Tracking-assisted Weakly Supervised Online Visual Object Segmentation in Unconstrained Videos

Zongpu Zhang[1], Yang Hua[2], Tao Song[1,*]
Zhengui Xue[3,1], Ruhui Ma[1], Neil Robertson[2], Haibing Guan[1]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]Queen's University Belfast, Belfast, UK
[3]Ulster University, Belfast, UK
{zhangz-z-p, songt333, zhenguixue, ruhuima, hbguan}@sjtu.edu.cn, {Y.Hua, N.Robertson}@qub.ac.uk

ABSTRACT

This paper tackles the task of online video object segmentation with weak supervision, i.e., labeling the target object and background with pixel-level accuracy in unconstrained videos, given only one bounding box information in the first frame. We present a novel tracking-assisted visual object segmentation framework to achieve this. On the one hand, initialized with a given bounding box in the first frame, the auxiliary object tracking module guides the segmentation module frame by frame by providing motion and region information, which is usually missing in semi-supervised methods. Moreover, compared with the unsupervised approach, our approach with such minimum supervision can focus on the target object without bringing unrelated objects into the final results. On the other hand, the video object segmentation module also improves the robustness of the visual object tracking module by pixel-level localization and objectness information. Thus, segmentation and tracking in our framework can mutually help each other in an online manner. To verify the generality and effectiveness of the proposed framework, we evaluate our weakly supervised method on two cross-domain datasets, i.e., the DAVIS and VOT2016 datasets, with the same configuration and parameter setting. Experimental results show the top performance of our method, which is even better than the leading semi-supervised methods. Furthermore, we conduct the extensive ablation study on our approach to investigate the influence of each component and main parameters.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**;

KEYWORDS

Video Object Segmentation; Visual Object Tracking; Video Analysis; Deep Learning

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '18, October 22–26, 2018, Seoul, Republic of Korea
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240638

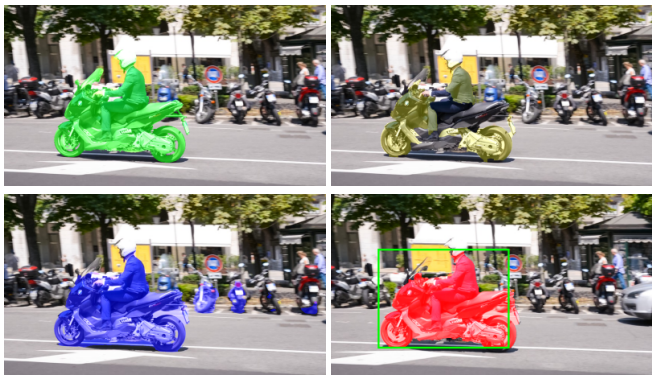

Figure 1: Sample results on the DAVIS dataset. Top-left: Ground truth; Top-right: OSVOS [5] (semi-supervised method); Bottom-left: LVO [39] (unsupervised method); Bottom-right: Ours (weakly supervised method), overlaid on the video frame. *Best viewed in color.*

ACM Reference Format: Zongpu Zhang, Yang Hua, Tao Song, and Zhengui Xue, Ruhui Ma, Neil Robertson, Haibing Guan. 2018. Tracking-assisted Weakly Supervised Online Visual Object Segmentation in Unconstrained Videos. In 2018 ACM Multimedia Conference (MM'18), October 22-26, 2018, Seoul, Republic of Korea. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240638

1 INTRODUCTION

Video object segmentation is the process of extracting a target object from the background with pixel-level accuracy in video data. It has been successfully applied to many multimedia applications, such as content-based video coding [25, 45] and video editing [23], and other real-world scenarios including video surveillance [37], autonomous driving [7, 16]. However, due to the extensive user scenarios, high accuracy object segmentation technique with minimum human interaction in unconstrained videos (e.g., 4K movies or low-quality surveillance data) is still heavily desired.

Generally speaking, video object segmentation techniques can be classified into three groups: supervised, semi-supervised and unsupervised methods. Supervised video segmentation methods [4, 15] usually require continuous user inputs, i.e., interactive annotation, during the segmentation procedure. These methods can output fine results eventually, however, they also bring tedious workload to users. In contrast, unsupervised video segmentation methods [13, 24, 30, 38, 39], also known as automatic methods, do

not require any annotation information but only rely on the information of intrinsic cues, such as motion, saliency and objectness. Early methods of unsupervised video segmentation [30] only utilize motion information (i.e., optical flow) between pairs of frames. They assume that the object (foreground) motion is dissimilar from the surroundings (background). Therefore, they are susceptible to the motion errors, and they also cannot *identify* the object if it has similar motion with the background. Recently, the two-stream model [13, 39] combining the information of motion and appearance becomes popular in unsupervised video segmentation. Although it has achieved promising results on several public datasets, this approach still suffers from unrelated objects included in the final results, as shown in the bottom-left of Figure 1. Furthermore, though tremendous progress has been made in optical flow in both accuracy and speed, e.g., FlowNet 2.0[21], it still performs unstable in low-quality videos, which holds back the real-world applications of this approach.

Semi-supervised methods [5, 8, 22, 31] try to balance between supervised methods and unsupervised methods. On the one hand, a semi-supervised method significantly reduces the requirement of annotation. It only needs a full object mask in the first frame of the whole video sequence. On the other hand, with the information of one-frame mask, a semi-supervised method can focus on the target object without bringing in unwanted objects in the segmentation results, thus it solves the issue mentioned above in unsupervised methods. However, as illustrated in the top-right of Figure 1, the results of semi-supervised methods tend to degenerate into small pieces due to the lack of continuously updated guidance information used in supervised methods. Moreover, it is still burdensome for users to prepare full object mask in the first frame.

