Graph-to-Graph Energy Minimization for Video Object Segmentation

Yuezun Li[1]*, Longyin Wen[2]*, Ming-Ching Chang[1] and Siwei Lyu[1]
[1] University at Albany, State University of New York, USA
[2] JD Finance AI Lab, USA

Abstract

*We describe a new unsupervised video object segmentation (VOS) method based on the graph-to-graph energy minimization, which focuses on exploiting the mutual bootstrapping information between bottom-up (*i.e.*, using pixel/superpixel attributes) and top-down (*i.e.*, using learned appearance and motion cues) processes in a unified framework. Specifically, we construct a graph-to-graph energy function to encode the spatial similarities among superpixels (superpixel-graph) and temporal consistency among regions (region-graph). An efficient heuristic iterative algorithm is used to minimize the energy function to get the optimal assignment of superpixel and region labels to complete the VOS task. Experiments on two challenging benchmarks (*i.e.*, SegTrack v2 and DAVIS) show that the proposed method achieves favorable performance against the state-of-the-art unsupervised VOS methods and comparable performance with the state-of-the-art semi-supervised methods.*

1. Introduction

Video object segmentation (VOS) [18, 24, 13] concerns the problem of extracting foreground objects from video frames. It is a crucial step for many video analysis tasks such as video editing, video summarization and scene understanding. Most existing VOS methods require different degrees of human interaction. Some need users to interactively correct segmentation errors [27, 1, 35, 9] and others need users to provide initial delineation of the foreground object [13, 37, 34, 22, 17, 42]. While leading to more accurate segmentation results, the requirement of human intervention puts burden on the user. As such, recent years have also seen several *unsupervised* VOS methods, which aim to obtain complete space-time segmentation of moving objects fully automatically from a video without user annotations or any prior information about the object (*e.g.*, its class).

Existing unsupervised VOS methods [41, 43, 18, 24, 23, 10] use either *bottom-up* (*i.e.*, using pixel/superpixel attributes) or *top-down* (*i.e.*, using learned appearance and motion cues) processes to group the pixels/superpixels into spatio-temporal tubes that may belong to the same object. Notably, the two processes can mutually benefit from each other, and combining them can lead to further improvement in performance. Thus, Xiao *et al.* [39] recently attempt to combine the bottom-up and top-down cues, *i.e.*, iteratively discover the harder instances in adjacent frames and update the appearance model of objects in a self-paced manner to complete the VOS task. However, this bounding box based propagation mechanism is not accurate enough to generate correct tube proposals, especially for videos with cluttered and complex backgrounds.

In this work, we describe a new *unsupervised* VOS method based on the graph-to-graph energy minimization (GEM). Our method focuses on combining the bottom-up and top-down cues based on superpixels and regions (rather than pixels and bounding boxes in [39]) in a unified framework. Specifically, we construct a graph-to-graph energy function to encode the spatial similarities among superpixels (superpixel-graph) and temporal consistency among regions (region-graph), see Figure 1. Starting with the extracted superpixels [12] and region proposals [6] from each video frame, our method first connects 'object-like" *regions* across different frames to generate the candidate *tubes* of objects based on the attributes of superpixels, and then classifies superpixels within each frame to refine the regions of the extracted tubes based on the appearance and motion features. These two steps are iterated to minimize the overall graph-to-graph energy function, which leads to an optimal assignment of superpixel and region labels to obtain the video segmentation results. We perform several experiments on the SegTrack v2 [20] and DAVIS [25] datasets, to demonstrate that our method achieves favorable performance against the state-of-the-art unsupervised VOS methods and comparable performance with the state-of-the-art semi-supervised methods.

The main contributions of this paper are summarized as follows. First, we provide a new formulation of the unsupervised VOS problem based on the minimization of the graph-to-graph energy function. Our method combines the

*The authors contribute equally.

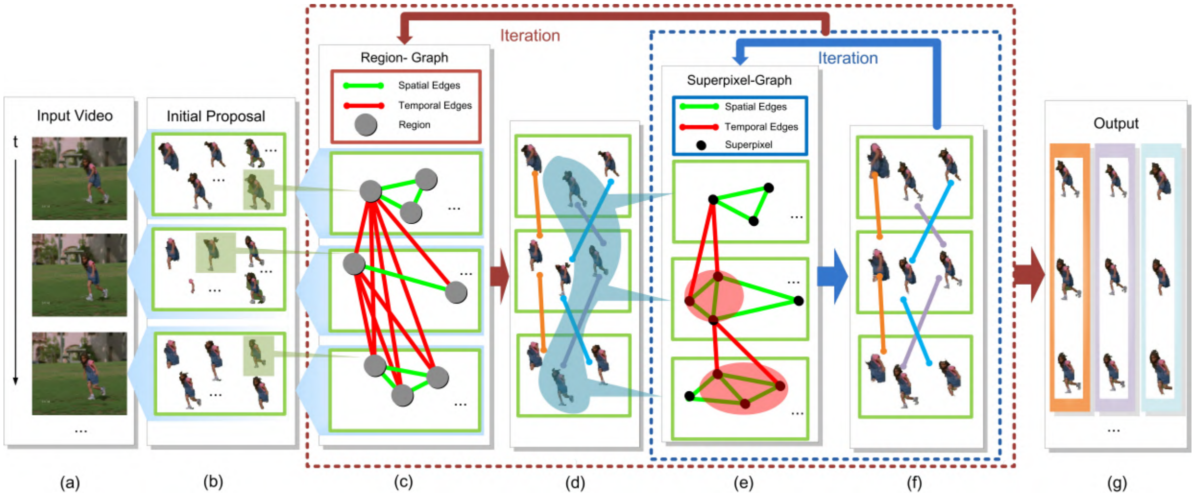

Figure 1. Overview of the proposed unsupervised VOS algorithm. (a) Input video frames. (b) Initial object proposals in each frame (regions in the green box) generated by [5]. (c) Region-graph construction. (d) Object tubes generated by optimizing region-energy. (e) Superpixel-graph construction. (f) Object tubes optimized by minimizing the superpixel-energy. (g) Ranked object tubes in the video.

bottom-up and top-down cues in a unified framework. Second, we present an efficient iterative algorithm to solve the energy minimization problem. Third, our algorithm achieves favorable performance against the state-of-the-art unsupervised VOS methods and comparable performance with the state-of-the-art semi-supervised methods.

