Multimedia Tools and Applications (2019) 78:10353–10372
https://doi.org/10.1007/s11042-018-6614-0

Video co-segmentation based on directed graph

Yufeng Xie[1,2] • Zhi Liu[1,2] • Xiaofei Zhou[1,2,3] • Wei Liu[4] • Xuemei Zou[2]

Received: 15 December 2016 / Revised: 14 May 2018 / Accepted: 27 August 2018 /
Published online: 6 September 2018

Abstract

This paper proposes a novel video co-segmentation method, which aims to extract multi-class objects from a group of videos. A set of tracklets are first generated based on object proposals, and then a novel directed graph is constructed to connect object tracklets. The directed graph is transformed to an undirected graph, and the extraction of common object tracklets is solved by using maximum weighted clique. The obtained common object tracklets are used as seed regions to perform manifold ranking and to generate the object-level saliency maps. Based on common object tracklets and object-level saliency maps, GrabCut is exploited to get the refined co-segmentation results. Experimental results on a public video dataset show that the proposed video co-segmentation method consistently outperforms the state-of-the-art methods.

Keywords Video co-segmentation · Saliency map · Directed graph

✉ Zhi Liu
liuzhisjtu@163.com

Yufeng Xie
xieyufeng0227@163.com

Xiaofei Zhou
zxforchid@outlook.com

Wei Liu
liuwei.1989@sjtu.edu.cn

Xuemei Zou
zxm@staff.shu.edu.cn

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

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

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

[4] Key Laboratory of Ministry of Education for System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China

1 Introduction

Nowadays, video object segmentation is an important research area owning to its wide variety of applications such as content-aware retargeting [5], content-based retrieval [3] and video surveillance, just to name a few. Continuous improvements on video segmentation methods have been achieved in recent years [20]. Unsupervised video segmentation methods were proposed in [7, 18], which exploited appearance information and motion cues simultaneously. In [13, 33], object proposals are introduced as a preprocessing step to catch high-level features and obtain more robust segmentation results. However, unsupervised methods may be invalid due to complex situations in different videos. In contrast, supervised methods [19, 28] provide manually annotated labels for objects in some frames and design the appearance models with motion cues, for the better segmentation quality. Nevertheless, supervised methods are hard to be extended due to the need of manual intervention. Therefore, in order to achieve a better video segmentation performance in an unsupervised manner, video co-segmentation methods [2, 4, 8–11, 16, 23, 25, 27, 34, 35] were introduced to compensate for the lack of supervision by exploiting additional information across multiple videos.

The aim of video object co-segmentation is to pursue a better segmentation for each object via mining the similarity information among different videos. It is also regarded as an extension of image co-segmentation [22, 24] and image co-saliency [12, 15, 32]. The video co-segmentation methods in [2, 23] enforced a cooperative constraint with a global appearance model for common objects. In [9, 10], the constraints of trajectory co-saliency and coherent moving for the foreground local parts were introduced into video co-segmentation respectively. A multi-class video co-segmentation method was proposed in [4], which was based on a nonparametric Bayesian model. This model used a video segmentation prior as well as a global appearance model that grouped dense image patches to obtain segments, which could potentially yield noisy results. In [25], a spatiotemporal SIFT flow was designed to generate estimations of common objects. Then, intra-frame saliency, inter-frame consistency and estimations of common objects were incorporated into an energy optimization framework to obtain the common object regions.

Recently, the object-based methods were introduced in [8, 16, 29, 34], which discovered the common objects by mining the consistency information among the segmentation proposals. Specifically, bounding boxes are used in [29] to initialize a set of easy instances clusters for an input video, and then an iterative growing process is exploited to detect the harder instances throughout the entire video for each cluster. However, this method only focuses on segmenting objects in a single video. The method [16] built a probabilistic graphical model across a set of videos based on object proposals in each frame. Then an energy function incorporating appearance, spatial and temporal consistency of primary objects was solved to obtain common objects. The method in [8] formulated a multi-state selection graph in which each proposal was a state in each frame node. Object score, region overlapping, intra-video and inter-video coherence were considered in the graph to optimize the segmentation of different objects jointly. However, this method suffers from restrictive assumptions that the number of object classes in each video should be the same. It cannot deal with the situation that the common object does not appear in some videos. In [34], object proposals were used to form a number of tracklets, and then each tracklet was treated as a node to construct an undirected graph; inter-video and intra-video constraints were considered to set an edge connecting each pair of nodes, and then edges with the similarity lower than a manually set threshold were removed; last, the multi-class object extraction problem was solved by obtaining the regularized maximum weight cliques iteratively. However, in [34], a manual setting is needed to indicate the similarity of common objects in all videos, and this makes the method hard to be extended to practical applications and may cause failure in the situation that the common objects have a large appearance difference among videos.

As an improvement work on [8], in [35], object clusters are used to construct a weighted graph, which is employed to highlight the common object. In [11], a co-saliency based segmentation scheme is built based on superpixels and employed to find out the co-salient object regions among videos. In [27], video co-segmentation is designed by minimizing an energy function, which incorporates co-saliency term, intra-video appearance term, inter-video appearance term and spatiotemporal smoothness term. However, since these methods [11, 27, 35] utilize the co-saliency maps to segment the common objects, their common limitation is incapable of identifying different classes of objects in the video set.

Therefore, in this paper, we propose a novel video co-segmentation method, which aims at achieving more effective and automatic co-segmentation of multi-class objects. Our main contribution lies in the following two aspects. First, a novel directed graph is designed to effectively connect object tracklets via the tracklet-wise co-saliency maps. In [34], objects with similarity lager than a threshold from different videos will be considered as the common objects. Different from [34], in our method two salient objects that are in different videos and point to each other by the directed edges are treated as belonging to the same class. This is an obvious superiority over [34] since the proposed directed graph can avoid the fine-tuning on the threshold for the object similarity in [34]. Besides, different from the co-saliency based methods [11, 27, 35], our tracklet-wise co-saliency maps are used to connect similar object tracklets and can deal with the segmentation of multi-class objects. Second, we propose to perform video object co-segmentation via a new pipeline, which first extracts common object tracklets by solving a maximum clique problem, then generates object-level saliency maps by manifold ranking [31] and finally exploits GrabCut [21] to obtain the object co-segmentation results.

Different from the method in [29], which focuses on segmenting objects in a single video, our method aims at co-segmenting objects in multiple videos. In [29], after iterative growing, the obtained harder instances are used as weak supervision to segment foreground objects in each video frame. Differently, our method first adopts the obtained tracklets to generate tracklet-wise co-saliency maps by combining the initial saliency maps and tracklet-wise similarity maps, where the tracklet-wise similarity maps are generated based on similarity between different tracklets in different videos. Then, we construct the directed graph by connecting tracklets based on the tracklet-wise co-saliency maps between different videos, and the maximum weight clique (MWC) is extracted from the directed graph for the final segmentation.

The rest of this paper is organized as follows. Section 2 details the proposed video co-segmentation method illustrated in Fig. 1. Experimental results and analysis are presented in Section 3, and conclusions are given in Section 4.