In this paper, we propose a novel framework by combining the object segmentation module with a general object tracking module. On one side, the general object tracking module supplies continuous guidance for the segmentation module. It can provide certain motion information without calculating optical flow and region information avoiding the degenerating issue in the semi-supervised approach. All these benefits only require one bounding box as input in the first frame. In this way, the annotation burden is dramatically reduced compared with full object mask needed by the semi-supervised approach. On the other side, the output of the object segmentation module can also improve the robustness of the general object tracking module. It overcomes common issues that cause to drift, such as heavy motion blur and abrupt motion. In short, the tasks of segmentation and tracking in our framework can mutually improve each other in an online manner.

The contributions of this paper are two-fold. Firstly, we propose a weakly supervised visual object segmentation framework in unconstrained videos supported by a general tracking module that only requires one bounding box as input in the first frame. Secondly, we present state-of-the-art results on different domain datasets including the DAVIS dataset [32] from video segmentation domain and the VOT2016 dataset [26] from visual object tracking domain (see §4.4). In addition, we also provide an extensive ablation study to show the impact and influence of the components and parameters in our framework (see §4.3). The code and pre-trained models are publicly available at https://github.com/Maphist0/TWS-VOS.

2 RELATED WORK

Semi-supervised video object segmentation. Semi-supervised video object segmentation assumes that the full object mask is given in the first frame. Following up this setting, most of the existing semi-supervised methods focus on propagating the initial object mask into the following frames, using temporal superpixels [6], video seams [3], co-clustering [41], or optical flow [8, 40]. Recently, two CNN-based semi-supervised approaches, named OSVOS [5] and MSK [31], have achieved state-of-the-art results on the DAVIS dataset. Both of these methods pre-trained their networks with extra image data and fine-tuned them in the first frame. MSK further utilizes optical flow to provide complementary motion information.

Unsupervised video object segmentation. Unsupervised video object segmentation methods directly process the video without any human supervision. In the early stage, there were two major methods based on supervoxel [17, 44] and motion boundary [30], respectively. In recent years, video object segmentation with two-steam fashion has become popular. FSEG [13] proposed an end-to-end two-stream deep learning framework to combine appearance and motion information. Later, LVO [39] adopted this two-stream framework and built a novel memory module based on ConvGRU, which represents all the video frames jointly. Though these two-steam unsupervised methods can achieve impressive performance on popular video segmentation dataset without any human annotation, it is liable to bring in unexpected objects into the final results. Different from semi-supervised and unsupervised approaches, our framework using minimum supervision information can target on the correct object, thus overcomes the intrinsic problem of the unsupervised approach and could be easily adopted to wide applications with less human efforts. Furthermore, by replacing optical flow with a general object tracking module, our segmentation framework also has overall guidance with motion information and is more stable in real-world, specifically low-resolution, video sequences.

Weakly-supervised image segmentation. Weakly-supervised image segmentation methods produce masks of objects given the bounding boxes. In recent years, iterative methods are commonly used in the process [9, 34]. Grabcut [34] extended graph-cut approaches by proposing an iterative algorithm of the optimization utilizing a bounding box. BoxSup [9] proposed a training procedure where the network is trained with automatically generated region proposals and after that refines the segmentations for training in an iterative manner. On the other hand, SimpleDoesIt [1] proposed an approach for normal segmentation training procedure by using well designed masks. Our proposed iterative algorithm for generating the mask of the first frame extends predecessor by choosing high quality masks while training, in order to prevent failure cases.

Visual object tracking. Visual object tracking is one of the fundamental tasks in computer vision, commonly used as assistance in multimedia applications such as surveillance system [14]. Given an initial bounding box in the first frame of a video sequence, it follows the target object in the following frames. Recently, deep learning based tracking methods have received significant attention and archived dominated performance on general visual object tracking benchmarks [26, 27, 43]. ECO [10] is an improved version based on C-COT [12]. It introduces a factorized convolution operator

based on Discriminative Correlation Filter (DCF), which significantly reduces the complexity of the model and memory usage while improves robustness.

3 OUR APPROACH

The overall structure of our tracking-assisted video object segmentation framework (except the first frame) is illustrated in Figure 2. More details of the first-frame processing will be depicted in §3.2. For each frame t in the sequence (except the first frame), the tracker (i.e., Figure 2-(2)) first predicts the target's location, illustrated as the yellow box in Figure 2-(3). The tracker guides the segmentation to focus on a smaller region around the target, i.e., the cyan box in Figure 2-(3). After one-round forwarding both appearance network and contour network in segmentation (see §3.1), the initial segmentation results are obtained, which shows with the red mask in Figure 2-(5). As shown in Figure 2-(6) (see §3.3), the tracker refines segmentation by locating the connected mask around the predicted target's location while the segmentation updates tracker's target position according to the outer bound of the mask, which leads to the outputs of both tracker and segmentation in Figure 2-(7).

3.1 Baseline

Our tracking-assisted segmentation framework is flexible and in general applicable for different existing video object segmentation modules, e.g., OSVOS [5] and MSK [31], and online tracking modules, such as ECO [10], C-COT [12] and the other trackers with hand-crafted feature [11, 18, 20]. In this paper, we adopt OSVOS and ECO as our segmentation module and general tracking module considering their high performance and flexibility, respectively.

