# CDTS: Collaborative Detection, Tracking, and Segmentation for Online Multiple Object Segmentation in Videos

Yeong Jun Koh
Korea University
yjkoh@mcl.korea.ac.kr

Chang-Su Kim
Korea University
changsukim@korea.ac.kr

## Abstract

*A novel online algorithm to segment multiple objects in a video sequence is proposed in this work. We develop the collaborative detection, tracking, and segmentation (CDTS) technique to extract multiple segment tracks accurately. First, we jointly use object detector and tracker to generate multiple bounding box tracks for objects. Second, we transform each bounding box into a pixel-wise segment, by employing the alternate shrinking and expansion (ASE) segmentation. Third, we refine the segment tracks, by detecting object disappearance and reappearance cases and merging overlapping segment tracks. Experimental results show that the proposed algorithm significantly surpasses the state-of-the-art conventional algorithms on benchmark datasets.*

## 1. Introduction

Multiple object segmentation (MOS) is the task to extract multiple segment tracks by separating objects from one another and from the background in a video sequence. It is challenging to segment multiple objects in a video, without any prior information about objects or user annotations. Furthermore, MOS is difficult due to background clutters, object overlapping, occlusion, disappearance, appearance variation, and fast motion.

MOS algorithms [1, 2, 4, 15, 16, 21–23, 30–32, 34, 37, 42] attempt to achieve pixel-wise delineation of as many objects as possible without user annotations. Most conventional algorithms [1, 2, 4, 15, 16, 21–23, 30, 31, 34, 37, 42] are offline approaches, which process all frames at once. In other words, the entire information in a video is required to achieve the segmentation. These algorithms gather motion trajectories [1, 2, 4, 21–23, 31] or region proposals [15, 16, 30, 34, 37, 42] from all frames, and then cluster them into segments. On the other hand, an online approach, such as [32], extracts objects from each frame using the information in the current and past frames only. An offline approach may achieve more accurate segmentation, but it demands future frames, thereby increasing memory and computational complexities. In contrast, an online approach can extract objects sequentially from the first to the last frames. In this regard, online MOS is more practical than offline one.

We propose a novel online MOS algorithm to extract segment tracks for multiple objects in a video sequence. We develop the collaborative detection, tracking, and segmentation (CDTS) technique to achieve accurate online MOS. First, we use object detector and tracker jointly to generate bounding box tracks for multiple objects. Second, we transform each bounding box into a pixel-wise segment, by shrinking and expanding foreground regions alternately, and then link it to the corresponding segment track. This alternate shrinking and expansion (ASE) is performed sequentially to achieve online MOS. Third, we present a scheme for segment track management, which exploits segmentation results to detect object disappearance, object reappearance, and duplicated segment tracks. Experimental results demonstrate that the proposed algorithm significantly outperforms the state-of-the-art MOS algorithms on the YouTube-Objects dataset [27] and FBMS dataset [23]. To summarize, this work has the following contributions.

- We simultaneously perform object detection, tracking, and segmentation to achieve MOS online, whereas most conventional algorithms are offline approaches.
- We develop the segment track management scheme to detect disappearing and reappearing objects and merge duplicated segment tracks for the same object.
- The proposed algorithm yields remarkable performances on the YouTube-Objects and FBMS benchmarks datasets.

## 2. Related Work

### 2.1. Video Object Segmentation

Video object segmentation (VOS) can be classified into three categories: semi-supervised VOS, single VOS, and MOS.

**Semi-Supervised VOS:** In semi-supervised VOS, user annotations are required about target objects in the first frame of a video sequence. Then, the manually annotated mask is propagated temporally to achieve segmentation in subsequent frames [19, 28, 33, 35, 39]. The region-based particle filter [35], the seam carving [28], and the occluder-occluded relationship [39] are used to perform non-rigid tracking of target objects. Märki *et al*. [19] propagate user annotations in the bilateral space. Tsai *et al*. [33] jointly optimize VOS and optical flow estimation. Also, Perazzi *et al*. [25] construct an appearance model (*i.e*. a support vector machine classifier) of a target object, by employing user annotations. However, note that the manual delineation or annotation is exhausting and inconvenient.

**Single VOS:** Single VOS algorithms automatically separate a single primary object from the background in a video [9, 11, 18, 24, 36, 38, 40], where the primary object refers to the most frequently appearing object in the video. They use various cues for the primary object segmentation. Papazoglou and Ferrari [24] generate motion boundaries to find moving objects, but they may fail to extract static objects. In [11, 18, 40], object proposal techniques are adopted to determine candidate regions of a primary object. However, many false positives, such as background proposals, are also generated, degrading the segmentation performance. In [9, 36, 38], saliency maps are used for the initial estimation of a primary object. These saliency-based techniques are vulnerable to inaccurate saliency detection results due to background clutters or background motions.

**MOS:** MOS algorithms produce multiple segment tracks. An approach is the motion segmentation that clusters point trajectories in a video [1, 2, 4, 21–23, 31]. Each motion segment becomes one segment track. Shi and Malik [31] adopt the normalized cuts to divide a frame into motion segments. Brox and Malik [1] perform point tracking to construct sparse long-term trajectories and divide them into multiple clusters. Ochs and Brox [21] develop a sparse-to-dense interpolation scheme to transform sparse clusters into dense motion segmentation. Ochs *et al*. [23] segment moving objects based on both [1] and [21]. Ochs and Brox [22] employ the spectral clustering to group trajectories based on a higher-order motion model. Fragkiadaki *et al*. [4] improve the segmentation performance on boundaries of moving objects, by analyzing trajectory discontinuities. Chen *et al*. [2] adopt a Markov random field model to refine initial trajectory clusters. These motion segmentation algorithms, however, cannot segment static objects effectively.

Another approach is based on region proposals. Specifically, region proposals are generated in each frame, and they are matched between neighboring frames [16] or clustered [15, 37] to yield multiple segment tracks. However, in [15, 16, 37], different segment tracks may overlap with one another. In other words, there may be multiple segment tracks including the same object. Moreover, false positives may occur, which mostly include background regions. Among these overlapping or erroneous segment tracks, Tsai *et al*. [34] attempt to select true object tracks, by comparing them with segment tracks in other videos. They also employ a semantic segmentation scheme to obtain region proposals.

