# Motion-Guided Cascaded Refinement Network for Video Object Segmentation

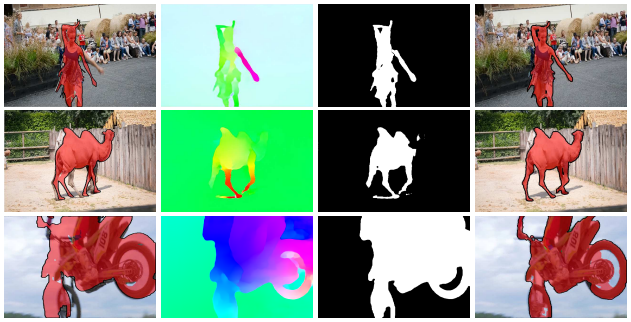
Ping Hu[†] Gang Wang[‡] Xiangfei Kong[†] Jason Kuen[†] Yap-Peng Tan[†]
[†]Nanyang Technological University [‡]Alibaba AI Labs
{phu005, xfkong, jkuen001, eyptan}@ntu.edu.sg, gangwang6@gmail.com

## Abstract

*Deep CNNs have achieved superior performance in many tasks of computer vision and image understanding. However, it is still difficult to effectively apply deep CNNs to video object segmentation(VOS) since treating video frames as separate and static will lose the information hidden in motion. To tackle this problem, we propose a Motion-guided Cascaded Refinement Network for VOS. By assuming the object motion is normally different from the background motion, for a video frame we first apply an active contour model on optical flow to coarsely segment objects of interest. Then, the proposed Cascaded Refinement Network(CRN) takes the coarse segmentation as guidance to generate an accurate segmentation of full resolution. In this way, the motion information and the deep CNNs can well complement each other to accurately segment objects from video frames. Furthermore, in CRN we introduce a Single-channel Residual Attention Module to incorporate the coarse segmentation map as attention, making our network effective and efficient in both training and testing. We perform experiments on the popular benchmarks and the results show that our method achieves state-of-the-art performance at a much faster speed.*

## 1. Introduction

Video object segmentation (VOS) is an important problem in computer vision, since it benefits other tasks like object tracking [72], video retrieval [26], activity recognition [20], video editing [38] and so on. Due to the strong spatiotemporal correlation between consecutive video frames, motion plays a key role in many state-of-the-art methods for video object segmentation [61, 68, 1, 62, 36, 15]. Motion estimations like optical flow [2, 27, 25] and pixel trajectory [52, 57] reveal the pixel correspondence between frames and enable the propagation of foreground/background labels from one frame to the next. Furthermore, motion contains rich spatiotemporal structure information which can benefit the segmentation of moving objects. However, motion estimation itself is still a very difficult task and often produces inaccurate results. Some common situations like noise, blurring, deformation, and occlusion can further exacerbate the difficulty.



(a) (b) (c) (d)

Figure 1. Examples of our method. (a) Input frame and the initial active contour. (b) Optical flow. (c) Segmentation by evolving the active contour on (b). (d) Final results with (c) as guidance.

Different from previous methods which mainly rely on motion, recent attempts based on CNNs[63, 8, 3, 54, 59, 30, 45, 11, 19] tackle the problem of VOS by learning. Due to the powerful learning ability and the large amounts of training data, deep CNNs have achieved very good performance in static image segmentation [7, 39]. While for VOS, the annotated training data is lacking and treating frames as static will lose the information hidden in motion. It has been shown in [3, 63] that after finetuning on the first frame, deep CNNs can "recognize" the object with similar appearance from subsequent frames. However, only relying on "memorizing" the appearance of the target object may suffer from several limitations. For example, the object's appearance may change along with the time, and objects in the background may share similar appearance to the target object.

To utilize the spatiotemporal structure information hidden in motion and the superior learning ability of CNNs, in this paper we propose a motion-guided cascaded refinement network for video object segmentation. The proposed method is composed of two parts: optical flow-based moving object segmentation and *Cascaded Refinement Network* (CRN). Specifically, for an input frame, a coarse segmenta-

tion of the target object is first extracted from optical flow. The CRN then takes the coarse segmentation as guidance and outputs an accurate segmentation.

To generate the coarse segmentation, which provides information about coarse shape and location of the target object, we apply the active contour [35, 5, 6] to segment the optical flow estimated by [27]. Active contour is a classical tool for image segmentation and works by finding the optimal segmentation that maximises the homogeneity in foreground region and background region respectively. Since the target object normally has a different motion pattern from background regions, we apply the active contour to segment the optical flow. Furthermore, with a proper initialization, active contour model can converge very efficiently. At each time frame, we first compute optical flow between the current frame and next one, and then initialize an active contour on the optical flow image using the final segmentation result of the last frame. After iteratively evolving the active contour, we can obtain a coarse segmentation of the target object. Examples are shown in Fig. 1(a)-(c).

Given the coarse segmentation, we propose a *Cascaded Refinement Network* that takes as guidance the coarse map to generate an accurate segmentation (Fig. 1(d)). In the *Cascaded Refinement Network*, the guidance map serves as a priori knowledge of the target object to help the network to focus on object regions and ignore background regions, thus benefit both training and testing. Furthermore, since the *Cascaded Refinement Network* tackles a problem of segmentation in static images, we are able to effectively train it using datasets for other tasks like instance segmentation [12]. Experimental results on benchmark datasets validate the effectiveness and efficiency of our method. In summary, we make the following contributions: (1) We propose a optical flow-based active contour model that can effectively and efficiently segment moving objects from video. (2) Our *Cascaded Refinement Network*(CRN) is effective and efficient in both training and testing. In CRN, we propose a *Single-channel Residual Attention Module* that effectively utilizes the guidance map as attention so as to help CRN to focus on regions of interest and ease the burden of training and model size. (3) Our method achieves state-of-the-art performance on three benchmarks. On the DAVIS dataset [46], we achieve mIOU of $84.4\%$ at 0.73 second/frame for semi-supervised task, and $76.4\%$ at 0.36 second/frame for unsupervised task, outperforming the current methods without post processing at a much faster speed.

