Primary Object Segmentation in Videos Based on Region Augmentation and Reduction

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

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

Abstract

A novel algorithm to segment a primary object in a video sequence is proposed in this work. First, we generate candidate regions for the primary object using both color and motion edges. Second, we estimate initial primary object regions, by exploiting the recurrence property of the primary object. Third, we augment the initial regions with missing parts or reducing them by excluding noisy parts repeatedly. This augmentation and reduction process (ARP) identifies the primary object region in each frame. Experimental results demonstrate that the proposed algorithm significantly outperforms the state-of-the-art conventional algorithms on recent benchmark datasets.

1. Introduction

Primary object segmentation (POS) is the task to segment a single primary object from the background in a video sequence, where the primary object means the most frequently appearing and salient object in the video. POS is applicable to many vision tasks, including video summarization, action recognition, and object class learning. POS, however, is challenging due to many difficulties, such as object deformation, occlusion, and background clutters. Especially, without user annotations or any prior information about a primary object (*e.g.* its class), it is difficult to separate the primary object from the background.

POS is closely related to video object segmentation (VOS). VOS can be classified into three categories: semi-supervised, multiple, and unsupervised VOS. Semi-supervised VOS [17, 22, 23, 26, 28, 35] requires manual annotations at the first frame to segment target objects in subsequent frames. Multiple VOS [2, 7, 13, 14, 18, 19, 25, 32] does not demand user annotations about objects, but yields multiple segment tracks. In other words, multiple VOS cannot identify the primary object among the multiple tracks without the ground-truth. In contrast, unsupervised VOS [5, 9, 16, 20, 30, 33, 36] aims to find only a single segment track, corresponding to a primary object, automatically. Thus, unsupervised VOS has the same purpose as POS does. In this regard, we use the terms 'POS' and 'unsupervised VOS' exchangeably. Without manual annotations or ground-truth about a primary object, POS is more challenging than semi-supervised or multiple VOS.

In general, POS methods estimate initial regions of primary objects, and refine the initial results using primary object cues, *e.g.* colors or positions. Object proposals are often used to find initial primary object regions [16, 36]. However, the conventional techniques [16, 36] strongly depend on the objectness score of each proposal, which is computed in each frame, without exploiting the recurrence property of the primary object. For the refinement of initial regions, many POS methods [9, 16, 20, 36] extend the interactive image segmentation technique [24] to the VOS task. They construct Gaussian mixture models (GMMs) for a primary object and the background, respectively, based on initial region estimates. However, these GMMs may fail to model temporally varying object and background information accurately and are vulnerable to incorrect initial estimates.

In this work, we propose a novel POS algorithm to yield a segment track for a primary object in a video sequence. First, we generate a pool of candidate regions for the primary object. To this end, both color and motion edges are used to increase the recall rates of static and moving objects. Second, we estimate initial regions for the primary object, by exploiting the recurrence property of the primary object. Third, instead of the GMM-based refinement, we refine the initial regions by augmenting them with missing parts or reducing them by excluding noisy parts. This augmentation and reduction process (ARP) is performed based on a cost function. By minimizing the cost function iteratively, we achieve the POS. Experimental results demonstrate that the proposed algorithm significantly outperforms the state-of-the-art semi-supervised, multiple, and unsupervised VOS algorithms on the DAVIS [21] and FBMS [2] benchmark datasets.

2. Related work

2.1. Video Object Segmentation

Semi-Supervised VOS: Semi-supervised VOS requires user annotations about target objects. An object is manually delineated in the first frame and then tracked in successive frames [23, 26, 28, 35]. Varas *et al*. [28] employ a region-based particle filter to track a target object. Ramakanth *et al*. [23] adopt the seam carving to detect object boundaries. Yang *et al*. [35] propagate annotated segments using the occluder-occluded relationship. Tsai *et al*. [26] consider VOS and optical flow estimation simultaneously, and refine optical flow vectors using segmentation results. Also, in [17, 22], user annotations are utilized to construct appearance models for objects. Perazzi *et al*. [22] construct a support vector machine (SVM) classifier for a target object. Märki *et al*. [17] optimize the two-class (*i.e*. foreground or background) labeling problem in the bilateral space.

Multiple VOS: Multiple VOS algorithms do not require any manual annotation, but they provide multiple segment tracks. They yield motion segmentation results [2, 7, 18, 19, 25] or video object proposals [13, 14, 32].

Shi and Malik [25] construct a graph based on motion characteristics and divide a frame into segments using the normalized cuts. Brox and Malik [2] form sparse long-term trajectories and cluster them. Ochs and Brox [18] convert the clusters of sparse trajectories in [2] into dense segmentation results, by solving a variational problem. Ochs and Brox [19] adopt the spectral clustering to segment point trajectories. Fragkiadaki *et al*. [7] analyze discontinuities between neighboring trajectories to segment moving objects.

Object proposals are sampled [13, 32] or matched [14] to generate video object proposals. Lee *et al*. [13] cluster object proposals, extracted from an entire sequence, and rank each cluster according to the average objectness score of the elements. They select high rank clusters to yield segment tracks. Xiao and Lee [32] form a proposal group, by gathering the k-nearest neighbors of each proposal, and then train an SVM classifier using the proposal group to extract a segment track. Li *et al*. [14] extract several figure-ground segments in a frame and match those segments in subsequent frames to provide multiple segment tracks.

For the performance assessment, these multiple VOS algorithms [2, 7, 13, 14, 18, 19, 25, 32] require the ground-truth to choose the best segment track among multiple tracks, since they do not consider which track is the most salient.

POS: POS automatically discovers a single primary segment track in a video sequence. Many POS algorithms [9, 16, 20, 30, 36] formulate the segmentation as the two-class labeling problem by constructing models for the primary object and the background, *e.g*. GMMs. To construct those models, they obtain initial regions of the primary object using motion boundaries [20], object proposals [16, 36], or saliency maps [9, 30, 33].

