Video Supervoxels Using Partially Absorbing Random Walks

Yuling Liang, Jianbing Shen, *Senior Member, IEEE*, Xingping Dong, Hanqiu Sun, *Member, IEEE*, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Supervoxels have been widely used as a preprocessing step to exploit object boundaries to improve the performance of video processing tasks. However, most of the traditional supervoxel algorithms do not perform well in regions with complex textures or weak boundaries. These methods may generate supervoxels with overlapping boundaries. In this paper, we present the novel video supervoxel generation algorithm using partially absorbing random walks to get more accurate supervoxels in these regions. Our spatial–temporal framework is introduced by making full use of the appearance and motion cues, which effectively exploits the temporal consistency in video sequence. Moreover, we build a novel Laplacian optimization structure using two adjacent frames to make our approach more efficient. Experimental results demonstrated that our method achieved better performance than the state-of-the-art supervoxel algorithms.**

***Index Terms*— Laplacian graph, optimization, partially absorbing random walk (PARW), supervoxel, video segmentation.**

I. Introduction

A SUPERVOXEL is commonly defined as a subset of 3-D grids of video volume, which contains voxels that are similar in color, texture [13], etc. It is regarded as a perceptually meaningful atomic volume in a spatial–temporal domain. Similar to superpixels [33], supervoxels carry richer information such as spatial–temporal consistency, and they are often more manageable than simple voxels. Thus, supervoxels have been generally used for many video analysis and processing applications [19], [35], [36], such as video tracking, object recognition [32], motion analysis [31], and scene reconstruction [28], [34], [37]. A good supervoxel should own a regular shape with a uniform size and adhere well to the boundaries of object, which will improve the performance and decrease the computing cost of video processing applications.

There are many methods to offer supervoxels, such as segmentation by weighted aggregation (SWA) [1], Nystrom normalized cuts (NNCs) [3], the graph-based (GB) method [4], meanshift (MS) [8], graph-cut segmentation [15], simple linear iterative clustering (SLIC) [21], multiple hypothesis from superpixel flows [20], streaming hierarchical video segmentation [26], and online VideoSEEDS [29]. In general, supervoxels by these methods adhere well to the boundary of object in many cases. However, in the complicated regions with weak boundary or complex texture, most of them perform an unsatisfying contour support in the video [1], [2], [4], [15], [21], [38]. Some fail to preserve uniform size of the supervoxels, such as the graphcut-based supervoxel (GCSV) algorithms [15], [29].

To overcome the above limitations, we first want to find a good superpixel method in an image, similar to most methods extended from superpixels. The random walk (RW) segmentation method is a good solution for the weak boundary problem, which has been studied widely in the society of video processing and computer vision. Essentially, the RW is a stochastic process and computes the probability from each pixel to a set of seed points in the image. Then the pixel with highest probability reaching a seed is assigned with the same label of this seed. Thus, the RW can be used for superpixels segmentation through specifying the different seeds in an image. Sometimes the RW can be misleading [5], [9], [25], [27], [38], which may make the boundaries of superpixels not adhere to the object well. Recently, Wu *et al.* [25] proposed the partially absorbing RWs (PARWs), which can keep the good property of RW and decrease the error of misleading. Then the excellent properties of PARW algorithm [10], [11], [17], [25] can be utilized to design the supervoxels segmentation for videos. In fact, the solution of most RW algorithms can be transformed to solve the Laplacian linear equations or the inverse of the Laplacian matrix. Thus, our supervoxel algorithm is designed by the Laplacian matrix of PARW between adjacent frames.

In this paper, we propose a spatial–temporal framework using the PARW algorithm to generate the video supervoxel segmentations. Our method is able to produce high-quality compact and nearly uniform supervoxels within this framework. Fig. 1 shows a visual comparison between our algorithm

Manuscript received August 1, 2014; revised November 11, 2014, December 26, 2014, February 2, 2015, and February 8, 2015; accepted February 13, 2015. Date of publication February 24, 2015; date of current version May 3, 2016. This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2013CB328805, in part by the National Natural Science Foundation of China under Grant 61272359 and Grant 61125106, in part by the Research Grants Council under Grant 416311, in part by the Program for New Century Excellent Talents in University under Grant NCET-11-0789, in part by the Fok Ying-Tong Education Foundation for Young Teachers, in part by the Key Research Program through the Chinese Academy of Sciences, Xi'an, China, under Grant KGZD-EW-T03, and in part by the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. This paper was recommended by Associate Editor J. M. Martinez.

Y. Liang, J. Shen, and X. Dong are with Beijing Key Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenjianbing@bit.edu.cn).

