Primary Object Segmentation in Videos via Alternate Convex Optimization of Foreground and Background Distributions

Won-Dong Jang
Korea University
wdjang@mcl.korea.ac.kr

Chulwoo Lee
Northumbria University
chulwoo.lee@northumbria.ac.uk

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

Abstract

An unsupervised video object segmentation algorithm, which discovers a primary object in a video sequence automatically, is proposed in this work. We introduce three energies in terms of foreground and background probability distributions: Markov, spatiotemporal, and antagonistic energies. Then, we minimize a hybrid of the three energies to separate a primary object from its background. However, the hybrid energy is nonconvex. Therefore, we develop the alternate convex optimization (ACO) scheme, which decomposes the nonconvex optimization into two quadratic programs. Moreover, we propose the forward-backward strategy, which performs the segmentation sequentially from the first to the last frames and then vice versa, to exploit temporal correlations. Experimental results on extensive datasets demonstrate that the proposed ACO algorithm outperforms the state-of-the-art techniques significantly.

1. Introduction

Video object segmentation is the process to separate a primary object from the background in a video sequence. It is applicable as a preliminary to various vision applications, such as action recognition, content-based video retrieval, targeted content replacement, and video summarization. It is hence important to develop robust video object segmentation techniques. However, video object segmentation is challenging due to a variety of difficulties, *e.g.*, cluttered background, occlusion, and non-rigid object deformation. To overcome these issues, many attempts have been made.

Video object segmentation methods can be categorized into supervised or unsupervised approaches. Supervised methods [3, 11, 30, 33] address the problem by employing user annotations on a few selected frames. In contrast, unsupervised methods [25, 29, 34, 37] identify an object automatically. Without the prior information about the object, the unsupervised segmentation is more difficult than the supervised one.

In this work, we propose a novel unsupervised algorithm for video object segmentation. To segment a primary foreground object from the background, we define three energies in terms of foreground and background probability distributions: Markov energy, spatiotemporal energy, and antagonistic energy. Then, we minimize the hybrid energy of the three energy terms to achieve the segmentation. More specifically, since the hybrid energy is nonconvex, we develop the alternate convex optimization (ACO) scheme that converts the nonconvex problem into two quadratic programs. We perform the segmentation forwardly from the first to the last frames, and then backwardly from the last to the first frames. Experimental results demonstrate that the proposed ACO algorithm outperforms the state-of-the-art conventional algorithms in [29,37] on the SegTrack [33], SegTrack v2 [23], and VidSeg datasets. To summarize, this paper has three main contributions.

- Introduction of the hybrid energy of foreground and background distributions and its ACO to segment a primary object from the background.
- The forward-backward strategy, to transfer object information sequentially from the first frame to the last frame and vice versa, for accurate object segmentation.
- Remarkable performance achievement on the three datasets, including the proposed VidSeg dataset that consists of challenging sequences.

2. Related Work

Supervised methods for video object segmentation, requiring user annotations about a primary object, include non-rigid object tracking and interactive video segmentation. In non-rigid object tracking, a primary object is manually delineated at the first frame and then tracked at subsequent frames [11, 33]. In interactive video segmentation, user annotations on a few selected frames are utilized to separate an object from its background [3, 30]. However, the manual delineation or annotation is exhausting. Hence, in this work, we focus on unsupervised methods.

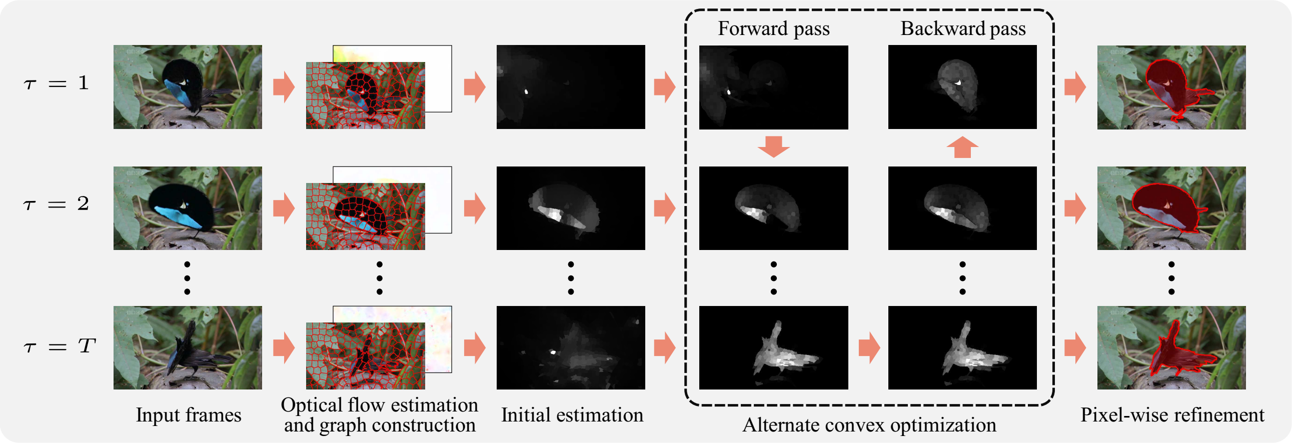


Figure 1. Overview of the proposed ACO algorithm. In the initial estimation and the alternate convex optimization, both foreground and background distributions are determined. However, only foreground distributions are shown here.

Unsupervised methods automatically extract an object from a video sequence. Brendel and Todorovic [7] proposed a bottom-up video over-segmentation algorithm, which determines the temporal connection between per-frame image segmentation results. Grundmann *et al.* [18] developed another video over-segmentation algorithm based on the graph-based optimization. Video over-segmentation is a versatile tool for various computer vision tasks, but it does not resolve the object-level segmentation problem.

To detect moving objects in a video sequence, Barnich and Droogenbroeck [5] proposed a background subtraction algorithm, which uses a background model based on interframe temporal consistencies. Also, Han and Davis [19] trained an SVM classifier, using multiple features, to build a background model. However, since background subtraction assumes fixed or slowly panning cameras, these algorithms [5, 19] are applicable to limited situations only.

Shi and Malik [31] constructed a graph for pixels in a video and then segmented motions using normalized cuts. By grouping long-term point trajectories, Brox and Malik [9] achieved object segmentation. Ochs and Brox [27] exploited sparse point trajectories to extract dense object regions. Ochs and Brox [28] also performed spectral clustering on point trajectories for object segmentation.

