VIDEO OBJECT CO-SEGMENTATION FROM NOISY VIDEOS BY A MULTI-LEVEL HYPERGRAPH MODEL

Xin Lv[1], *Le Wang*[1]*, *Qilin Zhang*[2], *Zhenxing Niu*[3], *Nanning Zheng*[1], *and Gang Hua*[4]

[1]Xi'an Jiaotong University [2]HERE Technologies [3]Alibaba Group [4]Microsoft Research

ABSTRACT

Defined as simultaneously segmenting a set of related videos to identify the common objects, video co-segmentation has attracted the attention of researchers in recent years. Existing methods are primarily based on pair-wise relations between adjacent pixels/regions, which are susceptible to performance degradation from "empty" video frames (*e.g.*, due to transient/intermittent common objects). In this paper, a new multi-level hypergraph based method, termed the full Video object Co-Segmentation method (VCS), is proposed, which incorporates both a high-level semantics object model and a low-level appearance/motion/saliency object model 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 co-segmentation. Experiments on three datasets demonstrate the efficacy of the proposed VCS method.

Index Terms— Object co-segmentation, Object model, Hypergraph cut, Fully convolutional network

1. INTRODUCTION

Video object co-segmentation aims at segmenting a common category of objects from multiple videos, which is utilized in computer vision tasks such as object centric video summarization, spatio-temporal action localization, and content-based video retrieval. Unlike single video based methods (*e.g.*, [1]), the primary advantage of video co-segmentation is the availability of video semantics (*e.g.*, categorical video labels) shared among multiple videos. In recent years, most research efforts [2,3,4,5,6,7,8,9,10] are based on energy minimization by exploring pair-wise relations between two adjacent pixels/regions [2, 3, 5, 6, 8, 10]. They either leverage the low-level image features (*e.g.*, color and motion) [2, 4, 7, 9], mid-level contextual features [8, 10] or object proposals [3, 5, 6]. Inspired by the recent success of Convolutional Neural Network (CNN)-based methods [11, 12], fully convolutional network (FCN)-based [13,14] methods are also introduced for video object segmentation.

*Corresponding Author, *lewang@xjtu.edu.cn*

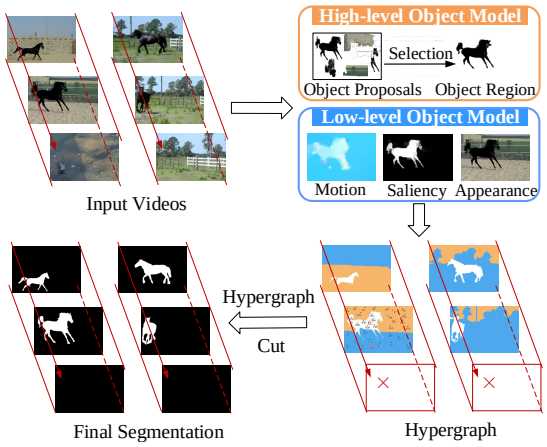

Fig. 1: Illustration of the proposed VCS method.

Despite the success of the aforementioned methods, most of them only exploit pair-wise correlations between pixels/regions, neglecting the higher order correlations between multiple ones. Besides, object proposal based methods ubiquitously utilize a single object proposal per video frame, which will consistently fail to localize common objects once the object proposal is inaccurate. Moreover, most existing methods require (almost) all video frames containing the common objects. With the percentage of "empty" frames without the common objects increasing, their performances degrade dramatically.

To resolve such limitations, we propose a multi-level hypergraph based full Video object Co-Segmentation method (VCS, as summarized in Fig. 1), which accounts for high order correlations, incorporates multiple object proposals per video frame, and robust to videos with large portions of "empty" frames. The hyperedge computation in our VCS method benefits from a hybrid object model (incorporating multi-modality information [15]), with one high-level model focused on video semantics, and a separate low-level model dedicated for video appearance/motion/saliency. Specifically, the high-level object model merges multiple object proposals to generate a more reliable object region per frame, thus producing more robust high-level features. The low-level features (appearance, motion and saliency) naturally complement the high-level ones, jointly contributing to a better video

representation. The hypergraph cut algorithm [16] is subsequently utilized to achieve the final object co-segmentation.

The contributions of the paper are as follows. 1) The VCS method is robust against large percentage of "empty" video frames (without the common objects). 2) The VCS method incorporates a multi-level hypergraph based hybrid object model, which accounts for both the high-level semantics and the low-level features. 3) A new, challenging video co-segmentation dataset is collected with both ground-truth categorical labels and pixel-wise foreground labels.

