our method. The propagation method of (Jain and Grauman 2014), which we also use as our propagation engine, sees an increase of 4% in accuracy when propagated from human-labeled object outlines. On the other hand, our method which is initialized from slightly imperfect—but much quicker to obtain—object boundaries achieves comparable performance. Using computer generated segmentations coupled with our Click Carving interactive selection algorithm is sufficient to obtain high performance.

Moving on to the methods that require less human supervision, i.e., bounding boxes and clicks, we see that Click Carving continues to hold advantages. In particular, BBox-VidGrabCut and Click-VidGrabCut result in poor performance, indicating that more nuanced propagation methods are needed than just relying on appearance-based segmentation alone. Click-STProp, which obtains a spatial prior by propagating the impact of a single click to the entire video volume, results in much better performance than solely appearance based methods. However, our method, which first translates clicks into accurate per-frame segmentation before propagating them, yields a 17% gain (37% relative gain).

All these trends show that our method offers an excellent trade-off between segmentation performance and annotation time. Figure 4 (left), visually depicts this trade-off. All methods which result in better segmentation accuracy than ours need substantially more human effort. Even then the gap in the performance in relatively small. On the flip side, the methods which require less annotation effort than us also result in a significant degradation in segmentation performance.

**Video segmentation propagation on VSB100:** Next, we test on VSB100. This is an even more challenging dataset and very few existing methods have reported foreground propagation results on it. Since this dataset includes several videos that contain multiple interacting objects in challenging conditions, Click Carving tends to require more clicks (4.35 on average). Our method again outperforms all baselines which require less human effort and results in comparable performance with (Jain and Grauman 2014), but at a much lower cost. Figure 4 (center) again reflects this trend.

**Video segmentation propagation on iVideoSeg:** We also compare our method on the recently proposed iVideoSeg dataset (Shankar Nagaraja, Schmidt, and Brox 2015). We compare with 3 methods (Grundmann et al. 2010; Godec, Roth, and Bischof 2011; Shankar Nagaraja, Schmidt, and Brox 2015) out of which (Shankar Nagaraja, Schmidt, and Brox 2015) is the current state-of-the-art method for interactive foreground segmentation in videos. Since the videos in the dataset are much longer (between 300-1000 frames), we use our Click Carving method to reinitialize the segmentation propagation every 100 frames. Figure 4 (right) shows the results. We outperform both (Grundmann et al. 2010; Godec, Roth, and Bischof 2011) by a considerable margin. When compared with (Shankar Nagaraja, Schmidt, and Brox 2015), our method achieves similar segmentation accuracy but with less than half the total annotation time. On average over all 24 videos, (Shankar Nagaraja, Schmidt, and Brox 2015) takes 110.05 seconds to achieve an IoU score of 80.04. In comparison our method only takes 54.35 seconds to reach an IoU score of 77.68.

**Comparison with TouchCut:** To our knowledge TouchCut (Wang, Han, and Collomosse 2014) is the only prior work which utilizes clicks for video segmentation. In that work, the user places a click somewhere on the object, then a level-sets technique transforms the click to an object contour. This transformed contour is then propagated to the remaining frames. Very few experimental results about video segmentation are discussed in the paper, and code is not available. Therefore, we are only able to compare with TouchCut on the 3 Segtrack videos reported in their paper. Table 4 shows the result. When initialized with a single click, our method outperforms TouchCut in 2 out of 3 videos. With 1 more click, we perform better in all 3 videos.

**Qualitative results on video segmentation propagation:** Figure 5 shows some qualitative results. The leftmost image in each row shows the best region proposal chosen by a human annotator using Click Carving. Subsequent images show the results of segmentation propagation, when initialized from the selected proposal.

| | TouchCut | Ours (1-click) | Ours (2-clicks) |
|---|---|---|---|
| birdfall2 | 248 | 213 | 187 |
| girl | 1691 | 2213 | 1541 |
| parachute | 228 | 225 | 198 |

Table 4: Comparison with TouchCut (Wang, Han, and Collomosse 2014) in terms of pixel error (lower is better).

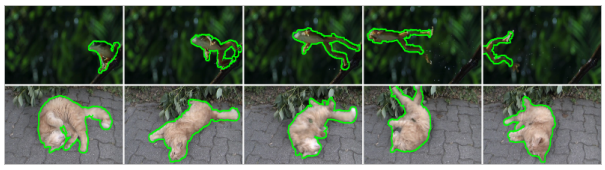


Figure 5: Qualitative results for video segmentation. Best viewed on pdf. See project webpage for videos.

**Conclusion:** We presented a novel interactive video segmentation technique *Click Carving* which can accurately segment objects in videos with only a few clicks. It strikes an excellent balance between accuracy and human effort resulting in large savings. Its ease of use even for non-experts offers great promise for scaling up video segmentation which can be beneficial for several research communities.

**Acknowledgments:** This research is supported in part by ONR YIP N00014-12-1-0754. We thank Shankar Nagaraja for providing the iVideoseg dataset timing data. We also thank all the participants in our user studies.