With advances in object proposal techniques [2, 10, 13], Lee *et al.* [21] first applied them to the video object segmentation task by ranking proposals in a video. Ma and Latecki [25] identified a primary object by determining maximum weight cliques on a series of object proposals. Zhang *et al.* [37] introduced a layered acyclic graph for object proposals and discovered an optimal path using dynamic programming. Banica *et al.* [4] constructed salient segment chains by performing matching between object proposals. Levinshtein *et al.* [22] applied the parametric maxflow to group spatiotemporal superpixels. Papazoglou and Ferrari [29] estimated motion-based inside-outside maps to delineate moving objects. Li *et al.* [23] generated multiple segment tubes by tracking many hypotheses.

Recently, Wang *et al.* [34] proposed a saliency-driven video object segmentation algorithm using geodesic distances. Giordano *et al.* [14] exploited a continuity of superpixels across consecutive frames to extract moving objects. Also, Taylor *et al.* [32] inferred long-term occlusions to discover objects in a video.

3. Proposed Algorithm

This section proposes a novel unsupervised video object segmentation algorithm, referred to as ACO. The input is a set of consecutive video frames $\{I^{(1)}, ..., I^{(T)}\}$, and the output is a set of pixel-wise binary label maps, discriminating a primary foreground object from its background.

Figure 1 shows an overview of the proposed ACO algorithm. First, we estimate initial probability distributions of the foreground and background by performing the manifold ranking processes with the boundary priors for each frame. Second, we optimize the foreground and background distributions, by employing the ACO, and then determine the label maps. This is done frame-by-frame, and the label map of a previous frame is exploited to obtain that of a current frame. More specifically, the forward-backward strategy is adopted, which obtains the label maps from the first to the last frames and then improves their accuracies from the last to the first frames. Finally, by refining the superpixel-level segmentation results, we obtain pixel-wise object segments.

3.1. Motion Estimation and Graph Construction

For each frame τ, we estimate the optical flows [24] from $I^{(\tau)}$ to $I^{(\tau-1)}$ and $I^{(\tau-2)}$, respectively. Also, we apply the SLIC algorithm [1] to over-segment each frame.

Let $G = (V, E)$ be a graph, where $V = \{x_1, \ldots, x_N\}$ is the set of nodes and $E = \{e_{ij}\}$ is the set of edges. The superpixels become nodes. Each edge e_{ij}, connecting x_i

and x_j, is assigned with weight (or affinity) w_{ij},

$$w_{ij} = \begin{cases} \exp\left(-\frac{d^2(x_i,x_j)}{\sigma^2}\right) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where d denotes a distance between x_i and x_j in a feature space, and σ^2 is a scale parameter.

We define the k-ring graph. In the k-ring graph, two nodes x_i and x_j are connected by an edge, if there is a sequence $(x_i = x_{g_1}, x_{g_2}, \ldots, x_{g_n} = x_j)$ for $n \leq k+1$ and every pair of consecutive nodes (or superpixels) in the sequence share a boundary. As k increases, the k-ring graph connects a node to a larger number of nodes.

3.2. Initial Probability Estimation

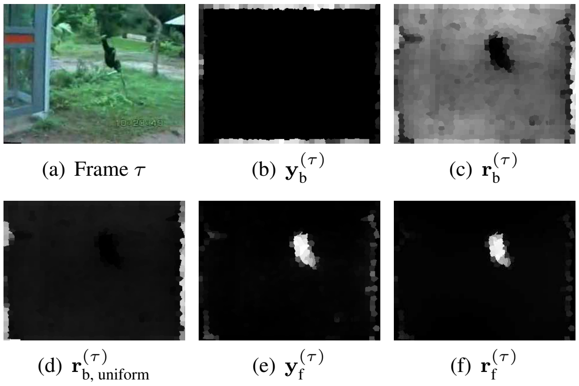
We generate initial probability distributions of the foreground and background, respectively. While a foreground object is likely to be near the center of a frame, boundary regions tend to belong to the background [35,36]. By adopting this boundary prior, we perform the manifold ranking algorithm [38]. Let us briefly review the manifold ranking. First, the affinity matrix $\mathbf{W} = [w_{ij}]$ is computed, and a query vector $\mathbf{y}$ is defined. For instance, $y_i = 1$ if node i is a query, and $y_i = 0$ otherwise. Then, the affinity matrix $\mathbf{W}$ is symmetrically normalized by $\mathbf{\Pi} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal matrix whose element d_{ii} equals the sum of the elements in the ith row of $\mathbf{W}$. Then, the ranking vector $\mathbf{r}$ is given by

$$\mathbf{r} = (\mathbf{I} - \alpha\mathbf{\Pi})^{-1}\mathbf{y} \quad (2)$$

where α is a parameter within $[0,1]$. The element r_i in $\mathbf{r}$ represents the ranking of the ith node toward the query node, *i.e.* the similarity of the ith node to the query node on the graph.

In this work, to determine initial foreground and background distributions at frame τ, we construct the 4-ring graph. We also compute each element $w_{ij}^{(\tau)}$ of the affinity matrix $\mathbf{W}^{(\tau)}$ using both the average LAB color difference and the average optical flow difference between x_i and x_j.

For the background distribution, we set a query vector $\mathbf{y}_{\text{b}}^{(\tau)}$ based on the boundary prior. Specifically, we set the query amount $y_{\text{b},j}^{(\tau)}$ at node j to be proportional to $1/\max_i\{\pi_{ij}^{(\tau)}\}$ if node j is at the image boundary, and 0 otherwise, as shown in Figure 2(b). $\pi_{ij}^{(\tau)}$ is an element of the normalized affinity matrix $\mathbf{\Pi}^{(\tau)}$. Thus, we assign a large query amount to a highly spreadable node, which is connected to many similar nodes. Then, by employing $\mathbf{y}_{\text{b}}^{(\tau)}$ as the query vector in (2), we obtain the ranking vector $\mathbf{r}_{\text{b}}^{(\tau)}$ in Figure 2(c), which is used as the initial background distribution. Notice that, if we assign query amounts uniformly to all boundary nodes, the background distribution $\mathbf{r}_{\text{b, uniform}}^{(\tau)}$ may be confined within the boundaries, as in Figure 2(d).


(a) Frame τ (b) $\mathbf{y}_{\text{b}}^{(\tau)}$ (c) $\mathbf{r}_{\text{b}}^{(\tau)}$

(d) $\mathbf{r}_{\text{b, uniform}}^{(\tau)}$ (e) $\mathbf{y}_{\text{f}}^{(\tau)}$ (f) $\mathbf{r}_{\text{f}}^{(\tau)}$

Figure 2. Computation of initial probability distributions: $\mathbf{r}_{\text{b}}^{(\tau)}$ in (c) and $\mathbf{r}_{\text{f}}^{(\tau)}$ in (f) are the initial background and foreground distributions, respectively, which are obtained using the query vectors $\mathbf{y}_{\text{b}}^{(\tau)}$ in (b) and $\mathbf{y}_{\text{f}}^{(\tau)}$ in (e). Note that $\mathbf{r}_{\text{b, uniform}}^{(\tau)}$ in (d) is the background distribution, when uniform query amounts are assigned to all boundary nodes.

