Semi-supervised Video Object Segmentation Using Multiple Random Walkers

Won-Dong Jang
wdjang@mcl.korea.ac.kr
Chang-Su Kim
changsukim@korea.ac.kr

School of Electrical Engineering
Korea University
Seoul, Korea

Abstract

A semi-supervised video object segmentation algorithm using multiple random walkers (MRW) is proposed in this work. We develop an initial probability estimation scheme that minimizes an objective function to roughly separate the foreground from the background. Then, we simulate MRW by employing the foreground and background agents. During the MRW process, we update restart distributions using a hybrid of inference restart rule and interactive restart rule. By performing these processes from the second to the last frames, we obtain a segment track of the target object. Furthermore, we optionally refine the segment track by performing Markov random field optimization. Experimental results demonstrate that the proposed algorithm significantly outperforms the state-of-the-art conventional algorithms on the SegTrack v2 dataset.

1 Introduction

Video object segmentation, the task to separate objects from the background in a video sequence, is challenging due to various difficulties, *e.g.* object deformation, ambiguous boundaries, motion blur, and occlusion. To overcome these issues, many attempts have been made, which can be classified into three categories according to the types of user annotation: unsupervised algorithms [13, 17, 21, 35, 38], semi-supervised ones [11, 18, 39, 45], and supervised ones [2, 14, 15, 42].

The unsupervised algorithms automatically discover objects without requiring any user input. They usually assume that the object to be segmented has distinct motion or appear frequently in a video. The semi-supervised algorithms require user annotations about the desired object at the first frame. They then perform segmentation and tracking jointly. The supervised algorithms demand user annotations repeatedly during the segmentation process to extract precise segment tracks. As the supervised ones need too much user effort, the unsupervised or semi-supervised ones are more desirable in practical applications.

In this work, we propose a novel semi-supervised video object segmentation algorithm. First, for each frame, we estimate initial distributions of the foreground and the background based on the segmentation results of previous frames. Second, we simulate multiple random walkers (MRW) [27] using a hybrid of inference restart rule and interactive restart rule. We perform these two-step processes from the second to the last frames sequentially to yield a segment track. Optionally, we refine the segmentation results by performing Markov random

field (MRF) optimization. Experimental results demonstrate that the proposed algorithm outperforms the state-of-the-art conventional algorithms [8, 18, 36, 41] on the SegTrack v2 dataset [29]. To summarize, this paper has three main contributions:

- Development of an effective restart rule for MRW that yields spatially precise and temporally consistent segment tracks.
- Fixation of all parameters, which ensures promising performances on general video sequences without exhaustive parameter tuning.
- Remarkable performance achievement on the SegTrack v2 dataset, which is composed of challenging video sequences.

2 Related Work

2.1 Unsupervised Video Segmentation and Video Object Segmentation

Unsupervised video segmentation is a problem to divide all pixels in a video into mutually disjoint clusters of spatiotemporally coherent pixels. To this end, motion segmentation techniques [5, 33, 38] cluster long point trajectories, which can be converted into dense segments [32]. Inspired by [16], Grundmann *et al*. [21] perform agglomerative clustering on a spatiotemporal graph to achieve video segmentation. Galasso *et al*. [17] apply spectral clustering to divide a video sequence. Khoreva *et al*. [26] propose a video segmentation algorithm based on machine learning that trains affinities between superpixels.

On the other hand, unsupervised video object segmentation yields segment tracks of salient objects. By assuming that an important object appears frequently across frames, video object segmentation algorithms [28, 31, 47], which exploit the object proposal technique in [12], have been proposed. Also, visual and motion saliency maps are adopted for video object segmentation in [13, 35, 43]. Li *et al*. [29] first generate many hypotheses and then track them to yield multiple segment tracks. They also provide the SegTrack v2 dataset. Jang *et al*. [25] estimate initial distributions of the foreground and background by using the boundary priors, and then optimize them alternately to obtain a precise segment track.

2.2 Semi-supervised Video Object Segmentation

In semi-supervised video object segmentation, a user specifies an object region to be tracked. It is possible to annotate the target object using interactive image segmentation techniques, such as [22, 37]. Alternatively, tracking-by-segmentation algorithms [8, 11, 18, 41] take an object box as input. Their objective is, however, to track the box instead of accurate segmentation. Chockalingam *et al*. [9] divide an object into fragments and build its Gaussian mixture models for non-rigid object tracking in sequential frames. Tsai *et al*. [39] perform MRF optimization using appearance similarity and motion coherence to separate a foreground object from the background. They also gather the SegTrack dataset, composed of six video sequences. Budvytis *et al*. [6] construct a tree-structured graphical model for segmenting an object, which is manually labeled in the first and last frames. Also, Budvytis *et al*. [7] infer foreground labels from the mixture of tree-structured graphical models, and combine them with the prediction of a self-trained classifier. Ramakanth and Babu [36] extract video seams to transfer labels in a previous frame to a current frame. Varas and Marques [40] apply a particle filter [23] for video object segmentation. Jain and Grauman [24] over-segment a

