Fast and Accurate Online Video Object Segmentation via Tracking Parts

Jingchun Cheng[1,2] Yi-Hsuan Tsai[3] Wei-Chih Hung[2] Shengjin Wang[1*] Ming-Hsuan Yang[2]
[1]Tsinghua University [2]University of California, Merced [3]NEC Laboratories America

Abstract

Online video object segmentation is a challenging task as it entails to process the image sequence timely and accurately. To segment a target object through the video, numerous CNN-based methods have been developed by heavily finetuning on the object mask in the first frame, which is time-consuming for online applications. In this paper, we propose a fast and accurate video object segmentation algorithm that can immediately start the segmentation process once receiving the images. We first utilize a part-based tracking method to deal with challenging factors such as large deformation, occlusion, and cluttered background. Based on the tracked bounding boxes of parts, we construct a region-of-interest segmentation network to generate part masks. Finally, a similarity-based scoring function is adopted to refine these object parts by comparing them to the visual information in the first frame. Our method performs favorably against state-of-the-art algorithms in accuracy on the DAVIS benchmark dataset, while achieving much faster runtime performance.

1. Introduction

Video object segmentation aims at separating target objects from the background and other instances on the pixel level. Segmenting objects in videos is a fundamental task in computer vision because of its wide applications such as video surveillance, video editing, and autonomous driving. However, it is a challenging task due to camera motion, object deformation, occlusion between instances and cluttered background. Particularly for online applications, significant different issues arise when the methods are required to be robust and fast without given access to future frames. In this paper, we focus on solving the problem of online video object segmentation. Given the object in the first frame, our goal is to immediately perform online segmentation on this target object without knowing future frames. For real application usages, the difficulties lie in the requirement of efficient runtime performance while maintaining accurate segmentation. Figure 1 illustrates comparisons of the state-of-the-art methods in terms of speed and performance, where we show that the proposed algorithm is fast, accurate and applicable to online tasks.

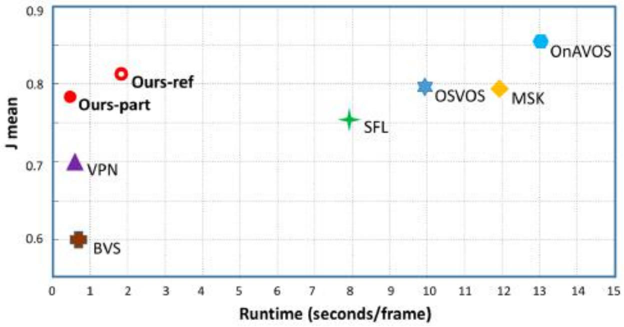

Figure 1. Accuracy versus runtime comparisons on the DAVIS 2016 dataset. We evaluate the state-of-the-art methods and demonstrate that our approach is significantly faster, while maintaining high accuracy. Note that the runtime includes the pre-processing steps averaged on all frames for fair comparisons.

Existing video object segmentation algorithms can be broadly classified into unsupervised and semi-supervised settings. Unsupervised methods [9, 14, 18, 35] mainly segment moving objects from the background without any prior knowledge of the target, e.g., initial object masks. However, these methods cannot handle multiple object segmentation as they are not capable of identifying a specific instance. In addition, several methods require batch model processing (i.e., all the frames are available) before segmenting the object [21, 41], which cannot be applied to online applications. On the other hand, semi-supervised methods [6, 16, 19, 20, 44] are given with an initial object mask which provides critical visual cues of the target. Thus, these methods can handle multi-instance cases and usually perform better than the unsupervised approaches. However, many state-of-the-art semi-supervised methods heavily rely on the segmentation mask in the first frame. For instance, before making predictions on the test video, the state-of-the-art methods need to finetune the networks for each video [4, 6, 19, 44], or the model for each instance [5, 34]. This finetuning step on the video or instance level is computationally expensive, where it usually takes more than ten minutes to update a model [4, 6]. In ad-