H. Sun is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

X. Li is with State Key Laboratory of Transient Optics and Photonics, Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710075, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2406232

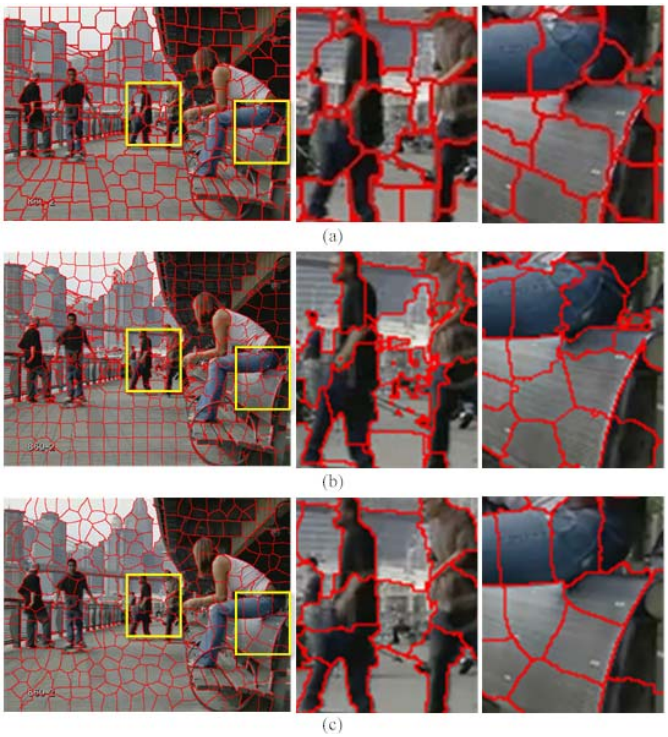


Fig. 1. Supervoxels generated from three methods. (a) GCSV [15]. (b) SLIC algorithm [21]. (c) Our method. Left column: the supervoxel results by these three algorithms. Middle and right columns: the zoomed-in results for the two yellow rectangles in the left column.

and the other two well-known methods: GCSV [15] and SLIC [21]. It is obvious that our supervoxel method achieves better segmentation results of the boundary region with a more regular shape and a more uniform size. First, we build a novel Laplacian optimization structure using two consecutive frames, which is formulated and incorporated into our PARW algorithm. Based on this Laplacian optimization matrix, our supervoxel segmentation can be performed on two adjacent frames, instead of processing the whole video frames such as [1], [14], [18], and [30]. Our method only needs to initialize the seeds in the first frame, and then these seeds are automatically propagated to the next frames using optical flow, which will maintain the temporal consistency very well. To the best of the authors' knowledge, we are the first to successfully formulate the supervoxel generation problem as an PARW algorithm by utilizing the motion information. Our algorithm not only maintains the spatial–temporal consistency in the regions with weak boundary or complex texture, but also ensures the temporal consistency by adding the motion information in video. Our source code will be publicly available online.[1]

II. Our Algorithm

We propose a supervoxel segmentation method for videos using the PARW algorithm and spatial–temporal information. The PARW algorithm is applied to each video frame twice, where the initial segmentation results are first obtained by running the PARW algorithm and then these initial results are further optimized. A novel Laplacian optimization structure is constructed by considering the motion consistency between frames, which will make the supervoxel adhere well to the boundary as well as maintain the temporal continuity. We define S as the set of seeds, and $S(f_t)$ denotes the set of initial K seeds of each frame f_t. Our method begins by placing the initial seeds $S(f_1)$ just on the first frame f_1 of the input video sequence. The initial seeds of the following frames are spread from previous frames. We use the method in [12] and [33] to obtain the initial seeds $S(f_1)$ of the first frame f_1, where these seeds are evenly placed on a rectangular lattice. The number K of initial seeds is prefixed by user, where a larger K leads to smaller superpixels at each frame. Thus, the user can adjust the size of superpixels by changing the number K. Fig. 2 shows the main steps of the proposed supervoxel algorithm, and the adjacent two frames (f_t and f_{t+1}) are considered and processed together to enhance the spatial–temporal consistency.

As shown in Fig. 2, the Laplacian matrix $L(f_t)$ is first built and the initial supervoxel results ($\mathrm{SP}(f_t)$) are obtained by performing the $\mathrm{PARW}(f_t)$ segmentation algorithm. Then an adjustment step ($\mathrm{Adjust}(f_t)$) is made on the initial segmentation result to generate new positions of seeds ($S^{\mathrm{new}}(f_t)$). The motion information of the next frame f_{t+1} is fully utilized to guide the segmentation process of the current frame f_t. This is done by building a Laplacian matrix ($L(f_t, f_{t+1})$) in Fig. 3. The final supervoxel results ($\mathrm{SP}^{\mathrm{new}}(f_t)$) of f_t are generated by running the PARW segmentation algorithm twice using a different Laplacian matrix. The construction process of our Laplacian matrix considers both the color and motion cues of two adjacent frames (f_t, f_{t+1}), which ensures the spatial–temporal consistency of the supervoxel. According to the optical flow information ($\mathrm{OPT}(f_t, f_{t+1})$), the initial seeds ($S(f_t)$) of each frame are propagated from f_t to f_{t+1} naturally. With these new seeds, the $\mathrm{PARW}(f_t, f_{t+1})$ segmentation algorithm is performed again. The purpose of this adjustment is to make the supervoxel more homogeneous and uniform with color and texture information, and we will discuss this in detail in Section II-C. By repeating the above segmentation-propagation-segmentation procedure, our approach finally generates consistent supervoxel results.

A. Laplacian Graph Structure

Here, we construct two kinds of graphs. The first graph structure is defined on single frame and the second one is based on two adjacent frames. First, an undirected graph $G^{(t)} = (V^{(t)}, E^{(t)}, W^{(t)})$ is constructed on a given frame f_t. In this graph, each node $v_i^{(t)} \in V^{(t)}$ uniquely identifies an image pixel $x_i^{(t)}$, and $e(v_i^{(t)}, v_j^{(t)}) \in E^{(t)}$ denotes the edge connecting to two neighboring nodes. $w^{(t)}(v_i, v_j) \in W^{(t)}$ measures the similarity of two nodes by the weighting function. Then the Laplacian matrix $L(f_t)$ of the graph is defined as

$$L(f_t) = D^{(t)} - W^{(t)} \tag{1}$$

where $D^{(t)}$ is a diagonal matrix and $D = \mathrm{diag}(d_1, d_2, \ldots, d_N)$, $d_i^{(t)} = \sum_j w_{ij}^{(t)}$ is the degree that all edges $e_{ij}^{(t)}$ incident on $v_i^{(t)}$. N is the number of pixels in one frame.

