Semantic Object Segmentation in Tagged Videos via Detection

Yu Zhang, *Student Member, IEEE*, Xiaowu Chen, *Senior Member, IEEE*, Jia Li, *Senior Member, IEEE*, Chen Wang, Changqun Xia, and Jun Li

Abstract—Semantic object segmentation (SOS) is a challenging task in computer vision that aims to detect and segment all pixels of the objects within predefined semantic categories. In image-based SOS, many supervised models have been proposed and achieved impressive performances due to the rapid advances of well-annotated training images and machine learning theories. However, in video-based SOS it is often difficult to directly train a supervised model since most videos are weakly annotated by tags. To handle such tagged videos, this paper proposes a novel approach that adopts a segmentation-by-detection framework. In this framework, object detection and segment proposals are first generated using the models pre-trained on still images, which provide useful cues to roughly localize the semantic objects. Based on these proposals, we propose an efficient algorithm to initialize object tracks by solving a joint assignment problem. As such tracks provide rough spatiotemporal configurations of the semantic objects, a voting-based refinement algorithm is further proposed to improve their spatiotemporal consistency. Extensive experiments demonstrate that the proposed framework can robustly and effectively segment semantic objects in tagged videos, even when the image-based object detectors provide inaccurate proposals. On various public benchmarks, the proposed approach obtains substantial improvements over the state-of-the-arts.

Index Terms—Video segmentation, semantic object, detection-based segmentation, weakly supervised segmentation

1 Introduction

Although still a young area, semantic object segmentation (SOS) is among the most popular research topics in the computer vision community. Briefly speaking, SOS aims to jointly detect and segment all pixels of the objects from predefined semantic categories with a unified framework. In this framework, the detection and segmentation parts are strongly coupled and responsible for roughly locating the semantic objects and aligning them with the physical boundaries of images/videos, respectively. With SOS, many subsequent visual understanding tasks like action recognition [44] and scene modeling [12] may benefit from the precisely segmented semantic objects in images/videos.

In the past years, the booming of large-scale image datasets [17], [55] and machine learning theories [30] lead to a rapid advance in image-based SOS. In existing image-based SOS approaches, the detection part has been proven very helpful for revealing the high-level structures of the semantic objects and fully identifying their spatial extents. For example, various sophisticated object representations (e.g., deep neural networks [23], [38], [39], and-or graph [40], [67]) were learned from massive training images with manually segmented semantic objects. These representations are capable for roughly detecting the semantic objects with various poses, camera viewpoints and scales even in complex and cluttered background, which improves the performance of image-based SOS significantly.

Despite the remarkable success of image-based SOS, the same story, unfortunately, fails to repeat in video-based SOS. Unlike images, it is much more difficult to manually segment semantic objects in large-scale videos frame-by-frame. For example, the widely-used video-based SOS dataset [6] contains only 701 frames with pixel-level annotations. It is somewhat unclear whether the supervised SOS models directly trained on small datasets can well generalize to real-world scenarios. Moreover, some unsupervised works [15], [18], [34], [48] advocated to segment the salient and dominant objects in videos, while such foreground objects are not necessarily from the predefined semantic categories.

As it is difficult to manually generate pixel-level annotations in videos, some works [26], [42], [62] turned to the tagged videos that are vastly available on the internet. They adopted weakly supervised learning methods (e.g., multi-instance learning [26], negative mining [62] and label transfer [42]) to locate objects that appear frequently in videos with the same tags but rarely in videos with other tags. Usually, these models can capture the discriminative parts of semantic objects. However, they often have difficulties segmenting a semantic object as a whole since such

- *Y. Zhang, X. Chen, C. Wang, C. Xia, and J. Li are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. E-mail: zhangyulb@gmail.com, {chen, wangc, xiacq, junmuzi}@buaa.edu.cn.*
- *J. Li is also with the the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beijing 100191, China and the International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China. E-mail: jiali@buaa.edu.cn.*

Manuscript received 17 June 2016; revised 24 May 2017; accepted 4 July 2017. Date of publication 19 July 2017; date of current version 12 June 2018. (Corresponding author: Xiaowu Chen.) Recommended for acceptance by T. Brox. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2017.2727049

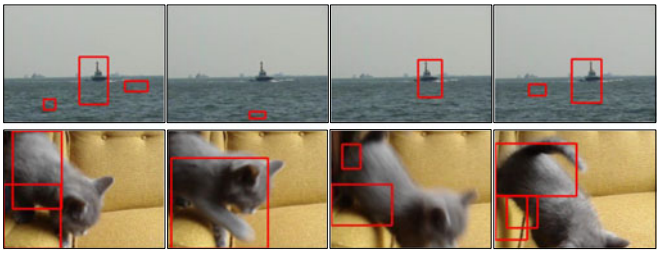


Fig. 1. Detection proposals generated on two videos from the Youtube-Objects dataset [51], by directly applying the DPM detector [19] pre-trained on Pascal VOC dataset [17]. The detections provide rough, noisy, but important cues for video object localization.

segmentation-only frameworks may fail to fully capture the spatial extent of an object. Moreover, these models require multiple videos as the input, which make them incapable of handling a single tagged video.

To segment semantic objects in a single tagged video, this paper proposes a novel approach that adopts a segmentation-by-detection framework. In this framework, we use the object detectors pre-trained on still images to generate a set of detection proposals. Due to the probable motion blur, compression effect, object occlusion and deformation in videos, such proposals are often noisy and only roughly reveal the probable locations of semantic objects (see Fig. 1). We propose an efficient algorithm to initialize a number of object tracks from such noisy proposals by solving a joint assignment problem. These tracks provide coarse configurations of semantic objects, which are fed into a voting-based algorithm for spatiotemporally consistent segmentation. Extensive evaluations on videos from Youtube-Objects [28], [62], SegTrack v2 [36] and FBMS-59 [47] datasets show that the proposed approach outperforms various state-of-the-art approaches.

The main contributions are summarized as follows:

1) We propose a novel segmentation-by-detection framework for video-based SOS, which significantly improves the performance of SOS on tagged videos.
2) We propose an efficient algorithm to initialize object tracks from the noisy detection proposals by solving a joint assignment problem.
3) We present a novel voting-based algorithm to refine the initial object tracks, which can produce spatiotemporally consistent object segmentations.

In the rest of this paper, we first conduct a brief review of the previous studies on SOS in Section 2. In Section 3, we present the technical details of the proposed segmentation-by-detection framework. Experimental results are shown in Section 4. At last, we conclude with a discussion in Section 5.

2 Related Work

Existing video object segmentation models can be categorized into supervised, unsupervised and weakly supervised groups. In the following, we mainly review approaches from these groups, and also some tightly correlated approaches that adopt a segmentation-by-detection framework.

2.1 Supervised Approaches

Most supervised SOS approaches have two steps. In the first step, each pixel is fed into a classifier that generates category-specific confidence scores. Based on these scores, an inference step is further taken to obtain final labels. While the first step was usually implemented using off-the-shelf classifiers (e.g., [58]), the second step receives more attention, especially with an interest on incorporating various object-level and scene-level cues. For example, Taylor et al. [63] explored occlusion cues in a convex framework to jointly segment semantic objects and estimate their depths. Floros et al. [20] and Kundu et al. [32] lifted semantic video segmentation into 3D space to impose geometric consistency. Liu et al. [41] proposed to jointly reason about pixel labels and object tracks, and incorporated depth ordering in an augmented CRF. Another direction is to accelerate the inference speed on long videos, such as dynamic graph reduction [10] and coarse-to-fine reasoning [27].

To sum up, supervised models for video-based SOS mainly take efforts towards sophisticated and efficient inference techniques, which may be caused by the fact that there lacks a large-scale and well-annotated dataset for video-based SOS. Actually, insufficient training data is the main obstacle that prevent the rapid development of video-based SOS models. To address this problem, Xie et al. [70] proposed to transfer the annotations in reconstructed 3D street scenes to 2D image plane so that a large amount of accurate pixel-level video annotations can be rapidly generated. However, this strategy is limited to work with several types of scenes and objects. Gathering pixel labels for videos in more wild setting, currently, is still largely unsolved.

2.2 Unsupervised Approaches

Unsupervised approaches mainly focus on segmenting the salient foreground objects in videos. For instance, a series of approaches [15], [34], [45] proposed to localize the underlying objects by objectness proposals [3], [8], which is widely considered as a branch of visual saliency [37]. After extracting large amounts of proposals from various frames, they formulated object segmentation in video as selecting a subset of proposals with coherent appearance and motion. This objective was addressed with various graph theories (e.g., spectral clustering [34], maximum weighted clique [45] and directed acyclic graph [15]). Beyond proposal extraction, several works [18], [48] estimated coarse foreground maps on sparse spatiotemporal locations through bottom-up cues (e.g., image fixation [48] and motion boundary [18]), and refine them with long-range propagation. Other solutions include motion analysis [47], foreground/background modeling [56], [71], [74], and low-rank region grouping [35].

Despite the remarkable success of unsupervised approaches, a key limitation is that there still exists a large gap between foreground objects and semantic objects. However, for tagged videos the foreground objects do have tight correlations with the video tags, and such inherent correlation deserves being further explored in video-based SOS.