2. Related Work

Interactive Video Object Segmentation. The goal of interactive VOS is to extract objects in videos with the help of user input to correct the algorithm's mistakes, *e.g.*, [1, 27, 35, 9]. Price *et al.* [27] extract multiple video cues (*e.g.*, shape, color, temporal coherence, and gradient) to select objects in video sequences interactively using a graph-cut optimization framework. In [1], a set of local classifiers are integrated with multiple local image features to complete the segmentation task. To minimize user effort, the work of [35] uses active learning technique to intelligently query a user to label only certain objects in certain frames that are likely to improve the performance. Similarly, Fathi *et al.* [9] combine self-learning and active learning to guide the user about where to annotate, and use harmonic functions to solve the problem efficiently.

Semi-Supervised Video Object Segmentation. Semi-supervised VOS aims to segment video objects based on foreground regions annotated in the initial video frames, and propagate them to the remaining frames. Jain *et al.* [13] design a high order supervoxel label consistency potential for the foreground region propagation, which leverages bottom-up supervoxels to guide the estimation towards the long-range coherent regions. Wen *et al.* [37] integrate multi-part tracking and segmentation into a unified energy objective, which is efficiently optimized by a RANSAC-style approach. Tsai *et al.* [34] jointly optimize VOS and optical flow estimation in a unified framework using an iterative scheme to exploit the mutually bootstrapping information between the two tasks to obtain better performance. Caelles *et al.* [3] use the fully-convolutional neural network to transfer generic semantic information to generate video segmentation, which fine-tunes the pre-trained network with the appearance of a single annotated object in the test sequence. In [42], a multiple granularity analysis method is presented to segment the provided object in a coarse-to-fine manner, which estimates superpixel labeling using multiple instance learning to obtain the coarse segmentation mask. Graph-cut is used to refine the pixelwise mask to produce the final results.

Unsupervised Video Segmentation. Some unsupervised video segmentation algorithms use a bottom-up strategy to group spatial-temporal coherent tubes without any prior information. Xu *et al.* [41] implement a graph-based hierarchical segmentation method within a streaming framework, which enforces a Markovian assumption on the video stream to approximate full video segmentation. Yu *et al.* [43] propose an efficient and robust algorithm based on parametric graph partitioning that identifies and removes between-cluster edges to generate node clusters for video segmentation.

Several other unsupervised VOS methods upgrade bottom-up video segmentation to object-level segments. Lee *et al.* [18] use static and dynamic cues to identify object-like regions, and discover hypothesis object groups with persistent appearance and motion. Then, each ranked hypothesis is used to estimate a pixel-level object labeling across all frames. Li *et al.* [20] track multiple holistic figure-ground segments simultaneously to generate video object proposals, and train an online non-local appearance models for each track using a multi-output regularized least squares formulation. Papazoglou *et al.* [24] present a fast unsupervised VOS method, which generates object proposals by simply selecting pixels in the video by combining two kinds of motion boundaries extracted from optical flow. In [39], a series of easy-to-group object instances are discovered, and the appearance model of the instances are iteratively updated to detect the harder-to-find instances in temporally-adjacent frames.

In recent years, some unsupervised VOS methods [14, 31, 32] rely on deep convolutional neural network (CNN) to learn the appearance and motion patterns for video segmentation. Jain *et al.* [14] formulate the video segmentation task as a structured prediction problem and design a two-stream fully convolutional network to fuse motion and appearance, which is trained using available image segmentation annotations together with weakly annotated video data. Tokmakov *et al.* [31] also use a fully convolutional network to learn motion patterns in videos to handle the unsupervised VOS task, which is trained entirely using synthetic video sequences with ground-truth optical flow and motion segmentation. In [32], a two-stream neural network with an explicit memory module is used to segment moving objects in unconstrained videos, where the two-stream structure encodes the spatial and temporal features in a video sequence respectively, while the memory module captures the evolution of objects over time.

3. Problem Formulation

Given the video sequence, we aim to generate several object tubes $\mathbf{O}$. As discussed above, we first use the method in [12, 40] to generate coherent superpixels in each frame with enhanced temporal consistency. Let $\mathbf{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_\mathrm{T}\}$ to be the generated superpixels, where T is the total number of frames in the video, and $\mathcal{P}_t$ is the set of superpixels at frame t. Then, generating video object proposals is formulated as a label assignment problem to each superpixel in $\mathbf{P}$.

Superpixels at different frames are assembled to form tubes through *regions*, which are groups of superpixels that may correspond to the objects. As such, the assignment of superpixels (*i.e.*, indicating which tube it belongs to) are performed in two steps: assigning region labels to superpixels and linking regions to constructing tubes in the temporal domain. We integrate these two steps into a unified framework to make them mutually benefit from each other. We use $\mathbf{R} = \{\mathcal{R}_1, \cdots, \mathcal{R}_\mathrm{T}\}$ to denote the region set in video frames, where $\mathcal{R}_t = \{R_{1,t}, \cdots, R_{\mathrm{K}_t,t}\}$ $(1 \leq t \leq \mathrm{T})$ is the set of regions at frame t, $R_{i,t}$ is the i-th region, and K_t is total number of regions in the video frame. Let $\eta_{i,t}$ be the identity of region $R_{i,t}$, and Φ be the clusters of region identities (*i.e.*, the regions in the same cluster belonging to the same tube). The overall energy is formed by two terms, the superpixel-energy $E_\Phi(\mathbf{L}, \mathbf{R})$ and the region-energy $E_\mathbf{R}(\Phi)$, and the overall label assignment problem is formulated as

$$\min_{\mathbf{L},\mathbf{R},\Phi} E_\mathbf{R}(\Phi) + E_\Phi(\mathbf{L}, \mathbf{R}). \tag{1}$$

$E_\mathbf{R}(\Phi)$ encodes the energy of linking regions to construct tubes, $E_\Phi(\mathbf{L}, \mathbf{R})$ encodes the energy of assigning region labels to superpixels, and $\mathbf{L}$ is the set of binary labels ($\{0, 1\}$) indicating whether a superpixel belongs to a region.

3.1. Region-Energy

The region-energy $E_\mathbf{R}(\Phi)$ models the energy of partitioning regions to generate the object tubes. If two regions in consecutive frames have similar appearance and size, and overlapping with each other in the spatial domain, we expect that the corresponding energy $E_\mathbf{R}(\Phi)$ that groups them into the same tube to be small. To this end, we associate each region with a node in the region-graph $\mathbf{G}^\mathrm{rg} = (\mathbf{R}, \mathcal{E}^\mathrm{rg})$ (see Figure 1(c)), where $\mathcal{E}^\mathrm{rg}$ is the edge set describing the interactions between regions. Similar to [38], the region partition task is formulated as exploiting the dense clusters on $\mathbf{G}^\mathrm{rg}$, and each exploited cluster corresponds to a specific object tube.