[1] http://github.com/shenjianbing/parwsupervoxel

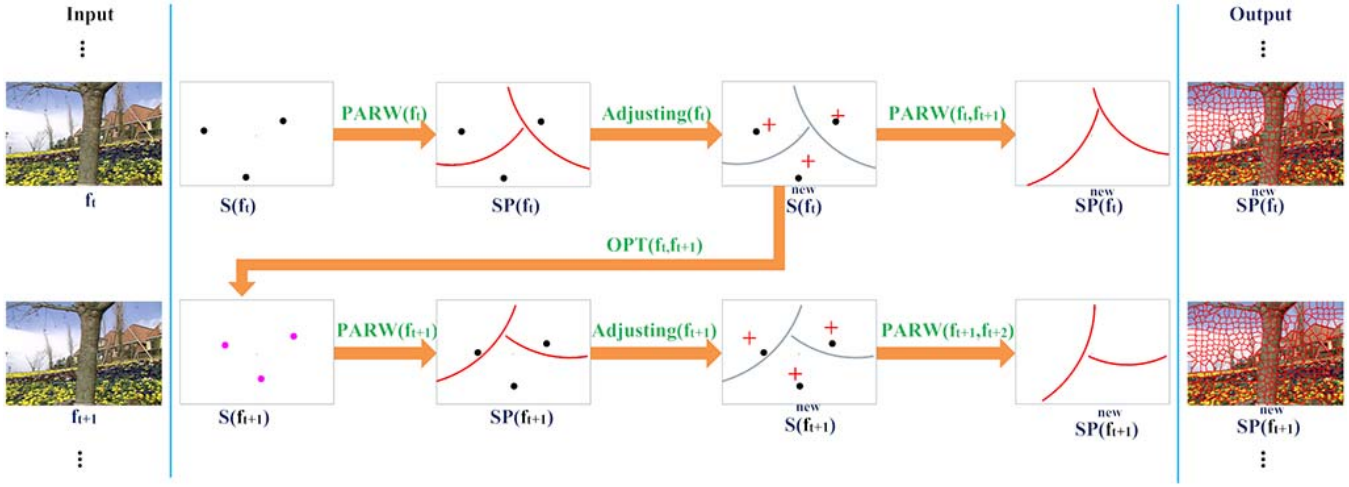


Fig. 2. Framework of our spatial–temporal supervoxel method. $S(f_t)$ denotes the initial seeds of frame f_t. SP(f_t) denotes the initial superpixel segmentation results of f_t using the PARW algorithm (PARW(f_t)). $S^{\text{new}}(f_t)$ denotes the new adjusted seeds by Adjusting(f_t). SP$^{\text{new}}(f_t)$ is the final superpixel results by performing the PARW algorithm (PARW(f_t, f_{t+1})) again. OPT(f_t, f_{t+1}) denotes seed propagation by the optical flow of f_t, we then obtain the initial seeds $(S(f_{t+1}))$ of frame f_{t+1}. The first row (second row) denotes the segmentation process of frame f_t (f_{t+1}).

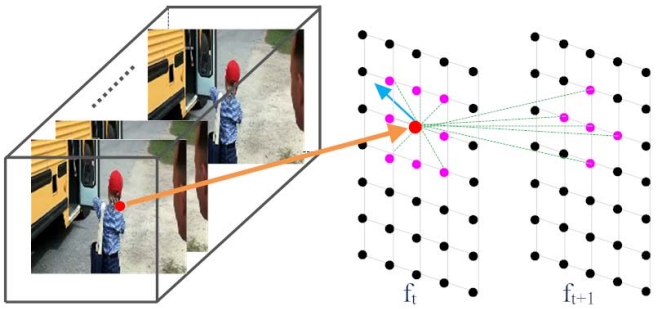


Fig. 3. Proposed Laplacian graph structure $L(f_t, f_{t+1})$ of two neighboring frames in a video. Red node: the current pixel. Pink nodes: their neighboring pixels. Black nodes: the other pixels.

Furthermore, we construct a new Laplacian graph $G'^{(t)} = (V'^{(t)}, E'^{(t)}, W'^{(t)})$ on two consecutive frames of the video, where $V'^{(t)}$ has $2N$ nodes and $W'^{(t)}$ denotes the edge weight matrix that measures the spatial–temporal similarity of adjacent nodes. As shown in Fig. 3, our Laplacian graph structure $L(f_t, f_{t+1})$ considers both the spatial and temporal connections. Each node $v_i^{(t)}$ in the current frame f_t utilizes the spatial connections (pink nodes in f_t) and temporal connections (pink nodes in f_{t+1}) to construct the new Laplacian matrix, where the optical flow information (blue arrow in Fig. 3) of the current node is also incorporated into $L(f_t, f_{t+1})$.