Papazoglou and Ferrari [20] generate motion boundaries for each frame to separate moving objects, but they may fail to segment static objects. Ma and Latecki [16] construct a locally connected graph of object proposals, and select primary object proposals for all frames by optimizing the maximum weight clique problem. Zhang *et al*. [36] design a layered directed acyclic graph of object proposals and find an optimal path in the graph. However, these methods [16, 36] do not consider the recurrence characteristic of a primary object, since they depend on proposal scores, which are computed frame-by-frame.

Wang *et al*. [30] estimate saliency maps using geodesic distances and delineate salient objects. Jang *et al*. [9] estimate initial probability distributions of foreground and background using boundary priors, and refine the probability distributions by optimizing a hybrid of the Markov, spatiotemporal, and antagonistic energies. Yang *et al*. [33] design a graph, which performs the segmentation and appearance modeling simultaneously. Also, Faktor and Irani [5] propose a non-local consensus voting scheme. They perform random walk simulation on a non-locally connected graph for all frames, by employing a saliency map as the initial distribution of the walker. However, these saliency-dependent techniques [5, 9, 30] may face difficulties, when the saliency maps are inaccurate due to background clutters or background motions.

2.2. Primary Object Discovery

Similar to POS, primary object discovery (POD) [10, 15, 34] also attempts to identify the locations of a primary object in a video sequence. However, it locates the primary object with bounding boxes, instead of pixel-wise delineation. POD algorithms also use saliency maps [15, 34] or object proposals [10]. Luo *et al*. [15] and Yang *et al*. [34] generate candidate boxes based on saliency scores in each frame, and then find the optimal path maximizing the sum of the saliency scores. Yang *et al*. [34] employ six kinds of saliency maps to overcome the limitations of individual saliency cues. Koh *et al*. [10] discover a primary object by combining object proposals based on the evolutionary primary object model.

3. Proposed Algorithm

We segment a primary object in a sequence of video frames $\mathcal{I} = \{I^{(1)}, \dots, I^{(T)}\}$, based on the assumption that the primary object appears in most frames. The output is a set of pixel-wise binary maps to delineate the primary object in the corresponding frames.

Figure 1 shows an overview of the proposed algorithm. First, we generate a pool of candidate regions for each frame. Second, we select initial primary object regions,

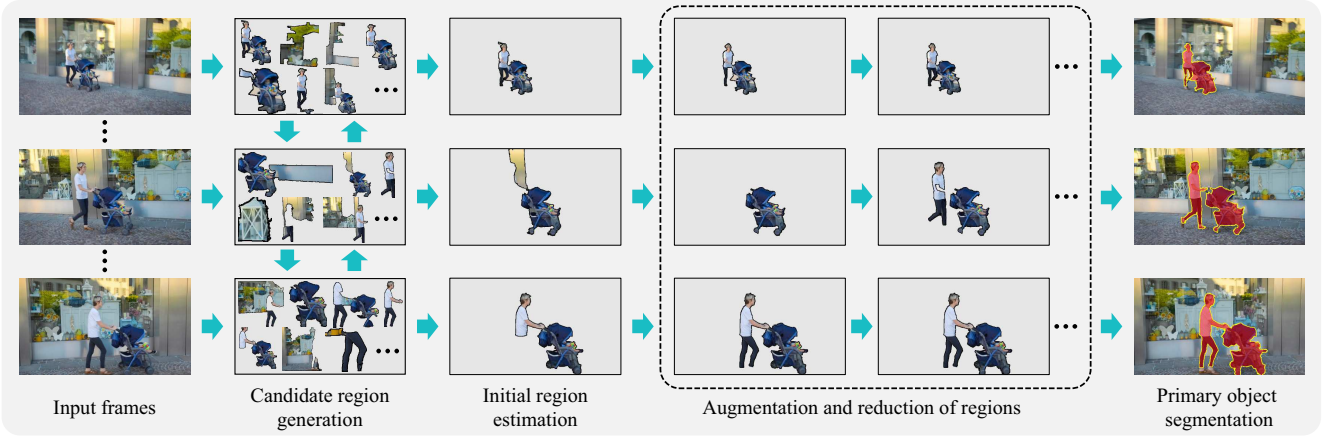


Figure 1. An overview of the proposed algorithm.

by exploiting the recurrence property of a primary object. Third, we refine the initial regions, by augmenting and reducing those regions progressively.

3.1. Generating Candidate Regions

Candidate Regions: After over-segmenting each frame into superpixels, we generate a pool of candidate regions by merging neighboring superpixels recursively [27, 29].

For the over-segmentation, we extract two ultrametric contour maps (UCMs) [1] for each frame. The original UCM method [1] generates a contour map using color edges. In this work, we extract another UCM using motion edges as well. The motion edges are obtained by the learning-based detector [31] using optical flow data [8]. Figure 2 illustrates these UCMs. Contours are more concentrated around the moving car in the motion-based UCM in Figure 2(c) than they are in the color-based UCM in Figure 2(b). Each region, delineated by a closed boundary in a UCM, becomes a superpixel. Thus, we have color-based superpixels and motion-based superpixels, which are shown in Figures 2(e) and (f), respectively. We initialize the set of candidate regions $\mathcal{Q}^{(t)}$ for frame $I^{(t)}$, by gathering all these superpixels.

Note that each boundary, shared by neighboring superpixels, is associated with the boundary strength in the UCM method. We recursively merge neighboring superpixels according to their boundary strengths, and include the merged superpixel into $\mathcal{Q}^{(t)}$. More specifically, let us consider the color-based superpixels first. We determine the pair of superpixels, s_m and s_n, which share the weakest boundary. Then, to improve the diversity of candidate regions in $\mathcal{Q}^{(t)}$, we generate additional candidate regions as follows:

- We put the union of superpixels $s_m \cup s_l$ into $\mathcal{Q}^{(t)}$ for each $s_l \in \mathcal{N}_m$, where $\mathcal{N}_m$ denotes the set of the neighboring superpixels of s_m.
- Similarly, we put the union of superpixels $s_n \cup s_{l'}$ into $\mathcal{Q}^{(t)}$ for each $s_{l'} \in \mathcal{N}_n$.