2.3 Weakly Supervised Approaches

Nowadays, videos on the internet are widely accompanied with tags, which often indicate the presence/absence of semantic categories in the videos. To explore these tagged videos, recent studies [42], [59], [62] adopted various weakly supervised learning methods for video-based SOS. The pioneer work [26] adopted multi-instance learning to learn object appearance models from video-level tags. Later,

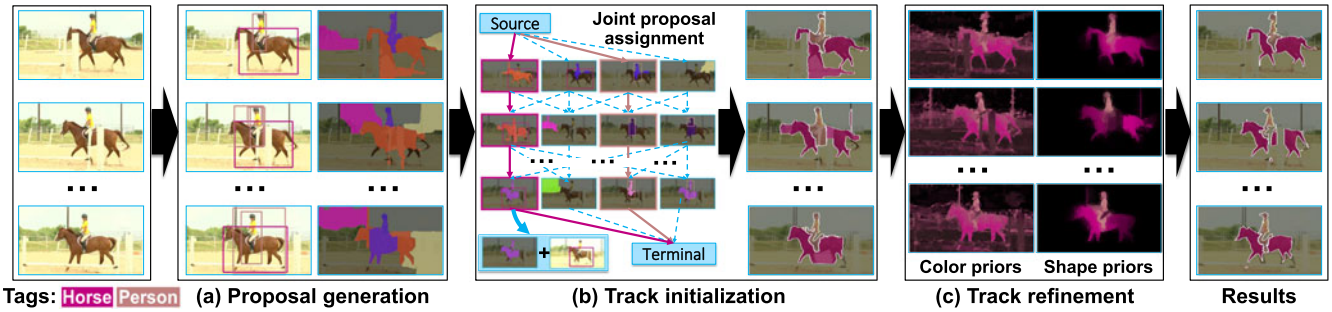


Fig. 2. The system framework of our approach. It consists of three major stages, including 1) *proposal generation:* generating a set of detection and segment proposals for each frame; 2) *track initialization:* initializing object tracks from noisy proposals by solving a joint assignment problem; 3) *track refinement:* refining the spatiotemporal consistency of object tracks by using shape priors and color cues.

Tang et al. [62] proposed to rank the spatiotemporal segments in tagged videos via their dissimilarities to massive segments sampled in irrelevant videos. As the former works only focus on foreground/background segmentation, Liu et al. [42] proposed a multi-class approach that can handle videos with multiple tags, in which category-aware feature distance was learned so as to build nearest neighbor classifiers.

Video object co-segmentation [11], [13], [16], [21], [54], [66] tightly correlates with weakly supervised SOS, with the small difference that there are no background videos, i.e., the same object categories are assumed to appear in all videos. Thus, co-segmentation mainly relies on inter-video similarities to segment the common objects. Following this idea, prior works [12], [54] initialized foreground/background segmentation independently for each video, then iteratively refined them to improve the appearance and motion similarity of the segmented regions in different videos. In more recent studies [16], [21] proposed to extract many object proposals and select the ones with coherent appearance and motion in different videos. Beyond focusing on inference, Wang et al. [66] proposed to use spatiotemporal auto-context feature and multi-instance boosting to represent and learn the object appearance. Chiu et al. [13] assumed that the distribution of object classes in different videos follow a known probabilistic prior, which enables their approach to handle unknown number of object classes and objects.

Weakly supervised SOS is capable of handling tagged videos but still have two main limitations. First, object parts often match more consistently across different videos than the whole objects, especially for categories with large intra-class variance. In this case, it may be difficult to segment semantic objects as a whole. Second, most weakly supervised approaches require multiple videos as input to discover the common information hidden behind tags, which restricts the usage of such models in real-world applications.

2.4 Segmentation-by-Detection Approaches

To address the limitations above, our idea is to borrow the power of object detectors pre-trained on external images, while not introducing labeling efforts in video domain. The concept of segmentation-by-detection is not new, with many such efforts taken for image-based SOS. For example, [69], [72] proposed to derive shape priors in the detected bounding boxes to guide segmentation through voting [69] or layer ordering [72]. Without other cues, however, they lack the ability to prune false positive detections. In contrary, several approaches [33], [64] combined the detections with region-level or scene-level cues to generate coherent results. Beyond treating detection and segmentation separately, recent works [14], [25] unified them into a single stage considering the consistency between detected and segmented regions. These approaches are robust to outlier detections, but rely on strongly annotated training images.

Object segmentation-by-detection is relatively new for videos. However, it is promising as videos provide spatiotemporal cues that can effectively eliminate outlier detections with minimal supervision. Our previous work [73] makes an attempt towards this direction and obtains state-of-the-art results on Youtube-Objects dataset [28], [51]. This extended version improves it in terms of both performance and speed, see Section 3.4 for detailed comparisons. More recently, Seguin and Laptev et al. [57] proposed a detection-based approach to segment multiple object instances in a video. To achieve this, they assumed that reliable detections are given, and demonstrated the effectiveness of their approach on segmenting humans. In contrast, our approach is designed to work with inaccurate detections and general object categories in more wild setting.

3 Video Object Segmentation via Detection

3.1 Detection/Segment Proposal Extraction

Given a video with T frames, the first stage of our approach extracts detection and segment proposals for each frame, as shown in Fig. 2. Taking the tth frame as example, this stage works as follows:

Detection Proposals. We apply image-based object detectors (e.g., [19], [22]) to generate a set of object detections $\mathbb{D}_t^+$. For each detection proposal $\mathcal{D} \in \mathbb{D}_t^+$, its normalized detection response is represented with $r(\mathcal{D})$. Normalization is straightforward when the video is tagged with a single category. For multiple categories, one can apply Platt scaling [50] to calibrate detectors of different categories so as to make their outputs comparable.

It would be too ideal to assume that the target objects are always hit by some detections in $\mathbb{D}_t^+$. To handle detection missing, a "dummy" detection $\mathcal{D}_\phi$ is appended, i.e., $\mathbb{D}_t = \mathbb{D}_t^+ \cup \{\mathcal{D}_\phi\}$. $\mathcal{D}_\phi$ does not have specific spatial position or detection response. As shown in the next section, it is just introduced for algorithmic convenience.

Segment Proposals. We apply the MCG proposals [3] for each frame. To incorporate motion cues, we compute the point-wise maximum between the edge response maps of the original image and optical flow gradients, and feed the

combined map into the rest stages of [3]. After extracting object proposals, we eliminate the ones that are either too small (i.e., with less than 200 pixels) or too large (i.e., cover more than 80 percent image area). After that, we retain the top 500 proposals ranked by [3] and discard the rest. We further follow [15] to re-score these proposals considering both their objectness and motion saliency. The top 300 proposals after re-ranking are picked up, which form the final proposal set $\mathbb{S}_t$. For each segment, we compute the HSV colors and texture features [46] at each pixel inside, accumulate and quantize them into 96-bin and 64-bin histograms, respectively. We denote the concatenated features as $\mathbf{f}(\mathcal{S})$, and the "objectness" score given by the MCG algorithm as $o(\mathcal{S})$.

For convenience, we give here the Intersection-over-Union (IoU) score that measures the overlap between two regions (detection or segment) $\mathcal{R}_1$ and $\mathcal{R}_2$, computed as $\pi(\mathcal{R}_1, \mathcal{R}_2) = \frac{|\mathcal{R}_1 \cap \mathcal{R}_2|}{|\mathcal{R}_1 \cup \mathcal{R}_2|}$. Here $|\cdot|$ denotes the number of pixels in the input region, and operators $\cap$ and $\cup$ generate the intersected and union regions of $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. If at least one of $\mathcal{R}_1$ and $\mathcal{R}_2$ equals $\mathcal{D}_\phi$, i.e., the missed underlying ground-truth detection, we simply set $\pi(\mathcal{R}_1, \mathcal{R}_2) = 1$.

3.2 Track Initialization

In the *track initialization*, we aim to link the proposals to one or several object tracks that best cover the semantic objects. To this end, an immediate way is to apply the tracking-by-detection models (e.g., [9], [49]) to link detections into tracks. However, since the detections of the same object are often missing and changing locations/scales across different frames, this strategy can only generate fragmented tracks in our preliminary study. To address this issue, we notice that segment proposals often evolve more smoothly along time, which could be helpful for robust tracking. Therefore, we propose to jointly assign the detection and segment proposals to tracks instead of treating them separately, which thus lead to more consistent and complete tracks.

3.2.1 Joint Assignment Problem

Assuming that there are K object tracks in a video (the value of K is unknown), we define the following variables

$$\begin{aligned}\mathbb{A} &= \{a_{\mathcal{D}}^k | k \in \{1, 2, \ldots, K\}, t \in \{1, 2, \ldots, T\}, \mathcal{D} \in \mathbb{D}_t\},\\ \mathbb{B} &= \{b_{\mathcal{S}}^k | k \in \{1, 2, \ldots, K\}, t \in \{1, 2, \ldots, T\}, \mathcal{S} \in \mathbb{S}_t\},\end{aligned} \tag{1}$$

where $a_{\mathcal{D}}^k$ ($b_{\mathcal{S}}^k$) equals to 1 if $\mathcal{D}$ ($\mathcal{S}$) is assigned to the kth track and 0 otherwise. To optimize $\mathbb{A}$ and $\mathbb{B}$ we propose to solve the following minimization problem:

$$\begin{aligned}&\min_{\mathbb{A},\mathbb{B},K} \mathcal{L}(\mathbb{A}, \mathbb{B}) + \lambda_t \Omega_t(\mathbb{A}, \mathbb{B}) + \lambda_s \Omega_s(\mathbb{B}), \text{ s.t.},\\ &1)\ \forall t, \mathcal{D} \in \mathbb{D}_t, \mathcal{S} \in \mathbb{S}_t, \sum_k a_{\mathcal{D}}^k \le 1, \sum_k b_{\mathcal{S}}^k \le 1,\\ &2)\ \forall k, t, \sum_{\mathcal{D} \in \mathbb{D}_t} a_{\mathcal{D}}^k = \sum_{\mathcal{S} \in \mathbb{S}_t} b_{\mathcal{S}}^k \le 1,\\ &3)\ \sum_t \sum_{\mathcal{D} \in \mathbb{D}_t} a_{\mathcal{D}}^k = \sum_t \sum_{\mathcal{S} \in \mathbb{S}_t} b_{\mathcal{S}}^k \ge 1, \text{ and } 4)\ \forall k, t_0 < t < t_1,\\ &\left(\sum_{\mathcal{D} \in \mathbb{D}_{t_0}} a_{\mathcal{D}}^k\right)\left(1 - \sum_{\mathcal{D} \in \mathbb{D}_t} a_{\mathcal{D}}^k\right)\left(\sum_{\mathcal{D} \in \mathbb{D}_{t_1}} a_{\mathcal{D}}^k\right) = 0,\end{aligned} \tag{2}$$

where $\mathcal{L}$ denotes the quality of selected proposals, Ω_t and Ω_s penalize the temporal inconsistency and spatial conflict of tracks, while their influences are controlled by λ_t and λ_s. The four sets of constraints, from top to bottom, indicate that 1) a proposal can be assigned to at most one track; 2) a track can select at most one detection/segment proposal from each frame; 3) tracks are non-empty and 4) consecutive along time. To show that the last constraint implies consecutiveness, note that $\sum_{\mathcal{D} \in \mathbb{D}_t} a_{\mathcal{D}}^k$ takes 1 if the kth track passes the tth frame, and 0 otherwise. Thus, it is impossible for a track to be consecutive if there exists $t_0 < t < t_1$ so that it appears at t_0th and t_1th frame but breaks at the tth frame.