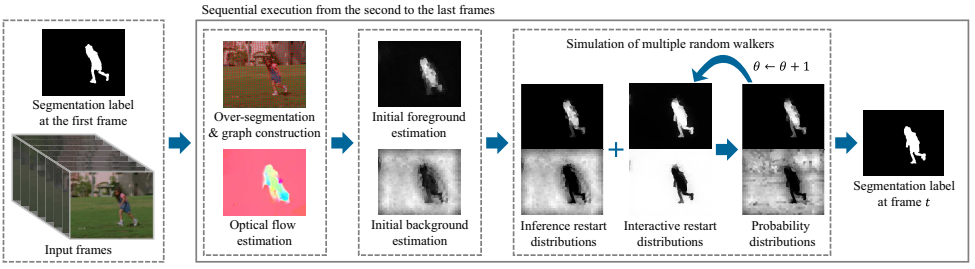


Figure 1: An overview of the proposed algorithm. Using segmentation labels at previous frames, we initialize foreground and background distributions for each frame. Then, we simulate MRW using the inference restart rule and the interactive restart rule. For the segmentation, we compare the foreground and background probabilities at each superpixel.

video into supervoxels and minimize an MRF energy function using supervoxel characteristics. Wen *et al.* [45] roughly predict foreground regions based on the superpixel matching, and then update a multi-part model of a target object via iterative energy optimization.

3 Proposed Algorithm

We propose a novel semi-supervised video object segmentation algorithm. The input is a set of video frames $\{I^{(1)},\dots,I^{(T)}\}$ and a pixel-level object annotation at the first frame $\mathbf{s}^{(1)}$. The output is a segment track of the object. We annotate an object in the first frame by employing the interactive image segmentation technique in [22], which accepts scribbles as input.

Figure 1 is an overview of the proposed algorithm. First, we extract color and motion features for each superpixel, and construct a graph using the superpixels. Second, we estimate initial foreground and background distributions. By simulating the MRW process, we separate the foreground from the background. During the iteration, we fix the inference restart rule but adapt the interactive restart rule. To yield a segment track, we execute these processes sequentially from the second to the last frames.

3.1 Feature Extraction and Graph Construction

We over-segment each frame into about 1,000 SLIC superpixels [1], which become nodes in a graph. Also, we estimate backward optical flows from $I^{(t)}$ to $I^{(t-1)}$ and $I^{(t-2)}$ [30]. For each superpixel, we extract its features, by computing the average LAB color and the average motion vector. For each frame, we construct a graph $G=(V,E)$, where $V=\{v_1,\dots,v_N\}$ is a set of nodes (or superpixels) and $E=\{e_{ij}\}$ is a set of edges. We link edges between superpixels according to the k-ring connectivity. In the k-ring graph, two nodes are connected if we can traverse from one to the other through connected neighbors within k transitions.

We set an edge weight (or affinity) between nodes i and j by

$$w_{ij}=\begin{cases}\exp\left(-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2/\sigma^2\right) & \text{if } e_{ij}\in E,\\ 0 & \text{otherwise,}\end{cases} \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ denote the features of nodes i and j, and σ is a scale parameter.

3.2 Initial Foreground and Background Estimation

We estimate initial foreground and background distributions by exploiting the segmentation results at previous frames. To this end, we formulate an energy function. By minimizing the energy function, we first estimate the initial distribution of background, $\mathbf{p}_{\mathrm{b}}^{(0)}$, and then that of foreground, $\mathbf{p}_{\mathrm{f}}^{(0)}$. Note that, in $\mathbf{p}_{\mathrm{b}}^{(0)}$ and $\mathbf{p}_{\mathrm{f}}^{(0)}$, the superscript '0' means the iteration index, not the frame index. The distributions will be refined in MRW iterations later.

The energy function consists of three terms: color Markov energy, motion Markov energy, and guidance energy. It is given by

$$\mathcal{E}(\mathbf{p},\mathbf{u}) = \lambda_{\mathrm{c}}\|\mathbf{A}_{\mathrm{c}}\mathbf{p}-\mathbf{p}\|_2^2 + \lambda_{\mathrm{m}}\|\mathbf{A}_{\mathrm{m}}\mathbf{p}-\mathbf{p}\|_2^2 + \lambda_{\mathrm{g}}\|\mathbf{p}-\mathbf{u}\|_2^2 \tag{2}$$

and is minimized with respect to the probability distribution $\mathbf{p}$, where

$$0 \le p_i \le 1, \qquad \textstyle\sum_{i=1}^{N} p_i = 1. \tag{3}$$

Also, $\mathbf{A}_{\mathrm{c}}$ and $\mathbf{A}_{\mathrm{m}}$ are color and motion transition matrices. By normalizing affinities in (1), we compute each element in the transition matrices, $a_{ij} = w_{ij}/\sum_l w_{lj}$, which indicates the proportion that a probability is transferred from node j to node i. The color and motion transition matrices, $\mathbf{A}_{\mathrm{c}}$ and $\mathbf{A}_{\mathrm{m}}$, disperse probabilities according to the color and motion features, respectively. Also, $\mathbf{u}$ denotes a guidance distribution, which is set differently for estimating the foreground and background distributions. The multipliers, λ_{c}, λ_{m}, and λ_{g}, adjust the trade-offs between the two Markov energies and the guidance energy. Let us describe these energy terms.