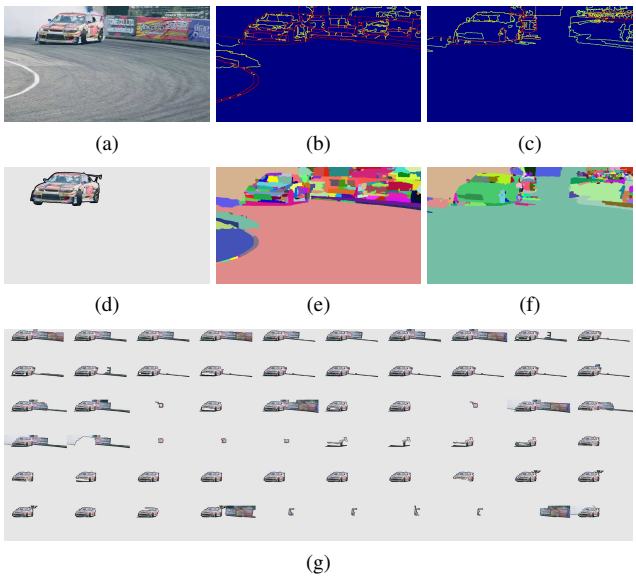


Figure 2. Candidate region generation for frame 26 in the "Drift-turn" sequence: (a) input frame $I^{(t)}$, (b) color-based UCM, (c) motion-based UCM, (d) the ground-truth, (e) color-based superpixels, (f) motion-based superpixels, and (g) the set of candidate regions $\mathcal{Q}^{(t)}$.

We then merge s_m and s_n into a single superpixel $s_m \cup s_n$. After the merging, we select the pair of superpixels with the weakest boundary and repeat the process. This recursive merging terminates, when all superpixels are merged into a single cluster. Next, using the motion-based superpixels, we perform the same process to further expand the set $\mathcal{Q}^{(t)}$ of candidate regions.

Foreground Confidence: After expanding $\mathcal{Q}^{(t)}$, we measure the foreground confidence $c_i^{(t)}$ of each candidate region $q_i^{(t)}$ in $\mathcal{Q}^{(t)}$, which is defined as

$$c_i^{(t)} = \phi_i^{(t)} + \psi_i^{(t)} \tag{1}$$

where $\phi_i^{(t)}$ and $\psi_i^{(t)}$ are the appearance confidence and the edge confidence, respectively.

To determine the appearance confidence $\phi_i^{(t)}$, we obtain a saliency map for frame $I^{(t)}$ using the preprocessing technique in [9]. Based on the boundary prior, [9] estimates the initial foreground distribution, which we regard as the saliency map. We then compute $\phi_i^{(t)}$ by averaging the saliency values within the candidate region $q_i^{(t)}$.

Also, we determine the edge confidence $\psi_i^{(t)}$, based on the color-based edge score map $E_{\mathrm{c}}^{(t)}$ [4] and the motion-based edge score map $E_{\mathrm{m}}^{(t)}$ [31], which were used to generate the UCMs above. Then, $\psi_i^{(t)}$ is given by

$$\psi_i^{(t)} = \frac{1}{\sqrt{|\mathcal{B}_i|}} \sum_{\mathbf{x} \in \mathcal{B}_i} \left(\beta_{\mathrm{c}} E_{\mathrm{c}}^{(t)}(\mathbf{x}) + \beta_{\mathrm{m}} E_{\mathrm{m}}^{(t)}(\mathbf{x}) \right) \quad (2)$$

where $\mathcal{B}_i$ is the set of the boundary pixels of the region $q_i^{(t)}$. As in [11, 12], we assume that an edge score map is more reliable if its scores are distributed more compactly. Thus, we set the weighting parameters β_{c} and β_{m} in (2) adaptively according to the corresponding spatial variances. Specifically, the spatial variance v_{c} of the color edge map $E_{\mathrm{c}}^{(t)}$ is

$$v_{\mathrm{c}} = \frac{\sum_{\mathbf{x}} \|\mathbf{x} - \mu_{\mathrm{c}}\|^2 \times E_{\mathrm{c}}^{(t)}(\mathbf{x})}{\sum_{\mathbf{x}} E_{\mathrm{c}}^{(t)}(\mathbf{x})} \quad (3)$$

where the summation is over all pixels $\mathbf{x}$ in the map, and μ_{c} is the centroid given by

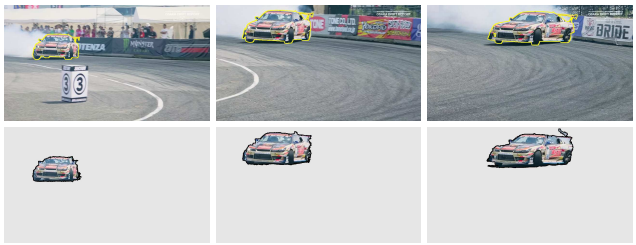
$$\mu_{\mathrm{c}} = \frac{\sum_{\mathbf{x}} \mathbf{x} \times E_{\mathrm{c}}^{(t)}(\mathbf{x})}{\sum_{\mathbf{x}} E_{\mathrm{c}}^{(t)}(\mathbf{x})}. \quad (4)$$

We also compute the variance v_{m} of the motion edge map $E_{\mathrm{m}}^{(t)}$ similarly. Then, we set β_{c} and β_{m} to be inversely proportional to the corresponding variances;

$$\beta_{\mathrm{c}} = \frac{v_{\mathrm{m}}}{v_{\mathrm{c}} + v_{\mathrm{m}}}, \quad \beta_{\mathrm{m}} = \frac{v_{\mathrm{c}}}{v_{\mathrm{c}} + v_{\mathrm{m}}}. \quad (5)$$