On the other hand, for the foreground distribution, we convert the background distribution $\mathbf{r}_{\text{b}}^{(\tau)}$ into the foreground query vector $\mathbf{y}_{\text{f}}^{(\tau)}$ via $y_{\text{f},i}^{(\tau)} \propto \exp(-r_{\text{b},i}^{(\tau)})$. Then, we compute the corresponding ranking vector $\mathbf{r}_{\text{f}}^{(\tau)}$, which is the initial foreground distribution. In Figure 2(f), we see that $\mathbf{r}_{\text{f}}^{(\tau)}$ roughly indicates the shape of the foreground object.

3.3. Forward Pass of Video Object Segmentation

Next, we delineate a primary object in each frame, sequentially from the first to the last frames. We use the segmentation results of previous frames to process a current frame. For simplicity, let us describe the algorithm mainly in terms of the foreground distribution $\mathbf{p}_{\text{f}}^{(\tau)}$, in which $p_{\text{f},i}^{(\tau)}$ is the probability that the foreground object is found at node i at frame τ. The background distribution $\mathbf{p}_{\text{b}}^{(\tau)}$ is handled in a symmetrical manner.

For the video object segmentation, we define an energy function, composed of three terms: Markov energy, spatiotemporal energy, and antagonistic energy. Let us describe these three energy terms subsequently.

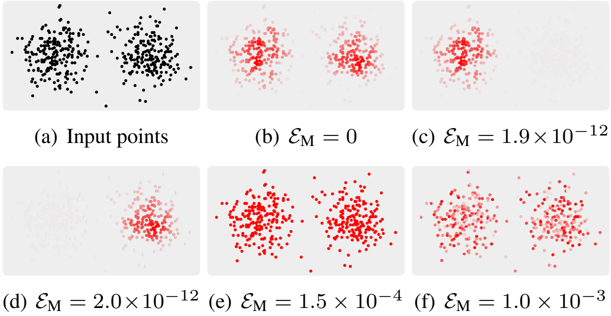
Markov Energy: The Markov random walk process simulates movements of an agent on a graph [12]. A movement is made with a higher probability, as the two nodes have more similar features. Many properties, including the stationary distribution of the agent, provide useful information for data clustering [8, 15, 20, 26].

An agent moves from node j to node i according to the transition probability

$$a_{ij} = w_{ij} / \textstyle\sum_{k=1}^{N} w_{kj}. \quad (3)$$

The movements of the agent are modeled by

$$\mathbf{p}^{(\theta+1)} = \mathbf{A}\mathbf{p}^{(\theta)} \quad (4)$$


(a) Input points (b) $\mathcal{E}_\mathrm{M} = 0$ (c) $\mathcal{E}_\mathrm{M} = 1.9 \times 10^{-12}$

(d) $\mathcal{E}_\mathrm{M} = 2.0 \times 10^{-12}$ (e) $\mathcal{E}_\mathrm{M} = 1.5 \times 10^{-4}$ (f) $\mathcal{E}_\mathrm{M} = 1.0 \times 10^{-3}$

Figure 3. (a) Input points, (b) stationary distribution, (c) left cluster, (d) right cluster, (e) uniform distribution, and (f) random distribution. Note that each distribution is normalized, and high probabilities are depicted by highly saturated red colors.

where θ is a time instance and $\mathbf{A} = [a_{ij}]$ is the transition matrix. The stationary distribution $\mathbf{p}^{(\infty)}$ can be found when the squared distance $||\mathbf{A}\mathbf{p}^{(\infty)} - \mathbf{p}^{(\infty)}||^2$ is minimized to 0. Figure 3(b) shows an example of the stationary distribution. We observe two desirable properties for clustering: First, it has high probabilities near the center of a cluster (*i.e.* a region of high point density) and low probabilities along the boundary of a cluster (*i.e.* a region of low point density). Second, nearby points have similar probabilities, thus facilitating the assignment of those points to the same cluster.

To enforce these two properties, we define the Markov energy $\mathcal{E}_\mathrm{M}$ as

$$\mathcal{E}_\mathrm{M}(\mathbf{p}_\mathrm{f}^{(\tau)}) = ||\mathbf{A}^{(\tau)}\mathbf{p}_\mathrm{f}^{(\tau)} - \mathbf{p}_\mathrm{f}^{(\tau)}||^2 \tag{5}$$

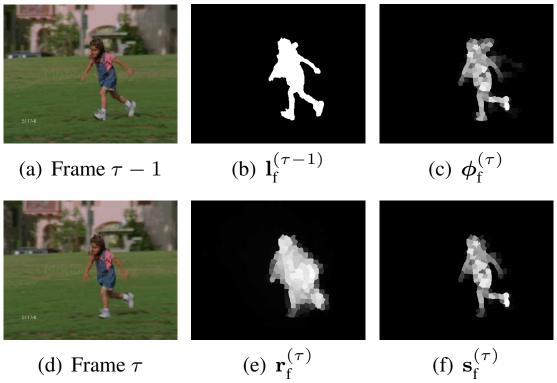
where $\mathbf{A}^{(\tau)}$ is the transition matrix at frame τ, derived from the affinity matrix $\mathbf{W}^{(\tau)}$. Notice that, when the agent moves within a single cluster in Figure 3(c) or (d), the Markov energy is also small and the aforementioned two properties are also satisfied. In contrast, the uniform and random distributions in Figures 3(e) and (f) yield much higher energies. In a feature space, the left and right clusters in Figures 3(c) and (d) may correspond to the foreground and background, respectively. We can make the foreground and background distributions form such separate clusters, by minimizing the spatiotemporal energy $\mathcal{E}_\mathrm{S}$ and the antagonistic energy $\mathcal{E}_\mathrm{A}$, in addition to the Markov energy $\mathcal{E}_\mathrm{M}$. Let us describe the additional energies $\mathcal{E}_\mathrm{S}$ and $\mathcal{E}_\mathrm{A}$ subsequently.

Spatiotemporal Energy: For each frame, the initial distribution $\mathbf{r}_\mathrm{f}^{(\tau)}$ in Section 3.2 provides a rough estimate of the foreground distribution. However, the per-frame estimate is insufficient for the video object segmentation due to its lack of temporal consistency. We hence combine the initial distribution with the segmentation results of previous frames to yield spatially accurate and temporally coherent segments.