We introduce an indicator vector $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,\mathrm{u}})$ to describe the i-th dense cluster ($i = 1, \cdots, \mathrm{N}$, where N is the total number of exploited dense clusters), where $\mathrm{u} = \mathrm{T} \cdot \mathrm{K}_t$ is the total number of regions, $x_{i,j} = 1/\beta_i$ if the j-th node is included in the i-th dense structure, and $x_{i,j} = 0$ otherwise, β_i is the total number of nodes in the i-th dense structure that will be inferred in the optimization. Then, the region-energy $E_\mathbf{R}(\Phi)$ is defined as

$$E_\mathbf{R}(\Phi) = \omega_1 \cdot \exp\left(-\textstyle\sum_{\mathcal{C}_i \in \mathbf{C}} \mathbf{x}_i^\top \mathbf{M} \mathbf{x}_i\right), \tag{2}$$

where $\mathcal{C}_i$ is the i-th cluster of nodes found in $\mathbf{G}^\mathrm{rg}$, and $\mathbf{C}$ is the set of all clusters, $\mathbf{M}$ is the edge weight matrix of $\mathbf{G}^\mathrm{rg}$, and ω_1 is the predefined parameter used to balance the influence of region-energy.

The edge weight matrix $\mathbf{M}$ in (2) captures the similarities between regions, which is defined as

$$\mathbf{M}(i,j) = \omega_2 M^\mathrm{c}_{i,j} + \omega_3 M^\mathrm{d}_{i,j} + \omega_4 M^\mathrm{o}_{i,j} + \omega_5 M^\mathrm{s}_{i,j}, \tag{3}$$

where ω_2, ω_3, ω_4, and ω_5 are preset parameters balancing the four terms, $M^\mathrm{c}_{i,j}$ is the cosine distance between the LAB color histograms of the i-th and j-th regions, $M^\mathrm{d}_{i,j}$ is the cosine distance between the average convolutional neural network (CNN) features[1] of the pixels in the i-th and j-th regions, $M^\mathrm{o}_{i,j}$ is the interaction-over-union (IoU) between the i-th and j-th regions, and $M^\mathrm{s}_{i,j}$ is the size similarity. We compute the size similarity as $M^\mathrm{s}_{i,j} = 1 - \frac{|\mathbf{U}_i - \mathbf{U}_j|}{\max(\mathbf{U}_i, \mathbf{U}_j)}$, where $\mathbf{U}_i$ and $\mathbf{U}_j$ are the areas of the i-th and j-th regions, respectively.

[1] We use the method in [21] to extract hierarchical CNN features for each pixel in each video frame.

3.2. Superpixel-Energy

The superpixel-energy $E_{\Phi}(\mathbf{L}, \mathbf{R})$ corresponds to the cost of a particular region label assignment to superpixels. Specifically, we form a graph $\mathbf{G}^{\mathrm{sp}} = \{\mathbf{P}, \mathcal{E}^{\mathrm{sp}}\}$ (see Figure 1(e)), where $\mathbf{P}$ is the set of superpixels in the video, and $\mathcal{E}^{\mathrm{sp}}$ is the edge set including two types of edges: the temporal edges $\mathcal{E}^{\mathrm{sp}}_{\mathcal{X}}$ and the spatial edges $\mathcal{E}^{\mathrm{sp}}_{\mathcal{S}}$, *i.e.*, $\mathcal{E}^{\mathrm{sp}} = \mathcal{E}^{\mathrm{sp}}_{\mathcal{S}} \cup \mathcal{E}^{\mathrm{sp}}_{\mathcal{X}}$. Then, the superpixel-energy is defined as:

$$E_{\Phi}(\mathbf{L}, \mathbf{R}) = \sum_{\phi_j \in \Phi} \Big(\sum_{t=1}^{\mathrm{T}} \sum_{R_{i,t} \in \mathcal{R}_t} \delta\big(\eta_{i,t} \in \phi_j\big) \sum_{p \in \mathcal{P}_t} D(\ell_t^p, R_{i,t}) + \sum_{t=1}^{\mathrm{T}} \sum_{(p,q) \in \mathcal{E}^{\mathrm{sp}}_{\mathcal{S}}} V_{\mathcal{S}}(\ell_t^p, \ell_t^q) + \sum_{t=1}^{\mathrm{T}-1} \sum_{(p,q) \in \mathcal{E}^{\mathrm{sp}}_{\mathcal{X}}} V_{\mathcal{X}}(\ell_t^p, \ell_{t+1}^q) \Big), \tag{4}$$

where ϕ_j is the j-th cluster in Φ, and $\delta(\eta_{i,t} \in \phi_j) = 1$, if $\eta_{i,t} \in \phi_j$, *i.e.*, $R_{i,t}$ belongs to cluster ϕ_j, and 0 otherwise. $D(\ell_t^p, R_{i,t})$ models the unary data cost that the superpixel p belongs to the region $R_{i,t}$ (*i.e.*, $\ell_t^p \in \{0, 1\}$), considering both the appearance and location cues. $V_{\mathcal{S}}(\ell_t^p, \ell_t^q)$ and $V_{\mathcal{X}}(\ell_t^p, \ell_{t+1}^q)$ are the smoothness cost encoding the spatial and temporal consistency of superpixels, respectively.

Data cost. If superpixel p belongs to region $R_{i,t}$ ($\ell_t^p = 1$), we expect that it has small energy regarding the color, CNN features and spatial arrangements of the region models. As such, we define the data cost as:

$$\begin{aligned} D(\ell_t^p, R_{i,t}) \; &= \omega_6 \Psi_{\mathrm{c}}(\ell_t^p, R_{i,t}) + \omega_7 \Psi_{\mathrm{d}}(\ell_t^p, R_{i,t}) \\ &+ \omega_8 \Psi_{\mathrm{l}}(\ell_t^p, R_{i,t}), \end{aligned} \tag{5}$$

where $\Psi_{\mathrm{c}}(\ell_t^p, R_{i,t})$, $\Psi_{\mathrm{d}}(\ell_t^p, R_{i,t})$, and $\Psi_{\mathrm{l}}(\ell_t^p, R_{i,t})$ are the energy terms regarding color features, CNN features, and location cue, respectively. Weights ω_6, ω_7, and ω_8 are the predefined parameters balancing the three terms.