OSVOS contains two main parts, namely appearance network and contour network. OSVOS constructs the appearance network with a VGG Net [36] as a backbone (named as 'base network'), and connects it with a series of deconvolutional layers trained with DAVIS dataset [32] for pixel-level output. Furthermore, since OSVOS is a semi-supervised method, it utilizes a full annotated ground truth mask in the first frame to fine-tune the base network into a more specific network, namely 'parent network'. Meanwhile, OSVOS builds the contour network featuring VGG Net, which is trained with PASCAL-Context [33]. The contour network is then used to refine the outputs of the appearance network by means of Ultrametric Contour Map (UCM) [2] in order to generate the final segmentation results.

It is worth noticing that sequences in video object segmentation dataset, such as DAVIS, are generally well chosen for their high resolution, clear object appearance, and limited camera movement. The network trained with DAVIS in OSVOS is incapable of handling more general video sequences, such as surveillance videos. Therefore, in order to further improve the generality of the proposed framework, we adopt 101-layer Residual Network [19] (ResNet) to replace VGG Net in OSVOS and train it with Microsoft COCO 2017 dataset [28], which contains more objects and sceneries. For clearer comparison with the baseline, we leave the contour network in OSVOS and the tracking module (i.e., ECO) unchanged.

3.2 Initialization of Segmentation Task

Following up the setting of OSVOS, in order to fit specific objects in different sequences, the appearance network should be further fine-tuned from parent network with a pixel-level mask in the first frame. However, in our weakly-supervised framework, if we use the given bounding box in the first frame directly, the segmentation performance will degrade dramatically. Therefore, similar to the seminal GrabCut algorithm [34] and its recent successors [1, 9], we propose a simple but effective algorithm to generate fine object mask from an input bounding box to help construct the appearance network. Given the first frame i_0 of a sequence, the output of our initialization algorithm is a mask m_0 generated with the help of ground-truth bounding box b_0. Our proposed method first collects a set of **c**andidate masks $M_c = \{m : m \subset R^2\}$ by iteratively training the parent network. In each iteration, the prediction from forward propagating the current network is refined based on the contour and the bounding box to generate a new mask for the next iteration. Then a subset of masks is **s**elected $M_s \subseteq M_c$ with high probability of covering the target. Finally, to combine all masks in M_s together, the intersection of masks in M_s is returned as the mask m_0 for the first frame and it is used to fine-tune the parent network.

Candidate masks generation. Suppose totally we have n_t iterations to generate initial mask from input bounding box, in each iteration $i \in [1, n_t]$ the raw possibility map $p^{(i)} \subset R^2$ of the foreground object is obtained by forward propagating the current network with i_0. The first refinement we adopt is using the contour map $c_0 \subset R^2$ from contour network to snap onto $p^{(i)}$ by means of the superpixels using Ultrametric Contour Map (UCM) [2]. The threshold $\theta^{(i)}_{UCM}$ for assigning the 4-connected components from UCM controls the fineness of information from the contour, specifically, higher $\theta^{(i)}_{UCM}$ produces coarser information, resulting in lower controllability of c_0 and higher controllability of $p^{(i)}$ to the snapped mask. Secondly, all predictions outside b_0 are set to background. Then the resulting mask $m^{(i)}$ for iteration i is stored in M_c, and its **w**eight, calculated by dividing the area it covers by the area of b_0, is stored in W_c for later usage. At the end of each iteration, the network from previous iteration is trained for $\lfloor N/n_t \rfloor$ steps given the total training steps N and the training pair $\{i_0,\ m^{(i)}\}$.

To prevent an initialization failure on the first iteration caused by empty mask generated from the first forward propagation, which is common especially in real-world video sequences where the object is relatively small, the UCM threshold $\theta^{(i)}_{UCM}$ is initialized to a low value and gently increase with step size $\Delta\theta_{UCM}$ as the iteration goes larger. The algorithm which takes the bounding box as input and generates M_c described above is illustrated in Algorithm 1.

Final mask generation. With the set of candidate masks M_c and their corresponding weights W_c, we further choose a subset M_s of them, and obtain a final mask m_s by taking the intersection of M_s. To exclude empty masks in the mask candidates, we apply bi-class clustering to the weight of all candidates, and exclude the set with lower average weight, which typically consists of empty masks in the candidate set. Formally speaking, providing two subsets of weights for their corresponding masks $W_{c,1}$ and $W_{c,2}$,

