# Object Cosegmentation in Noisy Videos with Multilevel Hypergraph

Le Wang, *Member, IEEE,* Xin Lv, Qilin Zhang, *Member, IEEE,* Zhenxing Niu, *Member, IEEE,* Nanning Zheng, *Fellow, IEEE,* and Gang Hua, *Fellow, IEEE*

***Abstract*—With the target of simultaneously segmenting semantically related videos to identify the common objects, video object cosegmentation has attracted the attention of researchers in recent years. Existing methods are primarily based on pair-wise relations between adjacent pixels and regions, which are susceptible to performance degradation from object entries/exists or occlusions. Specifically, we refer these video frames without the common objects present as the "empty" frames. In this paper, we propose a multilevel hypergraph-based full Video object CoSegmentation (VCS) method, which incorporates high-level semantics and low-level appearance/motion/saliency to construct the hyperedge among multiple spatially and temporally adjacent regions. Specifically, the high-level semantic model fuses multiple object proposals from each frame instead of relying on a single object proposal per frame. A hypergraph cut is subsequently utilized to calculate the object cosegmentation. Experiments on four video object segmentation/cosegmentation datasets against state-of-the-art methods with both objective and subjective results manifest the effectiveness of the proposed VCS method, including the SegTrack and VCoSeg datasets without "empty" frames, the XJTU-Stevens dataset with 3.7% "empty" frames, and the Noisy-ViCoSeg dataset proposed together with our method with 30.3% "empty" frames.**

***Index Terms*—Object cosegmentation, Hypergraph cut, Object model, Fully convolutional network.**

## I. Introduction

VIDEO object cosegmentation refers to the problem that separates a common category of objects from multiple videos, which can be utilized in a number of computer vision tasks, *e.g.*, spatio-temporal action localization and video content understanding. Unlike single video based object segmentation, the video cosegmentation can benefit from semantic or structure information shared among multiple videos. The idea of leveraging shared information is also commonly used in other fields, such as motif detection [1] and action detection [2].

Currently, most methods aiming at this task [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] are cast into the energy minimization framework by exploiting pair-wise relations between two adjacent pixels/regions. They either utilize the low-level features (such as color and optical flow) [3], [4], [5], [9], [10], mid-level contextual features [11] or object proposals [6], [7], [8] to contribute the common object cosegmentation from multiple videos. Recently, Convolutional Neural Network (CNN)-based [13], [14] and Fully Convolutional Network (FCN)-based [15], [16] methods are also explored for video object segmentation/cosegmentation.

Despite the breakthroughs these methods achieved, most of them only focus on pair-wise correlations between pixels/regions while neglect the higher order correlations among multiple videos, which is critical for distinguishing foreground and background. Moreover, methods based on object proposal ubiquitously utilize a single object proposal per video frame, which will consistently fail to localize the common objects once the selected object proposal is inaccurate. Besides, most existing methods require all video frames capturing the common objects, which is an unrealistic assumption. Their performances will degrade dramatically, if the percentage of "empty" frames increases. In this paper, "empty" video frame is defined as frame without the common objects, as denoted by red cross in Figure 1.

To address these challenges, we propose a multilevel hypergraph based full Video object CoSegmentation (VCS) method, which accounts for high order correlations, incorporates multiple object proposals per video frame, and is robust to the presence of large amount of "empty" video frames (*i.e.*, object entries/exists/occlusions). Figure 1 summarizes the flowchart of the proposed VCS method. Given multiple noisy videos containing a common category of objects with the existence of many "empty" video frames, our proposed VCS method incorporates a hybrid object model for hyperedge computation, with one high-level model focused on video semantics, and a separate low-level model dedicated to video motion/saliency/appearance. Specifically, the high-level object model is designed for merging multiple object proposals to generate a more reliable frame-wise object region, thus producing more robust high-level features. The low-level features (*i.e.*, appearance, motion and saliency) naturally complement the high-level ones, jointly contributing to a better video representation. The hypergraph cut algorithm [17] is subsequently leveraged to obtain the final object cosegmentation result.

Manuscript received July 17, 2019; revised January 25, 2020 and March 16, 2020; accepted May 12, 2020. Date of publication XXX, XX, 2020; date of current version XXX, XX, 2020. This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 61629301, 61773312, and 61976171, China Postdoctoral Science Foundation Grant 2019M653642, and Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Pradeep Atrey. (*Corresponding author: Gang Hua.*)

L. Wang, X. Lv and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: {lewang,lvxin1,nnzheng}@mail.xjtu.edu.cn).

Q. Zhang is with HERE Technologies, Chicago, IL 60606, USA (e-mail: samqzhang@gmail.com).

Z. Niu is with Alibaba Group, Hangzhou, Zhejiang 311121, China (e-mail: niuzhenxing@gmail.com).

G. Hua is with Wormpex AI Research, Bellevue, WA 98004, USA (e-mail: ganghua@gmail.com).

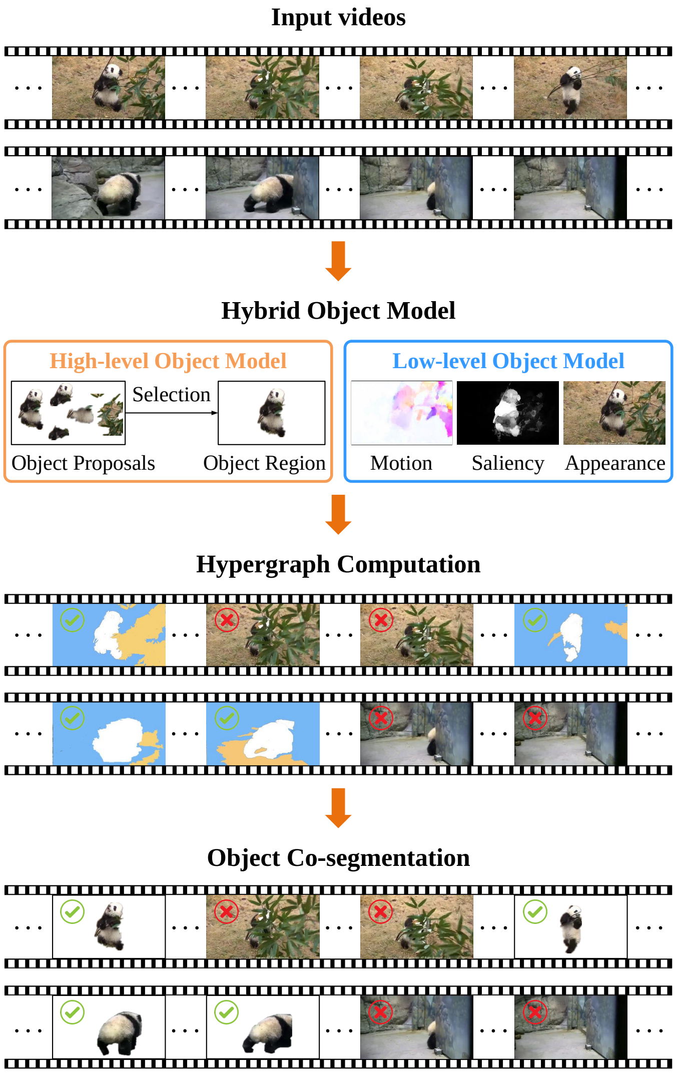


Fig. 1. The flowchart of the proposed multilevel hypergraph based full Video object CoSegmentation method (VCS). The input are multiple noisy videos containing a common category of objects with the existence of many "empty" video frames. After hypergraph computation with a hybrid object model, the final object cosegmentation results are obtained, where the "empty" video frames are marked by red cross, and the common objects are remained in the relevant video frames marked by green tick.

Extensive experiments are executed to evaluate the proposed VCS against state-of-the-art methods with both objective and subjective results on four video object segmentation/cosegmentation datasets, including 1) single-video object segmentation without "empty" frames on the SegTrack dataset [18], [19], 2) multi-video object cosegmentation without "empty" frames on the VCoSeg dataset [20], [21], [4], 3) multi-video object cosegmentation with a few (3.7%) "empty" frames on the XJTU-Stevens dataset [11], and 4) multi-video object cosegmentation with many (30.3%) "empty" frames on the Noisy-ViCoSeg dataset newly proposed in this paper. The experimental results clearly manifested the efficacy of the proposed VCS method against the competing ones, especially the ability of separating the common objects from multiple noisy videos with large portions of "empty" frames.

To isolate the contributions of each major components of our proposed VCS method, we further conduct ablation experiments to evaluate the effects of the high-level object model, the object proposal generation method, and the parameter setting.

The key contributions of this paper include:

- We propose a multilevel hypergraph-based full Video object CoSegmentation (VCS) method, which is robust against videos with "empty" video frames (*i.e.*, frames with object entries/exists/occlusions).
- We propose a hybrid object model consisting of both a high-level and a low-level object model, which accounts for both the high-level semantics and low-level features.
- By introducing the concept of object region in the high-level object model, our method achieves state-of-the-art performance even if the percentage of "empty" video frames is up to 30.3%[1].

This paper is an extension of our previous conference paper [22] with more technical details and improved readability. In contrast, there are three major changes. 1) A more comprehensive review is further made on related work. 2) More explanations are added for the problem formulation and implementation details. 3) The experiments and discussions section is fully reorganized by introducing new evaluation results and comparisons with state-of-the-arts.

The remainder of the paper is organized as follows. Section II briefly review the related work. The preliminaries of the hypergraph are presented in Section III. We formulate the video object cosegmentation from noisy videos in Section IV. Extensive experiments with detailed discussions are presented in Section V. In the last section, the whole paper is concluded.

## II. Related Work

In this section, we briefly survey recent related work in video object cosegmentation and video object segmentation, both with special emphasis on object proposal based methods, since our method tries to address the problem of video object cosegmentation from multiple noisy videos by leveraging object proposals.

### A. Video Object Cosegmentation

In recent years, a large amount of video object cosegmentation methods have been proposed to simultaneously separate the common object from two or more videos. These methods can be divided into two categories roughly, *i.e.*, video object cosegmentation from multiple videos without and with "empty" frames. Moreover, we specifically summarize the object proposal based methods.

*1) Methods for videos without "empty" frames.* Plenty of methods [3], [4], [23], [9], [24], [14] have been proposed to handle the traditional video object cosegmentation task, where each of the videos contains the common object in all frames, *i.e.*, there are no "empty" video frames. Chen *et al.* [3] identified object regions with coherent motion and then found

[1] This challenging video cosegmentation dataset is collected in this paper and will be publicly available, with both ground-truth categorical labels and pixel-wise foreground labels. Details can be found in Section V-G

the common object based on similar chroma and texture features. Rubio *et al.* [4] proposed an iterative process for figure-ground separation based on feature matching among frame regions and spatio-temporal tubes. Kowdle *et al.* [23] proposed an unsupervised hierarchical reasoning framework, which can combine appearance cues and multi-view cues. Wang *et al.* [9] proposed a subsequent quadratic pseudo-boolean optimization and a subspace clustering algorithm for video object cosegmentation. Wang *et al.* [24] explored multiple co-occurring matching and employed semantic information for video object cosegmentation. Li *et al.* [14] proposed a hierarchical deep cosegmentation method for aerial videos, that can segment primary video objects accurately and efficiently. The above methods almost all focus on videos without "empty" frames, and cannot handlle videos with "empty" frames.