Taylor *et al*. [32] identify occluders to segment multiple objects online. Also, some MOS algorithms [30, 42] employ pre-trained object detection techniques. Zhang *et al*. [42] propose a segmentation-by-detection framework, which combines object detection and semantic segmentation. However, they need the prior information of video-level object categories. Seguin *et al*. [30] perform MOS, guided by object detection and tracking, but they focus on the segmentation of the person category. In this work, we also adopt a pre-trained object detector [17] to initialize locations of multiple objects.

### 2.2. Multiple Object Tracking

Similar to MOS, multiple object tracking (MOT) also attempts to locate multiple objects in a video. However, it identifies the location of each object with a bounding box, instead of pixel-wise segmentation. The MOT problem is often decomposed into object detection and global data association. The MOT algorithms in [7, 10, 20, 41] focus on the global data association, which finds the optimal path to link object detection results across frames. On the other hand, Kim and Kim [12] propose the cooperative detection and tracking algorithm, in which the tracker restores undetected objects while the detector guides the tracker to recognize the disappearance or occlusion of target objects.

## 3. Proposed Algorithm

The proposed CDTS algorithm yields multiple segment tracks online without requiring any user annotations. Each segment track (or spatiotemporal segment) is composed of segmentation masks of a detected object in the frames, in which the object occurs. Thus, the output is a sequence of pixel-wise label maps for frames, and all pixels in a spatiotemporal segment are assigned the same label. The proposed algorithm is an online approach in the sense that it produces a pixel-wise label map for each frame causally without using the information in future frames.

Figure 1 is an overview of the proposed CDTS algorithm. First, we generate bounding box tracks for objects, called object tracks, by employing object detector and tracker jointly. At the first frame, we use the object detector to yield a bounding box for each object and initialize an object track. At each subsequent frame, we generate two kinds of bounding boxes using the detector and the tracker, respectively, and match them. For each matched pair, we select the more accurate box to represent the object, and

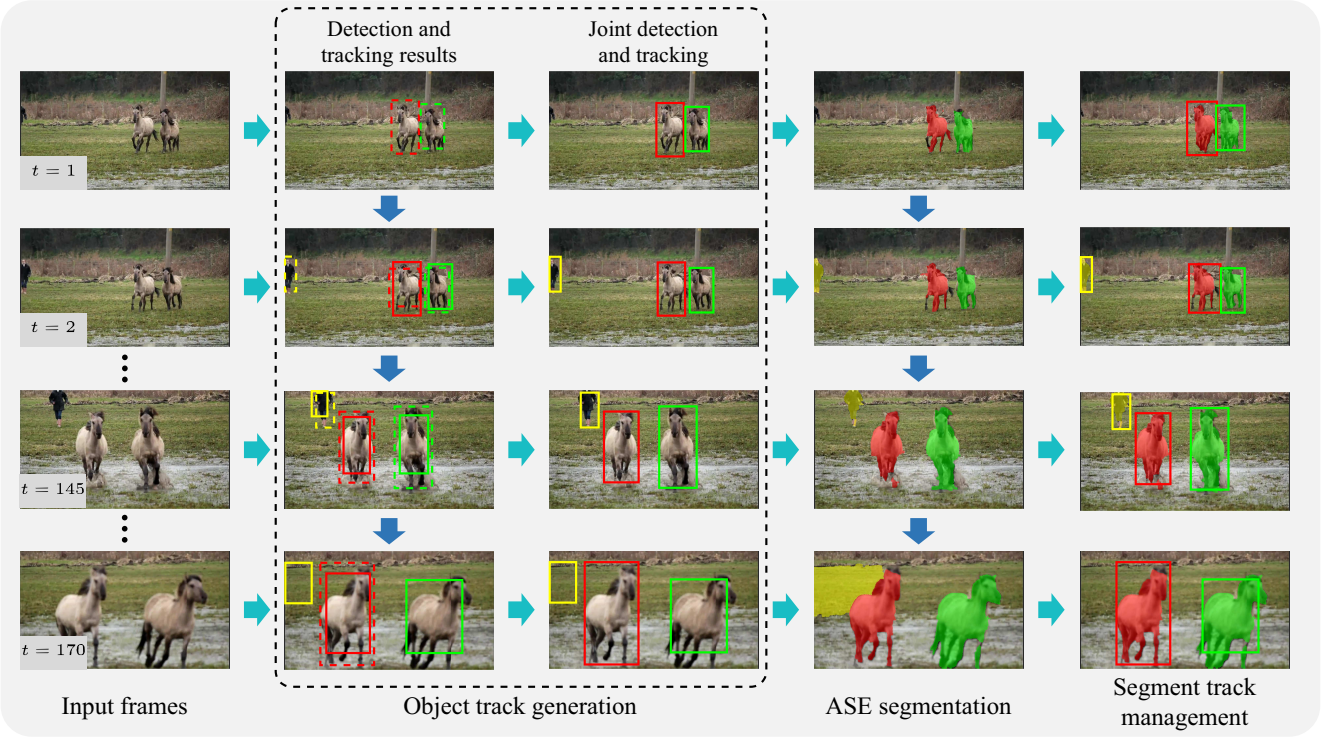



Figure 1. An overview of the proposed algorithm. In 2nd column, dotted and solid boxes depict detection and tracking results, respectively.

link the selected box to the corresponding object track. We regard unmatched detection boxes as newly appearing objects, and initialize new object tracks for them. Second, we identify a pixel-wise segment from each bounding box in an object track, by shrinking and expanding foreground regions alternately, and add it to the corresponding segment track. Third, we refine the segment tracks, by handling object disappearance and reappearance cases and merging overlapped segment tracks.