[*] Corresponding Author

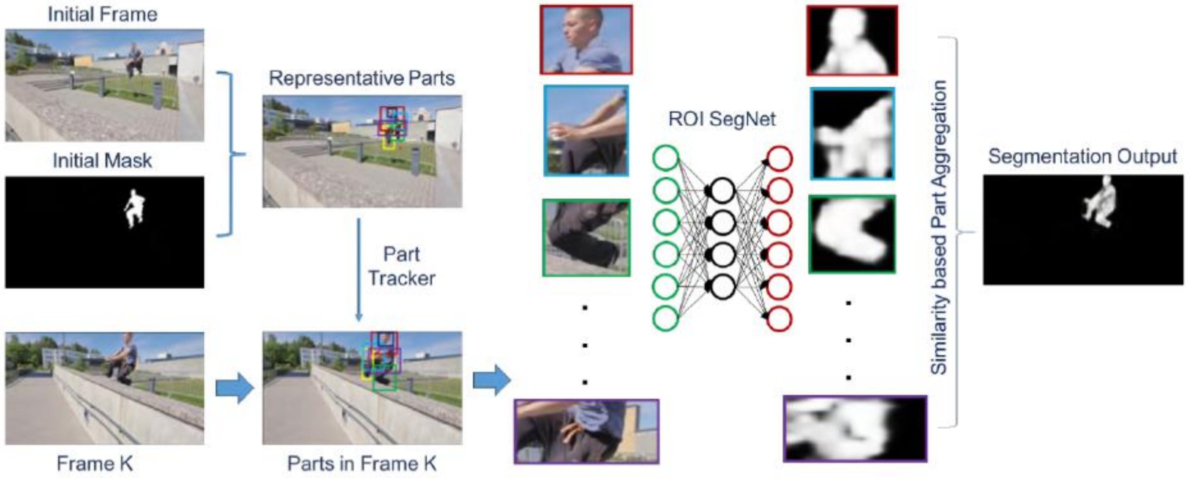

Figure 2. Proposed framework for online video object segmentation. Our algorithm first generates parts of the target object in the first frame. These parts are then tracked in the next frame to obtain tracking boxes. With our ROI segmentation network and a similarity-based scoring function, final segmentation outputs are generated through the entire video.

dition, data preparation (e.g., optical flow generation [42]) and training data augmentation [19] require additional processing time. As such, these methods cannot be used for time-sensitive online applications that require fast and accurate segmentation results of a specific target object (see Figure 1).

In this paper, we propose a video object segmentation algorithm that can immediately start to segment a specific object through the entire video fast and accurately. To this end, we utilize a part-based tracking method and exploit a convolutional neural network (CNN) for representations but does not need the time-consuming finetuning stage on the target video. The proposed method mainly consists of three parts: part-based tracking, region-of-interest segmentation, and similarity-based aggregation.

Part-based Tracking. Naturally, object tracking is an effective way to localize the target in the next frame. However, non-rigid objects often have large deformation with fast movement, thereby making it difficult to accurately localize the target [2, 8, 30]. To better utilize the tracking cues, we adopt a part-based tracking scheme to resolve challenging issues such as occlusions and appearance changes [27]. We first randomly generate object proposals around the target in the first frame, and select representative parts based on the overlapping scores with the initial mask. We then apply the tracker for each part to provide temporally consistent region of interests (ROIs) for subsequent frames.

ROI Segmentation. Once each part is localized in the next frame, we construct a CNN-based ROI SegNet to predict the segmentation mask that belongs to the target object. Different from conventional foreground segmentation networks [4, 6, 26] that focus on segmenting the entire object, our ROI SegNet learns to segment partial objects given the bounding box of part.

Similarity-based Aggregation. With part tracking and ROI segmentation, the object location and segmentation mask can be roughly identified. However, there could be false positives due to incorrect tracking results. To reduce noisy segmentation parts, we design a similarity-based method to aggregate parts by computing the feature distance between tracked parts and the initial object mask. Figure 2 shows the main steps of the proposed algorithm.

To validate the proposed algorithm, we conduct extensive experiments with comparisons and ablation study on the DAVIS benchmark datasets [36, 38]. We show that the proposed method performs favorably against state-of-the-art approaches in accuracy, while achieving much better runtime performance. The contributions of this work are as the following. First, we propose a fast and accurate video object segmentation method that is applicable to online tasks. Second, we develop the part-based tracking and similarity-based aggregation methods that effectively utilize the information contained in the first frame, without adding much computational load. Third, we design an ROI SegNet that takes bounding boxes of parts as the input, and outputs the segmentation mask for each part.

2. Related Work

Unsupervised Video Object Segmentation. Unsupervised video object segmentation methods aim to automatically discover and separate prominent objects from the background. These methods are based on probabilistic models [23, 31], motions [18, 17, 35], and object proposals [24, 46]. Existing approaches often rely on visual cues such as superpixels, saliency maps or optical flow to obtain initial object regions, and need to process the entire video in batch mode for refining object segmentation. In addition, generating and processing thousands of candidate regions in each frame is usually time-consuming. Recently, CNN-based methods [14, 40, 41] exploit learning rich hierarchical features (e.g., ImageNet pre-training) and large augmented data to achieve the state-of-the-art segmentation results. However, these unsupervised methods are not able to segment a specific object due to motion confusions between different instances and dynamic background.

Semi-supervised Video Object Segmentation. Semi-supervised methods aim to segment a specific object with an initial mask. Numerous algorithms have been proposed based on tracking [10], object proposals [37], graphical model [32], and optical flow [42]. Similar to the unsupervised approaches, CNN-based methods [4, 6, 20] have achieved significant improvement for video object segmentation. However, these methods usually heavily rely on finetuning models through the first frame [4, 20], data augmentation [19], online model adaptation [44] and joint training with optical flow [6]. These steps are computationally expensive (e.g., it takes more than 10 minutes for finetuning on the first frame in each video) and are not suitable for online vision applications.

To alleviate the issue of computational loads, a few methods are developed by propagating the object mask in the first frame through the entire video [15, 16]. Without exploiting much information in the first frame, these approaches suffer from the error accumulation after propagating a long period of time and thus do not perform as well as other methods. In contrast, the proposed algorithm incorporates part-based tracking and always keeps eyes on the first frame by a similarity-based part aggregation strategy.