*2) Methods for videos with "empty" frames*. Recently, many methods [25], [10], [15], [26], [11], [27], [12] focus on cosegmentation of the common objects from multiple videos with several "empty" frames. Chiu and Fritz [25] presented a non-parametric bayesian model that employs a spatio-temporal segmentation prior together with a global appearance model to cluster pixels into different regions. Wang *et al.* [10] integrated inter-frame consistency, intra-frame saliency, and across-video similarity into an energy optimization framework to segment the common object in multiple videos. Tsai *et al.* [15] utilized a fully convolutional network to extract semantic information, and thus facilitated the cosegmentation of objects within the same category from multiple videos. Wang *et al.* [26], [11] formulated the task using a spatio-temporal energy minimization framework that incorporates a spatio-temporal context model for joint discovery and cosegmentation of the common video object. Ma *et al.* [27] proposed a multi-class cosegmentation method by constructing a powerful hypergraph joint-cut framework, which utilizes intra-image feature representation as mid-level feature and $\ell_1$-manifold graph-based inter-image coherency exploration. However, these methods almost always dramatically deteriorate when facing videos with a large number of "empty" frames, although can handle videos with a few "empty" frames. Han *et al.* [12] introduced the concept of union background to improve the robustness by suppressing the image backgrounds.

*3) Object proposal-based methods*. Several methods [6], [8], [28], [7] proposed to leverage object proposals for video object cosegmentation. Zhang *et al.* [6] leveraged a regulated maximum weight clique extraction scheme to cosegment the common objects in video by sampling, tracking and matching object proposals. Lou and Gevers [8] constructed a probabilistic graphical model by employing the appearance, motion consistency, and saliency of object proposals to locate the common video objects. Li *et al.* [28] introduced an adaptive multi-search strategy to extend the previous object proposal selection based methods to realize unsupervised cosegmentation of indefinite numbers of common objects. Nevertheless, all of them are based on the assumption of a single object proposal per video frame, where inaccurate object proposal inevitably leads to performance degradation. Fu *et al.* [7] used object proposals as basic elements and then extracted multiple common objects by utilizing a graph model with multi-state selection.

### *B. Video Object Segmentation*

Video object segmentation addresses the problem that separates relevant objects from their surroundings in a video. Most segmentation methods are based on handcrafted features, deep features, or object proposals.

*1) Handcrafted feature-based methods*. Most such research efforts leveraged the low-level features, such as motion characteristics [29], appearance [30], [31], or saliency [29], [32]. Recently, Tsai *et al.* [33] simultaneously estimated video object segmentation and optical flow, and improved performance in both tasks iteratively. Wang *et al.* [34] introduced temporal saliency consistency measurement of superpixels as a prior for pixel-wise labeling, and then utilized graph cut to obtain the final segmentation results. Wang *et al.* [35] proposed a semi-supervised video segmentation method based on a video representation super-trajectory. Chen *et al.* [36] proposed a supervised object segmentation algorithm with multilevel model based on a more reasonable frame selection manner called supervision optimization.

*2) Deep feature-based methods*. With the popularity of deep learning, deep features have been applied in both unsupervised [37], [38], [39] and semi-supervised [40], [41] video segmentation methods. Hu *et al.* [37] proposed an unsupervised video object segmentation approach using saliency estimation and a graph neighborhood. Lu *et al.* [38] proposed a global co-attention siamese network for the unsupervised video object segmentation task. Wang *et al.* [39] proposed a novel attentive graph neural network (AGNN) for zero-shot video object segmentation. Cheng *et al.* [40] proposed a fast and accurate video object segmentation algorithm, which is able to start the segmentation of a specific object immediately. Griffin and Corso [41] introduced a deep sorting network termed BubbleNets that learns to select the guidance frame in video object segmentation. Thanks to the learning abilities of neural networks, deep feature-based video object segmentation methods usually outperforms their counterparts based on heuristics or handcrafted features.

*3) Object proposal-based methods*. Many methods [30], [42], [43], [44], [45] have been proposed to extract object-like regions or object proposals to facilitate video object segmentation. Lee *et al.* [30] proposed to discover object-like key-segments automatically and then predicted the foreground objects in a video by grouping techniques. Ma and Latecki [42] proposed to select object region candidates by finding the maximum weight clique in a weighted region graph, and utilized mutual constraints to obtain reliable segmentation of foreground object. Fragkiadaki *et al.* [43] proposed to separate moving objects in videos under the help of multiple segment proposal generation and ranking by utilizing moving objectness. Liu *et al.* [44] utilized a principled probabilistic model to discover and segment the object jointly by coupling a superpixel graph and an object proposal graph. Xu *et al.* [45] presented a detection based multiple hypotheses propagation method for video object segmentation, in which a proposal decision in one frame is delayed and augmented with long-term information to reduce ambiguity. However, almost all of them assume one object proposal per video frame, which could introduce potential performance degradation due to inaccurate proposal generation.

In contrast, there are three primary distinctions between our proposed VCS method and the above ones.

- The proposed VCS method is robust to substantial amount of "empty" video frames devoid of the common objects.
- Unlike conventional graph models where only pairwise relations between two vertices are explicitly addressed, the hypergraph represents complex correlations among multiple vertices in the proposed VCS method.
- The VCS method introduces the concept of object region by merging multiple object proposals in a video frame.

## III. Preliminaries of Hypergraph

Defining a weighted hypergraph as $G = \{V, E, \omega\}$, with the node set $V$ and the hyperedge set $E$, where $V = \{v_i\}$ denotes a finite set of nodes, $E$ denotes the hyperedge set containing a family of subsets of $V$ (such that $\cup_{e \in E} = V$), and each hyperedge $e$ is assigned a positive weight $\omega(e)$ [46]. A hyperedge $e$ is incident with a node $v$ when $v \in e$. For a node $v \in V$, its degree is defined as $d(v) = \sum_{e \in E | v \in e} \omega(e)$. For a hyperedge $e \in E$, its degree is defined as $\delta(e) = |e|$, which denotes the number of nodes that the hyperedge $e$ contains. A hypergraph $G$ can be represented by a $|V| \times |E|$ incidence matrix $H$ with entries $h(v, e) = 1$ if $v \in e$ and 0 otherwise. Then $d(v) = \sum_{e \in E} \omega(e) h(v, e)$, and $\delta(e) = \sum_{v \in V} h(v, e)$. Let $D_v$ and $D_e$ denote the diagonal matrices containing the node and hyperedge degrees, respectively. Let $W$ be the diagonal matrix containing the weighted hyperedges.

## IV. Problem Formulation

In this section, we cast the video object cosegmentation into the hypergraph cut framework. Therefore, separating the common objects is equivalent to partition the nodes (superpixels) $V$ of the hypergraph $G = (V, E, \omega)$ into a common object subset $S$ and a background (complement) subset $S^c$. If the hyperedge $e$ contains nodes from both $S$ and $S^c$, this hyperedge $e$ should be cut. The hyperedge boundary $\partial S := \{e \in E | e \cap S \neq \emptyset, e \cap S^c \neq \emptyset\}$ is a set of hyperedges. The volume of $S$ is the sum of the degrees of the nodes in $S$, which is defined as $vol(S) = \sum_{v \in S} d(v)$. The partition of the hypergraph leads to the hyperedge boundaries,

$$vol(\partial S) := \sum_{e \in S} \omega(e) \frac{|e \cap S||e \cap S^c|}{\delta(e)}, \tag{1}$$

where $\delta(e)$ is the degree of hyperedge $e$. It is clear that $vol(\partial S) = vol(\partial S^c)$. Like the normalized cut [47], a natural partition is achieved where internode connections within the same cluster are dense, while those across different clusters are sparse. Therefore, the two-way normalized hypergraph partition minimizes the bias of unbalanced partitioning as

$$\text{argmin}_{S \subset V} \text{Cut}(S) := vol(\partial S)\left(\frac{1}{vol(S)} + \frac{1}{vol(S^c)}\right). \tag{2}$$

Following the approximate solution in [17], we retain the first three eigenvectors with non-zero eigenvalues of $\Delta$ as the indicators and use the k-means ($k = 2$, object and background) on these eigenspace to get the final clustering/object cosegmentation results. The Laplacian matrix is defined as

$$\Delta = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} \tag{3}$$

where $D_v$, $H$, $W$ and $D_e$ are as described in Section III.

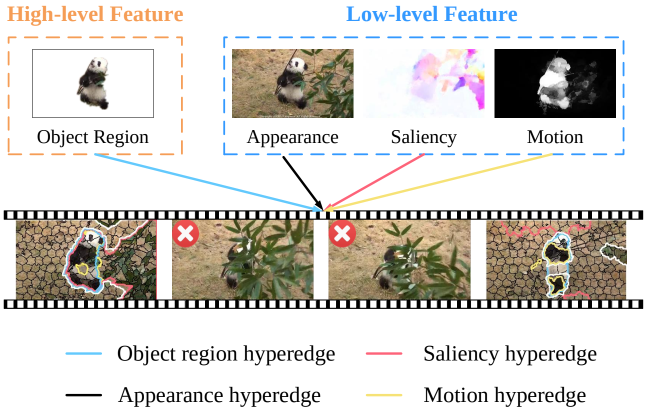


Fig. 2. Illustration of multilevel feature hyperedge.

Specifically, given a set of videos $\mathbf{F} = \{F^n\}_{n=1}^N$ containing a common object with the existence of "empty" frames, our objective is to find a binary cosegmentation labeling $\mathbf{B} = \{B^n\}_{n=1}^N$ of the common object from $\mathbf{F}$. Each video $F^n = \{f_t^n\}_{t=1}^T$ consists of $T$ frames, and similarly for its segmentation $B^n = \{B_t^n\}_{t=1}^T$. $B_t^n = \{b_{t,k}^n\}_{k=1}^K$ is the binary label of frame $f_t^n$, where $b_{t,k}^n \in \{0, 1\}$ denotes the segmentation label of superpixel $s_{t,k}^n \in f_t^n$ either belonging to the common object (its segmentation label $b_{t,k}^n = 1$) or the background (its segmentation label $b_{t,k}^n = 0$).

We proceed to present the hypergraph construction, the hypergraph computation by coupling a low-level and a high-level object model, and the hyperedge weights computation.

### A. Hypergraph Construction

Since we utilize superpixels[2] as the nodes of the hypergraph, for ease of presentation, we use nodes of $p$ and $q$ instead of superpixels $s_{t,k}^n$ and $s_{t',k'}^n$ from now on. Nodes with similar features are clustered into the same hyperedge with eigenvalue decomposition of Laplacian matrix $L = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}$. $A$ is the affinity matrix. $D$ is the diagonal matrix with

$$D(p, p) = \sum_q A(p, q), \tag{4}$$

where $A(p, q)$ means the affinity between the two nodes $p$ and $q$, and is obtained by coupling a low-level object and a high-level object model.