### 3.1. Object Track Generation

**Detector:** We adopt the object detector R-FCN [17] to locate objects without manual annotations. Note that other detectors, such as [6, 29], also can be used instead of R-FCN. In each frame, we measure the detection scores for region proposals, obtained by the region proposal network in [29], using the R-FCN detector and choose only the proposals whose scores are higher than a threshold $\theta_{\text{det}} = 0.5$. Although R-FCN provides category-wise scores, we use the maximum of those scores. Since the purpose of the proposed algorithm is to extract and delineate objects regardless of their categories, we use the maximum score as the objectness of the proposal.

**Tracker:** Since many proposals are discarded by the detection threshold $\theta_{\text{det}}$, some objects may remain undetected. For example, the detector cannot detect the person within a green box in Figure 2, since he is small and partially occluded. To boost the recall rate of objects using temporal correlations in a video, we employ a model-free tracker [13]. For each detected object in a previous frame $t-1$, the tracker estimates its location in the current frame $t$. Specifically, given the $i$th bounding box $b_i^{(t-1)}$ in frame $t-1$, a search region is set in frame $t$ and candidate boxes within the search region are sampled by the sliding window method. The feature $\phi(b)$ of each candidate box $b$ is described by the combination of an RGB histogram and an HOG histogram. Then, the bounding box $b_i^{(t-1)}$ is traced to the new location in the current frame $t$, which is given by

Figure 2. Joint detection and tracking. Dotted and solid rectangles represent detection and tracking boxes, respectively.

$$b_i^{(t)} = \arg\max_{b \in \mathcal{R}} \mathbf{w}_i^T \phi(b) \tag{1}$$

where $\mathcal{R}$ denotes the search region and $\mathbf{w}_i$ is the appearance model of the target object.

**Joint Detection and Tracking:** At the first frame, we locate objects using the detector and initialize an object track for each detected object. From the second frames, we extend the object tracks by employing the detector and the

tracker jointly. As illustrated in Figure 2, there are three cases after the detection and tracking: an object is included by 1) both detection and tracking boxes, 2) only detection box, or 3) only tracking box.

In case 1), a detection box in the current frame and a tracking box traced from the previous frame include the same object. In this case, we select the more accurate box between the two boxes. For example, dotted and solid yellow rectangles in Figure 2 are detection and tracking boxes, respectively. Since the cow has a scale variation, the detector [17] provides a better result than the fixed-scale tracker [13]. We match detection and tracking boxes and choose the better box for each matching pair similarly to [12]. The matching cost $C_{\mathrm{m}}(b_{\mathrm{d}}, b_{\mathrm{t}})$ between a detection box $b_{\mathrm{d}}$ and a tracking box $b_{\mathrm{t}}$ is defined as

$$C_{\mathrm{m}}(b_{\mathrm{d}}, b_{\mathrm{t}}) = \begin{cases} ||\phi(b_{\mathrm{d}}) - \phi(b_{\mathrm{t}})||^2 & \text{if } \Omega(b_{\mathrm{d}}, b_{\mathrm{t}}) \geq \theta_{\mathrm{iou}} \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

where $\Omega(b_{\mathrm{d}}, b_{\mathrm{t}})$ is the intersection over union (IoU) ratio and $\theta_{\mathrm{iou}}$ is set to 0.3. Thus, we match detection and tracking boxes only if their IoU ratio is greater than $\theta_{\mathrm{iou}}$. After computing all matching costs between detection and tracking boxes, we find the optimal set of matching pairs using the Hungarian algorithm [14], which carries out greedy one-to-one matching in a bipartite graph. For each matching pair, we compare the detection scores for the two boxes and select the better box with the higher score. Suppose that the tracking box is traced from the $i$th object track in the previous frame. Then, the selected box is denoted by $b_i^{(t)}$ and is linked to the $i$th object track to form the extended track $\mathcal{O}_i^{(t)} = \{b_i^{(t_i)}, b_i^{(t_i+1)}, \ldots, b_i^{(t)}\}$, where $t_i$ is the first appearance time of the track.

In case 2), an unmatched detection box is regarded as a newly appearing object and is used to initialize a new object track. A dotted cyan box in Figure 2 shows a detection result including a new object. In case 3), an unmatched tracking box is simply linked to the corresponding object track. Finally, for each object track $\mathcal{O}_i^{(t)}$, we update the appearance model $\mathbf{w}_i$ in (1) using the feature vector $\phi(b_i^{(t)})$ of the bounding box $b_i^{(t)}$, as done in [13].

### 3.2. ASE Segmentation

From the bounding box $b_i^{(t)}$ of the $i$th object track in frame $t$, we segment out foreground pixels to transform the object track into a segment track. After over-segmenting frame $t$ into superpixels, we dichotomize each superpixel within and near the box $b_i^{(t)}$ into either foreground or background class. Notice that we perform the segmentation independently for each object track.

**Over-Segmentation:** For the over-segmentation, we compute an ultrametric contour map (UCM) [26] as in Figure 3(b). Each region, enclosed by a boundary in the UCM, becomes a superpixel in Figure 3(c). Let $\mathcal{S} = \{s_1, \ldots, s_M\}$ be the set of superpixels. For each superpixel $s_m$, we calculate the size ratio $\Delta(s_m, b_i^{(t)})$ of the intersection region between $s_m$ and $b_i^{(t)}$ to the superpixel region $s_m$. In other words, $\Delta(s_m, b_i^{(t)})$ represents the percentage of pixels in the superpixel $s_m$ that are included in the bounding box $b_i^{(t)}$. Then, we roughly divide $\mathcal{S}$ into the foreground region $\mathcal{F}_i^{(t)}$ and the background region $\mathcal{B}_i^{(t)}$ by

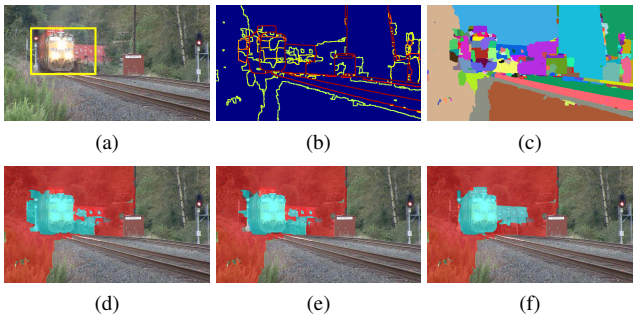


Figure 3. The ASE segmentation of a frame in the "Train0001" sequence: (a) input frame with a bounding box $b_i^{(t)}$, (b) UCM, (c) superpixels, (d) preliminary classification, (e) intra-frame refinement, and (f) inter-frame refinement.

