Video Object Segmentation via Cellular Automata Refinement

Ding-Jie Chen, Hwann-Tzong Chen, and Long-Wen Chang
Department of Computer Science
National Tsing Hua University, Taiwan

Abstract

We present a robust algorithm for unsupervised video object segmentation using foreground prior estimated from optical flow. Optical flow is an important cue for predicting the region of foreground object in a video. However, the estimation of flow is inherently inaccurate near the occluded object boundaries. We show that, even though the foreground prior might be unreliable due to the inaccurately estimated flow, Cellular Automata can be used to refine the foreground prior and thus is helpful to define the energy function for better segmentation accuracy. The experiments on the recently proposed DAVIS dataset show that our method performs favorably against the existing ones.

1. Introduction

Video object segmentation is a fundamental computer vision task of separating the foreground objects from the background region in a video. It is important for a wide range of applications, including video editing, scene understanding [17], object class model learning [28], video summarization, and action recognition [14]. The rapidly growing number of video sequences on the web draws more attention to this labeling task.

Typical video object segmentation tasks have different levels of user intervention: A user may provide only the video for unsupervised segmentation [3, 5, 13, 18, 20, 24, 25, 42], or may additionally annotate the object position in some frames for semi-supervised segmentation [15, 22, 29, 31, 36, 37, 40, 41]. Various segmentation algorithms have been proposed, for example, tracking based [40, 41], clustering based [3, 15, 24], ranking based [18, 20, 42], or propagation based [29, 36, 37] at pixel level, superpixel level, or object level. To address the video segmentation task, the temporal information is one important cue for maintaining the segmentation consistency over the whole video or estimating the potential foreground objects. The most widely used technique for this purpose is optical flow estimation, which models the motion of pixels between two frames.

We aim to address the video object segmentation task in fully unsupervised manner. We propose a foreground prior refinement approach and apply it to the unsupervised video object segmentation task. Under the unsupervised constraint, the segmentation task usually treats an image region that has different motion from its surrounding regions as a potential foreground object. Hence, the quality of optical flow is usually proportional to the accuracy of foreground object estimation. However, optical flow is often deteriorated by large displacements or occlusions [4] and thus might result in unsatisfying foreground prior. Here we employ Cellular Automata to design a foreground prior refining method that makes the prior estimation more robust to the inaccurate optical flow around object boundaries. After being refined with Cellular Automata, the foreground prior is enforced to be consistent within the homogeneous region and fit to the object boundaries. Based on the refined prior, we are able to ensure the quality of the corresponding energy function for figure-ground segmentation. We then build a spatio-temporal graphical model of the entire video, and extract all refined foreground object priors from video frames to learn the appearance Gaussian mixture model (GMM). The learned GMM is thus used to gradually refine the segmentations.

2. Related Work

The literature related to video object segmentation can be divided into three categories with respect to the level of required supervision: unsupervised methods, semi-supervised methods, and supervised methods.

2.1. Unsupervised Methods

Unsupervised video segmentation methods [3, 5, 13, 18, 20, 24, 25, 42] have no requirement of any user annotation and usually assume that different objects have different appearances or motions. Based on the clustering concept, the methods of [3, 24] track keypoints to form trajectories in video sequence and thus cluster these trajectories to separate the keypoints of the object from the background region. Brutzer *et al.* [5] assume that the appearance of background changes slowly over time and thus consider the rapidly changing pixels to be the foreground. With the guidance of object proposals [6, 8], the methods of [13, 18, 20, 42] rank several candidate combinations from object-like image regions to reason out the potential object segmentations. Papazoglou and Ferrari [25] assume that the foreground object has different motion with respect to its surrounding regions, and hence separate the object via the graph cuts technique. The unsupervised methods are suited for large scale datesets; however, the segmentation quality may degrade if the underlying assumption does not hold.

2.2. Semi-supervised Methods

Semi-supervised video segmentation methods [15, 22, 29, 31, 36, 37, 40, 41] require the user annotation in one or few frames and then propagate the annotation to the entire video sequence. Grundmann *et al.* [15] represent a video sequence as a set of supervoxels, and then the supervoxels belonging to the foreground object indicating by the user annotations are grouped to represent the object. Marki *et al.* [22] minimize the energy in bilateral space to approximates non-local connections for separating the object from the background. Given the manually annotations on a few frames, several methods [29,36,37] can propagate the annotations to all other frames mainly based on the optical flow. Ramakanth and Babu [31] cast the segmentation task as an optimization problem, which defines the energy of a graph over the entire video. Given some annotated superpixels, Wen *et al.* [40] and Yang *et al.* [41] carry out the segmentation via tracking the object segments. The semi-supervised methods usually focus on improving the quality of annotation propagation, but how to make the annotation easier to get is a potential issue.