### B. Hyperedge Computed with a Low-level Object Model

The low-level object model computes the hyperedge by combining the motion, appearance and saliency cues, as shown in Figure 2.

[2] The superpixels used in our model is obtained by SLIC [48].

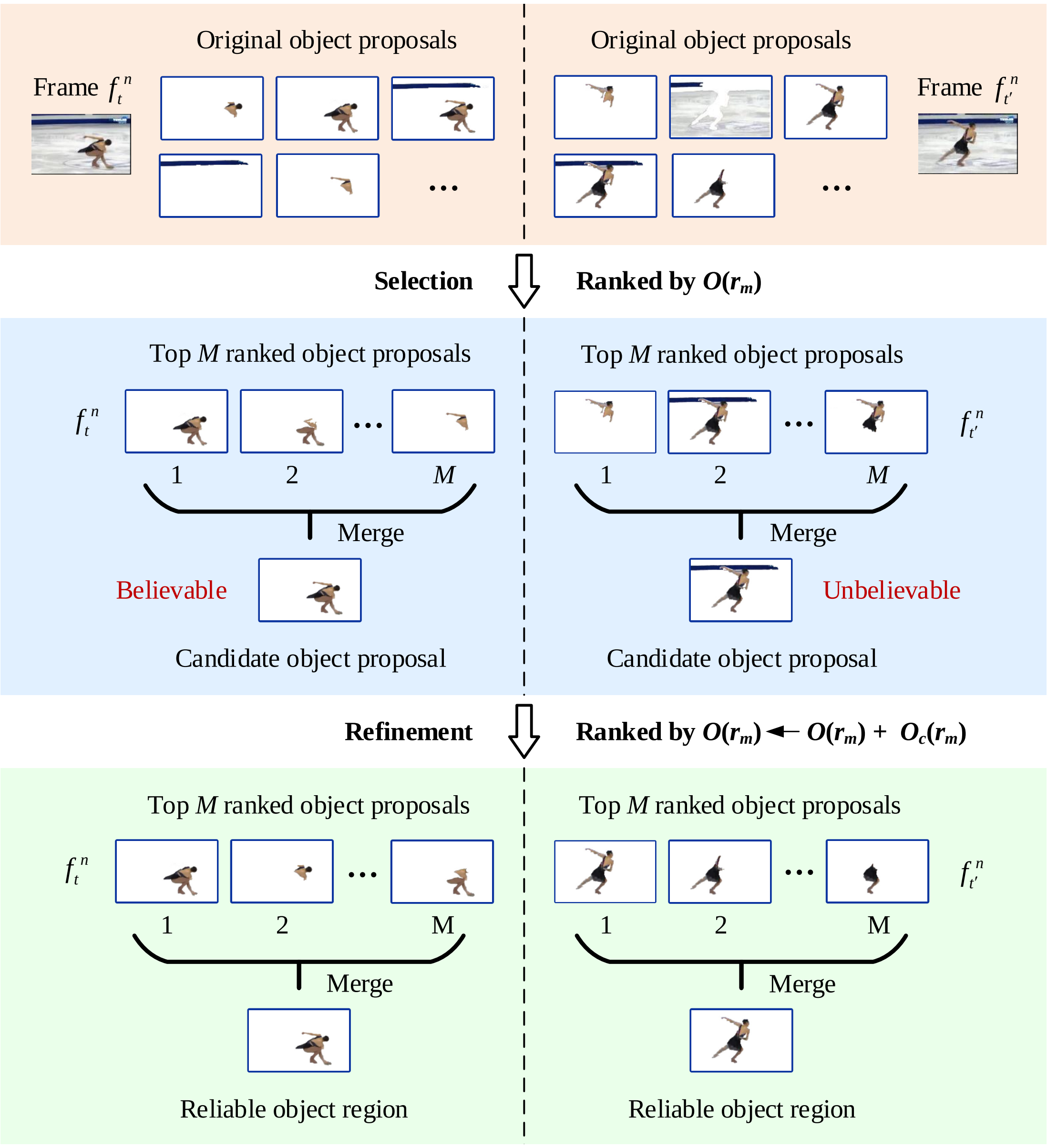

Original object proposals
Frame $f_t^n$
Original object proposals
Frame $f_{t'}^n$
Selection
Ranked by $O(r_m)$
Top $M$ ranked object proposals
$f_t^n$
1
2
$M$
Merge
Believable
Candidate object proposal
Top $M$ ranked object proposals
$f_{t'}^n$
1
2
$M$
Merge
Unbelievable
Candidate object proposal
Refinement
Ranked by $O(r_m) \leftarrow O(r_m) + O_c(r_m)$
Top $M$ ranked object proposals
$f_t^n$
1
2
M
Merge
Reliable object region
Top $M$ ranked object proposals
$f_{t'}^n$
1
2
M
Merge
Reliable object region

Fig. 3. Reliable object region generation. The left column and the right column represent the different frames of the same video, respectively. The orange area represents the original object proposals generated from each frame. The blue area represents that the highest ranking $M$ object proposals are merged to generate a candidate object proposal for each frame. The green area represents that the new top $M$ ranked object proposals are merged into a reliable object region for each frame.

*1) Motion.* Assume pixels from the same superpixel share identical motion pattern, the motion feature $P_m = (P_u, P_d)$ of each superpixel is calculated by using optical flow [49]. The motion intensity $P_u$ and direction $P_d$ of each superpixel can be computed as

$$P_u = \frac{1}{N_s}\sum_j \omega_j u_j, \tag{5}$$

$$P_d = \frac{1}{N_s}\sum_j \omega_j d_j, \tag{6}$$

where $N_s$ is the number of pixels in each superpixel. $\omega_j$ is a weight generated from a low-pass 2D Gaussian filter centered on the centroid of the superpixel. $u_j$ and $d_j$ are the motion intensity and direction of the $j$th pixel, respectively.

*2) Appearance.* Pixels from the same superpixel are highly likely to have similar color, the color feature $P_c$ of each superpixel are computed in $Lab$ color space as

$$P_c = \frac{1}{N_s}\sum_j c_j, \tag{7}$$

where $c_j$ is the color value of the $j$th pixel.

*3) Saliency.* The saliency detection method introduced in [50] is employed to generate the saliency map for each video frame. Then, the saliency value $P_s$ of each superpixel is computed by averaging the saliency values of all the pixels in it as

$$P_s = \frac{1}{N_s}\sum_j s_j, \tag{8}$$

where $s_j$ is the saliency value of the $j$th pixel.

The affinity between two nodes (superpixels) $p$ and $q$ based on the low-level object model is

$$A_l(p,q) = e^{-\frac{||P_l(p)-P_l(q)||_2}{\sigma^l}}, \tag{9}$$

where $l \in \{m, c, s\}$ denotes the motion, appearance and saliency, respectively. $\sigma^l$ is the deviation of $||P_l(p) - P_l(q)||_2$.

We argue that pure low-level information is insufficient for co-segmentation, because there is no concept of common object in hyperedges (as validated by experiments in Section V-C) computed purely with the low-level object model, which leads to the degeneration of our framework into a single-video object segmentation method. To segment the common object, we introduce the concept of object region in the high-level object model as presented immediately below.

### C. Hyperedge Computed with a High-level Object Model

The high-level object model creates a more reliable object region for each video frame to guide the hyperedge computation, as shown in Figure 2. The reliable object region generation procedure is illustrated in Figure 3. First, multiple object proposals are generated per frame [51], and a *video object score* $O(r_m)$ for each object proposal $r_m$ is estimated by combining appearance, motion, and semantic cues,

$$O(r_m) = O_a(r_m) + O_m(r_m) + O_s(r_m), \tag{10}$$

where $O_a(r_m)$ denotes the appearance score of $r_m$ with the objectness [51]. $O_a(r_m)$ will be assigned a high score if $r_m$ has a well-defined closed boundary and exhibits large distinction from its surroundings. $O_m(r_m)$ is the motion score of $r_m$, which is calculated by the average Frobenius norm of the gradient of optical flow around the boundary of $r_m$ [52]. $O_s(r_m)$ is the semantic score of $r_m$. Using an FCN [53] with ImageNet [54] pre-trained weights as initialization, we randomly select one video from each video category as the training set and obtain a meta network, which will be further finetuned on the first object frame of each video to get the final model for segmentation. After obtaining the segmentation results, the semantic score can be calculated based on them. The training details are presented in Section V-A.

Once the object proposals are measured and sorted by the video object score, the highest ranking $M$ object proposals (empirically $M = 20$) are merged to generate a candidate object region for each frame, which will be further refined to obtain a more reliable object region. Specifically, all candidate object proposals are clustered into two sets by k-means, *i.e.* a believable (dependable) set $Q_b$ and an unbelievable (undependable) set $Q_u$. By regarding the original object proposals that are used to merge the top $M$ ranked candidate object proposals in $Q_b$ as positive samples, and the remaining ones as negative samples, a linear SVM classifier can be trained. Specifically, the training samples of the SVM classifier are the output of the last convolutional layer of ResNet [55] after L2 normalization. Subsequently, all the original proposals can be classified by this SVM classifier, and the classification score of $r_m$ is noted as $O_c(r_m)$. Finally, we refine the *video object score* $O(r_m)$ of $r_m$ by

$$O(r_m) \leftarrow O(r_m) + O_c(r_m), \tag{11}$$

and merge the new top $M$ ranked object proposals into a reliable object region $\hat{r}$ for each frame. The whole generation process is illustrated as Figure 3, the object proposals with both object and background can be filtered out.

After acquiring the reliable object region $\hat{r}$ of each frame, the hyperedge is calculated under the guidance of $\hat{r}$. The nodes (superpixels) belonging to the reliable object region contribute to one hyperedge and the rest of nodes contribute to the alternative hyperedge. Therefore, the affinity between nodes $p$ and $q$ based on the high-level object model is calculated as

$$A_h(p,q) = \begin{cases} \frac{1}{M}\sum_m O(\hat{r}_m), & \text{if } p, q \in \hat{r}/\hat{r}^c \\ \frac{1}{N_p+N_q}, & \text{otherwise} \end{cases}, \tag{12}$$

where $\hat{r}_m$ denotes one of the $M$ object proposals that are merged into $\hat{r}$, and $\hat{r}^c$ denotes the remaining parts per frame. $N_p$ and $N_q$ are the numbers of pixels in $p$ and $q$, respectively.