The proposal quality term $\mathcal{L}$ is defined as

$$\begin{aligned}\mathcal{L}(\mathbb{A}, \mathbb{B}) &= -\sum_{t,k} \sum_{\mathcal{D} \in \mathbb{D}_t} \sum_{\mathcal{S} \in \mathbb{S}_t} \xi(\mathcal{D}, \mathcal{S}) a_{\mathcal{D}}^k b_{\mathcal{S}}^k, \text{ where}\\ \xi(\mathcal{D}, \mathcal{S}) &= \begin{cases} \log \frac{r(\mathcal{D})^{\alpha_1} o(\mathcal{S})^{\alpha_2} \pi(\mathcal{D}, \mathcal{S})^{\alpha_3}}{1 - r(\mathcal{D})^{\alpha_1} o(\mathcal{S})^{\alpha_2} \pi(\mathcal{D}, \mathcal{S})^{\alpha_3}}, & \text{if } \mathcal{D} \neq \mathcal{D}_\phi \\ 0, & \text{otherwise,} \end{cases}\end{aligned} \tag{3}$$

which are designed with two considerations. First, the selected proposals are expected to have high detection objectness scores. Second, for each track, the selected segment and detection on a frame should be spatially close. The positive weights α_1, α_2 and α_3 are empirically set as $\alpha_1 = \alpha_3 = 0.25$, and $\alpha_2 = 0.5$. For dummy detection proposal, we set the outcome of selecting it to zero as it should not contribute to the overall objective.

The penalty Ω_t encourages smoothly selecting segment proposals on adjacent frames, which is computed as

$$\Omega_t(\mathbb{A}, \mathbb{B}) = \sum_{t,k} \sum_{\mathcal{S} \in \mathbb{S}_t} \sum_{\mathcal{S}_0 \in \mathbb{S}_{t+1}} \frac{\chi^2(\mathbf{f}(\mathcal{S}), \mathbf{f}(\mathcal{S}_0))}{1 + \pi(\bar{\mathcal{S}}, \mathcal{S}_0)} b_{\mathcal{S}}^k b_{\mathcal{S}_0}^k. \tag{4}$$

The smoothness term $\eta(\mathcal{S}, \mathcal{S}_0)$ considers both appearance and shape similarities. The former is calculated by the χ^2 distance between the features of $\mathcal{S}$ and $\mathcal{S}_0$. To measure shape similarity, we warp a segment $\mathcal{S}$ to the next frame through optical flows, then calculate the overlap between $\mathcal{S}_0$ and the warped region $\bar{\mathcal{S}}$, i.e., $\pi(\bar{\mathcal{S}}, \mathcal{S}_0)$.

The penalty term Ω_s prevents different tracks from selecting segments with large spatial overlaps:

$$\Omega_s(\mathbb{B}) = \frac{1}{2} \sum_t \sum_{k \neq k_0} \sum_{\mathcal{S}, \mathcal{S}_0 \in \mathbb{S}_t} b_{\mathcal{S}}^k b_{\mathcal{S}_0}^{k_0} \pi(\mathcal{S}, \mathcal{S}_0). \tag{5}$$

The problem obtained by incorporating (3)$\sim$(5) into (2), is combinatorial and higher-order, which is very challenging to solve. As a result, we reformulate this problem as a special form of min-cost flow [2], which can be solved efficiently.

3.2.2 Reformulation as Quadratic Min-Cost flow

We re-express the objective functions and constraints of (2) using the following two sets of variables[1]:

$$\delta_x(\mathcal{D}, \mathcal{S}) = \sum_k a_{\mathcal{D}}^k b_{\mathcal{S}}^k, \delta_y(\mathcal{D}, \mathcal{D}_0, \mathcal{S}, \mathcal{S}_0) = \sum_k a_{\mathcal{D}}^k a_{\mathcal{D}_0}^k b_{\mathcal{S}}^k b_{\mathcal{S}_0}^k. \tag{6}$$

1. Due to space limit, we briefly describe the reformulation here while leave the strict analysis in supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2017.2727049.

Note that δ_x and δ_y are also binary given the first constraint in (2). This substitution marginalizes out k, i.e., the index of tracks. $\delta_x(\mathcal{D},\mathcal{S})=1$ means that the proposal pair $(\mathcal{D},\mathcal{S})$ is selected by a track. Similarly, $\delta_y(\mathcal{D},\mathcal{D}_0,\mathcal{S},\mathcal{S}_0)$ indicates the status of jointly selecting the pairs $(\mathcal{D},\mathcal{S})$ and $(\mathcal{D}_0,\mathcal{S}_0)$. In other words, we can treat δ_x and δ_y as the activation variables of nodes and edges in a graph, in which a node is a pair of a detection and segment proposal in the same frame, and an edge links such pairs for adjacent frames.

Given this graph, the solution of track initialization can be characterized from two aspects. First, there are K tracks traversing from the virtual source node to sink node along the nodes and edges defined above. Second, these tracks do not conflict in selecting segment or detection proposals. From the perspective of network flow theory [2], the first requirement can be well modeled by the *flow conservation* and *flow requirement* constraints of unit-capacity flows [49]. The second requirement can be reformulated as quadratic constraints: $\delta_x(\mathcal{D},\mathcal{S})\delta_x(\mathcal{D}_0,\mathcal{S}_0)=0$ if $\mathcal{D}=\mathcal{D}_0$ or $\mathcal{S}=\mathcal{S}_0$ and written in compact form $(\boldsymbol{\delta}_x)^{\mathrm{T}}\mathbf{M}'\boldsymbol{\delta}_x=0$, where an entry of $\mathbf{M}'$ takes 1 if a quadratic constraint holds for a pair of nodes, and 0 otherwise.

Let $\boldsymbol{\delta}=\left[\boldsymbol{\delta}_x,\boldsymbol{\delta}_y\right]^{\mathrm{T}}$ be a column vector concatenating the new optimization variables. By substituting it into (2) and rewriting the constraints, we have

$$\begin{aligned}&\min_{\boldsymbol{\delta}\in\{0,1\}^{|\boldsymbol{\delta}|}}\frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}\underbrace{\begin{bmatrix}\boldsymbol{\Pi}' & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}}_{\boldsymbol{\Pi}}\boldsymbol{\delta}-\underbrace{\begin{bmatrix}\boldsymbol{\xi}\\ \boldsymbol{\eta}\end{bmatrix}}_{\mathbf{c}}\boldsymbol{\delta},\\ &\text{s.t. }\boldsymbol{\delta}\in\mathbb{F}(K)\ \wedge\ \boldsymbol{\delta}^{\mathrm{T}}\underbrace{\begin{bmatrix}\mathbf{M}' & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}}_{\mathbf{M}}\boldsymbol{\delta}=0.\end{aligned}\tag{7}$$

The column vectors $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ concatenate the costs defined in (3) and (4), and the matrix $\boldsymbol{\Pi}'$ collects the quadratic costs in (5). $\mathbb{F}(K)$ denote the feasible region of flow conservation and requirement constraints.

What distinguish this problem with standard min-cost flows are the quadratic costs and constraints. In [9], an effective solution is proposed for quadratic flows with moderate size. As the size of our problem is much larger, we propose a novel algorithm to efficiently solve (7).

3.2.3 The Optimization Algorithm

The difficulties for optimizing (7) lie in two aspects: 1) how to determine the number of tracks K and 2) how to obtain the configuration of each track. To address the first problem efficiently, our algorithm works in greedy manner. At first, it establishes a single flow using existing min-cost flow solver. In each of the remaining steps, it tries to establish one more flow. It stops when the objective value starts to increase.

The second problem is addressed in each step of increasing the number of flows. Assume that $K-1$ flows are obtained, whose configuration variable is written as $\boldsymbol{\delta}^{(K-1)}$. To obtain $\boldsymbol{\delta}^{(K)}$, we propose an algorithm that finds a local minimum of (7) using $\boldsymbol{\delta}^{(K-1)}$ as initialization, which can jointly optimize all the obtained flows.

To solve (7), we first relax it into continuous domain and put the quadratic constraints into the objective function:

$$\min_{\boldsymbol{\delta}\in[0,1]^{|\boldsymbol{\delta}|}}\frac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}(\boldsymbol{\Pi}+\mu\mathbf{M})\boldsymbol{\delta}-\mathbf{c}^{\mathrm{T}}\boldsymbol{\delta},\ \text{s.t. }\boldsymbol{\delta}\in\mathbb{F}(K).\tag{8}$$

The relaxed problem is quadratic, which is still inefficient to solve. Inspired by recent advances on finding maximum independent set [68], we instead iteratively solve its first-order approximations. Starting from $\boldsymbol{\delta}^{(K-1)}$, this algorithm visits a sequence of solutions. Given the current solution $\boldsymbol{\delta}'$, it seeks for the next one around the first-order neighborhood of $\boldsymbol{\delta}'$ through Taylor expansion:

$$\min_{\boldsymbol{\delta}\in[0,1]^{|\boldsymbol{\delta}|}}\boldsymbol{\delta}^{\mathrm{T}}[(\boldsymbol{\Pi}+\mu\mathbf{M})\boldsymbol{\delta}'-\mathbf{c}],\ \text{s.t.}\boldsymbol{\delta}\in\mathbb{F}(K).\tag{9}$$

This approximation is a tractable linear sub-problem w.r.t. $\boldsymbol{\delta}$. Moreover, since that the constraint matrix $\mathbb{F}(K)$ of min-cost flows is *totally unimodular* [2], the sub-problem actually leads to discrete solution although solved in continuous domain. After solving (9), we check whether the obtained solution $\widetilde{\boldsymbol{\delta}}$ decreases the value of (8). If it does, we accept it as the new local minimum. Otherwise, a local minimum must exist along the solution path linearly interpolated by $\boldsymbol{\delta}'$ and $\widetilde{\boldsymbol{\delta}}$ given the smoothness of the objective function. Line search is thus performed in this case, in which the new solution is expressed as $\boldsymbol{\delta}'+\rho(\widetilde{\boldsymbol{\delta}}-\boldsymbol{\delta}')$, $\rho\in[0,1]$. It is shown in [68] that the optimal step length ρ^* is calculated in closed form

$$\rho^*=\min\left(\max\left(\frac{(\mathbf{c}-(\boldsymbol{\Pi}+\mu\mathbf{M})\boldsymbol{\delta}')^{\mathrm{T}}(\widetilde{\boldsymbol{\delta}}-\boldsymbol{\delta}')}{(\widetilde{\boldsymbol{\delta}}-\boldsymbol{\delta}')^{\mathrm{T}}\boldsymbol{\Pi}(\widetilde{\boldsymbol{\delta}}-\boldsymbol{\delta}')},0\right),1\right).\tag{10}$$

The interpolated solution is not necessarily binary, but can be used to start the next iteration. In this way, the objective is guaranteed to decrease monotonically. To stop the iterations, we compute the objective values in (8) of the last two solutions and check whether they are sufficiently close (i.e., their absolute difference is within 10^{-8}). After convergence, the last binary solution is taken as $\boldsymbol{\delta}^{(K)}$. Note that the obtained local minimum in this manner satisfies the relaxed quadratic constraints as long as μ is sufficiently large (see proof in supplementary material, available online).

