Multimedia Tools and Applications (2019) 78:33617–33631
https://doi.org/10.1007/s11042-019-08146-3

Effective online refinement for video object segmentation

Gongyang Li [1,2] · Zhi Liu [1,2] · Xiaofei Zhou [3]

Received: 7 November 2018 / Revised: 5 July 2019 / Accepted: 2 September 2019 /
Published online: 9 September 2019

Abstract
In this paper, we propose a novel framework, which deeply explores the motion cue and the online fine-tuning strategy to tackle the task of semi-supervised video object segmentation. First, in order to filter out the irrelevant background regions in the initial segmentation results, which are generated by an existing semi-supervised segmentation model, a motion based background suppression method is exploited to obtain the purified segmentation results. Second, a set of key frames with high-quality segmentation results are selected based on several metrics of segmentation quality in the purified segmentation results. Finally, the selected key frames are combined with the manually annotated first frame to efficiently retrain the segmentation model online, so as to obtain more accurate segmentation results. Our experimental results on two challenging datasets demonstrate that the proposed framework achieves the state-of-the-art performance.

Keywords Video object segmentation · Motion cue · Online fine-tuning

1 Introduction

Video object segmentation has become a booming research topic in recent years. The pixel-level object segmentation results provide a better understanding of video and have many useful

✉ Zhi Liu
liuzhisjtu@163.com

Gongyang Li
ligongyang@shu.edu.cn

Xiaofei Zhou
zxforchid@outlook.com

[1] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China

[2] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

[3] Institute of Information and Control, Hangzhou Dianzi University, Hangzhou 310018, China

applications in the real world, e.g. video editing, video summarization, video compression and video content retrieval. Up to now, numerous video object segmentation methods have been proposed, and they can be divided into two categories: unsupervised and semi-supervised. For the former, the unsupervised approaches [3, 6, 10, 11, 13, 14, 21, 22, 26, 28] are designed to obtain pixel-level masks of objects in a video without human input during test time. Different from the former, the latter one, i.e. semi-supervised approaches [1, 2, 7, 8, 16, 18, 23–25, 27], relies on the pixel-level mask of the first frame, which is annotated by human, to segment the pixels belonging to the specific object in a video. In this paper, we focus on the semi-supervised video object segmentation.

Recently, convolutional neural network (CNN) based semi-supervised approaches, which utilizes large image/video datasets for pre-training a segmentation model offline and then modifies it by online fine-tuning on the test video with the ground truth pixel mask of the first frame, improved the segmentation performance significantly. The recently proposed one-shot video object segmentation (OSVOS) [1] approach has received numerous attentions. In general, OSVOS adopts online fine-tuning strategy for video object segmentation and it consists of three stages including Base Network, Parent Network, and Test Network. Concretely, firstly, OSVOS adopts a pre-trained CNN model for image labeling on ImageNet to obtain the Base Network. Then, for video object segmentation, the Base Network is trained on the training set of DAVIS dataset, yielding the Parent Network. Finally, in the testing phase, OSVOS fine-tunes the Parent Network on the first frame of the test sequence, and obtains the Test Network that can be employed to segment the specific target object. However, OSVOS uses the Test Network in unchanged form and is inadequate for large appearance change, deformation and fast motion. Another concurrent work named learning video object segmentation from static images (MSK) [18], which also uses online fine-tuning strategy for video object segmentation and considers the temporal information among adjacent video frames, i.e. optical flow. MSK employs the current frame and the optical flow with a rough mask estimate from the previous frame to provide a refined mask output for the current frame.

Inspired by [1, 18], in this paper we propose a novel framework which deeply explores the motion cue, i.e. optical flow, and the online fine-tuning strategy. We propose a motion based background suppression method to filter out the irrelevant background regions and to purify the initial segmentation results, which are generated from an existing semi-supervised segmentation model. Besides, we also take full advantage of the online fine-tuning strategy. As demonstrated in the previous works [1, 18], if more pixel-level ground truth masks for the test video are provided at the test time, the segmentation performance will be progressively improved. However, it is time-consuming to obtain such pixel-level ground truth masks by manual annotation. Therefore, we propose a key frame selection method to choose the high-quality segmentation results in the purified segmentation results, and use them to retrain the segmentation model. Benefiting from the motion based background suppression method and the key frame selection method, the selected key frames have a higher segmentation quality with sufficient foreground information. Finally, by deploying the online fine-tuning strategy with the selected key frames and the manually annotated first frame, the segmentation model is retrained online and is used to generate more accurate segmentation results.

Overall, our main contributions are summarized as follows:

1) We propose a novel framework, which deeply explores motion cue and online fine-tuning strategy, to promote the video object segmentation quality. Our framework consists of three key components including motion based background suppression, key frame selection and model update.

2) We propose a motion based background suppression method, which utilizes motion cue directly to filter out irrelevant background regions and purify the segmentation results.
3) We propose an effective key frame selection method to select the confident frames with high segmentation quality, for fine-tuning segmentation model online.
4) We test our framework on two public video datasets, and the results firmly demonstrate the effectiveness and superiority of our framework.

The rest of this paper is organized as follows. Section 2 reviews existing representative models for video object segmentation. Section 3 details the proposed framework. Experimental results and analysis are presented in Section 4, and conclusions are given in Section 5.