### D. Hyperedge Weights Computation

With the obtained low-level and high-level affinity matrices ($A_l$ and $A_h$, respectively, in addition to $A_m$, $A_c$, and $A_s$), the corresponding Laplacian matrices $L_l$ and $L_h$ (also $L_m$, $L_c$, and $L_s$) can be obtained. Having obtained these Laplacian matrices, eigenvalue decomposition leads to the hyperedges, and the weight $\omega(e)$ for hyperedge $e$ is,

$$\omega(e) = c \cdot \frac{\sum_{p,q\in e} A(p,q)}{\sum_{p\in e, q\notin e} A(p,q)}, \tag{13}$$

where $c$ is a normalization constant to ensure $\sum_{e\in E}\omega(e) = 1$. A larger weight is assigned to a hyperedge $e$ if the affinity of nodes within this hyperedge (*i.e.* the numerator in Eq. (13)) is high, while the affinity of the nodes across different hyperedges (*i.e.* the denominator in Eq. (13)) is low, and vice versa. A large weight $\omega(e)$ prevents hyperedge $e$ from being cut, while a small weight $\omega(e)$ allows hyperedge $e$ to be cut. Depending on a hyperedge being low-level or high-level, $A(p,q)$ is computed by Eq. (9) or Eq. (12), respectively.

## V. Experiments and Discussions

### A. Implementation Details

The proposed Video object CoSegmentation (VCS) method is implemented in MATLAB, while the FCN is trained and fine-tuned with the Caffe framework with NVIDIA CuDNN libraries. The ImageNet pre-trained FCN model is trained on one video sampled from each video category and fine-tuned on the first object frame of each video with stochastic gradient descent (SGD) at a fixed learning rate of $10^{-14}$. The weight decay and momentum parameters are fixed to $5\times10^{-4}$ and 0.99. All experiments are conducted on an Intel Xeon 2.1GHz CPU with 64GB RAM and a NVIDIA Titan Xp GPU, for code on MATLAB and Caffe, respectively. During object cosegmentation inference, the proposed VCS method achieves a processing speed of approximately 0.5 frame per second.

### *B. Experimental Setting*

*1) Tasks and Evaluation Datasets.* Extensive experiments are conducted to evaluate the proposed VCS method against state-of-the-art methods with both objective and subjective results. Specifically, we evaluate our method on four different tasks, with corresponding dataset on each task as follows.

- **Task 1** - single-video object segmentation without "empty" frames on the SegTrack dataset [18], [19].
- **Task 2** - multi-video object cosegmentation without "empty" frames on the VCoSeg dataset [4], [20], [21].
- **Task 3** - multi-video object cosegmentation on the XJTU-Stevens dataset [11] with a few (3.7%) "empty" frames.
- **Task 4** - multi-video object cosegmentation on the Noisy-ViCoSeg dataset with many (30.3%) "empty" frames, which is proposed in this paper.

Table I summarizes the statistics of the above four datasets along with their application task.

*2) Evaluation Metric.* The segmentation performance is measured by the intersection-over-union (IoU) score, which is defined as

$$\text{IoU} = \frac{|Seg \cap GT|}{|Seg \cup GT|}, \tag{14}$$

where $Seg$ is the binary segmentation mask obtained by a video object segmentation/cosegmentation method, $GT$ is the binary ground truth segmentation mask obtained by human annotations, and $|\cdot|$ indicates the cardinality (*i.e.*, number of pixels).

We use the labeling accuracy to evaluate the performance of identifying the "empty" frames, which is defined as

$$L_{acc} = \frac{N_{tp} + N_{tn}}{N_{total}}, \tag{15}$$

where $N_{tp}$, $N_{tn}$ and $N_{total}$ denote the numbers of true positive, true negative and all video frames, respectively.

*3) Baselines.* Three methods for single-video object segmentation (VOS [30], FOS [29], and BVS [56]) and five methods for multi-video object cosegmentation (MVC [25], VOC [6], RVC [10], CBP [12] and MSG [7]) are selected as competing algorithms.

- VOS [30], a method for single-video object segmentation that can discover and group key segments automatically to separate the foreground object.
- FOS [29], a method for single-video object segmentation that can segment the foreground object by the efficient foreground estimation and figure/ground labeling refinement.
- BVS [56], a method for single-video object segmentation that can achieve foreground object segmentation via bilateral space operations with the usage of the object ground truth in the first frame.
- MVC [25], a method for multi-video object cosegmentation that can achieve multi-class object cosegmentation via a nonparametric bayesian model across multiple videos.
- VOC [6], a method for multi-video object cosegmentation that leverages a regulated maximum weight clique extraction scheme to cosegment the common objects in videos by sampling, tracking and matching object proposals.
- RVC [10] a method for multi-video object cosegmentation that integrates inter-frame consistency, intra-frame saliency, and across-video similarity into an energy optimization framework to cosegment the common objects in multiple videos.
- CBP [12] a method for multi-image object cosegmentation that introduces the concept of union background to improve the robustness by suppressing the image backgrounds.
- MSG [7] a method for multi-video object cosegmentation that uses object proposals as basic elements and then extracts multiple common objects by utilizing a graph model with multi-state selection.

### *C. Ablation Studies*

To isolate the contributions of individual high-level or low-level object model of the proposed VCS method, we conduct ablation experiments on multiple tasks, with ablated variants described as follow.

- VCS-L, an ablated variant of VCS with only the low-level object model. VCS-L are purely based on the low-level appearance/motion/saliency model, without any high-level semantic cues. Therefore, VCS-L is a degenerated single-video object segmentation method. We evaluate VCS-L on Task 1 and Task 2 but not on Task 3 or Task 4, considering VCS-L cannot handle "empty" frames without the high-level object model.
- VCS-O, an ablated variant of VCS with only the high-level object model. Without low-level cues, VCS-O relies purely on the reliable object regions generated by the high-level object model for segmentation. We evaluate VCS-O on all four tasks.
- VCS, the full version of the proposed method as shown in Fig. 1, which is evaluated on all four tasks and it is compared against other state-of-the-art methods.

As presented in Table II–III, VCS-L achieves comparable performance compared with other low-level features-based methods (VOS [30] and FOS [29]). Comparing the performance of VCS-L and VCS-O quantitatively (Table II–III) and qualitatively (Fig. 4–5), we find that the high-level semantics-based VCS-O significantly outperform the low-level features-based VCS-L. However, only considering high-level cues may result in the missing of segmentation details, such as the bird's feet and monkey's hands in Fig. 4.

With all components intact, the full version VCS achieves the best performance. From Table II–III, we observe that the "high-level object model" accounts more for the performance improvement. Additionally, by comparing VCS-O with VCS, we observe that the incorporation of low-level information further boosts the segmentation performance.

### *D. Single-video Object Segmentation without "Empty" Frames*

To evaluate the single-video object segmentation performance without "empty" frames, we carry out experiments on the SegTrack dataset [18], [19], which contains 14 videos, of which 8 videos have only one object, and the others capture multiple objects. Since our method is designed for single object segmentation, our method is only evaluated on the 8

TABLE I
THE STATISTICAL DETAILS AND APPLICATION TASK OF FOUR BENCHMARKS OR THEIR SUBSETS FOR EVALUATION OF THE PROPOSED VCS METHOD.

| **Dataset** | Task | Input | Group | Video | Frame | | | "Empty" Frame |
|---|---|---|---|---|---|---|---|---|
| | | | | | Total | Pos. | Neg. | |
| SegTrack [18], [19] | Segmentation | Single video | 8 | 8 | 785 | 785 | 0 | 0% |
| VCoSeg [4], [20], [21] | Cosegmentation | Multiple videos | 3 | 11 | 512 | 512 | 0 | 0% |
| XJTU-Stevens [11] | Cosegmentation | Multiple videos | 10 | 101 | 13398 | 12907 | 491 | 3.7% |
| Noisy-ViCoSeg | Cosegmentation | Multiple videos | 12 | 35 | 4835 | 3518 | 1465 | 30.3% |

TABLE II
THE SEGMENTATION PERFORMANCES OF OUR METHODS (VCS, VCS-L, AND VCS-O), AND THREE METHODS FOR SINGLE-VIDEO OBJECT SEGMENTATION (VOS [30], FOS [29], AND BVS [56]) ON THE SEGTRACK DATASET.

| **Video** | VOS [30] | FOS [29] | BVS [56] | VCS-L | VCS-O | VCS |
|---|---|---|---|---|---|---|
| birdfall2 | 49.4 | 17.5 | **63.5** | 51.8 | 61.9 | 62.0 |
| bird of paradise | 92.4 | 81.8 | 91.7 | 86.2 | 92.4 | **92.6** |
| frog | 75.7 | 54.1 | **76.4** | 57.0 | 73.6 | 73.5 |
| girl | 64.2 | 54.9 | 79.1 | 68.8 | **79.6** | **79.6** |
| monkey | 82.6 | 65.0 | **85.9** | 68.2 | 75.9 | 75.9 |
| parachute | **94.6** | 76.3 | 93.8 | 87.8 | 90.8 | 91.5 |
| soldier | 60.8 | 39.8 | 56.4 | 49.9 | 69.1 | **69.4** |
| worm | 62.2 | **72.8** | 65.5 | 62.0 | 68.1 | 68.1 |
| **Avg.** | 72.7 | 57.8 | 76.5 | 66.5 | 76.4 | **76.6** |

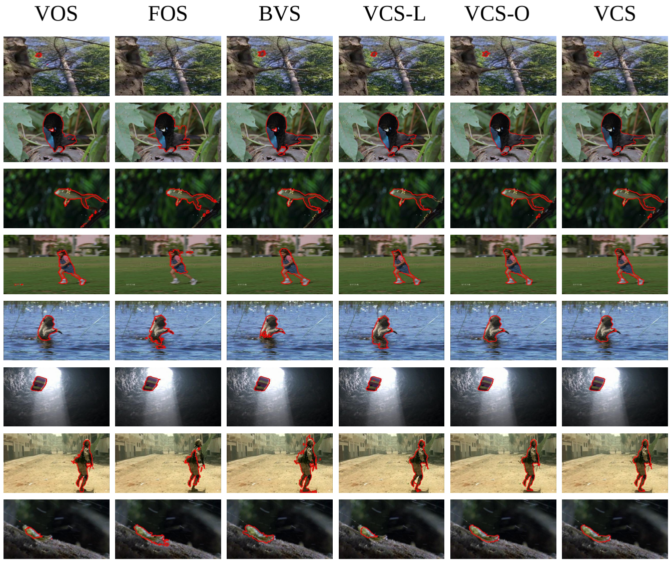


Fig. 4. Subjective segmentation results on the SegTrack dataset.

videos containing only one object. We compare our proposed method with VOS [30], FOS [29], and BVS [56], which are all proposed for the single-video object segmentation task.

We present IoU scores of these methods in Table II, and some subjective segmentation results of them in Figure 4. The results showed that our proposed VCS outperforms all the competing methods, and the VCS-L performs poorly compared to the proposed VCS and VCS-O, VOS [30] and BVS [56] by margins from 7.2% to 11%. While VCS-O is better than VOS [30] and FOS [29], but slightly underperforms BVS [56]. These demonstrate the efficacy of the reliable object region generated by the high-level semantics object model. Besides, although our method is designed for multi-video object cosegmentation, it still achieves state-of-the-art performance on single-video object segmentation.