$$\mathcal{F}_i^{(t)} = \{s_m : \Delta(s_m, b_i^{(t)}) \geq \theta_{\mathrm{ratio}}\}, \tag{3}$$

$$\mathcal{B}_i^{(t)} = \{s_m : 0 < \Delta(s_m, b_i^{(t)}) < \theta_{\mathrm{ratio}}\}, \tag{4}$$

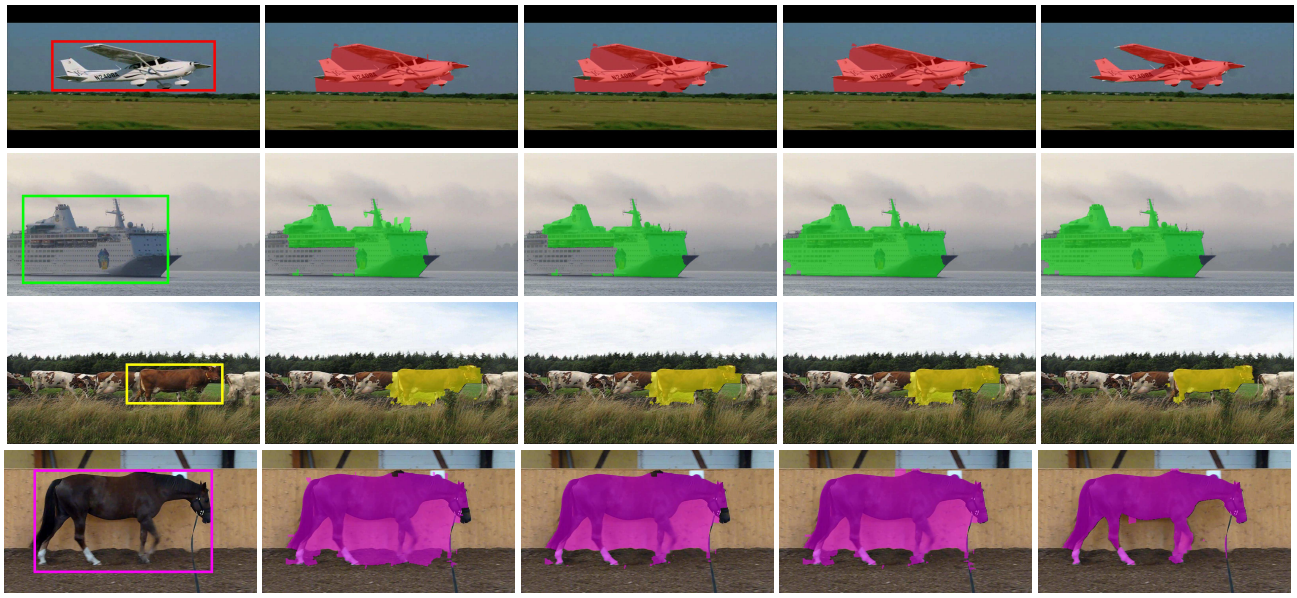
where $\theta_{\mathrm{ratio}}$ is set to 0.9.

In Figure 3(d), $\mathcal{F}_i^{(t)}$ and $\mathcal{B}_i^{(t)}$ are colored in cyan and red, respectively. Being constrained by the rectangular shape of $b_i^{(t)}$ in Figure 3(a), the preliminarily classified $\mathcal{F}_i^{(t)}$ misses parts of the object. Also, $\mathcal{F}_i^{(t)}$ includes background pixels. To improve the segmentation performance, we shrink and expand the preliminary foreground region alternately in two steps: intra-frame refinement and inter-frame refinement.

**Intra-Frame Refinement:** Suppose that the object in $b_i^{(t)}$ first appears in the current frame $t$. In other words, $t_i = t$. Since the proposed algorithm operates online, it can use only the information in frame $t$ to refine the foreground region $\mathcal{F}_i^{(t)}$ for $b_i^{(t)}$. For this intra-frame refinement, we constrain the foreground region to have intense edge strengths along its boundary. To this end, we define the boundary cost, by employing the UCM $U^{(t)}$ for frame $t$, which is given by

$$C_{\mathrm{bnd}}(\mathcal{F}_i^{(t)}) = -\sum_{\mathbf{x} \in \partial \mathcal{F}_i^{(t)}} U^{(t)}(\mathbf{x}) \tag{5}$$

where $\partial \mathcal{F}_i^{(t)}$ denotes the set of the boundary pixels of the foreground region $\mathcal{F}_i^{(t)}$. Due to the minus sign in (5), the



(a) Input with bounding boxes (b) Preliminary classification (c) Intra-frame shrinking (d) Inter-frame expansion (e) Inter-frame shrinking

Figure 4. Step-by-step segmentation results in the proposed ASE segmentation. From top to bottom, the "Aeroplane0002," "Boat0003," "Cow0002," and "Horse0014" sequences. Note that each step improves the segmentation performance.

minimization of $C_{\text{bnd}}(\mathcal{F}_i^{(t)})$ enforces $\mathcal{F}_i^{(t)}$ to have the maximum edge strengths at the boundary pixels.

We refine the foreground region $\mathcal{F}_i^{(t)}$ in (3) to reduce the boundary cost in (5), by removing superpixels from $\mathcal{F}_i^{(t)}$ in a greedy manner. Since the joint detection and tracking in Section 3.1 attempts to include a whole object within a bounding box, the initial foreground region $\mathcal{F}_i^{(t)}$ in (3), as well as the bounding box $b_i^{(t)}$, includes more false positives than false negatives. Hence, in the intra-frame refinement, we only shrink the foreground region by removing superpixels from $\mathcal{F}_i^{(t)}$. Adding new superpixels into $\mathcal{F}_i^{(t)}$ is performed in the inter-frame refinement.