## 2. Related Work

**Video Object Segmentation (VOS).** Due to today's need of automatically processing the huge amount of video data, related research works in recent years mainly focus on unsupervised methods and semi-supervised methods for VOS. Unsupervised methods [51, 43, 44, 67, 58] assume no manually annotation about the target objects. In order to automatically identify primary objects in a video, cues like motion [73, 40, 13], object proposals [40, 36, 14, 70, 47], and saliency [67, 73] are utilized. In [44, 73] the authors first locate moving objects via motion boundaries and then segment the object region with appearance-based models. The recurrence of objects and the coherence of the appearance are considered in [16, 18, 31] to segment primary objects from frames across the video. Semi-supervised approaches [66, 10, 47, 42, 29] accept target objects identified by user at the first frame and then segment the objects from subsequent frames. To propagate the labels, dense point trajectory is adopted in [68, 1]. Graphs are defined on superpixels locally [69, 61] or globally [65, 47] to efficiently propagate labels in spatiotemporal space. Based on the bilateral formulation, segmentation is performed in bilateral space [42] and bilateral networks [29] are trained to propagate more general information.

Recently, deep learning based methods[63, 41, 8, 3, 54, 59, 58, 30, 45, 11, 54, 33] have advanced the state-of-the-art performance for VOS. These methods can be grouped into two types based on whether motion information is used. One class of methods train network to incorporate motion information explicitly [33, 11, 59, 45, 58] or implicitly [8]. Although motion contributes to the performance in these methods, directly applying networks to extract target object from motion may be suboptimal due to the lack of training data and the quality of optical flow estimation. The other category of methods ignores motion information and only relies on appearance learning [63, 3] or matching [54]. By driving the network to "memorize" the appearance of the target object, this type of models can achieve state-of-the-art performance. However, these methods are still limited by object deformation, interference of background objects, and the time-consuming training process. Different from these methods, we first coarsely extract the object's segmentation from motion, then apply the *Cascaded Refinement Network* to refine the coarse segmentation into an accurate one. Since both components of our method can work effectively and complementarily, we achieves state-of-the-art performance at a much faster speed.

**Active Contour.** Active contour [32] is a classical model for segmentation. It detects object regions by iteratively evolving a curve under constraints from the given image. Due to its efficiency and advantages [9], active contour has been widely used in image segmentation [5, 35, 24, 71, 49, 74] and tracking [23, 53, 56]. In general, there are two types of active contours: edge-based models and region-based models. Edge-based models [4, 35] utilize image gradient and converge to objects boundaries. However, these methods are sensitive to initial state and may fail when object boundaries are weak. Region-based models [5, 60, 6] focus

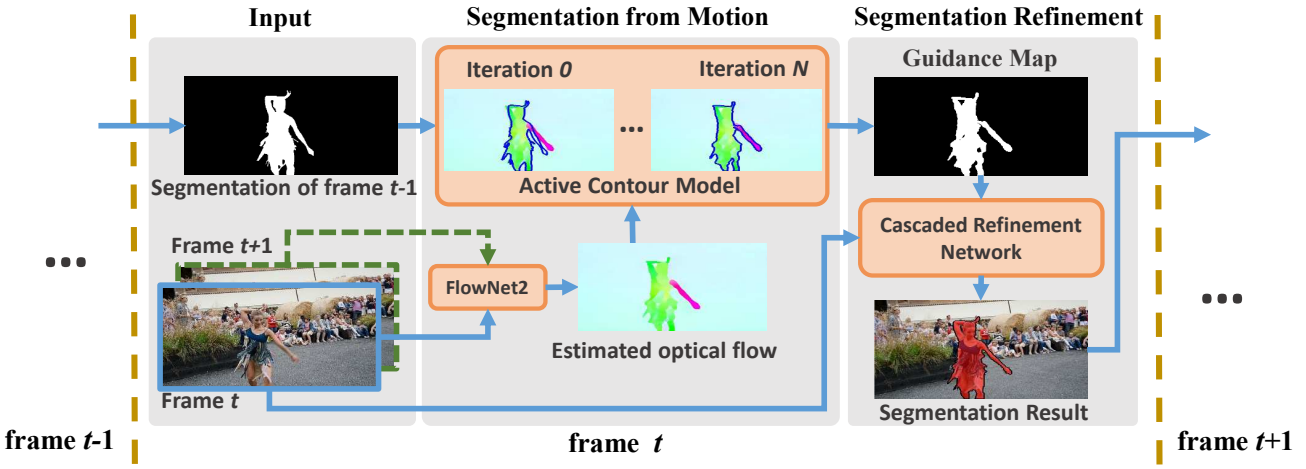



Figure 2. An overview of the proposed method. For each frame, optical flow [27] is estimated at first. Then we use the segmentation result of the last frame to initialize an active contour (shown as the blue curve in Active Contour Model ) on the optical flow, and evolve it $N$ steps to minimize an energy function so that to coarsely segment the object. Finally, the coarse object mask is used as guidance to help the Cascaded Refinement Network to accurately segment the target object instance. To begin this process, user annotation is used to initialize $frame\ 0$ in semi-supervised VOS, and a predefined rectangle is used to initialize $frame\ 0$ in unsupervised VOS.

on region homogeneity rather than gradient, and therefore work better for situations like weak boundary and less sensitive to the initial state. In this work, we build our method based on the region-based model [5].

## 3. Method

An overview of the proposed algorithm is shown in Fig. 2. Video frames are processed sequentially. For each frame, we first segment the target object from optical flow, then apply the Cascaded Refinement Network(CRN) to produce an accurate result.

### 3.1. Object Segmentation from Motion

In the task of VOS, extracting the spatiotemporal structure information hidden in motion [44, 73, 55] is popular but difficult for situations like inaccurate motion estimation and static objects. To make better use of motion information, we propose to apply the active contour model on optical flow. In videos, objects of interests normally have different motion patterns from the background. This makes region-based active contour [5, 6] models, which segment images by maximizing the homogeneity within each of the segmented regions, suitable for video object segmentation. In this section, we first introduce how to formulate the active contour using level set, and then present the active contour model for segmenting objects from optical flow.