$$W_{c,1} \cup W_{c,2} = W_c, \quad W_{c,1} \cap W_{c,2} = \emptyset \tag{1}$$

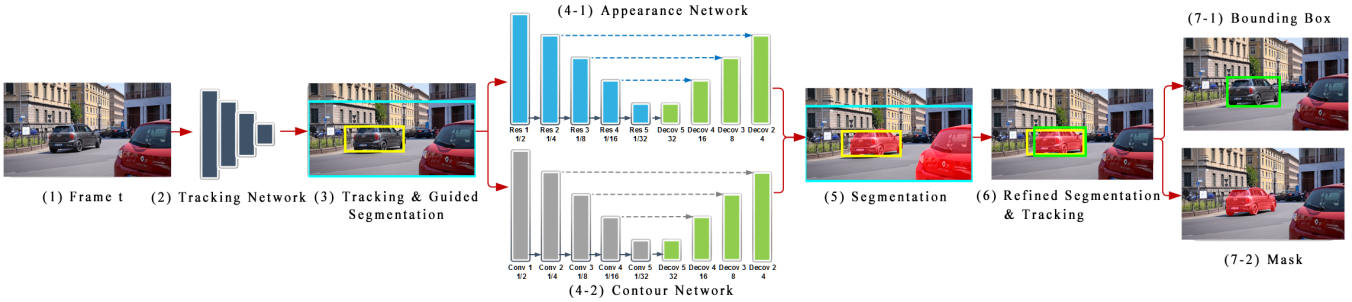

Figure 2: Tracking-assisted Weakly Supervised Segmentation Framework: (1) Input frame t. (2) A general tracking module, adopted from ECO [10]. (3) The tracker first helps to guide the segmentation. The yellow bounding box represents the predicted target position and size by tracker, while the cyan bounding box represents the area that is cropped for segmentation. (4) Segmentation module contains appearance network and contour network. (5) Initial segmentation results indicating with the red mask. (6) Tracking output and segmentation results help refine each other mutually. (7) Finally, two outputs are given, a bounding box for tracking, and a mask for segmentation. *Best viewed in color.*

Algorithm 1 Generate candidate set of masks for the first frame with bounding box

Given the first frame i_0 and the bounding box b_0.
Given the parent network $Net^{(1)}$ and UCM threshold $\theta^{(1)}_{UCM}$.
Given the contour network $Cont\_Net$.
Initialize M_c with {}.
Initialize W_c with {}.
$c_0 \leftarrow$ Forward($Cont\_Net$, i_0).
for $i = 1$ to n_t **do**
  $p^{(i)} \leftarrow$ Forward($Net^{(i)}$, i_0).
  $m^{(i)}_{raw} \leftarrow$ Snap($p^{(i)}$, c_0, $\theta^{(i)}_{UCM}$).
  $m^{(i)} \leftarrow \{m^{(i)}_{raw}(y, x) : (y, x)$ in $b_0\}$.
  $w^{(i)} \leftarrow$ Area($m^{(i)}$) / Area(b_0).
  $M_c \leftarrow M_c \cup \{m^{(i)}\}$.
  $W_c \leftarrow W_c \cup \{w^{(i)}\}$.
  $Net^{(i+1)} \leftarrow$ Fine-tune($Net^{(i)}$, $m^{(i)}$, $\lfloor N/n_t \rfloor$).
  $\theta^{(i+1)}_{UCM} \leftarrow \theta^{(i)}_{UCM} + \Delta\theta_{UCM}$.
end for
return $\{M_c, W_c\}$.

the subset of masks with higher average weight is chosen as M_s.

$$M_s = \{m^{(i)} \mid w^{(i)} \in argmax_{W_{c,i}}\{mean(W_{c,i})\}\} \tag{2}$$

To further exclude failed cases where the parent network is unable to separate the object from its surroundings, the final mask m_0 is then the pixel-wise intersection of M_s,

$$m_0 = \{\mathbb{1}_{(x,y)\ s.t.\ \forall m^{(i)} \in M_s,\ m^{(i)}(x,y)=1}\} \tag{3}$$

As a backup strategy, in case the parent network completely failed to detect the object, the region bounded by bounding box is filled with ones (denoting the foreground object) when less than 5% pixels inside the bounding box are labeled as foreground.

3.3 Tracking-assisted Segmentation Framework

At the beginning of each sequence, the parent network is fine-tuned with our generated mask derived from the input bounding box. In the following frames, the region of segmentation is guided by cropping around the target position obtained from tracker. Then the segmentation network generates a mask, and snaps with contour response from the contour network. After that, tracker and segmentation jointly refine the results by: (1) moving the bounding box provided by the tracker to cover as many pixels connected with the mask inside the bounding box as possible, and (2) excluding pixels outside the bounding box provided by segmentation to better focus on the target of interest. The output for each frame consists of both segmentation results and tracking results, illustrated in Figure 2.

To guide segmentation, we choose a crop region three times larger than the tracking bounding box. Then the cropped region is resized to fit the input dimension of the segmentation network. The benefits of this guiding strategy are two-fold. Firstly, cropping the frame helps the segmentation network focus on the target better. On the other hand, it helps to avoid irrelevant background objects from affecting the segmentation results. Instead of forward propagating the entire frame, this strategy preserves much more details especially in real-world video sequences with small targets.

We further use the tracker to refine the segmentation results. Starting from the tracking result bounding box $\{P, S\}$, where P and S stand for center location and size respectively, we first move and resize the bounding box such that all pixels connected to the mask originally inside $\{P, S\}$ are included in the afterward box $\{P', S'\}$. Then, due to segmentation's instability of including background noises, the update of bounding box is smoothened by two parameters θ_p and θ_s, controlling the impact on position and size, respectively. The updated bounding box $\{\widehat{P}, \widehat{S}\}$ is given by equation 4. Finally the mask inside $\{\widehat{P}, \widehat{S}\}$ is set as segmentation results.

$$\widehat{P} = P + \theta_p * (P' - P), \quad \widehat{S} = S + \theta_s * (S' - S) \tag{4}$$

4 EXPERIMENTS

4.1 Datasets and evaluation

We use two datasets to evaluate the segmentation performance of our framework, i.e., Densely Annotated VIdeo Segmentation (DAVIS) [32] and VOT2016 pixel-wise annotations [42].