At frame τ, we first obtain the temporal foreground confidence map $\boldsymbol{\phi}_\mathrm{f}^{(\tau)}$. To this end, we compute the propagation matrix $\mathbf{C}^{(\tau,\tilde{\tau})}$ that transfers the foreground labels at frame $\tilde{\tau}$ to frame τ. An element $c_{ij}^{(\tau,\tilde{\tau})}$ in $\mathbf{C}^{(\tau,\tilde{\tau})}$ is 1 if at least one pixel within node i at frame τ is matched to a pixel within node j at frame $\tilde{\tau}$ according to the optical flow, and 0 otherwise. Then, we transfer the segmentation results of the previous two frames to the current frame τ to generate the temporal confidence map $\boldsymbol{\phi}_\mathrm{f}^{(\tau)}$, which is given by

$$\boldsymbol{\phi}_\mathrm{f}^{(\tau)} = \mathbf{C}^{(\tau,\tau-1)}\mathbf{l}_\mathrm{f}^{(\tau-1)} + \mathbf{C}^{(\tau,\tau-2)}\mathbf{l}_\mathrm{f}^{(\tau-2)} \tag{6}$$

where $\mathbf{l}_\mathrm{f}^{(\tau-1)}$ and $\mathbf{l}_\mathrm{f}^{(\tau-2)}$ denote the foreground binary label vectors at frames $\tau-1$ and $\tau-2$, respectively.


(a) Frame $\tau-1$ (b) $\mathbf{l}_\mathrm{f}^{(\tau-1)}$ (c) $\boldsymbol{\phi}_\mathrm{f}^{(\tau)}$

(d) Frame τ (e) $\mathbf{r}_\mathrm{f}^{(\tau)}$ (f) $\mathbf{s}_\mathrm{f}^{(\tau)}$

Figure 4. The spatiotemporal distribution $\mathbf{s}_\mathrm{f}^{(\tau)}$ provides a reliable estimate of the foreground, by multiplying the temporal estimate $\boldsymbol{\phi}_\mathrm{f}^{(\tau)}$ and the spatial estimate $\mathbf{r}_\mathrm{f}^{(\tau)}$.

Then, we compute the spatiotemporal distribution

$$\mathbf{s}_\mathrm{f}^{(\tau)} = \beta \times \boldsymbol{\phi}_\mathrm{f}^{(\tau)} \otimes \mathbf{r}_\mathrm{f}^{(\tau)} \tag{7}$$

where $\otimes$ denotes the element-wise multiplication, and β is a constant to normalize $\mathbf{s}_\mathrm{f}^{(\tau)}$. By combining the spatial and temporal estimates, the spatiotemporal distribution $\mathbf{s}_\mathrm{f}^{(\tau)}$ provides a more reliable estimate of the foreground, as shown in Figure 4(f). Therefore, we define the spatiotemporal energy $\mathcal{E}_\mathrm{S}$ as

$$\mathcal{E}_\mathrm{S}(\mathbf{p}_\mathrm{f}^{(\tau)}) = ||\mathbf{p}_\mathrm{f}^{(\tau)} - \mathbf{s}_\mathrm{f}^{(\tau)}||^2 \tag{8}$$

in order to enforce the foreground distribution to be similar to the spatiotemporal distribution.

Antagonistic Energy: We segment a foreground object from its background by comparing the foreground and background distributions, $\mathbf{p}_\mathrm{f}^{(\tau)}$ and $\mathbf{p}_\mathrm{b}^{(\tau)}$. For reliable and accurate segmentation, these two distributions should have high probabilities at mutually exclusive regions. In other words, they should not have high probabilities at the same region. To formulate this mutual exclusiveness between $\mathbf{p}_\mathrm{f}^{(\tau)}$ and $\mathbf{p}_\mathrm{b}^{(\tau)}$, we define the antagonistic energy $\mathcal{E}_\mathrm{A}$ as

$$\mathcal{E}_\mathrm{A}(\mathbf{p}_\mathrm{f}^{(\tau)}, \mathbf{p}_\mathrm{b}^{(\tau)}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{M}_i} w_{ij}^{(\tau)} p_{\mathrm{f},i}^{(\tau)} p_{\mathrm{b},j}^{(\tau)} \tag{9}$$

Algorithm 1 Alternate Convex Optimization (ACO)

```
Input: Graph $G$ at frame $\tau$ and label vectors at previous frames
Compute affinity and transition matrices                          ▷ (1) and (3)
Compute initial distributions $\mathbf{r}_{\text{f}}^{(\tau)}$ and $\mathbf{r}_{\text{b}}^{(\tau)}$          ▷ (2)
Compute spatiotemporal distributions $\mathbf{s}_{\text{f}}^{(\tau)}$ and $\mathbf{s}_{\text{b}}^{(\tau)}$     ▷ (7)
repeat
    Optimization for foreground distribution $\mathbf{p}_{\text{f}}^{(\tau)}$                ▷ (15)
    Optimization for background distribution $\mathbf{p}_{\text{b}}^{(\tau)}$                ▷ (16)
until $\mathcal{E}(\mathbf{p}_{\text{f}}^{(\tau)}, \mathbf{p}_{\text{b}}^{(\tau)})$ stops decreasing                      ▷ (10)
Output: Foreground and background distributions $\mathbf{p}_{\text{f}}^{(\tau)}$ and $\mathbf{p}_{\text{b}}^{(\tau)}$
```

where $\mathcal{M}_i$ denotes the set of the neighbors of node i and $w_{ij}^{(\tau)}$ is the affinity between nodes i and j at frame τ. The antagonistic energy is reduced, when a highly probable foreground node is surrounded by unlikely background neighbors. Thus, for a low antagonistic energy, the foreground and the background should form their own dominant regions.

Alternate Convex Optimization: We combine the three energy terms into a hybrid energy. For notational simplicity, let us omit the superscripts for frame indices. Then, to obtain the optimal foreground and background distributions $\mathbf{p}_{\text{f}}$ and $\mathbf{p}_{\text{b}}$, we minimize the hybrid energy

$$\mathcal{E}(\mathbf{p}_{\text{f}}, \mathbf{p}_{\text{b}}) = \mathcal{E}_{\text{M}}(\mathbf{p}_{\text{f}}) + \mathcal{E}_{\text{M}}(\mathbf{p}_{\text{b}}) + \gamma \cdot \mathcal{E}_{\text{S}}(\mathbf{p}_{\text{f}}) + \gamma \cdot \mathcal{E}_{\text{S}}(\mathbf{p}_{\text{b}}) + \delta \cdot \mathcal{E}_{\text{A}}(\mathbf{p}_{\text{f}}, \mathbf{p}_{\text{b}}) \tag{10}$$

subject to the constraints

$$0 \le p_{\text{f},i} \le 1, \qquad \textstyle\sum_{i=1}^{N} p_{\text{f},i} = 1, \tag{11}$$

$$0 \le p_{\text{b},i} \le 1, \qquad \textstyle\sum_{i=1}^{N} p_{\text{b},i} = 1. \tag{12}$$

In (10), nonnegative parameters γ and δ control the tradeoffs between the three energy terms.