For each superpixel $s_m \in \mathcal{F}_i^{(t)}$ along the boundary of $\mathcal{F}_i^{(t)}$, we compute the boundary cost $C_{\text{bnd}}(\mathcal{F}_i^{(t)} \setminus s_m)$ of the set difference $\mathcal{F}_i^{(t)} \setminus s_m$. We then select the optimal superpixel $s_m^*$ to minimize the cost. Then, after removing $s_m^*$ from $\mathcal{F}_i^{(t)}$, we repeat this process until the boundary cost stops decreasing. Figure 3(e) shows the shrinking result of the intra-frame refinement.

**Inter-Frame Refinement:** In the inter-frame refinement, we constrain that the refined foreground region should be similar to the segmentation results in previous frames, while dissimilar from the background region in the current frame. To quantify similarity between regions, we extract the features $\mathbf{f}_{\text{f},i}^{(t)}$ and $\mathbf{f}_{\text{b},i}^{(t)}$ of $\mathcal{F}_i^{(t)}$ and $\mathcal{B}_i^{(t)}$, respectively, by employing the bag-of-visual-words (BoW) approach [3]. We design the LAB BoW feature using the 40 training sequences in the VSB100 dataset [5]. We quantize the LAB colors, extracted from the training sequences, into 300 codewords using the K-means algorithm. By associating each pixel with the nearest codeword, we obtain the BoW histogram of the codewords for the pixels in $\mathcal{F}_i^{(t)}$, and normalize it into the feature vector $\mathbf{f}_{\text{f},i}^{(t)}$. Also, $\mathbf{f}_{\text{b},i}^{(t)}$ is obtained in the same way.

For the inter-frame refinement, we define a cost function

$$C_{\text{inter}}(\mathcal{F}_i^{(t)}, \mathcal{B}_i^{(t)}) \quad (6)$$
$$= \alpha \cdot C_{\text{tmp}}(\mathcal{F}_i^{(t)}) + C_{\text{seg}}(\mathcal{F}_i^{(t)}, \mathcal{B}_i^{(t)}) + C_{\text{bnd}}(\mathcal{F}_i^{(t)})$$

where $C_{\text{tmp}}$ is the temporal cost, $C_{\text{seg}}$ is the segmentation cost, $C_{\text{bnd}}$ is the boundary cost in (5), and $\alpha$ is set to 5.

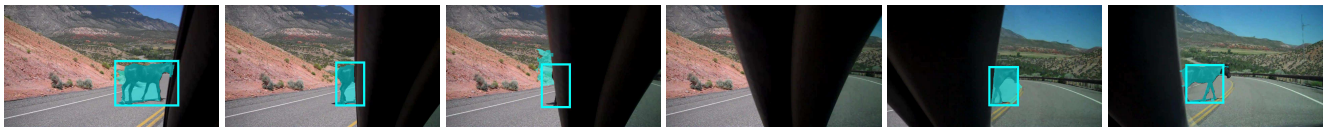
To achieve temporal consistency, the temporal cost $C_{\text{tmp}}(\mathcal{F}_i^{(t)})$ in (6) enforces the feature of the foreground region $\mathcal{F}_i^{(t)}$ to be similar to those of the foreground regions in the previous frames. Specifically,

$$C_{\text{tmp}}(\mathcal{F}_i^{(t)}) = \frac{1}{N_c} \sum_{\tau=t-N_c}^{t-1} d_\chi(\mathbf{f}_{\text{f},i}^{(t)}, \mathbf{f}_{\text{f},i}^{(\tau)}) \quad (7)$$

where $N_c$ specifies the temporal range and is fixed to 10 in this work. We employ the chi-square distance $d_\chi$, which compares two histograms effectively. On the other hand, to minimize the similarity between the foreground region $\mathcal{F}_i^{(t)}$ and background region $\mathcal{B}_i^{(t)}$, the segmentation cost $C_{\text{seg}}(\mathcal{F}_i^{(t)}, \mathcal{B}_i^{(t)})$ is defined as

$$C_{\text{seg}}(\mathcal{F}_i^{(t)}, \mathcal{B}_i^{(t)}) = -d_\chi(\mathbf{f}_{\text{f},i}^{(t)}, \mathbf{f}_{\text{b},i}^{(t)}). \quad (8)$$

Based on the overall inter cost $C_{\text{inter}}$ in (6), we further refine the foreground region $\mathcal{F}_i^{(t)}$, which is already processed by the intra-frame refinement. In the inter-frame refinement, we perform expansion and shrinking alternately.



(a) Frame 99 (b) Frame 103 (c) Frame 105 (d) Frame 107 (e) Frame 113 (f) Frame 116

Figure 5. A disappearing and reappearing horse in the "Horse0012" sequence. Bounding boxes and segmented foreground regions are depicted in cyan. The segment track for the horse is discontinued at frame 105 and then reconnected at frame 113.

**Algorithm 1** ASE Segmentation

**Input:** Bounding box $b_i^{(t)}$ in frame $t$

Obtain the set of superpixels $\mathcal{S} = \{s_1, \ldots, s_M\}$ using UCM
Classify each $s_m$ into $\mathcal{F}_i^{(t)}$ and $\mathcal{B}_i^{(t)}$ according to $\Delta(s_m, b_i^{(t)})$
**if** the object in $b_i^{(t)}$ first appears in frame $t$ **then**
    Update $\mathcal{F}_i^{(t)}$ and $\mathcal{B}_i^{(t)}$ based on $C_{\text{bnd}}(\mathcal{F}_i^{(t)})$
**else**
    Update $\mathcal{F}_i^{(t)}$ and $\mathcal{B}_i^{(t)}$ based on $C_{\text{bnd}}(\mathcal{F}_i^{(t)})$
    Update $\mathcal{F}_i^{(t)}$ and $\mathcal{B}_i^{(t)}$ based on $C_{\text{inter}}(\mathcal{F}_i^{(t)}, \mathcal{B}_i^{(t)})$
**end if**