2.3. Supervised Methods

Supervised video segmentation methods [10, 19, 38] require user annotations while segmenting. Since the user frequently corrects the segmentation results, a high segmentation quality is guaranteed. The supervised methods are suited for specific scenarios, for instance, the professional rotoscoping in the film industry.

In sum, the unsupervised methods enable the processing of large amounts of video sequences without human intervention, but the semi-supervised video segmentation methods have relatively better segmentation accuracy. Although the supervised methods achieve the best segmentation quality, the cost of time-consuming interaction is unavoidable.

3. Approach

The goal of our unsupervised video segmentation approach is to segment objects that move differently with respect to the surroundings. Our approach includes two main phases, namely *foreground prior estimation* and *figure-ground segmentation*. In the first phase, we first estimate the motion boundaries from the optical flow cue and infer the foreground prior region of each frame. Then, Cellular Automata is used to refine all foreground prior regions. In the second phase, we aim to collect all *refined* foreground prior regions to construct the global appearance Gaussian mixture model for separating the object region from the background region via graph cuts. Note that, while defining the appearance data term and the prior data term of energy function, the foreground prior regions and the propagated foreground prior are also refined via Cellular Automata.

3.1. Foreground Prior Estimation

This phase estimates a foreground prior region based on the motion cue. The optical flow of each pair of consecutive frames are calculated first, and then the corresponding per-frame motion boundaries are defined and refined to reason out the foreground prior region.

3.1.1 Motion Boundaries

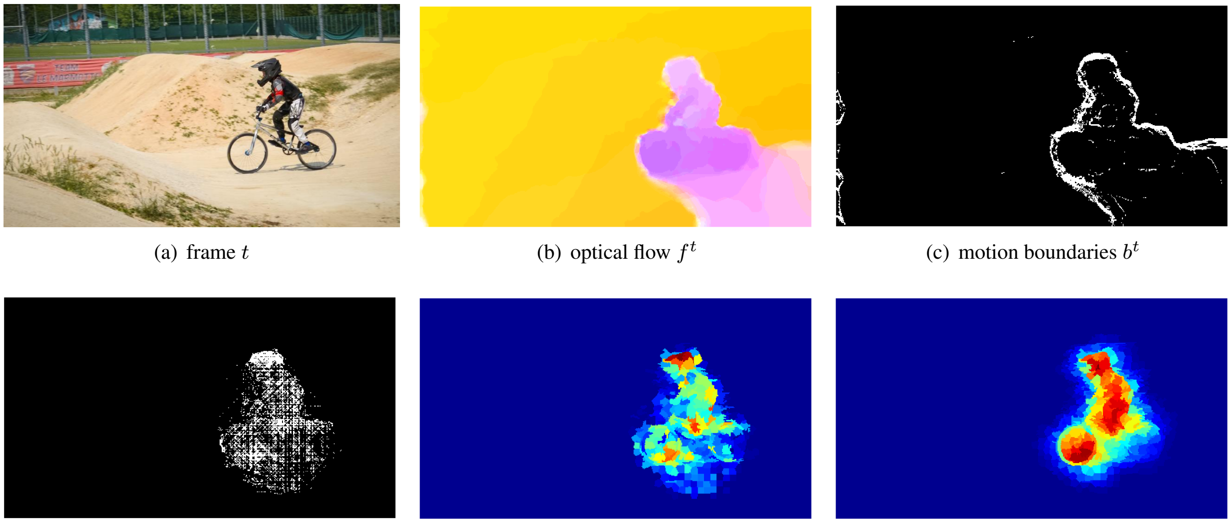
We first compute the optical flow f^t [2, 3, 33] of each pair of consecutive frames t and $t+1$. The motion boundary b_p^t [25, 34] of frame t, which indicates that the image pixel p has different motion with respect to its neighboring pixels, is defined as