Object Tracking. Tracking has been widely used to localize objects in videos as an additional cue for performing object segmentation [43]. Conventional methods [3, 13] adopt correlation filter-based trackers to account for appearance changes. Recently, numerous methods have been developed based on deep neural networks and classifiers. The CF2 method [30] learns correlation filters adaptively based on CNN features, thereby enhancing the ability to handle challenging factors such as deformation and occlusion. In addition, the SINT scheme [39] utilizes a Siamese network to learn feature similarities between proposals and the initial observation of target object. The SiaFC algorithm [2] develops an end-to-end Siamese tracking network with fully-convolutional layers, which allows the tracker to compute similarity scores for all the proposals in one forward pass. In this work, we adopt the Siamese network for tracking object parts, where each part is locally representative and endures less deformation through the video.

3. Proposed Algorithm

In this section, we describe each component of the proposed method. First, we present the part-based tracker, where the goal is to localize object parts through the entire video. Second, we construct the ROI SegNet, a general and robust network to predict segmentation results for object parts. Third, we introduce our part aggregation method to generate final segmentation results by computing similarity scores in the feature space.

3.1. Part-based Tracking

Object tracking is a difficult task due to challenging factors such as object deformation, fast movement, occlusion, and background noise. To deal with these issues, part-based methods [27] have been developed to track local regions instead of the entire object with larger appearance changes. Since our goal is to localize most object regions in the next frame for further segmentation, utilizing a part-based method matches our need and can effectively maintain a high recall rate.

Part Generation. In order to track parts, one critical problem is how to generate these parts in the first place. Conventional object parts are discovered from a large amount of intra-class data via discriminability and consistency. However, this assumption does not hold for online video segmentation, as only one object mask is provided in the first frame of the target video. To resolve this issue, we propose a simple yet effective way to generate representative parts guided by the object mask. First, we randomly generate part proposals with various sizes and locations around the object, and remove the ones with low overlapping ratio to the object mask. We compute the intersection-over-union (IoU) score between the proposal and the object, and keep the ones with scores larger than a threshold (i.e., 0.3 in this work). To ensure that each part contains mostly pixels from the object, we further measure the score: $\mathcal{S}_p = \frac{bbox \cap gtbox}{bbox}$, where $bbox$ is the bounding box of a proposal and $gtbox$ is the known object box in the first frame. Part proposals with $\mathcal{S}_p > 0.7$ are used as candidates for a non-maximum suppression (NMS) step. Based on the proposed selection process, we reduce thousands of proposals to only $50 \sim 300$ representative parts depending on the object size. Note that, we also transform the bounding box for each part to be tight within the object mask, reducing background noise for more effective tracking and segmentation. Some example results are shown in Figure 3 for generated parts (with high scores) in the first frame.

Part Tracking. Given a set of parts $\mathcal{P}_t = \{P_t^1, P_t^2, ..., P_t^i\}$ in frame I_t, our goal is to output a score map $\mathcal{S}_t$ that measures the location likelihood of part P_t^i appearing in the next frame I_{t+1}:

$$\mathcal{S}_t = \mathcal{T}(P_t^i, I_{t+1}), \tag{1}$$

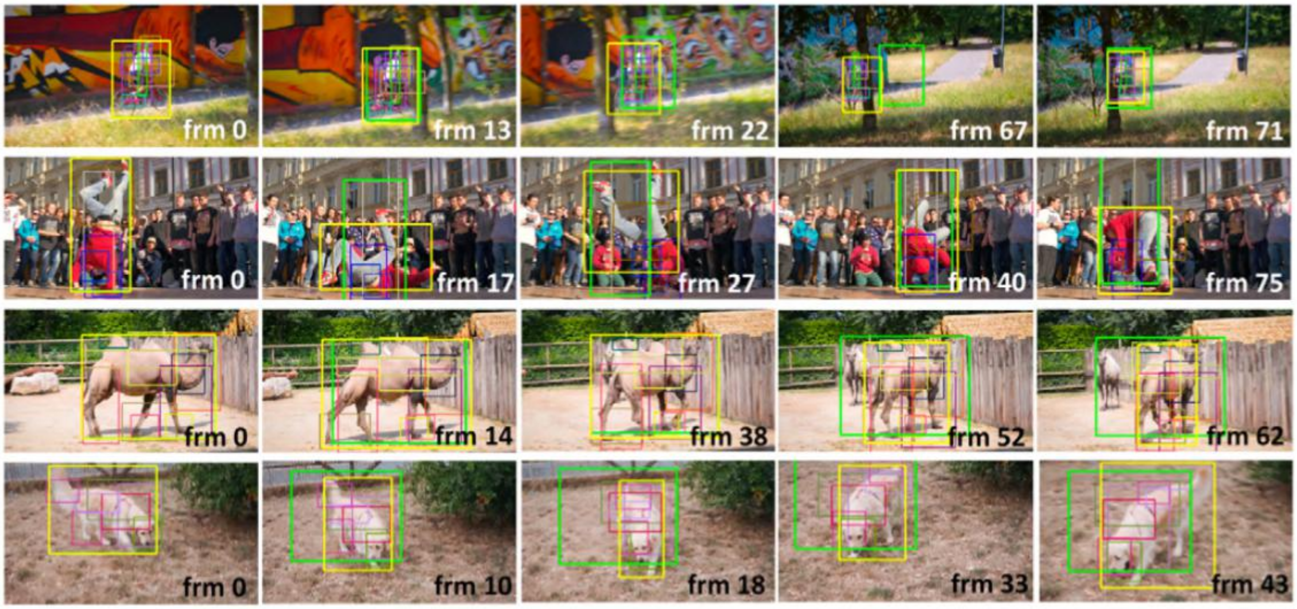

Figure 3. Sample results for part tracking. We show some high-scored parts and their tracking results. Green and yellow boxes are the results by applying object tracker [2] and by our method via aggregating parts, respectively. It shows that our result (yellow boxes) are robust to object deformation and occlusion, due to the stability of tracking parts.

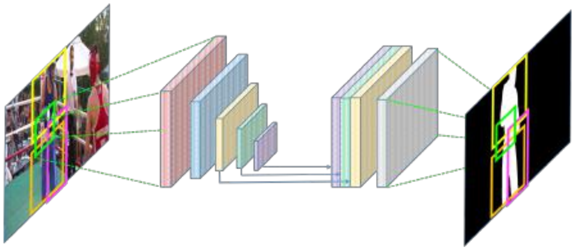

Figure 4. Illustration of the proposed ROI SegNet. Given an image and their parts, we resize and align each part as the input to the network. We use the ResNet-101 architecture containing 5 convolution modules. We up-sample and concatenate feature maps from the last three modules. An additional convolution layer is utilized for the binary prediction of parts.

where $\mathcal{T}$ is a function to compute similarity scores between the part P_t^i and the image I_{t+1}. We use the SiaFC method [2] as our baseline tracker $\mathcal{T}$ to compute the score map $\mathcal{S}_t$. Due to its fully-convolutional architecture, we compute score maps for multiple parts in one forward pass. Once obtaining the score map, we select the bounding box with the largest response as the tracking result. Some tracking results are shown in Figure 3.

3.2. ROI SegNet

Based on the tracking results of object parts, the next task is to segment partial object within the bounding box. Recent instance-level segmentation methods [11, 7] have demonstrated the state-of-the-art results by training networks for certain categories and output their segmentations. Our part segmentation problem is similar to the instance-level segmentation task but for the partial object. In addition, training such a network would require an alignment step for different parts as they may vary significantly in size, shape, and appearance for different instances or object categories. Hence, we utilize an ROI data layer by cropping image patches from parts as inputs to the network, in which these patches are aligned through resizing. Similar to semantic segmentation, our objective is to minimize the weighted cross-entropy loss for a binary (foreground/background) task:

$$\begin{aligned}\mathcal{L}(P) = -(1-w)\sum_{i,j\in fg}\log\mathbb{E}(y_{ij}=1;\theta)\\ -w\sum_{i,j\in bg}\log\mathbb{E}(y_{ij}=0;\theta),\qquad(2)\end{aligned}$$

where θ denotes CNN parameters, y_{ij} denotes the network prediction for the input part P at pixel (i,j) and w is the foreground-background pixel-number ratio used to balance the weights [45].

Network Architecture. We utilize the ResNet-101 architecture [12] as the base network for segmentation and transform it to fully-convolutional layers [29]. To enhance feature representations, we up-sample feature maps from the last three convolution modules and concatenate them together. The concatenated features are then followed by a convolution layer for the binary prediction. Figure 4 shows the architecture of our ROI SegNet.

Network Training. To train the proposed network, we first augment images from the training set of the DAVIS dataset [36] via random scaling and affine transformations (i.e., flipping, ±10% shifting, ±10% scaling, ±30° rotation). Then, parts are extracted for each instance as the same method as introduced in part-based tracking. We use the Stochastic Gradient Descent (SGD) optimizer with the patch size 80×80 and the batch size of 100 for training. The initial learning rate starts from 10^{-6} and decreases by half for every 50,000 iterations. We train the network for 200,000 iterations.

3.3. Similarity-based Part Aggregation

After obtaining all the segmentation results from parts, one simple way to generate the final segmentation is to compute an averaging score map from each part. However, parts may be tracked off the object or include background noise, resulting in inaccurate part segments. To avoid adding these false positives, we develop a scoring function by looking back to the initial object mask. That is, we seek to know if the current part is similar to any of the parts in the first frame. Although objects may appear quite differently from the first frame, we find that local parts are actually more robust to such appearance changes.