2 Proposed video co-segmentation method

2.1 Preprocessing

Given a set of videos $\{V_m\}_{m=1}^{M}$, in which each video V_m contains a set of frames $\left\{F_{m,t}\right\}_{t=1}^{N_m}$. We transform each frame $F_{m,\ t}$ into the *Lab* color space, which is more correlated with the human perception. For each frame $F_{m,\ t}$, each color channel is uniformly quantized into 16 bins

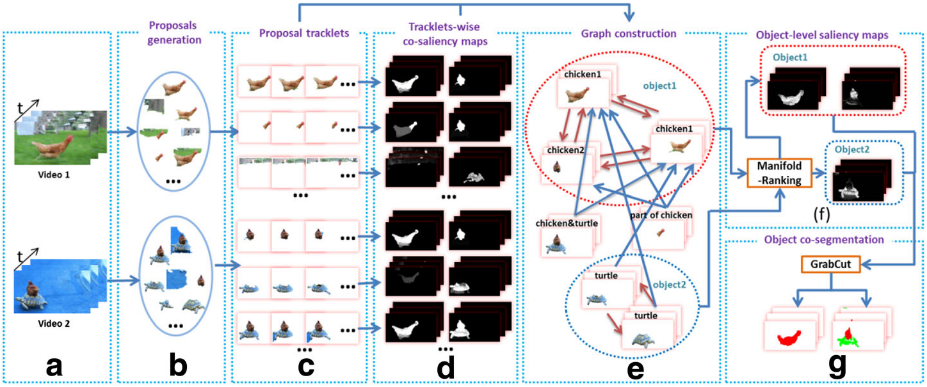


Fig. 1 Overview of the proposed video co-segmentation method. (**a**) Input videos; (**b**) object proposals; (**c**) object tracklets; (**d**) tracklet-wise co-saliency maps; (**e**) graph construction; (**f**) object-level saliency maps; (**g**) object co-segmentation results

for generating the color histogram. Then the category independent object proposal algorithm [6] is used to generate a set of candidate regions as object proposals $\{x_{m,t,i}\}_{i=1}^{q}$ for each frame $F_{m,\,t}$. Let H_a denote the color histogram calculated for region R_a, the color similarity between any pair of regions, R_a and R_b, is defined as follows:

$$Sim(R_a, R_b) = 1 - \chi^2[H_a, H_b] \tag{1}$$

where $\chi^2[\cdot]$ denotes the chi-square distance between histograms. For the videos shown in Fig. 1(a), the object proposals generated for video frames are shown in Fig. 1 (b).

2.2 Tracklet generation

A number of proposals have been generated for each frame, and each proposal may correspond to a real object region, a background region, or a region with a part of object and a part of background. In order to evaluate the likelihood of each proposal belonging to a real object, we firstly obtain the initial saliency map $IS_{m,\,t}$ for each frame $F_{m,\,t}$ by using the intra saliency map generation step in [30], which combines spatial saliency, temporal saliency and object prior for a maximal preservation of salient objects in terms of both appearance and motion. Specifically, the initial saliency map is defined as follows:

$$IS_{m,t}(p) = \left[SS_{m,t}(p) + ST_{m,t}(p)\right] \cdot OP_{m,t}(p) \tag{2}$$

where p denotes each pixel in the video frame. $SS_{m,\,t}(p)$ denotes the spatial saliency, which is computed based on color contrast and spatial sparsity within the frame, and $ST_{m,\,t}(p)$ denotes the temporal saliency, which is generated based on motion distinctiveness from background and temporal coherence in a period of consecutive frames. Besides, $OP_{m,\,t}(p)$ denotes the object prior, which is calculated based on the observation that salient object regions connect with image borders less than background regions, and such a prior can effectively highlight salient object regions and suppress background regions. Then for each proposal $x_{m,\,t,\,i}$, the object score is calculated as follows:

$$IOS\left(x_{m,t,i}\right) = \frac{\sum_{p \in x_{m,t,i}} IS_{m,t}(p)}{\sum_{p \in F_{m,t}} IS_{m,t}(p)} \cdot \frac{\sum_{p \in x_{m,t,i}} IS_{m,t}(p)}{\left|x_{m,t,i}\right|} \tag{3}$$

where p denotes each pixel in the video frame, $|x_{m,t,i}|$ denotes the number of pixels in the proposal $x_{m,t,i}$. According to Eq. (3), the proposal with a higher object score contains a larger object region with a smaller background region. Then we track each proposal $x_{m,t,i}$ backward and forward along the whole video to form the corresponding track $X_{m,t,i}$.

For tracking object proposals, a similarity function combining color and location between any pair of proposals extracted from adjacent frames is defined as follows:

$$S\left(x_{m,t,i},x_{m,u,j}\right) = Sim\left(x_{m,t,i},x_{m,u,j}\right)\cdot\frac{\left|x_{m,t,i}\cap warp_{u,t}\left(x_{m,u,j}\right)\right|}{\left|x_{m,t,i}\cup warp_{u,t}\left(x_{m,u,j}\right)\right|} \tag{4}$$

where $warp_{u,t}(x_{m,u,j})$ denotes the warped region from proposal j of frame u by using optical flow [14] to frame t. For example, based on proposal i in frame u, we can find the most similar proposal j in frame $t+1$ according to Eq. (4); and then based on proposal j, we can also find the most similar proposal l in frame $t+2$. This process will be performed iteratively in consecutive frames, and it is also performed similarly in the temporally backward direction. Through this way, we can generate a track $X_{m,t,i}$ from the whole video for the proposal $x_{m,t,i}$. As shown in Fig. 2, two tracking proposals examples are used to generate corresponding tracks. This process will generate a large number of tracks since each object proposal corresponds to a track. Most of the generated tracks are overlapping and therefore are redundant. A non-maximum suppression process is then performed to remove the redundant tracks for each video. Specifically, we first calculate the object score for each track as follows:

$$ITS\left(X_{m,t,i}\right) = \sum\nolimits_{x\in X_{m,t,i}} IOS(x) \tag{5}$$

Based on Eq. (5), the track with the highest score is selected as a reference track X^R. Then we calculate the overlap ratio for all the other tracks $Y=\{Y^1, \ldots, Y^{NT}\}$ with respect to the reference track X^R as follows:

$$O\left(X^R, Y^l\right) = \frac{\sum_{t=1}^{N_m}\sum\nolimits_{x_{m,t,i}\in X^R, y_{m,t,j}\in Y^l}\left|x_{m,t,i}\cap y_{m,t,j}\right|}{\sum_{t=1}^{N_m}\sum\nolimits_{x_{m,t,i}\in X^R, y_{m,t,j}\in Y^l}\left|x_{m,t,i}\cup y_{m,t,j}\right|} \tag{6}$$

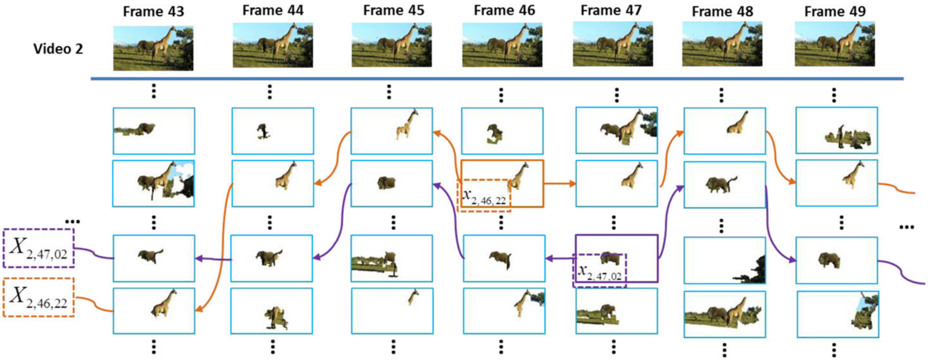


Fig. 2 Illustration of the track generation. This picture only shows two examples of tracking proposals backward and forward to generate corresponding tracks. Each line with arrow points to the most similar proposal in the next frame or previous frame

Based on Eq. (6), the tracks which have the overlap ratio larger than 0.5 with X^R will be removed. In the remaining track set, we select the track with the second highest score as a reference track, and then a selection operation as above will be performed again; such a process will be performed continuously until each of the remaining tracks has been used as the reference track. Then each remaining track is split into some tracklets to ensure the consistency of object proposals. Specially, the similarity between two proposals in adjacent frames is used to determine whether the track should be split between two proposals or not. For each track, the mean of such similarity measures is calculated, if the similarity between any two proposals is 1.5 times the standard deviation away from the mean similarity measure, the track is split between the two proposals in the adjacent frames as shown in Fig. 3. Using the above split operation, we obtain the tracklets set $X = \{X_{m,k}\}_{m=1,\dots,M;k=1,\dots,K_m}$ for the whole video set $\{V_m\}_{m=1}^{M}$, where K_m is the number of final tracklets in V_m. The examples of generated tracklets is shown in Fig. 1(c).