TABLE III
THE SEGMENTATION PERFORMANCES OF OUR METHODS (VCS, VCS-L, AND VCS-O), THREE METHODS FOR SINGLE-VIDEO OBJECT SEGMENTATION (VOS [30], FOS [29], AND BVS [56]), AND TWO METHODS FOR MULTI-VIDEO OBJECT COSEGMENTATION (MVC [25] AND VOC [6]) ON THE VCOSEG DATASET.

| **Video** | VOS [30] | FOS [29] | BVS [56] | MVC [25] | VOC [6] | VCS-L | VCS-O | VCS |
|---|---|---|---|---|---|---|---|---|
| chachacha | 55.3 | 61.0 | 71.7 | 56.3 | 53.2 | 67.2 | **75.2** | **75.2** |
| ice skater | 82.1 | 81.6 | **83.5** | 69.1 | 65.3 | 66.3 | 80.1 | 80.7 |
| kite surfer | 69.1 | 35.0 | 65.9 | 38.2 | 51.6 | 52.9 | 69.1 | **69.3** |
| **Avg.** | 68.8 | 59.2 | 73.7 | 54.5 | 56.7 | 62.1 | 74.8 | **75.1** |

## *E. Multi-video Object Cosegmentation without "Empty" Frames*

The performance of object cosegmentation from multiple videos without "empty" frames is then evaluated on the VCoSeg dataset [4], [20], [21], which consists of 10 videos within 3 categories. And all video frames contain the common object. We compare our proposed method with two multi-video object cosegmentation methods (MVC [25] and VOC [6]), and three single-video object segmentation methods (*i.e.*, VOS [30], FOS [29], and BVS [56]). The reasons why we choose the three single-video object segmentation methods for comparison are two-fold, 1) to fairly compare with VCS-L, which is essentially a single-video object segmentation method degenerated from VCS, and 2) to explore the advantages of multi-video object cosegmentation over single-video object segmentation. For these single-video object segmentation methods, each video is individually segmented. The subsequent experiments on the XJTU-Stevens and the Noisy-ViCoSeg datasets follow the same setting.

Their IoU scores are compared in Table III with additional subjective segmentation results in Figure 5. The results reveal that, 1) our VCS and VSC-O are better than all the competing methods, and outperforms VOS [30], FOS [29], MVC [25], and VOC [6] by 6% $\sim$ 20.3%. This demonstrates that the multilevel hypergraph model (especially the high-level object model) in VCS contributes significantly to cosegmentation performance. 2) Our VCS-L perform better than FOS [29], MVC [25], and VOC [6], but poorer than VOS [30] and BVS [56]. We speculate that the performance gap could be attributed to the degenerated VCS-L being a single-video object segmentation method, which encapsulates only low-level appearance/motion/saliency cues in each individual video, but ignores high-level semantics across multiple videos. 3) Our VCS and VCS-O perform slightly better than BVS [56], we

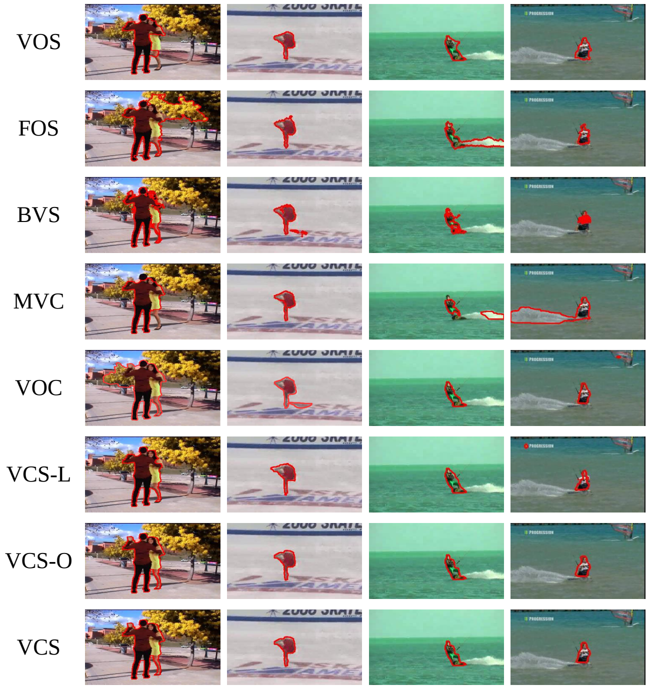


Fig. 5. Subjective segmentation results on the VCoSeg dataset.

TABLE IV
THE OBJECT DISCOVERY ACCURACIES OF OUR METHODS (VCS AND VCS-O), ONE METHOD FOR SINGLE-VIDEO OBJECT SEGMENTATION (BVS [56]), AND FIVE METHODS FOR MULTI-VIDEO OBJECT COSEGMENTATION (MVC [25], VOC [6], RVC [10], CBP [12] AND MSG [7]) ON THE XJTU-STEVENS DATASET.

| **Video** | APR | BVS<br>[56] | MVC<br>[25] | VOC<br>[6] | MSG<br>[7] | RVC<br>[10] | CBP<br>[12] | VCS |
|---|---|---|---|---|---|---|---|---|
| airplane | 97.1 | 95.7 | 97.1 | 97.1 | 97.1 | 98.9 | **99.7** | 99.6 |
| balloon | 95.9 | 97.3 | 96.5 | 95.9 | 95.9 | 98.9 | 96.2 | **99.1** |
| bear | 96.8 | 97.6 | 96.8 | 96.8 | 96.8 | 99.1 | 98.9 | **99.2** |
| cat | 96.5 | 96.5 | 96.5 | 96.5 | 96.5 | **97.3** | 90.0 | 96.5 |
| eagle | 97.7 | 97.0 | 97.7 | 97.7 | 97.7 | **98.8** | 97.7 | 97.7 |
| ferrari | 97.7 | 97.8 | 98.1 | 97.7 | 97.7 | 98.1 | 98.1 | **98.3** |
| figure skating | 96.3 | 96.3 | 96.3 | 96.3 | 96.3 | **98.3** | 94.2 | 97.5 |
| horse | 95.7 | 97.2 | 97.3 | 95.7 | 95.7 | 97.6 | 97.6 | **98.6** |
| parachute | 96.8 | 97.7 | 96.8 | 96.8 | 96.8 | 98.0 | 95.6 | **98.2** |
| single diving | 94.5 | 95.5 | 95.6 | 94.5 | 94.5 | 97.4 | 93.3 | **97.8** |
| **Avg.** | 96.5 | 96.9 | 96.7 | 96.5 | 96.5 | 98.2 | 96.1 | **98.3** |

speculate that the competitive performance of BVS [56] could originate from its leverage of the first-frame segmentation mask.

### *F. Multi-video Object Cosegmentation with a Few "Empty" Frames*

The performance of object cosegmentation from multiple videos with a few "empty" frames is further evaluated on the XJTU-Stevens dataset[11], which contains 101 publicly available internet videos of 10 categories. These videos contain 3.7% "empty" frames (with the common object absent) and varieties of challenging scenarios, including large variations in object appearance, scale, and angle of view. We test and compare our method against one method for single-video object segmentation (BVS [56]), four methods for multi-video object cosegmentation (MVC [25], VOC [6], RVC [10], and MSG [7]), and one method for multi-image object cosegmentation (CBP [12]). Among the multi-video object cosegmentation methods, VOC [6] and MSG [7] are also based on the refinement of object proposals, which is similar to our VCS method. Meanwhile, in order to explore how well single-video object segmentation methods perform on multiple videos with a few "empty" frames, we select BVS [56] for comparison, because it performs best among the competing single-video object segmentation methods on both the SegTrack [18], [19] and VCoSeg [4], [20], [21] datasets above in Table II–III. For CBP [12], all frames of each category are processed as individual images, and then are simultaneously segmented, where the temporal correlations among frames are ignored. This setting explores whether the temporal consistency can benefit the multi-video object cosegmentation.

We first evaluate the binary classification performance of distinguishing frames with the common object from "empty" ones. The object discovery accuracies are summarized in Table IV. Our proposed VCS achieves the highest accuracy among all methods. Specifically, the accuracies of BVS [56] are equivalent to or even lower than the APR[3] in some video categories, and the accuracies of VOS [30], MVC [25], CBP [12] and MSG [7] are merely marginally higher than the APR. Overall, RVC [10] achieves the second highest performance, mainly due to its discovery energy function that focuses on common objects. Compared with the proposed VCS method, these competing methods achieve object discovery accuracies on par with or slightly better than the random guess (APR), possibly suffering from performance penalties incurred by the "empty" frames.

The average IoU scores are presented in Table V with additional subjective segmentation results in Figure 6. Note that, 1) the ablated VCS-L (with low-level object model only) cannot handle videos with "empty" frames, thus it is excluded. 2) The average IoU scores are computed on video frames containing the common objects. As shown in Table V, methods for multi-video/multi-image object cosegmentation (MVC [25], VOC [6], RVC [10], CBP [12] and MSG [7]) perform better than the single-video object segmentation B-VS [56] method. 3) From Table III–V, single-video object segmentation method BVS [56] degenerates a lot when tested on datasets with "empty" frames, which is also indicated in Figure 6. 4) The multi-image object cosegmentation methd CBP [12] underperforms most of the multi-video object cosegmentation (MVC [25], VOC [6], RVC [10], and our VCS and VCS-O) except MSG [7], because it cannot employ motion cue to enhance the temporal consistency of the common object. 5) Both our VCS and VCS-O are superior to all other competing methods, especially to the single-video object segmentation method BVS [56] by an obvious margin.

### *G. Multi-video Object Cosegmentation with Many "Empty" Frames*

The performance of object cosegmentation from multiple videos with many "empty" frames is evaluated on the Noisy-

[3] A random guess classifier that always predicts positive will achieve the actual positive rate (APR) in Table IV.

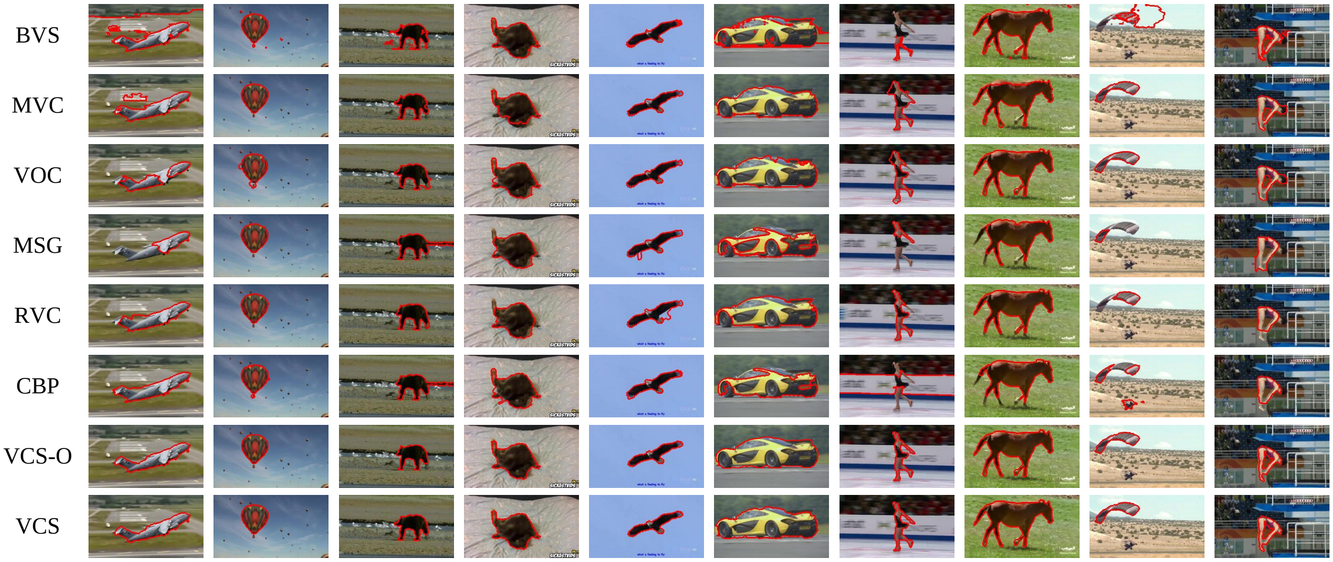


Fig. 6. Subjective segmentation results on the XJTU-STEVENS dataset.