Specifically, we first compute the similarity score between each part in $\mathcal{P}_t$ at frame t and initial parts $\mathcal{P}_0$ in the feature space. Then we select part P_0^n with the highest similarity for the current part P_t^m by:

$$n = \underset{i \in N}{\operatorname{argmin}} \|f(P_t^m) - f(P_0^i))\|_2^2, \tag{3}$$

where f is the feature vector representing each part, extracted from the last layer in our ROI SegNet with an average pooling on the part mask. Overall, our scoring function consists of three components:

$$\mathcal{S}_{seg}(\mathcal{P}_t) = \mathcal{S}_{ave}(\mathcal{P}_t) \cdot \mathcal{S}_{sim}(\mathcal{P}_t, \mathcal{P}_0^n) \cdot \mathcal{S}_{con}(\mathcal{P}_0^n), \tag{4}$$

where $\mathcal{P}_0^n$ is a set of initial parts selected based on Equation (3) and $\cdot$ is the element-wise multiplication operation. The first function $\mathcal{S}_{ave}$ is the simple averaging score of part segments in the current frame t:

$$\mathcal{S}_{ave}(\mathcal{P}_t) = \sum_{i \in \mathcal{P}_t} S^i / |\mathcal{P}_t|, \tag{5}$$

where $\mathcal{P}_t$ is the set of parts at frame t and S^i is the segmentation score for each part i. Second, $\mathcal{S}_{sim}$ is the similarity score between current and initial parts in the feature space based on (3). Since the selected initial part segment may have poor quality, we add $\mathcal{S}_{con}$ by forwarding $\mathcal{P}_0^n$ to the ROI SegNet and measuring its segmentation overlapping ratio to the initial mask as the confidence score:

$$\mathcal{S}_{con}(\mathcal{P}_0^n) = J(G(\mathcal{P}_0^n), gt), \tag{6}$$

where J is the IoU measurement, G is the ROI SegNet and gt is the object mask in the first frame. With the guidance of the initial object mask and parts without using expensive model finetuning step, our part aggregation method can effectively remove false positives. Figure 5 shows some examples of score maps with different scoring functions.

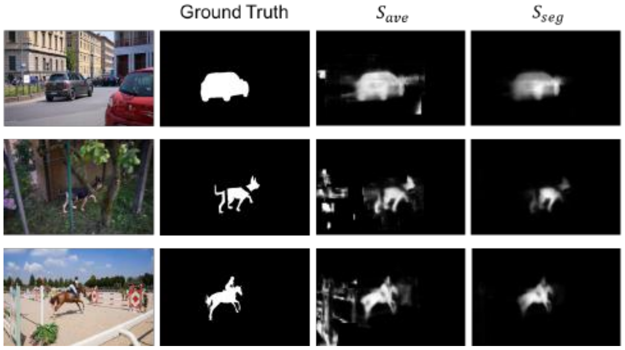

Figure 5. Part aggregation results. We compare score maps via the functions of $\mathcal{S}_{ave}$ and $\mathcal{S}_{seg}$. Without computing the similarity score to the first frame, the result of $\mathcal{S}_{ave}$ contains noisy segments, while our aggregation algorithm performs segmentation more precisely.

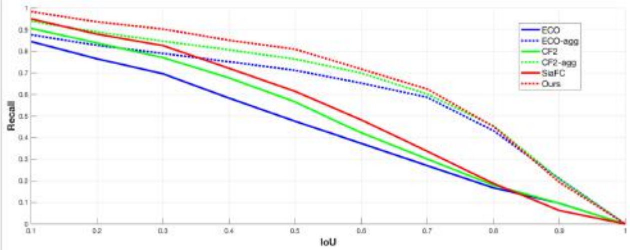

Figure 6. IoU-Recall curve for trackers on the DAVIS 2016 dataset. Dashed lines (-agg) denote results by utilizing the proposed part-based tracking.

4. Experimental Results

4.1. Dataset and Evaluation Metrics

We conduct experiments on the DAVIS benchmark datasets [38, 36] which contain high-quality videos with dense pixel-level object segmentation annotations. The DAVIS 2016 dataset consists of 50 sequences (30 for training and 20 for validation), with 3,455 annotated frames of real-world moving objects. Each video in the DAVIS 2016 dataset contains a single annotated foreground object, so both semi-supervised and unsupervised methods can be evaluated. The DAVIS 2017 dataset contains 150 videos with 10,459 annotated frames and 376 object instances. It is a challenging dataset as there are multiple instances in each video, where objects could occlude each other. In this setting, it is difficult for unsupervised methods to separate different instances. For performance evaluation, we use the mean region similarity (J mean), contour accuracy (F mean) and temporal stability (T mean) as in the benchmark setting [38, 36]. The source code

Table 1. Ablation study on DAVIS 2016. "+ $\mathcal{S}_{seg}$" and "+ $\mathcal{S}_{seg}$ + Tracker + CRF" denote results for **Ours-part** and **Ours-ref** in Figure 1, respectively.

Method	Baseline	+ [2]	+ $\mathcal{S}_{ave}$	+ $\mathcal{S}_{seg}$	+ $\mathcal{S}_{seg}$ + Tracker	+ $\mathcal{S}_{seg}$ + Tracker + CRF
J Mean ↑	0.707	0.696	0.739	0.779	0.786	0.824
J Recall ↑	0.840	0.860	0.874	0.924	0.929	0.965
J Decay ↓	-0.005	0.008	0.072	0.067	0.054	0.045
F Mean ↑	0.695	0.671	0.727	0.760	0.772	0.795
F Recall ↑	0.786	0.790	0.792	0.849	0.869	0.894
F Decay ↓	-0.004	-0.003	0.089	0.076	0.060	0.055
T Mean ↓	0.260	0.321	0.240	0.229	0.219	0.263

and models are available at `https://github.com/JingchunCheng/FAVOS`. More results and analysis are presented in the supplementary material.