2 Related works

2.1 Unsupervised video segmentation

Unsupervised video segmentation methods aim to segment a primary object without human inputs. Before adopting CNN for video object segmentation, some classic approaches, by minimizing a manually designed energy function to segment objects [13] and utilizing visual saliency [3, 26] and motion cue [3, 10, 11, 14, 26, 28], have achieved good performance. In [13], Märki et al. solved the video object segmentation problem in a bilateral space with the designed energy function. Nagaraja et al. [14] combined sparse user inputs with long-term motion cues and color consistency constrains for video object segmentation. Wang et al. [26] introduced a saliency-aware geodesic distance based method, which considers spatial edges and temporal motion boundaries simultaneously, for video salient object segmentation. Luo et al. [11] constructed a primary object segmentation method based on the complexity awareness of video clips and their segmentation propagation. Koh et al. [10] used both color and motion edges to generate candidate regions for primary object, and proposed the augmentation and reduction processing to segment the primary object in each frame. Based on optical flow and edge cues, Hu et al. [3] developed a novel saliency estimation technique and a novel neighborhood graph to segment objects. Zhang et al. [28] integrated both point trajectories and region trajectories to segment objects in video.

Recently, convolutional neural networks greatly promote the development of video object segmentation. In the CNN-based approaches, motion cue still plays an important role. In [21], the optical flow is treated as the only input to a complicated encoder-decoder network to extract the moving object. Based on [21], Tokmakov et al. [22] introduced a two-stream neural network with an explicit memory module to segment moving objects in unconstrained videos. The two streams of the network are appearance network and motion network, which encode spatial and temporal features, respectively. Similar to [22], Jain et al. [6] combined appearance and motion information to produce pixel-level segmentation masks for all prominent objects. Different from previous methods which use motion cue as extra inputs [6, 21, 22], we exploit the optical flow to suppress the irrelevant background and purify the segmentation results.

2.2 Semi-supervised video segmentation

It is well known that manually labeling a video sequence with pixel-level labels is a time-consuming process. The semi-supervised video segmentation methods only require the pixel-

level labels of the first frame of the video sequence. One common framework for semi-supervised video segmentation is mask propagation. Wang et al. [25] proposed a super-trajectory based video representation to accurately propagate the initial annotations in the first frame to the remaining video frames. Jampani et al. [7] combined the temporal bilateral network and the spatial network to propagate information across video frames. Due to the large error of directly using the optical flow to propagate the labels from previous frames to the current frame [7, 25], Jang et al. [8] developed the convolutional trident network with Markov random field optimization to attenuate the error.

As mentioned above, OSVOS [1] and MSK [18] are two typical CNN-based methods, which sufficiently use online fine-tuning strategy. Voigtlaender et al. [24] updated the pre-trained network online using training samples selected based on the confidence of the network and the spatial configuration. Tsai et al. [23] and Cheng et al. [2] both proposed methods to simultaneously predict pixel-wise object segmentation and optical flow in videos. Based on the idea of matching, Yoon et al. [27] proposed a pixel-level matching based convolutional neural network for video object segmentation. Perazzi et al. [16] formulated the video object segmentation problem as the minimization of an energy function with a fully connected graph of object proposals.

3 Proposed framework

The section details our proposed video object segmentation framework in the following four subsections. Firstly, we first describe an overview of the proposed framework in Section 3.1. Then we present the motion based background suppression method in Section 3.2. In Section 3.3, we give the detailed formulas of the key frame selection method. In the end, we provide model update details of our framework in Section 3.4.

3.1 Architecture overview

The overall architecture of the proposed framework is illustrated in Fig. 1. The video frame $\mathbf{I}_t$ and the previous frame $\mathbf{I}_{t-1}$ are passed to a semi-supervised segmentation model, namely the VGG-16 [20] based OSVOS test model [1], yielding the initial segmentation result $\mathbf{IR}_t$. And they are also sent to an optical flow estimation network i.e. FlowNet [4] to generate the optical flow $\mathbf{F}_t$. We employ the motion based background suppression method to purify $\mathbf{IR}_t$ with $\mathbf{F}_t$

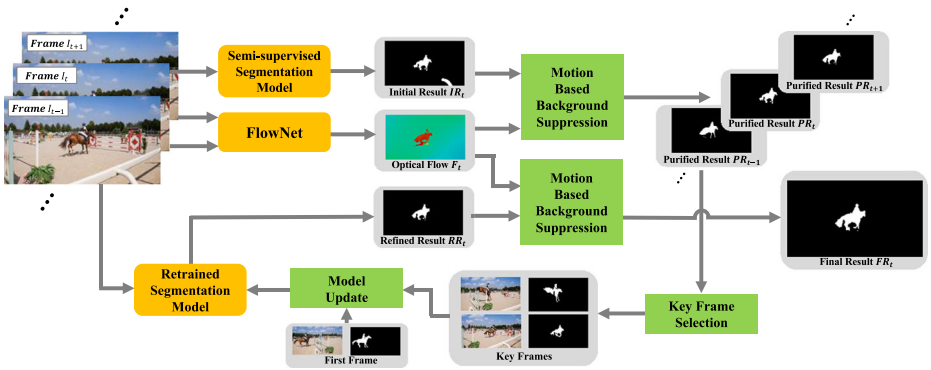

Fig. 1 (Better viewed in color) Illustration of the proposed framework

(Section 3.2), and generate the purified segmentation result $\mathbf{PR}_t$. Then based on the purified segmentation results, the key frame selection method is exploited to select the key frames with high-quality segmentation results (Section 3.3). With the selected key frames and the manually annotated first frame, the segmentation model is updated online (Section 3.4). Finally, the video frame $\mathbf{I}_t$ is fed to the retrained segmentation model, outputting the refined segmentation result $\mathbf{RR}_t$, which is further improved by the motion based background suppression method to obtain the final segmentation result $\mathbf{FR}_t$.