#### 3.1.1 Level Set Formulation for Active Contour

Level Set is a tool for implementing active contours [5, 6]. Given a 2D pixel domain $\Omega$, a curve $C$ is defined as the boundary of an open subset $\omega$ of the 2D pixel space $\Omega$ (i.e. $\omega \in \Omega$, $C = \partial\omega$). Subsequently, the image is segmented into two subregions: region $\omega$ denoted by $inside(C)$ and region $\Omega \setminus \omega$ denoted by $outside(C)$. With level set formulation, the curve $C$ can be represented by the zero level set of a Lipschitz function $\phi : \Omega \to \mathbb{R}$ such that,

$$\begin{cases} C = \{(x, y) \in \Omega : \phi(x, y) = 0\}, \\ inside(C) = \{(x, y) \in \Omega : \phi(x, y) > 0\}, \\ outside(C) = \{(x, y) \in \Omega : \phi(x, y) < 0\}, \end{cases} \tag{1}$$

With this formulation, evolving the curve $C$ on the image can be achieved by gradually changing the value of the level set function $\phi(\cdot)$.

Since the sign of $\phi(\cdot)$ indicates the whether a pixel or inside or outside the contour, $\phi(\cdot)$ can be converted into the binary foreground/background labels via a Heaviside step function $H(\phi)$, which projects nonnegative input to 1 and negative input to 0. In practice, to avoid local minima, an approximated version of the Heaviside Function is used,

$$H_\varepsilon(z) = \frac{1}{2}(1 + \frac{2}{\pi}\arctan(\frac{z}{\varepsilon})), \quad \delta_\varepsilon = \frac{\partial H_\varepsilon(z)}{\partial z} = \frac{1}{\pi} \cdot \frac{\varepsilon}{\varepsilon^2 + z^2} \tag{2}$$

#### 3.1.2 Applying Active Contour on Optical Flow

To our best knowledge, this is the first attempt to apply active contour model on optical flow for moving object segmentation. Given a frame $t$, we begin by estimating optical flow between frame pairs of $(t, t + 1)$ with the state-of-the-art approach FlowNet2 [27], which runs efficiently and is sensitive to objects as well as motion boundaries (Fig. 1(b)). Since the original 2-dimensional optical flow has a relatively narrow range of values, we convert the optical flow into a color image [1] and apply the active contour model on it.

[1] Expressing the orientation and the magnitude of the vector by varying hue and saturation.

Given an image and an initial contour on it, at first an initial level set function is defined by computing the signed distance between pixels and the initial contour, then the level set function is iteratively updated to minimizing an energy defined on the image. Traditionally, the energy function is composed of two parts [5, 6]. One is the geometry constraints that control the shape of the contour according Gestalt Principle of Simplicity. The other is a data term that forces the divided subregions to be smooth and homogenous. In our method, we empirically found that the geometry constraints don't contribute to the final performance. Therefore, we only use the data term for simplicity. Given a color image $\mathbf{u}_0$ converted from optical flow and an initial level set function $\phi$, we iteratively update the level set function $\phi$ to minimize

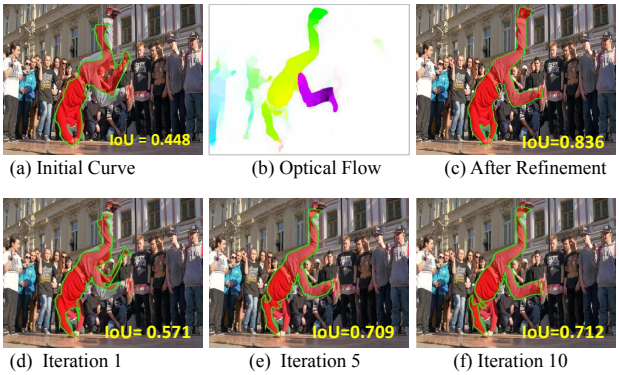
$$\begin{aligned} E_{vos} =& \lambda_1 \sum_{i\in\{r,g,b\}} \int_\Omega |u_{0,i}(\cdot) - c_{1,i}|^2 \cdot H_\varepsilon(\phi(\cdot)) \\ &+ \lambda_2 \sum_{i\in\{r,g,b\}} \int_\Omega |u_{0,i}(\cdot) - c_{2,i}|^2 \cdot [1 - H_\varepsilon(\phi(\cdot))] \end{aligned} \tag{3}$$

where $\phi(\cdot)$ is initialized by the segmentation result of last frame. $\lambda_1$ and $\lambda_2$ are two parameters. $H_\varepsilon$ is the Approximated Heaviside Function as in Eq. 2 with $\varepsilon = 1$. $u_{0,i}$ is the intensity of channel $i$ in the optical flow image $\boldsymbol{u}_0$, $c_{1,i} = \frac{\int_\Omega u_{0,i}(\cdot)\cdot H_\varepsilon(\phi(\cdot))}{\int_\Omega H_\varepsilon(\phi(\cdot))}$ is the average intensity of foreground regions on $u_{0,i}$ and $c_{2,i} = \frac{\int_\Omega u_{0,i}(\cdot)\cdot(1-H_\varepsilon(\phi(\cdot)))}{\int_\Omega(1-H_\varepsilon(\phi(\cdot)))}$ is the average intensity of background regions on $u_{0,i}$.

