Sequential Clique Optimization for Video Object Segmentation

Yeong Jun Koh[1], Young-Yoon Lee[2], and Chang-Su Kim[1]

[1] School of Electrical Engineering, Korea University, Korea
yjkoh@mcl.korea.ac.kr, changsukim@korea.ac.kr
[2] Samsung Electronics Co., Ltd
yy77lee@gmail.com

Abstract. A novel algorithm to segment out objects in a video sequence is proposed in this work. First, we extract object instances in each frame. Then, we select a visually important object instance in each frame to construct the salient object track through the sequence. This can be formulated as finding the maximal weight clique in a complete k-partite graph, which is NP hard. Therefore, we develop the sequential clique optimization (SCO) technique to efficiently determine the cliques corresponding to salient object tracks. We convert these tracks into video object segmentation results. Experimental results show that the proposed algorithm significantly outperforms the state-of-the-art video object segmentation and video salient object detection algorithms on recent benchmark datasets.

Keywords: Video object segmentation, primary object segmentation, salient object detection, sequential clique optimization

1 Introduction

Video object segmentation (VOS) [1–4] is the task to segment out primary objects from the background in a video sequence, where a 'primary' object refers to the most salient one in the sequence [5, 6]. In this regard, VOS is closely related to video salient object detection (SOD) [7–10], in which the objective is to detect salient objects in a video. Note that the 'salient' objects mean that they appear frequently in the video and have dominant color and motion features. VOS can be used in many vision applications, including object recognition, action recognition, and video summarization. However, it is challenging to delineate salient objects in videos without any user annotations. Also, various factors, such as background clutter, fast motion, and object occlusion, make VOS even more difficult.

Recent object instance segmentation techniques for still images achieve remarkable performances, by employing convolutional neural networks (CNNs) [11–14]. On the other hand, many VOS techniques [15, 16] and video SOD techniques [9, 10, 17] focus on the combination of spatial and temporal results. However, the fusion processes often cause temporal inconsistency and may fail to segment out primary objects properly when either spatial or temporal results are inaccurate. Also, although these techniques can effectively extract objects with dominant color and motion features, they do not consider the appearance frequency of an object in a video sequence. In other words, they may fail to detect primary objects, which have less dominant features in each frame but appear frequently in the sequence.

In this work, we propose a novel approach to segment out foreground objects in a video sequence. First, we generate object instances in each frame. Then, we perform instance matching, by selecting one object instance from each frame, in order to construct the most salient object track. This is formulated as finding a clique in a complete k-partite graph [18] of object instances. Note that the clique should contain the instances over frames, corresponding to an identical object. Thus, the instances should be similar to one another. However, finding the optimal clique with the maximal similarity weights is NP hard. We hence develop the sequential clique optimization (SCO) process, which considers both the node energy and the edge energy. By repeating the SCO process, we can extract multiple salient object tracks. Finally, we convert these salient object tracks into VOS results in unsupervised and semi-supervised settings. Experimental results demonstrate that the proposed algorithm significantly outperforms the state-of-the-art VOS and video SOD algorithms on the DAVIS [19] and FBMS [20] datasets.

This work has the following major contributions:

- We develop the SCO process that determines a suboptimal clique efficiently with time complexity $O(NT^2)$, where T is the number of frames in a video and N is the number of instances in each frame.
- The proposed algorithm can extract multiple primary objects effectively, whereas most conventional algorithms assume a single primary object.
- The proposed algorithm provides remarkable performances on the DAVIS 2016, DAVIS 2017, and FBMS benchmark datasets.

2 Related Work

Video Object Segmentation: VOS attempts to separate foreground objects from the background in a video. Many VOS algorithms extract a single primary object. Papazoglou and Ferrari [1] generate motion boundaries using optical flows, construct a foreground model for the regions within the motion boundaries, and then use it to extract moving objects. Lee *et al.* [21] extract object proposals with the objectness scores from all frames. The proposals are clustered, and each cluster is ranked according to the objectness score. In [3, 22], object proposals are used to construct a locally connected graph, and the optimal path in the graph is determined to describe a primary object. Koh *et al.* [23] consider the recurrence property of a primary object to choose proposals. Also, saliency detection techniques are widely employed to estimate initial regions of a primary object [2, 4, 24, 25]. Wang *et al.* [24] adopt geodesic distances for saliency estimation and design an energy function to enforce the temporal smoothness of a primary object. Jang *et al.* [4] obtain foreground and background distributions by adopting the boundary prior, and dichotomize each region into the primary object or background class by minimizing a hybrid energy function. Yang *et al.* [25] use saliency maps to build an appearance model. Faktor and Irani [2] employ saliency maps as the initial distribution of random walk simulation.

Another approach to VOS is motion segmentation [20, 26–30], which clusters point trajectories. Shi and Malik [26] divide a video into motion segments using the normalized cuts. Brox and Malik [27] construct sparse long-term trajectories and cluster them. Ochs and Brox [28] convert sparse motion clusters into dense segmentation via the sparse-to-dense interpolation scheme. They also adopt the spectral clustering based on a higher-order motion model [29]. Fragkiadaki *et al.* [30] analyze trajectory discontinuities at object boundaries to improve the segmentation accuracy.

Recently, deep learning techniques have been developed for VOS [15, 16, 31–35]. Jain *et al.* [15] propose an end-to-end learning framework, which combines appearance and motion information to provide pixel-wise segmentation results for salient objects. Tokmakov *et al.* [16, 31] learn motion patterns with a fully convolutional network by employing synthetic video sequences. Deep learning models are also used in semi-supervised VOS, which requires manually annotated masks at the first frame to segment out target objects in subsequent frames [33–35]. Caelles *et al.* [33] fine-tune a CNN using user annotations to extract a target object. In [34, 35], propagation errors of segmentation masks are recovered by deep learning models.

Salient Object Detection: Early SOD algorithms [36–42] for still images are based on bottom-up models, which use global or local contrast of image features. Some algorithms [40–42] adopt *a priori* knowledge, such as the boundary prior that boundary regions tend to belong to the background and thus be less salient than center regions. Recently, deep learning techniques have been adopted prevalently for SOD. Many deep fully convolutional networks are trained in an end-to-end manner to yield pixel-wise saliency maps [43–45]. Also, an instance-level segmentation algorithm for salient objects was proposed in [14], which uses both saliency maps and object proposals.