3.2 Motion based background suppression

When testing on a video sequence, the segmentation model may segment out some confusing background (e.g. visually similar background regions) and produces false predictions. Nevertheless, in most videos, the motion discontinuity is generally observed around the object boundaries, and the confusing background regions are usually unconnected with the object boundaries. According to the observation, we exploit the motion cue, i.e. optical flow, and propose a novel motion based background suppression method to filter out the irrelevant background regions. In this way, we can obtain the purified segmentation results. An example of the proposed method is shown in Fig. 2. In the following, we will provide a detailed description of the proposed method.

For the current frame $\mathbf{I}_t$, some background regions, which have similar appearances with the foreground, are also segmented out, such as Fig. 2(b). By contrast, the optical flow $\mathbf{F}_t$ differentiates foreground from background obviously, as shown in Fig. 2(c). By using the contour network in OSVOS [1], the motion boundary probability map of the optical flow can be generated as shown in Fig. 2(d), which shows a rough prediction of the object motion boundaries. Then we use the adaptive thresholding method [15] to generate the binary motion boundary map. And we adopt morphological dilation operation to obtain the dilated motion boundary map as shown in Fig. 2(e), which can better accommodate the foreground boundaries. Successively, based on dilated motion boundary map, we preserve regions connected to the dilated area in the initial result $\mathbf{IR}_t$, and drop those unconnected regions to obtain the purified result $\mathbf{PR}_t$ shown as Fig. 2(f). As a result, the proposed motion based background suppression method filters out the irrelevant background regions effectively and the purified result preserves the foreground regions accurately.

3.3 Key frame selection

To efficiently utilize the online fine-tuning strategy, we establish key frame selection criterion to select the key frames, and combine them with the manually annotated first frame to retrain the segmentation model.

In general, the motion boundary is useful to filter out the irrelevant background regions. However, due to the camera movement, the motion boundary usually mixes with some

Fig. 2 Example of our motion based background suppression method. (a) Current frame $\mathbf{I}_t$. (b) Initial result $\mathbf{IR}_t$. (c) Optical flow $\mathbf{F}_t$. (d) Motion boundary probability map. (e) Dilated motion boundary map. (f) Purified result $\mathbf{PR}_t$

contours of background regions, which will cause some purified results to contain irrelevant background. Therefore, the relatively high-quality purified results usually contain fewer non-connected regions than low-quality ones. Inspired by this, we exploit the region number (RN), which is the number of non-connected regions in each purified result, to remove the low-quality results. We first count RN of each purified result in a video sequence, and calculate the average region number (ARN) over the video sequence. Then, according to ARN, we remove those purified results, with RNs bigger than ARN, since they usually contain background. In this way, the remaining purified results constitute a key frame candidate list (KFCL).

Then, we consider the temporal continuity of the video sequence and exploit the temporal change rate (TCR) of purified result, which can reflect the rate of change in the number of foreground pixels between $\mathbf{PR}_t$ and $\mathbf{PR}_{t-1}$, to further condense KFCL. Specifically, for each $\mathbf{PR}_t$, the temporal change rate TCR_t is defined as

$$TCR_t = \frac{N(\mathbf{PR}_t) - N(\mathbf{PR}_{t-1})}{N(\mathbf{PR}_{t-1})}, \tag{1}$$

where $N(\cdot)$ represents the number of foreground pixels. Correspondingly, if TCR_t changes drastically, it indicates the segmentation result of $\mathbf{PR}_t$ is unreliable. Hence, we first calculate the TCR of each purified result in KFCL, and sort the TCRs in the ascending order. Then we delete those purified results with TCR in the first $\alpha\%$ (a higher shrinkage rate of foreground) and the last $\alpha\%$ (a higher expansion rate of foreground) from KFCL.

Besides, a high-quality segmentation result usually concentrates its foreground pixels in a compact region. Similar to the saliency compactness [12, 29], we calculate the segmentation compactness (SC) to evaluate the quality of each purified result $\mathbf{PR}_t$ as follows:

$$SC_t = \frac{N(\mathbf{PR}_t)}{A_t}, \tag{2}$$

where A_t is the area of the smallest bounding box, which covers the whole foreground pixels in $\mathbf{PR}_t$. We calculate the segmentation compactness for each purified result in KFCL, and sort all segmentation compactness values in the ascending order. Considering that KFCL has been condensed based on temporal change rate, we only delete from KFCL relatively fewer purified results with segmentation compactness in the first $\beta\%$. The remaining purified results in KFCL are used as the final key frame candidates.