Our Laplacian matrix utilizes both the color information of neighboring pixels and the motion information of optical flow in each frame. The weight w_{ij}^a measures the color similarity between neighboring pixels, and the motion weight w_{ij}^m measures the difference of optical flow vectors between x_i and x_j. These edge weights are calculated by the following weighting functions:

$$w_{ij}^a = \exp\left(-\frac{\|I_i - I_j\|^2}{\sigma^2}\right) + \epsilon$$
$$w_{ij}^m = 1 - \frac{\|M_i - M_j\|}{\max\{\|M_i\|, \|M_j\|\}} \tag{2}$$

where I_i and I_j denote the intensity values at two pixels x_i and x_j, respectively, and σ is a normalizing constant. ϵ is a user-defined constant with 10^{-6}. M_i and M_j denote the optical flow in x_i and x_j, respectively, where $M_i = [v(x)_i, v(y)_i]^T$ is a 2-D direction vector. In our approach, we adopt the Lucas–Kanade optical flow method [6] to obtain M_i and M_j for computation simplicity. As mentioned above, the texture homogeneity contains two parts: color homogeneity and motion homogeneity. Only when two neighbors own similar color and similar motion, they would be prone to be assigned to the same superpixel. Our method can find the weight value to keep the superpixel as homogenous as possible. To maintain the texture homogeneity, we define the spatial connection weight as

$$w_{ij}^{\text{spa}} = \min\left(w_{ij}^a, w_{ij}^m\right). \tag{3}$$

Using (3), the spatial affinity matrixes of frames f_t and f_{t+1} are denoted by $W^{\text{spa}(t)}$ and $W^{\text{spa}(t+1)}$, respectively. The temporal connection measures the similarity of two pixels between two consecutive frames (f_t and f_{t+1}). The weight b_{ij}^{tmp} of one temporal edge denotes the color similarity of two pixels x_i and x_j, where $x_i \in f_t$ and $x_j \in f_{t+1}$. As shown in Fig. 3, each pixel x_i in frame f_t is connected to five pixels in the next frame f_{t+1}. The temporal affinity matrix B^{tmp} is defined as

$$b_{ij}^{\text{tmp}} = \exp\left(-\frac{\|I_i - I_j\|^2}{\sigma^2}\right) + \epsilon$$
$$\text{s.t. } i \in f_t, \quad j \in f_{t+1}, \quad i \sim j, \quad b_{ij}^{\text{tmp}} \in B^{\text{tmp}} \tag{4}$$

where $i \sim j$ indicates that pixel x_i is connected to x_j. Note that the motion information of optical flow is not included in the temporal affinity matrix B^{tmp}. This is because the spatial affinity matrix $W^{\text{spa}(t)}$ has included this motion information and the final weight matrix includes these two affinity matrixes. For computational simplicity, we just use the color information for temporal affinity matrix $B^{\text{tmp}(t)}$.

According to the aforementioned spatial–temporal edge weights and their affinity matrixes, we define the new weight

matrix W' as

$$W'^{(t)} = \begin{bmatrix} W^{\mathrm{spa}(t)} & \mu B^{\mathrm{tmp}(t)^T} \\ \mu B^{\mathrm{tmp}(t)} & W^{\mathrm{spa}(t+1)} \end{bmatrix} \tag{5}$$

where $W^{\mathrm{spa}(t)}$ and $W^{\mathrm{spa}(t+1)}$ measure the local spatial similarities, and B^{tmp} denotes the temporal similarity in the two adjacent frames. μ is a weighting parameter between these measurements.

Thus, the new Laplacian graph structure $L(f_t, f_{t+1})$ is defined as

$$L(f_t, f_{t+1}) = D'^{(t)} - W'^{(t)}. \tag{6}$$

B. PARW Supervoxel Segmentation

After building the Laplacian matrix $L(f_t)$ on each frame and the corresponding Laplacian optimization structure $L(f_t, f_{t+1})$ on two adjacent frames, we perform PARW(f_t) and PARW(f_t, f_{t+1}) supervoxel segmentation on them, respectively.

1) PARW Algorithm: Our segmentation approach is based on PARW [25], which is originally used in data clustering and analysis [9], [10], [25]. PARW suggests a stochastic process that a RW would be absorbed at current state i with probability α and with probability $1-\alpha$ out to other states. When a random worker starts from a homogeneous region with proper absorption rates α, the PARW algorithm has one good property that the absorption probabilities vary slowly inside this region and drop greatly outside this region. In other words, this RW is mostly absorbed in this region at last. According to this property, PARW can be used for superpixel or supervoxel segmentation, since a superpixel or supervoxel is a homogeneous region with similar color and texture inside this region and drops sharply across the boundaries. In fact, our PARW method has a proper probability measurement on the graph by considering the global information, which avoids the misleading problem of RW and solves the weak boundary problem well for supervoxel segmentation.

Our supervoxel segmentation performs as a labeling problem. First, we just consider our method on one frame. Assume that each seed in set S of the frame has been assigned a distinct label h, $h \in H = \{h_1, h_2, \ldots, h_k\}$, where k is the number of labels. We define a $N \times 1$ indicating vector s^h for each seed, N is number of pixel in the graph

$$s_i^h = \begin{cases} 1 & \text{if } x_i \text{ is assigned to label } h \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

denote p_i^h as the likelihood of pixel x_i to be labeled with h. p^h is an $N \times 1$ vector that indicates the probabilities of all the nodes in the graph to be assigned to the label h. Our supervoxel segmentation is achieved by assigning each pixel x_i to the label h having the highest probability. p^h is defined as

$$\begin{aligned} p^h &= As^h \\ A &= (\Lambda + L)^{-1}\Lambda = (\alpha I + L)^{-1}(\alpha I) \end{aligned} \tag{8}$$

where A is the absorption probability matrix, which is defined in [25]. $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$.