In the energy function Eq. 3, the first term constrains the homogeneity and smoothness of foreground regions. The second term constrains the background regions to be smooth and homogeneous. In each iteration, we minimize the energy with respect to $\phi$, yields the following Euler-Lagrange equation for $\phi$,

$$\begin{aligned} \frac{\partial\phi}{\partial t} = \delta_\varepsilon(\phi) \cdot \Big[ & -\lambda_1 \sum_{i\in\{r,g,b\}} \int_\Omega |u_{0,i}(\cdot) - c_{1,i}|^2 \\ & + \lambda_2 \sum_{i\in\{r,g,b\}} \int_\Omega |u_{0,i}(\cdot) - c_{2,i}|^2 \Big] \end{aligned} \tag{4}$$

For a frame $t$ we first initialize the active contour on optical flow using last frame's final segmentation, since a proper initialization may greatly decrease the time to convergence and result in a good segmentation. Then we perform the iterative minimization $N$ steps, and treat the region within the final curve as a coarse segmentation of target object. An example is shown in Fig. 3. As can be seen from the example, our model can deal with situations such as incoherent motion and moving background objects. It should be noted that at this step,we can only generate a coarse segmentation. In next subsection, we will show how to generate an accurate segmentation based on the coarse one. In our implement, we also segment the optical flow for frame pair $(t,t-1)$ and combine the two binary masks by OR operation for each pixel. Furthermore, we constrain the coarse segmentation using a mask resulting from applying dilatation operation on the last frame's segmentation.




(a) Initial Curve (b) Optical Flow (c) After Refinement

(d) Iteration 1 (e) Iteration 5 (f) Iteration 10

Figure 3. An example active contour on optical flow. (a) A curve initialized with the final segmentation of the last frame. (b) The optical flow used. (c) The final segmentation using the coarse segmentation in (f) as guidance. (d)-(f) curve at different iteration.

## 3.2. Cascaded Refinement with Guidance

In this section, we present the Cascaded Refinement Network(CRN) which can effectively segment an object under the guidance of the coarse segmentation from optical flow-based active contour model. Since the guidance map provides coarse information about location and shape of target objects, the network doesn't need to assiduously learn how to define and locate a target object, but can focus only on segmenting the dominant object in the given region and with the given coarse shape. Furthermore, since the task for CRN is to segment object instance from static image, it can be effectively trained using datasets for instance segmentation like PASCAL VOC.

### 3.2.1 Cascaded Refinement Network (CRN)

As shown in Fig. 4 (a), our CRN utilizes ResNet101 [22] for feature encoding (i.e., *Conv1, Conv2_x, Conv3_x, Conv4_x, Conv5_x*) and takes a coarse-to-fine scheme. The workflow is formed by five stages of Refining Modules (i.e. $RM^5$,$RM^4$,$RM^3$,$RM^2$,$RM^1$), which are structurally identical. The resolution is 16*16 for the beginning module $RM^5$, and doubled between two consecutive modules. Given an 512*512 input, we first down-sample the coarse segmentation by active contour model to 16*16 as a guidance map. Then, we feed the input image into the network and feed the guidance map into $RM^5$. From $RM^5$ to $RM^1$, the five Refining Modules sequentially operate at their corresponding resolutions, and finally the network outputs a refined segmentation map of full resolution. We rescale the guidance map to such a small size because spatial down-sampling suppresses the inaccuracy of the guidance map

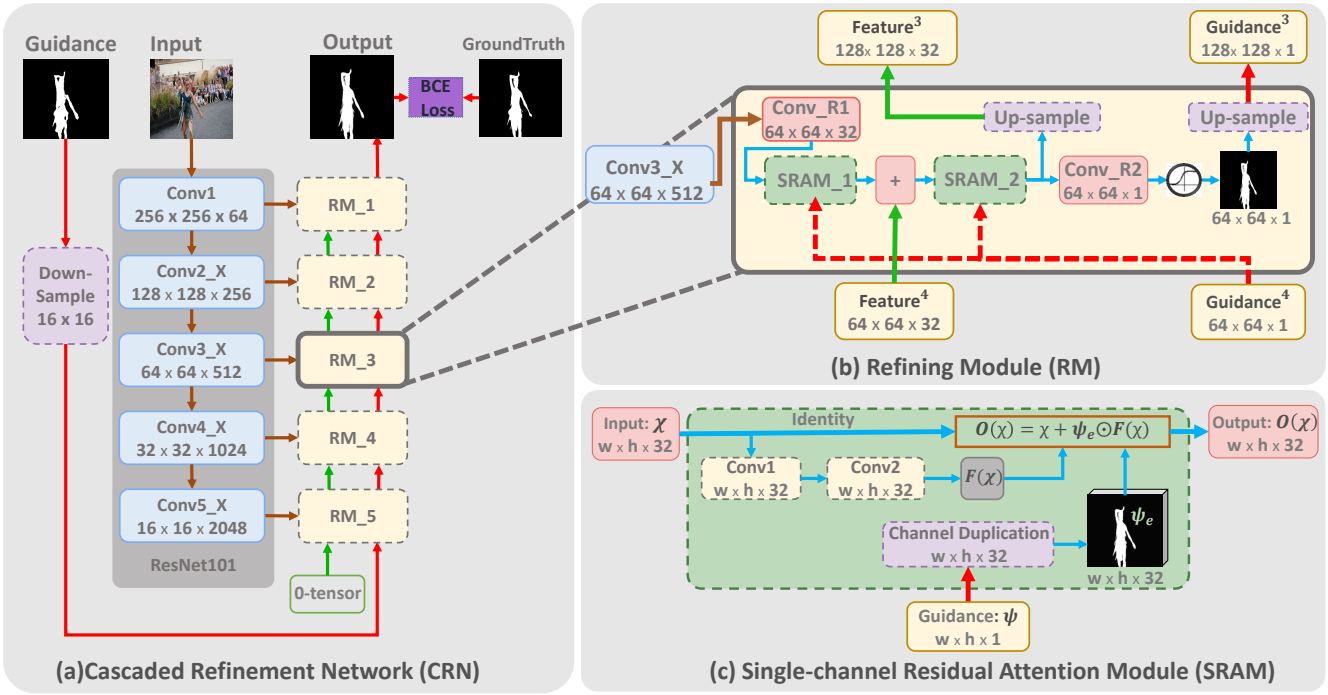



Figure 4. Network overview. (a) An overview of the *Cascaded Refinement Network*(CRN). The "0-tensor" below *RM_5* is a 0-padding tensor used as an input for *RM_5*. (b) Details of the Refining Module *RM_3*. All Refining Modules share the same structure. (c) Details of the Single-channel Residual Attention Module(SRAM) used in RM.