4.2. Tracker Evaluation

Our part-based tracker focuses on tracking local regions and cannot directly output the object location in the next frame. However, we can roughly find the object center based on the aggregated part segments. Motivated by the tracking-by-detection algorithms [1], we utilize detection proposals [28] as candidates of object bounding boxes, and select the one closest to the object center as the tracking result. We then validate this part-based tracker on the DAVIS 2016 dataset with comparisons to our baseline SiaFC method [2] and other tracking algorithms including CF2 [13], ECO [8], and MDNet [33]. Experimental results are presented in Figure 3 and 6, where we show that our part-based trackers consistently maintain better IoU-recall curves for localizing objects.

Although our ultimate goal is for video object segmentation, this evaluation is useful for understanding the challenges on the DAVIS dataset. One interesting fact is that if there is a good tracker, it should be able to help the segmentation task. Thus, a high recall rate under a high IoU is required as once partial object is missing, it is not possible to recover the corresponding segment. As shown in Figure 6, most trackers achieve around 60% recall rate under a 0.5 IoU while ours is 80%, which enables potential usages of applying our tracker to improve segmentation results. We will present our results by integrating this tracker in the ablation study section.

4.3. Ablation Study on Segmentation

We present ablation study in Table 1 on the DAVIS 2016 validation set to evaluate the effectiveness of each component in the proposed video object segmentation framework. We start with the unsupervised version of SFL [6] as our baseline due to its balance between speed and accuracy. To demonstrate the usefulness of using part, we first conduct an experiment by combining the baseline result and the score map from [2] via tracking an entire object. Specifically, we average the foreground probability from [6] and the segmentation map of [2] through the ROI SegNet. However, we find that the tracking accuracy is highly unstable, which usually loses objects and even results in a worse performance than the baseline segmentation (1.1% drop in J Mean). It shows that combining tracking and segmentation is not a trivial task, and we use part-based model to achieve a better combination.

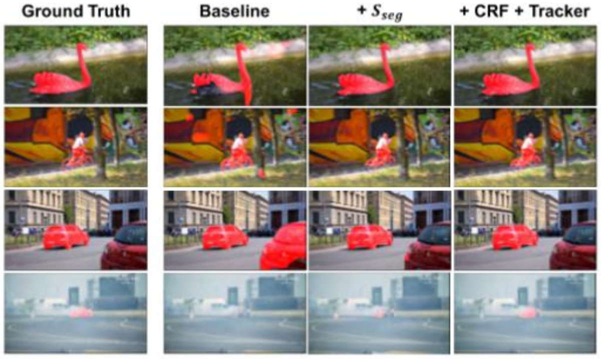

Figure 7. Sample results of using different components in the proposed method. We show gradual improvement over baseline with part aggregation, CRF refinement and an object tracker.

After adopting our part-based tracker and ROI SegNet to obtain part segments, we compare results with or without part aggregation. The one that utilizes part aggregation via the function $\mathcal{S}_{seg}$ in Equation (4) performs better (4% improvement in J Mean) than only computing the score function $\mathcal{S}_{ave}$. It shows that with the consideration of initial object mask, false part segmentations can be largely reduced as they are not similar to any of the object parts in the first frame. In addition, we take advantage of our tracker combined with detection proposals as mentioned in Section 4.2 and use it to further improve our results, denoted as "+Tracker" in Table 1. To further improve the boundary accuracy, we add a refinement step using dense CRF [22]. In Figure 1, we denote the result of using $\mathcal{S}_{seg}$ as *Ours-part*, and the one combined with our tracker and CRF with refinement as *Ours-ref*.

4.4. Segmentation Results

DAVIS 2016. We evaluate our proposed method on the validation set of DAVIS 2016 [36] with comparisons to state-of-the-art algorithms, including semi-supervised and unsupervised settings. In Table 2, we show results with different settings, including the need of initial object mask, future frames and pre-processing steps. Based on these requirements and their runtime speed, we then analyze the capability for online applications.

Table 2. Overall segmentation results on DAVIS 2016. We analyze various settings for different algorithms as well as provide online applicability based on their runtime speed (with different colors).

Method	Initial mask	Future frames	Pre-processing	Online	Speed	J mean	F mean	T mean
OnAVOS [44]	✓		finetuning	weak	13s	0.861	0.849	0.190
Lucid [19]	✓		data, finetuning	weak	40s	0.848	0.823	0.158
Ours-ref	✓		no	strong	1.8s	0.824	0.795	0.263
OSVOS [4]	✓		finetuning	weak	10s	0.798	0.806	0.378
MSK [20]	✓		flow, finetuning	weak	12s	0.797	0.754	0.218
Ours-part	✓		no	strong	0.60s	0.779	0.760	0.229
ARP [21]		✓	data	no	-	0.762	0.706	0.393
SFL [6]	✓		finetuning	weak	7.9s	0.761	0.760	0.189
LVO [41]		✓	flow	no	-	0.759	0.721	0.265
CTN [16]	✓		flow	weak	29.95s	0.735	0.693	0.220
FSEG [14]			flow	weak	7s	0.707	0.653	0.328
VPN [15]	✓		no	strong	0.63s	0.702	0.655	0.324
LMP [40]			flow	weak	18s	0.700	0.659	0.572
OFL [42]	✓		flow	weak	60s	0.680	0.634	0.222
BVS [32]	✓		no	strong	0.84s	0.600	0.588	0.347