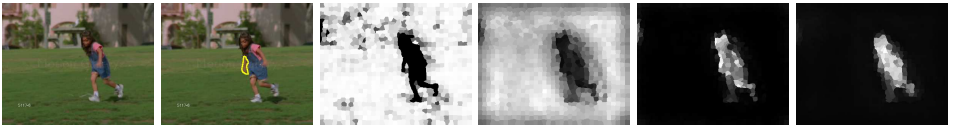
The color and motion Markov energies compel the probabilities to be distributed based on the random walk theory [10]. In the random walk simulation, a random walker (or agent) travels on a graph according to a transition matrix, and the recursion is given by

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

where θ indicates an iteration index. The stationary distribution can be found when the squared distance $\|\mathbf{A}\mathbf{p}^{(\infty)} - \mathbf{p}^{(\infty)}\|_2^2$ is minimized to 0. In the stationary distribution, neighboring nodes with similar features tend to have similar probabilities, since there are frequent transitions between them. Therefore, we adopt the two Markov energies to encourage similar neighboring nodes to have the same segmentation label (foreground or background), based on the color and motion attributes.

By adopting the guidance energy, we make $\mathbf{p}$ similar to the guidance distribution $\mathbf{u}$. Different guidance distributions are used for estimating the initial foreground and background distributions, while the color and motion Markov terms are used in common. To obtain the initial background distribution $\mathbf{p}_{\mathrm{b}}^{(0)}$, we compute the ith element $u_{\mathrm{b},i}$ of the guidance distribution using the segmentation result at the previous frame. To this end, we define the propagation matrix $\mathbf{H}^{(t,\tilde{t})} = [h_{ij}^{(t,\tilde{t})}]$, which computes overlap ratios between superpixels at frame t and frame $\tilde{t}$, matched by the optical flows. Note that each row in the propagation matrix is normalized so that $\sum_j h_{i,j}^{(t,\tilde{t})} = 1$. We transfer the inverses of the segmentation labels at the previous frame $t-1$ to the current frame t by

$$u_{\mathrm{b},i} = \alpha \sum_j w_{ij}^{(t,t-1)} h_{ij}^{(t,t-1)} (1 - s_j^{(t-1)}) \tag{5}$$



(a) Frame $t-1$ (b) Frame t (c) $\mathbf{u}_{\mathrm{b}}$ (d) $\mathbf{p}_{\mathrm{b}}^{(0)}$ (e) $\mathbf{u}_{\mathrm{f}}$ (f) $\mathbf{p}_{\mathrm{f}}^{(0)}$

Figure 2: An example of the initial background and foreground estimation. The foreground guidance distribution $\mathbf{u}_{\mathrm{f}}$ is obtained from the initial background distribution $\mathbf{p}_{\mathrm{b}}^{(0)}$. High initial foreground probabilities $\mathbf{p}_{\mathrm{f}}^{(0)}$ are assigned to the girl's right arm as well, which is not visible in the frame $t-1$. We mark the arm with yellow boundaries at frame t.

where $w_{ij}^{(t,t-1)}$ is an inter-frame affinity between node i at frame t and node j at frame $t-1$. We measure the inter-frame affinity by comparing their average LAB colors. Also, $s_j^{(t-1)}$ is the segmentation label of node j at frame $t-1$. Specifically, $s_j^{(t-1)}=1$ if the superpixel is labeled as the foreground and $s_j^{(t-1)}=0$ otherwise. The parameter α is set to normalize the guidance distribution $\mathbf{u}_{\mathrm{b}}$ into a probability distribution. By plugging this guidance distribution into (2) and minimizing the energy function, we exploit the segmentation labels at the previous frame to infer the initial background distribution $\mathbf{p}_{\mathrm{b}}^{(0)}$ at the current frame.

After determining the initial background distribution $\mathbf{p}_{\mathrm{b}}^{(0)}$, we estimate the initial foreground distribution $\mathbf{p}_{\mathrm{f}}^{(0)}$. For the foreground estimation, we exploit the information in $\mathbf{p}_{\mathrm{b}}^{(0)}$, instead of the segmentation labels at the previous frame. The initial background distribution $\mathbf{p}_{\mathrm{b}}^{(0)}$ tends to yield very low probabilities on the nodes corresponding to the target object. We, hence, simply set the ith element of the foreground guidance distribution via $u_{\mathrm{f},i} \propto \exp(-p_{\mathrm{b},i}^{(0)})$. In this way, newly appearing parts, as well as already visible parts, of the target object can have high foreground guidance probabilities. If we set the guidance distribution by propagating the foreground labels at the previous frame, newly appearing parts may be misguided by low guidance probabilities.