Image SOD has been extended to video SOD [7,9,17,46,47]. Kim *et al.* [7] produce a spatiotemporal saliency map via the pixel-wise multiplication of spatial and temporal saliency maps. In the multiplication, adaptive weights can be used to yield more robust results. For instance, Fang *et al.* [46] fuse spatial and temporal maps using entropy-based uncertainty weights. Also, Yang *et al.* [17] generate six kinds of saliency maps and superpose them adaptively. Chen *et al.* [9] combine color and motion saliency maps based on the salient foreground model and the non-salient background model.

Some algorithms [8, 10, 48] exploit spatial and temporal features jointly to detect spatiotemporally salient regions. Wang *et al.* [48] propose the gradient flow field to merge intra-frame boundaries and inter-frame features. Kim *et al.* [8] exploit spatial and temporal features in the random walk with restart framework. Wang *et al.* [10] design two networks for static saliency and dynamic saliency, respectively. They feed the output of the static network into the dynamic network to obtain a saliency map.

3 Proposed Algorithm

We segment primary objects in a sequence of video frames $\mathcal{I} = \{I_1, \ldots, I_T\}$. The output is the corresponding sequence of pixel-wise maps, which locate the primary objects in the frames. Fig. 1 shows an overview of the proposed algorithm. First, we generate object instances in each frame. Second, we construct a complete k-partite graph using the set of object instances. Third, we extract salient object tracks by finding cliques in the graph and convert the tracks into VOS results.

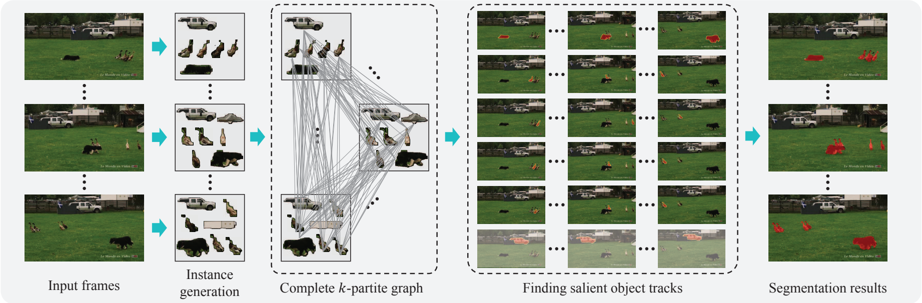


Fig. 1. An overview of the proposed algorithm.

3.1 Generating Object Instances

To detect object instances without manual annotations, we employ the instance-aware semantic segmentation method FCIS [11]. FCIS measures the category-wise detection scores of region proposals, generated by the network in [49], and segments out foreground regions from the proposals. We choose only the proposals whose detection scores are higher than 0.5 and declare the corresponding foreground regions as object instances. For each proposal, we use the maximum of the category-wise scores, since the purpose of the proposed algorithm is to segment out salient objects in videos regardless of their categories. Fig. 2(b) shows object instances in a frame. The set of detected object instances in frame I_t is denoted by $\mathcal{O}_t = \{o_{t,\theta} \,|\, \theta \in \mathbb{N}_{N_t}\}$, where $\mathbb{N}_m = \{1, 2, \ldots, m\}$ is the finite index set and N_t is the number of object instances in frame I_t. The θth object instance $o_{t,\theta}$ in frame I_t has two attributes: saliency score $s_{t,\theta}$ and feature vector $\mathbf{f}_{t,\theta}$.

To determine the saliency score $s_{t,\theta}$, we simply estimate a foreground distribution based on the boundary prior. We over-segment a frame into superpixels using [50], and construct a 4-ring graph of superpixels. We compute edge weights by summing up the average LAB color difference and the average optical flow difference between two superpixels. We then obtain the background distribution using the random walk with restart (RWR) simulation [51], where only superpixels at the frame boundary are assigned nonzero restart probabilities. Finally, we invert the background distribution to yield the foreground distribution, as illustrated in Fig. 2(c). We then compute $s_{t,\theta}$ by averaging the foreground probabilities within the instance $o_{t,\theta}$.

Also, we construct the feature vector $\mathbf{f}_{t,\theta}$ using the bag-of-visual-words (BoW) approach [52]. We quantize the LAB colors, extracted from the 40 training sequences in the VSB100 dataset [53], into 300 codewords using the K-means algorithm. We then construct the histogram of the codewords for the pixels within $o_{t,\theta}$, and normalize it into the feature vector $\mathbf{f}_{t,\theta}$.

3.2 Finding Salient Object Tracks

Problem: The set of all object instances, $\mathcal{V} = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_T$, includes non-salient objects, as well as salient ones. From $\mathcal{V}$, we extract as many salient objects as possible,

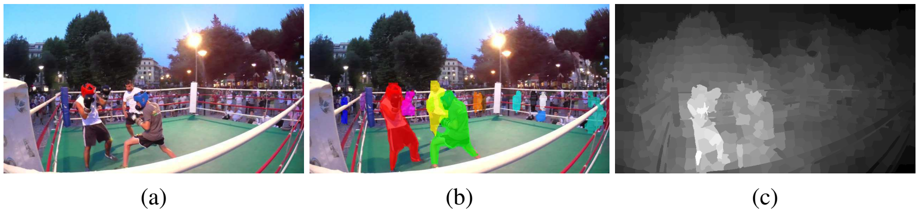


Fig. 2. Object instance generation in the “Boxing-fisheye” sequence: (a) input frame, (b) object instances, and (c) foreground distribution.

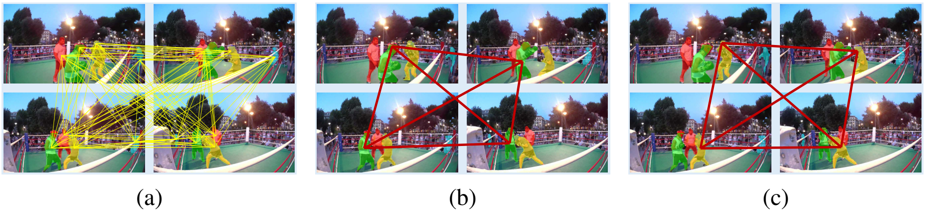


Fig. 3. Illustration of finding salient object tracks over four frames in “Boxing-fisheye”: (a) complete 4-partite graph, (b) 1st salient object track Θ_1, and (c) 2nd salient object track Θ_2.

while excluding non-salient ones, assuming that a salient object should have dominant features in each frame and appear frequently through the sequence. To this end, we construct the most salient object track, by selecting an object instance in each frame, which corresponds to one identical salient object. Then, after removing all instances in the track from $\mathcal{V}$, we repeat the process to extract the next salient track, and so on.