Let $\mathbf{p} = [\mathbf{p}_{\text{f}}^T, \mathbf{p}_{\text{b}}^T]^T$. Then, the hybrid energy in (10) can be rewritten as

$$\mathcal{E}(\mathbf{p}) = \mathbf{p}^T\mathbf{B}\mathbf{p} - 2\gamma[\mathbf{s}_{\text{f}}^T, \mathbf{s}_{\text{b}}^T]\mathbf{p} + \gamma\mathbf{s}_{\text{f}}^T\mathbf{s}_{\text{f}} + \gamma\mathbf{s}_{\text{b}}^T\mathbf{s}_{\text{b}} \tag{13}$$

where

$$\mathbf{B} = \begin{bmatrix} (\mathbf{A}-\mathbf{I})^T(\mathbf{A}-\mathbf{I})+\gamma\mathbf{I} & \frac{\delta}{2}\mathbf{W} \\ \frac{\delta}{2}\mathbf{W} & (\mathbf{A}-\mathbf{I})^T(\mathbf{A}-\mathbf{I})+\gamma\mathbf{I} \end{bmatrix}. \tag{14}$$

Notice that, without the antagonistic energy $\mathcal{E}_{\text{A}}(\mathbf{p}_{\text{f}}, \mathbf{p}_{\text{b}})$, the non-diagonal sub-matrices in (14) would be zero and $\mathbf{B}$ would be positive semidefinite. In such a case, the minimization of $\mathcal{E}(\mathbf{p})$ subject to the constraints in (11) and (12) becomes a quadratic program [6], which can be solved easily, *e.g.*, using Lagrange multipliers.

However, the antagonistic energy makes $\mathbf{B}$ indefinite, and the minimization problem is a nonconvex one, which is difficult to solve. To overcome this difficulty, we develop the ACO scheme, which decomposes the nonconvex problem into two convex subproblems. First, after fixing the background distribution $\mathbf{p}_{\text{b}}$, we solve a quadratic program:

$$\min_{\mathbf{p}_{\text{f}}} \big\{\mathbf{p}_{\text{f}}^T \left((\mathbf{A}-\mathbf{I})^T(\mathbf{A}-\mathbf{I}) + \gamma\mathbf{I}\right)\mathbf{p}_{\text{f}} - \left(2\gamma\mathbf{s}_{\text{f}}^T - \delta\mathbf{p}_{\text{b}}^T\mathbf{W}\right)\mathbf{p}_{\text{f}}\big\} \tag{15}$$

subject to the constraints in (11). Then, using the resultant $\mathbf{p}_{\text{f}}$, we update the background distribution $\mathbf{p}_{\text{b}}$ by solving the other quadratic program:

$$\min_{\mathbf{p}_{\text{b}}} \big\{\mathbf{p}_{\text{b}}^T \left((\mathbf{A}-\mathbf{I})^T(\mathbf{A}-\mathbf{I}) + \gamma\mathbf{I}\right)\mathbf{p}_{\text{b}} - \left(2\gamma\mathbf{s}_{\text{b}}^T - \delta\mathbf{p}_{\text{f}}^T\mathbf{W}\right)\mathbf{p}_{\text{b}}\big\} \tag{16}$$

subject to the constraints in (12). We solve the two quadratic programs alternately, using the software in [16, 17]. When we first solve (15), the initial distribution $\mathbf{r}_{\text{b}}$ is used as the background distribution $\mathbf{p}_{\text{b}}$. The alternate scheme is guaranteed to converge and yield a locally optimal solution, since each quadratic program in (15) or (16) monotonically decreases the hybrid energy in (13) that is bounded below. **Algorithm 1** summarizes the ACO scheme.

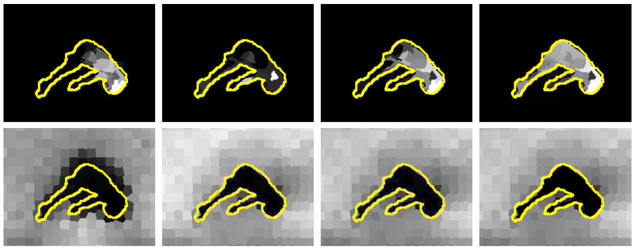

(a) $\mathcal{E}_{\text{S}} + \mathcal{E}_{\text{A}}$ (b) $\mathcal{E}_{\text{M}} + \mathcal{E}_{\text{A}}$ (c) $\mathcal{E}_{\text{M}} + \mathcal{E}_{\text{S}}$ (d) $\mathcal{E}_{\text{M}} + \mathcal{E}_{\text{S}} + \mathcal{E}_{\text{A}}$

Figure 5. The resulting distributions of minimizing various energy combinations on "Cliff Diving." The upper and lower rows are the foreground and background distributions, respectively. Yellow boundaries are the outlines of the ground-truth object. The images are cropped for better visualization.

Figure 5 exemplifies how each of the three energy terms affects the resulting distributions. Specifically, we exclude one of the three terms to analyze its efficacy. In Figure 5(a), without the Markov energy $\mathcal{E}_{\text{M}}$, the distributions do not spread according to the node affinities and the background distribution fails to cover the region near the object boundary. In Figure 5(b), the spatiotemporal energy $\mathcal{E}_{\text{S}}$ is omitted. The foreground distribution is concentrated on only a few superpixels, since the spatial and temporal correlations are not exploited. In Figure 5(c), the antagonistic energy $\mathcal{E}_{\text{A}}$ is excluded. The two distributions discover the object and the background relatively well. However, without the interaction between the two distributions, the foreground distribution fails to find the diver's legs. In contrast, in Figure 5(d), the diver is accurately separated from the background, by combining all three terms.

Table 1. The average numbers of mislabelled pixels per frame on the SegTrack dataset [33]. Lower values are better. The best and the second best results are boldfaced and underlined, respectively.