Algorithm 1 Pseudo Code of Similarity Maps Generation

```
Input: The tracklet $X_{m,k}$ for which the similarity maps are calculated; all the other tracklets in $\mathrm{X} = \{X_{n,k}\}_{n=1,\dots,M;k=1,\dots,K_n}$.
Output: A set of similarity maps $SM = \{SM_{m,k}^{n,t}\}_{n=1,\dots,M;t=1,\dots,N_n}$ for $X_{m,k}$.
Begin
    Define and initialize the similarity map in each frame: $SM_{m,k}^{n,t}(p) = 0, \forall p \in F_{n,t}$.
    Define and initialize the mask in each frame: $MASK_{m,k}^{n,t}(p) = 1, \forall p \in F_{n,t}$.
    For $n \leftarrow 1\ to\ M$
        For $l \leftarrow 1\ to\ K_n$
            Find the tracklet $Z$ which has the highest similarity with $X_{m,k}$ :
            $Z = \arg\max_{X_{n,l}}\left[Sim\left(X_{m,k}, X_{n,l}\right)\right]$.
            For $t \leftarrow 1\ to\ N_n$
                Obtain the region $CR_{m,k}^{n,t}$ in frame $F_{n,t}$ for comparing with $X_{m,k}$ as follows:
                If $z_{n,t,i} = \varnothing$
                    Continue;
                Else
                    $CR_{m,k}^{n,t} = MASK_{m,k}^{n,t} \cap z_{n,t,i}, z_{n,t,i} \in Z$.
                End
                Set the pixels' values in $CR_{m,k}^{n,t}$ to the similarity between $CR_{m,k}^{n,t}$ and $X_{m,k}$ as follows:
                $SM_{m,k}^{n,t}(p) = Sim\left(X_{m,k}, CR_{m,k}^{n,t}\right), \forall p \in CR_{m,k}^{n,t}$.
                Reset the mask as follows:
                $MASK_{m,k}^{n,t}(p) = 0, p \in CR_{m,k}^{n,t}$.
            End
            Delete the tracklet $Z$ from tracklet set $\mathrm{X}$.
        End
        For $t \leftarrow 1\ to\ N_n$
            Set the pixels' values in $MASK_{m,k}^{n,t}$ to the similarity between $MASK_{m,k}^{n,t}$ and $X_{m,k}$ as follows:
            $SM_{m,k}^{n,t}(p) = Sim\left(X_{m,k}, MASK_{m,k}^{n,t}\right), \forall p \in (MASK_{m,k}^{n,t} = 1)$.
        End
    End
End
```

2.3 Graph construction

We obtained a lot of tracklets in Section 2.2, and each tracklet represents some kind of object or background. Video co-segmentation aims at finding the relationship among videos and

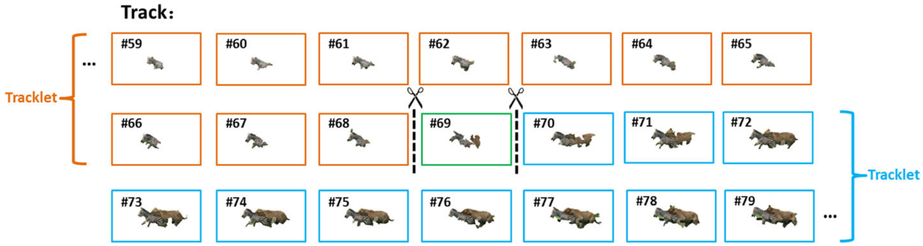


Fig. 3 Illustration of splitting one remaining track into some tracklets. Splitting situations occur in two positions (marked with a scissor) in this example, i.e., one is between frame 68 and 69, and another one is between frame 69 and 70

extracting multi-class objects from all videos. For this purpose, we propose a directed graph, which tries to connect each tracklet belonging to the same class and extract the salient ones as objects. We use $G = (V, E, W)$ to denote the directed graph, where V is the set of vertices, E is the set of directed edges. Each vertex $v_{m,\,k} \in V$ represents a tracklet $X_{m,\,k}$, and $e = (v_{m,\,k}, v_{n,\,l})$ denotes the edge directing from $v_{m,\,k}$ to $v_{n,\,l}$.

Based on the definition of directed graph, we try to connect each tracklet with the best-matched tracklet obtained from all frames in all videos. Here we make an assumption that each tracklet $X_{m,\,k}$ tends to connect to the best-matched tracklet which contains complete and similar objects. This means that a tracklet containing a part of object may connect to a tracklet containing complete and similar objects, and two tracklets containing complete and similar objects tend to connect with each other. For our purpose, the matching score of one tracklet $X_{m,\,k}$ with respect to another tracklet $X_{n,\,l}$ is defined as follows:

$$FTS_{m,k}\left(X_{n,l}\right) = Sim\left(X_{m,k}, X_{n,l}\right)\cdot\left\{\sum\nolimits_{x_{n,t,i}\in X_{n,l}}\left(\frac{\sum\nolimits_{p\in x_{n,t,i}} TCS_{m,k}^{n,t}(p)}{\sum\nolimits_{p\in F_{n,t}} TCS_{m,k}^{n,t}(p)}\cdot\frac{\sum\nolimits_{p\in x_{n,t,i}} TCS_{m,k}^{n,t}(p)}{\left|x_{n,t,i}\right|}\right)\right\} \tag{7}$$

In Eq. (7), the tracklet-level similarity $Sim(X_{m,\,k}, X_{n,\,l})$ takes the form of Eq. (1), in which the two color histograms are calculated based on the pixels of all proposals in $X_{m,\,k}$ and $X_{n,\,l}$ respectively. The tracklet-wise co-saliency map of $X_{m,\,k}$ with respect to the frame $F_{n,\,t}$ is defined as follows:

$$TCS_{m,k}^{n,t}(p) = SM_{m,k}^{n,t}(p)\cdot IS_{n,t}(p) \tag{8}$$

where $SM_{m,k}^{n,t}$ is the tracklet-wise similarity map of $X_{m,\,k}$ with respect to $F_{n,\,t}$. Eq. (8) indicates that the tracklet-wise co-saliency map is generated by integrating the initial saliency map with the tracklet-wise similarity map by multiplication operation.