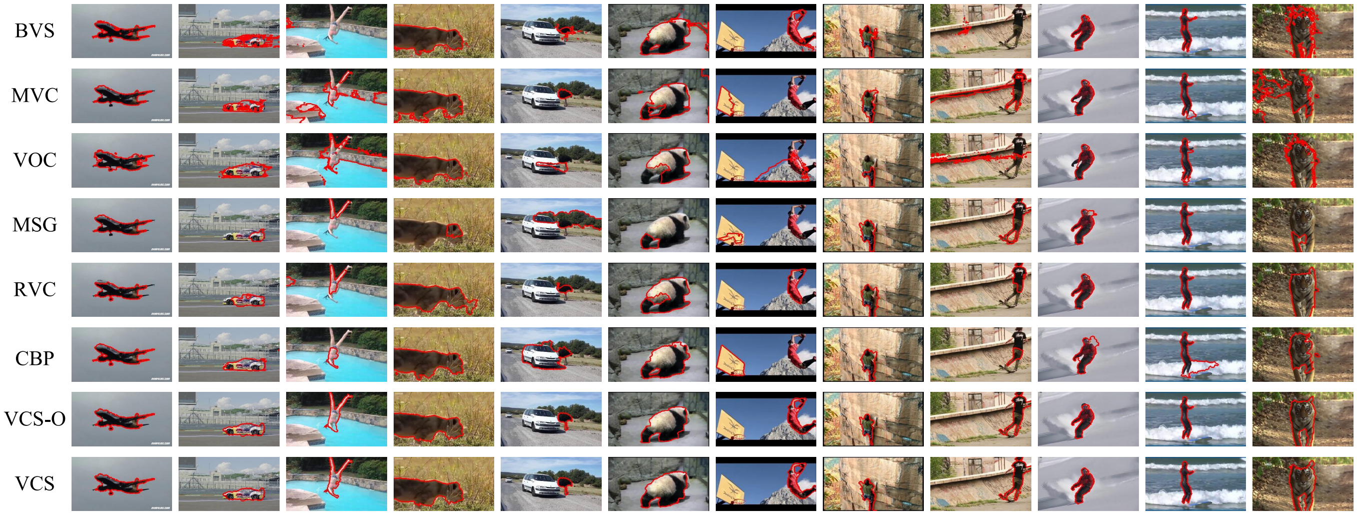


Fig. 9. Subjective segmentation results on the proposed Noisy-ViCoSeg dataset.

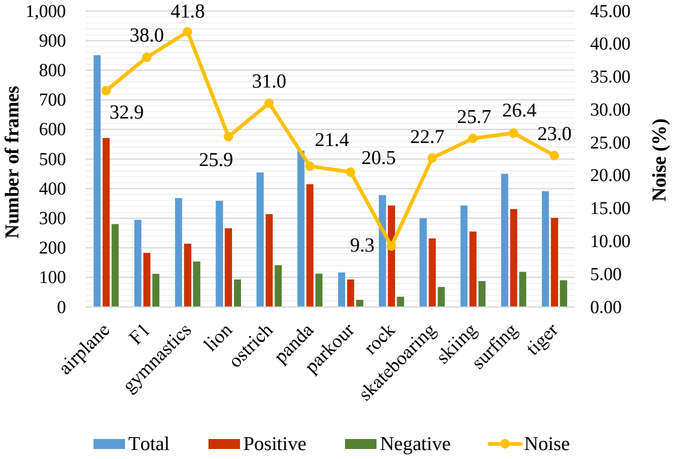


Fig. 7. The numbers of total frames, positive frames and negative frames with the percentage of "empty" frames of each category of the newly proposed Noisy-ViCoSeg dataset.

ViCoSeg Dataset. This dataset is newly collected and proposed in this paper, and it consists of 35 videos with 4,835 frames of 12 categories in total, where each video contains a large proportion of "empty" frames. 30.3% of video frames are "empty" ones without a common object on average. Figure 7 details the statistics. We manually assign a per-frame label indicating whether each individual frame contains the common object or not, and each non-"empty" video frame is also annotated with a per-pixel segmentation mask, as illustrated in Figure 8. We evaluate our proposed VCS method against one method for single-video object segmentation (BVS [56]), four methods for multi-video object cosegmentation (MVC [25], VOC [6], RVC [10], and MSG [7]), and one multi-image object cosegmentation method (CBP [12]). Here, the competing methods are the same as the ones for the above comparison on multi-video object cosegmentation with a few "empty" frames, in order to further explore their abilities to handle multiple videos with more "empty" frames.

TABLE V
THE SEGMENTATION PERFORMANCES OF OUR METHODS (VCS AND VCS-O), ONE METHOD FOR SINGLE-VIDEO OBJECT SEGMENTATION (BVS [56]), AND FIVE METHODS FOR MULTI-VIDEO OBJECT COSEGMENTATION (MVC [25], VOC [6], RVC [10], CBP [12] AND MSG [7]) ON THE XJTU-STEVENS DATASET.

| **Video** | BVS [56] | MVC [25] | VOC [6] | MSG [7] | RVC [10] | CBP [12] | VCS-O | VCS |
|---|---|---|---|---|---|---|---|---|
| airplane | 35.1 | 57.6 | 58.6 | 53.2 | 48.4 | **70.7** | 66.2 | 66.7 |
| balloon | 78.7 | 86.8 | 86.8 | 81.8 | 90.2 | 73.8 | 90.2 | **93.2** |
| bear | 85.7 | 80.8 | 83.2 | 84.9 | **96.7** | 81.7 | 84.9 | 88.0 |
| cat | 71.2 | 75.2 | **78.8** | 64.3 | 55.3 | 56.8 | 73.8 | 73.7 |
| eagle | 59.4 | 72.2 | 78.4 | 70.1 | 69.4 | 78.1 | 77.9 | **79.3** |
| ferrari | 61.3 | 75.4 | 61.8 | 41.3 | 69.7 | 67.4 | **83.2** | 82.9 |
| figure skating | 48.7 | 61.7 | 65.3 | 32.9 | **76.1** | 36.1 | 69.5 | 69.9 |
| horse | 79.5 | 80.3 | 85.1 | 70.9 | **85.6** | 77.4 | 80.2 | 82.7 |
| parachute | 76.4 | 80.8 | **83.4** | 43.6 | 66.9 | 53.0 | 78.7 | 79.1 |
| single diving | 35.6 | 59.1 | **69.5** | 42.3 | 62.9 | 52.4 | 63.5 | 63.0 |
| **Avg.** | 63.2 | 73.0 | 75.1 | 58.5 | 72.1 | 64.7 | 76.8 | **77.9** |

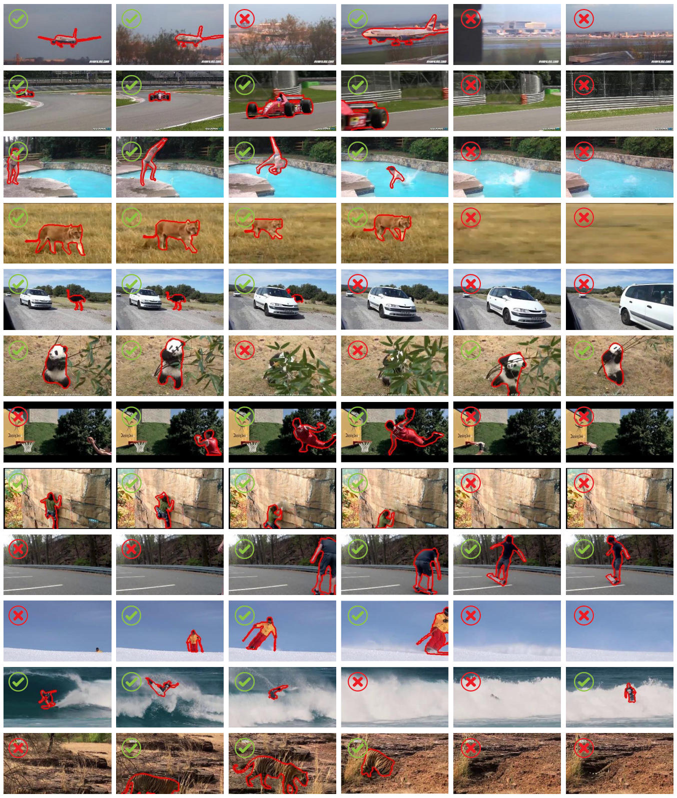


Fig. 8. Some sample frames with annotations of the Noisy-ViCoSeg dataset. The red cross indicates “empty” frames; the green tick indicates positive frames containing the common object enveloped by the red edge.

The object discovery accuracies (distinguishing frames with the common object from the “empty” ones) are presented in Table VI. Our VCS method outperforms all other methods by a significant margin of $4.0\% \sim 18.8\%$, mainly due to its designed robustness against “empty” frames.

With their IoU scores in Table VII and some subjective segmentation results in Figure 9, both the proposed VCS and VCS-O are superior to the competing methods by a margin of $5.9\% \sim 30.7\%$. As illustrated in Figure 9, other object segmentation or co-segmentation methods repeatedly fail to focus on the common object. However, our methods (VCS-O and VCS) manage to segment the common object, which indicates their robustness against distractions from “empty” frames and appearance/scale/deformation variations.