2. PRELIMINARIES OF HYPERGRAPH

Let $V = \{v_i\}$ denote the node set comprising a finite set of nodes and E denote the hyperedge set comprising a family of subsets of V, such that $\cup_{e\in E}= V$. $G = \{V, E, \omega\}$ is a weighted hypergraph with the node set V and hyperedge set E, and each hyperedge e is assigned a positive weight $\omega(e)$ [17]. A hyperedge e is incident with a node v when $v \in e$. For a node $v \in V$, its degree is $d(v) = \sum_{e\in E|v\in e} \omega(e)$, For a hyperedge $e \in E$, its degree is $\delta(e) = |e|$. 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 $\delta = \sum_{v\in V} h(v, e)$. 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.

3. PROBLEM FORMULATION

Given a set of videos $\mathbf{F} = \{F^n\}_{n=1}^N$, our goal is to find a binary co-segmentation 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 $B^n = \{B_t^n\}_{t=1}^T$. $B_t^n = \{b_{t,k}\}_{k=1}^K$ is the binary labels of frame f_t^n, where $b_{t,s}^n \in \{0, 1\}$ denotes the segmentation label of superpixel $s_{t,k}^n$ either belonging to the common object ($b_{t,k}^n = 1$) or the background ($b_{t,k}^n = 0$).

Video object co-segmentation can be cast into the hypergraph cut framework. Thus, the segmentation of the common object is 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 nodes of hyperedge e are included in S and S^c simultaneously, this hyperedge 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 lead 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 [18], we try to get a natural partition 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,

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

With the approximate algorithm of spectral analysis in [16], the hypergraph is partitioned and the final object co-segmentation result is obtained.

3.1. Hypergraph Construction

The superpixels generated by SLIC [19] are utilized as the nodes of the hypergraph. 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, where $A(p, q)$ represents the affinity between two nodes p and q, and is calculated by coupling a high-level and a low-level object model. D is the diagonal matrix with $D(p, p) = \sum_q A(p, q)$.

3.2. Hyperedge Computed with High-level Object Model

The high-level object model creates a more reliable object region for each video frame to guide the hyperedge computation. First, multiple object proposals are generated per frame [20], and a *video object score* $O(r_m)$ for each object proposal r_m is estimated by combining appearance, motion, and semantic cues,

$$O(r_m) = O_a(r_m) + O_m(r_m) + O_s(r_m), \tag{3}$$

where $O_a(r_m)$ denotes the appearance score of r_m with the objectness [20]. r_m will be assigned a high objectness score if it exhibits large distinction from surroundings and has a well-defined closed boundary. $O_m(r_m)$ 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 [21]. $O_s(r_m)$ is the semantic score of r_m. By using a FCN [22] pre-trained on ImageNet [23] as base network, we train a meta network on one video that randomly selected from each video category, and then fine-tuned the meta network on the first object frame of each video to obtain the test network. The segmentation results obtained by the fine-tuned network are utilized to compute the semantic score.

After sorting the object proposals by their video object scores, we merge the top M (empirically $M = 10$) ranked object proposals to obtain a candidate object region for each frame, and further refine it to obtain a reliable object region. Specifically, we use k-means to cluster the candidate object proposals of all video frames into two sets, *i.e.* a believable set Q_b and an unbelievable set Q_u. By treating the original object proposals that are used to merge the top M ranked candidate

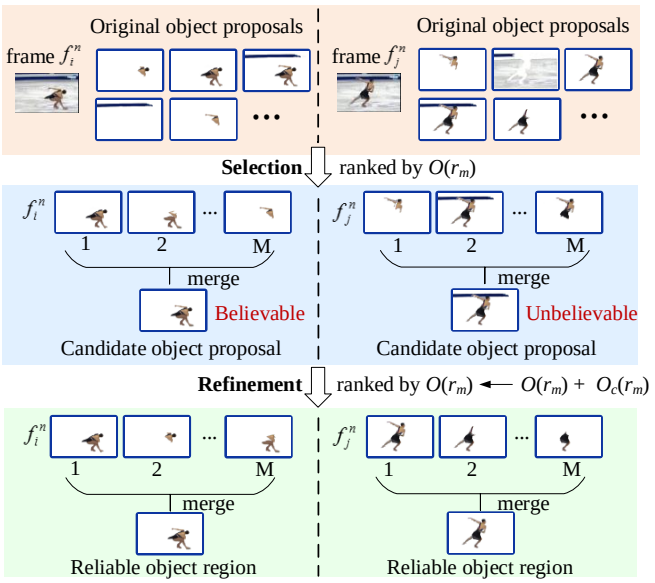

Fig. 2: Reliable object region generation.