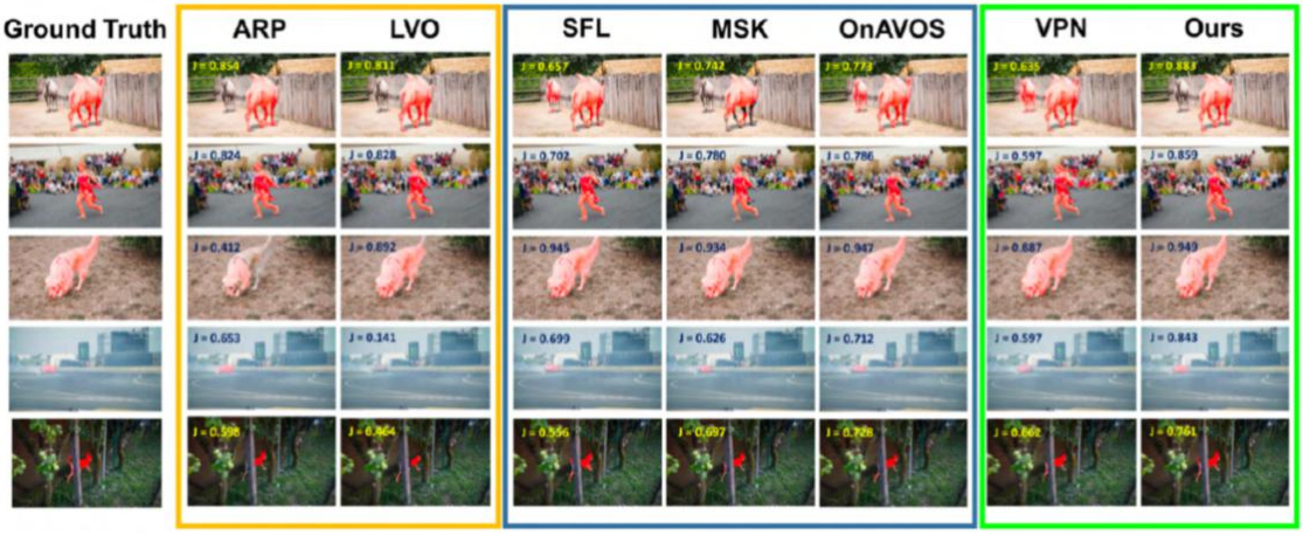

Figure 8. Example results of comparisons between state-of-the-art methods on DAVIS 2016. Approaches with *no*, *weak*, *strong* online applicability are marked in yellow, blue and green, respectively.

For unsupervised methods that do not need the initial mask, they usually need to compute optical flow as the motion cue (FSEG [14] and LMP [40]) or foresee the entire video (LVO [41] and ARP [21]) to improve the performance, which is not applicable to online usages. In addition, these methods cannot distinguish different instances and perform segmentation on a specific object.

In the semi-supervised setting, recent methods require various pre-processing steps before starting to segment the object in the video, which weaken the ability for online applications. These pre-processing steps include model finetuning (OnAVOS [44], Lucid [19], OSVOS [4], MSK [20], SFL [6]), data synthesis (Lucid [19]) and flow computing (MSK [20], CTN [16], and OFL [42]). For fair comparisons in the online setting, these pre-processing steps are included in the runtime by averaging on all the frames.

The most closest setting to our method is VPN [15] and BVS [32] that do not have heavy pre-processing steps. However, these approaches may propagate segmentation errors after tracking for a long period of time. In contrast, our algorithm always constantly refers to the initial object mask via parts and can reduce such errors in the long run, improving more than 12% in J Mean against VPN [15]. Overall, the proposed video object segmentation framework

Table 3. Segmentation results on DAVIS 2017 validation set. We show our baseline results with different modules, including foreground/background regularization (FG), Spatial Propagation Network (SPN) and a refinement procedure.

	Finetuning	Method	Baseline	+ FG	+ FG	+FG
					+ SPN	+SPN
						+Refine
Ours		J Mean ↑	0.451	0.462	0.481	0.546
SPN [5]	✓		0.442	0.457	0.506	0.540
Ours		F Mean ↑	0.554	0.571	0.574	0.618
SPN [5]	✓		0.453	0.504	0.568	0.611

runs at the fastest speed, and can achieve J Mean in the 3rd place with further refinement, while still maintaining a fast runtime speed compared to state-of-the-art methods. Some qualitative comparisons are presented in Figure 8.