After we obtain the probabilities, it is straightforward to assign x_i to the label h having the largest probability

$$\begin{aligned} R_i &= \arg\max_h p_i^h \\ &= \arg\max_h As^h \\ &= \arg\max_h (\alpha(\alpha I + L)^{-1} s^h) \end{aligned} \tag{9}$$

where R_i denotes the final label of pixel x_i. The supervoxel segmentation is obtained by assigning the label R_i to each pixel x_i.

2) PARW Supervoxel Segmentation: Our supervoxel segmentation framework is illustrated in Fig. 2. In this framework, we introduce a two-step PARW segmentation for each frame, one for initial segmentation and another for supervoxel optimization. That is, frame f_t is segmented by

$$\begin{cases} SP(f_t) = PARW(f_t) & \text{using } L(f_t) \\ \mathrm{SP}^{\mathrm{new}}(f_t) = PARW(f_t, f_{t+1}) & \text{using } L(f_t, f_{t+1}) \end{cases} \tag{10}$$

where SP(f_t) is the initial segmentation results of frame f_t by performing PARW(f_t) and $\mathrm{SP}^{\mathrm{new}}(f_t)$ is the final segmentation results by PARW(f_t, f_{t+1}).

From (8), (9), and (10), we get the initial segmentation label of each pixel x_i in frame f_t

$$\begin{aligned} SP(f_t)_i &= \mathrm{PARW}(f_t)_i = \arg\max_{h_l} p^{h_l}(f_t)_i \\ &= \arg\max_{h_l} (\alpha(\alpha I + L(f_t))^{-1} s^{h_l}(f_t))_i \end{aligned} \tag{11}$$

where $s^{h_l}(f_t)$ is the indicating vector of seed set $S(f_t)$.

The final supervoxel result of frame f_t is defined as

$$\begin{aligned} \mathrm{SP}^{\mathrm{new}}(f_t)_i &= \mathrm{PARW}(f_t, f_{t+1})_i = \arg\max_{h_l} p^{h_l}(f_t, f_{t+1})_i \\ &= \arg\max_{h_l} (\alpha(\alpha I + L(f_t, f_{t+1}))^{-1} s^{h_l}(f_t, f_{t+1}))_i \end{aligned} \tag{12}$$

where $s^{h_l}(f_t, f_{t+1})$ is the indication vector of seed set $S(f_t, f_{t+1})$.

C. Spatio-Temporal Supervoxel

1) Supervoxel: To further ensure the homogeneous properties for each supervoxel, we make an optimization to the initial segmentation result SP(f_t) before performing PARW(f_t, f_{t+1}) segmentation again. This strategy makes an adjustment operation Adjusting(f_t) on the initial result SP(f_t) and generates new seeds $S^{\mathrm{new}}(f_t)$. In this adjustment step, our approach finds the suitable relocation center positions (seeds) $S^{\mathrm{new}}(f_t)$ of superpixels, which makes the boundaries of superpixels more consistent with the object edges

$$\begin{aligned} S_{h_l}^{\mathrm{new}}(f_t) &= \frac{\sum_{\mathrm{SP}(f_t)_i = h_l} \omega_i^t y_i^t}{\sum_{\mathrm{SP}(f_t)_i = h_l} \omega_i^t} \\ \omega_i^t &= \frac{\max_{h_l} p_i^{h_l}}{||y_i^t - S(f_t)_{\arg\max_{h_l} p_i^{h_l}}||} \end{aligned} \tag{13}$$

where $S_{h_l}^{\mathrm{new}}(f_t)$ are the new seeds with label h_l of the current processing frame f_t. $S(f_t)$ is the set of original seeds. $p_i^{h_l}$ is the likelihood of a node x_i to be labeled with h_l in (8). y_i is the coordinate vector of a node x_i in frame f_t.

Algorithm 1 Video Supervoxels Using PARW

Input: An input video $F = \{f_1, f_2, \cdots, f_n\}$;
Output: Supervoxel results $SV = \{SP^{\text{new}}(f_1), SP^{\text{new}}(f_2), \cdots, SP^{\text{new}}(f_n)\}$;
1: Calculate the initial seeds $S(f_1)$ for the first frame f_1;
2: Compute the optical flow for each frame;
3: **for** t = 1 : n **do**
4: Build the Laplacian matrix $L(f_t)$ by (1) and the Laplacian optimization matrix $L(f_t, f_{t+1})$ by (6);
5: Perform the $PARW(f_t)$ algorithm by (11) and obtain the initial supervoxel results $SP(f_t)$;
6: Generate new seeds $S^{\text{new}}(f_t)$ by (13);
7: Execute the PARW(f_t, f_{t+1}) algorithm again by (12), and obtain the final supervoxel results $SP^{\text{new}}(f_t)$ of frame f_t;
8: Generate initial seeds $S(f_{t+1})$ of frame f_{t+1} by optical flow OPT(f_t, f_{t+1}) and (13);
9: **end for**
10: Run steps 1 to 8 until all the frames are performed;
11: Obtain the final supervoxel results $\{\text{SP}^{\text{new}}(f_1), \text{SP}^{\text{new}}(f_2), \cdots, \text{SP}^{\text{new}}(f_n)\}$.