We summarize the whole process in Algorithm 1. Since that the number of semantic objects in a video is usually small, the algorithm typically terminates in less than 5 runs.

Algorithm 1. The Track Initialization Algorithm.

Input: Problem structure $\mathbf{c}$, $\boldsymbol{\Pi}$ and $\mathbf{M}$, and parameter μ.
Output: The local optimum $\boldsymbol{\delta}^*$ and the flow number K^*.

1: Initialize a flow by solving
   $\boldsymbol{\delta}^{(0)}=\min_{\boldsymbol{\delta}}-\mathbf{c}^{\mathrm{T}}\boldsymbol{\delta},\ \text{s.t. }\boldsymbol{\delta}\in\{0,1\}^{|\boldsymbol{\delta}|}\cap\mathbb{F}(1)$.
2: Let $K=0$, $\hat{\boldsymbol{\delta}}=\boldsymbol{\delta}^{(0)}$;
3: **Do**
4: Let $K=K+1$;
5: Using $\boldsymbol{\delta}^{(K)}$ as initialization, Solve (8) to obtain $\boldsymbol{\delta}^{(K+1)}$.
6: **While** $\boldsymbol{\delta}^{(K+1)}$ decreases the objective defined in (8)
7: **return** $\boldsymbol{\delta}^*=\boldsymbol{\delta}^{(K)}$, $K^*=K$.

Implementation Details. Note that the size of the problem is large since it operates on proposal pairs. To reduce the size, we only consider the nodes where the IoU overlap between the segment and detection bounding boxes exceeds

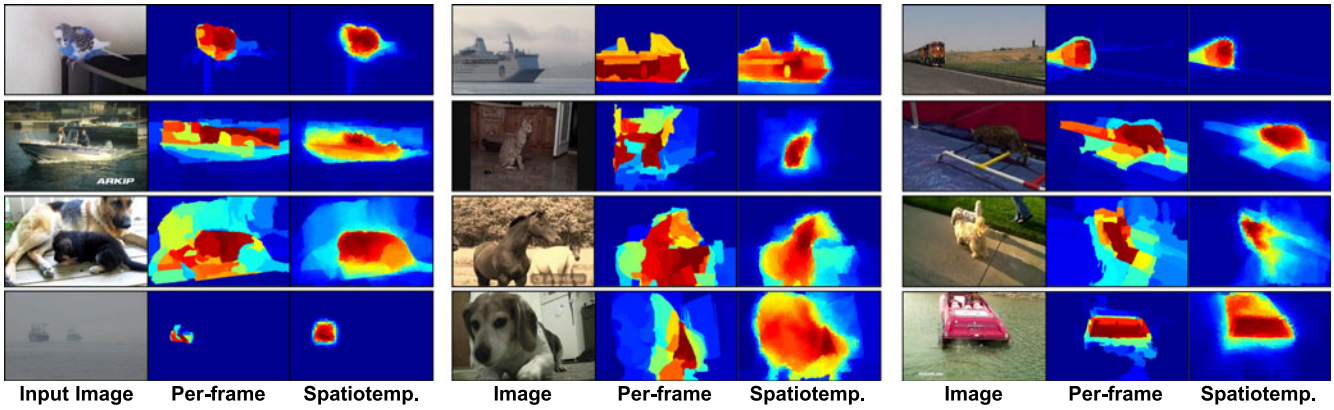


Fig. 3. Visual comparison between the per-frame voting and spatiotemporal voting. Row 1: trivial cases with clean background, where high-quality estimations are obtained by both voting schemes. Row $2 \sim 3$: when background clutter and object occlusion present, only the spatiotemporal voting gives meaningful results. Row 4: spatiotemporal voting fixes per-frame failures and generates more complete shape priors.

30 percent. Temporal edges whose segments have less than 30 percent area overlapped after optical flow mapping are also pruned. In this manner, the number of optimization variables can be restricted into several millions even for long videos with hundreds of frames. Such a scale is tractable for commercial solvers such as Gurobi [24]. After solving (7), the tracks that span less than 3 frames are considered unreliable and removed. The remaining tracks may still cover a small portion of the video, thus we extend them temporally through tracking segment proposals. We apply the efficient greedy tracking algorithm [16], and stop it when no region in the successive frame has more than 50 percent area mapped to the last tracked segment. We perform tracking independently for each track, in both forward and backward directions.

3.3 Track Refinement

Although the initial tracks can locate the objects well, they often exhibit inconsistent appearance in different frames. It is due to the limitation that segment proposal extraction is performed individually for each frame, thus is sensitive to various levels of occlusion, background clutter and motion blur on different frames. To address this problem, we propose a novel voting-based algorithm to improve the spatiotemporal consistency of the initial segmentations.

The algorithm starts from the observation that the ranking scores given by segment proposal extractors are often designed to match the spatial overlaps with the underlying ground-truth object [3], [8]. Thus, these scores are natural estimators of object shapes. To implement this idea, one can simply aggregate these scores onto pixels. To this end, we first retain the segment proposals with more than 50 percent area overlapped by the initial segment in each frame. Let the ith segment on the tth frame be denoted with $\mathcal{S}_{t,i}$. The likelihood of the ith pixel (superpixel) on the tth frame being part of the object, denoted with $C_t(\mathcal{P}_t^i)$, is calculated as

$$C_t(\mathcal{P}_t^i) = \sum_j \mathbb{1}(\mathcal{P}_t^i \in \mathcal{S}_t^j) o(\mathcal{S}_t^j), \tag{11}$$

where $\mathbb{1}(\cdot)$ is the characteristic function which takes 1 if the condition holds, or 0 otherwise. A similar formula was introduced in [69] without considering the objectness scores, and obtained promising performance on the Pascal VOC dataset. Representative object shape priors estimated by 11 are illustrated in the top row of Fig. 3. However, in case of severe occlusion or background clutter as shown by the 2nd and 3rd rows in Fig. 3, the estimations are often unreliable since the background distractors and "occluder" objects are often more salient than the foreground objects. To address this issue, notice that the foreground regions often appear more consistently than the background ones when looking across the whole spatiotemporal region defined by the initial track. To instantiate this idea, we allow a segment proposal to give votes to pixels temporally far away from it. Such extension modifies (11) by

$$C(\mathcal{P}_t^i) = \sum_{t_0} \sum_j \mathbb{1}\left(\mathcal{P}_t^i \in \mathscr{T}_{t_0,t}(\mathcal{S}_{t_0}^j)\right) o(\mathcal{S}_t^j), \tag{12}$$

in which a warping function $\mathscr{T}_{t_0,t}$ is introduced to propagate the segment proposals on the t_0th frame to tth frame. Note that the voting process in (12) can be performed by first computing the per-frame estimates using (11), and then propagating them to every other frame through optical flows. The final estimates for each frame are computed as the summation of the propagated confidence maps from all other frames and its own, and normalized.

To speed up and compensate boundary information, we operate on SLIC superpixels [1]. The propagation from the t_0th to the tth frame ($t > t_0$) is recursive, computed by

$$C_{t_0,t}(\mathcal{P}_t^i) = \frac{\sum_j \omega(\mathcal{P}_{t-1}^j, \mathcal{P}_t^i) \Phi_{t_0,t-1}(\mathcal{P}_{t-1}^j) C_{t_0,t-1}(\mathcal{P}_{t-1}^j)}{\sum_j \omega(\mathcal{P}_{t-1}^j, \mathcal{P}_t^i) \Phi_{t_0,t-1}(\mathcal{P}_{t-1}^j)}. \tag{13}$$

In (13), $C_{t_0,t}(\mathcal{P}_t^i)$ denotes the propagated confidence from the t_0th frame onto superpixel $\mathcal{P}_t^i$ on the tth frame. The connection weight $\omega(\mathcal{P}_{t-1}^j, \mathcal{P}_t^i)$ is the number of pixels in $\mathcal{P}_{t-1}^j$ mapped to $\mathcal{P}_t^j$ through optical flows. To quantify the transfer error of optical flows, we introduce a re-weighting term $\Phi_{t_0,t}(\mathcal{P}_t^i)$ for each superpixel, measuring the quality of propagated confidences from the t_0th frame to $\mathcal{P}_t^i$. Large value of Φ indicates high confidence and should contribute more to the propagation to the next frame. We follow [65] and interpret $\Phi_{t_0,t}(\mathcal{P}_t^i)$ as the percentage of pixels in $\mathcal{P}_t^i$ correctly tracked during the propagation. In this manner, the computation of Φ has the following recursive structure:

$$1-\Phi_{t_0,t}(\mathcal{P}_t^i)=\sum_j \frac{\omega(\mathcal{P}_{t-1}^j,\mathcal{P}_t^i)}{|\mathcal{P}_t^i|}(1-\Phi_{t_0,t-1}) + \sum_j \frac{\omega(\mathcal{P}_{t-1}^j,\mathcal{P}_t^i)}{|\mathcal{P}_t^i|}\Phi_{t_0,t-1}(\mathcal{P}_{t-1}^j)\Psi(\mathcal{P}_{t-1}^j). \quad (14)$$

The right hand of (14) quantifies the percentage of pixels in $\mathcal{P}_t^i$ that are mis-tracked during propagation, and is derived from two parts. The first part counts the pixels that already lose identities in the previous steps, which will certainly introduce errors when doing propagation to $\mathcal{P}_t^i$. The second part accounts for the other pixels correctly tracked during all the time from the t_0th to the $(t-1)$th frame, but wrongly propagated in the last hop. We quantify the percentage of mis-tracked pixels in a single hop as $\Psi(\mathcal{P}_{t-1}^j)=\exp(-\frac{1}{\lambda_g}\|\nabla \mathbf{f}(\mathcal{P}_{t-1}^j)\|)$ where $\mathbf{f}(\mathcal{P}_{t-1}^j)$ is the mean magnitude of the flow gradients, a quantity often used to measure the reliability of optical flow tracking [48], [65]. It is efficient to compute although it does not directly imply the definition of Ψ, i.e., the number of correctly tracked pixels. More intuitive definitions (e.g., the count of consistent pixels through forward-backward check [59]) could be used for more principled definition.