The energy $\Psi_{\mathrm{c}}(\ell_t^p, R_{i,t})$ represents the likelihood of superpixel p belonging to region $R_{i,t}$ regarding the color cue. We use the Gaussian mixture model (GMM) to describe the color information of $R_{i,t}$ and the background (the area that excludes the region from the corresponding frame). Specifically, we construct the GMM of RGB colors for $R_{i,t}$ and the background, then, $\Psi_{\mathrm{c}}(\ell_t^p, R_{i,t})$ is defined as

$$\Psi_{\mathrm{c}}(\ell_t^p, R_{i,t}) = -\log\big(\mathbf{H}(\ell_t^p; h_p)\big), \tag{6}$$

where $\mathbf{H}(\ell_t^p; h_p)$ is the GMM of $R_{i,t}$ or the background, and h_p is the average RGB values of pixels in p.

The energy $\Psi_{\mathrm{d}}(\ell_t^p, R_{i,t})$ is derived from the likelihood of superpixel p belonging to the region $R_{i,t}$ according to the CNN features. We construct two logistic regression models of CNN features of the pixels in region $R_{i,t}$ and the background, respectively. Then, $\Psi_{\mathrm{d}}(\ell_t^p, R_{i,t})$ is defined as

$$\Psi_{\mathrm{d}}(\ell_t^p, R_{i,t}) = -\log\big(\Lambda(\ell_t^p; g_p)\big), \tag{7}$$

where $\Lambda(\ell_t^p; g_p)$ is the logistic regression model of $R_{i,t}$ or its corresponding background, and g_p is the average CNN features of the pixels in superpixel p.

If superpixel p belongs to region $R_{i,t}$, we expect that p locates near $R_{i,t}$ in the spatial domain. Thus, we define $\Psi_{\mathrm{l}}(\ell_t^p, R_{i,t})$ by the distance between p and $R_{i,t}$, *i.e.*,

$$\Psi_{\mathrm{l}}(\ell_t^p, R_{i,t}) = \begin{cases} \frac{\mathbf{Q}(p, R_{i,t})}{\max_{\tilde{p} \in \mathcal{P}_t} \mathbf{Q}(\tilde{p}, R_{i,t})}, & \ell_t^p = 1, \\ 1 - \frac{\mathbf{Q}(p, R_{i,t})}{\max_{\tilde{p} \in \mathcal{P}_t} \mathbf{Q}(\tilde{p}, R_{i,t})}, & \ell_t^p = 0, \end{cases} \tag{8}$$

where $\mathbf{Q}(p, R_{i,t})$ is the nearest Euclidean distance in the image plane between the pixels in p and the boundary of region $R_{i,t}$.

Smoothness cost. If two spatial or temporal adjacent superpixels have similar appearance and motion patterns, it is preferable to assign them to the same region label. To this end, we introduce the smoothness cost in (4) to model the consistences of two spatially adjacent superpixels p and q, which is calculated as

$$\begin{aligned} V_{\mathcal{S}}(\ell_t^p, \ell_t^q) \; &= \delta(\ell_t^p \neq \ell_t^q) \cdot \Big(\omega_9 \cdot \Theta_{\mathrm{c}}(p, q) \\ &+ \omega_{10} \cdot \Theta_{\mathrm{d}}(p, q) + \omega_{11} \cdot \Theta_{\mathrm{m}}(p, q)\Big), \end{aligned} \tag{9}$$

where $\delta(\ell_t^p \neq \ell_t^q) = 1$ if $\ell_t^p \neq \ell_t^q$, and 0 if $\ell_t^p = \ell_t^q$. $\Theta_{\mathrm{c}}(p, q)$ is the Euclidean distance between the the average RGB values of the pixels in p and q, $\Theta_{\mathrm{d}}(p, q)$ is the cosine distance between the average CNN features [21] of the pixels in p and q, $\Theta_{\mathrm{m}}(p, q)$ is the Euclidean distance between the superpixels p and q in the optical flow space [29], and ω_9, ω_{10}, and ω_{11} are preset parameters balancing the three terms. The smoothness cost $V_{\mathcal{X}}(\ell_t^p, \ell_{t+1}^q)$ in (4) of the temporal adjacent superpixels is similarly defined.

4. Optimization

The energy minimization problem in (1) is challenging because the objective involves three sets of variables, *i.e.*, the first term involves the clusters of region identities Φ, and the second term involves the superpixel labels $\mathbf{L}$, and the region set $\mathbf{R}$. It is difficult to optimize them simultaneously. We propose a heuristic iterative algorithm to minimize the energy. That is, we first minimize the first term $E_{\mathbf{R}}(\Phi)$ to estimate the optimal $\hat{\Phi}$ with the current $\tilde{\mathbf{R}}$. Next, we minimize the second term $E_{\Phi}(\mathbf{L}, \mathbf{R})$ to estimate the optimal $\hat{\mathbf{L}}$ and $\hat{\mathbf{R}}$ based the optimal $\hat{\Phi}$. Then if the overall energy $\hat{\mathbf{E}} = E_{\hat{\mathbf{R}}}(\hat{\Phi}) + E_{\hat{\Phi}}(\hat{\mathbf{L}}, \hat{\mathbf{R}})$ is reduced, we use the estimated $\hat{\Phi}$, $\hat{\mathbf{L}}$ and $\hat{\mathbf{R}}$ to replace the current optimal variables, *i.e.*, $\Phi^* = \hat{\Phi}$, $\mathbf{L}^* = \hat{\mathbf{L}}$ and $\mathbf{R}^* = \hat{\mathbf{R}}$; otherwise, the current ones are retained. This two steps are performed iteratively until convergence, *i.e.*, the overall energy $E_{\mathbf{R}}(\Phi) + E_{\Phi}(\mathbf{L}, \mathbf{R})$ is no longer reduced. Meanwhile, for the subproblem of minimizing $E_{\Phi}(\mathbf{L}, \mathbf{R})$ regarding the variables $\mathbf{L}$ and $\mathbf{R}$, we use the block coordinate descent algorithm to optimize them iteratively. Notably, the energy function in (1) is non-negative