Next, we rank the candidate regions in $\mathcal{Q}^{(t)}$ according to their foreground confidence levels. We select the top 20 candidate regions and discard the other ones. To boost the recall rate of the primary object for frame $I^{(t)}$, we also warp the selected candidate regions at $I^{(t-1)}$ and $I^{(t+1)}$ to $I^{(t)}$ using pixel-wise optical flow vectors [8], respectively. We then rearrange $\mathcal{Q}^{(t)} = \{q_1^{(t)}, q_2^{(t)}, \ldots, q_N^{(t)}\}$ so that it consists of the top 20 candidate regions in $I^{(t)}$ and the 40 warped regions from $I^{(t-1)}$ or $I^{(t+1)}$. Thus, $N = 60$. Also, we define the confidence vector $\mathbf{c}^{(t)}$, whose ith element is the foreground confidence $c_i^{(t)}$ of $q_i^{(t)}$ in $\mathcal{Q}^{(t)}$.

Figure 2(g) shows the candidate regions in $\mathcal{Q}^{(t)}$, sorted in the raster scan order according to the foreground confidence levels. We see that many candidate regions contain both the car (*i.e.* primary object) and the signboard. If we rely on the foreground confidence levels only, both objects can be regarded as primary due to their distinctive colors and dominant motions. Therefore, we should also exploit the recurrence property of the primary object to separate it accurately, as will be discussed in Section 3.2.



(a) frame 16 (b) frame 26 (c) frame 36

Figure 3. The initial primary object regions for frames 16, 26, and 36 in the "Drift-turn" sequence. In the top row, the ground-truth boundaries of the primary object are depicted in yellow.

Feature Description: We describe the feature $\mathbf{f}_i^{(t)}$ of each candidate region $q_i^{(t)}$ in $\mathcal{Q}^{(t)}$ using the bag-of-visual-words approach [6]. Given the video sequence $\mathcal{I}$, we quantize the average LAB colors of all superpixels into 100 codewords, and associate each pixel with the nearest codeword. We then construct the histogram of the codewords for the pixels within $q_i^{(t)}$, and normalize it into the feature vector $\mathbf{f}_i^{(t)}$.

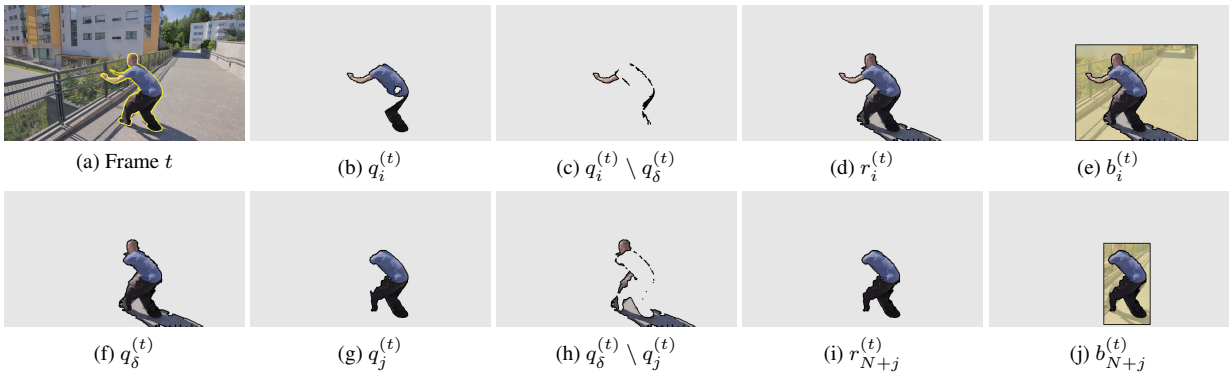
3.2. Selecting Initial Primary Object Regions

Among the candidate regions in $\mathcal{Q}^{(t)}$, we choose the main region $q_\delta^{(t)}$ and regard it as the initial primary object region in frame $I^{(t)}$. Noisy environments, such as background clutters and non-primary objects, make it difficult to decide the main region. To overcome this issue, we exploit the recurrence property that a primary object appears repeatedly in a video sequence. In other words, we decide the main region by finding recurring candidate regions in the sequence.

The main region is discovered in the feature space. Based on the recurrence property, we assume that the feature of the main region $q_\delta^{(t)}$ in $I^{(t)}$ should be similar to those of the main regions in the other frames. Thus, we obtain the index δ of $q_\delta^{(t)}$ by

$$\delta = \arg \min_{i: q_i^{(t)} \in \mathcal{Q}^{(t)}} \sum_{\tau=1, \tau \neq t}^{T} d_\chi(\mathbf{f}_i^{(t)}, \mathbf{p}^{(\tau)}) \quad (6)$$

where $\mathbf{p}^{(\tau)}$ denotes the feature vector of the main region in frame $I^{(\tau)}$, and the chi-square distance d_χ is adopted to compare two histograms. Without any prior information, at the beginning, we set the feature $\mathbf{p}^{(\tau)}$ by superposing the



(a) Frame t (b) $q_i^{(t)}$ (c) $q_i^{(t)} \setminus q_\delta^{(t)}$ (d) $r_i^{(t)}$ (e) $b_i^{(t)}$

(f) $q_\delta^{(t)}$ (g) $q_j^{(t)}$ (h) $q_\delta^{(t)} \setminus q_j^{(t)}$ (i) $r_{N+j}^{(t)}$ (j) $b_{N+j}^{(t)}$

Figure 4. Augmentation and reduction of the initial primary object region $q_\delta^{(t)}$ in frame 57 in the "Parkour" sequence.

features of all candidate regions in $\mathcal{Q}^{(\tau)}$. To take into account the foreground confidence levels, we combine these features of the candidate regions using the confidence vector $\mathbf{c}^{(\tau)} = [c_1^{(\tau)}, \ldots, c_N^{(\tau)}]^T$ in Section 3.1. Specifically, we can write the feature vector $\mathbf{p}^{(\tau)}$ as

$$\mathbf{p}^{(\tau)} = \mathbf{F}^{(\tau)}\mathbf{c}^{(\tau)} \tag{7}$$

where $\mathbf{F}^{(\tau)} = [\mathbf{f}_1^{(\tau)}, \ldots, \mathbf{f}_N^{(\tau)}]$ is the matrix whose ith column is the feature vector of the ith candidate region in $\mathcal{Q}^{(\tau)}$.