Finally, we select the key frames, which are exploited to update the segmentation model, from the final key frame candidates. In a video, the closer the two video frames are, the smaller the difference between them will be. In order to learn different appearance information about foreground over the whole video, the temporal distance between key frames is important. The number of key frames i.e. *Num* is first determined based on the length of all purified results, T, as follows:

$$Num = \begin{cases} 1, & T \le d \\ 2, & d < T \le 2d \\ & \vdots \\ n, & T > (n-1)d \end{cases}, \tag{3}$$

where d and n are the parameters to determine *Num*, and these two parameters will be analyzed in Section 4.3. The temporal indices of possible key frames are then determined by uniform sampling on the purified results sequence. For example, in the case of $Num = 3$, the temporal indices of possible key frames are set to $[T/Num]$, $[2T/Num]$ and $[3T/Num]$ where $[\cdot]$ is the round operation.

Finally, a total of *Num* frames, which are the nearest to the above temporal indices, are selected from KFCL to constitute the key frames for updating the segmentation model.

3.4 Model update

As demonstrated in [1, 18, 24], the online fine-tuning strategy can improve the video object segmentation performance significantly. With the increase of pixel-level object masks of the test video in the test phase, the performance will be further improved. As aforementioned, OSVOS [1] contains three stages including Base Network, Parent Network and Test Network. We follow the first two phases of OSVOS and make changes during the testing phase, i.e. Test Network. As for the generation of Test Network, OSVOS only fine-tunes the Parent Network with the manually annotated first frame online. Differently, we retrain the Parent Network online using two kinds of samples including the selected key frames with high-quality segmentation results and the manually annotated first frame. Though for some videos, the segmentation results of the selected key frames are not sufficiently accurate, they can still provide the detailed and enough foreground information to perform online fine-tuning. Notably, the segmentation results of the selected key frames are useful for some challenging scenes such as deformation, fast motion and appearance change.

We use the stochastic gradient descent (SGD) with momentum 0.9 for 560 iterations, and the learning rate is set to 10^{-8} at the stage of model update. The data augmentation operations including conventional mirroring and resizing are only performed on the first frame, and no data augmentation operations are performed on the selected key frames. Specifically, we first train the segmentation model using the key frames for 60 iterations, and then train using with the addition of the manually annotated first frame for 500 iterations. This training strategy achieves stable results at test time. Considering the randomness of online fine-tuning, we conduct the retraining process four runs and report the corresponding mean in the experimental results of Section 4.

4 Experimental results

4.1 Datasets

We evaluate the performance of the proposed framework on two public datasets, i.e. DAVIS [17] and Youtube-Objects [5, 19]. DAVIS consists of 50 full-HD video sequences (30 for training and 20 for validation) with 3455 annotated frames. Youtube-Object includes 126 videos with more than 20,000 frames for 10 objects categories.

4.2 Evaluation metric

To evaluate the segmentation performance, we adopt the standard evaluation metrics [17], namely region similarity, contour accuracy and temporal stability.

Region Similarity $\mathcal{J}$. The region similarity is also called intersection-over-union (IoU), which has been widely used for evaluating the quality of the predicted foreground segmentation mask M with the reference of the ground truth mask G. It is defined as follows:

$$\mathcal{J} = \frac{|M \cap G|}{|M \cup G|}. \tag{4}$$

Contour Accuracy $\mathcal{F}$. The predicted foreground segmentation mask M can be treated as a lot of closed contour regions $c(M)$. And the contour-based precision and recall P_c and R_c between the contour $c(M)$ and $c(G)$ are used to compute the contour-based F-measure as follows:

$$\mathcal{F} = \frac{2P_c R_c}{P_c + R_c}. \tag{5}$$

Temporal Stability $\mathcal{T}$. The temporal stability of results is an important aspect in video object segmentation. And it is measured by computing the distance between the shape context descriptors that describe the shape of the boundary of the segmentations between two successive frames. Intuitively, temporal stability measures the turbulence and inaccuracy of the contours.

4.3 Parameters analysis

As described in Section 3.3, the parameters, α, β, d and n, together determine how to select key frames. In this section, we make a detailed analysis for these parameters and select the optimal parameters via experiments on DAVIS dataset.

First, we choose the values for α and β from {7.5, 15, 22.5, 30} to determine the final key frame candidates. Then, we can determine the number of key frames, i.e. *Num*, using Eq. 3. But Eq. 3 is related to d and n, we choose a value for d from {20, 30, 40}, and n can be determined by d as $n = 2, \cdots, \left\lceil \frac{T_{max}}{d} \right\rceil$. The $\lceil \cdot \rceil$ is the ceil operation and T_{max} is the maximum length of the purified results sequence in DAVIS, which is 103. For example, if d is 30, n can be determined to be 2, 3 and 4. Then Eq. 3 has the three forms, i.e. $Num = \begin{cases} 1, T \leq 30 \\ 2, T > 30 \end{cases}$, $Num = \begin{cases} 1, T \leq 30 \\ 2, 30 < T \leq 60 \\ 3, T > 60 \end{cases}$ and $Num = \begin{cases} 1, T \leq 30 \\ 2, 30 < T \leq 60 \\ 3, 60 < T \leq 90 \\ 4, T > 90 \end{cases}$. After selecting the key frames, we use them to update the model with the manually annotated first frame, and test the video sequence. Based on each set of parameters α, β, d and n, we test four times on DAVIS and report the mean in Table 1 with mean $\mathcal{J}$.