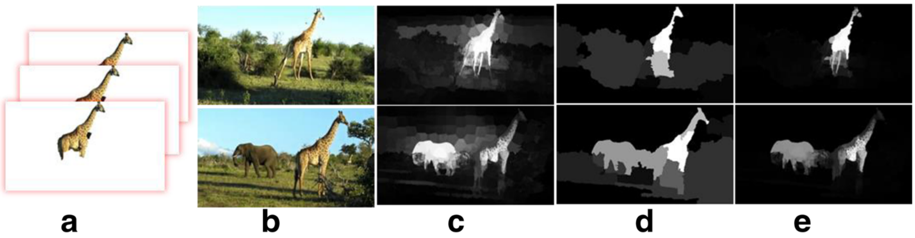


Fig. 4 Illustration of generating the tracklet-wise co-saliency maps. (**a**) One tracklet; (**b**) original frames; (**c**) initial saliency maps; (**d**) similarity maps; (**e**) tracklet-wise co-saliency maps

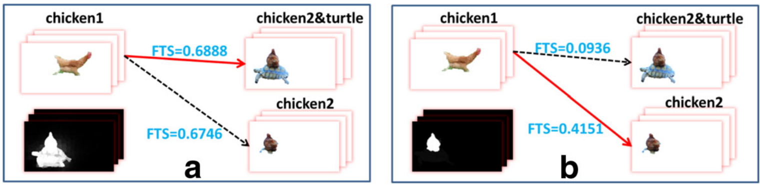


Fig. 5 Illustration of difference between using initial saliency map and tracklet-wise co-saliency map in the directed graph construction. (**a**) Tracklet matching process using initial saliency maps; (**b**) tracklet matching process using tracklet-wise co-saliency maps. The red arrow line in (**a**) or (**b**) denotes the best matched tracklet

The tracklet-wise co-saliency map $TCS_{m,k}^{n,t}$ can highlight the regions, which are salient in the frame $F_{n,\ t}$ and similar with the proposals in tracklet $X_{m,\ k}$. As shown in Fig. 4, the initial saliency maps highlight both objects, i.e., giraffe and elephant, based on appearance and motion, while the similarity maps of the tracklet corresponding to giraffe can highlight the regions of giraffe. By combining initial saliency maps with similarity maps, the obtained tracklet-wise co-saliency maps can better highlight the regions of giraffe, and suppress other regions irrelevant to the tracklet. It can be observed in Fig. 4 that the similarity map is important to generate the tracklet-wise co-saliency map. The generation process of similarity map for each tracklet is described in Algorithm 1.

To intuitively demonstrate the effect of tracklet-wise co-saliency map, a comparison is shown in Fig. 5. Figure 5(a) shows the connection of two tracklets using Eq. (7) based on initial saliency map. As shown in Fig. 5(a), since the initial saliency maps highlight the regions of both the chicken and the turtle, the matching score of '*chicken1*' with respect to '*chicken2&turtle*' is larger than that with respect to '*chicken2*', which results in the wrong matching. In contrast, as show in Fig. 5(b), since the tracklet-wise co-saliency maps of '*chicken1*' only highlight the region of chicken, the matching score of '*chicken1*' with respect to '*chicken2*' is larger, which leads to the correct matching.

By using Eqs. (7–8), we can obtain the matching scores of each tracklet $X_{m,\ k}$ with respect to all the other tracklets. The matching scores are exploited to find the best-matched tracklets for $X_{m,\ k}$. For each tracklet $X_{m,\ k}$, we search in each video frame to find another tracklet $X_{n,\ l}$ with the matching score $FTS_{m,\ k}(X_{n,\ l})$ as the highest one. Specifically, in video V_m, in which $X_{m,\ k}$ exists, we only search for $X_{m,\ k}$ its best matched tracklet $X_{n,\ l}$ with $Sim(X_{m,\ k}, X_{n,\ l}) > 0.9$. Then a connection is added from $X_{m,\ k}$ to $X_{n,\ l}$ with a directed edge. The latter constraint of similarity with a high threshold, 0.9, is exploited to discard tracklets in those video frames where similar objects are unlikely to appear.

Figure 1(e) gives the illustration of graph construction. It can be seen that, based on the matching score function, a tracklet of '*chicken1*' directs to the same object class '*chicken2*' in video2 instead of '*chicken2&turtle*' or '*turtle*'. Similarly, a tracklet of '*chicken2*' directs to the same object class '*chicken1*' instead of '*part of chicken*' or '*chicken2&turtle*'. It indicates that the tracklets of complete and similar objects direct to each other, and such two objects pointing to each other by the directed edges can be treated as belonging to the same class.

2.4 Common object extraction

After connecting each tracklet with the best-matched tracklet, we transform the directed graph $G = (V, E, W)$ into an undirected graph $G' = (V, E', W)$. Based on the assumption that tracklets connecting to each other contain complete and similar objects. Any pair of tracklets is connected with an undirected edge if they are connected with each other by two directed edges. The pairwise connected tracklets in $G^{'}$ belong to a complete subgraph. Thus, we can regard the extraction of common object tracklets as the problem of Maximum Weight Clique (MWC). Those tracklets corresponding to the primary objects will have high object scores. We use Bron-Kerbosch algorithm [1] to find all maximal cliques,$C_h(h = 1, ..., H)$, from $G^{'}$. The weight of each clique is defined as follows:

$$W(C_h) = \sum_{X_{m,k} \in C_h} ITS\left(X_{m,k}\right) \tag{9}$$

Then the MWC from all cliques is obtained as follows:

$$MWC = \arg\max_{C_h} \left[W(C_h)\right] \tag{10}$$

The proposals in MWC are regarded as object queries to compute object-level saliency maps by using the manifold ranking algorithm [31]. Then for each frame $F_{m,\ t}$, we threshold the object-level saliency map adaptively by using the Otsu's method [17]. The obtained background pixels are used to estimate the background Gaussian mixture model $GMM^b_{m,t}$. The object Gaussian mixture model GMM^f_m is estimated for each video V_m based on the proposals which belong to V_m and MWC. Finally, the GrabCut [21] based on GMM^f_m and $GMM^b_{m,t}$ is used to obtain the co-segmentation result of primary object class for each video frame $F_{m,\ t}$. For the object extraction of the second class, we set the pixels of the extracted primary object regions in the initial saliency maps to zero. Then we update the graph by recalculating the object scores of proposals and tracklets by using Eq. (3) and (5) as well as the weights of cliques by using Eq. (9). The objects of the second class can be extracted as a new MWC based on Eq. (10). The above process can be performed multiple times in order to extract multiple classes of objects from the video set. Examples of the final co-segmentation result are shown in Fig. 1(g), in which multi-class objects are represented by using different colors.

3 Experimental results

We performed experiments on two public video datasets MOViCS [4] and Safari [34]. The experimental settings, objective and subjective evaluations including comparisons with two state-of-the-art video co-segmentation methods and video segmentation method on MOViCS and Safari are presented in Section 3.1, Section 3.2 and Section 3.3, respectively. The computation issue of video co-segmentation methods is discussed in Section 3.4.