Sequential Clique Optimization: Using the set of object instances $\mathcal{V} = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_T$, we construct a complete k-partite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Thus, $\mathcal{V}$ becomes the node set, and each object instance becomes a node in the graph $\mathcal{G}$. Since $\mathcal{O}_t \cap \mathcal{O}_\tau = \emptyset$ for $t \neq \tau$, $\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_T$ form a partition of $\mathcal{V}$. Also, we define the edge set as $\mathcal{E} = \{(o_{t,i}, o_{\tau,j}) \,|\, t \neq \tau\}$. In other words, every pair of object instances in different frames are connected by an edge in $\mathcal{E}$, whereas two instances in the same frame are not adjacent in the graph $\mathcal{G}$. As a result, $\mathcal{G}$ is complete k-partite [18], where $k = T$. For example, Fig. 3(a) illustrates the complete k-partite graph for four frames, *i.e.* $k = 4$. We assign a weight to edge $(o_{t,i}, o_{\tau,j})$ by

$$w(o_{t,i}, o_{\tau,j}) = \exp\left(-\frac{d_{\chi^2}(\mathbf{f}_{t,i}, \mathbf{f}_{\tau,j})}{\sigma^2}\right) \tag{1}$$

where $\sigma^2 = 0.01$ is a scaling parameter and d_{χ^2} denotes the chi-square distance.

To extract the most salient object, we perform the instance matching by selecting one object instance (one node) from each frame (each node subset) $\mathcal{O}_t$. This process of finding an object track is equivalent to finding a clique in the graph $\mathcal{G}$. Notice that selecting one node from each frame satisfies the condition of a clique [18]: every pair of nodes within the clique are adjacent. In the clique, which represents the track of an identical object in the video sequence, the features of the member nodes should be similar to one another. Therefore, we determine the clique to maximize the sum of the edge weights. Let $\Theta = \{\theta_t\}_{t=1}^{T}$ denote a clique, which is represented by the sequence of the node indices in the clique. Here, $\theta_t \in \mathbb{N}_{N_t}$ is the index of the selected node from the tth frame $\mathcal{O}_t$. Then, we define the similarity $E_{\text{similarity}}(\Theta)$ of clique Θ as

$$E_{\text{similarity}}(\Theta) = \sum_{t=1}^{T} \sum_{\tau=1,\tau\neq t}^{T} w(o_{t,\theta_t}, o_{\tau,\theta_\tau}), \tag{2}$$

which is the sum of all edge weights in Θ. We attempt to maximize the similarity, assuming that the features of an identical object do not change drastically over frames.

Also, object instances in a clique, representing a salient object track, should have high saliency scores. We hence define the saliency $E_{\text{saliency}}(\Theta)$ of clique Θ as

$$E_{\text{saliency}}(\Theta) = \sum_{t=1}^{T} s_{t,\theta_t}. \tag{3}$$

We attempt to find the maximal weight clique Θ^* that maximizes the similarity energy:

$$\Theta^* = \arg\max_{\Theta} E_{\text{similarity}}(\Theta) \tag{4}$$

subject to the constraint that the saliency $E_{\text{saliency}}(\Theta)$ is also high. However, even the unconstrained problem in (4) is NP hard [54,55]. There are $N_1 \times N_2 \times \cdots \times N_T$ possible cliques, which make the exhaustive search infeasible. Some approximation methods, *e.g.* multi-greedy heuristics [56], local search [55], and binary integer program [57], have been developed to obtain suboptimal cliques in complete k-partite graphs. But, these methods are still computationally expensive and do not consider the node energy (*i.e.* the saliency E_{saliency} in this work). Instead, we develop an efficient optimization technique, called SCO, to find the clique that considers both the node energy E_{saliency} and the edge energy $E_{\text{similarity}}$.

In SCO, we first initialize the clique $\Theta^{(0)}$ to maximize the saliency E_{saliency} in (3). Specifically, the tth element in $\Theta^{(0)}$ is determined by

$$\theta_t^{(0)} = \arg\max_{\theta \in \mathbb{N}_{N_t}} s_{t,\theta}. \tag{5}$$

Then, at iteration κ, we update $\theta_t^{(\kappa)}$, by selecting the node that is the most similar to the nodes in the other frames,

$$\theta_t^{(\kappa)} = \arg\max_{\theta \in \mathbb{N}_{N_t}} \sum_{\tau=1,\tau\neq t}^{T} w(o_{t,\theta}, o_{\tau,\theta_\tau}), \tag{6}$$

and then set θ_t to be $\theta_t^{(\kappa)}$ for each t sequentially from 1 to T. We repeat this sequential update of the nodes in all frames, until $\Theta^{(\kappa)} = \{\theta_t^{(\kappa)}\}_{t=1}^{T}$ is unaltered from $\Theta^{(\kappa-1)} = \{\theta_t^{(\kappa-1)}\}_{t=1}^{T}$. This process is theoretically guaranteed to converge, since $E_{\text{similarity}}(\Theta^{(\kappa)})$ is a monotonically increasing function of κ. To summarize, SCO initializes the clique to maximize the saliency E_{saliency} and then refines it iteratively to

Algorithm 1 (SCO) Sequential Clique Optimization

```
Input: Sets of object instances $\mathcal{V} = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_T$
Construct a complete $k$-partite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
for each frame $I_t$ do
    Initialize the node index in clique $\Theta$ via
    $\theta_t \leftarrow \arg\max_{\theta \in \mathbb{N}_{N_t}} s_{t,\theta}$
end for
repeat
    for each frame $I_t$ do
        Update the node index via
        $\theta_t \leftarrow \arg\max_{\theta \in \mathbb{N}_{N_t}} \sum_{\tau=1,\tau \neq t}^{T} w(o_{t,\theta}, o_{\tau,\theta_\tau})$
    end for
until node indices are unaltered
Output: Optimized clique $\Theta = \{\theta_1, \theta_2, \ldots, \theta_T\}$
```

achieve a local maximum of $E_{\text{similarity}}$. Thus, at the initialization, the clique consists of salient object instances over frames, which may not represent an identical object. However, as the iteration goes on, the clique converges to a salient object track, in which the nodes represent an identical object and thus exhibit high similarity weights in general. **Algorithm 1** summarizes the proposed SCO technique. In most cases, less than 10 iterations are required for the convergence.