As mentioned above, we first remove some purified results of the first $\alpha\%$ and the last $\alpha\%$ of TCR in KFCL. Then according to SC, we only delete the first $\beta\%$ purified results in KFCL. We can see from Table 1 that when α is fixed, a larger mean $\mathcal{J}$ appears if β is a small value like 7.5 or 15; and when β is fixed, a larger mean $\mathcal{J}$ appears if α is a small value like 7.5 or 15. The reason behind this is that when both α and β are relatively large values like 22.5 or 30, the KFCL will lose more high-quality candidate frames, which will result in the lack of object information in the model update phase. When d is fixed, we can obtain a good performance if n is set to 3. This is because with a large value of n, the information provided by the key frames has more background information than object information, while with a small value of n, the key frames cannot provide sufficient object information. Therefore, α, β, d and n are set to 15, 7.5, 30 and 3, respectively. And based on this set of parameters, our model achieves the best performance.

In addition, on the basis of the optimal parameters, we do not remove the purified results with RNs larger than ARN, and the mean $\mathcal{J}$ is 81.6%, which is 1.8% lower than that of the proposed framework. Thus, we can conclude that the strategy of removing the purified results with RNs larger than ARN is valid.

Table 1 Parameters analysis on DAVIS validation set with Mean $\mathcal{J}$

α	β	d									
		20					30			40	
		n					n			n	
		2	3	4	5	6	2	3	4	2	3
7.5	7.5	82.0	82.8	82.7	82.8	82.8	82.0	82.2	81.9	81.7	82.2
7.5	15	82.1	82.4	82.2	82.4	82.3	82.1	81.9	81.5	82.1	82.2
7.5	22.5	81.8	82.4	81.9	82.2	82.1	81.8	82.0	81.6	81.6	82.0
7.5	30	81.7	82.6	82.1	82.4	82.4	81.7	82.2	81.8	81.6	82.2
15	7.5	82.5	83.3	82.4	82.9	82.8	82.5	83.4	82.7	82.2	83.0
15	15	81.8	82.1	81.6	81.9	81.9	81.8	82.1	81.7	81.3	81.9
15	22.5	81.4	82.1	81.4	81.6	81.6	81.4	82.0	81.6	81.3	81.9
15	30	81.4	81.6	80.9	81.4	81.3	81.4	81.8	81.4	81.1	81.6
22.5	7.5	81.6	82.3	82.1	82.4	82.3	81.6	81.9	81.6	81.4	82.0
22.5	15	81.5	81.9	81.5	81.7	81.7	81.5	81.7	81.4	81.9	82.4
22.5	22.5	81.5	82.0	81.1	81.7	81.6	81.5	82.1	81.7	81.6	82.2
22.5	30	80.7	81.8	81.0	81.3	81.3	80.7	81.4	81.0	80.7	81.5
30	7.5	80.7	81.7	81.2	81.5	81.5	80.7	81.0	80.6	80.7	81.0
30	15	81.1	81.8	81.5	81.6	81.5	81.1	81.3	80.9	80.7	80.8
30	22.5	81.4	81.9	81.5	81.8	81.8	81.4	81.4	81.1	81.3	81.4
30	30	81.6	81.9	81.8	82.1	82.0	81.6	81.8	81.3	81.2	81.4

4.4 Ablation study

Table 2 summarizes the contributions of motion based background suppression (MBBS) and model update with the selected key frames to the segmentation performance. The baseline in Table 2 is OSVOS, which achieves 79.8% in mean $\mathcal{J}$. Our MBBS effectively brings 1.5% improvement over OSVOS in mean $\mathcal{J}$. After efficient online fine-tuning with the purified results of the selected key frames and the manually annotated first frame, i.e. Model Update in Table 2, our retrained segmentation model reaches 82.9%, which is 3.1% higher than the baseline. This shows the priority of our model update method. We can find that the MBBS and Model Update effectively improve per-sequence performance on DAVIS validation set in Table 3. Besides, we fine-tune online with the pixel-level ground truths of the selected key frames (Model Update with GT in Table 2), and the mean $\mathcal{J}$ reaches 83.5%, which is the upper-bound of fine-tuning in our case and is similar to our 82.9%. In addition, we also retrained the segmentation model with the key frames by uniform sampling (Model Update with Uniform Sampling in Table 2), and achieves 81.4%, which is 1.5% lower than our 82.9%. It clearly demonstrates the effectiveness of our key frame selection method. Finally, by

Table 2 Ablation study on DAVIS validation set. MBBS denotes the motion based background suppression

Method	DAVIS, Mean $\mathcal{J}\uparrow$
OSVOS (Baseline)	79.8
+ MBBS	81.3
Model Update	82.9
Model Update with GT	83.5
Model Update with Uniform Sampling	81.4
Model Update + MBBS (Ours)	83.4

deploying the MBBS on the results of Model Update, our complete framework brings another 0.5% improvement. In Table 3, we can find that our method performs better than the baseline OSVOS on most sequences. In addition, our method is based on temporal changes, and we analyze the performance of our method on Fast-Motion video sequences. In DAVIS validation set, the Fast-Motion videos contain *bmx-trees*, *breakdance*, *dog*, *drift-chicane*, *drift-straight*, *motocross-jump* and *parkour*. According to Table 3, we calculate to obtain that our method is 80.2% in terms of mean $\mathcal{J}$ on Fast-Motion videos, and in contrast, OSVOS only achieves 76.5%. This demonstrates that our method is more effective for Fast-Motion videos.