		Unsupervised - Single					Unsupervised - Multiple						Supervised	
Video (Number of frames)		ACO	[34]	[29]	[37]	[25]	[14]	[23]	[28]	[5]	[21]	[9]	[33]	[11]
SegTrack	Birdfall (30)	**144**	209	217	$\underline{155}$	189	278	199	468	606	288	468	252	454
	Cheetah (29)	**617**	796	890	$\underline{633}$	806	824	599	1175	11210	905	1968	1142	1217
	Girl (21)	$\underline{1195}$	**1040**	3859	1488	1598	1029	1164	5683	26409	1785	7595	1304	1755
	Monkeydog (71)	$\underline{354}$	562	**284**	365	472	192	322	1434	12662	521	1434	563	683
	Parachute (51)	**200**	$\underline{207}$	855	220	221	251	242	1595	40251	201	1113	235	502

Foreground and Background Labeling: After the ACO of the foreground and background distributions, we use the maximum a posteriori (MAP) criterion to determine the binary segmentation label of each superpixel [20]. The probability $p_{\mathrm{f},i}^{(\tau)}$ that the foreground is found on node i at frame τ is regarded as the likelihood $p(x_i|\omega_{\mathrm{f}}^{(\tau)})$, and $p(\omega_{\mathrm{f}}^{(\tau)})$ is the prior probability of the foreground at frame τ. Also, $p(x_i|\omega_{\mathrm{b}}^{(\tau)})$ and $p(\omega_{\mathrm{b}}^{(\tau)})$ are similarly defined. Then, we compute the posterior by

$$p(\omega_{\mathrm{f}}^{(\tau)}|x_i) = \frac{p(x_i|\omega_{\mathrm{f}}^{(\tau)})p(\omega_{\mathrm{f}}^{(\tau)})}{p(x_i|\omega_{\mathrm{f}}^{(\tau)})p(\omega_{\mathrm{f}}^{(\tau)}) + p(x_i|\omega_{\mathrm{b}}^{(\tau)})p(\omega_{\mathrm{b}}^{(\tau)})}, \tag{17}$$

which represents the probability that node i is occupied by the foreground. Then, the segmentation labels $l_{\mathrm{f},i}^{(\tau)}$ and $l_{\mathrm{b},i}^{(\tau)}$ at node i are determined by

$$(l_{\mathrm{f},i}^{(\tau)}, l_{\mathrm{b},i}^{(\tau)}) = \begin{cases} (1,0) & \text{if } p(\omega_{\mathrm{f}}^{(\tau)}|x_i) > p(\omega_{\mathrm{b}}^{(\tau)}|x_i), \\ (0,1) & \text{otherwise.} \end{cases} \tag{18}$$

We estimate the prior probabilities $p(\omega_{\mathrm{f}}^{(\tau)})$ and $p(\omega_{\mathrm{b}}^{(\tau)})$ of the foreground and background at frame τ, by employing the distributions at the previous frame $(\tau-1)$. Suppose that the two distributions are completely separated and that each distribution is uniformly spread in its own region. Then, the number of the nodes that each distribution occupies is equal to the inverse of its uniform probability. Inspired by this, we estimate the priors by

$$p(\omega_{\mathrm{f}}^{(\tau)}) = \frac{1}{\max_i p_{\mathrm{f},i}^{(\tau-1)}}, \quad p(\omega_{\mathrm{b}}^{(\tau)}) = \frac{1}{\max_i p_{\mathrm{b},i}^{(\tau-1)}}. \tag{19}$$

3.4. Backward Pass of Video Object Segmentation

In the forward pass, the probability distributions in later frames are more reliable than those in earlier frames, since the segmentation results in past frames are used to improve the clustering performance for a current frame. Thus, we further carry out a backward pass, which progresses from the last to the first frames. The backward pass is the same as the forward pass, except for primary object selection. The backward pass selects a primary object in each frame among multiple connected components of foreground superpixels. We determine the priority score for each component, by summing up the foreground probabilities of the corresponding superpixels. We then declare the component with the highest priority as the primary object.

3.5. Pixel-wise Refinement

Notice that each frame is over-segmented into superpixels to reduce the number of graph nodes. Thus, we refine segmentation results at the superpixel-level into those at the pixel-level by employing the Markov random field optimization scheme in [37].

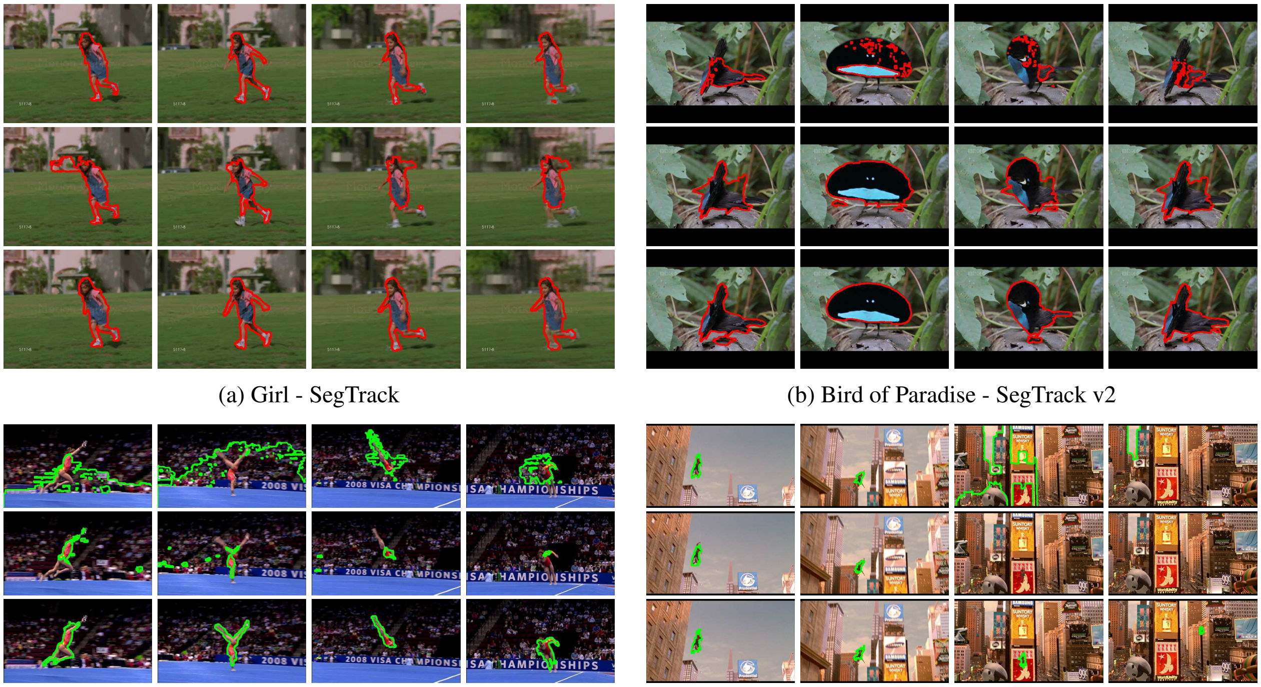
4. Experimental Results

We test the proposed ACO video object segmentation algorithm on three datasets: SegTrack [33], SegTrack v2 [23], and VidSeg. We use the same parameters in all experiments.