Given the recursive structures of (13) and (14), the propagation is efficient with two-pass dynamic programming: one pass for the weights Φ and one pass for the confidence scores. Compared with the per-frame voting scheme (11), it suppresses background confidence effectively and produces complete object shapes, see Fig. 3 for visual comparisons.

GrabCut Labeling. As this paper focuses on class-level object segmentation, the initial tracks and the inferred shape priors from the same category are merged together before further processing. They are then integrated into the classic GrabCut framework [53] to generate final results: the unary cost linearly combines the color likelihoods and shape prior maps, while the definition of the pairwise term follows [48], which models color contrasts among spatiotemporally adjacent superpixels. The initial color models are learned on the (merged) masks for each label. Learning the color models and optimizing the labels are repeated until convergence. As the energy is submodular, label optimization can be done efficiently with the alpha-beta move algorithm [5].

3.4 Comparison with Previous Work

This work is an extension of our previous work [73], while the main difference lies in three aspects:

First, a new algorithm is proposed for track initialization. The algorithm benefits from the special problem structure and powerful optimization techniques, and is thus efficient. In contrast, our previous algorithm forms multiple candidate tracks and then selects a subset of tracks from these candidates by solving a quadratic binary problem. Although this strategy also obtains promising results, it is somewhat heuristic and less efficient in practice.

Second, we propose a novel algorithm for shape prior estimation. In our algorithm, each pixel location aggregates votes from a wide coverage of segment proposals in the video, which generate complete and consistent shape priors. In contrary, our previous work votes for each pixel using many segment tracks that are expected to cover the objects or object parts. This strategy suppresses background well, but may wrongly suppress deformable object parts that are not tracked easily, leading to incomplete shape priors.

Finally, in this work we conduct extensive evaluations on large numbers of videos from Youtube-Objects [28], [62], SegTrack v2 [36] and FBMS-59 datasets [47] and various semantic categories. Results on various datasets demonstrate the effectiveness of the proposed approach and show improved performance over our previous work in both accuracy and speed.

4 Experiments

4.1 Experimental Settings

We use videos from Youtube-Objects dataset [51] for evaluation as it shares 10 Pascal VOC [17] object categories where off-the-shelf detectors are available. On this dataset, pixel-level groundtruths are avaliable for two subsets collected by Jain and Grauman [28] and Tang et al. [62], which are referred as YTO-Jain and YTO-Tang, respectively. YTO-Jain dataset consists of 126 videos, which are accurately labeled every 10 frames. In contrary, YTO-Tang dataset is comprised of 151 videos but objects are roughly and densely labeled on supervoxel level. Each of them is among the largest video segmentation benchmarks, with more than 20,000 frames in total and up to 400 frames per video. We also use a subset of the YTO-Jain dataset only consisting of the annotated frames, denoted with YTO-Jain-Sub. On this smaller dataset, we extensively evaluate the proposed approach to see its performance under various conditions.

We make comparisons with 8 existing approaches of three types with available results/codes:

1) *The unsupervised group LTV* [47], *FST* [48], *ACO* [29] and *NLC* [18]. These approaches segment one or more object instance(s) in unconstrained videos based on motion and saliency analysis, and are proven to show good results in various real-world settings.
2) *The weakly supervised approaches CRANE* [62] and *MWS* [42]. They simultaneously segment the common objects in a set of videos labeled with the same category(s) via weakly supervised learning, and are among the state-of-the-arts on YTO-Tang dataset.
3) *The supervised group DTM* [4] and *FCN* [43]. DTM is a recently proposed detection-based approach relying on the R-CNN detector [23], while FCN is the state-of-the-art semantic segmentation model trained on massive manual segmentations. DTM is directly comparable as it also only requires bounding box detectors.

We evaluate 2 variants of the proposed approach: *OSD-DPM*, *OSD-FRCNN* which are equipped with the DPM [19] and the state-of-the-art Fast R-CNN [22] detectors, respectively. Our previous work *OSD-P-DPM* [73] employs DPM detectors. These detectors are pre-trained on Pascal VOC dataset without being re-trained or fine-tuned with additional data. The free parameters λ_1 and λ_2 in (2) are set to 10 and 1,000, respectively. As for the metric, we follow previous works and use the Intersection-over-Union overlap ratio (IoU) to evaluate each video, which is computed as $\frac{TP}{TP+FP}$ where TP (FP) is the number of true (false) positive foreground pixel labels. On YTO-Tang dataset, mean

TABLE 1
Quantitative Results on YTO-Jain Dataset, Reported as IoU

Method	Plane	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Cls. Avg.	Vid. Avg.
LTV	0.137	0.122	0.108	0.237	0.186	0.163	0.180	0.115	0.106	0.196	0.155	0.156
FST	0.709	0.706	0.425	0.652	0.521	0.445	0.653	0.535	0.442	0.296	0.538	0.539
ACO	0.630	0.690	0.400	0.610	0.480	0.460	**0.670**	0.530	**0.470**	0.380	0.530	0.534
FCN	0.593	0.676	0.326	0.505	0.331	0.274	0.356	0.460	0.184	0.473	0.418	0.373
DTM	0.744	0.721	**0.585**	0.600	0.457	**0.612**	0.552	**0.566**	0.421	0.367	0.562	0.558
OSD-P-DPM	0.680	0.671	0.488	**0.709**	0.388	0.567	0.493	0.514	0.420	0.517	0.545	0.518
OSD-DPM	0.698	0.677	0.515	0.695	0.408	0.599	0.614	0.512	0.435	0.525	0.568	0.557
OSD-FRCNN	**0.899**	**0.750**	0.568	0.693	**0.555**	0.553	0.636	0.458	0.430	**0.631**	**0.617**	**0.591**

Along each column, bold highlights the top place while underline the second.

TABLE 2
Quantitative Results on YTO-Jain-Sub Dataset, Reported as IoU

Method	Plane	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Cls. Avg.	Vid. Avg.
FST	0.431	0.641	0.315	0.393	0.325	0.337	0.390	0.325	0.195	0.341	0.369	0.355
ACO	0.575	0.607	0.374	0.311	0.360	0.312	0.458	0.407	0.217	0.342	0.396	0.389
NLC	0.549	0.598	0.269	0.436	0.297	0.336	0.421	0.351	0.270	0.450	0.398	0.371
OSD-P-DPM	**0.718**	0.666	0.443	**0.668**	0.391	0.553	0.492	0.470	0.430	0.469	0.530	0.504
OSD-DPM	0.671	0.679	0.457	0.623	0.389	0.590	0.544	0.472	**0.434**	0.485	0.534	0.519
OSD-FRCNN	0.616	**0.680**	**0.553**	0.637	**0.527**	**0.619**	**0.614**	**0.489**	0.412	**0.511**	**0.566**	**0.568**

Along each column, bold highlights the top place while underline the second.

Average Precision (mAP) is adopted by treating segmentation as a pixel-level classification problem (computed on the produced soft segmentations before graph-cut smoothing), to make direct comparisons with [62] and [42].

4.2 Comparison with State-of-the-Arts

4.2.1 Comparison with Existing Approaches

Results on YTO-Jain dataset are shown in Table 1. Although equipped with weaker detectors, OSD-DPM performs similarly with the recent detection-based approach DTM. When using the same RCNN detector, OSD-FRCNN achieves significantly improved results. We suspect that jointly selecting detection and segment proposals obtains more robust initial tracks than solely linking the inconsistent detections. OSD-DPM performs consistently better than our previous work OSD-P-DPM on almost all the categories. The strong FCN model does not generalize well to the Youtube-Objects dataset, which may owe to the domain transfer from image to video, as models trained on clean high-quality images often perform poorly on low-quality web videos [52], [61]. FST and ACO perform favorably on object categories with salient motion, e.g., *cat*, dog and *horse*. However, due to the lack of high-level knowledge, they may fail to detecting several objects without sufficiently salient motion, leading to worse performance on *train*, *cow* and *boat*.

Table 2 summarizes the results of the proposed approach and several state-of-the-art unsupervised approaches on the YTO-Jain-Sub dataset. Since YTO-Jain-Sub exhibits significantly faster camera/object move but less visual information than YTO-Jain, unsupervised approaches have difficulties in correctly detecting the target objects. In contrary, the proposed approach only encounters slight degeneration thanks to the aid of detection cues. As a result, the proposed approach obtains more than 30 percent improvements over other models, with substantial gains on almost all the categories.

In Fig. 4, OSD-DPM and OSD-FRCNN outperforms the weakly supervised approaches on the YTO-Tang dataset by a large margin. As groundtruth objects on this dataset are weakly annotated at supervoxel-level, the results may bias towards the weakly supervised approaches which also operate on supervoxels. Thus, further improvements can be expected if accurate annotations become available. However, the precision-recall curves in Fig. 4 suggest that the maximal precisions of OSD-DPM and OSD-FRCNN still fall in $60 \sim 80$ percent, meaning that more than 20 percent semantic objects are missed. These statistics imply a large space to improve for both object detection and segmentation in videos.