Name	First Mask	Network VGG	Network ResNet	Combine Tracker?	$\mathcal{J}$
-GT	Groundtruth		✓	✓	**81.1**
-GT-V	Groundtruth	✓		✓	80.3
-GT-WoT	Groundtruth		✓		78.8
-BB	B-Box		✓	✓	62.6
-BB-V	B-Box	✓		✓	56.1
-V	Iterative	✓		✓	70.5
-WoT-V	Iterative	✓			71.7
-WoT	Iterative		✓		78.0
Ours	Iterative		✓	✓	**80.3**

Table 1: Ablation study on DAVIS: Evaluation of our framework using VGG Net (-V), without tracking (-WoT), fine-tuning with ground truth annotations (-GT), fine-tuning with masks obtained by filling the ground truth bounding box (-BB).

	θ_{UCM}	Training Steps N	With mask selection?	$\mathcal{J}$
SimpleDoesIt [1]	/	/	/	67.2
Grabcut [34]	/	/	/	58.6
Ours	0.4	50	✓	84.1
	0.6	50	✓	83.2
	Dy.	5	✓	84.4
	Dy.	100	✓	83.1
	Dy.	50		66.5
	Dy.	50	✓	**84.6**

Table 2: Ablation study on iterative algorithm for the first frame's mask: Evaluation of our iterative algorithm for generating the first frame from ground truth bounding box. Performance is compared on the first frame of each sequence in DAVIS 2016 train. In column θ_{UCM}, 'Dy.' means initializing θ_{UCM} with 0.1, and increasing by 0.1 in each iteration. Mask selection refers to the algorithm which generates M_s and process into final mask m_0 in §3.2

DAVIS. DAVIS is one of the most popular datasets for training and evaluating segmentation algorithms, which contains 50 high quality, Full HD video sequences with 3455 annotated frames in total. It covers multiple common segmentation challenges such as appearance changes, motion blur and occlusions. We report the average region similarity ($\mathcal{J}$) in the following discussion. Contour accuracy ($\mathcal{F}$) and temporal (in-)stability ($\mathcal{T}$) originally proposed in DAVIS [32] are not presented due to paper length limitation.

VOT2016. It contains 60 high-quality video sequences targeted at video tracking tasks, covering a wide range of challenges such as illumination changes, motion changes, occlusion and camera motion. In [42], Vojir et al. provided a set of manually segmented, pixel-level segmentation annotations for the VOT2016 dataset, constructing a challenging segmentation training and evaluation dataset. We evaluate with other state-of-the-art methods on this segmentation dataset, and report the average region similarity ($\mathcal{J}$).

4.2 Implementation details

We use a 101-layer Residual Network [19] pre-trained on ILSVRC [35] as the base network and replace its fully connected layers with deconvolutional layers. The settings for deconvolutional layers are kept the same as proposed in [5] except that all kernel sizes and strides in them are doubled to fit the output size of residual network blocks. We use masks from the COCO 2017 [28] dataset to train deconvolutional layers, with sigmoid cross entropy balance loss, base learning rate 0.0001 and step size 10,000. We use the contour network provided in [5] to snap contour onto appearance based prediction. We adopt the same parameters in [10] as our tracker.

While initializing at the test phase, we iteratively generate and refine the mask for 8 iterations, and fine-tune the current network for 50 steps in total to obtain the mask from the bounding box. We set $\theta^{(i)}_{UCM}$ to start from 0.1 and increase in each iteration with step size 0.1. We train the parent network with our generated mask for 500 steps at the initialization step, and set θ_p and θ_s to 0.8 and 0.1 in the combination step respectively. We use exactly the same configuration and parameter setting on all cross-domain datasets.

Our framework is tested on a machine with Intel Xeon @2.4GHz and Nvidia Tesla K80 graphics card. Our code are mainly written in MATLAB. We use Caffe for the segmentation network, Matconvnet for the ECO tracker. The combination of segmentation and tracking tasks is developed in MATLAB with the help of MatCaffe API. Following the above configuration, our framework takes around 60 (s) / 50 (iter) to generate the mask for the first frame consumes, and around 250 (s) / 500 (iter) to train the parent model for the current sequence. While testing, the joint framework runs at around 2 fps.

4.3 Ablation Study on DAVIS

In this section, firstly we analyze the impact of each component in our framework, i.e., the iterative weakly-supervised mask generating algorithm, the tracker-segmentation combination algorithm, and the use of ResNet trained with Microsoft COCO. Table 1 shows the evaluation of our framework with or without each component. We adopt the following abbreviations for various settings: (1) **-GT**: Fine-tuning parent network with ground truth mask. (2) **-BB**: Fine-tuning parent network with masks obtained by naively filling the

(Fix $\theta_s = 0.1$) θ_p	0.2	0.4	0.6	0.8	1.0
$\mathcal{J}$	79.3	79.5	80.2	**80.3**	79.8
(Fix $\theta_p = 0.8$) θ_s	0.04	0.06	0.1	0.3	0.5
$\mathcal{J}$	80.2	77.2	**80.3**	79.5	76.1

Table 3: Ablation study on DAVIS about combination parameter: Evaluation of our framework with different values of combination parameter θ_p and θ_s. While testing θ_p, θ_s is fixed to 0.1. While testing θ_s, θ_p is fixed to 0.8.