SegTrack [33] was initially announced to evaluate non-rigid object tracking algorithms. Among the six videos, five are typically used to assess video object segmentation performance [14, 25, 29, 34, 37], and their pixel-wise ground-truth maps are available. Since a primary object in each video maintains similar sizes over all frames, the average number of mislabelled pixels per frame is used as the performance metric in [14, 25, 29, 34, 37].

Table 1 compares the performance of the proposed ACO algorithm on the SegTrack dataset with those of 12 conventional algorithms: unsupervised-single [25, 29, 34, 37], unsupervised-multiple [5, 9, 14, 21, 23, 28], and supervised non-rigid object tracking [11, 33]. For the performance assessment, the unsupervised-multiple algorithms should select the best segment track, among the multiple tracks, that maximally matches with the ground-truth. The proposed ACO algorithm outperforms most conventional unsupervised-single algorithms. Furthermore, ACO even surpasses most unsupervised-multiple algorithms and two supervised trackers [11, 33].

SegTrack v2 [23] extends the SegTrack dataset by supplying eight video sequences and their ground-truth maps. We select five sequences, each of which contains a single prominent object. Also, we propose a new dataset, VidSeg. We collect eight videos from the YouTube and four movie clips. Except for the "Long Jump" sequence, we extract the ground-truth map for every fifth frame. For "Long Jump," we annotate the ground-truth labels for all frames because of its relatively short length. The VidSeg videos are chal-



(a) Girl - SegTrack

(b) Bird of Paradise - SegTrack v2

(c) Floor Exercise - VidSeg

(d) Spider Man - VidSeg

Figure 6. Comparison of video object segmentation results. Segmentation boundaries are depicted in red or green. Each sub-figure shows the results of [37], [29], and the proposed algorithm from top to bottom.

lenging due to complex object appearance, object deformation, cluttered background, and long video lengths.

The average number of mislabelled pixels per frame can be misleading, since it simply counts wrong labels regardless of an object size. Li *et al.* [23] pointed out this issue, and used the intersection over union (IoU) score, IoU $= 100 \cdot \frac{|T \cap R|}{|T \cup R|}$ where T and R are the sets of foreground pixels in a segmentation result and the corresponding ground-truth map, respectively. Since object sizes vary dramatically in SegTrack v2 and VidSeg, we measure the segmentation performance by IoU. We compute the IoU score for each frame and report the average IoU score over all frames.

Table 2 summarizes the IoU scores on all three datasets. Two conventional algorithms [29, 37] are compared, whose source codes are available. The proposed ACO algorithm outperforms these two algorithms. Especially, ACO is effective for long-duration videos, whose primary objects suffer from non-rigid appearance deformation and background variations. In terms of the average scores, ACO outperforms [29] and [37] by significant margins, about 19.0% and 14.2%, respectively.

Table 2. Comparison of IoU scores. Higher values are better. The best and the second best results are boldfaced and underlined.

Video (Number of frames)		[29]	[37]	ACO
SegTrack	Birdfall (30)	3.88	71.98	**73.29**
	Cheetah (29)	44.95	**65.48**	64.23
	Girl (21)	56.83	81.46	**86.75**
	Monkeydog (71)	72.62	72.06	**76.12**
	Parachute (51)	85.61	94.47	**94.68**
SegTrack v2	Bird of Paradise (98)	83.46	27.02	**93.92**
	Frog (279)	65.20	72.00	**81.58**
	Monkey (31)	**69.28**	63.28	63.96
	Soldier (32)	**46.48**	39.57	36.84
	Worm (243)	**73.62**	44.59	61.79
VidSeg	Bike Rampage (347)	21.82	0.59	**46.13**
	Bike Riding (631)	34.87	73.80	**77.41**
	Cliff Diving (104)	60.28	77.82	**84.56**
	Floor Exercise (114)	15.65	14.28	**61.84**
	Long Jump (84)	60.14	78.14	**79.85**
	Tennis (679)	33.89	45.81	**56.08**
	White Bird (628)	63.49	78.86	**79.60**
	Wolf (362)	17.15	**78.30**	75.91
	Hulk (160)	65.18	31.14	**78.82**
	Jurassic Park (118)	50.79	**83.60**	37.97
	Silver Surfer (100)	69.88	0.68	**75.02**
	Spider Man (118)	32.24	37.54	**58.84**
Average		51.24	56.02	**70.24**

Figure 6 shows segmentation results, obtained by [37], [29], and the proposed ACO algorithm. ACO delineates primary objects precisely and robustly in these sequences, even though they have appearance deformation, motion blur (“Floor Exercise”), and cluttered background (“Spider Man”). Since [37] relies on object proposals, it fails when the proposal scheme does not identify small or complex objects in “Spider Man.” Also, on “Floor Exercise,” [29] provides unsuccessful results due to its strong dependency on motion boundaries. In contrast, ACO robustly identi-

Table 3. Average IoU score after each proposed step.

	Initial	Forward	Backward	Pixel-wise
Average	44.31	64.80	67.71	70.24

Table 4. Average IoU scores of the proposed ACO algorithm using various energy combinations.

	$\mathcal{E}_\mathrm{S}$	$\mathcal{E}_\mathrm{S}+\mathcal{E}_\mathrm{A}$	$\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{A}$	$\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{S}$	$\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{S}+\mathcal{E}_\mathrm{A}$
Average	52.81	52.92	11.30	57.87	67.71

fies primary objects by minimizing the hybrid energy of the Markov, spatiotemporal, and antagonistic terms.

Step Analysis: Table 3 lists the average IOU score of the proposed ACO algorithm, after performing each step. The forward pass improves the initial score by exploiting temporal, as well as spatial, correlations. Then, the backward pass corrects misjudged segments in the forward pass. Finally, the pixel-wise refinement further boosts the performance to the final score of 70.24.