By applying $\mathbf{p}^{(\tau)}$ in (7) to (6), we obtain the main region $q_\delta^{(t)}$ for each frame $I^{(t)}$. After obtaining the main regions for all frames, we update them as follows:

1. We update the feature $\mathbf{p}^{(t)}$ for each frame $I^{(t)}$ by $\mathbf{p}^{(t)} \leftarrow \mathbf{f}_\delta^{(t)}$.

2. Using the updated features of the main regions, we choose the main region $q_\delta^{(t)}$ for each frame $I^{(t)}$ via (6).

We repeat these two steps alternately until the features of the main regions are unchanged. Consequently, we obtain the initial primary object regions for all frames.

Figure 3 shows the initial primary object regions for three frames in the "Drift-turn" sequence. In particular, in Figure 3(b), we see that the initial region for the car is well selected from the 60 candidate regions in Figure 2(g).

3.3. Refining Primary Object Regions

Initial regions in Section 3.2 roughly delineate primary objects. They, however, may exclude parts of primary objects or include noisy regions (background or other objects), as shown in Figure 4(f). Hence, we attempt to refine the initial regions, by augmenting them with missing regions and reducing them by removing noisy regions.

Augmented and Reduced Regions: For each frame $I^{(t)}$, we have the initial estimate $q_\delta^{(t)}$ of the primary object region. By augmenting the candidate regions in $\mathcal{Q}^{(t)} = \{q_1^{(t)}, \ldots, q_N^{(t)}\}$, we obtain $\mathcal{R}_{\text{aug}}^{(t)} = \{r_1^{(t)}, \ldots, r_N^{(t)}\}$, whose ith element is given by

$$r_i^{(t)} = q_\delta^{(t)} \cup q_i^{(t)} = q_\delta^{(t)} \cup (q_i^{(t)} \setminus q_\delta^{(t)}). \tag{8}$$

Figure 4(d) illustrates the augmented region $r_i^{(t)}$, which is the union of the original candidate $q_\delta^{(t)}$ in Figure 4(f) and a possibly missing region $q_i^{(t)} \setminus q_\delta^{(t)}$ in Figure 4(c).

We also reduce the candidate regions in $\mathcal{Q}^{(t)} = \{q_1^{(t)}, \ldots, q_N^{(t)}\}$ to obtain $\mathcal{R}_{\text{red}}^{(t)} = \{r_{N+1}^{(t)}, \ldots, r_{2N}^{(t)}\}$, whose element $r_{N+j}^{(t)}$ is given by

$$r_{N+j}^{(t)} = q_\delta^{(t)} \cap q_j^{(t)} = q_\delta^{(t)} \setminus (q_\delta^{(t)} \setminus q_j^{(t)}). \tag{9}$$

Figure 4(i) shows the reduced region $r_{N+j}^{(t)}$. The set difference $q_\delta^{(t)} \setminus q_j^{(t)}$ in Figure 4(h) contains background parts. By subtracting $q_\delta^{(t)} \setminus q_j^{(t)}$ from the original candidate $q_\delta^{(t)}$ in Figure 4(f), we obtain the background-free region $r_{N+j}^{(t)}$.

By combining $\mathcal{R}_{\text{aug}}^{(t)}$ and $\mathcal{R}_{\text{red}}^{(t)}$, we form the set of refined regions

$$\begin{aligned}\mathcal{R}^{(t)} &= \mathcal{R}_{\text{aug}}^{(t)} \cup \mathcal{R}_{\text{red}}^{(t)} \cup \{q_\delta^{(t)}\} \\ &= \{r_1^{(t)}, \ldots, r_N^{(t)}, r_{N+1}^{(t)}, \ldots, r_{2N}^{(t)}, r_{2N+1}^{(t)}\}\end{aligned} \tag{10}$$

where $r_{2N+1}^{(t)} = q_\delta^{(t)}$. For each refined region $r_i^{(t)}$, we define a background region $b_i^{(t)}$. Specifically, as shown in Figure 4(e), we place a bounding box surrounding $r_i^{(t)}$ with a margin, and exclude $r_i^{(t)}$ from the bounding box to obtain the background region $b_i^{(t)}$, which is depicted in yellow. We extract the features $\mathbf{f}_{\text{r},i}^{(t)}$ and $\mathbf{f}_{\text{b},i}^{(t)}$ of $r_i^{(t)}$ and $b_i^{(t)}$, respectively, by employing the feature description scheme in Section 3.1.

Primary Object Regions: To determine whether to augment or reduce $q_\delta^{(t)}$ in order to delineate the primary object, we define a cost function,

$$C(r_i^{(t)}) = C_{\text{data}}(r_i^{(t)}) + \gamma \cdot C_{\text{seg}}(r_i^{(t)}), \tag{11}$$

(a) Input frames (b) Initial regions (c) 1st iteration (d) 2nd iteration (e) 3rd iteration (f) Convergence

Figure 5. Evolution of refined regions in the iterative augmentation and reduction process (ARP). From top to bottom, "Parkour," "Motocross-jump," "Mallard-water," "Libby," and "Stroller" sequences. As the iteration goes on, refined regions represent the primary objects more accurately. In (a), the ground-truth boundaries of the primary objects are depicted in yellow.

where C_{data} and C_{seg} are the data and segmentation costs, respectively, and γ is an adaptive weight to balance the influence of the two terms.