4.5 Performance comparison

We compare our framework with nine semi-supervised methods including OnAVOS [24], OSVOS [1], MSK [18], SegFlow [2], CTN [8], VPN [7], PLM [27], OFL [23], FCP [16] and six unsupervised methods including LVO [22], ARP [10], FSEG [6], MPNet [21], CALP [11],

Table 3 Ablation study of per-sequence results on DAVIS validation set. The **red** is the best, and the **blue** is the second best

Sequence	Method, $\mathcal{J}$ ↑			
	OSVOS	+MBBS	Model Update	Ours
blackswan	**94.2**	**94.2**	93.9	93.9
bmx-trees	55.5	**59.4**	54.8	**57.9**
breakdance	70.8	70.8	**75.3**	**75.3**
camel	85.1	92.1	**94.9**	**94.9**
car-roundabout	95.3	96.1	**96.5**	**96.7**
car-shawdow	93.7	**94.7**	94.0	**94.0**
cows	**94.6**	**94.6**	93.5	93.5
dance-twirl	67.0	67.4	**79.6**	**79.6**
dog	**90.7**	**90.7**	83.6	85.1
drift-chicane	83.5	83.5	**85.0**	**86.4**
drift-straight	67.6	67.6	**85.4**	**85.5**
goat	**88.0**	**88.0**	87.0	87.0
horsejump-high	78.0	**86.7**	85.9	**85.9**
kite-surf	**68.6**	**69.1**	66.9	67.3
libby	80.8	**84.9**	83.7	**83.9**
motocross-jump	**81.6**	**82.4**	79.4	81.3
paragliding	62.5	63.1	**63.1**	**63.2**
parkour	85.6	88.4	**89.4**	**89.6**
scooter-black	71.1	71.4	**78.4**	**78.5**
soapbox	81.2	81.3	**87.5**	**87.5**
Mean	79.8	81.3	**82.9**	**83.4**

Table 4 Quantitative comparison of semi-supervised and unsupervised methods on DAVIS validation set. The **red** is the best, and the **blue** is the second best in each category

		Semi-supervised									Unsupervised					
	Metric	Ours	OnAVOS	OSVOS	MSK	SFL	CTN	VPN	OFL	FCP	ARP	LVO	FSEG	MPNet	CALP	SAGE
$\mathcal{J}$	Mean $\mathcal{M}\uparrow$	**83.4**	**83.2**	79.8	79.7	76.1	73.5	70.2	68.0	58.4	**76.2**	**75.9**	70.7	70.0	40.3	34.9
	Recall $\mathcal{O}\uparrow$	**95.9**	**95.5**	93.6	93.1	90.6	87.4	82.3	75.6	71.5	**91.1**	**89.1**	83.5	85.0	40.8	18.8
	Decay $\mathcal{D}\downarrow$	6.2	**5.0**	14.9	8.9	12.1	15.6	12.4	26.4	**-2.0**	7.0	**0.0**	1.5	**1.3**	10.8	7.7
$\mathcal{F}$	Mean $\mathcal{M}\uparrow$	**82.5**	**85.1**	80.6	75.4	76.0	69.3	65.5	63.4	49.2	**70.6**	**72.1**	65.3	65.9	33.9	37.9
	Recall $\mathcal{O}\uparrow$	**93.3**	**92.8**	92.6	87.1	85.5	79.6	69.0	70.4	49.5	**83.5**	**83.4**	73.8	79.2	28.5	17.4
	Decay $\mathcal{D}\downarrow$	7.2	**6.0**	15.0	9.0	10.4	12.9	14.4	27.2	**-1.1**	7.9	**1.3**	**1.8**	2.5	9.5	4.8
$\mathcal{T}$	Mean $\mathcal{M}\downarrow$	33.0	22.1	37.6	**21.8**	**18.9**	21.3	32.4	22.2	30.6	39.3	**26.5**	**32.8**	57.2	42.4	73.3

SAGE [26]. We also compare with non-deep learning based methods including JFS [14] and BVS [13].

4.5.1 Quantitative comparison

For a quantitative comparison, Table 4 shows that the proposed framework achieves the best performance compared to other state-of-the-art video object segmentation methods on mean $\mathcal{J}$. It can be seen that our framework improves OSVOS by 3.6%. Compared with OnAVOS which does not apply test-time augmentation and post processing, ours is 0.2% higher in terms of mean $\mathcal{J}$. Table 5 reports per-category mean $\mathcal{J}$ on the Youtube-Objects dataset [5, 19]. Our framework achieves the best performance and improves OSVOS by 1.3%, and our method is 0.3% higher than OnAVOS.