Since the minimization of $\mathcal{E}(\mathbf{p},\mathbf{u})$ in (2) subject to the constraints in (3) is a quadratic program [3], we minimize $\mathcal{E}(\mathbf{p}_{\mathrm{b}}^{(0)},\mathbf{u}_{\mathrm{b}})$ and $\mathcal{E}(\mathbf{p}_{\mathrm{f}}^{(0)},\mathbf{u}_{\mathrm{f}})$ using [19, 20]. Figure 2 exemplifies the initial background and foreground estimation. The girl's right arm, which is not visible in frame $t-1$, yields low probabilities in $\mathbf{p}_{\mathrm{b}}^{(0)}$. Thus, by exploiting the information in $\mathbf{p}_{\mathrm{b}}^{(0)}$ to set the guidance distribution $\mathbf{u}_{\mathrm{f}}$, we obtain high initial foreground probabilities on the arm. Note that $\mathbf{p}_{\mathrm{b}}^{(0)}$ and $\mathbf{p}_{\mathrm{f}}^{(0)}$ roughly delineate the background and the girl, respectively.

3.3 Segmentation Using Multiple Random Walkers

We refine the foreground and background distributions, using MRW, in order to separate the foreground from the background. MRW simulates movements of multiple agents [27]. In the MRW system, agents interact with one another by adapting their restart distributions, while the random walk with restart [34] uses a fixed restart rule. For the video object segmentation, we define double agents, *i.e.* foreground and background agents. To simulate interactions between them, we combine a time-invariant restart rule with a time-varying restart rule.

MRW System: In this work, we employ the foreground and background agents on a graph. Let $\mathbf{p}_{\mathrm{f}}^{(\theta)}=[p_{\mathrm{f},1}^{(\theta)},\cdots,p_{\mathrm{f},N}^{(\theta)}]^{\mathrm{T}}$ and $\mathbf{p}_{\mathrm{b}}^{(\theta)}=[p_{\mathrm{b},1}^{(\theta)},\cdots,p_{\mathrm{b},N}^{(\theta)}]^{\mathrm{T}}$ be vectors, in which $p_{\mathrm{f},i}^{(\theta)}$ and $p_{\mathrm{b},i}^{(\theta)}$ are

the probabilities that the foreground and background agents are found at node i at iteration θ, respectively. The movements of the foreground and background agents are modeled by

$$\mathbf{p}_{\mathrm{f}}^{(\theta+1)} = (1-\varepsilon)\mathbf{A}_{\mathrm{c}}\mathbf{p}_{\mathrm{f}}^{(\theta)} + \varepsilon\mathbf{r}_{\mathrm{f}}^{(\theta)}, \tag{6}$$

$$\mathbf{p}_{\mathrm{b}}^{(\theta+1)} = (1-\varepsilon)\mathbf{A}_{\mathrm{c}}\mathbf{p}_{\mathrm{b}}^{(\theta)} + \varepsilon\mathbf{r}_{\mathrm{b}}^{(\theta)}, \tag{7}$$

where $\mathbf{r}_{\mathrm{f}}^{(\theta)} = [r_{\mathrm{f},1}^{(\theta)}, \cdots, r_{\mathrm{f},N}^{(\theta)}]^{\mathrm{T}}$ and $\mathbf{r}_{\mathrm{b}}^{(\theta)} = [r_{\mathrm{b},1}^{(\theta)}, \cdots, r_{\mathrm{b},N}^{(\theta)}]^{\mathrm{T}}$ are the foreground and background restart distributions. $\mathbf{A}_{\mathrm{c}}$ is the color transition matrix in (2). With probability $1-\varepsilon$, the agents move on the graph according to the transition matrix $\mathbf{A}_{\mathrm{c}}$. On the other hand, with probability ε, the foreground and background agents are forced to restart with the distributions $\mathbf{r}_{\mathrm{f}}^{(\theta)}$ and $\mathbf{r}_{\mathrm{b}}^{(\theta)}$, respectively. The MRW system makes the agents interact with each other via these restart distributions. Specifically, to achieve the interactions, a restart rule determines the restart distributions $\mathbf{r}_{\mathrm{f}}^{(\theta)}$ and $\mathbf{r}_{\mathrm{b}}^{(\theta)}$ by considering both $\mathbf{p}_{\mathrm{f}}^{(\theta)}$ and $\mathbf{p}_{\mathrm{b}}^{(\theta)}$. It was proven in [27] that the MRW process converges to a stationary distribution π_{f} and π_{b} for any restart rule, by introducing a cooling factor. We set the cooling factor to 0.995. While the 1-ring graph is used for the foreground distribution in (6), the 4-ring graph is employed for the background one in (7). Note that the four sides of an image usually belong to the background [44, 46], and superpixels along the sides are connected to a small number of nodes. Hence, we provide a wider connection for the background agent than for the foreground agent.

Restart Rules: We propose two restart rules: inference restart rule and interactive restart rule. For simplicity, let us describe the algorithm from the viewpoint of the foreground agent. The background agent is handled in a symmetrical manner.