**Refining Module (RM).** All Refining Modules share the same structure. The $RM^3$ is taken as an example and shown in Fig. 4(b). For a Refining Module $RM^i$ in the network, it receives three inputs and produces two outputs. The two outputs are a feature map $Feature^i$ and a segmentation prediction $Guidance^i$. The three inputs include a feature map from corresponding block of ResNet, and the two outputs of last module $RM^{i+1}$, which are $Feature^{i+1}$ and $Guidance^{i+1}$. For the beginning refining module $RM^5$, since only two inputs are available, we make a 0-padding tensor to play the role the lacking input. For the last refining module $RM^1$, we take its segmentation prediction $Guidance^1$ as our final output.

In a Refining Module $RM^i$, we first apply channel reduction on the feature map from ResNet via *Conv_R1*, which is composed of an 1*7 conv layer and a 7*1 conv layer. Then the coarse segmentation $Guidance^{i+1}$, is utilized to guide the features to focus on target regions via a Single-channel Residual Attention Module (i.e. *SRAM_1*). The resulting feature map has the same shape as $Feature^{i+1}$, hence we add them element-wisely. The merged feature is then processed by another Single-channel Residual Attention Module (i.e. *SRAM_2*) to help further focus on the region of interest. And then, the feature from *SRAM_2* is utilized to predict a refined segmentation map of current resolution via a 3*3 convolutional layer *Conv_R2*. Finally both the feature from *SRAM_2* and the refined segmentation map are up-scaled by 2 as outputs. These two outputs (i.e. $Feature^i$ and $Guidance^i$) and the feature map from a higher ResNet block are used as the inputs for $RM^{i-1}$ that works with a larger resolution.

**Single-channel Residual Attention Module (SRAM).** This module is a residual model with a light-weight single-channel attention and inspired by [64, 37]. The framework of the Single-channel Residual Attention Module is shown in Fig. 4(c). SRAM is an important component in Refining Modules, since it incorporates a coarse segmentation of target object as a single-channel attention to help the network focus on regions of interest. As shown in Fig. 4(c), the SRAM takes a feature map $\chi$ as well as a guidance map $\psi$ as input. As in Residual Unit [22], there are two pathways, one is the identity path, the other path is formed by two 3*3 convolutional layers (i.e. *Conv1* and *Conv2*) which convert the input $\chi$ into $F(\chi)$. To highlight the regions of interest, the values of the single-channel guidance map $\psi$ are used as attention coefficients to multiply the feature vectors of corresponding spatial positions in $F(\chi)$. The two feature maps are then summed together element-wisely to be the output,

$$O(\chi) = \chi + \psi_e \odot F(\chi) \tag{5}$$

where $\psi_e$ is a tensor made by duplicating the single-channel guidance map $\psi$ to have the same channel as $F(\chi)$. This module works well because it allows network to focus on regions of interest and learning features for object instance segmentation. Furthermore, since we combine residual unit with a single-channel attention, this module has a very

small number of parameters.

#### 3.2.2 Network Implementation and Training

In our implementation, ResNet101 [22] pretrained on ImageNet [50] is used to initialize the feature encoding network, and for all other convolutional layers we apply Xavier Initialization [17]. In our network, besides the *Conv_R2* in Refining Modules, all the conv layers are followed by BatchNormalization and ReLU layers.

**Training Cascaded Refinement Network (CRN).** The task for CRN is to segment objects from an input image under the guidance of a coarse segmentation, therefore it allows us to exploit existing datasets of other tasks like instance segmentation. We train the network with 11355 images from PASCAL VOC2012 dataset [12] and their instance annotations provided by [21]. To train the CRN, each training sample comprises three components: an input image of resolution 512*512, a binary groundtruth of size 512*512, and a binary guidance map of size 16*16. In each training iteration, we take one image from the PASCAL VOC2012 dataset, and utilize it to make a training batch of size 4. Each sample in the batch is created by randomly choosing an instance as foreground and treating all other regions as background. To generate the guidance maps in the training samples, we apply random morphological operations, including both dilation and erosion combined with three types of kernels (rect, eclipse, cross) of different sizes between 8 - 24 pixels, on the foreground mask and then scale it to 16*16. During training, predictions of all stages are jointly trained with Binary Cross Entropy loss and optimized using SGD with initial learning rate 0.0001 and momentum 0.9 for 10 epochs.

**Offline training for VOS.** After the initial training on the PASCAL VOC dataset, CRN is able to segment generic object instance given a guidance. To adapt the network for video object segmentation, we further train the network on the training split of DAVIS2016 as in [63, 3, 8, 45]. The CRN is finetuned using SGD with batchsize 4, learning rate 0.0001 and momentum 0.9 for 40 epochs. Guidance maps of size 16*16 are made after applying random morphological operations on the groundtruth. Data augmentations like flipping and rescaling are applied in training.

**Online training for VOS.** In CNN-based methods, finetuning on the first frame of a testing video can greatly help the network to focus on the target object and suppress the background [63, 3, 8, 45]. Therefore, for semi-supervised VOS, we also finetune our network on the first frame. During online training, we make the guidance map and apply data augmentations as in the offline training step. The CRN is optimized by SGD with batchsize 1, learning rate 0.002 and momentum 0.9 for 100 iterations. Since the coarse guidance map by active contour model relived the burden of learning for locating the target object, it is unnecessary to heavily finetune our CRN on the first frame. As a result, our method can perform efficiently and effectively.

## 4. Experiments

We perform experiments on two public benchmarks. **DAVIS2016** [46] is a challenging video object segmentation dataset composed of 30 training videos and 20 validation videos. The region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$ [46] are used for quantitative assessment on this dataset. **YoutubeObjects** [48, 28] contains 10 object categories with 126 challenging videos. Each frame is provided with pixel-level annotation. The mean of Intersection of Union (mIoU) metric is adopted for evaluation on this dataset.

Our method is implemented using Python/C with PyTorch. All experiments are performed on a PC with a Nividia TitanX GPU and a 3.3GHz CPU.