DAVIS 2017. To evaluate how our method deals with multiple instances in videos, we conduct experiments on the DAVIS 2017 validation set [38] which consists of 30 challenging videos and each one has two instances on average. Existing methods all rely on sophisticated processing steps [25] to achieve better performance, and hence we compare our method with SPN [5] that only involves the finetuning step in Table 3. For the baseline algorithm, we start with our part-based aggregation method via part-based tracker and ROI SegNet, while [5] finetunes a CNN-based model for each instance. The baseline results show that, without the need of the computationally expensive finetuning process, our method even outperforms the existing method. One reason is that as the video becomes more complicated, finetuning-based methods may suffer from confusions between instances. In contrast, our method employs a part-based tracker that can effectively capture local cues for further segmentation.

Following [5], we then sequentially add different components, including foreground/background regularization, a Spatial Propagation Network and a region-based refinement step. In addition, we integrate the object tracker proposed in Section 4.2 to further refine the segmentation. Overall, without the need of finetuning on each instance, our approach achieves a similar performance or outperforms the one that requires finetuning. We also note that finetuning is expensive not only in speed but also in stored size, as hundreds of objects would result in a huge number of stored models, which is not practical in real-world applications. In Figure 9, we present some example results on the DAVIS 2017 dataset.

Runtime Analysis. In the proposed framework, our method runs at 0.60 seconds on average per instance per frame without the refinement step, including part-based tracking (0.2s), ROI segmentation (0.3s), and part aggregation (0.1s). With CRF (1s) and tracker (0.2s) refinements, our method runs at 1.8 seconds per instance per frame with better performance. We note that for tracking and segmenting parts, we parallelly use Titan X GPUs to handle hundreds of parts for faster inference.

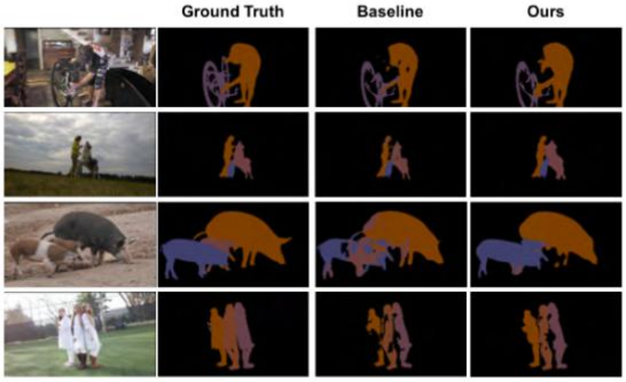

Figure 9. Example results for multiple instances on DAVIS 2017.

5. Concluding Remarks

In this paper, we propose a fast and accurate video object segmentation method that is applicable to online applications. Different from existing algorithms that heavily rely on pre-processing the object mask in the first frame, our method exploits the initial mask via a part-based tracker and an effective part aggregation strategy. The part-based tracker provides good localization for local regions surrounding the object, ensuring that most portion of the object is retained for further segmentation purpose. We then design an ROI segmentation network to accurately output partial object segmentations. Finally, a similarity-based scoring function is developed to aggregate parts and generate the final result. Our algorithm exploits the strength of CNN-based frameworks for tracking and segmentation to achieve fast runtime speed, while closely monitoring the information contained in the first frame for the state-of-the-art performance. The proposed algorithm can be applied to other video analytic tasks that require fast and accurate online video object segmentation.

Acknowledgments. This project is supported in part by the NSF CAREER Grant #1149783, gifts from Adobe and NVIDIA.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009. 6

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016. 2, 3, 4, 6

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 3

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 2, 3, 7

[5] J. Cheng, S. Liu, Y.-H. Tsai, W.-C. Hung, S. De Mello, J. Gu, J. Kautz, S. Wang, and M.-H. Yang. Learning to segment instances in videos with spatial propagation network. In *CVPR Workshop*, 2017. 1, 8

[6] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017. 1, 2, 3, 6, 7

[7] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 4

[8] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 2, 6

[9] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 1

[10] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011. 3

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *CVPR*, 2017. 4

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 2015. 3, 6

[14] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017. 1, 3, 7

[15] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, 2017. 3, 7

[16] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *CVPR*, 2017. 1, 3, 7

[17] W.-D. Jang, C. Lee, and C.-S. Kim. Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In *CVPR*, 2016. 2

[18] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015. 1, 2

[19] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. *arXiv:1703.09554*, 2017. 1, 2, 3, 7

[20] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 1, 3, 7

[21] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 1, 7

[22] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 6

[23] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 2

[24] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2

[25] X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. C. Loy, and X. Tang. Video object segmentation with re-identification. In *CVPR Workshop*, 2017. 8

[26] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2

[27] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*, 2015. 2, 3

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 6

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 4

[30] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015. 2, 3

[31] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012. 2

[32] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 3, 7

[33] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 6

[34] A. Newswanger and C. Xu. One-shot video object segmentation with iterative online fine-tuning. In *CVPR Workshop*, 2017. 1

[35] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 1, 2

[36] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 5, 6

[37] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *CVPR*, 2015. 3

[38] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2, 5, 8

[39] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016. 3

[40] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 3, 7

[41] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 1, 3, 7

[42] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 2, 3, 7

[43] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *ECCV*, 2016. 3

[44] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 1, 3, 7

[45] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 4

[46] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 2