The inference restart rule is time-invariant and inferred from the previous segmentation labels. We propagate the segmentation results $\mathbf{s}^{(t-2)}$ and $\mathbf{s}^{(t-1)}$ at the two previous frames to the current frame to determine the inference restart rule, which is given by

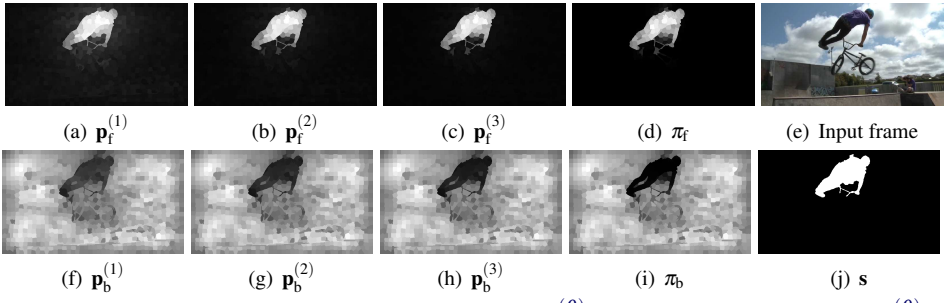
$$\mathbf{q}_{\mathrm{f}} = \beta \times \mathbf{p}_{\mathrm{f}}^{(0)} \otimes (\mathbf{H}^{(t,t-2)}\mathbf{s}^{(t-2)} + \mathbf{H}^{(t,t-1)}\mathbf{s}^{(t-1)}), \tag{8}$$

where $\otimes$ indicates the element-wise multiplication, $\mathbf{p}_{\mathrm{f}}^{(0)}$ is the initial foreground distribution, $\mathbf{H}^{(t,\tilde{t})}$ is the propagation matrix defined in Section 3.2, and β normalizes the inference restart distribution $\mathbf{q}_{\mathrm{f}}$. Since the inference restart rule is time-invariant, we allocate high probabilities to reliable superpixels only, where both initial and propagated probabilities are high.

Next, we define the interactive restart rule that encourages interactions between the agents. Specifically, at each iteration θ, we update the interactive restart distribution $\mathbf{y}_{\mathrm{f}}^{(\theta)} = [y_{\mathrm{f},1}^{(\theta)}, \cdots, y_{\mathrm{f},N}^{(\theta)}]^{\mathrm{T}}$ as follows. By considering the probabilities at neighboring nodes, we compute the amount of interactive restart at node i via

$$y_{\mathrm{f},i}^{(\theta)} = \frac{p_{\mathrm{f},i}^{(\theta)} + \sum_{j\in\mathcal{N}_i} w_{ij} p_{\mathrm{f},j}^{(\theta)}}{(p_{\mathrm{f},i}^{(\theta)} + \sum_{j\in\mathcal{N}_i} w_{ij} p_{\mathrm{f},j}^{(\theta)}) + (p_{\mathrm{b},i}^{(\theta)} + \sum_{j\in\mathcal{N}_i} w_{ij} p_{\mathrm{b},j}^{(\theta)})} \tag{9}$$

where $\mathcal{N}_i$ is the set of connected neighbors of node i, and w_{ij} denotes the LAB color affinity between node i and node j. A high amount of foreground restart is assigned to a node if its neighbors, which yield similar features, have higher foreground probabilities than background ones. While the repulsive restart rule in [27] determines the restart amount at each node using the probabilities on that node only, we also exploit the probabilities on the neighboring nodes to suppress noisy restarts.



(a) $\mathbf{p}_{\mathrm{f}}^{(1)}$ (b) $\mathbf{p}_{\mathrm{f}}^{(2)}$ (c) $\mathbf{p}_{\mathrm{f}}^{(3)}$ (d) π_{f} (e) Input frame

(f) $\mathbf{p}_{\mathrm{b}}^{(1)}$ (g) $\mathbf{p}_{\mathrm{b}}^{(2)}$ (h) $\mathbf{p}_{\mathrm{b}}^{(3)}$ (i) π_{b} (j) $\mathbf{s}$

Figure 3: The distributions of the foreground agent $\mathbf{p}_{\mathrm{f}}^{(\theta)}$ and the background agent $\mathbf{p}_{\mathrm{b}}^{(\theta)}$ are visualized for the first three MRW iterations. The segmentation labels in $\mathbf{s}$ are determined by comparing the two stationary distributions, π_{f} and π_{b}.

Then, we set the foreground restart distribution $\mathbf{r}_{\mathrm{f}}^{(\theta)}$ at each iteration θ in (6), by combining (8) and (9),

$$\mathbf{r}_{\mathrm{f}}^{(\theta)} = (1-\delta)\cdot\mathbf{q}_{\mathrm{f}} + \delta\cdot\mathbf{y}_{\mathrm{f}}^{(\theta)}, \tag{10}$$

where δ is a weight to blend the time-varying distribution $\mathbf{y}_{\mathrm{f}}^{(\theta)}$ with the invariant one $\mathbf{q}_{\mathrm{f}}$. Using this hybrid restart rule, we perform the MRW iterations to obtain the stationary distributions π_{f} and π_{b} of the foreground and background agents, respectively.