$$b_p^t = \begin{cases} 1, & \text{if } \hat{b}_p^t \cdot \tilde{b}_p^t > 0.5 \ , \\ 0, & \text{otherwise}\,, \end{cases} \qquad (1)$$

where $\hat{b}_p^t = 1 - \exp(-\theta_1 \|\nabla f_p^t\|)$ means the moving difference of the motion boundary at pixel p in frame t, $\|\nabla f_p^t\|$ denotes the magnitude of the flow vector f_p^t at pixel p in frame t, $\tilde{b}_p^t = 1 - \exp(-\theta_2 \max_{q \in \mathcal{N}}(\angle f_{pq}^t))$ means the orientation difference of the motion boundary at pixel p in frame t, $\mathcal{N}$ denotes the 8-connected neighborhood of p, $\angle f_{pq}^t$ denotes the angle between f_p^t and f_q^t. We empirically set and fix the parameters $\theta_1 = 0.7$ and $\theta_2 = 1$ to control the steepness of their corresponding functions in all experiments. The Eq. (1) is used to indicate the pixel that has different motion speed and motion direction from its surroundings. Fig. 1(c) shows one example of motion boundaries b^t.

3.1.2 Foreground Prior

In computational geometry [11], the point-in-polygon problem deals with the question of determining whether a pixel is inside a polygon. Papazoglou and Ferrari propose the *integral intersections* algorithm [25] to identify whether a pixel is inside the incomplete boundaries. Given the motion boundary b^t, we employ the *integral intersections* algorithm to obtain a binary map m^t. We then represent each frame t as a set of superpixels $\mathcal{S}^t = \{s_1^t, s_2^t, ..., s_{|\mathcal{S}^t|}^t\}$ using the SLIC algorithm [1] with roughly 2,000 superpixels. For each superpixel s_i, we average the pixels $\{m_p^t | p \in s_i\}$



(a) frame t

(b) optical flow f^t

(c) motion boundaries b^t

(d) binary map m^t

(e) foreground prior M^t

(f) refined foreground prior $\mathbf{M}^t$

Figure 1. An example of foreground prior estimation. (a) One frame of the sequence `bmx-bumps`. (b) Optical flow computed using [3] from frame t to $t+1$. (c) The estimated motion boundaries using (b). (d) The calculated binary map m^t using (c). (e) The superpixel-level foreground prior M^t corresponding to (d). (f) The refined foreground prior using (e). The foreground prior inside the object is clearer, in comparison with the initial foreground prior M^t.

to represent its foreground prior belief, and thus form the superpixel-level foreground prior M^t. Fig. 1(d)-(e) show examples of binary map m^t and foreground prior M^t.

3.1.3 Foreground Prior Refinement

As can be observed in the Fig. 1(b), the quality of optical flow is often degraded by large displacements or occlusions, particularly around object boundaries. Here, we propose the following model to make the foreground prior more fit to the object boundaries.

Cellular Automata [23] is a self-organizing evolution model. The model can be used to propagate information [21, 30] via exploiting the intrinsic relevance among similar neighbors. The Cellular Automata model has a set of cells with discrete states evolving with time. We use superpixels as cells, and the superpixel-level foreground prior defines the initial state of each cell.

To simulate the evolving state of each cell, we employ the updating rule as [21, 30] for updating the states of all cells simultaneously. The rule is defined as

$$\mathbf{S}^{r+1} = \mathbf{C}\cdot\mathbf{S}^r + (\mathbf{I}-\mathbf{C})\cdot\mathbf{A}\cdot\mathbf{S}^r\,, \tag{2}$$

where $\mathbf{S}^r$ and $\mathbf{S}^{r+1}$ respectively denote the current state and the next state of size $|\mathcal{S}^t|$-by-1, and $\mathbf{I}$ denotes the $|\mathcal{S}^t|$-by-$|\mathcal{S}^t|$ identity matrix.

The *impact factor* matrix $\mathbf{A} = [a_{ij}]_{|\mathcal{S}^t|\times|\mathcal{S}^t|}$ defines the influence power of each cell s_i to its neighboring cell s_j. In general, the influence power a_{ij} of the cell s_i to the cell s_j is defined proportional to the feature similarity between s_i and s_j. We define a_{ij} as

$$a_{ij} = \begin{cases} e^{-\|\mathbf{f}_i-\mathbf{f}_j\|_2^2/\sigma^2}\,, & \text{if } j\in\Omega(i)\,, \\ 0, & i=j \text{ or otherwise}\,, \end{cases} \tag{3}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ denote the mean CIE LAB color of superpixel, and $\Omega(i)$ denotes the adjacent superpixels of s_i. We set $\sigma^2 = 0.1$. Note that, the matrix $\mathbf{A}$ should be row-normalized to ensure that the row sum is one.