4.5.2 Qualitative comparison

Figure 3 shows some segmentation results of several videos from DAVIS [17] and Youtube-Objects [5, 19]. In Fig. 3, the odd rows show the segmentation results of OSVOS (blue masks), and the even rows show the segmentation results (green masks) generated using our framework. These examples feature typical challenges in video sequences, e.g., object deformation, scale variation, fast motion, appearance change and shape complexity. Compared with OSVOS results, our segmentation results are more accurate and more complete. Specially, the results of a typical Fast-Motion video, *drift-chicane*, are shown in the third and fourth rows of Fig. 3. It can be found that OSVOS lose the car in the last several frames, while our method can still segment the car accurately.

4.6 Computational cost

Our framework is implemented by the publicly available Caffe library [9]. All the experiments and analyses are conducted on a workstation with a Nvidia Titan X (Pascal) GPU and an Intel

Table 5 Quantitative comparison of per-category region similarity $\mathcal{J}$ on Youtube-Objects dataset

Metric	Ours	OnAVOS	OSVOS	MSK	OFL	JFS	BVS	CALP	SAGE
Mean $\mathcal{J}\uparrow$	79.6	79.3	78.3	77.7	77.6	74.0	68.0	36.3	35.8

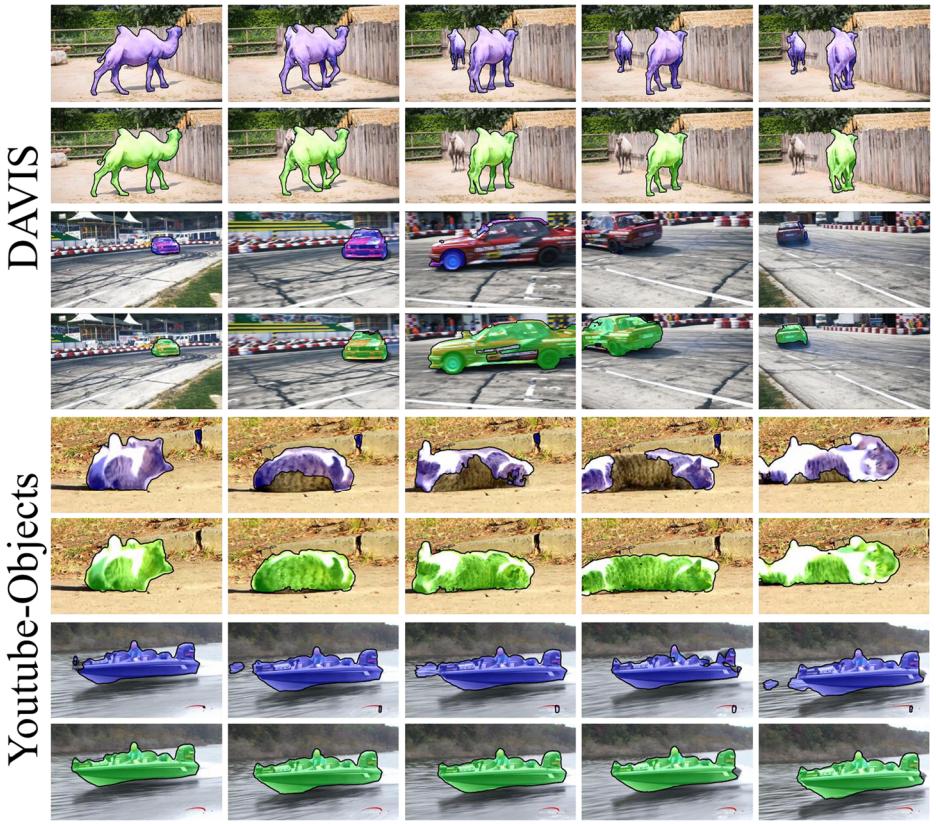

Fig. 3 Segmentation results of some videos from DAVIS and Youtube-Objects. The segmentation of OSVOS are blue masks (odd rows), and our segmentation results are green masks (even rows)

Core i7-6700K CPU 4.0GHZ with 16 GB RAM. It takes around 9.87 s to process a video frame with a resolution of 480×854. Specifically, the initial segmentation including the optical flow estimation takes 5.45 s, the motion based background suppression takes 0.24 s (Section 3.2), the key frame selection takes 0.18 s (Section 3.3), and the model update with the second motion based background suppression takes 4.00 s (Section 3.4).

5 Conclusion

This paper proposes a novel video object segmentation framework by exploiting the motion cue and the online fine-tuning strategy. Firstly, the motion based background suppression method can purify the foreground clearly using motion boundary. Secondly, the key frame selection method provides confident samples for the model update process. Lastly, the model update module effectively improves the segmentation results using online fine-tuning strategy with the selected key frames and the manually annotated first frame. Experimental results on two challenging datasets have demonstrated the effectiveness of our framework.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No. 61771301.