object proposals in Q_b as positive samples, and the remaining ones as negative samples, we train a linear SVM classifier on the L2-normalized fc7 feature space (fully-connected 7th layer activation feature of Resnet [24]). We subsequently classify all the original proposals by this SVM classifier, and obtain a classification score $O_c(r_m)$ for each original object proposal r_m. Finally, we refine the *video object score* $O(r_m)$ of r_m by

$$O(r_m) \leftarrow O(r_m) + O_c(r_m), \tag{4}$$

and merge the new top M ranked object proposals into a reliable object region $\hat{r}$ for each frame. In this way, the object proposals that simultaneously contain the object and background can be filtered out. Fig. 2 illustrates the reliable object region generation procedure.

The reliable object region $\hat{r}$ of each frame is utilized to guide the hyperedge computation. 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,

$$A_h(p,q) = \begin{cases} \frac{1}{M}\sum_m O(\hat{r}_m), & \text{if } p,q \in \hat{r}/\hat{r}^c \\ \frac{1}{N_p+N_q}, & \text{otherwise} \end{cases}, \tag{5}$$

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 number of pixels in p and q, respectively.

3.3. Hyperedge Computed with Low-level Object Model

The low-level object model computes the hyperedge based on the appearance, motion and saliency cues. Assume pixels from the same superpixel have identical motion, the motion feature $P_m = (P_u, P_d)$ of each superpixel is computed by optical flow [25], which consists of the motion intensity $P_u = \frac{1}{N_s}\sum_j \omega_j u_j$ and direction $P_d = \frac{1}{N_s}\sum_j \omega_j d_j$. N_s is the number of pixels in the superpixel. ω_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 jth pixel, respectively.

Pixels from the same superpixel are highly likely to have similar color, color feature P_c are computed in Lab color space per superpixel as $P_c = \frac{1}{N_s}\sum_j c_j$, where c_j is the color value of the jth pixel. Additionally, the saliency detection introduced in [26] is used to generate the saliency map for each frame. Per-superpixel saliency value P_s is obtained by averaging all per-pixel saliency values. The affinity between two superpixels p and q based on the low-level object model is,

$$A_l(p,q) = e^{-\frac{||P_l(p)-P_l(q)||_2}{\sigma^l}}, \tag{6}$$

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

3.4. Hyperedge Weights Computation

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

$$\omega(e) = c \cdot \frac{\sum_{p,q\in e} A(p,q)}{\sum_{p\in e, q\notin e} A(p,q)}, \tag{7}$$

where c is a normalization constant to ensure $\sum_{e\in E}\omega(e) = 1$. Depending on a hyperedge being high-level or low-level, $A(p,q)$ is computed by Eq.(5) or Eq.(6), accordingly. Per Eq.(7), a large weight will be assigned to the hyperedge if the similarity of the superpixels is high, and vice versa.

4. EXPERIMENTS

All video object co-segmentation methods are evaluated on three datasets, *i.e.*, the VCoSeg dataset [7, 27, 28], XJTU-Stevens dataset [10] and a newly proposed Noisy-ViCoSeg dataset. Three individual video object segmentation methods (VOS [29], FOS [30], BVS [31]), and two multi-video object co-segmentation methods (MVC [32], VOC [3]) are used as competing algorithms. To better understand the contribution of hybrid object model in the proposed VCS method, an ablation study is also included. We removed the high-level semantics object model in VCS and denote it as (VCS-H). The ablated VCS-H is not evaluated on XJTU-Stevens or Noisy-ViCoSeg datasets, due to both datasets

contain "empty" frames that cannot be gracefully handled by the low-level object model alone.

Experiment 1. VCoSeg dataset [7, 27, 28] consists of 3 categories of 10 videos, and results are summarized in Tab 1 in terms of average IoU scores. On average, the proposed VCS method outperforms competing ones evidently, and the value of the hybrid object model is evident by comparing the performance gap between VCS and VCS-H.

Table 1: IoU scores on VCoSeg dataset.

Video	VOS	FOS	BVS	MVC	VOC	VCS-H	VCS
chachacha	55.3	61.0	71.7	56.3	53.2	67.2	**74.7**
ice skater	82.1	81.6	**83.5**	69.1	65.3	66.3	80.7
kite surfer	69.1	35.0	65.9	38.2	51.6	52.9	**69.3**
Avg.	68.8	59.2	73.7	54.5	56.7	62.1	**74.9**

Experiment 2. XJTU-Stevens dataset [10] consists of 10 categories of 101 publicly available internet videos, including 3.7% "empty" frames and many difficult frames (where the common object exhibits large variations in appearance, size, and shape). The average IoU scores and some sample results are presented in Table2 and Fig. 3, respectively. The performance advantage of the proposed VCS method is evident and the robustness of VCS is verified.