The *coherence* matrix $\mathbf{C} = diag\{c_1, c_2, \cdots, c_{|\mathcal{S}^t|}\}$ defines the strength of coherence of each cell towards its current state. In general, the strength of coherence c_i of the cell s_i is defined to be inversely proportional to the similarity of the most similar neighboring cell. To ensure $c_i \in [\alpha, \alpha+\beta]$, we define c_i as

$$c_i = \alpha + \frac{1/\max(a_{ij}) - \min(c_j)}{\max(c_j) - \min(c_j)}\cdot\beta\,. \tag{4}$$

We set $\alpha = 0.2$ and $\beta = 0.6$ as in [21, 30].

To refine the foreground prior of frame t via Cellular Automata, we set the initial state $\mathbf{S}^r$ in round $r = 0$ as the foreground prior M^t. The final state after R rounds is denoted as $\mathbf{S}^R$. Fig. 1(f) shows an example of refined foreground prior $\mathbf{M}^t$.

3.2. Figure-ground Segmentation

This phase defines an energy function based on the foreground prior. We first represent a video sequence as a superpixel-level spatio-temporal graph, and then the energy function for the graph is defined with respect to the foreground prior over frames. Finally, figure-ground segmentation is achieved using graph cuts.

3.2.1 Spatio-temporal Graph

Given a video with T frames, we define its spatio-temporal graph as a weighted connected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \omega)$ with the vertex set $\mathcal{S} = \mathcal{S}^1 \cup \mathcal{S}^2 \cup \dots \cup \mathcal{S}^T$ and the edge set $\mathcal{E}$. Each edge $e_{ij} \in \mathcal{E}$ denotes the adjacency relationship between superpixels s_i and s_j. Note that, two superpixels $s_i^t \in \mathcal{S}^t$ and $s_j^{t+1} \in \mathcal{S}^{t+1}$ are adjacent if s_i^t can cover s_j^{t+1} after being warped by the optical flow. The weighting function $\omega : \mathcal{E} \rightarrow [0, 1]$ is defined as

$$\omega_{ij} = e^{-\|\mathbf{f}_i - \mathbf{f}_j\|_2^2/\sigma^2} \,, \tag{5}$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ denote the mean CIE LAB color of superpixel.

3.2.2 Energy Function

Segmenting a video $\{s_i^t\}_{t,i}$ is equal to a labeling $\mathcal{L} = \{l_i^t\}_{t,i}$ among all superpixels. This work uses the binary label $l_i^t \in \{0, 1\}$. For evaluating a labeling, we define the energy function as

$$E(\mathcal{L}) = E^A + \alpha_1 E^P + \alpha_2 E^S + \alpha_3 E^T \,. \tag{6}$$

The appearance data term E^A evaluates how likely a superpixel belongs to the foreground or background. The prior data term E^P encourages foreground labeling in areas where independent motion has been observed. The spatial smoothness term E^S and the temporal smoothness term E^T encourage spatial and temporal smoothness, respectively. The parameters $\alpha_1, \alpha_2, \alpha_3$ are the weights for different terms. We set $\alpha_1 = 1.5$, $\alpha_2 = 2{,}000$, and $\alpha_3 = 1{,}000$.

Appearance Term: The appearance data term consists of two Gaussian mixture models in RGB color space. We consider all other superpixels to define the potential of superpixel s_i^t in frame t with respect to the foreground GMM as

$$E^A = \sum_{i,t} \exp(-\beta_1 \cdot (t - t')^2) \cdot \mathbf{M}_i^{t'} \,, \tag{7}$$

where $\exp(\cdot)$ computes the influence of $s_i^{t'}$ over time, t' denotes the other frame, and $\mathbf{M}_i^{t'}$ denotes the *refined* foreground prior of the superpixel s_i in frame t'. The potential of s_i^t with respect to the background GMM is defined analogously, *i.e.*, the $\mathbf{M}_i^{t'}$ in Eq. (7) is replaced with $1 - \mathbf{M}_i^{t'}$. We set $\beta_1 = 0.0001$.