2) Seeds Propagation: In general, the shape and appearance features of objects in each video scene vary slowly from frame to frame. We assume that the seeds of each frame are spread by its previous frame, jointly with optical flow information to strengthen the consistency of supervoxels between the next two frames.

As shown in Fig. 2, the initial seeds $S(f_{t+1})$ of the next frame f_{t+1} are obtained by propagating the seeds $S^{\text{new}}(f_t)$ from the current frame f_t using optical flow OPT(f_t, f_{t+1}). We use the Lucas–Kanade optical flow algorithm to compute the value of reliability flow for each pixel for computational simplicity. Thus, we can make a judgement about whether each seed in $S^{\text{new}}(f_t)$ is on the reliable flow. If not, we just directly project those seeds to the next frame f_{t+1}. If there are some seeds on the reliable flow, we make a modification to the position of those seeds according to its optical flow after the projection procedure. It is defined as

$$S(f_{t+1})_{i_x} = \begin{cases} S^{\text{new}}\bar{(}f_t)_{i_x} & \text{seeds not on reliable flow} \\ S^{\text{new}}\hat{(}f_t)_{i_x} + v(x)_i & \text{otherwise} \end{cases}$$
$$S(f_{t+1})_{i_y} = \begin{cases} S^{\text{new}}\bar{(}f_t)_{i_y} & \text{seeds not on reliable flow} \\ S^{\text{new}}\hat{(}f_t)_{i_y} + v(y)_i & \text{otherwise} \end{cases} \tag{14}$$

where $S^{\text{new}}\hat{(}f_t)_{i_x}$ and $S^{\text{new}}\hat{(}f_t)_{i_y}$ denote the seeds on the reliable flow. $S^{\text{new}}\bar{(}f_t)_{t_i}$ and $S^{\text{new}}\bar{(}f_t)_{i_x}$ denote the seeds that are not on the reliable flow. $v(x)_i$ and $v(y)_i$ are components of optical flow M_i of each pixel i i.e., $M_i = [v(x)_i, v(y)_i]^T$.

At last, we summarize our supervoxel algorithm with the pseudocodes in Algorithm 1.

III. Experiments and Evaluation

In this section, we evaluate the performance of the proposed supervoxel segmentation approach and compare it with the state-of-the-art supervoxel segmentation methods. The main parameters are first introduced in our algorithm and then the influence of these parameters with different values is also discussed. Thus, the optimal values for these parameters are suggested through our experiments. To get an objective and intuitive comparison, we quantitatively evaluate our algorithm with other eight well-known supervoxel algorithms: GB [4], NNCs [3], SWA [1], MS [8], Graph-Based Hierarchical (GBH) [14], GCSV [15], SLIC [21], and VideoSEEDS [29]. Moreover, a qualitative comparison of the supervoxels results is performed on more challenging examples with the above seven well-known methods.

A. Parameter Settings

There are two main parameters in our algorithm: 1) α and 2) μ. The parameter α controls the performance of the PARW algorithm. When α is close to zero, the PARW algorithm will converge to the constant distribution regardless of the starting state. We find that $\alpha = 10^{-4}$ produces a sufficiently good performance of the supervoxel results in the boundary adherence. The parameter μ measures the ratio of the spatial weights to temporal weights. We first analyze the weighting parameter μ. As shown in Fig. 4, our method achieves the similar supervoxel results when we vary the values of μ from μ^{-3} to μ^{0}. However, there are small differences for the boundary adherence in some complex texture regions [see the pink rectangles in Fig. 4(e)–(h)]. We empirically set μ to 10^{-2} so as to produce high-quality supervoxel results in our experiments. Another parameter is the initial number of supervoxels K, which is predefined according to the size of videos. Thus, this number is also given in the descriptions of most qualitative results.

The parameter α in (8) controls the absorption rates of PARW. With a suitable absorption rate, a RW starting from a homogeneous region stops in this region with a high probability. Fig. 5 gives an illustration of different values of parameter α and their corresponding supervoxel segmentation results. Our algorithm achieves the satisfying supervoxel results when we set α from 10^{-6} to 10^{0} [Fig. 5(e)–(h)], and the supervoxels generated by our method are stable. Moreover, our approach still preserves regular supervoxels even in the weak boundaries. We have found that our method can generate sufficient good supervoxels with accurate boundary adherence when we empirically set $\alpha = 10^{-4}$ through our experiments.

The boundaries of supervoxels by our approach adhere well to the object boundaries in the video, and the size of supervoxels is regular and uniform. Fig. 6 demonstrates the qualitative effects for the supervoxels obtained by our algorithm on three representative video clips, and we show ten frames for each video. We validate the approach on the problem of video occlusion in the first video clip and it produces good supervoxel results. In the second video clip, our supervoxel method successfully detects the correct boundaries, even when the boundaries around the hair of person and the background textures are very weak. The new incoming object can be well processed. The third video clip illustrates the fast motion segmentation.