3.1 Experiments on MOViCS dataset

The proposed video co-segmentation method is evaluated on the public video dataset MOViCS [4], which contains a total of 11 videos grouped into 4 video sets. Each video set contains one or two object classes. Originally, the manually annotated ground truths are provided for only 5 frames in each video. For a comprehensive comparison, we extended the ground truths by uniformly sampling 20 frames in each video and manually annotated the pixel-level ground truths of all objects in these video frames. The proposed method is compared with the two state-of-the-art video co-segmentation methods including MVC [4]

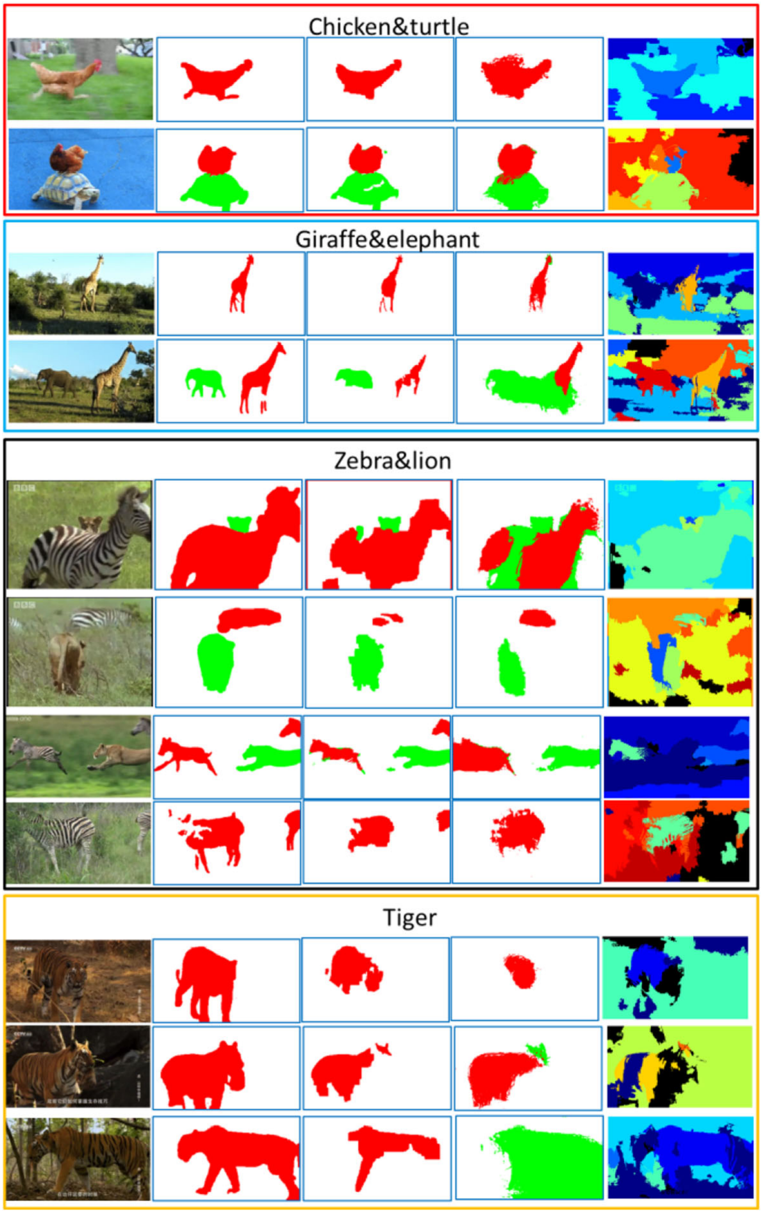


Fig. 6 Video co-segmentation results on the MOViCS dataset. From left to right: original video frames, ground truths, co-segmentation results generated using the proposed method, RMWC [34] and MVC [4], respectively. From the 2nd to the 4th column, red regions correspond to the objects of the primary class and green regions correspond to the objects of the second class

and RMWC [34], which can extract multi-class objects. We tested both MVC [4] and RMWC [34] with their publicly available codes. For RMWC [34], we used the best thresholds as suggested in [34] for the four video sets, and for MVC [4], we used the default parameter setting in the code.

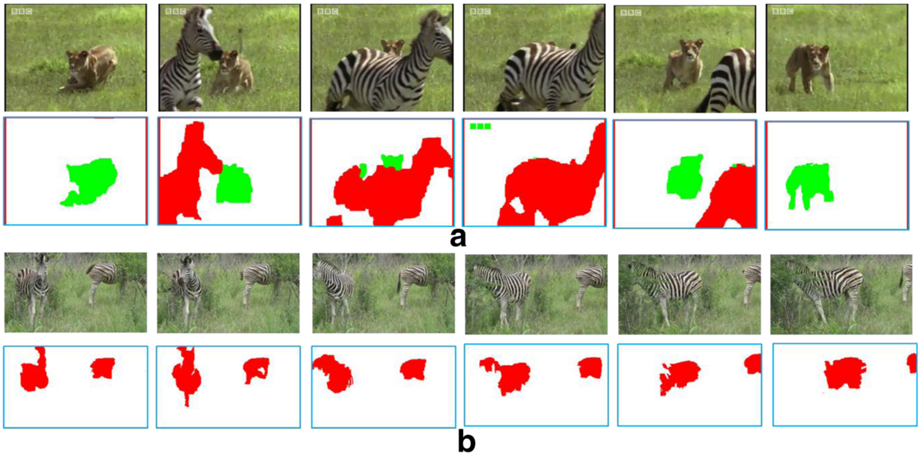


Fig. 7 Experimental results on the video set '*Zebra&lion*' with two situations: (a) the appearance, occlusion and disappearance of the lion and the zebra in some frames; (b) the absence of the lion in the whole video

The multi-class object co-segmentation results generated using our method and the other two methods are shown in Fig. 6 for an intuitive comparison. It can be observed that the results of MVC cannot provide a distinctive segmentation of real objects. Meanwhile, MVC tends to generate results which break one object into a number of fragments. RMWC fails to segment the common object into one class in the video set '*Tiger*'. Note that RMWC uses the manually set thresholds for similarity measures between common objects. This may result in errors when the common objects show a larger difference on appearance. Compared with the other two methods, our method can correctly segment each class of object and can obtain more complete segmentation results by using the directed graph, which avoids the manually parameter setting for similarity measures between common objects.

Fig. 7 shows the effectiveness of our method in dealing with the absence of target objects in some frames or the whole video. Fig. 7(a) shows the appearance, occlusion and disappearance of lion and zebra in some frames, and Fig. 7(b) shows the absence of lion in the whole video. The segmentation results demonstrate that our method can deal with the situations of object appearance and disappearance in some frames or in the whole video.

To objectively evaluate the video co-segmentation performance, following the setup in [4], we use the intersection-over-union metric to quantify the results as follows:

$$M(S,G) = \frac{S \cap G}{S \cup G} \tag{11}$$

Table 1 Quantitative evaluation on the MOViCS dataset

Video Set	MVC	RMWC	Ours
Chicken&turtle	0.622	0.785	**0.837**
Zebra&lion	0.454	0.575	**0.677**
Giraffe&elephant	**0.553**	0.454	0.404
Tiger	0.243	0.361	**0.462**
Average	0.468	0.544	**0.595**