**Output:** Foreground region $\mathcal{F}_i^{(t)}$

We first expand the foreground region $\mathcal{F}_i^{(t)}$ using adjacent superpixels in $\mathcal{B}_i^{(t)}$. More specifically, for each superpixel $s_n \in \mathcal{B}_i^{(t)}$ sharing a boundary with $\mathcal{F}_i^{(t)}$, we compute the inter cost $C_{\text{inter}}(\mathcal{F}_i^{(t)} \cup s_n, \mathcal{B}_i^{(t)} \setminus s_n)$. Then, we select the optimal superpixel $s_n^*$ to minimize the cost. We then augment $\mathcal{F}_i^{(t)}$ by moving $s_n^*$ from $\mathcal{B}_i^{(t)}$ to $\mathcal{F}_i^{(t)}$. This expansion step is repeated until the inter cost stops decreasing. After the expansion, we perform the shrinking in a similar way, by considering the inter cost $C_{\text{inter}}(\mathcal{F}_i^{(t)} \setminus s_m, \mathcal{B}_i^{(t)} \cup s_m)$ for removing a superpixel $s_m$.

Notice that both expansion and shrinking steps in the inter-frame refinement use the same cost function and decrease it monotonically. Thus, the alternate application of the expansion and the shrinking eventually converges when neither expansion nor shrinking can reduce the cost function. However, in practice, we observe that one iteration of the expansion and the shrinking provides sufficiently good segmentation performance. Figure 3(f) shows this inter-frame refinement example. Figure 4 shows more step-by-step segmentation results. **Algorithm 1** summarizes the ASE segmentation algorithm.

### 3.3. Segment Track Management

We manage segment tracks, by observing segmentation results. As illustrated in Figure 5, a target object may disappear from the view, and a disappeared object may reappear into the view. Also, as shown in Figure 6(a), a part of an existent object may be detected as a new object. In this work, we use segmentation results to detect the disappearance or reappearance of a target object and to merge duplicated segment tracks for the same object.

**Disappearance and Reappearance:** A target object may be out of the view or be occluded by other objects or the background. A detection score can be used to identify the disappearance of an object [12]. However, a severely occluded object, *e.g.* the horse in Figure 5(b), yields a very low detection score, and it may be incorrectly declared as a disappearance case. We detect disappearance cases more effectively using the segmentation information. When an object disappears in frame $t$, the segmentation result is likely to include nearby background regions, as in Figure 5(c). Thus, we expect that the segmentation result of a disappeared object includes background features rather than foreground features. Therefore, using the feature vectors of the foreground and background regions in previous frames, we compute two dissimilarities, given by

$$d_{\text{fore}} = \frac{1}{N_c} \sum_{\tau=t-N_c}^{t-1} d_\chi(\mathbf{f}_{\text{f},i}^{(t)}, \mathbf{f}_{\text{f},i}^{(\tau)}), \tag{9}$$

$$d_{\text{back}} = \frac{1}{N_c} \sum_{\tau=t-N_c}^{t-1} d_\chi(\mathbf{f}_{\text{f},i}^{(t)}, \mathbf{f}_{\text{b},i}^{(\tau)}). \tag{10}$$

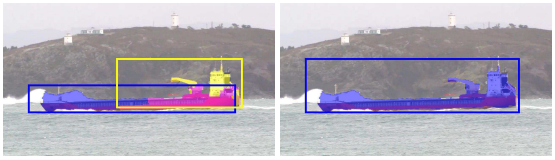
When $d_{\text{fore}}$ is larger than $d_{\text{back}}$, we declare that the object has disappeared and stop tracking it.

Also, a disappeared object may reappear into the view after some frames, as in Figure 5(e). We attempt to reconnect a reappearing object to its previous segment track. When object $j$ is newly detected in frame $k$, we check whether it is compatible with a discontinued segment track $i$. Specifically, we compare the foreground region $\mathcal{F}_j^{(k)}$ of object $j$ with $\mathcal{F}_i^{(\tilde{t}_i - N_c)}, \ldots, \mathcal{F}_i^{(\tilde{t}_i - 1)}$, where $\tilde{t}_i$ is the index of the frame when object $i$ disappears. We compute the dissimilarity

$$\frac{1}{N_c} \sum_{\tau=\tilde{t}_i-N_c}^{\tilde{t}_i-1} d_\chi(\mathbf{f}_{\text{f},j}^{(k)}, \mathbf{f}_{\text{f},i}^{(\tau)}) \tag{11}$$

and reconnect object $j$ to segment track $i$ if the dissimilarity is smaller than $\theta_{\text{dis}} = 0.5$.

**Mergence:** Even though an object in segment track $i$ has been tracked and segmented up to frame $t$, its partial region can be detected as another segment track $j$. This happens when the detected bounding box of the partial region



(a) (b)

Figure 6. Overlapped segment tracks for the same object. Two segments in (a), covering the same object, are integrated into one segment in (b) based on the mergence score.

is unmatched by the Hungarian algorithm in Section 3.1. In Figure 6(a), an existent object $\mathcal{F}_i^{(t)}$ and a newly detected overlapping region $\mathcal{F}_j^{(t)}$ are colored in blue and yellow, respectively, and their intersection is in magenta. These two regions, $\mathcal{F}_i^{(t)}$ and $\mathcal{F}_j^{(t)}$, should be merged. When $\mathcal{F}_i^{(t)}$ and $\mathcal{F}_j^{(t)}$ overlap each other, we determine whether to merge them using the mergence score $\Psi_{ij}$, given by

$$\Psi_{ij} = \frac{\lambda_{i\cup j}}{\frac{1}{2}(\lambda_i + \lambda_j)}(1 - d_\chi(\mathbf{f}_{\mathrm{f},i}^{(t)}, \mathbf{f}_{\mathrm{f},j}^{(t)})) \tag{12}$$

where $\lambda_i$ and $\lambda_j$ are the detection scores of the bounding boxes $b_i^{(t)}$ and $b_j^{(t)}$. Also, $\lambda_{i\cup j}$ is the detection score of the merged box $b_{i\cup j}^{(t)}$, depicted by a blue rectangle in Figure 6(b). The mergence score in (12) considers the objectness of the merged box as well as the similarity between the two regions $\mathcal{F}_i^{(t)}$ and $\mathcal{F}_j^{(t)}$. If $\Psi_{ij}$ is greater than $\theta_{\mathrm{mrg}} = 0.9$, we merge $\mathcal{F}_j^{(t)}$ into $\mathcal{F}_i^{(t)}$ as in Figure 6(b).