Algorithm 1 Graph-to-graph energy based VOS.

```
Input: Input video; maximal number of iterations $\tau^{\text{rg}}$; maximal number of superpixel-energy iterations $\tau^{\text{sp}}$.
Generate initial regions using category-independent object proposals generating method [6] and superpixels using supervoxel method [12].
Initialize the overall energy $\mathbf{E}^* = \infty$, and $\zeta^{\text{rg}} = 0$.
while $\zeta^{\text{rg}} \leq \tau^{\text{rg}}$ do
    Construct the region-graph $\mathbf{G}^{\text{rg}}$.
    Extract dense clusters $\mathbf{C}$ on $\mathbf{G}^{\text{rg}}$ to estimate the optimal clusters of region identities $\hat{\Phi}$.
    $\zeta^{\text{sp}} = 0$.
    Initialize the energy $E_{\hat{\Phi}}(\hat{\mathbf{L}}, \hat{\mathbf{R}}) = \infty$, and the variables $\hat{\mathbf{L}} = \mathbf{L}^*$ and $\hat{\mathbf{R}} = \mathbf{R}^*$.
    while $\zeta^{\text{sp}} \leq \tau^{\text{sp}}$ do
        Construct superpixel-graph $\mathbf{G}^{\text{sp}}$ based on $\hat{\mathbf{R}}$.
        Use graph cut algorithm [2] and maximal likelihood estimation method to get $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{R}}$.
        Compute the corresponding energy $E_{\hat{\Phi}}(\tilde{\mathbf{L}}, \tilde{\mathbf{R}})$ in (4).
        if $E_{\hat{\Phi}}(\tilde{\mathbf{L}}, \tilde{\mathbf{R}}) < E_{\hat{\Phi}}(\hat{\mathbf{L}}, \hat{\mathbf{R}})$ then
            $E_{\hat{\Phi}}(\hat{\mathbf{L}}, \hat{\mathbf{R}}) = E_{\hat{\Phi}}(\tilde{\mathbf{L}}, \tilde{\mathbf{R}})$, $\hat{\mathbf{L}} = \tilde{\mathbf{L}}$, $\hat{\mathbf{R}} = \tilde{\mathbf{R}}$.
        else
            break
        end if
        $\zeta^{\text{sp}} = \zeta^{\text{sp}} + 1$.
    end while
    Compute $E_{\hat{\mathbf{R}}}(\hat{\Phi})$, and $\hat{\mathbf{E}} = E_{\hat{\mathbf{R}}}(\hat{\Phi}) + E_{\hat{\Phi}}(\hat{\mathbf{L}}, \hat{\mathbf{R}})$
    if $\hat{\mathbf{E}} < \mathbf{E}^*$ then
        $\mathbf{E}^* = \hat{\mathbf{E}}$, $\Phi^* = \hat{\Phi}$, $\mathbf{L}^* = \hat{\mathbf{L}}$, and $\mathbf{R}^* = \hat{\mathbf{R}}$.
        Calculate the corresponding optimal tubes $\mathbf{O}^*$.
    else
        break
    end if
    $\zeta^{\text{rg}} = \zeta^{\text{rg}} + 1$.
end while
Output: Optimal video object proposals $\mathbf{O}^*$.
```

with a natural lower bound of 0. This non-increasing property over the iterations ensures the convergence of this optimization algorithm. Algorithm 1 shows the main steps of the proposed graph-to-graph energy minimization.

Minimizing $E_{\mathbf{R}}(\Phi)$. We minimize the first energy term $E_{\mathbf{R}}(\Phi)$ in (1) with the given $\mathbf{R}$, *i.e.*,

$$\begin{aligned}&\text{argmin}_{\Phi}\, E_{\mathbf{R}}(\Phi) = \text{argmax}_{\mathbf{x}_1,\cdots,\mathbf{x}_{\text{N}}} \textstyle\sum_{\mathcal{C}_i \in \mathbf{C}} \mathbf{x}_i^\top \mathbf{M}\mathbf{x}_i,\\ &\text{s.t.} \forall i \in \{1, \cdots, \text{N}\}, \textstyle\sum_j x_{i,j} = 1, x_{i,j} \in \{0, 1/\beta_i\},\end{aligned} \tag{10}$$

where N is the total number of exploited dense clusters. It is difficult to optimize the above problem since we do not know the number of clusters beforehand. We adopt a greedy scheme used in [38] to solve (10). That is, we set each node on $\mathbf{G}^{\text{rg}}$ as a starting point, and search the dense structures of it, which is formulated as:

$$\begin{aligned}&\mathbf{x}_i^* = \text{argmax}_{\mathbf{x}_i}\, \mathbf{x}_i^\top \mathbf{M}\mathbf{x}_i\\ &\text{s.t.} \textstyle\sum_j x_{i,j} = 1, x_{i,j} \in \{0, 1/\beta_i\}.\end{aligned} \tag{11}$$

And then, to reduce the complexity and avoid degeneracy, we require the minimum size of dense clusters $\tilde{\beta}_i \leq \beta_i$ and relax the discrete constraint $x_{i,j} \in \{0, 1/\beta_i\}$ to be a continuous one as $x_{i,j} \in [0, 1/\tilde{\beta}_i]$. Please refer to [38] for the pairwise updating and post-processing schemes. If energy $E_{\mathbf{R}}(\Phi)$ of (10) is reduced, we use the obtained set of region identities to replace the existing one; otherwise, the existing one will be retained.

Minimizing $E_{\Phi}(\mathbf{L}, \mathbf{R})$. Given the set of region identities Φ, we minimize the second energy term $E_{\Phi}(\mathbf{L}, \mathbf{R})$ in (1), which is formulated as

$$\begin{aligned}&\underset{\mathbf{L},\mathbf{R}}{\text{argmin}} \sum_{\phi_j \in \Phi} \Big(\sum_{t=1}^{\text{T}} \sum_{R_{i,t} \in \mathcal{R}_t} \delta\big(\eta_{i,t} \in \phi_j\big) \sum_{p \in \mathcal{P}_t} D(\ell_t^p, R_{i,t})\\ &+ \sum_{t=1}^{\text{T}} \sum_{(p,q) \in \mathcal{E}_{\mathcal{S}}^{\text{sp}}} V_{\mathcal{S}}(\ell_t^p, \ell_t^q) + \sum_{t=1}^{\text{T}-1} \sum_{(p,q) \in \mathcal{E}_{\mathcal{X}}^{\text{sp}}} V_{\mathcal{X}}(\ell_t^p, \ell_{t+1}^q)\Big),\end{aligned} \tag{12}$$

where ϕ_j is the j-th cluster of the region identities in Φ. We optimize this subproblem using the block coordinate descent algorithm in two steps: 1) optimize $\mathbf{L}$ with fixed $\mathbf{R}$, and 2) optimize $\mathbf{R}$ with fixed $\mathbf{L}$ iteratively until convergence.