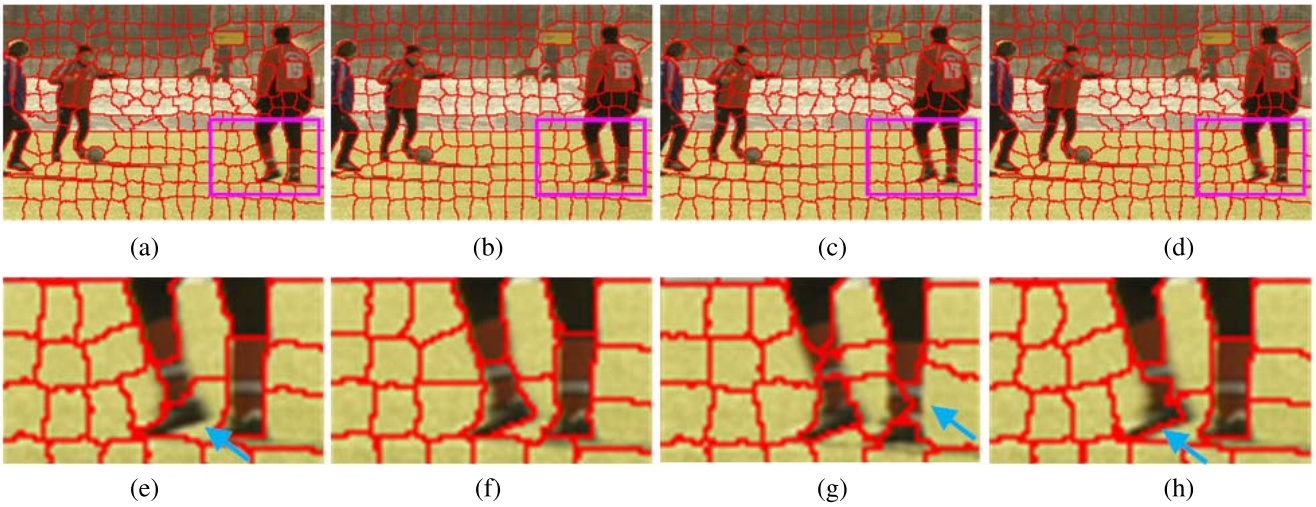


Fig. 4. Supervoxel results with different values of parameter μ (a) $\mu = 10^{-3}$. (b) $\mu = 10^{-2}$. (c) $\mu = 10^{-1}$. (d) $\mu = 10^{0}$. (e)–(h) Close-ups of (a)–(d) in pink rectangles.

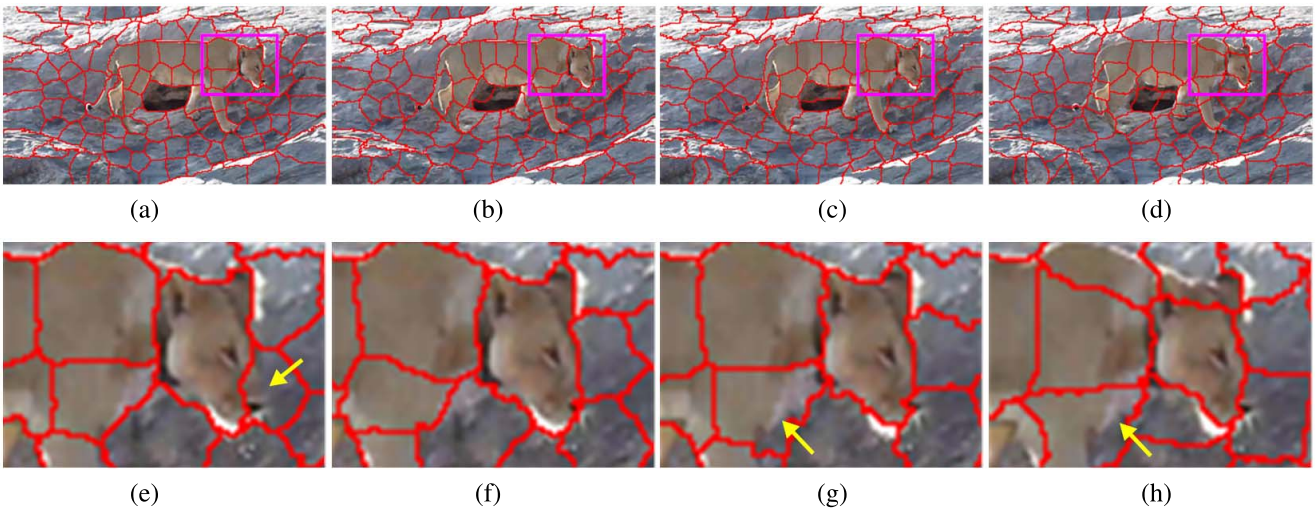


Fig. 5. Supervoxel results with different values of parameter α from 10^{-6} to 10^{0}. (a) $\alpha = 10^{-6}$. (b) $\alpha = 10^{-4}$. (c) $\alpha = 10^{-2}$. (d) $\alpha = 10^{0}$. (e)–(h) Close-ups of (a)–(d) in yellow rectangles.

B. Quantitative Comparison With Other Algorithms

In Fig. 7, a quantitative comparison is given with other eight well-known algorithms: GB [4], NNCs [3], SWA [1] and MS [8], GBH [14], GCSV [15], SLIC [21], and VideoSEEDS [29]. In our experiments, we use both the data set from [16] and the data set from [24], which totally include eight plus six video sequences. We varied the supervoxel numbers from 200 to 900 for each video sequence to get different supervoxel segmentation results. The evaluation metrics are based on [24], and we use the source code of this evaluation method provided on the website.[2] The video sets are evaluated on a suite of metrics to assess object boundaries and spatial–temporal consistency. The metrics include 2-D undersegmentation error, 3-D segmentation error, 2-D segmentation accuracy, and 3-D segmentation accuracy.