Energy Term Analysis: We analyze the efficacy of each energy term, by testing various combinations of the three energy terms. For each combination, Table 4 reports the average IoU score without the pixel-wise refinement to focus on the energy terms only. Note that the energies, $\mathcal{E}_\mathrm{S}$ and $\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{S}$, are minimized by the global convex optimization, instead of the ACO. The spatiotemporal energy $\mathcal{E}_\mathrm{S}$ plays an essential role by exploiting motion information, as well as spatial correlations, in video sequences. Thus, the combination $\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{A}$ provides the worst performance. However, the Markov energy $\mathcal{E}_\mathrm{M}$ and the antagonistic energy $\mathcal{E}_\mathrm{A}$ are also important, and thus $\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{S}+\mathcal{E}_\mathrm{A}$ outperforms both $\mathcal{E}_\mathrm{S}+\mathcal{E}_\mathrm{A}$ and $\mathcal{E}_\mathrm{M}+\mathcal{E}_\mathrm{S}$ significantly. Table 4, together with Figure 5, indicates that the three energy terms are complementary to one another. The energy terms yield excellent segmentation performance, when they are jointly minimized.

Multiple Primary Object Segmentation: Next, we apply the proposed ACO algorithm to videos that contain multiple primary objects. The number of primary objects, k, is assumed to be known. For the initialization, we first perform the scheme in Section 3.2 to obtain initial foreground and background distributions. Then, we divide the initial foreground distribution into k distributions using the k-means clustering technique. Consequently, we use k foreground distributions (one for each object) and a single background one. The remaining steps are straightforwardly generalized from those of the single object segmentation. For example, in the ACO, we optimize each distribution, after fixing the other k distributions, repeatedly in a round-robin manner.

Table 5 presents the IoU scores on the three sequences in SegTrack v2 [23], each of which has two primary objects. Three conventional algorithms [9, 21, 28] are compared, whose source codes are publicly available and applicable to the multiple object segmentation task. [21] extracts a primary object by selecting a hypothesis, which is a set of object proposals across frames. It yields hypotheses with priorities. We test [21] in two ways. In the column [21]-T, we compute the IoU score of the two hypotheses with the highest priorities. On the other hand, in [21]-A, we report the IoU score of the best combination of two hypotheses, among all combinations. The best combination is selected using the ground truth. [9] and [28] are the motion segmentation algorithms to yield dense segments. To measure their primary object segmentation performances, we find the maximally matched segment for each ground-truth object using the IoU criterion. It is hence unfair to compare the proposed ACO algorithm, requiring only the number k of objects, with [21]-A, [9] and [28] that use the ground-truth data. However, ACO significantly outperforms [21]-T, [9] and [28], and provides comparable performance to [21]-A. Figure 7 shows examples of the multiple primary object segmentation results of the proposed ACO algorithm.

Table 5. Multiple primary object segmentation performances (in IoU scores) of the proposed ACO algorithm and the conventional algorithms [9, 21, 28]. The best and the second best results are boldfaced and underlined, respectively.

Video (Number of frames)		[9]	[21]-T	[21]-A	[28]	Ours
SegTrack v2	BMX (36)	4.90	37.25	**63.76**	4.90	<u>55.42</u>
	Drifting Car (74)	**59.04**	35.52	50.91	21.49	<u>57.72</u>
	Hummingbird (29)	32.42	31.46	**44.28**	27.46	<u>35.92</u>
	Average	32.12	34.74	**52.98**	17.95	<u>49.69</u>

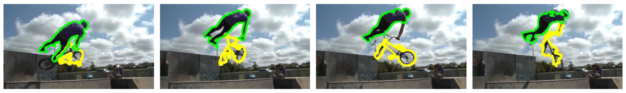


Figure 7. Multiple primary object segmentation results of the proposed ACO algorithm. The boundaries of the two primary objects are depicted in yellow and green, respectively. The frames are from the "BMX" in SegTrack v2 [23].

5. Conclusions

We proposed a novel unsupervised video object segmentation algorithm. We first defined a hybrid of the Markov, spatiotemporal, and antagonistic energies, and then minimized the hybrid energy to delineate a primary object. To minimize the nonconvex hybrid energy, we developed the ACO scheme, which optimized the foreground and background distributions alternately by solving quadratic programs. We also proposed the forward-backward strategy to yield temporally consistent segmentation results. Experiments showed that the proposed ACO algorithm outperforms the state-of-the-art techniques significantly.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A10055037).

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. Mach. Intell.*, 34(11):2274–2282, 2012. 2

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010. 2

[3] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. *ACM Trans. Graphics*, 28(3):70, 2009. 1

[4] D. Banica, A. Agape, A. Ion, and C. Sminchisescu. Video object segmentation by salient segment chain composition. In *ICCVW*, pages 283–290, 2013. 2

[5] O. Barnich and M. V. Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.*, 20(6):1709–1724, 2011. 2, 6

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 5

[7] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, pages 833–840, 2009. 2

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998. 3

[9] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. 2010. 2, 6, 8

[10] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, pages 3241–3248, 2010. 2

[11] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, pages 1530–1537, 2009. 1, 6

[12] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 3

[13] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588. 2010. 2

[14] D. Giordano, F. Murabito, S. Palazzo, and C. Spampinato. Superpixel-based video object segmentation using perceptual organization and location prior. In *CVPR*, pages 4814–4822, 2015. 2, 6

[15] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006. 3

[16] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, pages 95–110. Springer, 2008. 5

[17] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, 2014. 5

[18] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, pages 2141–2148, 2010. 2

[19] B. Han and L. S. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):1017–1023, 2012. 2

[20] C. Lee, W.-D. Jang, J.-Y. Sim, and C.-S. Kim. Multiple random walkers and their application to image cosegmentation. In *CVPR*, pages 3837–3845, 2015. 3, 6

[21] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002, 2011. 2, 6, 8

[22] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Optimal image and video closure by superpixel grouping. *Int. J. Comput. Vis.*, 100(1):99–119, 2012. 2

[23] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 1, 2, 6, 7, 8

[24] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 2

[25] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, 2012. 1, 2, 6

[26] M. Meila and J. Shi. Learning segmentation by random walks. In *Adv. Neural Inf. Process. Syst.*, pages 873–879, 2001. 3

[27] 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. 2

[28] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *CVPR*, pages 614–621, 2012. 2, 6, 8

[29] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, pages 1777–1784, 2013. 1, 2, 6, 7

[30] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, pages 779–786, 2009. 1

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

[32] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, pages 4268–4276, 2015. 2

[33] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label MRF optimization. In *BMVC*, pages 1–11, 2010. 1, 6

[34] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, pages 3395–3402, 2015. 1, 2, 6

[35] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, pages 29–42. 2012. 3

[36] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013. 3

[37] 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, 6, 7

[38] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *Adv. Neural Inf. Process. Syst.*, pages 169–176, 2004. 3