Let Θ_1 denote the most salient object track, obtained by this SCO process. To extract the next track Θ_2, we exclude the nodes in Θ_1 from $\mathcal{G}$ and perform SCO again. This is repeated to yield the set of tracks, $\{\Theta_1, \Theta_2, \ldots, \Theta_M\}$, until no node remains in $\mathcal{G}$. In general, if $p < q$, Θ_p is more salient than Θ_q. Thus, the subscript p in Θ_p is the saliency rank of the track. Figs. 3(b) and (c) illustrate the first two tracks Θ_1 and Θ_2.

Postprocessing The track selection is greedy in the sense that, if an object instance is mistakenly included in a track Θ_p, it cannot be included in a later track Θ_q even though it indeed belongs to Θ_q. To alleviate this problem, we perform postprocessing to maximize the sum of the similarities

$$\sum_{m=1}^{M} E_{\text{similarity}}(\Theta_m) \tag{7}$$

as follows. At each frame I_t, we match object instances in $\mathcal{O}_t$ to the tracks in $\{\Theta_p\}_{p=1}^{M}$. The matching cost $C(o_{t,i}, \Theta_p)$ between an instance $o_{t,i}$ and a track Θ_p is defined as the sum of the feature distances from $o_{t,i}$ to all object instances in Θ_p, except for the instance at the same frame t. After computing the matching costs, we find the optimal matching pairs using the Hungarian algorithm [58], and update the tracks to include the matched instances. This is performed for all frames. As a result, we obtain the set of the refined salient object tracks $\{\tilde{\Theta}_1, \tilde{\Theta}_2, \ldots, \tilde{\Theta}_M\}$.

Disappearance Detection Also, we detect disappearing events for each refined salient object track. When an object disappears or is fully occluded at some frames, noisy objects are selected at those frames. Given a refined salient object track $\tilde{\Theta} = \{\tilde{\theta}_t\}_{t=1}^{T}$, we determine whether to discard $o_{t,\tilde{\theta}_t}$ at frame t from $\tilde{\Theta}$. To this end, for each $\tau \neq t$, we compare the weight $w(o_{\tau,\tilde{\theta}_\tau}, o_{t,\tilde{\theta}_t})$ against the average weight. Specifically, we count the number of object instances $o_{\tau,\tilde{\theta}_\tau}$ for $\tau \neq t$, which satisfy $w(o_{\tau,\tilde{\theta}_\tau}, o_{t,\tilde{\theta}_t}) < \frac{1}{N-2}\sum_{k=1,k\neq t}^{T} w(o_{\tau,\tilde{\theta}_\tau}, o_{k,\tilde{\theta}_k})$. If the number is larger than $0.7T$, we declare $o_{t,\tilde{\theta}_t}$ to be noisy and discard it.

3.3 Segmentation Results

Using the object tracks in $\{\tilde{\Theta}_1, \tilde{\Theta}_2, \ldots, \tilde{\Theta}_M\}$, we generate a pixel-wise segmentation map for each frame, which delineates primary objects in the frame. We propose four schemes to yield final segmentation results: Proposed-F, Proposed-O, and Proposed-M for unsupervised VOS and Proposed-S for semi-supervised VOS.

- **Proposed-F:** The first track $\tilde{\Theta}_1$ extracts the primary object in a video in general. Thus, Proposed-F selects $\tilde{\Theta}_1$. However, it may fail to extract spatially connected objects. For example, given a motorbike and its rider, it may detect only one of them. Therefore, Proposed-F additionally picks another salient object track $\tilde{\Theta}_p$, only when $\tilde{\Theta}_1$ and $\tilde{\Theta}_p$ are spatially adjacent in most frames in a video.
- **Proposed-O:** The aforementioned Proposed-F is an offline approach, which constructs the global T-partite graph for an entire video. In contrast, Proposed-O is an online approach, which uses the t-partite graph for frames $1, \ldots, t$ to obtain the segmentation result for the current frame t. In other words, Proposed-O uses the information in the current and past frames only to yield the segmentation result.
- **Proposed-M:** To handle multiple primary objects, which are not spatially connected, we choose multiple tracks from $\{\tilde{\Theta}_1, \tilde{\Theta}_2, \ldots, \tilde{\Theta}_M\}$. We compute the mean saliency score of object instances in each track, and discard the tracks whose mean scores are lower than a pre-specified threshold δ. We fix $\delta = 0.1$ in all experiments.
- **Proposed-S:** Proposed-S is for semi-supervised VOS, which chooses the ground-truth segment in the first frame and fixes it in SCO. Proposed-S is based on the online approach, Proposed-O. Moreover, we warp a segment result to the next frame using optical flow, and then add the warped segment to the set of object instances.

Finally, we improve the segmentation qualities of object instances in the selected tracks using a two-class Markov random field (MRF) optimizer in [59].

3.4 Complexity Analysis

Let us analyze the computational complexity of the proposed SCO process. For the convenience of analysis, we fix the number of object instances in each frame to N. Note that SCO has two steps: initialization and update. In the initialization, $N-1$ comparisons are made to find the maximum saliency in each frame, requiring $O(NT)$ comparisons in total. In the update step, $N(T-2)$ additions and $(N-1)$ comparisons are performed for each frame in one iteration. Thus, the update step demands $O(KNT^2)$ complexity, where K is the number of iterations and is restricted to be less than 10 in this work.

Table 1. Comparison with the conventional VOS algorithms on the DAVIS 2016 dataset. The best results are boldfaced, and the second best ones are underlined.