Figure 3 visualizes the evolvement of the foreground and background distributions in the MRW simulation. As the iteration goes on, the foreground and background agents yield more mutually exclusive probabilities. Eventually, the stationary distributions, π_{f} and π_{b}, accurately delineate the target object and the background regions.

Segmentation Labeling: After obtaining the foreground and background stationary distributions π_{f} and π_{b}, we label each superpixel by comparing the two probabilities on it. As the foreground and the background have different sizes in general, the corresponding distributions should be scaled for adequate comparison. Thus, we scale the stationary distributions by their the maximum probabilities, respectively. Then, we decide segmentation label $s_i^{(t)}$ of node i at frame t by

$$s_i^{(t)} = \begin{cases} 1 & \text{if } \pi_{\mathrm{f},i}/\max_j(\pi_{\mathrm{f},j}) > \pi_{\mathrm{b},i}/\max_j(\pi_{\mathrm{b},j}), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

3.4 Pixel-wise Markov Random Field Optimization

Optionally, we further refine the segmentation results at the superpixel level into those at the pixel level. We follow the MRF optimization in [17]. Specifically, we construct RGB color and positional Gaussian mixture models for the foreground and background regions. Also, we encourage the spatiotemporal neighbors to have the same labels using a pairwise term. We employ the graph-cut algorithm [4] to perform the optimization. Note that the Gaussian mixture models and the pairwise term are computed using all pixels in all frames. Hence, this refinement is offline, while the other processes of the proposed algorithm are online.

Table 1: The average numbers of mislabelled pixels per frame on the SegTrack dataset [39]. Lower values are better. The best and the second best results among the fixed parameter methods are boldfaced and underlined, respectively.

Sequence	Tuned parameters			Fixed parameters									
	[8]	[34]	[45]	[8]	[39]	[8]	[40]	[13]	[40]	[24]	[8]	MRW	MRW-MRF
Birdfall	508	186	163	454	252	468	1204	466	243	189	481	215	**169**
Cheetah	855	535	806	1216	1142	1501	2765	1431	**391**	1170	2825	740	654
Girl	1200	761	1904	1755	1304	1705	10505	6338	1935	2883	7790	1527	**1203**
Monkeydog	412	358	342	683	563	736	2466	809	497	**333**	5361	428	337
Parachute	296	249	275	502	235	404	2369	1028	**187**	228	3105	331	198
Penguin	1736	355	571	6627	1705	19310	9078	6239	903	**443**	11669	883	1459

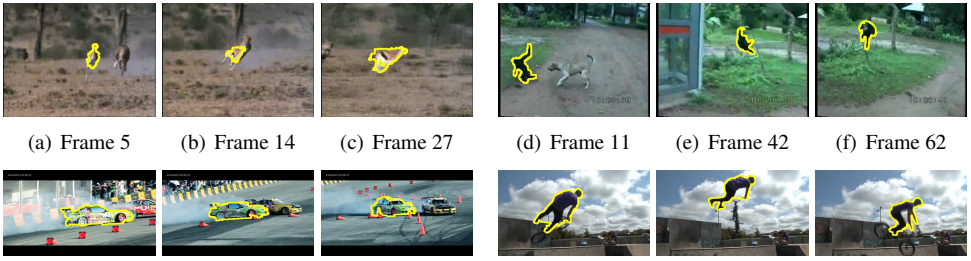
Table 2: IoU scores on the SegTrack v2 dataset [29]. The best and the second best results among the fixed parameter methods are boldfaced and underlined. (-) indicates segmentation failures, and the average scores (*) are computed without the failed sequences.