References

1. Caelles S, Maninis K, Pont-Tuset J, Leal-Taixé L, Cremers D, Van-Gool L (2017). One-shot video object segmentation. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 5320–5329
2. Cheng J, Tsai Y, Wang S, and Yang M (2017). Segflow: joint learning for video object segmentation and optical flow. In: Proc. of IEEE international conference on computer vision, pp. 686–695
3. Hu YT, Huang JB, Schwing AG (2018) Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In: Proc. of European conference on computer vision
4. Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, and Brox T (2017). Flownet 2.0: evolution of optical flow estimation with deep networks. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 1647–1655
5. Jain SD and Grauman K (2014). Supervoxel-consistent foreground propagation in video. In: Proc. of European conference on computer vision, pp. 656–671.
6. Jain SD, Xiong B, Grauman K (2017) Fusionseg: learning to combine motion and appearance for automatic segmenation. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 2117–2126
7. Jampani V, Gadde R, Gehler PV (2017) Video propagation networks. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 3154–3164
8. Jang W, Kim C (2017) Online video object segmentation via convolutional trident network. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 7474–7483
9. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. ACM international conference on Multimedia:675–678
10. Koh YJ, Kim C (2017) Primary object segmentation in videos based on region augmentation and reduction. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 7417–7425
11. Luo B, Li H, Meng F, Wu Q, Ngan KN (2018) An unsupervised method to extract video object via complexity awareness and object local parts. IEEE Trans on Circuits and Syst for Video Tech 28(7):1580–1594
12. Mai L and Liu F (2014). Comparing salient object detection results without ground truth. In: Proc. of European conference on computer vision, pp. 76–91
13. Märki N, Perazzi F, Wang O, and Sorkine-Hornung A (2016) Bilateral space video segmentation. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 743–751
14. Nagaraja NS, Schmidt FR, Brox T (2015) Video segmentation with just a few strokes. In: Proc. of IEEE international conference on computer vision, pp. 3235–3243
15. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
16. Perazzi F, Wang O, Gross M, Sorkine-Hornung A (2015). Fully connected object proposals for video segmentation. In: Proc. of IEEE international conference on computer vision, pp. 3227–3234.
17. Perazzi F, Pont-Tuset J, McWilliams B, Van-Gool L, Gross M, and Sorkine-Hornung A (2016). A benchmark dataset and evaluation methodology for video object segmentation. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 724–732
18. Perazzi F, Khoreva A, Benenson R, Schiele B, SorkineHornung A (2017) Learning video object segmentation from static images. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 3491–3500
19. Prest A, Leistner C, Civera J, Schmid C, and Ferrari V (2012). Learning object class detectors from weakly annotated video. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 3282–3289
20. Simonyan K, Zisserman A (2014). Very deep convolutional networks for large-scale image recognition. Computer Science
21. Tokmakov P, Alahari K, and Schmid C (2017). Learning motion patterns in videos. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 531–539
22. Tokmakov P, Alahari K, Schmid C (2017). Learning video object segmentation with visual memory. In: Proc. of IEEE international conference on computer vision, pp. 4491–4500
23. Tsai Y, Yang M, Black MJ (2016) Video segmentation via object flow. In: Proc. of IEEE conference on computer vision and pattern recognition, pp. 3899–3908
24. Voigtlaender P, Leibe B (2017). Online adaptation of convolutional neural networks for video object segmentation. In: Proc. of the British Machine Vision Conference
25. Wang W, Shen J, Xie J, Porikli F (2017) Super-trajectory for video segmentation. In: Proc. of IEEE international conference on computer vision, pp. 1680–1688
26. Wang W, Shen J, Yang R, Porikli F (2018) Saliency-aware video object segmentation. IEEE Trans on Pattern Anal Mach Intell 40(1):20–33
27. Yoon JS, Rameau F, Kim J, Lee S, Shin S, Kweon I. S. (2017). Pixel-level matching for video object segmentation using convolutional neural networks. In: Proc. of IEEE international conference on computer vision, pp. 2186–2195

28. Zhang G, Yuan Z, Liu Y, Ma L, Zheng N (2015) Video object segmentation by integrating trajectories from points and regions. Multimed Tools Appl 74(21):9665–9696
29. Zhou X, Liu Z, Sun G, Wang X (2017) Adaptive saliency fusion based on quality assessment. Multimed Tools Appl 76(22):23187–23211

Gongyang Li received the B.E. degree from Shanghai Normal University, Shanghai, China, in 2016. He is currently pursuing the Ph.D. degree at the School of Communication and Information Engineering, Shanghai University, Shanghai, China. His research interests include image/video object segmentation and saliency detection.

Zhi Liu received the B.E. and M.E. degrees from Tianjin University, Tianjin, China, and the Ph.D. degree from Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai, China, in 1999, 2002, and 2005, respectively. He is currently a Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai, China. From Aug. 2012 to Aug. 2014, he was a Visiting Researcher with the SIROCCO Team, IRISA/INRIA-Rennes, France, with the support by EU FP7 Marie Curie Actions. He has published more than 180 refereed technical papers in international journals and conferences. His research interests include image/video processing, machine learning, computer vision and multimedia communication. He was a TPC member/session chair in ICIP 2017, PCM 2016, VCIP 2016, ICME 2014, WIAMIS 2013, etc. He co-organized special sessions on visual attention, saliency models, and applications at WIAMIS 2013 and ICME 2014. He is an area editor of *Signal Processing: Image Communication* and served as a guest editor for the special issue on *Recent Advances in Saliency Models, Applications and Evaluations* in *Signal Processing: Image Communication*. He is a senior member of IEEE.

Xiaofei Zhou received the B.E. degree from Anhui Polytechnic University, Wuhu, China, in 2012, and the M.E. and Ph.D. degrees from Shanghai University, Shanghai, China, in 2015 and 2018, respectively. He is currently a Lecturer with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou, China. His research interests include saliency detection and image/video segmentation.