		Semi-Supervised				Unsupervised					Weakly-Supervised	
	Measure	OSVOS	MSK	SFLS	VPN	LVO	FSEG	LMP	FST	CUT	Ours-WoT	Ours
$\mathcal{J}$	Mean $\mathcal{M}\uparrow$	79.8	79.7	76.1	70.2	75.9	70.7	70.0	55.8	55.2	78.0	**80.3**
	Recall $\mathcal{O}\uparrow$	93.6	93.1	90.6	82.3	89.2	83.5	85.0	64.9	57.5	93.5	**95.2**
	Decay $\mathcal{D}\downarrow$	14.9	8.9	12.1	12.4	2.3	1.5	1.3	**-0.0**	2.2	8.5	5.4

Table 4: DAVIS Validation: Our method versus the state of art in terms of average region similarity $\mathcal{J}$. Ours-WoT represents the score of our segmentation network without the help of tracker. We can achieve state-of-art performance compared with both unsupervised and unsupervised methods. For rows with ↑, higher numbers are better, and vice versa for rows with ↓.

	Semi-Supervised				Unsupervised					Weakly-Supervised	
Measure	OSVOS	MSK	SFLS	VPN	LVO	FSEG	LMP	FST	CUT	Ours-WoT	Ours
AC	80.6	79.8	77.6	65.2	74.7	71.1	71.3	56.9	59.1	80.0	**83.6**
DB	**74.3**	74.1	54.9	44.4	55.7	50.0	58.3	46.9	35.4	68.4	70.2
FM	76.5	74.8	71.9	59.4	69.8	68.2	67.6	53.8	54.3	77.2	**78.1**
MB	73.7	73.4	74.1	64.8	70.6	63.6	64.8	47.6	52.4	74.8	**75.6**
OCC	**77.2**	75.5	71.0	71.2	73.0	61.5	69.2	46.4	41.2	75.8	75.6

Table 5: Per-attribute analysis on DAVIS Validation: Our method versus the state of art in terms of average region similarity $\mathcal{J}$ over all sequences with that specific attribute. AC stands for appearance change, DB for dynamic background, FM for fast motion, MB for motion blur, and OCC for occlusion.

ground truth bounding box. (3) **-WoT**: Without the help of tracker. (4) **-V**: Using VGG structure in the parent network.

Regarding the impact of our iterative algorithm for generating the first mask, two cases are compared: the ground truth annotation containing prior sequence-specific object information, and the mask obtained by naively filling the ground truth bounding box containing lots of background noises. The results of our framework using masks with different quality show a positive influence to the overall performance from the quality of the first mask. Our method with ground truth masks performs 1.0% higher than that with masks from our iterative method (-GT versus Ours). The score evaluated with our generated mask of the first frame achieves state-of-the-art performance compared with semi-supervised methods which use ground truth masks to fine-tune the parent network. One of the biggest benefits of our weakly-supervised framework is that our framework requires much less human effort on annotating the first frame, which has been one of the main problems limiting the application of semi-supervised video object segmentation algorithms in real world situations. The performance with our masks is 28.3% higher than that with masks obtained by naively filling the bounding box (Ours versus -BB), which shows the necessity of obtaining a mask from the bounding box on the first frame.

Table 1 also shows the impact of tracker assistance to the segmentation network. We achieve 2.9% better performance with tracker, with our generated first masks (Ours versus -WoT), and 2.9% better performance with tracker, and with ground truth masks (-GT versus -GT-WoT). Compared with traditional segmentation frameworks which tend to focus on objectness, the tracker in our framework focuses more on specific target object information besides objectness. This enables segmentation to locate the target better and separate out pixels with higher confidence belonging to the target.

Lastly, the performance of our tracking-assisted framework with ResNet is 13.9% higher than using VGG Net with tracker (Ours versus -V), and is 8.6% higher without tracker (-WoT versus -WoT-V), both showing the benefit of using ResNet as the base network with our iteratively generated mask. Our framework does encounter subtle performance drop using the ground truth mask and without tracker , 1.3% lower in our case (-GT-WoT versus OSVOS).

Iterative mask generation algorithm. To test the impact of each operation involved while generating the first frame mask with the ground truth bounding box, we examine the performance of our iterative algorithm on DAVIS 2016 train dataset, illustrated in Table 2. Besides, we also compare with other box-supervised segmentation methods, Grabcut [34] and SimpleDoesIt [1]. For dynamically changing θ_{UCM}, our strategy of increasing from 0.1 with step size of 0.1 outperforms static θ_{UCM} by 0.6% and 1.7% for $\theta_{UCM} = 0.4$ and $\theta_{UCM} = 0.6$, respectively. Regarding the total training steps N, we realize that the performance downgrades when N is too small or too large, with 0.2% and 1.8% comparing ours ($N = 50$) with $N = 5$ and $N = 100$. Besides, the mask selection and combination step play a key role in our iterative algorithm, where we observe significantly 27.2% better with this process enabled. Compared with other box-supervised segmentation algorithms, our method outperforms Grabcut and SimpleDoesIt by 44.4% and 25.9%, respectively. The main reasons for the improvement lie in the ability to use the pre-trained segmentation network, i.e., the parent network and generate the final mask from a stack of high-quality candidate masks in each iteration.