Target object (Number of frames)	Tuned parameters	Fixed parameters					
	[45]	[13]	[40]	[8]	[34]	MRW	MRW-MRF
Girl (21)	84.6	53.6	52.4	62.0	62.4	83.0	**86.6**
Birdfall (30)	78.7	56.0	32.5	36.4	9.36	61.6	**68.3**
Parachute (51)	94.4	85.6	69.9	59.3	92.5	91.1	**94.7**
Cheetah-Deer (29)	66.1	46.1	33.1	38.7	17.7	56.4	**63.0**
Cheetah-Cheetah (29)	35.3	**47.4**	14.0	19.7	0.69	30.0	33.0
Monkeydog-Monkey (71)	82.2	61.0	22.1	25.7	4.96	74.2	**77.3**
Monkeydog-Dog (71)	21.1	**18.9**	10.2	3.83	9.03	17.1	16.5
Penguin-#1 (42)	94.2	54.5	20.8	40.1	80.2	**88.1**	80.6
Penguin-#2 (42)	91.8	67.0	20.8	37.9	73.1	**87.4**	80.0
Penguin-#3 (42)	91.9	7.59	10.3	31.2	46.3	**84.8**	74.0
Penguin-#4 (42)	90.3	54.3	13.0	30.2	51.6	**79.2**	74.5
Penguin-#5 (42)	76.3	29.6	18.9	10.7	53.7	**74.2**	61.6
Penguin-#6 (42)	88.7	2.09	32.3	35.0	70.1	**86.7**	81.1
Drift-#1 (74)	67.3	62.6	43.5	57.2	42.9	**80.8**	80.7
Drift-#2 (74)	63.7	21.8	11.6	13.8	11.1	**37.8**	**37.8**
Hummingbird-#1 (29)	58.3	11.8	28.8	25.1	14.0	48.9	**56.6**
Hummingbird-#2 (29)	50.7	-	**45.9**	44.2	36.8	41.6	41.6
Frog (279)	56.3	14.5	45.2	38.8	**63.4**	42.8	48.4
Worm (243)	79.3	36.8	27.4	44.3	**72.4**	67.9	60.6
Soldier (32)	81.1	70.7	43.0	54.2	**71.9**	60.8	69.7
Monkey (31)	86.0	73.1	61.7	58.7	76.1	**86.1**	73.2
Bird of Paradise (98)	93.0	5.10	44.3	46.5	82.3	**85.3**	83.2
BMX-Person (36)	88.9	2.04	27.9	36.0	44.5	**78.8**	77.2
BMX-Bike (36)	5.70	-	6.04	3.86	0.00	**8.93**	6.94
Mean per object	71.8	40.1*	30.7	35.6	45.3	**64.7**	63.6
Mean per sequence	72.2	41.0*	37.0	40.4	48.8	64.2	**64.7**

4 Experimental Results

We evaluate the proposed algorithm on the SegTrack [39] and SegTrack v2 [29] datasets. Both datasets provide video sequences and the corresponding ground-truth label maps. We set σ^2 to 0.01 for both color and motion features in (1). In (2), we set λ_c, λ_m, and λ_g to 1, 2, 1, respectively. The restart probability ε is set to 0.4 in (6) and (7), and δ is set to 0.3 in (10). Notice that we fix all parameters. This fixation is desirable, since it is important to provide reliable performance without manual tuning. We will make the code publicly available.

SegTrack [39] is a subset of SegTrack v2 [29]. We measure the average number of mislabelled pixels per frame to evaluate the segmentation performance on the SegTrack dataset, as done in [24, 36, 39, 40, 45]. We compare the proposed algorithm with eleven algorithms: three tracking-by-segmentation methods [8, 18, 41] and eight semi-supervised video object segmentation methods [6, 7, 9, 24, 36, 39, 40, 45]. Also, we check whether the algorithms tune parameters for each sequence or not, and categorize them into *tuned parameters* or *fixed*



(a) Frame 5 (b) Frame 14 (c) Frame 27 (d) Frame 11 (e) Frame 42 (f) Frame 62

(g) Frame 7 (h) Frame 44 (i) Frame 70 (j) Frame 3 (k) Frame 22 (l) Frame 36

Figure 4: Qualitative results. Segmentation boundaries are depicted in yellow. (a)∼(c), (d)∼(f), (g)∼(i), and (j)∼(l) are from the "Cheetah-Deer," "Monkeydog-Monkey," "Drift-#1," and "BMX-Person," respectively.

parameters, respectively. Then, we do not directly compare the proposed algorithm with the methods that tune parameters for each sequence. Table 1 lists the performances of the proposed algorithm and the conventional algorithms on the SegTrack dataset. The scores of the conventional algorithms are from [7, 24, 36, 40, 45]. MRW and MRW-MRF denote the proposed algorithm without and with MRF optimization, respectively. Notice that the proposed algorithm outperforms most conventional algorithms on most sequences.

SegTrack v2 [29] consists of 14 video sequences and includes 24 objects. The sequences are challenging due to motion blur ("girl" and "hummingbird"), ambiguous boundaries ("birdfall," "cheetah," "penguin," and "soldier"), and object deformation ("cheetah," "monkeydog," "drift," "hummingbird," "frog," "worm," and "BMX"). While SegTrack consists of short video clips with 21∼71 frames, SegTrack v2 includes longer clips with up to 279 frames. We compute the intersection over union (IoU) score between a segmentation result and the ground-truth label map, as done in [45].

In Table 2, we compare the proposed algorithm with the five conventional algorithms in [8, 18, 36, 41, 45]. The IoU scores of the conventional algorithms are from [45], except for [36]. For [36], we use the default setting in their source code. We see that the proposed algorithm significantly outperforms all conventional algorithms on most sequences. More specifically, the proposed algorithm ranks 1st on 18 objects among the 24 objects. Also, we report two average scores, *mean per object* and *mean per sequence*. The mean per object assesses a video object segmentation algorithm by averaging its scores on all 24 objects. On the other hand, to compute the mean per sequence, we first calculate the mean score of objects within each sequence, and then average them. The proposed algorithm outperforms the conventional ones in terms of both metrics. The MRF refinement improves the mean per sequence performance to a small degree, but degrades the mean per object performance slightly.