Note that in all the evaluations, OSD-FRCNN improves over OSD-DPM significantly. It is predictable, as the Fast RCNN detector improves the detection mAP significantly over DPM (67.8 versus 43.6 percent, evaluated on the YTO-Jain dataset). Large improvements are observed on *aeroplane*, *bird*, *cat* and *train*, where several videos failed by DPM are succesfully handled via Fast RCNN. Interestingly, on many categories (e.g., *car*, *cow* and *motorbike*) OSD-DPM performs comparably with OSD-FRCNN. We observe that the final segmentations are reasonable in many cases even if the detections are far from satisfactory. In Fig. 5, we show

Method	mAP
CRANE	0.425
MWS	0.461
OSD-DPM	0.532
OSD-FRCNN	0.645

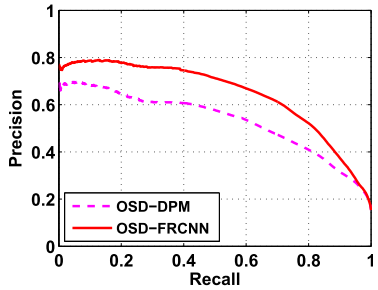


Fig. 4. Quantitative results on YTO-Tang dataset, reported as mAP (left) and precision-recall curves (right).

Fig. 5. Representative examples generated by our approach on Youtube-Objects dataset. Rows 1-4: successful cases. We show our results in case of high-quality detections (left), highly inconsistent detections (middle) and multiple objects (right). Row 5: partially successful examples. Our approach may miss some object parts when the detections are unreliable (left), fail to locate the objects in some frames in presence of large camera motion (middle) and fail to segment the undetected objects (right). Row 6: typical failure cases, including invalid bounding box assumption (left), missing detections (middle) and false positives (right). See text for details.

representative results generated by the proposed approach. Note that it can successfully segment the objects although the detections vary significantly in locations and scales.

In the last row of Fig. 5, we show several typical failure cases of the proposed approach. In the left, the aeroplane only occupies a small portion of its detection box. Without prior object knowledge, the track initialization stage tends to select the segment proposals that can fill the detection boxes, which may include large background area and confuse the refinement stage. The middle example shows that the proposed approach may fail if the detectors cannot hit the target object (e.g., the extremely small cow present in the video). In the right example, false positive detections appear consistently in the video and are considered reliable, which incurs large segmentation errors. These cases could be handled by incorporating category-specific object shape priors [72] and stronger detectors.

4.2.2 Comparison with the Previous Work [73]

The IoU of the proposed approach OSD-DPM is 2.3 and 1.5 percent higher than our previous work OSD-P-DPM on YTO-Jain and YTO-Jain-Sub datasets, respectively. Major improvements are observed on non-rigid categories (i.e., *cat*, *cow*, *dog*). To understand the rationale behind the improvements, two additional experiments are conducted on the YTO-Jain-Sub dataset, as summarized in Table 3 and Fig. 6, respectively. For ease of presentation, we refer to the track initialization and refinement algorithms proposed in this paper as TI and TR, while those proposed in the previous work [73] as TI-P and TR-P, respectively.

TABLE 3
Evaluating Track Initialization Algorithms on the YTO-Jain-Sub Dataset

Algorithm	Detector	IoU	Secs/frame
TI-P	DPM	0.519	1.64
	Fast RCNN	0.563	1.58
TI	DPM	0.519	0.76
	Fast RCNN	0.566	0.62

In Table 3, we show the final results when applying TI and TI-P for track initialization, respectively. For both scenarios, we adopt TR for track refinement. Similar accuracies are obtained by TI nad TI-P with different detectors, while TI costs only less than half the time taken by TI-P. We find that although both algorithms are efficient in many cases, TI-P is sometimes significantly slower on several long videos with multiple objects. We also note that the Fast RCNN detections are slightly faster to be processed than the DPM detections. We suspect that the stronger Fast-RCNN detector distinguishes from the foreground and background regions better and make the initialization process less confusing.

Table 3 also shows that the performance of OSD-P-DPM is improved if TR-P is replaced with TR. To better understand the contributions of TR, we compute the precisions and recalls of the shape priors generated by different refinement algorithms as well as their time cost in Fig. 6. A

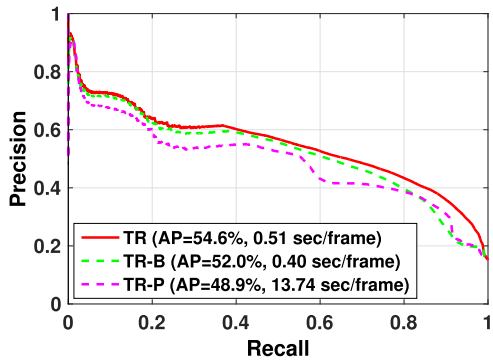


Fig. 6. Evaluating shape prior estimation algorithms on the YTO-Jain-Sub dataset as precision-recall curves. The legend shows the Average Precision (AP) and time cost of different algorithms.

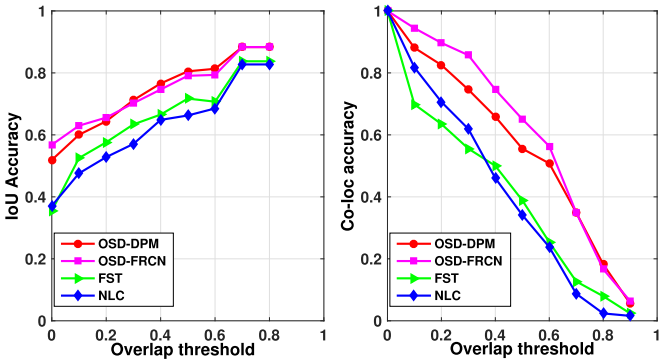


Fig. 7. Left: IoU scores (left) and object CorLoc accuracies of different approaches as functions of the overlap threshold on the YTO-Jain-Sub dataset. (Best viewed in color.)

baseline TR-B is also involved that excludes the error re-weighting term in (13). Fig. 6 shows that TR improves the Average Precision (AP) over TR-P while being a magnitude faster. Such improvement lies in that the novel voting scheme can generate more complete object shape priors by aggregating the information of all the proposals, while TR-P may ignore the proposals that are not consistently tracked and thus miss some object parts. TR also outperforms TR-B with negligible time burden, suggesting that handling long-range propagation errors is desirable for optical flow based estimation of shape priors.

4.3 Performance Analysis

In this section, we conduct additional experiments to see how the proposed approach works and further demonstrate its effectiveness. To comprehensively analyze different aspects of the proposed approach, the following evaluations are taken on the smaller YTO-Jain-Sub dataset. If not explicitly explained, OSD-DPM is used as the baseline approach.

4.3.1 Detection Cues versus Bottom-Up Cues

In the first experiment, we aim to understand how much the detection cues improve over the conventional bottom-up cues (e.g., motion and saliency) towards high-quality video object segmentation. To this end, we report the object segmentation and localization performance of the proposed approach and two state-of-the-art unsupervised approaches FST and NLC on the videos correctly handled by all the competitors. A video is considered to be correctly handled by an approach if its IoU score on this video is above a given threshold. In Fig. 7, such threshold is chosen uniformly from 0 to 1 and the segmentation and localization performances are reported accordingly. Segmentation accuracy is reported as IoU scores, while the localization task follows the CorLoc metric [60], i.e., percentage of the correctly localized objects under Pascal VOC object detection criterion [17]. Note again that at each threshold, the numbers are averaged on the videos correctly handled by all the approaches.

It is observed from Fig. 7 that the proposed OSD-DPM and OSD-FRCNN follow closely, while both achieve much higher segmentation and localization accuracies than the unsupervised approaches relying on bottom-up cues. This fact suggests that 1) the spatial extents of the video objects can be retrieved more accurately with top-down detection cues, while bottom-up cues only may lead to over(or under)-estimation by missing object parts or including additional background. 2) The right of Fig. 7 also shows that incorporating detection cues help detect more video objects that might be missed by bottom-up approaches.

4.3.2 Ablation Studies of Different Components

In the second experiment, we propose to see the contributions of the track initialization and track refinement components. Two baselines are implemented: *Base-INT* takes the initial tracks as final segmentation, in which the refinement process is excluded. *Base-GC* uses the bounding boxes of the initial segments to bootstrap the GrabCut [53] process, which is used to isolate the proposed shape priors.

The performances on the subset of [28] are summarized in Table 4. Note that the initial tracks selected by OSD-INT already obtains higher accuracy than the unsupervised approaches FST and NLC (see Table 2). The OSD-DPM improves over OSD-GC by 3.2 percent, demonstrating the effectiveness of the proposed shape priors than a simple bounding-box prior. OSD-GC even harms the results on *Boat* and *Train*, which attributes to the highly inconsistent detection boxes generated on these categories. On the contrary, the proposed shape priors improve the performance on all the categories.

4.3.3 Sensitiveness to Detection Thresholds

In the third experiment, we evaluate the sensitiveness of our approach w.r.t. different detection thresholds. We vary the thresholds uniformly from $[0, 1]$ and study both the IoU segmentation accuracies and detection F1-scores, which are summarized in Fig. 8. The detection F1-score is calculated as $\frac{2 \cdot P \cdot R}{P+R}$, where P and R are detection precisions and recalls computed following the Pascal object detection Criteria [17].

Fig. 8 shows that the segmentation accuracy only slightly changes while the detection curve has large variance. Notably is that the segmentation performance is only 0.9 percent worse when all the detections are fed into our algorithm (i.e., the threshold is zero). It shows the robustness of the track initialization process, and is desired in practice since choosing a proper detection threshold is usually not trivial.

In Table 1, the performance of OSD-DPM (IoU = 51.9 percent) is computed using the optimal threshold estimated from

TABLE 4
Results of Ablation Studies on the YTO-Jain-Sub Dataset, Reported as IoU

Method	Plane	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Cls. Avg.	Vid. Avg.
OSD-INT	0.457	0.570	0.407	0.601	0.306	0.450	0.408	0.357	0.344	0.437	0.434	0.413
OSD-GC	**0.696**	0.648	0.413	0.563	0.336	0.544	**0.546**	0.428	0.406	0.424	0.500	0.487
OSD-DPM	0.671	**0.679**	**0.457**	**0.623**	**0.389**	**0.590**	0.544	**0.472**	**0.434**	**0.485**	**0.534**	**0.519**

Bold highlights the top place while underline the second.