Prior Term: We define the prior data term to accumulate the *refined* superpixel-level foreground prior over the video. The propagation equation is defined as

$$\mathbf{P}_j^{t+1} = \mathbf{P}_j^{t+1} + \beta_2 \frac{\sum_i \omega(s_i^t, s_j^{t+1}) \cdot \psi(s_i^t)}{\sum_i \omega(s_i^t, s_j^{t+1})} \mathbf{P}_i^t \,, \tag{8}$$

where the value of $\mathbf{P}_i^t$ is initialed with $\mathbf{M}_i^t$, $\omega(s_i^t, s_j^{t+1})$ means the weight ω_{ij} defined in Eq. (5), ψ down-weights the propagation power of s_i^t if it covers the strong flow gradients. In [25], the function ϕ, which computes the overlap ratio between the two superpixels among two consecutive frames, is used to define their propagation equation. However, replacing the function ϕ with the function ω shows better segmentation quality in our experiment. We set $\beta_2 = 20$. The forward propagation and backward propagation of Eq. (8) respectively define the prior potential $\hat{\mathbf{P}}_i^t$ and $\tilde{\mathbf{P}}_i^t$. The prior data term can thus be defined as

$$E^P = \sum_{i,t} \frac{\hat{\mathbf{P}}_i^t + \tilde{\mathbf{P}}_i^t + \mathbf{M}_i^t}{3} \,. \tag{9}$$

Smoothness Term: The spatial smoothness term E^S is defined on the edge of the adjacent superpixels in the same frame, and the temporal smoothness term E^T is defined on the edge of the adjacent superpixels among two neighboring frames. We follow the contrast-modulated Potts potential [18, 25, 32] to define E^S and E^T as

$$E^S = \sum_{(i,j)\in\mathcal{E},t,l_i^t \neq l_j^t} d_1(s_i^t, s_j^t)^{-1} \exp(-\beta_3 d_2(s_i^t, s_j^t)) \,, \tag{10}$$

$$E^T = \sum_{(i,j)\in\mathcal{E},t,l_i^t \neq l_j^{t+1}} \phi(s_i^t, s_j^{t+1}) \exp(-\beta_4 d_2(s_i^t, s_j^{t+1})) \,, \tag{11}$$

where d_1 is the Euclidean distance between the centers of two superpixels, d_2 is the squared Euclidean distance between the mean RGB color of two superpixels, and ϕ [25] is the overlap ratio guided by optical flow between the two superpixels. The parameters β_3 and β_4 are set as [25].

Notice that, $\hat{\mathbf{P}}_i^t$, $\tilde{\mathbf{P}}_i^t$, and $\mathbf{M}_i^t$ are refined via Cellular Automata while defining the energy function. Therefore, the figure-ground segmentation is obtained by minimizing the energy function as GrabCut [32]. Our GMM model uses ten mixture components in RGB color space for each label, and we additionally use the guided filter [16] to reduce the under-segmentation error derived from over-segmentation.

4. Experimental Results

We compare our approach with several popular unsupervised video segmentation methods: MSG [3], NLC [9], TRC [12], KEY [18], FST [25], CVOS [35], and SAL [39]. The evaluations are performed with respect to the two metrics suggested in the DAVIS dataset [26]: *region similarity* ($\mathcal{J}$) and *contour accuracy* ($\mathcal{F}$). In our experiments, all parameters are fixed without further tuning.

The comparison is evaluated on the DAVIS (Densely Annotated VIdeo Segmentation) dataset [26], which contains 50 high-resolution sequences of 3,455 frames. This dataset covers a wide range of object segmentation challenges.

Table 1. Quantitative comparison (%) of region similarity ($\mathcal{J}$) and contour accuracy ($\mathcal{F}$) on the DAVIS dataset [26]. The 'mean' is the average dataset error. The 'recall' measures the fraction of sequences scoring higher than a threshold. The 'decay' quantifies the performance loss (or gain) over time. For rows with an upward pointing arrow, the higher numbers are better, and vice versa for rows with a downward pointing arrow. The best two scores among the unsupervised methods are colored in red and green. The scores of the semi-supervised methods that are better than ours are emphasized in boldface.