	Region similarity $\mathcal{J}$		Contour accuracy $\mathcal{F}$	
Algorithm	Mean	Recall	Mean	Recall
NLC [2]	0.641	0.731	0.593	0.658
CVOS [61]	0.514	0.581	0.490	0.578
TRC [30]	0.501	0.560	0.478	0.519
MSG [28]	0.543	0.636	0.525	0.613
KEY [21]	0.569	0.671	0.503	0.534
SAL [24]	0.426	0.386	0.383	0.264
FST [1]	0.575	0.652	0.536	0.579
ACO [4]	0.531	0.611	0.504	0.558
LMP [16]	0.697	0.829	0.663	0.783
FSEG [15]	0.716	0.877	0.658	0.790
ARP [23]	0.763	0.892	0.711	0.828
Proposed-F	**0.796**	**0.947**	**0.770**	**0.912**
Proposed-O	0.783	0.932	0.758	0.896
Proposed-M	0.787	0.928	0.764	0.895

We repeat the SCO process N times to extract N object tracks. Thus, the complexity is $O(KN^2T^2)$. Finally, in the postprocessing, the Hungarian matching of $O(N^3)$ complexity is performed for each frame. Hence the complexity of the postprocessing is $O(N^3T)$. The overall complexity of the proposed algorithm can be approximated to $O(KN^2T^2)$, since T is larger than N in general. This complexity is significantly lower than the binary integer program in [57], which requires $O(2^{T^2N^2})$ complexity in the worst case because of the depth-first node selection [60]. Moreover, the proposed SCO yields better segmentation performance than [57], as will be shown in Section 4.

4 Experimental Results

Given a video sequence, the proposed algorithm can yield a segmentation mask for each frame, which delineates primary objects at the pixel level. Hence, we compare the proposed algorithm with the conventional VOS algorithms in [1, 2, 4, 15, 16, 21, 23, 24, 28, 30, 33, 35, 61, 62] and the conventional SOD algorithms in [8, 10, 24, 48, 63–65], which also extract primary or salient objects from each frame at the pixel level. For the comparison, we use the DAVIS dataset [19] and the FBMS dataset [20].

DAVIS Dataset [19]: It has two versions, DAVIS 2016 and DAVIS 2017. DAVIS 2016 is a benchmark to evaluate VOS algorithms. It consists of 50 video sequences, which are divided into training and test sequences. We assess the proposed algorithm using both the training and test sequences. Each sequence contains a single object or spatially connected objects, *e.g.* a motorbike and its rider, which appear repeatedly in the sequence. The spatially connected objects are also regarded as a primary object.

DAVIS 2016 was extended to DAVIS 2017, which is for semi-supervised VOS. It is composed of the train-validation, test-develop, and test-challenge subsets, which contain 90, 30, and 30 videos, respectively. We evaluate the proposed algorithm on the train-validation set. Note that DAVIS 2017 is more challenging than DAVIS 2016,

Table 2. Comparison with the conventional VOS algorithms on the DAVIS 2017 dataset. The best results are boldfaced, and the second best ones are underlined.

	Region similarity $\mathcal{J}$		Contour accuracy $\mathcal{F}$	
Algorithm	Mean	Recall	Mean	Recall
A. Unsupervised VOS				
NLC [2]	0.514	0.555	0.486	0.494
FST [1]	0.496	0.529	0.480	0.468
ACO [4]	0.450	0.464	0.448	0.430
ARP [23]	0.633	0.729	0.612	0.678
Proposed-F	0.685	0.792	0.677	0.773
Proposed-O	0.674	0.785	0.663	0.759
Proposed-M	**0.714**	**0.839**	**0.705**	**0.813**
B. Semi-supervised VOS				
OSVOS [33]	0.566	0.638	0.639	0.738
OnAVOS [62]	0.616	0.674	0.691	0.754
Proposed-S	**0.665**	**0.797**	**0.688**	**0.821**

since multiple objects, which are not connected to one another, correspond to different targets. The union region of those multiple instances is regarded as the ground for the evaluation of the unsupervised algorithms, while multiple instance annotations are used for that of the semi-supervised algorithms.

FBMS Dataset: The FBMS dataset [20] is for segmenting out moving objects in videos, where multiple objects are labeled as the ground-truth. It consists of 59 video sequences, which are split into 29 training and 30 test sequences. We assess the proposed algorithm using the test sequences.

4.1 Comparison with Video Object Segmentation Techniques

We compare the proposed algorithm with the conventional VOS algorithms, by employing the metrics of the region similarity $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ [19]. The region similarity $\mathcal{J}$ is defined as the intersection over union (IoU) ratio $\mathcal{J} = \frac{|\mathcal{S}_\mathrm{p} \cap \mathcal{S}_\mathrm{gt}|}{|\mathcal{S}_\mathrm{p} \cup \mathcal{S}_\mathrm{gt}|}$, where $\mathcal{S}_\mathrm{p}$ and $\mathcal{S}_\mathrm{gt}$ are an estimated segment and the ground-truth. Also, the contour accuracy $\mathcal{F}$ is the F-measure, which is the harmonic mean of contour precision and recall rates. In these metrics, there are two statistics: 'Mean' measures the average score and 'Recall' denotes the proportion of the frames whose scores are higher than 0.5.

Evaluation on DAVIS 2016 Dataset: Table 1 compares the results on DAVIS 2016 dataset. The scores of the conventional algorithms are from the DAVIS dataset website [19]. All three versions of the proposed algorithm (Proposed-F, Proposed-O, and Proposed-M) outperform all conventional algorithms. Note that even the online version Proposed-O performs better than the state-of-the-art algorithm ARP [23], even though ARP is an offline approach. Also, Proposed-F surpasses all conventional algorithms significantly, *e.g.* by convincing margins of 3.3% and 5.9% against ARP in terms of Mean $\mathcal{J}$ and Mean $\mathcal{F}$. Especially, Proposed-F yields a very high recall score of the region similarity $\mathcal{J}$, which is almost as high as 95%. As compared with Proposed-F, Proposed-M provides lower performances, since it selects non-primary objects, as well as primary ones, in some videos.