Specifically, to distinguish multiple regions from the background, we assign superpixel p with a binary $\{0, 1\}$ label to determine whether p belongs to a corresponding region. Let $\mathbf{L} = \{\mathcal{L}_1, \cdots, \mathcal{L}_{\text{N}}\}$ be the binary labels assigned to the superpixels, where $\mathcal{L}_j = \{\ell_t^p\}_{t=1,\cdots,\text{T}}^{p \in \mathcal{P}_t}$ is the assigned labels corresponding to the j-th cluster (tube). To exploit the temporal consistency, we optimize the labels corresponding to the region of the same tube jointly. Then, for the regions in the j-th cluster (tube) ϕ_j of Φ, the superpixel labeling problem in (12) is converted to[2]:

$$\begin{aligned}&\underset{\mathcal{L}_j}{\text{argmin}} \sum_{t=1}^{\text{T}} \sum_{R_{i,t} \in \mathcal{R}_t} \delta\big(\eta_{i,t} \in \phi_j\big) \sum_{p \in \mathcal{P}_t} D(\ell_t^p, R_{i,t})\\ &+ \sum_{t=1}^{T} \sum_{(p,q) \in \mathcal{E}_{\mathcal{S}}^{\text{sp}}} V_{\mathcal{S}}(\ell_t^p, \ell_t^q) + \sum_{t=1}^{T-1} \sum_{(p,q) \in \mathcal{E}_{\mathcal{X}}^{\text{sp}}} V_{\mathcal{X}}(\ell_t^p, \ell_{t+1}^q).\end{aligned} \tag{13}$$

We use the α-expansion based graph-cut algorithm [2] to get the labels of superpixels with the tube. In this way, we can obtain the superpixel labels $\mathcal{L}_j$ effectively.

Once the superpixel labels $\mathbf{L}$ are computed, we re-estimate the region set $\mathbf{R}$ by minimizing the energy $E_{\Phi}(\mathbf{L}, \mathbf{R})$. The two smoothness terms $V_{\mathcal{S}}(\ell_t^p, \ell_t^q)$ and

[2] Since the labelings for the superpixels belonging to regions are independent for different tubes, we can optimize the labelings of superpixels of each tube individually.

Table 1. Segmentation results on the SegTrack v2 dataset using the IoU overlap metric. Bold fonts correspond to the best performance for the *unsupervised* VOS methods. **The detail performance of each sequence/object is shown in Supplementary Materials.**

Category	Semi-Supervised				Unsupervised				
Methods	[37]	[16]	[4]	[34]	[20]	[18]	[12]	[39]	GEM
Mean per Object	71.8	67.4	35.6	74.1	65.9	45.3	51.8	69.1	**71.3**
Mean per Sequence	72.2	68.8	40.4	75.3	71.2	57.3	50.8	73.9	**75.0**
Avg.# of Proposals	N/A	N/A	N/A	N/A	60.0	10.6	336.6	121.9	339.0

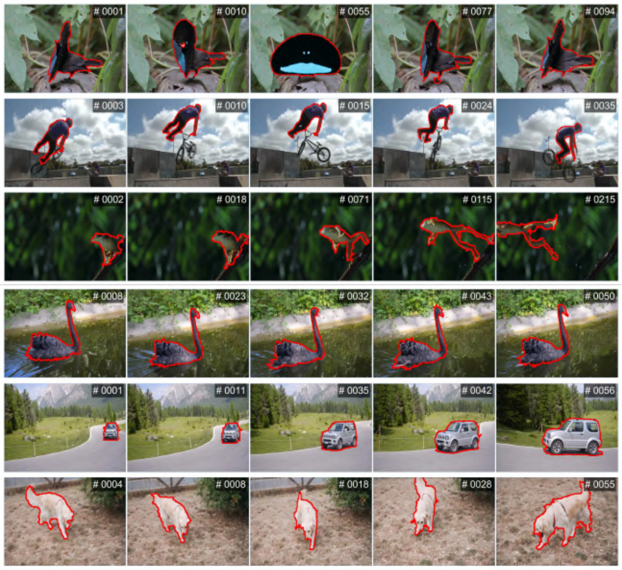

Figure 2. Segmentation results of the GEM algorithm in six sequences from SegTrack v2 (top) and DAVIS (bottom) datasets.

$V_{\mathcal{X}}(\ell_t^p, \ell_{t+1}^q)$ in (12) are fixed, and the problem is formulated as

$$\underset{\mathbf{R}}{\operatorname{argmin}} \sum_{\phi_j \in \Phi} \sum_{t=1}^{\mathrm{T}} \sum_{R_{i,t} \in \mathcal{R}_t} \delta(\eta_{i,t} \in \phi_j) \sum_{p \in \mathcal{P}_t} D(\ell_t^p, R_{i,t}). \tag{14}$$

We use the maximal likelihood estimation method to obtain the optimal regions. That is, we estimate the GMM models, logistic regression models, and extensions of the regions based on the computed superpixel labels $\mathbf{L}$. Then, if the energy $E_\Phi(\mathbf{L}, \mathbf{R})$ in (4) is reduced, we use the estimated regions to replace the existing one; otherwise, the existing regions are retained.

5. Experiments

We evaluate the proposed algorithm[3] against state-of-the-art semi-supervised and unsupervised VOS algorithms on two widely used datasets, namely SegTrack v2 [20] and DAVIS [25]. The results are presented in Table 1 and Table 2. Specifically, we compare quantitative performance of our method with eleven state-of-the-art unsupervised VOS methods [12, 18, 20, 11, 7, 30, 15, 36, 24, 39, 31]. We also compare with seven state-of-the-art semi-supervised VOS methods [37, 16, 4, 34, 8, 26, 22], that requires human annotation of the object's boundary in the first frame. Several qualitative segmentation results are shown in Figure 2.

Implementation Details. Similar to [18, 39], we use [5] to generate several object proposals in each frame of the video sequences. Meanwhile, to improve the accuracy of initial proposals, we append a patch of regions enclosed by the motion boundary and edge map [19]. We use a pre-trained VGG net [28] to extract hierarchial CNN features [21] for each pixel by combining the first 3 convolutional layers into 448 dimensional vectors.