## References

Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; and Malik, J. 2014. Multiscale combinatorial grouping. In *CVPR*.

Badrinarayanan, V.; Galasso, F.; and Cipolla, R. 2010. Label propagation in video sequences. In *CVPR*.

Bai, X.; Wang, J.; Simons, D.; and Sapiro, G. 2009. Video snapcut: Robust video object cutout using localized classifiers. In *SIGGRAPH*.

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2015. What's the point: Semantic segmentation with point supervision. *ArXiv e-prints*.

Bell, S.; Upchurch, P.; Snavely, N.; and Bala, K. 2015. Material recognition in the wild with the materials in context database. *Computer Vision and Pattern Recognition (CVPR)*.

Brox, T., and Malik, J. 2010. Object Segmentation by Long Term Analysis of Point Trajectories. In *ECCV*.

Carreira, J., and Sminchisescu, C. 2012. CPMC: Automatic object segmentation using constrained parametric min-cuts. *PAMI* 34(7):1312–1328.

Fathi, A.; Balcan, M.; Ren, X.; and Rehg, J. 2011. Combining self training and active learning for video segmentation. In *BMVC*.

Galasso, F.; Nagaraja, N.; Cardenas, T.; Brox, T.; and B.Schiele. 2013. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*.

Godec, M.; Roth, P. M.; and Bischof, H. 2011. Hough-based tracking of non-rigid objects. In *ICCV*.

Grundmann, M.; Kwatra, V.; Han, M.; and Essa, I. 2010. Efficient hierarchical graph based video segmentation. In *CVPR*.

Jain, S., and Grauman, K. 2013. Predicting sufficent annotation strength for interactive foreground segmentation. In *ICCV*.

Jain, S. D., and Grauman, K. 2014. Supervoxel-consistent foreground propagation in video. In *ECCV 2014*. 656–671.

Kohli, P.; Nickisch, H.; Rother, C.; and Rhemann, C. 2012. User-centric learning and evaluation of interactive segmentation systems. *IJCV* 100(3):261–274.

Krause, A., and Guestrin, C. 2007. Near-optimal observation selection using submodular functions. In *National Conference on Artificial Intelligence (AAAI), Nectar track*.

Lee, Y. J.; Kim, J.; and Grauman, K. 2011. Key-segments for video object segmentation. In *ICCV*.

Lezama, J.; Alahari, K.; Sivic, J.; and Laptev, I. 2011. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*.

Li, F.; Kim, T.; Humayun, A.; Tsai, D.; and Rehg, J. M. 2013. Video Segmentation by Tracking Many Figure-Ground Segments. In *ICCV*.

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

Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common objects in context. In *ECCV*.

McGuinness, K., and OConnor, N. E. 2010. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition* 43(2):434 – 444. Interactive Imaging and Vision.

Oneata, D.; Revaud, J.; Verbeek, J.; and Schmid, C. 2014. Spatio-temporal object detection proposals. In *ECCV*.

Papazoglou, A., and Ferrari, V. 2013. Fast object segmentation in unconstrained video. In *ICCV*.

Pont-Tuset, J.; Farré, M.; and Smolic, A. 2015. Semi-automatic video object segmentation by advanced manipulation of segmentation hierarchies. In *International Workshop on Content-Based Multimedia Indexing (CBMI)*.

Ren, X., and Malik, J. 2007. Tracking as repeated figure/ground segmentation. In *CVPR*.

Shankar Nagaraja, N.; Schmidt, F. R.; and Brox, T. 2015. Video segmentation with just a few strokes. In *ICCV*.

Sundberg, P.; Brox, T.; Maire, M.; Arbelaez, P.; and Malik, J. 2011. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*.

Tsai, D.; Flagg, M.; and Rehg, J. 2010. Motion coherent tracking with multi-label mrf optimization. In *BMVC*.

Vijayanarasimhan, S., and Grauman, K. 2012. Active frame selection for label propagation in videos. In *ECCV*.

Vondrick, C., and Ramanan, D. 2011. Video annotation and tracking with active learning. In *NIPS*.

Wang, J.; Bhat, P.; Colburn, A.; Agrawala, M.; and Cohen, M. F. 2005. Interactive video cutout. *ACM Trans. Graph.* 24(3):585–594.

Wang, T.; Han, B.; and Collomosse, J. 2014. Touchcut: Fast image and video segmentation using single-touch interaction. *Computer Vision and Image Understanding* 120:14 – 30.

Weinzaepfel, P.; Revaud, J.; Harchaoui, Z.; and Schmid, C. 2015. Learning to Detect Motion Boundaries. In *CVPR 2015*.

Wen, L.; Du, D.; Lei, Z.; Li, S. Z.; and Yang, M.-H. 2015. Jots: Joint online tracking and segmentation. In *CVPR*.

Xu, C.; Xiong, C.; and Corso, J. J. 2012. Streaming Hierarchical Video Segmentation. In *ECCV*.