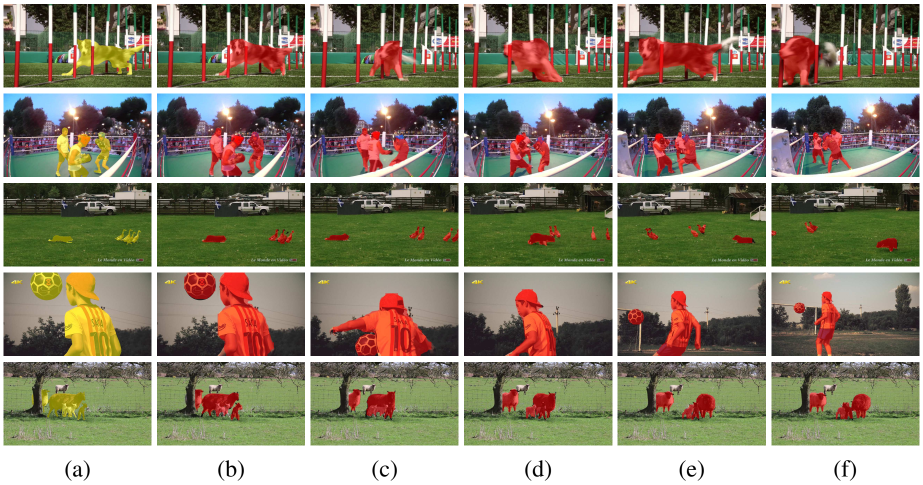


Fig. 4. VOS results of Proposed-M on the DAVIS 2017 dataset: (a) the ground-truth of primary objects in 1st frames, (b) the detection results in 1st frames, and (c)∼(f) the detection results in the subsequent frames. From top to bottom, "Dog-agility," "Boxing-fisheye," "Dog-gooses," "Kid-football," and "Sheep." Detected regions are depicted in red.

Table 3. Comparison of the Mean $\mathcal{J}$ scores on the FBMS dataset. The best result is boldfaced, and the second best one is underlined.

Video	[1]	[3]	[2]	[4]	[23]	Proposed-F	Proposed-M
Average	0.555	0.473	0.445	0.542	0.598	<u>0.625</u>	**0.686**

Evaluation on DAVIS 2017 Dataset: Table 2 compares the proposed algorithm with the conventional unsupervised algorithms [1,2,4,23] and semi-supervised ones [33,62] on DAVIS 2017. The train-validation set and validation set are used for evaluating unsupervised and semi-supervised algorithms, respectively. We compute the results of [1, 2, 4, 23] using the source codes, provided by the respective authors, and take the scores of [33,62] from the DAVIS website [19]. All unsupervised algorithms yield lower scores on DAVIS 2017 than on DAVIS 2016, since DAVIS 2017 is more challenging due to multiple primary objects. Nevertheless, the three versions of the proposed algorithm provide the best results in all metrics. Especially, Proposed-M undergoes the least degradation in the performance. This indicates that the proposed algorithm extracts multiple primary objects more reliably than the conventional ones. Fig. 4 presents examples of segmentation results.

Also, notice that Proposed-S outperforms the conventional semi-supervised algorithms [33, 62], even though [33, 62] involve the fine-tuning, which requires the high computational complexity.

Evaluation on FMBS Dataset: Table 3 compares the Mean $\mathcal{J}$ scores on the FBMS dataset. The scores of the conventional algorithms are from [23]. Compared with the state-of-the-art algorithm [23], Proposed-M improves the performance by 8.8%.

Table 4. Comparison of the proposed SCO with the conventional matching techniques [3, 57] on the DAVIS 2016 and DAVIS 2017 datasets.

	DAVIS2016		DAVIS2017	
Algorithm	Mean $\mathcal{J}$	Recall $\mathcal{J}$	Mean $\mathcal{J}$	Recall $\mathcal{J}$
FCIS+ [3]	0.737	0.869	0.635	0.734
FCIS+ [57]	0.710	0.829	0.623	0.725
FCIS + SCO + Proposed-M w/o PPMRF	0.769	0.921	0.694	0.827
FCIS + SCO + Proposed-M	0.787	0.928	0.714	0.839
FCIS + SCO + Oracle	0.799	0.959	0.769	0.923
FCIS + SCO + Proposed-M + DF	0.755	0.878	0.666	0.775

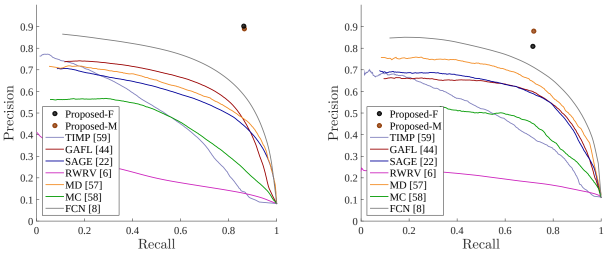
Efficacy of SCO: We analyze the efficacy of the proposed SCO. Note that SCO yields multiple salient object tracks, which are used to produce VOS results. We compare SCO with the conventional matching techniques for primary object segmentation [3] and multiple object tracking [57]. More specifically, given object instances from FCIS [11], we obtain multiple object tracks by employing the conventional techniques [3, 57]. Since [3] is designed to yield a single object track, we repeatedly perform [3] to obtain multiple object tracks. On the other hand, [57] solves the binary integer problem to produce multiple suboptimal cliques directly. Table 4 compares the proposed SCO with these conventional techniques. For all three methods, we produce segmentation results from the multiple object tracks using the method of Proposed-M.

We see that the proposed SCO outperforms the conventional methods [3, 57] significantly, even when the postprocessing and MRF (PPMRF) are not applied. Also, we perform oracle experiments for the performance upper bounds: we obtain segmentation results by matching object instances with the ground-truth segments. The proposed algorithm yields scores close to these oracle results. Finally, we use deep features (DF) instead of color-based bag-of-visual-words to compute edge weights in the graph. To this end, we extract feature maps by concatenating outputs of conv1, conv3, and conv5 in VGG-16 [66]. To generate a feature of an object instance, we average the values of pixels within the object for each channel of the feature maps. DF degrades the performance in this application, since deep semantic features undesirably yield high similarity weights between different objects in the same class.

4.2 Comparison with Salient Object Detection Techniques

To assess SOD results, we adopt three performance metrics: precision-recall (PR) curves, F-measure, and mean absolute error (MAE). The precision is the ratio $\frac{|\mathcal{S}_{\mathrm{p}} \cap \mathcal{S}_{\mathrm{gt}}|}{|\mathcal{S}_{\mathrm{p}}|}$ and the recall is $\frac{|\mathcal{S}_{\mathrm{p}} \cap \mathcal{S}_{\mathrm{gt}}|}{|\mathcal{S}_{\mathrm{gt}}|}$, where $\mathcal{S}_{\mathrm{p}}$ and $\mathcal{S}_{\mathrm{gt}}$ are an estimated result and the ground-truth, respectively. F-measure is defined as the harmonic mean of the precision and the recall, *i.e.* $\text{F-measure} = \frac{(1+\beta^2)\cdot\text{precision}\cdot\text{recall}}{\beta^2\cdot\text{precision}+\text{recall}}$ where β^2 is set to 0.3 as in [10]. Also, MAE is defined as the average of pixel-wise differences between $\mathcal{S}_{\mathrm{p}}$ and $\mathcal{S}_{\mathrm{gt}}$.