All parameters are fixed in the experiments. The parameters are chosen empirically, *i.e.*, changing one parameter with other parameters fixed. We set the number of initial proposals in each frame to 50 in our experiments, *i.e.*, $\forall t$, $\mathrm{K}_t = 50$. The weight in (2) is set to $\omega_1 = 10^6$. For the weights in (3), we set $\omega_2 = 3.0$, $\omega_3 = 3.0$, $\omega_4 = 5.0$, and $\omega_5 = 5.0$. Meanwhile, for the weights in (5), we set $\omega_6 = 150.0$, $\omega_7 = 20.0$, and $\omega_8 = 30.0$. The weights in (9) are set to $\omega_9 = 5.0$, $\omega_{10} = 5.0$, and $\omega_{11} = 2.5$. Taking both accuracy and efficiency into account, we set the maximal number of iterations $\tau^{\text{rg}} = \tau^{\text{sp}} = 2$. The minimal size of all dense clusters is set to $\tilde{\beta}_i = 3$, $\forall i$.

SegTrack v2 Dataset. SegTrack v2 [20] consists of 14 videos with 24 objects and 947 annotated frames, which includes various challenging sequences with large appearance variations, occlusions, complex deformations, clutter background, interactions between objects, etc. We use the IoU overlap metric to evaluate the segmentation accuracy of the proposed method and the state-of-the-art unsupervised VOS methods [20, 18, 12, 39] and semi-supervised VOS methods [4, 37, 34, 16][4], presented in Table 1.

As shown in Table 1, our GEM method improves 2.2% and 1.1% IoU values per object and per sequence with 339.0 proposals on average across the videos, compared to the second best unsupervised VOS method, *i.e.*, [39]. Our method also performs comparably to the best semi-supervised VOS method [34] in terms of the mean accuracy per sequence (75.0% of GEM *vs.* 75.3% of OFL [34]).

DAVIS Dataset. The DAVIS dataset [25] comprises 50 sequences, $3,455$ annotated frames with binary pixel-level foreground/background masks. Evaluation is performed on the 480p resolution set. This dataset includes all major challenges such as background clutter, fast-motion, edge ambiguity, and out-of-view. For comprehensive evaluation, we use three measures provided by the dataset, *i.e.*, region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$ and temporal stability $\mathcal{T}$. Specifically, region similarity $\mathcal{J}$ measures the number of mislabeled pixels, which is defined as the intersection-over-union of the estimated segmentation and the ground-

[3] We will release the source codes of the proposed method after the paper is accepted.

[4] The average number of proposals is not applicable in Semi-Supervised object segmentation as the object mask in first frame is provided.

Table 2. Segmentation results on the DAVIS dataset [25] using the provided evaluation protocol. Bold fonts correspond to the best performance for the *unsupervised* VOS methods.

Measure	Semi-Supervised					Unsupervised								
	MSK [16]	JMP [8]	FCP [26]	BVS [22]	OFL [34]	NLC [7]	CVOS [30]	TRC [11]	KEY [18]	SAL [36]	FST [24]	CUT [15]	LMP [31]	GEM
Mean ↑	0.797	0.607	0.631	0.665	0.711	0.641	0.514	0.501	0.569	0.426	0.575	0.552	**0.697**	0.696
$\mathcal{J}$ Recall ↑	0.931	0.693	0.778	0.764	0.800	0.731	0.581	0.560	0.671	0.386	0.652	0.575	**0.892**	0.867
Decay ↓	0.089	0.372	0.031	0.260	0.227	0.086	0.127	0.050	0.075	0.084	0.044	**0.022**	0.056	0.058
Mean ↑	0.754	0.586	0.546	0.656	0.679	0.593	0.490	0.478	0.503	0.383	0.536	0.552	**0.663**	0.596
$\mathcal{F}$ Recall ↑	0.871	0.656	0.604	0.774	0.780	0.658	0.578	0.519	0.534	0.264	0.579	0.610	**0.783**	0.662
Decay ↓	0.090	0.373	0.039	0.236	0.240	0.086	0.138	0.066	0.079	0.072	0.065	**0.034**	0.067	0.077
$\mathcal{T}$ Mean ↓	0.218	0.131	0.285	0.316	0.221	0.356	0.243	0.327	**0.190**	0.600	0.276	0.277	0.686	0.246

truth mask. Contour accuracy $\mathcal{F}$ computes the F-measure of the contour-based precision and recall between the contour points of estimated segmentation and the ground-truth mask. Temporal stability $\mathcal{T}$ measures the smoothness and stability of the object shapes in temporal domain.

Table 2 shows that our approach performs better than most of the unsupervised methods [18, 11, 24, 7, 30, 36, 15] except LMP [31], with 5.5% and 13.6% improvements on the mean and recall values of region similarity $\mathcal{J}$, and 0.3% and 0.4% improvements on the mean and recall values of contour accuracy compared to the second best method NLC [7]. LMP [31] performs a slightly better than our GEM in terms of region similarity $\mathcal{J}$ (*i.e.*, 0.697 *vs.* 0.696 mean values of $\mathcal{J}$) and contour accuracy $\mathcal{F}$ (0.892 *vs.* 0.867 mean values of $\mathcal{F}$). Regarding the temporal stability $\mathcal{T}$, LMP [31] produces worse results than GEM, which obtains almost $3\times$ larger $\mathcal{T}$ value than GEM (*i.e.*, 0.686 *vs.* 0.246). Unlike LMP [31] using only a pair of video frames at a time, our GEM jointly considers the spatial similarities among superpixels and temporal consistencies among regions in several consecutive frames, leading to more stable and smooth segmentation results.

Our method performs on a par with most of the state-of-the-art semi-supervised methods [8, 26, 22, 34] *without any supervision*, except the most recent approach [16], see Table 2. The method of [16] performs better than other methods by using a ConvNet trained with a large amount of additional static images with *segmentation annotations*. In addition, the semi-supervised approaches, *e.g.*, MSK [16] and OFL [34], propagate the initial manual segmentation iteratively in consecutive frames and thus achieve better temporal stability than unsupervised VOS methods, *e.g.*, SAL [36] and NLC [7]. Since our method considers the spatial-temporal relations of both superpixel and region levels by constructing two graphs, *i.e.*, region-graph and superpixel-graph (see Figure 1), it produces comparable performance with the semi-supervised methods in terms of the mean temporal stability.