The data cost $C_{\text{data}}(r_i^{(t)})$ in (11) constrains that the refined (*i.e.* augmented or reduced) region $r_i^{(t)}$ should be similar to the initial primary object regions in all frames. More specifically, it is defined as

$$C_{\text{data}}(r_i^{(t)}) = \frac{1}{T}\sum_{\tau=1}^{T} d_\chi(\mathbf{f}_{\text{r},i}^{(t)}, \mathbf{f}_\delta^{(\tau)}), \tag{12}$$

where $\mathbf{f}_\delta^{(\tau)}$ is the feature vector of the initial primary object region in frame $I^{(\tau)}$. On the other hand, the segmentation cost $C_{\text{seg}}(r_i^{(t)})$ is defined, based on the dissimilarity between the region $r_i^{(t)}$ and its background $b_i^{(t)}$, as

$$C_{\text{seg}}(r_i^{(t)}) = -d_\chi(\mathbf{f}_{\text{r},i}^{(t)}, \mathbf{f}_{\text{b},i}^{(t)}). \tag{13}$$

Notice that, because of the minus sign in (13), the minimization of $C_{\text{seg}}(r_i^{(t)})$ makes the region as dissimilar from its background as possible.

We minimize the cost function $C(r_i^{(t)})$ in (11) to select the optimal refined region $r_*^{(t)}$ from $\mathcal{R}^{(t)}$ in (10),

$$r_*^{(t)} = \arg\min_{r_i^{(t)} \in \mathcal{R}^{(t)}} C(r_i^{(t)}). \tag{14}$$

Note that the region is augmented if $r_*^{(t)} \in \mathcal{R}_{\text{aug}}^{(t)}$, while it is reduced if $r_*^{(t)} \in \mathcal{R}_{\text{red}}^{(t)}$.

We perform this augmentation and reduction process (ARP) iteratively.

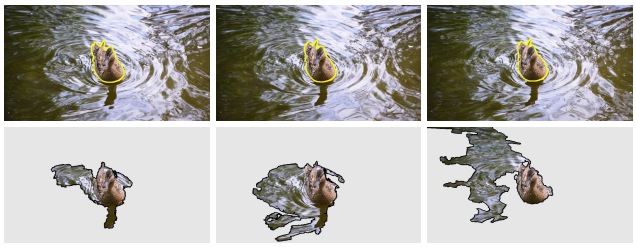
1. By employing $r_*^{(t)}$ as the initial region $q_\delta^{(t)}$ (*i.e.*, $q_\delta^{(t)} \leftarrow r_*^{(t)}$), we construct again the set of augmented or reduced regions, $\mathcal{R}^{(t)}$ in (10).

2. Then, we find the optimal $r_*^{(t)}$ again by minimizing $C(r_i^{(t)})$ in (14).

This is repeated until $r_*^{(t)}$ is unchanged. This refinement process is theoretically guaranteed to converge, since the cost function in (11) monotonically decreases at each iteration, and the candidate set $\mathcal{R}^{(t)}$ in (10) includes the optimal solution $q_\delta^{(t)}$ in the last iteration. Figure 5 illustrates how the optimal refined region $r_*^{(t)}$ evolves as the iteration goes on. We see that the initial regions in Figure 5(b) are augmented with missing regions or reduced by excluding noisy regions. Eventually, we obtain the faithful segmentation results in Figure 5(f). Even a disconnected part of a primary object is augmented in the "Libby' sequence.

Finally, after ARP converges, the proposed algorithm yields the set $\mathcal{R}^*$ of the primary object regions for all frames as output,

$$\mathcal{R}^* = \{r_*^{(1)}, r_*^{(2)}, \ldots, r_*^{(T)}\}. \tag{15}$$



(a) frame 62 (b) frame 63 (c) frame 64

Figure 6. Temporally inconsistent initial regions in the "Mallard-water" sequence. In the top row, the ground-truth object boundaries are depicted in yellow. The bottom row depicts the initial primary object regions.

Notice that a higher weight γ in (11) allows refined regions to be more different from initial primary object regions. Before the iterative ARP, we decide γ for an input video sequence, by analyzing the temporal consistency of initial primary object regions in different frames. When the video sequence yields inconsistent initial regions, as exemplified in Figure 6, those initial regions should be significantly modified with a high γ to provide satisfactory segmentation results. In contrast, consistent initial regions need be modified only slightly with a low γ. To quantify the temporal consistency, we obtain a warped region $q_\delta^{(t,t-1)}$ by mapping the pixels within $q_\delta^{(t-1)}$ in the previous frame $I^{(t-1)}$ to the current frame $I^{(t)}$ using optical flow vectors. Then, we measure the intersection over union (IoU) ratio between $q_\delta^{(t)}$ and $q_\delta^{(t,t-1)}$. We obtain the average λ of these IoU ratios over all frames, and then set γ as

$$\gamma = \exp\left(-\frac{\lambda}{\sigma^2}\right), \tag{16}$$

where $\sigma^2 = 0.6$. Consequently, we set γ to represent the overall inconsistency of the initial primary object regions.

4. Experimental Results

We compare the proposed algorithm with the conventional algorithms on the DAVIS dataset [21] and the FBMS dataset [2]. We use the same parameters in all experiments.

4.1. Evaluation on the DAVIS Dataset

The DAVIS dataset is a recent benchmark for evaluating VOS algorithms. It consists of 50 video sequences with 3,455 annotated frames. These sequences are challenging due to appearance change, fast-motion, occlusion, and so forth. Each sequence contains either a single object or two spatially connected objects, *e.g.* a horse and its rider, which appear repeatedly in the sequence. We also regard such connected objects as a single primary object.

For the assessment of segmentation results, we measure the region similarity $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ in [21].

Table 1. Comparison of the conventional GMM-based refinement techniques [20, 36] and the proposed augmentation and reduction process (ARP). 'IR' means that initial primary object regions in Section 3.2 are used as segmentation results.