TABLE VI
THE OBJECT DISCOVERY ACCURACIES OF OBJECT DISCOVERY OF OUR METHODS (VCS AND VCS-O), ONE METHOD FOR SINGLE-VIDEO OBJECT SEGMENTATION (BVS [56]), AND FIVE METHODS FOR MULTI-VIDEO OBJECT COSEGMENTATION (MVC [25], VOC [6], RVC [10], CBP [12] AND MSG [7]) ON THE PROPOSED NOISY-VICOSEG DATASET.

| **Video** | APR | BVS [56] | MVC [25] | VOC [6] | MSG [7] | RVC [10] | CBP [12] | VCS |
|---|---|---|---|---|---|---|---|---|
| airplane | 73.4 | 85.2 | 93.2 | 73.4 | 73.4 | 87.8 | 86.4 | **98.1** |
| F1 | 66.3 | 90.3 | 81.8 | 66.3 | 66.3 | 88.9 | 90.5 | **99.1** |
| gymnastics | 61.7 | 61.7 | 61.7 | 61.7 | 61.7 | 77.0 | 63.0 | **100.0** |
| lion | 74.9 | 74.9 | 90.2 | 74.9 | 74.9 | 90.4 | 81.9 | **92.3** |
| ostrich | 69.7 | 74.4 | 75.9 | 69.7 | 69.7 | 84.1 | 69.7 | **96.7** |
| panda | 75.7 | 75.7 | 82.2 | 75.7 | 75.7 | 82.2 | 79.2 | **83.0** |
| parkour | 77.1 | 63.2 | 77.1 | 77.1 | 77.1 | **91.6** | 77.1 | 91.4 |
| rock | 85.1 | 85.1 | 89.5 | 85.1 | 85.1 | 97.5 | 95.8 | **99.3** |
| skateboarding | 80.9 | 80.9 | 80.9 | 80.9 | 80.9 | 88.7 | 80.9 | **94.2** |
| skiing | 81.4 | 70.7 | 96.2 | 81.4 | 81.4 | 91.8 | 89.1 | **99.0** |
| surfing | 79.8 | 85.6 | 79.8 | 79.8 | 79.8 | 88.2 | 87.0 | **93.6** |
| tiger | 74.4 | 74.4 | 75.4 | 74.4 | 74.4 | **92.7** | 90.8 | 91.8 |
| **Avg.** | 73.7 | 77.2 | 82.1 | 73.7 | 73.7 | 88.5 | 82.6 | **92.5** |

TABLE VII
THE SEGMENTATION PERFORMANCES OF OUR METHODS (VCS AND VCS-O), ONE METHOD FOR SINGLE-VIDEO OBJECT SEGMENTATION (BVS [56]), AND FIVE METHODS FOR MULTI-VIDEO OBJECT COSEGMENTATION (MVC [25], VOC [6], RVC [10], CBP [12] AND MSG [7]) ON THE PROPOSED NOISY-VICOSEG DATASET.

| **Video** | BVS [56] | MVC [25] | VOC [6] | MSG [7] | RVC [10] | CBP [12] | VCS-O | VCS |
|---|---|---|---|---|---|---|---|---|
| airplane | 54.7 | 51.8 | 34.1 | 54.9 | 51.6 | **83.8** | 72.8 | 73.6 |
| F1 | 15.3 | 56.3 | 19.3 | 35.6 | 31.9 | **70.1** | 64.1 | 64.1 |
| gymnastics | 19.7 | 23.1 | 10.4 | 33.9 | 35.0 | 31.3 | 62.1 | **62.3** |
| lion | 50.2 | **74.7** | 70.9 | 46.2 | 73.8 | 72.3 | 48.8 | 49.0 |
| ostrich | **61.6** | 20.0 | 1.1 | 1.8 | 50.4 | 4.8 | 54.1 | 53.6 |
| panda | 51.7 | 30.6 | 45.7 | 28.2 | 68.5 | **83.9** | 70.3 | 72.9 |
| parkour | 39.8 | 60.6 | 38.9 | 10.3 | **70.1** | 25.0 | 65.5 | 65.5 |
| rock | 64.1 | 46.4 | 53.0 | 62.9 | 58.6 | 61.2 | 64.4 | **64.6** |
| skateboarding | 53.5 | 38.9 | 8.0 | 58.9 | 59.0 | 42.3 | 59.4 | **59.6** |
| skiing | 49.0 | 75.2 | 36.6 | 72.9 | **83.2** | 71.6 | 70.2 | 74.1 |
| surfing | 66.8 | 52.0 | 53.6 | 71.5 | 61.3 | 45.0 | **75.1** | 74.8 |
| tiger | 53.9 | 31.5 | 28.7 | 16.4 | **54.9** | 41.7 | 53.2 | 54.6 |
| **Avg.** | 48.4 | 46.8 | 33.4 | 41.1 | 58.2 | 52.8 | 63.3 | **64.1** |

### *H. Parameter Sensitivity Analysis*

As illustrated in Figure 3, top-$M$ ranked object proposals are selected and merged into the reliable object region in the high-level object model. To evaluate the parameter sensitivity on the choice of $M$, we test our VCS method with varying $M = 5, \cdots, 30$ on the Noisy-ViCoSeg dataset. We summarize the average IoU scores in Table VIII, and additional subjective segmentation results in Figure 10. As shown in Table VIII, our VCS performs better on most video categories with $M = 20$ than other choices. As shown in Figure 10, the merged object regions are often incomplete with $M < 20$, but may contain excessive amount of background with $M > 20$. We thus empirically set $M = 20$ throughout our experiments.

### *I. Computational Cost Analyses*

We proceed to evaluate the computational cost of our proposed VCS with the average execution time per frame

TABLE VIII
THE SEGMENTATION PERFORMANCES OF OUR PROPOSED VCS METHOD BY VARYING THE VALUE OF $M$ ON THE PROPOSED NOISY-VICOSEG DATASET.

| **Video** | $M = 5$ | $M = 10$ | $M = 15$ | $M = 20$ | $M = 25$ | $M = 30$ |
|---|---|---|---|---|---|---|
| F1 | 43.8 | 60.9 | 62.7 | **64.1** | 62.9 | 61.9 |
| gymnastics | 59.8 | **68.8** | 64.2 | 62.1 | 39.0 | 34.7 |
| lion | 32.9 | 33.8 | 47.2 | 48.8 | 50.2 | **52.1** |
| parkour | 38.5 | 48.3 | 62.3 | **65.5** | 60.6 | 38.9 |
| rock | 56.8 | 62.3 | 63.6 | **64.4** | 60.0 | 59.5 |
| skateboarding | 47.6 | 57.2 | 58.9 | **59.4** | 56.4 | 56.3 |
| surfing | 72.0 | 72.7 | 73.5 | **75.1** | 74.2 | 71.1 |
| tiger | 34.8 | 45.2 | 50.3 | 53.2 | 55.8 | **56.7** |
| **Avg.** | 48.3 | 56.2 | 60.3 | **61.6** | 57.4 | 53.9 |

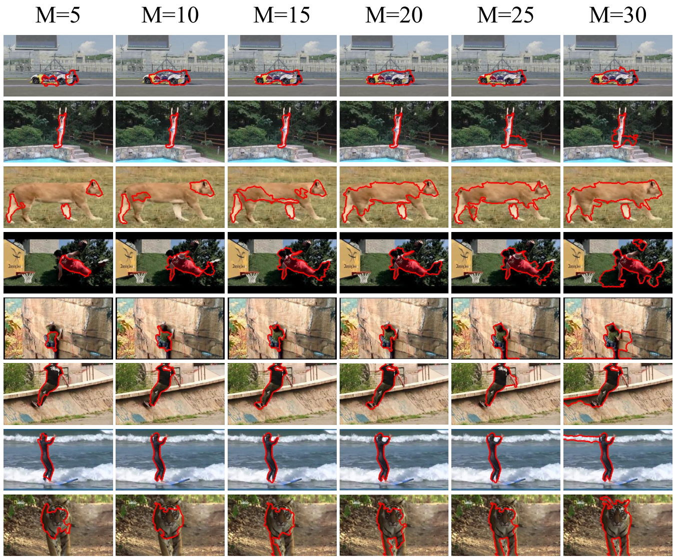


Fig. 10. Subjective segementation results of our proposed VCS method by varying the value of $M$ on the proposed Noisy-ViCoSeg dataset.

TABLE IX
AVERAGE EXECUTION TIME PER FRAME OF THE PREPARATION STEP AND THE COSEGMENTATION STEP.

| Steps | Preparation | | | | | Cosegmentation | | |
|---|---|---|---|---|---|---|---|---|
| | SG | MF | SF | PG | FCN | HEL | HEH | HC |
| Time(s) | 1.9 | 0.8 | 0.5 | 18.0 | 0.9 | 0.57 | 1.06 | 0.37 |

TABLE X
AVERAGE EXECUTION TIME PER FRAME OF OUR PROPOSED VCS METHOD AND FIVE METHODS FOR MULTI-VIDEO OBJECT COSEGMENTATION (MVC [25], VOC [6], RVC [10], CBP [12], AND MSG [7]) ON THE NOISY-VICOSEG DATASET.)

| **Method** | MVC [25] | VOC [6] | MSG [7] | RVC [10] | CBP [12] | VCS |
|---|---|---|---|---|---|---|
| **Time(s)** | 9.2 | 15.3 | 0.3 | 10.1 | 8.3 | 2.0 |

on the Noisy-ViCoSeg dataset. Our proposed VCS is divided into two main steps for computational cost analyses, *i.e.*, a preparation step and a cosegmentation step.

The preparation step includes superpixel generation (SG), motion feature computing (MF), saliency feature computing (SF), original object proposals generation (PG) and FCN segmentation (FCN). The left part of Table IX summarizes the average execution time per frame of the preparation step. Potentially, the procedures in preparation step can be accelerated using parallel computing.

The core computational cost of our VCS method comes from the cosegmentation step, which is further divided into hypergraph construction and hypergraph cut (HC). Hypergraph construction includes the hypergraph nodes construction, hyperedge computation with a low-level object model (HEL) and a high-level object model (HEH). Since hypergraph nodes construction is implemented during superpixel generation (S-G), and thus no extra time is consumed. The right part of Table IX lists the average execution time per frame of the cosegmentation step.

To further evaluate the efficiency of our proposed VCS against other competing methods, we compare the time cost of our proposed VCS with five multi-video object cosegmentation methods (MVC [25], VOC [6], RVC [10], CBP [12], and MSG [7]) on the Noisy-ViCoSeg dataset. Table X presents the average execution time per frame of them. The results show that our proposed VCS has competitive computational efficiency compared to the other state-of-the-art methods, and is only slightly slower than MSG [7]. However, MSG achieves the highest speed but meanwhile sacrificing the segmentation accuracy, as illustrated in Table VII; while our proposed VCS achieves a better balance between efficiency and accuracy.

In summary, the experiments on the above four datasets reveal that the proposed VCS method achieves competitive performance compared with the state-of-the-arts. Moreover, the ablation studies demonstrate the efficacy of the high-level object model. Most importantly, the experimental results indicate that the performance gap between our proposed VCS method and competing ones becomes larger when the portion of "empty" frames increases. This manifests that our proposed VCS method is capable of cosegmenting the common objects from multiple videos with large portions of "empty" video frames.

## VI. CONCLUSION

In this paper, we propose a full Video object CoSegmentation method (VCS), which is capable of cosegmenting the common objects in multiple videos automatically, with the robustness to substantial amount of "empty" frames. Benefiting from a multilevel hypergraph architecture, the proposed VCS method is capable of incorporating both a low-level object model and a high-level semantic model, which incorporates multiple object proposals per video frame, accounts for high order correlations, and thus contributes to its robustness against object entries/exists/occlusions. Empirical results on four video object segmentation/cosegmentation datasets verified the performance advantage of our VCS method.