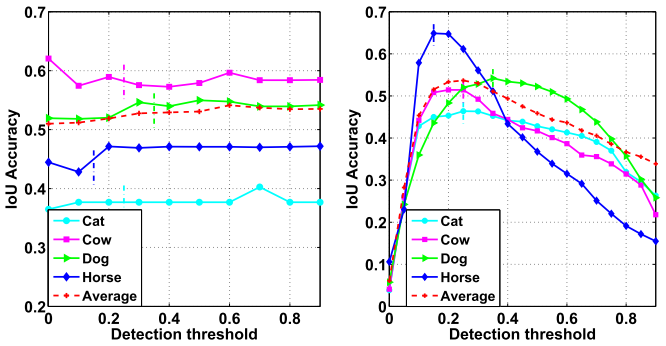


Fig. 8. IoU scores (left) and detection F1-scores (right) of the proposed approach as functions of detection thresholds on the YTO-Jain-Sub dataset. The optimal threshold for each category (where F1-score is maximized) is indicated using vertical dashed lines. For clarity, we illustrate four dominant categories with the most instances and the curves averaged on all classes. (Best viewed in color.)

the detection F1-scores (see the vertical dashed lines in Fig. 8). Interestingly, Fig. 8 shows that the optimal thresholds do not actually lead to the best segmentation performance. In fact, the accuracy of OSD-DPM reaches 54.1 percent when the thresholds of all categories are set to 0.6, which is even close to OSD-FRCNN, which reaches 56.9 percent. It partly owes to a limitation of detection-based approaches: if some false positive detections are not successfully suppressed, they may establish wrong segment tracks and thus incur large segmentation errors. As a result, the performance tends to improve when larger threshold (thus less false positives) is used. Also note that the *cow* category is an exception as it achieves best accuracy when the threshold is zero. In this category, multiple object instances are present in many videos, thus a smaller detection threshold is able to improve the recall and lead to higher segmentation accuracy.

4.3.4 Impact of Spatial and Temporal Terms

The fourth experiment aims to see the impact of the terms (4) and (5), which stands for the degree of tracking consistency and spatial exclusion of the selected tracks, respectively. To this end, we sample λ_1 and λ_2 of (2) uniformly in log scale, and compute the mean IoU scores at each parameter combination. The score distributions shown in Fig. 9 suggest that non-zero assignment of λ_1 and λ_2 increases the results, and the performance is stable for a wide range of parameters, i.e., $1 \leq \lambda_1 \leq 100$ and $\lambda_2 > 0$. Small λ_1 ignores tracking consistency, while large value tends to break the tracking at

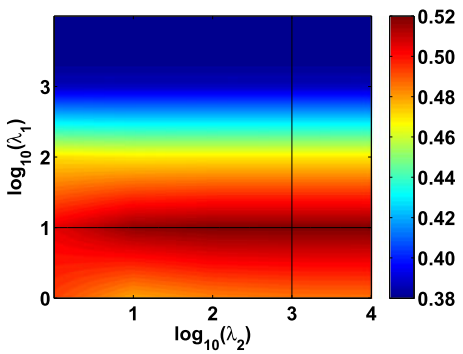


Fig. 9. Average IoU scores of the proposed approach as a function of value combinations of λ_1 and λ_2 on the YTO-Jain-Sub dataset. The cross of two slices shows the best accuracy. (Best viewed in color.)

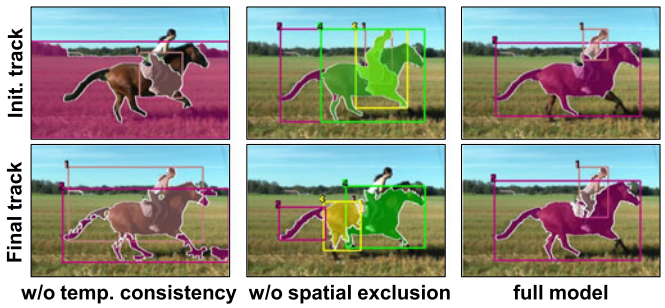


Fig. 10. The initial (top) and the final (bottom) object tracks generated by incorporating one and both of the spatial and temporal terms in Eq. (2), evaluated on the video *horses01* from the FBMS-59 dataset. Colors represent different objects. (Best viewed in color with zoom.)

weak links and produce fragmented tracks. Incorporating the spatial exclusion term has a positive effect in most cases, since it helps remove false positive tracks and distinguish between interacting objects. Fig. 10 illustrates the impacts of different terms on a sample video frame. Without enforcing temporal consistency, incorrect initial object segments are selected due to detection failures. If spatial mutual exclusion is not considered, many false positive tracks with high detection scores are established and the interacting objects are merged together. Both cases would lead to unexpected final results. The joint model, on the contrary, localizes the objects more accurately.

4.3.5 Multi-Class/Object Video Segmentation

Although the results are primarily shown on single-class video segmentation, the proposed approach naturally handles multiple classes/objects (to handle multiple object instances, one can assign a unique label for each initial track before GrabCut). To evaluate this setting on wild categories, we collect 9 additional videos from public benchmarks: 3 are collected from the SegTrack v2 [36] dataset (*bmx*, *drift* and *hummingbird*) while the others are from the FBMS-59 [47] dataset (*cars5*, *cars10*, *cats07*, *horses01*, *horses04* and *people04*). These videos are chosen to be comprised of multiple object instances from at least one Pascal VOC category, and exhibit varying time durations from 30 to 800 frames. We reuse their original annotations and re-annotate several of them so that ground-truth masks are available for each object instance.

In Table 5, the proposed approach OSD-FRCNN is compared with several existing multi-label segmentation approaches LTV, DTM, FCN and the MCM [31]. Following previous works [4], [36], for each annotated object we report the IoU of the best object track produced by each algorithm. Results show that LTV and MCM perform well on video objects with distinct and rigid video motion (e.g., *cars5*, *cars10*), but fail to separating objects with strong interactions (e.g., *horses01*, *horses04*). FCN does not recognize different objects from the same class, thus perform worse on videos such as *cars5* and *drift*. Our approach performs better or comparably in these cases. Fig. 11 shows some qualitative results generated by the proposed approach. Although reasonable results are achieved in various cases, the proposed approach still has difficulty segmenting heavily overlapping objects (e.g., the birds in *cats07*). Without stronger instance-specific cues, it is still challenging to accurately segment videos with such strong instance occlusions.

TABLE 5
Quantitative Results on Multi-Class/Object Segmentation on Videos from the FBMS-59 and SegTrack v2 Datasets, Reported as IoU

Video-Object	LTV	MCM	DTM	FCN	OSD
bmx-person	0.048	0.704	**0.907**	0.009	0.826
bmx-bicycle	0.012	0.173	**0.335**	0.200	0.297
drift-car#1	0.351	0.502	0.701	0.360	**0.813**
drift-car#2	0.124	0.003	0.602	0.369	**0.679**
hummingbird-bird#1	0.039	0.110	0.104	0.398	**0.652**
hummingbird-bird#2	0.554	0.324	0.094	0.302	**0.556**
cars5-car#1	0.026	0.010	-	0.003	**0.150**
cars5-car#2	**0.921**	0.920	-	0.151	0.855
cars5-car#3	0.760	**0.788**	-	0.229	0.758
cars10-bus	0.637	0.742	-	**0.785**	0.747
cars10-car	0.802	**0.803**	-	0.750	0.789
cats07-bird#1	0.014	**0.561**	-	0.315	0.492
cats07-bird#2	0.028	0.007	-	0.230	**0.318**
cats07-cat	0.019	**0.761**	-	0.119	0.746
horses01-person	0.272	0.233	-	0.452	**0.463**
horses01-horse	0.554	0.475	-	**0.714**	0.569
horses04-person	0.099	0.260	-	**0.509**	0.497
horses04-horse#1	0.645	0.506	-	0.667	**0.687**
horses04-horse#2	0.105	0.115	-	0.040	**0.280**
people04-person#1	0.021	**0.567**	-	0.399	0.164
people04-person#2	**0.311**	0.283	-	0.221	0.212
people04-motorbike	0.490	0.527	-	0.582	**0.618**
Average on videos	0.316	0.423	-	0.369	**0.569**
Average on objects	0.311	0.426	-	0.355	**0.553**

Bold highlights the top place while underline the second.

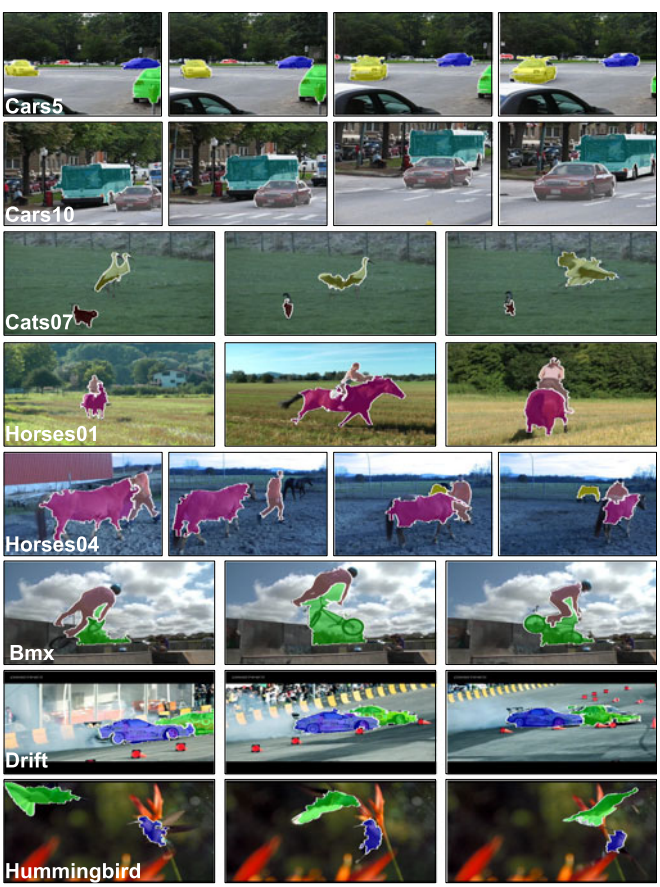


Fig. 11. Representative results generated by the proposed approach on segmenting multiple classes/objects. Colors represent different object instances. Best viewed in color with zoom.