### 4.1. Comparison to State-of-the-art

**Semi-supervised VOS.** During testing, the CRN is first trained on the PASCAL VOC dataset, then offline trained on the training split of DAVIS2016. Given a testing video, we first apply online training using the first frame. Then, the active contour for the first frame is initialized by the user annotation and following frames are sequentially processed as in Fig. 2. We compare our method ($\lambda_1 = 0.2$, $\lambda_2 = 0.4$, $N = 10$) with state-of-the-art semi-supervised approaches including OnAVOS [63], OSVOS [3], MSK [45], CTN [30], and SegFlow [8] on DAVIS2016. As shown in the left section of Table 1, without post processing, our method outperforms these state-of-the-art methods. Comparing to OnAVOS with post processing , which is the current state-of-the-art method, our approach achieves better performance in contour accuracy ($\mathcal{F}$),and comparable performance in region similarity ($\mathcal{J}$). However, OnAVOS applies techniques like online adaptation, test-time augmentation, and post processing with denseCRF [34], which rely on heavy consumption of time (15.57 seconds/frame) and computation resource. Our method is much more efficient (0.73 seconds/frame) and can accurately segment object from a video frame via a single feedforward process without any postprocessing steps. On YoutubeObjects, we also compare our method with state-of-the-art methods like OnAVOS [63], OSVOS [3], MSK [45], BVS [42], OFL [61], STV [68], JOT [69] . As presented in Table 2, the results show our method achieves state-of-the-art performance. The relatively weak performance on YoutubeObjects dataset is due to the low resolution and very large sampling gap of video frames. Some qualitative results of our method are shown in the first two rows of Fig. 5.

**Unsupervised VOS.** We also extend our method for unsupervised video object segmentation task. To generate a

| Metric | Semi-supervised | | | | | | | Unsupervised | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ours | OnAVOS | | OSVOS | MSK | CTN | SegFlow | Ours | ARP | LVO | FSEG | LMP | SegFlow |
| | | with CRF | w/o. CRF | | | | | | | | | | |
| $\mathcal{J}$ Mean ↑ | 0.844 | **0.861** | 0.832 | 0.798 | 0.797 | 0.735 | 0.761 | **0.764** | 0.762 | 0.759 | 0.707 | 0.700 | 0.674 |
| Recall ↑ | **0.971** | 0.961 | 0.955 | 0.936 | 0.931 | 0.874 | 0.906 | 0.900 | **0.911** | 0.891 | 0.835 | 0.850 | 0.814 |
| Decay ↓ | 0.056 | 0.052 | **0.050** | 0.149 | 0.089 | 0.156 | 0.121 | **-0.009** | 0.007 | 0.000 | 0.015 | 0.013 | 0.062 |
| $\mathcal{F}$ Mean ↑ | **0.857** | 0.849 | 0.851 | 0.806 | 0754 | 0.693 | 0.760 | **0.766** | 0.706 | 0.721 | 0.653 | 0.659 | 0.667 |
| Recall ↑ | **0.952** | 0.897 | 0.928 | 0.926 | 0.871 | 0.796 | 0.855 | **0.882** | 0.835 | 0.834 | 0.738 | 0.792 | 0.771 |
| Decay ↓ | **0.052** | 0.058 | 0.060 | 0.150 | 0.090 | 0.129 | 0.104 | **-0.014** | 0.079 | 0.013 | 0.018 | 0.025 | 0.051 |
| time(s/f) | **0.73** | 15.57 | 13.41 | 9.24 | (12) | (1.3) | (7.9) | **0.36** | - | - | - | - | (7.9) |

Table 1. Performance on the validation split of DAVIS2016. *Left*: performance for semi-supervised VOS. For OnAVOS, we compare with two versions including using postprocessing ('with crf') and not using postprocessing ('w/o crf'). *Right*: performance for unsupervised methods. In the last row, the numbers in parentheses are computation time reported in the original papers of corresponding methods.

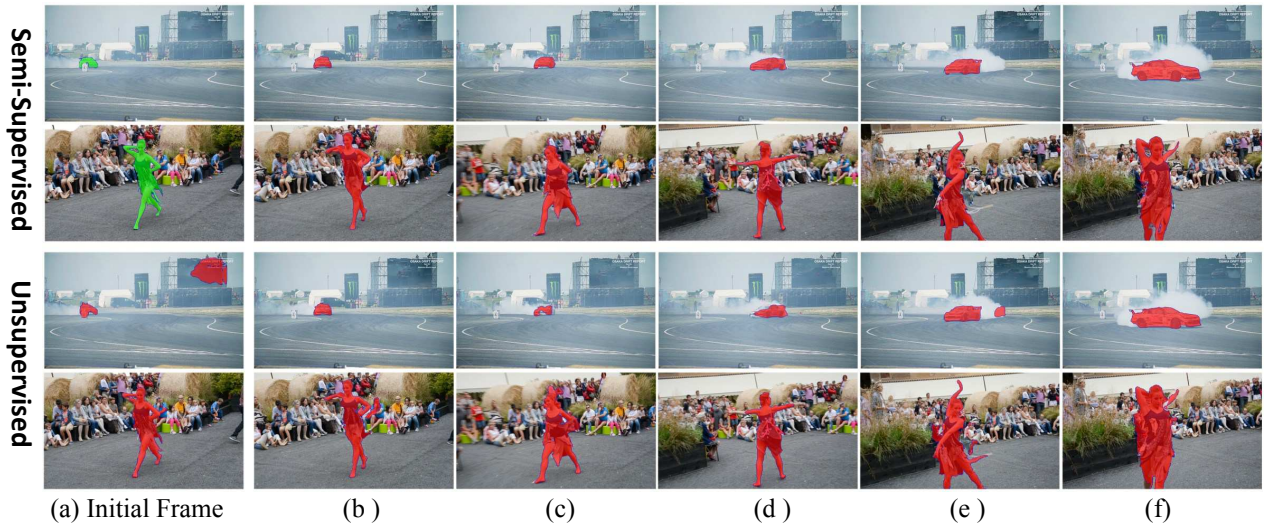



Figure 5. Qualitative results of the proposed method. From top to bottom, the first two rows are semi-supervised results, the last two rows are unsupervised results. From left to right: (a) The first frame overlaid with initial mask. Mask in the first and second rows of (a) are user annotations , and the masks in the third and fourth rows of (a) are our unsupervised results with a predefined rectangles as initial contour. (b-f) our segmentation results for subsequent frames.