## 4. Experimental Results

We assess the performance of the proposed CDTS algorithm on the YouTube-Objects dataset [27] and the FBMS dataset [23]. The proposed CDTS algorithm has five thresholds, which are fixed in all experiments: $\theta_{\mathrm{det}} = 0.5$, $\theta_{\mathrm{iou}} = 0.3$ in (2), $\theta_{\mathrm{ratio}} = 0.9$ in (3) and (4), $\theta_{\mathrm{dis}} = 0.5$, and $\theta_{\mathrm{mrg}} = 0.9$.

### 4.1. Evaluation on YouTube-Objects Dataset

YouTube-Objects is a large dataset, containing 126 videos for 10 object classes. These videos are challenging due to appearance change, fast-motion, occlusion, disappearance of objects, and so forth. The pixel-wise ground-truth is provided by [8]. To assess a segmentation result quantitatively, we measure the IoU ratio $\frac{|\mathcal{S}_{\mathrm{est}}\cap\mathcal{S}_{\mathrm{gt}}|}{|\mathcal{S}_{\mathrm{est}}\cup\mathcal{S}_{\mathrm{gt}}|}$, where $\mathcal{S}_{\mathrm{est}}$ and $\mathcal{S}_{\mathrm{gt}}$ are an estimated segment and the ground-truth, respectively. Since the proposed algorithm yields multiple segment tracks, different segment tracks may overlap in a frame. In such a case, we compute the average of frame-by-frame detection scores for each segment track. Then, we keep the foreground region of the best segment track with the highest average score and discard the foreground regions of the other segment tracks.

Table 1. Impacts of the intra-frame refinement (Intra), the inter-frame refinement (Inter), and the segment track management (STM) on the average IoU performance.

| | Baseline | Baseline + Intra | Baseline + Intra + Inter | Baseline + Intra + Inter + STM |
|---|---|---|---|---|
| Average | 0.581 | 0.638 | 0.661 | 0.672 |

**Ablation Study:** Table 1 analyzes the impacts of the intra-frame refinement, the inter-frame refinement, and the segment track management on the segmentation performance. We set preliminary classification results as the baseline. In Table 1, we observe that the intra-frame refinement improves the average IoU ratio significantly. Moreover, the inter-frame refinement and then the segment track management further improve the performances meaningfully. This indicates that all three steps are essential for the proposed algorithm to generate accurate segment tracks.

**Quantitative Comparison:** Table 2 compares the proposed CDTS algorithm with the conventional single VOS [9, 24] and MOS [23, 34, 42] algorithms. We obtain the results of [23, 24, 34, 42] from the paper [34]. We compute the result of [9] using the source code, provided by the respective author. For comparison, we measure the IoU ratios using the ground-truth in [8].

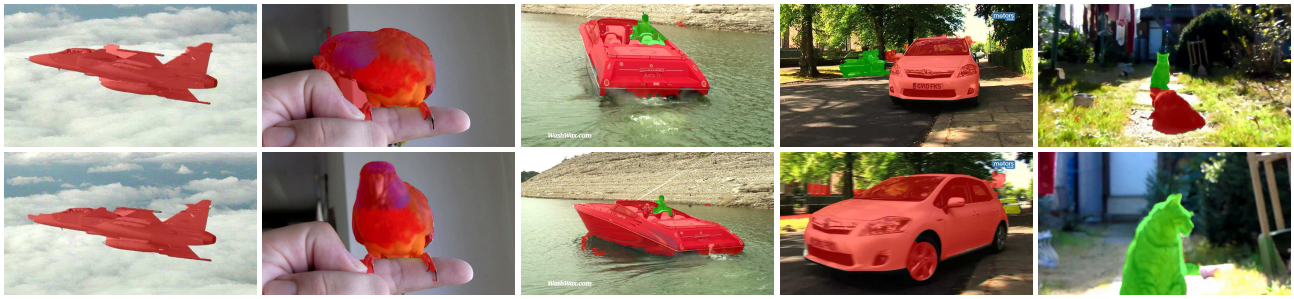
The single VOS algorithms [9,24] miss some foreground regions for videos including multiple objects, since they focus on the segmentation of a single primary object. In the MOS category, the proposed CDTS algorithm outperforms all conventional MOS algorithms [23, 34, 42] significantly. Specifically, the proposed algorithm yields 0.517, 0.148, and 0.091 better IoU ratio than [23], [42], and [34], respectively. It is worth pointing out that [42] and [34] require additional information of video-level object categories and other videos, respectively, whereas the proposed algorithm does not. Also, notice that the proposed algorithm is an online approach, whereas all conventional algorithms are offline ones.

**Qualitative Results:** Figure 7 shows examples of MOS results on the YouTube-Objects dataset. The proposed algorithm segments out multiple objects faithfully. Also, we see that the proposed algorithm deals with partially occluded objects in the "Cow" sequence and disappearing objects in the "Horse" sequence effectively. However, a motorbike and its rider in the "Motorbike" sequence are extracted together in some frames, whereas the ground-truth includes motorbike regions only. This is why the performance on the motorbike class is relatively low as compared with other classes in Table 2.

**More Results:** Although we use the ground-truth in [8] in this paper, they are too rough and need to delineate objects more tightly. We hence modified the ground-truth manually to improve it. More experimental results using this improve ground-truth are available in supplemental materials.