TABLE 6
Average Running Time of Our Approach on the Subset of [28]

Stage	Secs/frame
Pre-processing	63.11
Track init. (graph construction)	26.14
Track init. (optimization)	0.76
Track refine. (shape prior estimation)	0.51
Track refine. (grab-cut optimization)	1.57

4.3.6 Running Time

In Table 6, we summarize the average time cost of different stages of our approach on the YTO-Jain-Sub dataset. The evaluation is based on an unoptimized single-thread MATLAB implementation run on a 3.4 GHz processor. On this dataset, most video shots have resolution 640×360.

The time bottlenecks of our approach lie in the stages of pre-processing and graph-construction. In the current implementation, we use [7], [19] and [3] to compute the optical flows, detections and segment proposals, respectively. For acceleration, the pre-processing stage can integrate state-of-the-art fast implementations (e.g., [22], [59], [75]), while graph construction is highly parallelizable. Other stages are relatively efficient. Empirical comparisons suggest that the proposed approach is faster than state-of-the-art proposal-based approaches (e.g., [15], [34]). However, it is still slower than several unsupervised approaches [18], [48]. Exploring more efficient ways to incorporate image-based detections would be interesting for future research on semantic object segmentation in videos.

5 Conclusion and Discussion

This paper proposes a segmentation-by-detection approach for semantic object segmentation in tagged video. It starts with generating per-frame detection and segment proposals on various frames. After that, object tracks are initialized efficiently through solving a joint assignment problem, which is shown to be robust to the detection noise. The initial tracks are finally refined with a voting-based algorithm that can generate spatiotemporally consistent shape priors. From extensive experiments, we demonstrate that image-based object detections can significantly boost the segmentation performance in tagged videos while not introducing additional supervisions.

Our results reveal that the full spatial extents of semantic objects are not trivial to capture with only bottom-up models. We believe that exploring top-down cues provided by pre-trained image-based models could be effective for handling weakly labeled videos. In the future, we will extend this idea for object segmentation in multiple weakly labeled videos, exploring the inter-video similarities to jointly refine per-video detections/segmentations. Another interesting direction is to combine object occlusion cues and detection cues for robust instance-level video object segmentation.

Acknowledgments

We would like to thank the anonymous reviewers for their help in improving this work. This work was supported in part by grants from the National Natural Science

Foundation of China (61325011, 61532003 and 61421003). Earlier version of this work has been published in CVPR 2015 [73].

REFERENCES

[1] 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. and Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[3] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 328–335.

[4] B. Drayer and T. Brox, "Object detection, tracking, and motion segmentation for object-level video segmentation," *arXiv:1608.03066*, 2016.

[5] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[6] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[7] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[8] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.

[9] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, "On pairwise costs for network flow multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5537–5545.

[10] A. Y. C. Chen and J. J. Corso, "Temporally consistent multi-class video-object segmentation with the video graph-shifts algorithm," in *Proc. IEEE Workshop Motion Video Comput.*, 2011, pp. 614–621.

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

[12] X. Chen, J. Li, Q. Li, B. Gao, D. Zou, and Q. Zhao, "Image2scene: Transforming style of 3d room," in *Proc. ACM Conf. Multimedia Conf.*, 2015, pp. 321–330.

[13] 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.

[14] J. Dong, Q. Chen, S. Yan, and A. Yuille, "Towards unified object detection and semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 299–314.

[15] Z. Dong, J. Omar, 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.

[16] Z. Dong, J. Omar, and M. Shah, "Video object co-segmentation by regulated maximum weight cliques," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 551–566.

[17] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[18] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting," in *Proc. Brit. Mach. Vis. Conf.*, 2014, p. 34.

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[20] G. Floros and B. Leibe, "Joint 2d-3d temporally consistent semantic segmentation of street scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2823–2830.

[21] H. Fu, D. Xu, B. Zhang, and S. Lin, "Object-based multiple foreground video co-segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3166–3173.

[22] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[23] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[24] I. Gurobi Optimization , "Gurobi optimizer reference manual," 2016.

[25] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 297–312.

[26] G. Hartmann, et al., "Weakly supervised learning of object segmentations from web-scale video," in *Proc. Eur. Conf. Comput. Vis. Workshop Web-Scale Vis. Social Media*, 2012, pp. 198–208.

[27] A. Jain, S. Chatterjee, and R. Vidal, "Coarse-to-fine semantic video segmentation using supervoxel trees," in *Proc. IEEE Conf. Comput. Vis.*, 2013, pp. 1865–1872.

[28] S.-D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 656–671.

[29] W. D. Jang, C. Lee, and C. S. Kim, "Primary object segmentation in videos via alternate convex optimization of foreground and background distributions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 696–704.

[30] K. Kavukcuoglu, P. Sermanet, Y. L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition," in *Proc. Advances Neural Inform. Process. Syst.*, 2010, pp. 1090–1098.

[31] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3271–3279.

[32] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 703–718.

[33] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. Torr, "What, where and how many? combining object detectors and crfs," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 424–437.

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

[35] C. Li, L. Lin, W. Zuo, S. Yan, and J. Tang, "Sold: Sub-optimal low-rank decomposition for efficient video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5519–5527.

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

[37] J. Li, L. Duan, X. Chen, T. Huang, and Y. Tian, "Finding the secret of image saliency in the frequency domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2428–2440, Dec. 2015.

[38] X. Liang, et al., "Reversible recursive instance-level object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 633–641.

[39] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan, "Proposal-free network for instance-level object segmentation," *arXiv:1509.02636*, 2015.

[40] L. Lin, X. Wang, W. Yang, and J. Lai, "Discriminatively trained and-or graph models for object shape detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 959–972, May 2015.

[41] B. Liu and X. He, "Multiclass semantic video segmentation with object-level active inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4286–4294.

[42] X. Liu, D. Tao, M. Song, Y. Ruan, C. Chen, and J. Bu, "Weakly supervised multiclass video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 57–64.

[43] 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.

[44] J. Lu, R. Xu, and J. J. Corso, "Human action segmentation with hierarchical supervoxel consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3762–3771.

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

[46] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–27, 2001.

[47] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.

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

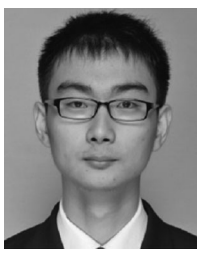
[49] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1201–1208.

[50] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularize likelihood methods," in *Proc. Advances Large Margin Classifiers*, 2000, pp. 61–74.
[51] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3282–3289.
[52] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video," *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[53] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": Interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, 2004, pp. 309–314.
[54] J. C. Rubio, J. Serrat, and A. López, "Video co-segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2013, pp. 13–24.
[55] O. Russakovsky, et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[56] K. Schindler and H. Wang, "Smooth foreground-background segmentation for video processing," in *Proc. Asian Conf. Comput. Vis.*, 2006, pp. 581–590.
[57] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev, "Instance-level video segmentation from object tracks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.
[58] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 1–15.
[59] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 438–451.
[60] K. Tang, A. Joulin, L. J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1464–1471.
[61] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller, "Shifting weights: Adapting object detectors from image to video," in *Proc. Advances Neural Inform. Process. Syst.*, 2012, pp. 638–646.
[62] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei, "Discriminative segment annotation in weakly labeled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2483–2490.
[63] B. Taylor, A. Ayvaci, A. Ravichandran, and S. Soatto, "Semantic video segmentation from occlusion relations within a convex optimization framework," in *Proc. Int. Conf. Energy Minimization Methods Comput. Vis. Pattern Recognit.*, 2013, pp. 195–208.
[64] J. Tighe and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3001–3008.
[65] S. Vijayanarasimhan and K. Grauman, "Active frame selection for label propagation in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 496–509.
[66] 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.
[67] P. Wang, X. Shen, Z. Lin, S. Cohen, B. L. Price, and A. L. Yuille, "Joint object and part segmentation using deep learned potentials," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1573–1581.
[68] W. Brendel and S. Todorovic, "Segmentation as maximum-weight independent set," in *Proc. Advances Neural Inform. Process. Syst.*, 2010, pp. 307–315.
[69] W. Xia, C. Domokos, J. Dong, L.-F. Cheong, and S. Yan, "Semantic segmentation without annotating segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2176–2183.
[70] J. Xie, M. Kiefel, M. T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3d to 2d label transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3688–3697.
[71] B. Xin, Y. Tian, Y. Wang, and W. Gao, "Background subtraction via generalized fused lasso foreground modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4676–4684.
[72] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1731–1743.
[73] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia, "Semantic object segmentation via detection in weakly labeled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3641–3649.
[74] B. Zhong, et al., "Background subtraction driven seeds selection for moving objects segmentation and matting," *Neurocomputing*, vol. 103, pp. 132–142, 2013.
[75] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

Yu Zhang received the BE degree in computer science from Beihang University, Beijing, China, in 2012. Since then, he is working towards the PhD degree with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Egnineering, Beihang University. His research interests include computer vision and image/video processing. He is a student member of the IEEE.

Xiaowu Chen received his Ph.D. degree in computer science from Beihang University, in 2001. He is currently a professor in the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. His research interests include virtual reality, augmented reality, computer graphics and computer vision. He is a senior member of the IEEE.

Jia Li received the BE degree from Tsinghua University, in 2005 and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2011. He is currently an associate professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. His research interests include computer vision and image/video processing. He is a senior member of the IEEE.



Chen Wang received the MS degree from the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, in 2015. His research interests during the master studies include computer vision and machine learning.

Changqun Xia is currently working towards the PhD degree with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. His research interests include computer vision and image processing.

Jun Li is currently working towards the PhD degree with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. His research interests are computer vision and machine learning.