Ablation Study. To demonstrate the effectiveness of combing bottom-up and top-down cues in a unified process and jointly optimizing variables $(\mathbf{L}, \mathbf{R})$, we construct three variants of the GEM algorithm, denoted as GEM-NGR, GEM-NG, and GEM-NR. We denote heuristic minimizing $E_{\mathbf{R}}(\Phi)$ and $E_{\Phi}(\mathbf{L}, \mathbf{R})$ iteratively as graph-to-graph iteration, and denote optimizing $\mathbf{L}$ and $\mathbf{R}$ iteratively as region iteration, where Φ is the clusters of region identities, $\mathbf{L}$ is the superpixel labels, and $\mathbf{R}$ is the region set. GEM-NGR indicates that we do not perform both graph-to-graph and region iterations in GEM, GEM-NG corresponds to that we do not perform graph-to-graph iteration, and GEM-NR indicates that we do not perform region iteration. We evaluate GEM as well as these three methods on the SegTrack v1 dataset [33], which forms by six sequences in SegTrack v2, *i.e.*, *Girl*, *Birdfall*, *Parachute*, *Cheetah*, *Monkeydog*, and *Penguin*, and report the average IoU overlap ratio in Table 3.

Table 3. Comparisons of three variants of the GEM algorithm in the SegTrack v1 dataset using the IoU overlap metric.

Sequence/Method	GEM-NGR	GEM-NG	GEM-NR	GEM
Mean per Object	65.2	67.4	67.7	**68.1**
Mean per Sequence	66.4	68.8	68.9	**70.2**

As shown in Table 3, GEM outperforms GEM-NGR, GEM-NG, and GEM-NR in both IoU per object and IoU per sequence metrics, *i.e.*, it improves 2.9% and 3.8% of IoU per object and per sequence comparing to GEM-NGR, 0.7% and 1.4% of IoU per object and per sequence comparing to GEM-NG, and 0.4% and 1.3% of IoU per object and per sequence compared to GEM-NR. Meanwhile, GEM-NR improves 2.5% and 2.5% of IoU per object and per sequence comparing to GEM-NGR, which indicates that the graph-to-graph iteration is crucial to improve the segmentation results. Without the graph-to-graph iteration, the performance drops significantly, as there is no combination of the top down and bottom up cues. Meanwhile, Table 3 shows that GEM-NG achieves 2.2% higher IoU per object and 2.4% higher IoU per sequence than GEM-NGR. The results demonstrate that the jointly optimizing of $\mathbf{L}$ and $\mathbf{R}$ in region iteration can significantly improve the performance.

6. Conclusion

In this paper, we present a new unsupervised VOS approach based on the graph-to-graph optimization based on the combination of the bottom-up and top-down cues in a unified framework. The graph-to-graph energy objective encodes the spatial similarities among superpixels and temporal consistency among regions. An efficient heuristic iterative algorithm is proposed to minimize the energy objective to generate video object proposals. We evaluate the proposed method on two challenging benchmarks, *i.e.*, SegTrack v2 and DAVIS, to demonstrate that the proposed method achieves favorable performance against the state-of-the-art unsupervised VOS methods and comparable performance with the state-of-the-art semi-supervised methods.

Acknowledgement This material is based upon work supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number 73042CS.

References

[1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. Graph.*, 28(3):70:1–70:11, 2009. 1, 2

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001. 5

[3] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. One-shot video object segmentation. In *CVPR*, pages 221–230, 2017. 2

[4] Z. Cai, L. Wen, J. Yang, Z. Lei, and S. Z. Li. Structured visual tracking with dynamic graph. In *ACCV*, pages 86–97, 2012. 6

[5] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588, 2010. 2, 6

[6] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *TPAMI*, 36(2):222–234, 2014. 1, 5

[7] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 6, 7

[8] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Transactions on Graphics*, 34(6):195, 2015. 6, 7

[9] A. Fathi, M.-F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, pages 1–11, 2011. 1, 2

[10] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *CVPR*, pages 4083–4090, 2015. 1

[11] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, pages 1846–1853, 2012. 6, 7

[12] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010. 1, 3, 5, 6

[13] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, pages 656–671, 2014. 1, 2

[14] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, pages 3664–3673, 2017. 3

[15] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, pages 3271–3279, 2015. 6, 7

[16] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 6, 7

[17] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, pages 3442–3450, 2017. 1

[18] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 1, 2, 6, 7

[19] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *ECCV*, pages 516–529, 2012. 6

[20] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 1, 2, 6

[21] C. Ma, J. Huang, X. Yang, and M. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, pages 3074–3082, 2015. 3, 4, 6

[22] N. Marki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, pages 743–751, 2016. 1, 6, 7

[23] D. Oneata, J. Revaud, J. J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, pages 737–752, 2014. 1

[24] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstraint video. In *ICCV*, pages 1–6, 2013. 1, 3, 6, 7

[25] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 1, 6, 7

[26] F. Perazzi, O. Wang, M. H. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, pages 3227–3234, 2015. 6, 7

[27] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786, 2009. 1, 2

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[29] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, pages 2432–2439, 2010. 4

[30] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, pages 4268–4276, 2015. 6, 7

[31] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *CVPR*, pages 3386–3394, 2017. 3, 6, 7

[32] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, pages 4481–4490, 2017. 3

[33] D. Tsai, M. Flagg, and J. M. Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, pages 1–11, 2010. 7

[34] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 1, 2, 6, 7

[35] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. In *NIPS*, pages 28–36, 2011. 1, 2

[36] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, pages 3395–3402, 2015. 6, 7

[37] L. Wen, D. Du, Z. Lei, S. Z. Li, and M. Yang. JOTS: joint online tracking and segmentation. In *CVPR*, pages 2226–2234, 2015. 1, 2, 6

[38] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *CVPR*, pages 1282–1289, 2014. 3, 5

[39] F. Xiao and Y. J. Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016. 1, 3, 6

[40] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, pages 1202–1209, 2012. 3

[41] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, pages 626–639, 2012. 1, 2

[42] R. Yang, B. Ni, C. Ma, Y. Xu, and X. Yang. Video segmentation via multiple granularity analysis. In *CVPR*, pages 3010–3019, 2017. 1, 2

[43] C. Yu, H. Le, G. J. Zelinsky, and D. Samaras. Efficient video segmentation using parametric graph partitioning. In *ICCV*, pages 3155–3163, 2015. 1, 2