	Unsupervised								Semi-Unsupervised					
	MSG	NLC	TRC	KEY	FST	CVOS	SAL	Ours	TSP	SEA	HVS	JMP	FCP	BVS
mean $\mathcal{J}\uparrow$	54.3	64.1	50.1	56.9	57.5	51.4	42.6	**66.4**	35.8	55.6	59.6	60.7	63.1	**66.5**
mean $\mathcal{F}\uparrow$	52.5	59.3	47.8	50.3	53.6	49.0	38.3	**61.1**	34.6	53.3	57.6	58.6	54.6	**65.6**
recall $\mathcal{J}\uparrow$	63.6	73.1	56.0	67.1	65.2	58.1	38.6	**81.2**	38.8	60.6	69.8	69.3	77.8	76.4
recall $\mathcal{F}\uparrow$	61.3	65.8	51.9	53.4	57.9	57.8	26.4	**73.2**	32.9	55.9	71.2	65.6	60.4	**77.4**
decay $\mathcal{J}\downarrow$	2.8	8.6	5.0	7.5	4.4	12.7	8.4	4.9	38.5	35.5	19.7	37.2	**3.1**	26.0
decay $\mathcal{F}\downarrow$	5.7	8.6	6.6	7.9	6.5	13.8	7.2	**5.2**	38.8	33.9	20.2	37.3	**3.9**	23.6

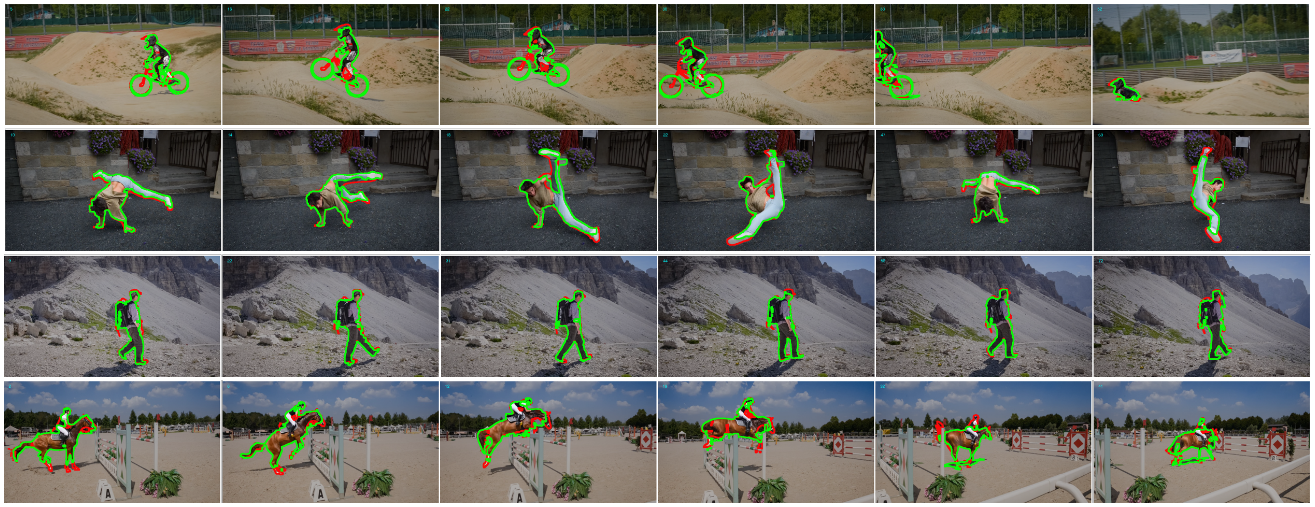


Figure 2. Qualitative video segmentation results from some sequences of DAVIS dataset [26]. The red contours depict the ground truth boundaries. The green contours depict the boundaries of our segmentations.

Table. 1 summarizes the average performance of each method over the entire dataset. As can be seen in Table. 1, our method outperforms all other unsupervised video segmentation methods excepts on the decay $\mathcal{J}$ evaluation. Our method achieves the best performance on the mean $\mathcal{J}$, recall $\mathcal{J}$, mean $\mathcal{F}$, recall $\mathcal{F}$, and decay $\mathcal{F}$, which demonstrates the superior performance of our method. We additionally provide the results of some semi-supervised methods for reference: TSP [7], SEA [31], HVS [15], JMP [10], FCP [27], and BVS [22]. The results also show the favorable quality of our method even if it is compared with semi-supervised methods.