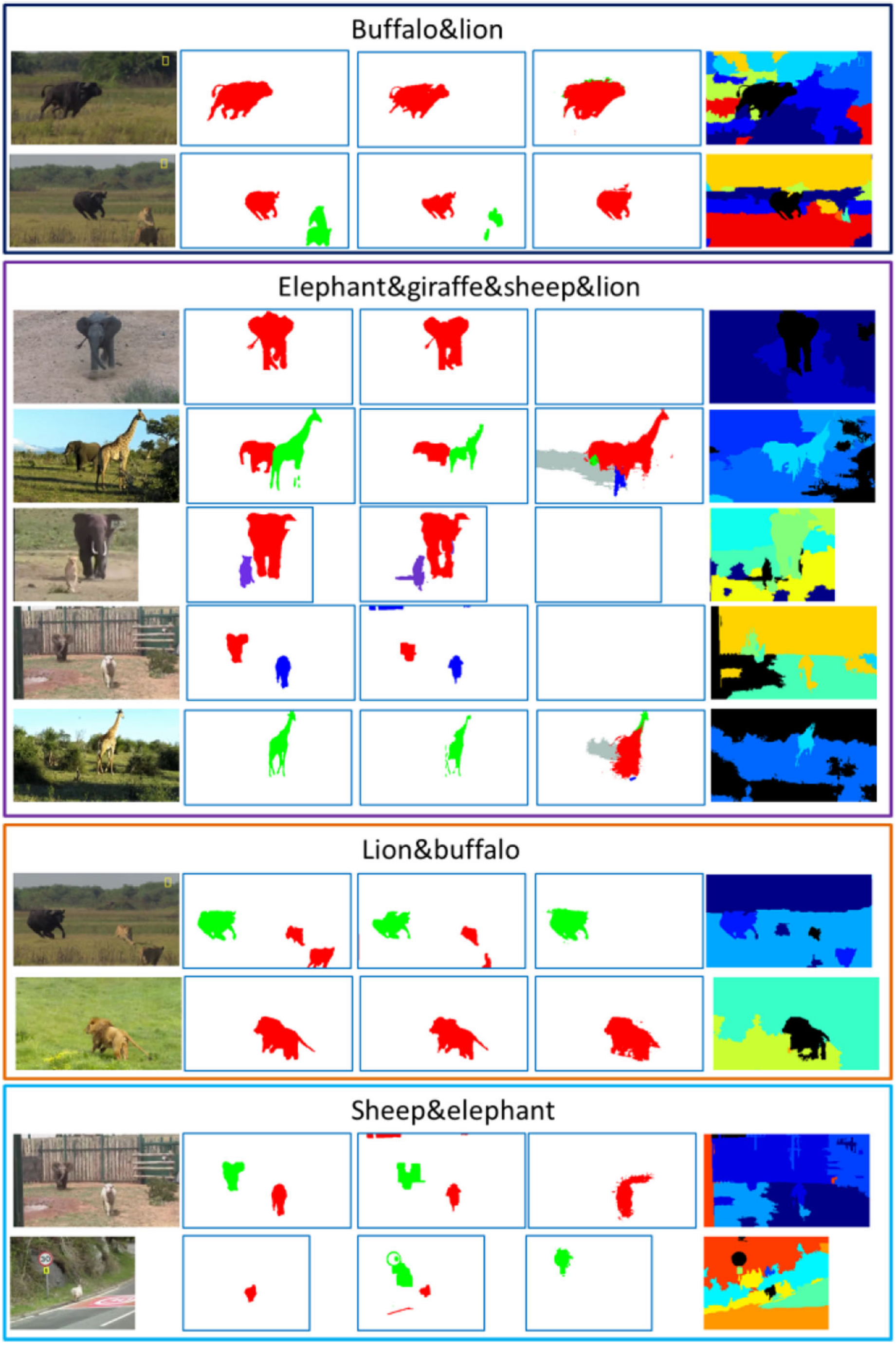


Fig. 8 Video co-segmentation results on the Safari dataset. From left to right: original video frames, ground truths, co-segmentation results generated using the proposed method, RMWC [34] and MVC [4], respectively. From the 2nd to the 4th column, different colors indicate different object classes

Table 2 Quantitative evaluation on the Safari dataset

Video Set	MVC	RMWC	Ours
Buffalo&lion	0.483	0.399	**0.574**
Elephant&giraffe&sheep&lion	0.195	0.151	**0.481**
Lion&buffalo	0.597	0.711	**0.731**
Sheep&elephant	0.120	0.090	**0.326**
Average	0.349	0.338	**0.528**

where S is a set of segments and G is the ground truth. The co-segmentation score of one object class in the video set is defined as follows:

$$Score_j = \max_{S_i}\left[M\left(S_i, G_j\right)\right] \tag{12}$$

where S_i denotes all segments in the object class i, and G_j is the ground truth for the object class j. The final co-segmentation score for the video set is defined as the average score on all object classes as follows:

$$Score = \frac{1}{C}\sum_j Score_j \tag{13}$$

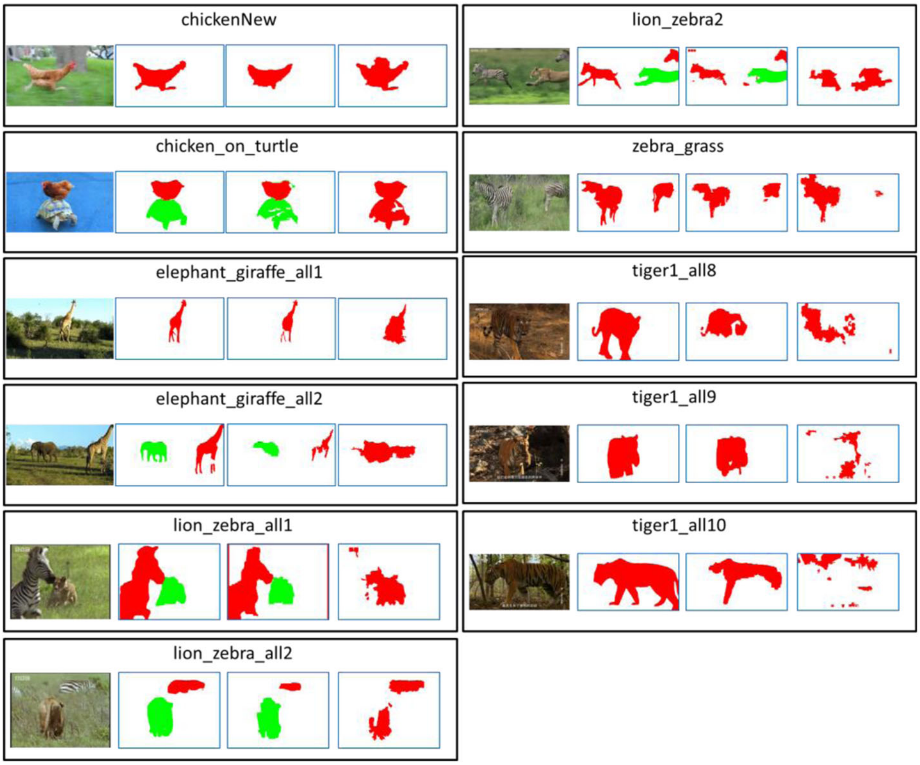


Fig. 9 Comparison between our method and SAGS [26] on the MOViCS dataset. For each video, from left to right: original video frames, ground truths, segmentation results generated using the proposed method and SAGS, respectively. From the 2nd to the 3th column, different colors indicate different object classes

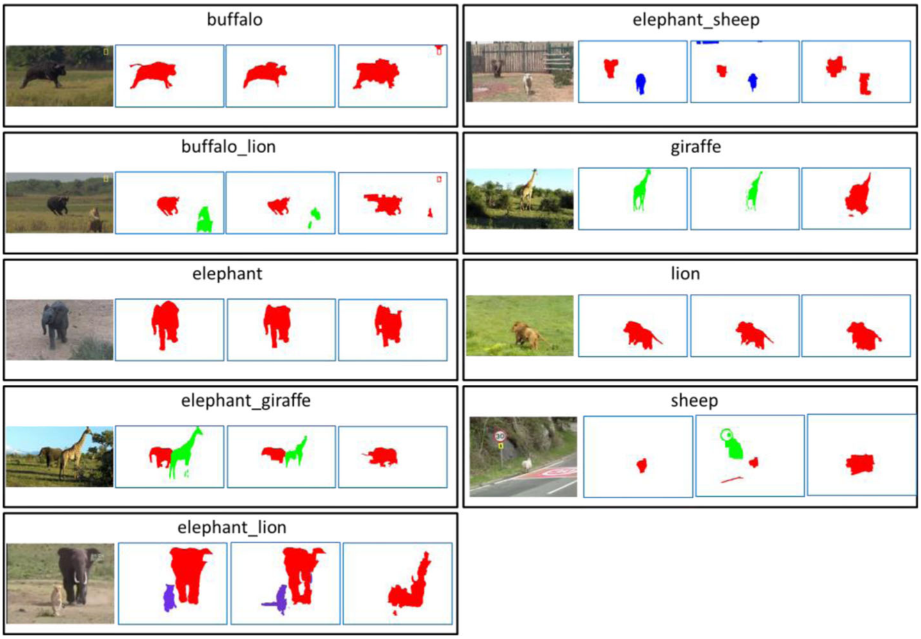


Fig. 10 Comparison between our method and SAGS [26] on the Safari dataset. For each video, from left to right: original video frames, ground truths, segmentation results generated using the proposed method and SAGS, respectively. From the 2nd to the 3th column, different colors indicate different object classes

where C is the number of object classes in the ground truth. Table 1 shows the quantitative comparison of our method with the other two methods. It can be seen that our method outperforms the other two methods in 3 out of 4 video sets. The only video set on which our method has a lower score is '*Giraffe&elephant*' because the segments of giraffe are not complete enough. On average, our method achieves the best performance