## REFERENCES

[1] A. Mueen and N. Chavoshi, "Enumeration of time series motifs of all lengths," *Knowledge and Information Systems*, vol. 45, no. 1, pp. 105–132, 2015.

[2] C. Panagiotakis, K. Papoutsakis, and A. Argyros, "A graph-based approach for detecting common actions in motion capture data and videos," *Pattern Recognition*, vol. 79, pp. 1–11, 2018.

[3] D.-J. Chen, H.-T. Chen, and L.-W. Chang, "Video object cosegmentation," in *Proc. ACM Multimedia*, 2012, pp. 805–808.

[4] J. C. Rubio, J. Serrat, and A. López, "Video co-segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 13–24.

[5] J. Guo, Z. Li, L.-F. Cheong, and S. Zhiying Zhou, "Video co-segmentation for meaningful action extraction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2232–2239.

[6] D. Zhang, O. Javed, and M. Shah, "Video object co-segmentation by regulated maximum weight cliques," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 551–566.

[7] H. Fu, D. Xu, B. Zhang, S. Lin, and R. K. Ward, "Object-based multiple foreground video co-segmentation via multi-state selection graph," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3415–3424, 2015.

[8] Z. Lou and T. Gevers, "Extracting primary objects by video co-segmentation," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2110–2117, 2014.

[9] C. Wang, Y. Guo, J. Zhu, L. Wang, and W. Wang, "Video object co-segmentation via subspace clustering and quadratic pseudo-boolean optimization in an mrf framework," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 903–916, 2014.

[10] W. Wang, J. Shen, X. Li, and F. Porikli, "Robust video object cosegmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, 2015.

[11] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 10, pp. 2074–2088, 2017.

[12] J. Han, R. Quan, D. Zhang, and F. Nie, "Robust object co-segmentation using background prior," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1639–1651, 2018.

[13] J. Li, A. Zheng, X. Chen, and B. Zhou, "Primary video object segmentation via complementary cnns and neighborhood reversible flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1417–1425.

[14] J. Li, P. Yuan, D. Gu, and Y. Tian, "Hierarchical deep cosegmentation of primary objects in aerial videos," *IEEE MultiMedia*, vol. 26, no. 3, pp. 9–18, 2018.

[15] Y.-H. Tsai, G. Zhong, and M.-H. Yang, "Semantic co-segmentation in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 760–775.

[16] A. Mustafa and A. Hilton, "Semantically coherent co-segmentation and reconstruction of dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 422–431.

[17] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1738–1745.

[18] D. Tsai, M. Flagg, and J. Rehg, "Motion coherent tracking with multi-label MRF optimization," in *Proc. British Mach. Vis. Conf.*, 2010, pp. 56–67.

[19] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2192–2199.

[20] F. Tiburzi, M. Escudero, J. Bescós, and J. M. Martínez, "A ground truth for motion-based video-object segmentation," in *Proc. IEEE Int. Conf. Image Process.*, 2008, pp. 17–20.

[21] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2141–2148.

[22] X. Lv, L. Wang, Q. Zhang, N. Zheng, and G. Hua, "Video object co-segmentation from noisy videos by a multi-level hypergraph model," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 2207–2211.

[23] A. Kowdle, S. N. Sinha, and R. Szeliski, "Multiple view object cosegmentation using appearance and stereo cues," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 789–803.

[24] C. Wang, H. Zhang, L. Yang, X. Cao, and H. Xiong, "Multiple semantic matching on augmented $n$-partite graph for object co-segmentation," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5825–5839, 2017.

[25] W.-C. Chiu and M. Fritz, "Multi-class video co-segmentation with a generative multi-video model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 321–328.

[26] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng, "Video object discovery and co-segmentation with extremely weak supervision," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 640–655.

[27] J. Ma, S. Li, H. Qin, and A. Hao, "Unsupervised multi-class co-segmentation via joint-cut over $l_1$-manifold hyper-graph of discriminative image regions," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1216–1230, 2017.

[28] K. Li, J. Zhang, and W. Tao, "Unsupervised co-segmentation for indefinite number of common foreground objects," *IEEE Trans. on Image Process.*, vol. 25, no. 4, pp. 1898–1909, 2016.

[29] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1777–1784.

[30] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1995–2002.

[31] A. Khoreva, F. Galasso, M. Hein, and B. Schiele, "Classifier based graph construction for video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 951–960.

[32] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *Proc. British Mach. Vis. Conf.*, 2014.

[33] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3899–3908.

[34] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, 2018.

[35] W. Wang, J. Shen, F. Porikli, and R. Yang, "Semi-supervised video object segmentation with super-trajectories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 985–998, 2018.

[36] Y. Chen, C. Hao, A. X. Liu, and E. Wu, "Multi-level model for video object segmentation based on supervision optimization," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 1934–1945, 2019.

[37] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 786–802.

[38] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3623–3632.

[39] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9236–9245.

[40] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang, "Fast and accurate online video object segmentation via tracking parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7415–7424.

[41] B. A. Griffin and J. J. Corso, "Bubblenets: Learning to select the guidance frame in video object segmentation by deep sorting frames," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8914–8923.

[42] T. Ma and L. J. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 670–677.

[43] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4083–4090.

[44] Z. Liu, L. Wang, G. Hua, Q. Zhang, Z. Niu, Y. Wu, and N. Zheng, "Joint video object discovery and segmentation by coupled dynamic markov networks," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5840–5853, 2018.

[45] S. Xu, D. Liu, L. Bao, W. Liu, and P. Zhou, "Mhp-vos: Multiple hypotheses propagation for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 314–323.

[46] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 1601–1608.

[47] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[48] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[49] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, 2012.

[50] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, 2015.

[51] I. Endres and D. Hoiem, "Category independent object proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 575–588.

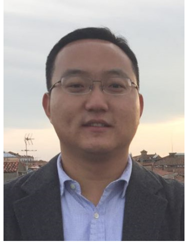
[52] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 628–635.

[53] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[55] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[56] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, “Bilateral space video segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 743–751.



**Le Wang** (M’14) received the B.S. and Ph.D. degrees in Control Science and Engineering from Xi’an Jiaotong University, Xi’an, China, in 2008 and 2014, respectively. From 2013 to 2014, he was a visiting Ph.D. student with Stevens Institute of Technology, Hoboken, New Jersey, USA. From 2016 to 2017, he is a visiting scholar with Northwestern University, Evanston, Illinois, USA. He is currently an Associate Professor with the Institute of Artificial Intelligence and Robotics of Xi’an Jiaotong University, Xi’an, China. His research interests include computer vision, pattern recognition, and machine learning. He is the author of more than 30 peer reviewed publications in prestigious international journals and conferences. He is a member of the IEEE.

**Xin Lv** received the B.S. degree in Automation Engineering from Guilin University of Electronic Technology, Guilin, China, in 2015, and the M.S. degree in Software Engineering from Xi’an Jiaotong University, Xi’an, China, in 2018. She is currently an Engineer with the Network Information Center of Xi’an Jiaotong University, Xi’an, China. Her research interests include computer vision and machine learning.

**Qilin Zhang** (M’18) received the B.E. degree in Electrical Information Engineering from the University of Science and Technology of China, Hefei, China, in 2009, and the M.S. degree in Electrical and Computer Engineering from University of Florida, Gainesville, Florida, USA, in 2011, and the Ph.D. degree in Computer Science from Stevens Institute of Technology, Hoboken, New Jersey, USA, in 2016. He is currently a lead research engineer with Here Technologies, Chicago, Illinois, USA. His research interests include computer vision, machine learning and multimedia signal processing. He is the author of more than 30 peer reviewed publications in international journals and conferences.

**Zhenxing Niu** (M’10) received the Ph.D. degree in Control Science and Engineering from Xidian University, Xi’an, China, in 2012. From 2013 to 2014, he was a visiting scholar with University of Texas at San Antonio, Texas, USA. He is a Researcher at Alibaba Group, Hangzhou, China. Before joining Alibaba Group, he is an Associate Professor of School of Electronic Engineering at Xidian University, Xi’an, China. His research interests include computer vision, machine learning, and their application in object discovery and localization. He served as PC member of CVPR, ICCV, and ACM Multimedia. He is a member of the IEEE.

**Nanning Zheng** (SM’94-F’06) graduated from the Department of Electrical Engineering of Xi’an Jiaotong University (XJTU), Xi’an, China, in 1975, received the M.E. degree in Information and Control Engineering from Xi’an Jiaotong University, Xi’an, China, in 1981, and the Ph. D. degree in electrical engineering from Keio University, Keio, Japan, in 1985. He is currently a Professor and the Director with the Institute of Artificial Intelligence and Robotics of Xi’an Jiaotong University, Xi’an, China. His research interests include computer vision, pattern recognition, computational intelligence, and hardware implementation of intelligent systems. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He became a member of the Chinese Academy Engineering in 1999. He is a fellow of the IEEE.

**Gang Hua** (M’03-SM’11-F’19) was enrolled in the Special Class for the Gifted Young of Xi’an Jiaotong University (XJTU), Xi’an, China, in 1994 and received the B.S. degree in Automatic Control Engineering from XJTU in 1999. He received the M.S. degree in Control Science and Engineering in 2002 from XJTU, and the Ph.D. degree in Electrical Engineering and Computer Science at Northwestern University, Evanston, Illinois, USA, in 2006. He is currently the Vice President and Chief Scientist of Wormpex AI Research. Before that, he served in various roles at Microsoft (2015-18) as the Science/Technical Adviser to the CVP of the Computer Vision Group, Director of Computer Vision Science Team in Redmond and Taipei ATL, and Principal Researcher/Research Manager at Microsoft Research. He was an Associate Professor at Stevens Institute of Technology (2011-15). During 2014-15, he took an on leave and worked on the Amazon-Go project. He was an Visiting Researcher (2011-14) and a Research Staff Member (2010-11) at IBM Research T. J. Watson Center, a Senior Researcher (2009-10) at Nokia Research Center Hollywood, and a Scientist (2006-09) at Microsoft Live labs Research. He is an associate editor of TIP, TCSVT, CVIU, IEEE Multimedia, TVCJ and MVA. He also served as the Lead Guest Editor on two special issues in TPAMI and IJCV, respectively. He is a program chair of CVPR’2019&2022. He is an area chair of CVPR’2015&2017, ICCV’2011&2017, ICIP’2012&2013&2016, ICASSP’2012&2013, and ACM MM 2011&2012&2015&2017. He is the author of more than 150 peer reviewed publications in prestigious international journals and conferences. He holds 19 US patents and has 15 more US patents pending. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award for his contribution on Unconstrained Face Recognition from Images and Videos, and a recipient of the 2013 Google Research Faculty Award. He is an IEEE Fellow, an IAPR Fellow, and an ACM Distinguished Scientist.