Tracking and segmentation combination. We further test the influence of two parameters θ_p and θ_s, which control the amount of influence each task gives to another. We find that updating tracker's

(a) Groundtruth (b) OSVOS[5] (c) MSK[31] (d) LVO[39] (e) FSEG[13] (f) Ours

Figure 3: Illustration of Segmentation Results on the DAVIS dataset: The green rectangle represents bounding box that tracker generate to help guide segmentation task. Our framework can achieve state-of-the-art performance compared with both semi-supervised (OSVOS and MSK) and unsupervised methods (LVO and FSEG). *Best viewed in color.*

center position in DAVIS dataset does not affect the final performance too much, as illustrated in Table 3. The reason is that the size of the target in DAVIS dataset is too big for an update to influence its memory of the target. However, the parameter θ_s, which controls the influence of size changes does affect the segmentation results. Because our framework is an online method and generic tracking methods tend to be not so stable, the performance with different combination parameters does not show linearity. We prove the generality of our method by using exactly the same set of parameters while testing on cross-domain datasets.

4.4 Comparison to State of the Art

Segmentation. To begin with, we first test our framework on DAVIS 2016 validation set, and the comparison of $\mathcal{J}$ is listed in Table 4. Per-attribute analysis is listed in Table 5. Our weakly-supervised framework can achieve state-of-the-art performance compared with both semi-supervised and unsupervised methods. Compared with unsupervised methods, our framework outperforms LVO by 5.8% and FSEG by 13.6%, while avoiding calculating Optical Flow, which is very computationally expensive and error-prone for real-world video sequences. Compared with semi-supervised methods, our framework achieves better results than OSVOS with 0.63%, given only a bounding box of the object instead of a pixel-level ground truth annotation for the first frame. Our framework without the help of tracker also achieves state-of-the-art performance compared with unsupervised methods, 2.8% better than LVO.

	OSVOS[5]	LVO[39]	FSEG[13]	Ours
$\mathcal{J}$	31.3	18.9	17.3	**31.8**

Table 6: VOT2016 pixel-wise annotations evaluation: Evaluation of our framework with other methods.

Then we test our framework on the VOT2016 pixel-wise annotations dataset, which is originally designed for tracking algorithm. It is much harder but closer to real-world scenarios. From Table 6, we observe that the results of our proposed framework achieve an improvement of 68.3% and 83.8% compared with unsupervised frameworks LVO and FusionSeg, respectively. We also achieve state-of-the-art performance compared with semi-supervised method OSVOS, with 1.6% improvement.

Tracking. We evaluate the tracker's performance in our framework on the DAVIS dataset and compare with both segmentation and tracking methods, including OSVOS [5], LVO [10], MDNET [29] and ECO [10]. Results listed in Table 7 show that the tracker in our framework achieves the best performance on DAVIS. Compared with tracking methods, our framework can overcome the difficulty of deformable objects and oversized objects, generating a better bounding box with the help of the segmentation mask. Compared with segmentation methods, our method can focus on the target object and provide a continuous and global guidance, preventing the mask of irrelevant objects from appearing in the final results.

Qualitative results. Because the objects in the DAVIS dataset is relatively big and clean, most recent segmentation methods can achieve fairly decent performance on it. But there are still some problems, for example, unsupervised methods work poorly when multiple objects appear in the frame. From the first row in Figure 3, the car in the background has a similar appearance to the target object, which confused unsupervised methods, leading to false positives. Although semi-supervised methods generally have better results than unsupervised, they work poorly on deformable objects. As shown in the last row in Figure 3, the appearance of motorbike changes dramatically compared with the first frame, causing semi-unsupervised method MSK to work poorly. Our approach further prevents these cases by providing guidance of the target's position.

(a) Groundtruth (b) OSVOS[5] (c) LVO[39] (d) FSEG[13] (e) Ours

Figure 4: Illustration of Segmentation Results on the VOT2016 pixel-wise annotation: Green rectangle represents bounding box that tracker generate to help guide segmentation task. Our framework outperforms both semi-supervised and unsupervised methods. But the failure of tracker is prone to affect the performance of segmentation, for example in the last row, the tracker failed to follow the girl but follows another man, causing the segmentation to lose the target. *Best viewed in color.*

Overlap	0.5	0.6	0.7	0.8	0.9	AVG
Ours	**88.1**	**83.8**	**76.3**	**66.8**	48.3	**72.7**
Ours-WoT	69.2	62.7	55.9	49.1	41.4	55.7
OSVOS[5]	78.2	72.2	65.8	59.4	**49.6**	65.0
LVO[39]	77.7	72.3	67.3	57.8	37.4	62.5
MDNET[29]	66.4	57.8	43.4	29.5	14.7	42.4
ECO[10]	59.7	49.0	36.2	22.8	12.4	36.0

Table 7: Tracking evaluation on DAVIS for percentage of overlap: Evaluation as a tracker of our framework with OSVOS and ECO on DAVIS. The percentage of bounding boxes which has no less than different threshold of overlap with ground truth bounding box is recorded.

Compared with DAVIS, sequences in VOT pixel-wise annotations dataset is more challenging and closer to real-world situations. Both semi-supervised and unsupervised methods have problems with these cross-domain datasets. For example, unsupervised methods cannot separate the main object from a frame and bring in a lot of irrelevant objects into their masks, like the scoreboard and car in the first and last row of Figure 4. In the third row of Figure 4, because the octopus is not moving, both optical-flow-based unsupervised methods LVO and FSEG fail to perform a segmentation. Semi-supervised method OSVOS also fails to segment the octopus because of lack of global constraints of the target. However, the instability of tracker on VOT pixel-wise annotations limits the segmentation results in some difficult sequences. For example, when tracker loses the target because of occlusion, the segmentation of our framework in the following frames will completely miss the target, which is shown in the last row of Figure 4.