Table 2. Performance comparison of the proposed CDTS algorithm with the conventional algorithms on the YouTube-Objects dataset in terms of the IoU metric, based on the original ground-truth in [8]. The best results are boldfaced.

| | Aeroplane | Bird | Boat | Car | Cat | Cow | Dog | Horse | Motorbike | Train | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A. Single VOS | | | | | | | | | | | |
| [24] | 0.736 | 0.561 | 0.578 | 0.339 | 0.305 | 0.418 | 0.368 | 0.443 | 0.489 | 0.392 | 0.463 |
| [9] | 0.504 | 0.625 | 0.312 | 0.528 | 0.453 | 0.367 | 0.472 | 0.406 | 0.240 | 0.343 | 0.425 |
| B. MOS | | | | | | | | | | | |
| [23] | 0.137 | 0.122 | 0.108 | 0.237 | 0.186 | 0.163 | 0.180 | 0.115 | 0.106 | 0.196 | 0.155 |
| [42] | 0.724 | 0.666 | 0.430 | 0.589 | 0.364 | 0.582 | 0.487 | 0.496 | 0.414 | 0.493 | 0.524 |
| [34] | 0.693 | 0.760 | 0.535 | 0.704 | **0.668** | 0.490 | 0.475 | **0.557** | 0.395 | 0.534 | 0.581 |
| CDTS | **0.786** | **0.758** | **0.649** | **0.762** | 0.634 | **0.642** | **0.720** | 0.523 | **0.562** | **0.680** | **0.672** |



(a) Airplane (b) Bird (c) Boat (d) Car (e) Cat

(f) Cow (g) Dog (h) Horse (i) Motorbike (j) Train

Figure 7. Segmentation results of the proposed CDTS algorithm on the YouTube-Objects dataset. Extracted objects are colored differently.

Table 3. Comparison of IoU scores on the test sequences in the FBMS dataset. The best results are boldfaced.

| | Precision | Recall | F-measure | F-measure$\geq 0.75$ |
|---|---|---|---|---|
| [23] | 0.749 | 0.601 | 0.667 | 20/69 |
| [32] | 0.779 | 0.591 | 0.672 | 15/69 |
| CDTS | **0.778** | **0.715** | **0.745** | **30/69** |

### 4.2. Evaluation on FBMS Dataset

The FBMS dataset [23] is another benchmark for MOS. It consists of 59 video sequences, which are divided into 29 training and 30 test sequences. Since we use the off-the-shelf object detector [17], we do not use the training sequences. For the performance assessment, we use the test sequences. We obtain the results of the conventional MOS algorithms [23, 32] from the respective papers. For comparison, we use the precision, recall, and F-measure as done in [23]. An object with F-measure $\geq 0.75$ is regarded as successfully segmented. In Table 3, the proposed algorithm outperforms the conventional algorithms [23, 32] in terms of all metrics and extracts more objects successfully.

## 5. Conclusions

We proposed a novel online MOS algorithm, referred to as CDTS. We first generated a set of object tracks using the object detector and tracker jointly. Then, to extract pixel-wise segments from the boxes in the object tracks, we developed the ASE segmentation technique. Finally, we performed the segment track management to refine segment tracks. Experimental results demonstrate that the proposed algorithm outperforms the state-of-the-art algorithms significantly on both the YouTube-Objects and FBMS datasets.

## Acknowledgements

This work was supported partly by the National Research Foundation of Korea grant funded by the Korea government (No. NRF2015R1A2A1A10055037), and partly by 'The Cross-Ministry Giga KOREA Project' grant funded by the Korea government (MSIT) (No. GK17P0200, Development of 4D reconstruction and dynamic deformable action model based hyper-realistic service technology).

## References


[1] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*. 2010. 1, 2

[2] L. Chen, J. Shen, W. Wang, and B. Ni. Video object segmentation via dense trajectories. *IEEE Trans. Multimedia*, 17(12):2225–2234, 2015. 1, 2

[3] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 5

[4] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012. 1, 2

[5] F. Galasso, N. S. Nagaraja, T. J. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013. 5

[6] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 3

[7] S. Hamid Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid. Joint probabilistic data association revisited. In *ICCV*, 2015. 2

[8] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 7, 8

[9] W.-D. Jang, C. Lee, and C.-S. Kim. Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In *CVPR*, 2016. 2, 7, 8

[10] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 2

[11] Y. Jun Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 2

[12] H.-U. Kim and C.-S. Kim. CDT: Cooperative detection and tracking for tracing multiple objects in video sequences. In *ECCV*, 2016. 2, 4, 6

[13] H.-U. Kim, D.-Y. Lee, J.-Y. Sim, and C.-S. Kim. SOWP: Spatially ordered and weighted patch descriptor for visual tracking. In *ICCV*, 2015. 3, 4

[14] H. W. Kuhn. The hungarian method for the assignment problem. *Nav. Res. Logist. Quart.*, 2(1-2):83–97, 1955. 4

[15] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 1, 2

[16] F. Li, T. Kim, A. Humayun, D. Tsai, and J. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 1, 2

[17] Y. Li, K. He, J. Sun, et al. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 2, 3, 4, 8

[18] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012. 2

[19] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 2

[20] A. Milan, K. Schindler, and S. Roth. Multi-target tracking by discrete-continuous energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2054–2068, 2016. 2

[21] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011. 1, 2

[22] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, 2012. 1, 2

[23] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1187–1200, 2014. 1, 2, 7, 8

[24] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 2, 7, 8

[25] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015. 2

[26] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(1):128–140, 2017. 4

[27] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 1, 7

[28] S. A. Ramakanth and R. V. Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, 2014. 2

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3

[30] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev. Instance-level video segmentation from object tracks. In *CVPR*, 2016. 1, 2

[31] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998. 1, 2

[32] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015. 1, 2, 8

[33] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 2

[34] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *ECCV*, 2016. 1, 2, 7, 8

[35] D. Varas and F. Marques. Region-based particle filter for video object segmentation. In *CVPR*, 2014. 2

[36] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 2

[37] F. Xiao and Y. J. Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016. 1, 2

[38] J. Yang, B. Price, X. Shen, Z. Lin, and J. Yuan. Fast appearance modeling for automatic primary video object segmentation. *IEEE Trans. Image Process.*, 25(2):503–515, 2016. 2

[39] Y. Yang, G. Sundaramoorthi, and S. Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *ICCV*, 2015. 2

[40] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 2

[41] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 2

[42] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *CVPR*, 2015. 1, 2, 7, 8