Table 3 Quantitative comparison between our method and SAGS [26] on the MOViCS dataset

Video Name (Object)	SAGS	Ours
chickenNew (chicken)	0.725	**0.841**
chicken_on_turtle (chicken)	0.355	**0.863**
chicken_on_turtle (turtle)	0.570	**0.804**
elephant_giraffe_all1 (giraffe)	0.407	**0.493**
elephant_giraffe_all2 (elephant)	0.300	**0.433**
elephant_giraffe_all2 (giraffe)	0.295	**0.426**
lion_zebra2 (lion)	0.415	**0.813**
lion_zebra2 (zebra)	0.312	**0.631**
lion_zebra_all1 (lion)	0.556	**0.773**
lion_zebra_all1 (zebra)	0.313	**0.792**
lion_zebra_all2 (lion)	0.357	**0.790**
lion_zebra_all2 (zebra)	**0.385**	0.246
zebra_grass (zebra)	0.473	**0.575**
tiger1_all8 (tiger)	0.361	**0.403**
tiger1_all9 (tiger)	0.163	**0.616**
tiger1_all10 (tiger)	0.044	**0.503**

Table 4 Quantitative Comparison between our method and SAGS [26] on the Safari dataset

Video Name (Object)	SAGS	Ours
buffalo (buffalo)	0.795	**0.847**
buffalo_lion (buffalo)	0.739	**0.804**
buffalo_lion (lion)	0.021	**0.316**
elephant (elephant)	0.813	**0.897**
elephant_giraffe (elephant)	0.601	**0.665**
elephant_giraffe (giraffe)	0.007	**0.418**
elephant_lion (elephant)	0.339	**0.860**
elephant_lion (lion)	0.159	**0.442**
elephant_sheep (elephant)	0.394	**0.571**
elephant_sheep (sheep)	0.175	**0.195**
giraffe (giraffe)	0.393	**0.595**
lion (lion)	0.813	**0.883**
sheep (sheep)	0.303	**0.480**

on the whole dataset, and this demonstrates the advantage of our directed graph based video co-segmentation.

3.2 Experiments on safari dataset

We also evaluated the proposed method on another dataset, i.e. Safari [34], which contains 9 videos with 5 classes of animals. The proposed method is also compared with the other two state-of-the-art methods, i.e. MVC [4] and RMWC [34]. For both MVC and RMWC, we used the default parameter settings in the codes provided by their authors.

We show the visual comparisons between our method and the other two methods in Fig. 8. It can be seen that for different classes of objects with dissimilar appearances, such as the elephants in the video set '*Elephant&sheep&giraffe&lion*', our method can achieve the better segmentation result than the other two methods. Our method does not perform well on the video set '*Sheep&elephant*', in which some background regions show very similar colors with the sheep and elephant, while the other two methods also do not perform well on this video set. Table 2 shows the quantitative comparison of our method with the other two methods. We can see that our method consistently outperforms the other two methods on all the four video sets, due to the effectiveness and superiority of our directed graph based video co-segmentation method.

3.3 Comparison with the common video segmentation method

Following the settings in the reference [34], we compared our method with the state-of-the-art video segmentation method, SAGS [26], which aims at segmenting objects in a single video.

Table 5 Comparison of average processing time per frame taken by different video co-segmentation methods

Method	MVC	RMWC	Ours
Time(second)	15.2	116	61.93
Code	Matlab	Matlab	Matlab

We performed segmentations on both datasets, i.e. MOViCS and Safari, and some segmentation results are shown in Figs. 9 and 10, respectively, for a visual comparison. It can be seen that our method can identify multi-class objects by incorporating the information originated from other videos, such as the chicken and the turtle in the video '*chicken_on_turtle*'. Differently, due that both the chicken and the turtle are moving objects in this video, SAGS treats them as an entire object and segments them together. Besides, in some low-contrast videos, such as '*tiger1_all8*', '*tiger1_all9*' and '*tiger1_all10*', SAGS fails to segment the tigers in these videos, but our method is able to segment the tigers by incorporating the information of these three videos simultaneously. Besides, we adopted Eq. (13) to evaluate the segmentation quality of the segmentation results generated using our method and SAGS, respectively, for each object class in each video, and the quantitative comparisons on the MOViCS dataset and the Safari dataset are shown in Tables 3 and 4, respectively. It can be seen that our method outperforms SAGS on most of the videos, and this clearly demonstrates the effectiveness of our method.

3.4 Computation cost

We analyze the computation cost of all video co-segmentation methods in this subsection. All experiments are performed on a PC with Intel Core i7–3770 3.4GHz CPU and 16GB RAM. Table 5 reports the average processing time per frame of each method on videos with a resolution of 640 × 360 in MOViCS. The average processing time per frame taken by the MATLAB implementation of our method is 61.93 s. It can be seen from Table 5 that our method has a higher computation cost than MVC. The reason behind this is that our method employs the object proposal generation algorithm which takes lots of time, while the MVC uses the superpixels for co-segmentation. However, our method has a lower computation cost than RMWC.

Therefore, in order to make our method more practical for applications with runtime requirements, the computational efficiency of the object proposal generation algorithm, which is the bottleneck of runtime, should be elevated with the highest priority. With the GPU-accelerated implementation of the object proposal generation algorithm and an optimized C/C++ implementation of other components in our method, we believe that the computation efficiency of our method can be substantially accelerated.

4 Conclusion

This paper proposes a novel video co-segmentation method, which enables an effective and automatic co-segmentation of multi-class objects in a set of videos via the proposed directed graph and a new pipeline for segmentation. Specifically, a novel directed graph is first constructed to connect similar object tracklets, which are generated by object proposals. Then, tracklet-wise co-saliency maps are generated to make the matching score more correct and the video co-segmentation is treated as the problem of MWC. Specifically, based on the extracted MWC, object-level saliency maps are generated by manifold ranking and the GrabCut is further exploited to obtain the final co-segmentation results. Experimental results show that the proposed method consistently outperforms the state-of-the-art video co-segmentation methods on both datasets.