Fig. 5 compares the proposed algorithm with the conventional SOD algorithms for still images [63, 64] and video sequences [8, 10, 24, 48, 65]. The scores of the conventional algorithms are from [10]. The conventional algorithms use thresholds to binarize continuous saliency maps to compute precision and recall rates. Thus, in Fig. 5, the PR curves are obtained by varying the thresholds from 0 to 255. In contrast, the proposed



(a) (b)

Fig. 5. Comparison of the precision-recall performances of the proposed algorithm with those of the conventional algorithms: (a) DAVIS 2016 and (b) FBMS datasets.

Table 5. Comparison with the conventional SOD algorithms. The best results are boldfaced.

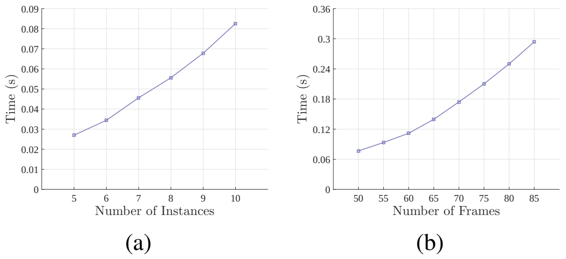
	DAVIS 2016		FBMS	
Algorithm	F-measure	MAE	F-measure	MAE
MD [63]	0.561	0.129	0.646	0.105
MC [64]	0.440	0.176	0.479	0.195
TIMP [65]	0.454	0.185	0.444	0.177
GAFL [48]	0.624	0.098	0.596	0.133
SAGE [24]	0.559	0.101	0.589	0.131
RWRV [8]	0.231	0.246	0.227	0.260
FCN [10]	0.699	0.064	0.696	0.077
Proposed-F	**0.896**	**0.017**	0.801	0.112
Proposed-M	0.888	0.020	**0.816**	**0.072**

algorithm provides a single binary map for primary objects, without requiring a threshold. Therefore, the performance of Proposed-F or Proposed-M is given by a single dot for the pair of the average precision and recall. Both Proposed-F and Proposed-M significantly surpass all conventional algorithms on both DAVIS and FBMS datasets.

Table 5 compares the F-measure and MAE performances. The proposed algorithm yields only one F-measure score, corresponding to the dot in Fig. 5. In contrast, each conventional algorithm yields 256 F-measure scores by varying the binarization threshold. Its maximum F-score is reported in Table 5 for impartial comparison. The proposed algorithm outperforms the conventional algorithms significantly. For example, Proposed-M yields about 0.19 and 0.12 higher F-measure scores than the state-of-the-art algorithm [10] on the DAVIS and FBMS datasets, respectively.

4.3 Running Time Analysis

We measure the running times of the SCO algorithm for finding cliques in a complete k-partite graph. In this test, we use the "Boxing-fisheye" sequence in the DAVIS 2017 dataset. Also, we use a computer with a 2.6GHz CPU. The running time of the proposed SCO algorithm is affected by two factors: 1) the number N of object instances in a frame and 2) the number T of frames in a sequence. Fig. 6(a) shows the running times according to N, when T is fixed to 50. Fig. 6(b) plots the running times according to



(a) (b)

Fig. 6. The running times according to the number of (a) object instances and (b) frames.

Table 6. Running times in seconds per frame (SPF).

	FCIS	Optical flow	Saliency estimation	Feature extraction	MRF	Total
Time (SPF)	0.24	0.93	1.16	0.15	0.96	3.44

T, when N is limited to 10. We see that the proposed algorithm is faster than the binary integer program in [57], which consumes about 1 second when $N = 10$ and $T = 50$.

Table 6 lists the average running times per frame on DAVIS 2016. The proposed algorithm performs FCIS [11] for generating object instances and also the optical flow estimation, saliency estimation, feature extraction at each frame. Then, it does SCO for the global optimization. SCO takes 0.304 second for the entire sequence, which is negligible. Then, the proposed algorithm also performs the MRF optimization at each frame. In total, the proposed algorithm takes 3.44 seconds per frame (SPF). It is much faster than the conventional deep-learning-based VOS algorithms [15] and [16], which take about 18 SPF and 7 SPF, respectively.

5 Conclusions

We proposed a novel algorithm to segment out primary objects in a video sequence, by solving the problem of finding cliques in a complete k-partite graph. We first generated object instances in each frame. Then, we chose a salient instance from each frame to construct the salient object track. For this purpose, we developed the SCO technique to consider both the saliency and similarity energies. By applying SCO repeatedly, we obtained multiple salient object tracks. Experimental results showed that the proposed algorithm significantly outperforms the state-of-the-art VOS and video SOD algorithms.

Acknowledgement

This work was supported partly by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2018-2016-0-00464) supervised by the Institute for Information & communications Technology Promotion, and the National Research Foundations of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A10055037 and No. NRF-2018R1A2B3003896).