Table 2: IoU scores on XJTU-STEVENS dataset.

Video	VOS	FOS	BVS	MVC	VOC	VCS
airplane	35.6	**70.1**	35.1	57.6	58.6	66.7
balloon	80.2	71.2	78.7	86.8	86.8	**93.2**
bear	87.4	81.0	85.7	80.8	83.2	**88.0**
cat	50.1	53.4	71.2	75.2	**78.8**	73.7
eagle	55.4	75.9	59.4	72.2	78.4	**79.3**
ferrari	55.3	67.4	61.3	75.4	61.8	**82.9**
figure skating	64.9	36.1	48.7	61.7	65.3	**69.9**
horse	78.4	74.5	79.5	80.3	**85.1**	82.7
parachute	57.8	48.3	76.4	80.8	**83.4**	79.1
single diving	52.7	49.8	35.6	59.1	**69.5**	63.0
Avg.	61.8	62.8	63.2	73.0	75.1	**77.9**

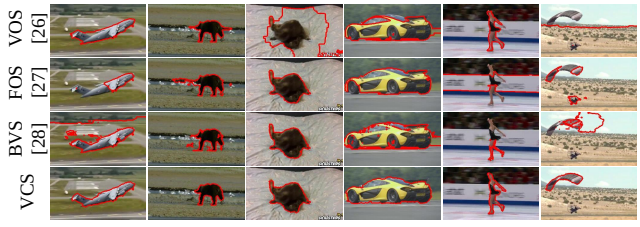

Fig. 3: Typical sample results on XJTU-STEVENS dataset.

Experiment 3. Noisy-ViCoSeg dataset is new collected and proposed in this paper, and it consists of 35 videos of 12 categories, where each video contains many "empty" frames. On average, 30.3% of video frames are without the common object. We manually assign each frame a per-frame label indicating whether it contains the common object. In addition, each video frame is also labeled with a per-pixel segmentation mask, indicating whether the common object is present at each pixel location. We compared our method with other five state-of-the-art methods. With the IoU scores in Table 3 and some example results in Fig. 4, the proposed VCS method is superior to the competing ones. One plausible explanation to such large performance gap is the designed robustness to "empty" frames in the proposed VCS method.

Table 3: IoU scores on the proposed Noisy-ViCoSeg dataset.

Video	VOS	FOS	BVS	MVC	VOC	VCS
airplane	46.1	**82.9**	54.7	51.8	34.1	73.6
F1	21.7	**70.3**	15.3	56.3	19.3	60.6
gymnastics	29.4	35.2	19.7	23.1	10.4	**62.3**
lion	61.6	**76.5**	50.2	74.7	70.9	49.0
ostrich	–	4.5	**61.6**	20.0	1.1	53.6
panda	63.9	**85.1**	51.7	30.6	45.7	72.9
parkour	60.2	28.1	39.8	60.6	38.9	**65.5**
rock	47.1	60.8	**64.1**	46.4	53.0	60.6
skateboarding	54.2	44.3	53.5	38.9	8.0	**59.6**
skiing	70.1	**84.7**	49.0	75.2	36.6	74.1
surfing	53.5	48.7	66.8	52.0	53.6	**73.0**
tiger	43.4	45.6	53.9	31.5	28.7	**54.6**
Avg.	45.9	55.6	48.4	46.8	33.4	**63.3**

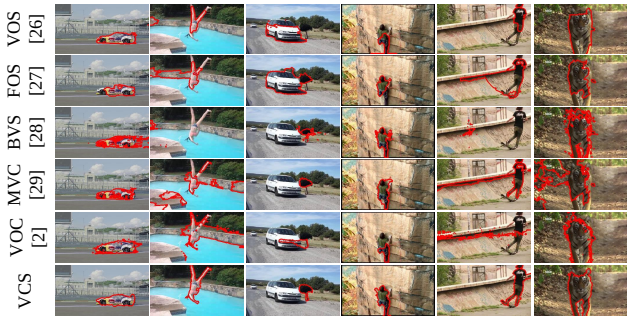

Fig. 4: Typical sample results on Noisy-ViCoSeg dataset.

5. CONCLUSION

In this paper, the full Video object Co-Segmentation method (VCS) is proposed to automatically co-segment the common objects in multiple noisy, cluttered videos. The proposed VCS method benefits from a multi-level hypergraph architecture incorporating both a high-level semantic model and a low-level object model, which contributes to its robustness against transient and intermittent objects. Empirical results on three video object co-segmentation datasets have verified the performance advantage of the proposed method.