5 CONCLUSION

In this paper, we have proposed a weakly supervised visual object segmentation framework assisted by a general object tracking module. By only inputting a bounding box in the first frame, our approach can overcome the intrinsic issue of the unsupervised approach. Meanwhile, it also provides continuous guidance to visual object segmentation module, which is usually missing in the semi-supervised approach. The generality and effectiveness of our method have been validated on the DAVIS and VOT2016 datasets, which usually belong to different research domains. With the same configuration and parameter setting, our method has obtained superior performance on both datasets. It has been shown that our approach with minimum supervision even outperforms the top semi-supervised methods.

ACKNOWLEDGMENT

This work was supported in part by National NSF of China (NO. 61525204, 61732010) and Shanghai Key Laboratory of Scalable Computing and Systems.

REFERENCES

[1] Khoreva Anna, Benenson Rodrigo, Hosang Jan, Hein Matthias, and Schiele Bernt. 2017. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. In *CVPR*.

[2] Pablo Arbelaez. 2006. Boundary extraction in natural images using ultrametric contour maps. In *CVPRW*.

[3] S Avinash Ramakanth and R Venkatesh Babu. 2014. Seamseg: Video object segmentation using patch seams. In *CVPR*.

[4] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. 2009. Video snapcut: robust video object cutout using localized classifiers. In *ACM Transactions on Graphics*, Vol. 28. ACM, 70.

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixe, Daniel Cremers, and Luc Van Gool. 2017. One-Shot Video Object Segmentation. In *CVPR*.

[6] Jason Chang, Donglai Wei, and John W Fisher. 2013. A video representation using temporal superpixels. In *CVPR*.

[7] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*.

[8] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. 2017. SegFlow: Joint Learning for Video Object Segmentation and Optical Flow. In *ICCV*.

[9] Jifeng Dai, Kaiming He, and Jian Sun. 2015. BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. In *ICCV*.

[10] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. 2017. ECO: Efficient Convolution Operators for Tracking. In *CVPR*.

[11] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*.

[12] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. 2016. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*.

[13] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. 2017. FusionSeg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos. In *CVPR*.

[14] L. Hanhe et al. 2016. Online Weighted Clustering for Real-time Abnormal Event Detection in Video Surveillance. In *ACMMM*.

[15] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2015. JumpCut: non-successive mask transfer and interpolation for video cutout. *ACM Transactions on Graphics* 34, 6 (2015), 195–1.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*.

[17] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. 2010. Efficient hierarchical graph-based video segmentation. In *CVPR*.

[18] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. 2016. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence* 38, 10 (2016), 2096–2109.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[20] Yang Hua, Karteek Alahari, and Cordelia Schmid. 2015. Online object tracking with proposal selection. In *ICCV*.

[21] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*.

[22] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. 2017. Video propagation networks. In *CVPR*.

[23] Hanqing Jiang, Guofeng Zhang, Huiyan Wang, and Hujun Bao. 2015. Spatio-temporal video segmentation of static scenes and its applications. *IEEE Transactions on Multimedia* 17, 1 (2015), 3–15.

[24] Margret Keuper, Bjoern Andres, and Thomas Brox. 2015. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*.

[25] Changick Kim and Jenq-Neng Hwang. 2002. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE transactions on circuits and systems for video technology* 12, 2 (2002), 122–129.

[26] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin, Tomás Vojír, Gustav Häger, and et al. Abdelrahman Eldesokey. 2017. The Visual Object Tracking VOT2017 Challenge Results. In *ICCV Workshop on Visual Object Tracking Challenge*.

[27] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin, Tomás Vojír, Gustav Häger, Alan Lukezic, Gustavo Fernández, Abhinav Gupta, and Alireza Memarmoghadam et al. 2016. The Visual Object Tracking VOT2016 Challenge Results. In *ECCV Workshop on Visual Object Tracking Challenge*.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

[29] Hyeonseob Nam and Bohyung Han. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*.

[30] Anestis Papazoglou and Vittorio Ferrari. 2013. Fast object segmentation in unconstrained video. In *ICCV*.

[31] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. 2017. Learning Video Object Segmentation From Static Images. In *CVPR*.

[32] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. 2016. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *CVPR*.

[33] X. Liu N.-G. Cho S.-W. Lee S. Fidler R. Urtasun R. Mottaghi, X. Chen and A. Yuille. 2014. The role of context for object detection and semantic segmentation in the wild. In *CVPR*.

[34] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* 23, 3 (2004), 309–314.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[36] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. (2015).

[37] Ying-Li Tian, Max Lu, and Arun Hampapur. 2005. Robust and efficient foreground analysis for real-time video surveillance. In *CVPR*.

[38] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. 2017. Learning motion patterns in videos. In *CVPR*.

[39] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. 2017. Learning Video Object Segmentation with Visual Memory. In *ICCV*.

[40] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. 2016. Video segmentation via object flow. In *CVPR*.

[41] David Varas and Ferran Marques. 2014. Region-based particle filter for video object segmentation. In *CVPR*.

[42] Tomas Vojir and Jiri Matas. 2017. *Pixel-Wise Object Segmentations for the VOT 2016 Dataset.* Research Report CTU–CMP–2017–01. Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic.

[43] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. 2015. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 9 (2015), 1834–1848.

[44] Chenliang Xu, Caiming Xiong, and Jason J Corso. 2012. Streaming hierarchical video segmentation. In *ECCV*.

[45] JY Zhou, Ee Ping Ong, and Chi Chung Ko. Video object segmentation and tracking for content-based video coding. In *ICME*.