| | Ours | OnAVOS | OSVOS | MSK | BVS | OFL |
|---|---|---|---|---|---|---|
| mIoU | 0.766 | **0.774** | 0.726 | 0.717 | 59.7 | 70.1 |

Table 2. Comparisons on YoutubeObjects

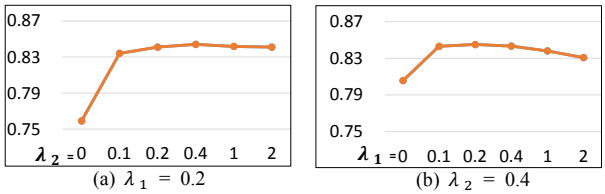
mask for the first frame, we initialize the active contour with a predefined rectangle on the optical flow image, and evolve it for 20 iterations to get a coarse segmentation. Then, our Cascaded Refinement Network takes the coarse segmentation as guidance map and generates a more accurate segmentation (The last two rows of Fig. 5(a)). With this segmentation as the initial mask for the first frame, subsequent frames are sequentially processed as in Fig. 2. The Cascaded Refinement Network used here is trained offline. In the right section of Table 1, we compare with state-of-the-art unsupervised methods including ARP [31], LVO [59], F-SEG [11], and LMP [58] on DAVIS2016. The results show that our method ($\lambda_1 = 0.2$, $\lambda_2 = 0.4$, $N = 10$) outperforms other methods and achieves state-of-the-art performance. Some examples of our unsupervised method are shown in the last two rows of Fig. 5. As shown in the figures, our unsupervised method can track the object regions reliably.

### 4.2. Method Analysis

**Active contour for coarse segmentation.** In this work, we apply active contour (Eq 4) on optical flow images to segment moving objects. There are three parameters: $\lambda_1$, $\lambda_2$, as well as the iteration number $N$. After performing some coarse manual tuning based on [6], we set $\lambda_1$=0.2, $\lambda_2$=0.4, and performance for different combination of these two parameters are shown in Fig. 6. To evaluate the effectiveness of the optical flow-based active contour model, we run our system without the Cascaded Refinement Network. The performance for different iteration number $N$ is shown in the first row (denoted by "AC-only") of Table 3. As shown in the Table, without the refinement by CRN, our optical flow-based active contour model itself can achieve a mIoU of 0.553 on DAVIS2016. Furthermore, the performance of the complete system with different iterations number $N$ is

| | N=0 | N=1 | N=5 | N=10 | N=20 |
|---|---|---|---|---|---|
| AC-only | 0.272 | 0.547 | 0.551 | **0.553** | 0.550 |
| AC+CRN | 0.824 (CRN-only) | 0.842 | 0.843 | **0.844** | 0.844 |

Table 3. mIoU for different iterations number $N$ in the Active Contour model. "AC-only" represents only using the optical flow-based active contour. "AC+CRN" denotes our complete system.




(a) $\lambda_1 = 0.2$ (b) $\lambda_2 = 0.4$

Figure 6. Performance (mIoU) on DAVIS2016 for different $\lambda_1$ and $\lambda_2$ in the active contour model.

| Metric | Training on PascalVOC | Offline Training | Online Training | Baseline |
|---|---|---|---|---|
| $\mathcal{J}$ Mean ↑ | 0.526 | 0.764 | **0.844** | 0.805 |
| Recall ↑ | 0.617 | 0.895 | **0.971** | 0.939 |
| Decay ↓ | 0.136 | 0.115 | **0.056** | 0.058 |
| $\mathcal{F}$ Mean ↑ | 0.472 | 0.757 | **0.857** | 0.821 |
| Recall ↑ | 0.522 | 0.871 | **0.952** | 0.924 |
| Decay ↓ | 0.118 | 0.112 | 0.052 | **0.049** |

Table 4. Performance on DAVIS2016 for different training phase of CRN and the baseline.

shown in the last row (denoted by "AC+CRN") of Table 3. In this row, "$N = 0$" represents that the segmentation of the last frame are directly used as guidance for CRN, thus the system runs only with CRN. As shown in the table, without optical flow-based active contour, our CRN itself achieves a performance of 0.824, which is already better than most of the state-of-the-art methods. When combining with an active contour model of 10 iteration, the performance further increases to 0.844. The improvement proves that the proposed optical flow-based active contour model is effective.

**Cascaded Refinement Network (CRN).** We first compare our method with a baseline. The baseline has the same structure as CRN except that the *SRAM_1* and *SRAM_2* in $RM^5$ are replaced by normal residual units [22]. Thus, the baseline doesn't accept guidance map from outside and ignore the motion between frames. We first train the baseline on PASCAL VOC for objectness [63], then finetune it on the training split of DAVIS2016, and finally perform testing with online training. As shown in the last column of Table 4, the baseline also achieves state-of-the-art performance. However, when comparing our motion-guided Cascaded Refinement Network, the baseline lags behind with a significant gap of 0.039 in mean($\mathcal{J}$) and 0.036 in mean($\mathcal{F}$).

As described in section 3.2.2, we first train the CRN on PascalVOC dataset, then perform offline training using training split of DAVIS2016, and finally apply online training with the first frame for testing. In Table 4, we present the performance for our system with a CRN of different training phases. As shown in the table, offline training step adapts the CRN to the task of video object segmentation, thus improving the performance by 0.238 in mean($\mathcal{J}$). Furthermore, online training with the first frame helps our CRN further adapt to the testing video and therefore increases the mean($\mathcal{J}$) by 0.08 . For the semi-supervised task, performance of different training iterations for the online training step is shown in Table 5. As we can see, too much finetuning on the first frame doesn't only increases running time of CRN, but also compromise the performance sometimes.

| #Iter | 10 | 50 | 100 | 150 | 200 | 500 |
|---|---|---|---|---|---|---|
| mIoU | 0.819 | 0.841 | **0.844** | 0.844 | 0.843 | 0.843 |
| time(s/f) | **0.40** | 0.55 | 0.73 | 0.92 | 1.09 | 2.18 |

Table 5. Performance for different iterations in the online training step of CRN.