Figure 4 shows qualitative segmentation results of the proposed algorithm. Even though the target objects suffer from superposition ("cheetah" and "drift"), ambiguous boundaries ("cheetah"), and deformation (all four examples), the resultant segment tracks delineate them accurately. We present more segmentation results in supplementary materials.

Initial Segmentation Method: We analyse segmentation qualities according to the initial interactive segmentation algorithm. For the "Bird of Paradise" sequence, we use GrabCut [37] instead of [22] in the first frame. Figure 5 exemplifies the segmentation results of the proposed algorithm according to the initial segmentation methods. While [37] misleads the

(a) User input (b) Frame 1 (c) Frame 9 (d) Frame 17 (e) Frame 25

Figure 5: Segmentation results of the proposed algorithm according to the interactive segmentation method in the first frame. The upper and lower rows are the resultant segmentations using [22] and [37], respectively. Segmentation boundaries are depicted in yellow. The frames are from the "Bird of Paradise."

(a) Frame 107 (b) Frame 219 (c) Frame 354 (d) Frame 439 (e) Frame 567

Figure 6: A segment track on the long sequence. Segmentation boundaries are depicted in yellow. The frames are from the "Bike Riding."

proposed algorithm initially, the target object is recovered in the later frames.

Long Sequence: We apply the proposed algorithm to a long sequence to observe evolvements of segments as time goes on. Among the sequences in the VidSeg dataset [25], we use the "Bike Riding" sequence, which consists of 631 frames. Figure 6 displays segmentation results on it. The proposed algorithm even yields robust segments in the later frames.

Running Time Analysis: We count a running time of the proposed algorithm by seconds per frame (SPF). For the "Girl" sequence at 400×320 resolution, we measure SPF of the proposed method on a PC with a 2.2GHz CPU. The proposed algorithm runs at 26.8SPF.

5 Conclusions

We proposed a novel semi-supervised video object segmentation algorithm. We first estimated initial distributions of the foreground and background. Then, we simulated MRW using a hybrid of the inference restart rule and the interactive restart rule. We performed these processes from the second to the last frames to extract a segment track of a target object. Furthermore, we optionally refined the segment track by performing the MRF optimization. Experimental results demonstrated that the proposed algorithm significantly outperforms the state-of-the-art conventional algorithms [8, 18, 36, 41] on the SegTrack v2 dataset [29].

Acknowledgement

This work was supported partly by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A10055037), and partly by the MSIP, Korea, under the ITRC support program supervised by the Institute for Information &communications Technology Promotion (No. IITP-2016-R2720-16-0007).

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] 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.

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

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

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

[6] I. Budvytis, V. Badrinarayanan, and R. Cipolla. Semi-supervised video segmentation using tree structured graphical models. In *CVPR*, pages 2257–2264, 2011.

[7] I. Budvytis, V. Badrinarayanan, and R. Cipolla. MoT-Mixture of trees probabilistic graphical model for video segmentation. In *BMVC*, pages 1–11, 2012.

[8] Z. Cai, L. Wen, Z. Lei, N. Vasconcelos, and S. Z. Li. Robust deformable and occluded object tracking with dynamic graph. *IEEE Trans. Image Process.*, 23(12):5497–5509, 2014.

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

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

[11] S. Duffner and C. Garcia. PixelTrack: A fast adaptive algorithm for tracking non-rigid objects. In *ICCV*, pages 2480–2487, 2013.

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

[13] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, pages 1–12, 2014.

[14] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. JumpCut: Non-successive mask transfer and interpolation for video cutout. *ACM Trans. Graphics*, 34(6):195, 2015.

[15] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, pages 1–11, 2011.

[16] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, 59(2):167–181, 2004.

[17] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, pages 760–774. 2012.

[18] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, pages 81–88, 2011.

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

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

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

[22] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, pages 3129–3136, 2010.

[23] M. Isard and A. Blake. CONDENSATION-Conditional density propagation for visual tracking. *Int. J. Comput. Vis.*, 29(1):5–28, 1998.

[24] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, pages 656–671. 2014.

[25] 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.

[26] A. Khoreva, F. Galasso, M. Hein, and B. Schiele. Classifier based graph construction for video segmentation. In *CVPR*, pages 951–960, 2015.

[27] 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.

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

[29] 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.

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

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

[32] 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.

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

[34] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *Proc. ACM SIGKDD*, pages 653–658, 2004.

[35] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, pages 1777–1784, 2013.

[36] S. A. Ramakanth and R. V. Babu. SeamSeg: Video object segmentation using patch seams. In *CVPR*, pages 376–383, 2014.

[37] 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.

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

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

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

[41] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330, 2011.

[42] T. Wang, B. Han, and J. Collomosse. TouchCut: Fast image and video segmentation using single-touch interaction. *Comput. Vis. Image Understand.*, 120:14–30, 2014.

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

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

[45] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. JOTS: Joint online tracking and segmentation. In *CVPR*, pages 2226–2234, 2015.

[46] 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.

[47] 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.