6. ACKNOWLEDGE

This work was supported partly by National Key Research and Development Program of China Grant 2017YFA0700800,

NSFC Grants 61629301, 61773312, 61503296, and 91748208, and China Postdoctoral Science Foundation Grant 2017T100752.

7. REFERENCES

[1] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li, "Video-based sign language recognition without temporal segmentation," in *AAAI*, 2018.

[2] Ding-Jie Chen, Hwann-Tzong Chen, and Long-Wen Chang, "Video object cosegmentation," in *ACM MM*, 2012.

[3] Dong Zhang, Omar Javed, and Mubarak Shah, "Video object co-segmentation by regulated maximum weight cliques," in *ECCV*, 2014.

[4] Jiaming Guo, Zhuwen Li, Loong-Fah Cheong, and Steven Zhiying Zhou, "Video co-segmentation for meaningful action extraction," in *ICCV*, 2013.

[5] Huazhu Fu, Dong Xu, Bao Zhang, and Stephen Lin, "Object-based multiple foreground video co-segmentation," in *CVPR*, 2014.

[6] Zhongyu Lou and Theo Gevers, "Extracting primary objects by video co-segmentation," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2110–2117, 2014.

[7] Jose C Rubio, Joan Serrat, and Antonio López, "Video co-segmentation," in *ACCV*, 2012, pp. 13–24.

[8] Le Wang, Gang Hua, Rahul Sukthankar, Jianru Xue, and Nanning Zheng, "Video object discovery and co-segmentation with extremely weak supervision," in *ECCV*, 2014, pp. 640–655.

[9] Wenguan Wang, Jianbing Shen, Xuelong Li, and Fatih Porikli, "Robust video object cosegmentation," *IEEE T-IP*, vol. 24, no. 10, pp. 3137–3148, 2015.

[10] Le Wang, Gang Hua, Rahul Sukthankar, Jianru Xue, Zhenxing Niu, and Nanning Zheng, "Video object discovery and co-segmentation with extremely weak supervision," *IEEE T-PAMI*, vol. 39, no. 10, pp. 2074–2088, 2017.

[11] Jia Li, Anlin Zheng, Xiaowu Chen, and Bin Zhou, "Primary video object segmentation via complementary cnns and neighborhood reversible flow," in *ICCV*, 2017.

[12] Lingyan Ran, Yanning Zhang, Wei Wei, and Qilin Zhang, "A hyperspectral image classification framework with spatial pixel pair features," *Sensors*, vol. 17, no. 10, pp. 2421, 2017.

[13] Yi-Hsuan Tsai, Guangyu Zhong, and Ming-Hsuan Yang, "Semantic co-segmentation in videos," in *ECCV*, 2016.

[14] Armin Mustafa and Adrian Hilton, "Semantically coherent co-segmentation and reconstruction of dynamic scenes," *CVPR*, 2017.

[15] Qilin Zhang and Gang Hua, "Multi-view visual recognition of imperfect testing data," in *ACM MM*, 2015, pp. 561–570.

[16] Yuchi Huang, Qingshan Liu, and Dimitris Metaxas, "Video object segmentation by hypergraph cut," in *CVPR*, 2009.

[17] Denny Zhou, Jiayuan Huang, and Bernhard Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *NIPS*, 2007.

[18] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE T-PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[19] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, "Slic superpixels," Tech. Rep., 2010.

[20] Ian Endres and Derek Hoiem, "Category independent object proposals," in *ECCV*, 2010.

[21] Dong Zhang, Omar Javed, and Mubarak Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *CVPR*, 2013.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, et al., "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[25] Bruce D Lucas, Takeo Kanade, et al., "An iterative image registration technique with an application to stereo vision," 1981.

[26] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li, "Salient object detection: A benchmark," *IEEE T-IP*, vol. 24, no. 12, pp. 5706–5722, 2015.

[27] Fabrizio Tiburzi, Marcos Escudero, Jesús Bescós, and José M Martínez, "A ground truth for motion-based video-object segmentation," in *ICIP*, 2008.

[28] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR*, 2010.

[29] Yong Jae Lee, Jaechul Kim, and Kristen Grauman, "Key-segments for video object segmentation," in *ICCV*, 2011.

[30] Anestis Papazoglou and Vittorio Ferrari, "Fast object segmentation in unconstrained video," in *ICCV*, 2013.

[31] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung, "Bilateral space video segmentation," in *CVPR*, 2016.

[32] Wei-Chen Chiu and Mario Fritz, "Multi-class video co-segmentation with a generative multi-video model," in *CVPR*, 2013.