Measure		IR	IR+ [20]	IR+ [36]	IR+ARP
$\mathcal{J}$	Mean	0.719	0.580	0.670	**0.763**
	Recall	0.855	0.665	0.810	**0.892**
$\mathcal{F}$	Mean	0.680	0.523	0.613	**0.711**
	Recall	0.802	0.541	0.740	**0.828**

The region similarity $\mathcal{J}$ is defined as the IoU ratio $\mathcal{J} = \frac{|\mathcal{S}_\mathrm{p} \cap \mathcal{S}_\mathrm{gt}|}{|\mathcal{S}_\mathrm{p} \cup \mathcal{S}_\mathrm{gt}|}$, where $\mathcal{S}_\mathrm{p}$ and $\mathcal{S}_\mathrm{gt}$ are an estimated segment and the ground-truth, respectively. Also, the contour accuracy $\mathcal{F}$ computes the F-measure that is the harmonic mean of the contour precision and recall rates.

Impacts of ARP: We analyze the impacts of the proposed ARP refinement in Section 3.3. Note that ARP refines initial primary object regions, by augmenting them with missing parts or reducing them by excluding noisy parts. We compare ARP with the conventional refinement techniques [20, 36]. They determine the class label (*i.e.* foreground or background) of each pixel or superpixel using the foreground and background GMMs [24], which are constructed from initial regions. Table 1 compares the $\mathcal{J}$ and $\mathcal{F}$ scores of the initial regions (IR) in Section 3.2 and the refined results of these initial regions, obtained by the conventional techniques and the proposed ARP. In Table 1, 'Mean' denotes the average score, while 'Recall' measures the proportion of the frames whose scores are larger than 0.5. We see that the conventional techniques [20, 36] rather degrade the VOS performance. This is because the GMMs are constructed from incomplete initial regions, unlike manual annotations. In such cases, the GMMs cannot model temporally varying objects and their background information reliably. In contrast, the proposed ARP improves the VOS performance significantly.

Quantitative Comparison: Table 2 compares the proposed algorithm with the conventional semi-supervised VOS [3, 17, 22, 23], multiple VOS [2, 7, 13], and POS [5, 9, 20, 30, 36] algorithms. We obtain the results of the conventional algorithms from the DAVIS dataset website [21], except for [9, 36]. For [9, 36], we compute the results using the source codes, provided by the respective authors.

In terms of the region similarity $\mathcal{J}$, the proposed algorithm outperforms all conventional algorithms significantly. For example, the proposed algorithm yields 0.122 and 0.098 higher 'Mean' $\mathcal{J}$ score than the state-of-the-art POS [5] and semi-supervised VOS [17] algorithms, respectively. In terms of the contour similarity $\mathcal{F}$, the proposed algorithm also provides much better performance. It is worth pointing out that the proposed algorithms even surpasses the semi-supervised and multiple VOS algorithms, even though the proposed algorithm does not require any manual annota-

Table 2. Comparison of the proposed algorithm with the conventional algorithms on the DAVIS dataset in terms of the region similarity $\mathcal{J}$ and the contour similarity $\mathcal{F}$.

Measure		Semi-supervised VOS				Multiple VOS			POS					
		[3]	[23]	[22]	[17]	[2]	[7]	[13]	[20]	[36]	[30]	[5]	[9]	Proposed
$\mathcal{J}$	Mean	0.358	0.556	0.631	0.665	0.543	0.501	0.569	0.575	0.466	0.426	0.641	0.531	**0.763**
	Recall	0.388	0.606	0.778	0.764	0.636	0.560	0.671	0.652	0.467	0.386	0.731	0.611	**0.892**
$\mathcal{F}$	Mean	0.346	0.533	0.546	0.656	0.525	0.478	0.503	0.536	0.445	0.383	0.593	0.504	**0.711**
	Recall	0.329	0.559	0.604	0.774	0.613	0.519	0.534	0.579	0.421	0.264	0.658	0.558	**0.828**

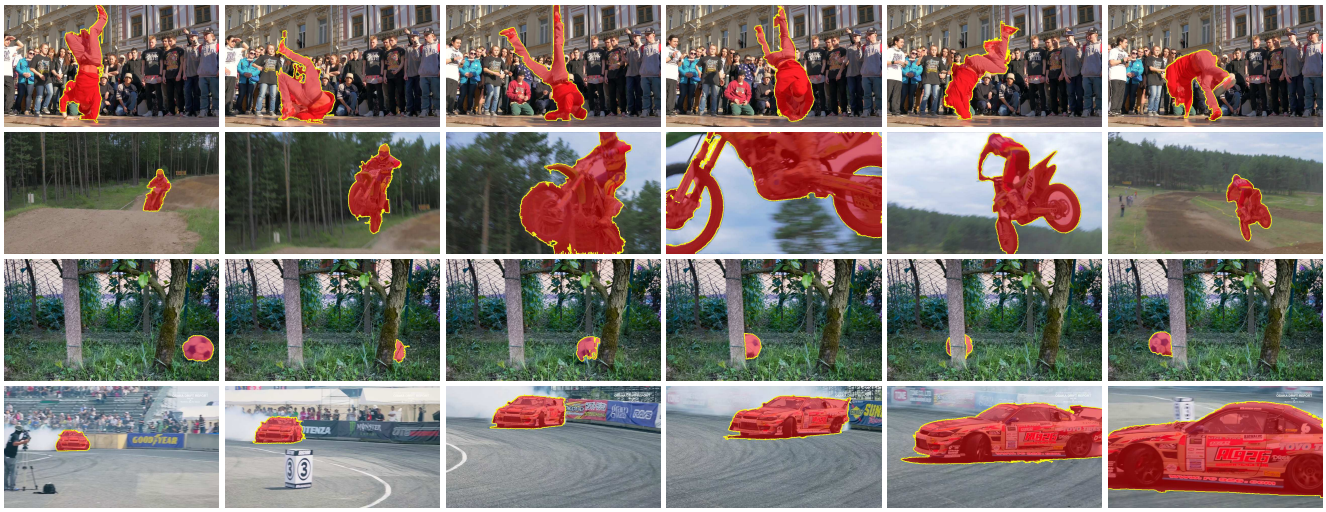


Figure 7. Primary object segmentation results of the proposed algorithm on the DAVIS dataset: "Breakdance," "Motocross-jump," "Soccerball" and "Drift-run" sequences from top to bottom. Segmentation regions and boundaries are depicted in red and yellow, respectively.