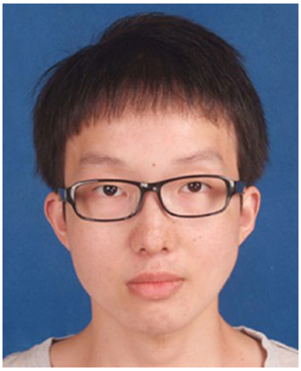
Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No. 61471230, and by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. Commun ACM 16(9):575–577
2. Chen DJ, Chen HT, Chang LW (2012) Video object cosegmentation. In Proc. of ACM International Conference on Multimedia 805–808
3. Cheng MM, Mitra NJ, Huang X, Hu SM (2014) SalientShape: group saliency in image collections. Vis Comput 30(4):443–453
4. Chiu WC, Fritz M (2013) Multi-class video co-segmentation with a generative multi-video model. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 321–328
5. Du H, Liu Z, Jiang J, Shen L (2013) Stretchability-aware block scaling for image retargeting. J Vis Commun Image Represent 24(4):499–508
6. Endres I, Hoiem D (2010) Category independent object proposals. In: Proc. of European Conference on Computer Vision 575–588
7. Faktor A, Irani M (2014) Video segmentation by non-local consensus voting. In: Proc. of British Machine Vision Conference, article 8
8. Fu H, Xu D, Zhang B, Lin S, Ward RK (2015) Object-based multiple foreground video co-segmentation via multi-state selection graph. IEEE Trans Image Process 24(11):3415–3424
9. Guo J, Li Z, Cheong LF, Zhou SZ (2013) Video co-segmentation for meaningful action extraction. In: Proc. of IEEE International Conference on Computer Vision 2232–2239
10. Guo J, Cheong LF, Tan RT, Zhou SZ (2014) Consistent foreground co-segmentation. In Proc. of Asian Conference on Computer Vision 241–257
11. Huang G, Pun CM, Lin C (2017) Unsupervised video co-segmentation based on superpixel co-saliency and region merging. Multimed Tools Appl 76(10):12941–12964
12. Jacobs D, Goldman D, Shechtman E (2010) Cosaliency: Where people look when comparing images. In: Proc. of ACM symposium on User interface software and technology 219–228
13. Lee YJ, Kim J, Grauman K (2011) Key-segments for video object segmentation. In: Proc. of IEEE International Conference on Computer Vision 1995–2002
14. Liu C (2009) Beyond pixels: Exploring new representations and applications for motion analysis. Ph.D. Dissertation, Massachusetts Inst. Technol., Cambridge
15. Liu Z, Zou W, Li L, Shen L, Le Meur O (2014) Co-saliency detection based on hierarchical segmentation. IEEE Sign Proc Lett 21(1):88–92
16. Lou Z, Gevers T (2014) Extracting primary objects by video co-segmentation. IEEE Trans Multimed 16(8): 2110–2117
17. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cyber 9(1):62–66
18. Papazoglou A, Ferrari V (2013) Fast object segmentation in unconstrained video. In: Proc. of IEEE International Conference on Computer Vision 1777–1784
19. Perazzi F, Wang O, Gross M, Sorkine-Hornung A (2015) Fully connected object proposals for video segmentation. In: Proc. of IEEE International Conference on Computer Vision 3227–3234
20. Perazzi F, Pont-Tuset J, McWilliams B, Van Gool L, Gross M, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 724–732
21. Rother C, Kolmogorov V, Blake A (2004) Grabcut: interactive foreground extraction using iterated graph cuts. ACM Trans Graph 23(3):309–314
22. Rother C, Minka T, Kolmogorov V, Blake A (2006) Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 993–1000
23. Rubio JC, Serrat J, López A (2012) Video co-segmentation. In Proc. of Asian Conference on Computer Vision 13–24

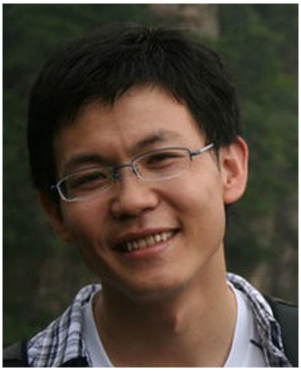
24. Vicente S, Rother C, Kolmogorov V (2011) Object cosegmentation. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 2217–2224
25. Wang W, Shen J, Li X, Porikli F (2015) Robust video object cosegmentation. IEEE Trans Image Process 24(10):3137–3148
26. Wang W, Shen J, Porikli F (2015) Saliency-aware geodesic video object segmentation. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 3395–3402
27. Wang W, Shen J, Sun H, Shao L (2017) ViCoS2: Video co-saliency guided co-segmentation. IEEE Trans. Circuits Syst. Video Technol. doi: https://doi.org/10.1109/TCSVT.2017.2701279
28. Wen L, Du D, Lei Z, Li SZ, Yang MH (2015) JOTS: Joint online tracking and segmentation. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 2226–2234
29. Xiao F, Lee YJ (2016) Track and segment: An iterative unsupervised approach for video object proposals. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 933–942
30. Y. Xie, L. Ye, Z. Liu, and X. Zhou (2016) Video co-saliency detection. In: Proc. of International Conference on Digital Image Processing, article 100335G
31. Xu B, Bu J, Chen C, Cai D, He X, Liu W, Luo J (2011) Efficient manifold ranking for image retrieval. In: Proc. of International ACM SIGIR Conference on Research and Development in Information Retrieval 525–534
32. Ye L, Liu Z, Li J, Zhao W, Shen L (2015) Co-saliency detection via co-salient object discovery and recovery. IEEE Sign Proc Lett 22(11):2073–2077
33. Zhang D, Javed O, Shah M (2013) Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In: Proc. of IEEE Conference on Computer Vision Pattern Recognition 628–635
34. Zhang D, Javed O, Shah M (2014) Video object co-segmentation by regulated maximum weight cliques. In: Proc. of European Conference on Computer Vision 551–566
35. Zhang J, Li K, Tao W (2016) Multivideo object cosegmentation for irrelevant frames involved videos. IEEE Sign Proc Lett 23(6):785–789



Yufeng Xie received the B.E. degree from China Jiliang University, Hangzhou, China, in 2014, and the M.E. degree from Shanghai University, Shanghai, China, in 2017. His research interests include video co-saliency models and video object co-segmentation methods.

Zhi Liu received the B.E. and M.E. degrees from Tianjin University, Tianjin, China, and the Ph.D. degree from Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai, China, in 1999, 2002, and 2005, respectively. He is currently a Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai, China. From Aug. 2012 to Aug. 2014, he was a Visiting Researcher with the SIROCCO Team, IRISA/INRIA-Rennes, France, with the support by EU FP7 Marie Curie Actions. He has published more than 150 refereed technical papers in international journals and conferences. His research interests include image/video processing, machine learning, computer vision and multimedia communication. He was a TPC member/session chair in ICIP 2017, PCM 2016, VCIP 2016, ICME 2014, WIAMIS 2013, etc. He co-organized special sessions on visual attention, saliency models, and applications at WIAMIS 2013 and ICME 2014. He is an area editor of *Signal Processing: Image Communication* and served as a guest editor for the special issue on *Recent Advances in Saliency Models, Applications and Evaluations* in *Signal Processing: Image Communication*. He is a senior member of IEEE.

Xiaofei Zhou received the B.E. degree from Anhui Polytechnic University, Wuhu, China, in 2012, the M.E. degree from Shanghai University, Shanghai, China, in 2015, and the Ph.D. degree from Shanghai University, Shanghai, China, in 2018. He is currently an Assistant Professor with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou, China. His research interests include image processing, pattern recognition and computer vision.



Wei Liu received the B.E. degree from the Department of Automation, Xi'an Jiao Tong University, in 2012. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include image processing and video processing.

Xuemei Zou received the B.E. degree from Shanghai University of Science and Technology, Shanghai, China, in 1982, and now is an Associate Professor at the School of Communication and Information Engineering, Shanghai University, Shanghai, China. Prior to join Shanghai University in 1994, she worked at Optical Fiber Research Institute, Shanghai University of Science and Technology, Shanghai, China, from 1982 to 1990, and then at the School of Communication and Information Engineering, Shanghai University of Technology, Shanghai, China, from 1990 to 1994. Her research interests including image processing and video compression.