Fig. 2 shows some qualitative results of the proposed video segmentation method. The challenging video sequences shown in Fig. 2 demonstrate that our method is robust to some intriguing scenarios such as complex objects and fast-motion. Regarding the computational cost, our approach takes about 3 seconds per frame for DAVIS dataset (480p) on an Intel Core i7-4770 3.40 GHz CPU, excluding the optical flow computation.

5. Conclusion

We have shown that using Cellular Automata to refine the foreground prior has a significant advantage to make the estimation of foreground prior more robust to the inaccurate optical flow. With the aid of the refined foreground prior, the energy function can be formulated to yield higher quality results. Thus, the proposed unsupervised video segmentation method is able to extract more suitable superpixels for learning the GMM appearance model to improve the segmentation accuracy. The experimental results demonstrate that our unsupervised video segmentation method, which benefits from the refined foreground prior, performs favorably against the existing methods.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012. 2

[2] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving patchmatch for large displacement optical flow. *IEEE TIP*, 23(12):4996–5006, 2014. 2

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 1, 2, 3, 4

[4] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE TPAMI*, 2011. 1

[5] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *CVPR*, 2011. 1

[6] J. Carreira and C. Sminchisescu. CPMC: automatic object segmentation using constrained parametric min-cuts. *IEEE TPAMI*, 2012. 2

[7] J. Chang, D. Wei, and J. W. F. III. A video representation using temporal superpixels. In *CVPR*, pages 2051–2058, 2013. 5

[8] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 2

[9] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 4

[10] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM TOG*, 34(6):195:1–195:10, 2015. 2, 5

[11] J. D. Foley, A. van Dam, S. Feiner, and J. F. Hughes. *Computer graphics - principles and practice, 2nd Edition*. Addison-Wesley, 1990. 2

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

[13] H. Fu, D. Xu, and S. Lin. Object-based multiple foreground segmentation in RGBD video. *IEEE Trans. Image Processing*, 26(3):1418–1427, 2017. 1, 2

[14] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE TPAMI*. 1

[15] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 1, 2, 5

[16] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE TPAMI*, 2013. 4

[17] A. Jain, S. Chatterjee, and R. Vidal. Coarse-to-fine semantic video segmentation using supervoxel trees. In *ICCV*, 2013. 1

[18] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 1, 2, 4

[19] Y. Lu, X. Bai, L. G. Shapiro, and J. Wang. Coherent parametric contours for interactive video object segmentation. In *CVPR*, pages 642–650, 2016. 2

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

[21] S. Mahmoudpour and M. Kim. Superpixel-based depth map estimation using defocus blur. In *ICIP*, pages 2613–2617, 2016. 3

[22] N. Marki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, pages 743–751, 2016. 1, 2, 5

[23] J. V. Neumann. The general and logical theory of automata. In *Cerebral Mechanisms in Behaviour*, pages 1–2. 1951. 3

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

[25] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 1, 2, 4

[26] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. J. V. Gool, M. H. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 4, 5

[27] F. Perazzi, O. Wang, M. H. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, pages 3227–3234, 2015. 5

[28] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, pages 3282–3289, 2012. 1

[29] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009. 1, 2

[30] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In *CVPR*, pages 110–119, 2015. 3

[31] S. A. Ramakanth and R. V. Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, pages 376–383, 2014. 1, 2, 5

[32] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM TOG*, 2004. 4

[33] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 2

[34] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, pages 2233–2240, 2011. 2

[35] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015. 4

[36] Y. Tsai, M. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 1, 2

[37] T. Wang and J. P. Collomosse. Probabilistic motion diffusion of labeling priors for coherent video segmentation. *IEEE TMM*, 2012. 1, 2

[38] T. Wang, B. Han, and J. P. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *CVIU*, 120:14–30, 2014. 2

[39] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 4

[40] L. Wen, D. Du, Z. Lei, S. Z. Li, and M. Yang. JOTS: joint online tracking and segmentation. In *CVPR*, 2015. 1, 2

[41] F. Yang, H. Lu, and M. Yang. Robust superpixel tracking. *IEEE TIP*, 2014. 1, 2

[42] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 1, 2