**Running time.** In experiments, we resize inputs to 512*512. For each frame, optical flow estimation with FlowNet2 [27] takes about 0.15 seconds. The active contour model with $N = 10$ takes 0.10 seconds, and the Cascaded Refinement Network takes about 0.11 seconds. For the semi-supervised task, performing online training with 100 iterations takes about 25 seconds for per video. As a result, our method runs at 0.73 seconds per frame (s/f) in average on DAVIS2016. Compared with other state-of-the-art methods (last row of Table 1) such as OSVOS (9.24 s/f), OnAVOS (15.57 s/f), our method achieves a state-of-the-art accuracy at a much faster speed. For the unsupervised task, since we don't need to finetune on the first frame, our method can achieve the state-of-the-art performance with an average speed of 0.36 seconds per frame.

## 5. Conclusion

In this paper, a motion guided Cascaded Refinement Network for video object segmentation is presented. We first propose to apply active contour on optical flow to segment moving object. We also present a Cascaded Refinement Network that generate accurate segmentations under the guidance of coarse results from the optical flow-based active contour. In the proposed system composed by these two components, motion information and deep CNN can well complement each other for the task of VOS. Experiments on benchmarks demonstrate that our method achieves state-of-the-art performance at a much faster speed.

## Acknowledgement

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the Infocomm Media Development Authority, Singapore. The authors gratefully acknowledge the support of NVIDIA AI Technology Center for their donation of a Titan Xp GPU used for our research.

## References


[1] S. Avinash Ramakanth and R. Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, 2014. 1, 2

[2] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. on PAMI*, 33(3):500–513, 2011. 1

[3] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 2, 6

[4] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997. 2

[5] T. F. Chan, B. Y. Sandberg, and L. A. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11(2):130–141, 2000. 2, 3, 4

[6] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001. 2, 3, 4, 7

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. on PAMI*, 2017. 1

[8] 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, 6

[9] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007. 2

[10] S. Duffner and C. Garcia. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In *ICCV*, 2013. 2

[11] S. Dutt Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017. 1, 2, 7

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 6

[13] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 2

[14] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *CVPR*, 2015. 2

[15] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, pages 1846–1853, 2012. 1

[16] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012. 2

[17] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 6

[18] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 2

[19] J. Gu et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 2017. 1

[20] J. Guo, Z. Li, L.-F. Cheong, and S. Zhiying Zhou. Video co-segmentation for meaningful action extraction. In *ICCV*, 2013. 1

[21] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5, 6, 8

[23] E. Horbert, K. Rematas, and B. Leibe. Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *ICCV*, 2011. 2

[24] P. Hu, B. Shuai, J. Liu, and G. Wang. Deep level sets for salient object detection. In *CVPR*, 2017. 2

[25] P. Hu, G. Wang, and Y.-P. Tan. Recurrent spatial pyramid cnn for optical flow estimation. *IEEE Trans. on Multimedia*, 2018. 1

[26] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics*, 41(6):797–819, 2011. 1

[27] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 1, 2, 3, 8

[28] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 6

[29] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, 2017. 2

[30] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *CVPR*, 2017. 1, 2, 6

[31] Y. Jun Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 2, 7

[32] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 2

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

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

[35] C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: a new variational formulation. In *CVPR*, 2005. 2

[36] J. Li, A. Zheng, X. Chen, and B. Zhou. Primary video object segmentation via complementary cnns and neighborhood reversible flow. In *ICCV*, 2017. 1, 2

[37] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*, 2017. 5

[38] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 595–600, 2005. 1

[39] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[40] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012. 2

[41] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *arXiv preprint arXiv:1709.06031*, 2017. 2

[42] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 2, 6

[43] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011. 2

[44] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 2, 3

[45] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 1, 2, 6

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

[47] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015. 2

[48] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 6

[49] T. Riklin-Raviv, N. Sochen, and N. Kiryati. Shape-based mutual segmentation. *International Journal of Computer Vision*, 79(3):231–245, 2008. 2

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 6

[51] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998. 2

[52] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994. 1

[53] Y. Shi and W. C. Karl. Real-time tracking using level sets. In *CVPR 2005*, 2005. 2

[54] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017. 1, 2

[55] A. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *ICCV*, 2007. 3

[56] X. Sun, H. Yao, S. Zhang, and D. Li. Non-rigid object contour tracking via a novel supervised level set model. *IEEE Trans. on Image Processing*, 24(11):3386–3399, 2015. 2

[57] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 1

[58] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 2, 7

[59] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 1, 2, 7

[60] A. Tsai, A. Yezzi, and A. S. Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE transactions on Image Processing*, 10(8):1169–1186, 2001. 2

[61] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 1, 2, 6

[62] D. Varas and F. Marques. Region-based particle filter for video object segmentation. In *CVPR*, 2014. 1

[63] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 1, 2, 6, 8

[64] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *CVPR*, 2017. 5

[65] H. Wang, T. Raiko, L. Lensu, T. Wang, and J. Karhunen. Semi-supervised domain adaptation for weakly labeled semantic video object segmentation. In *ACCV*, 2016. 2

[66] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011. 2

[67] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 2

[68] W. Wang, J. Shen, J. Xie, and F. Porikli. Super-trajectory for video segmentation. In *ICCV*, 2017. 1, 2, 6

[69] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015. 2, 6

[70] F. Xiao and Y. Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016. 2

[71] X. Yang, X. Gao, D. Tao, and X. Li. Improving level set method for fast auroral oval segmentation. *IEEE Trans. on Image Processing*, 23(7):2854–2865, 2014. 2

[72] D. Yeo, J. Son, B. Han, and J. Hee Han. Superpixel-based tracking-by-segmentation using markov chains. In *CVPR*, 2017. 1

[73] 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, 3

[74] X. Zhou, X. Huang, J. S. Duncan, and W. Yu. Active contours with group similarity. In *CVPR*, 2013. 2