References

1. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: ICCV. (2013) 1777–1784
2. Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: BMVC. (2014)
3. Zhang, D., Javed, O., Shah, M.: Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In: CVPR. (2013) 628–635
4. Jang, W.D., Lee, C., Kim, C.S.: Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In: CVPR. (2016) 696–704
5. Koh, Y.J., Jang, W.D., Kim, C.S.: POD: discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In: CVPR. (2016) 1068–1076
6. Koh, Y.J., Kim, C.S.: Unsupervised primary object discovery in videos based on evolutionary primary object modeling with reliable object proposals. IEEE Trans. Image Process. **26**(11) (2017) 5203–5216
7. Kim, W., Jung, C., Kim, C.: Spatiotemporal saliency detection and its applications in static and dynamic scenes. IEEE Trans. Circuits Syst. Video Technol. **21**(4) (2011) 446–456
8. Kim, H., Kim, Y., Sim, J.Y., Kim, C.S.: Spatiotemporal saliency detection for video sequences based on random walk with restart. IEEE Trans. Image Process. **24**(8) (2015) 2552–2564
9. Chen, C., Li, S., Wang, Y., Qin, H., Hao, A.: Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion. IEEE Trans. Image Process. **26**(7) (2017) 3156–3170
10. Wang, W., Shen, J., Shao, L.: Video salient object detection via fully convolutional networks. IEEE Trans. Image Process. **27**(1) (2018) 38–49
11. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR. (2017) 2359–2367
12. Arnab, A., Torr, P.H.: Pixelwise instance segmentation with a dynamically instantiated network. In: CVPR. (2017) 44–450
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. (2017) 2980–2988
14. Li, G., Xie, Y., Lin, L., Yu, Y.: Instance-level salient object segmentation. In: CVPR. (2017) 2386–2395
15. Jain, S.D., Xiong, B., Grauman, K.: FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: CVPR. (2017) 3664–3673
16. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: CVPR. (2017) 3386–3394
17. Yang, J., Zhao, G., Yuan, J., Shen, X., Lin, Z., Price, B., Brandt, J.: Discovering primary objects in videos by saliency fusion and iterative appearance estimation. IEEE Trans. Circuits Syst. Video Technol. **26**(6) (2016) 1070–1083
18. Chartrand, G., Zhang, P.: Chromatic Graph Theory. CRC press (2008)
19. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR. (2016) 724–732
20. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE Trans. Pattern Anal. Mach. Intell. **36**(6) (2014) 1187–1200
21. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: ICCV. (2011) 1995–2002
22. Ma, T., Latecki, L.J.: Maximum weight cliques with mutex constraints for video object segmentation. In: CVPR. (2012) 670–677

23. Koh, Y.J., Kim, C.S.: Primary object segmentation in videos based on region augmentation and reduction. In: CVPR. (2017) 3442–3450
24. Wang, W., Shen, J., Porikli, F.: Saliency-aware geodesic video object segmentation. In: CVPR. (2015) 3395–3402
25. Yang, J., Price, B., Shen, X., Lin, Z., Yuan, J.: Fast appearance modeling for automatic primary video object segmentation. IEEE Trans. Image Process. **25**(2) (2016) 503–515
26. Shi, J., Malik, J.: Motion segmentation and tracking using normalized cuts. In: ICCV. (1998) 1154–1160
27. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV. (2010) 282–295
28. Ochs, P., Brox, T.: Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In: ICCV. (2011) 1583–1590
29. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: CVPR. (2012) 614–621
30. Fragkiadaki, K., Zhang, G., Shi, J.: Video segmentation by tracing discontinuities in a trajectory embedding. In: CVPR. (2012) 1846–1853
31. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. In: ICCV. (2017) 4491–4500
32. Cheng, J., Tsai, Y.H., Wang, S., Yang, M.H.: Segflow: Joint learning for video object segmentation and optical flow. In: ICCV. (2017) 686–695
33. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixe, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR. (2017) 221–230
34. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR. (2017) 2663–2672
35. Jang, W.D., Kim, C.S.: Online video object segmentation via convolutional trident network. In: CVPR. (2017) 5849–5858
36. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: CVPR. (2013) 1155–1162
37. Goferman, S., Zelnik-Manor, L., Tal, A.: Context-aware saliency detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(10) (2012) 1915–1926
38. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3) (2015) 569–582
39. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: Contrast based filtering for salient region detection. In: CVPR. (2012) 733–740
40. Wei, Y., Wen, F., Zhu, W., Sun, J.: Geodesic saliency using background priors. ECCV (2012) 29–42
41. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: CVPR. (2014) 2814–2821
42. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR. (2013) 3166–3173
43. Li, G., Yu, Y.: Deep contrast learning for salient object detection. In: CVPR. (2016) 478–487
44. Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., Jodoin, P.M.: Non-local deep features for salient object detection. In: CVPR. (2017) 6609–6617
45. Hu, P., Shuai, B., Liu, J., Wang, G.: Deep level sets for salient object detection. In: CVPR. (2017) 2300–2309
46. Fang, Y., Wang, Z., Lin, W., Fang, Z.: Video saliency incorporating spatiotemporal cues and uncertainty weighting. IEEE Trans. Image Process. **23**(9) (2014) 3910–3921
47. Liu, Z., Zhang, X., Luo, S., Le Meur, O.: Superpixel-based spatiotemporal saliency detection. IEEE Trans. Circuits Syst. Video Technol. **24**(9) (2014) 1522–1540
48. Wang, W., Shen, J., Shao, L.: Consistent video saliency using local gradient flow optimization and global refinement. IEEE Trans. Image Process. **24**(11) (2015) 4185–4196

49. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Adv. Neural Inf. Process. Syst. (2015) 91–99
50. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11) (2012) 2274–2282
51. Pan, J.Y., Yang, H.J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: Proc. ACM SIGKDD. (2004) 653–658
52. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: CVPR. (2005) 524–531
53. Galasso, F., Nagaraja, N.S., Cardenas, T.J., Brox, T., Schiele, B.: A unified video segmentation benchmark: Annotation, metrics and analysis. In: ICCV. (2013) 3527–3534
54. Feremans, C., Labbé, M., Laporte, G.: Generalized network design problems. European Journal of Operational Research **148**(1) (2003) 1–13
55. Zamir, A.R., Dehghan, A., Shah, M.: GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In: ECCV. (2012) 343–356
56. Althaus, E., Kohlbacher, O., Lenhof, H.P., Müller, P.: A combinatorial approach to protein docking with flexible side chains. Journal of Computational Biology **9**(4) (2002) 597–612
57. Dehghan, A., Modiri Assari, S., Shah, M.: GMMCP Tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In: CVPR. (2015) 4091–4099
58. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Quart. **2**(1-2) (1955) 83–97
59. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM Trans. Graphics. Volume 23. (2004) 309–314
60. Chinneck, J.W.: Practical Optimization: A Gentle Introduction. Systems and Computer Engineering (2006)
61. Taylor, B., Karasev, V., Soattoc, S.: Causal video object segmentation from persistence of occlusions. In: CVPR. (2015) 4268–4276
62. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC. (2017)
63. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR. (2015) 5455–5463
64. Jiang, B., Zhang, L., Lu, H., Yang, C., Yang, M.H.: Saliency detection via absorbing Markov chain. In: ICCV. (2013) 1665–1672
65. Zhou, F., Kang, S.B., Cohen, M.F.: Time-mapping using space-time saliency. In: CVPR. (2014) 3358–3365
66. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)