Table 3. Comparison of IoU scores on the test sequences in the FBMS dataset.

Video	[20]	[36]	[5]	[9]	Proposed
Average	0.555	0.473	0.445	0.542	**0.598**

tions or ground-truth to identify primary objects.

Qualitative Results: Figure 7 shows examples of POS results on the DAVIS dataset. We see that the proposed algorithm yields accurate segment tracks for primary objects, even though those objects suffer from appearance deformation ("Breakdance"), fast motion ("Motocross-jump"), and occlusion ("Soccerball"). Furthermore, the proposed algorithm can deal with fast camera motion in the "Drift-run" sequence.

4.2. Evaluation on the FBMS Dataset

The FBMS dataset [2] is another benchmark for VOS. It consists of 59 video sequences, which are divided into 29 training and 30 test video sequences. We assess the POS algorithms using the test set. We obtain the results of the conventional POS algorithms [5, 9, 20, 36] using the source codes, provided by the respective authors. Table 3 lists the average IoU scores on the test set. As compared with the conventional POS algorithms [20], [36], [5], and [9], the proposed algorithms improves the average IoU by 0.043, 0.125, 0.153, and 0.056, respectively. Due to the page limitation, we provide more experimental results in the supplemental materials.

5. Conclusions

We proposed a novel POS algorithm based on ARP. We first generated candidate regions for each frame using color and motion edges. We then estimated initial regions for the primary object, based on the recurrence property of the primary object. Finally, we adopted the iterative ARP to refine the initial regions and delineate the primary object in each frame. Experimental results demonstrated that the proposed algorithm efficiently segments primary objects and significantly outperforms the state-of-the-art semi-supervised, multiple, and unsupervised VOS algorithms on the DAVIS and FBMS datasets.

Acknowledgements

This work was supported partly by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. NRF-2015R1A2A1A10055037), and partly by the Agency for Defense Development (ADD) and Defense Acquisition Program Administration (DAPA) of Korea (UC160016FD).

References

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011. 3

[2] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. 2010. 1, 2, 7, 8

[3] J. Chang, D. Wei, and J. W. Fisher III. A video representation using temporal superpixels. In *CVPR*, pages 2051–2058, 2013. 7, 8

[4] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, pages 1841–1848, 2013. 4

[5] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 1, 2, 7, 8

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005. 4

[7] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, pages 1846–1853, 2012. 1, 2, 7, 8

[8] Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *CVPR*, pages 5704–5712, 2016. 3, 4

[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*, pages 696–704, 2016. 1, 2, 4, 7, 8

[10] Y. J. Koh, W.-D. Jang, and C.-S. Kim. POD: Discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *CVPR*, pages 4268–4276, 2016. 2

[11] Y. J. Koh, C. Lee, and C.-S. Kim. Video stabilization based on feature trajectory augmentation and selection and robust mesh grid warping. *IEEE Trans. Image Process.*, 24(12):5260–5273, 2015. 4

[12] S.-H. Lee, J.-W. Kang, and C.-S. Kim. Compressed domain video saliency detection using global and local spatiotemporal features. *J. Vis. Commun. Image R.*, 35:169–183, 2016. 4

[13] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 1, 2, 7, 8

[14] F. Li, T. Kim, A. Humayun, D. Tsai, and J. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 1, 2

[15] Y. Luo, J. Yuan, and J. Lu. Finding spatio-temporal salient paths for video objects discovery. *J. Vis. Commun. Image R.*, 38:45–54, 2016. 2

[16] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012. 1, 2

[17] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, pages 743–751, 2016. 1, 2, 7, 8

[18] P. Ochs and T. Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, pages 1583–1590, 2011. 1, 2

[19] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, pages 614–621, 2012. 1, 2

[20] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, pages 1777–1784, 2013. 1, 2, 7, 8

[21] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 1, 7

[22] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, pages 3227–3234, 2015. 1, 2, 7, 8

[23] S. A. Ramakanth and R. V. Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, pages 376–383, 2014. 1, 2, 7, 8

[24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Trans. Graphics*, volume 23, pages 309–314, 2004. 1, 7

[25] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998. 1, 2

[26] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, pages 3899–3908, 2016. 1, 2

[27] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *Int. J. Comput. Vis.*, 104(2):154–171, 2013. 3

[28] D. Varas and F. Marques. Region-based particle filter for video object segmentation. In *CVPR*, pages 3470–3477, 2014. 1, 2

[29] C. Wang, L. Zhao, S. Liang, L. Zhang, J. Jia, and Y. Wei. Object proposal by multi-branch hierarchical segmentation. In *CVPR*, pages 3873–3881, 2015. 3

[30] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, pages 3395–3402, 2015. 1, 2, 7, 8

[31] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Learning to detect motion boundaries. In *CVPR*, pages 2578–2586, 2015. 3, 4

[32] F. Xiao and Y. J. Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, pages 933–942, 2016. 1, 2

[33] 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. 1, 2

[34] J. Yang, G. Zhao, J. Yuan, X. Shen, Z. Lin, B. Price, and J. Brandt. Discovering primary objects in videos by saliency fusion and iterative appearance estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 26(6):1070–1083, 2016. 2

[35] Y. Yang, G. Sundaramoorthi, and S. Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *ICCV*, pages 4408–4416, 2015. 1, 2

[36] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, pages 628–635, 2013. 1, 2, 7, 8