[2]http://www.cse.buffalo.edu/~jcorso/r/supervoxels/

1) Undersegmentation Error: A good supervoxel algorithm should ensure that one supervoxel overlaps only one object. This evaluation measurement checks the under-segmentation error that is calculated by the following [24]:

$$\mathrm{UE}(g_i) = \frac{\left[\sum_{s_j | s_j \bigcap g_i \neq \emptyset} \mathrm{Vol}(s_j)\right] - \mathrm{Vol}(g_i)}{\mathrm{Vol}(g_i)} \tag{15}$$

where g_i is the ground-truth segments $(g_1, g_2, \ldots, g_n)$, Vol is the segment volume, and s_i denotes a supervoxel segment. This metric defines the undersegmentation error of an input video sequence by averaging this quantity over all the ground-truth segmentations.

Fig. 7(a)–(d) gives the undersegmentation errors of the 2-D-superpixel and 3-D-supervoxel results, respectively. The plot of the undersegmentation error shows that GBH and our approach outperform other eight representative algorithms. In 2-D error rate, both VideoSEEDS, GCSV, GBH and our method have lower error, and our approach performs slightly

Fig. 6. Supervoxel segmentation using our PARW algorithm. Ten frames with their corresponding supervoxel results are displayed for each input video. Videos with occlusion problem (the first row), new incoming object problem (the third row), and fast motion (the fifth row). The numbers of supervoxels are 250, 240, and 200, respectively. Our supervoxel boundaries adhere well to object boundaries, and the regularity and homogeneity of supervoxels are also maintained very well.

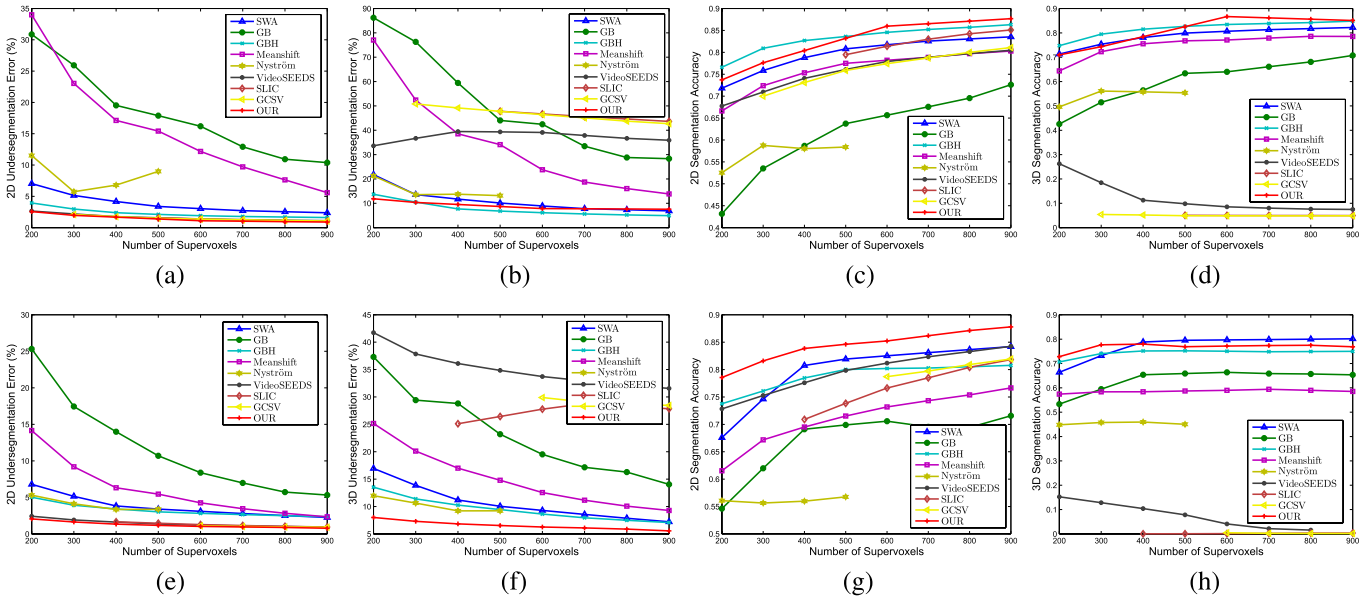


Fig. 7. Quantitative evaluation of the superovxel segmentation. We compare our algorithm with eight other well-known methods: GB [4], NNCs [3], SWA [1] and MS [8], GBH [14], GCSV [15], SLIC [21], and VideoSEEDS [29]. Top: result on the data set from [16]. Bottom: result on the SegTrack data set from [24]. Performance evaluation by the method in [24]. (a) and (e) 2-D under-segmentation error. (b) and (f) 3-D under-segmentation error. (c) and (g) 2-D segmentation accuracy. (d) and (h) 3-D segmentation accuracy.

better than the other three. The undersegmentation error rate of 2-D superpixel segmentation is less than 3%, and the 3-D error rate is 10% or less for our algorithm. Moreover, the error rate of our algorithm increases slightly when the number of supervoxels decreases. In contrast, both GB and MS increase sharply when the number of supervoxel increases.

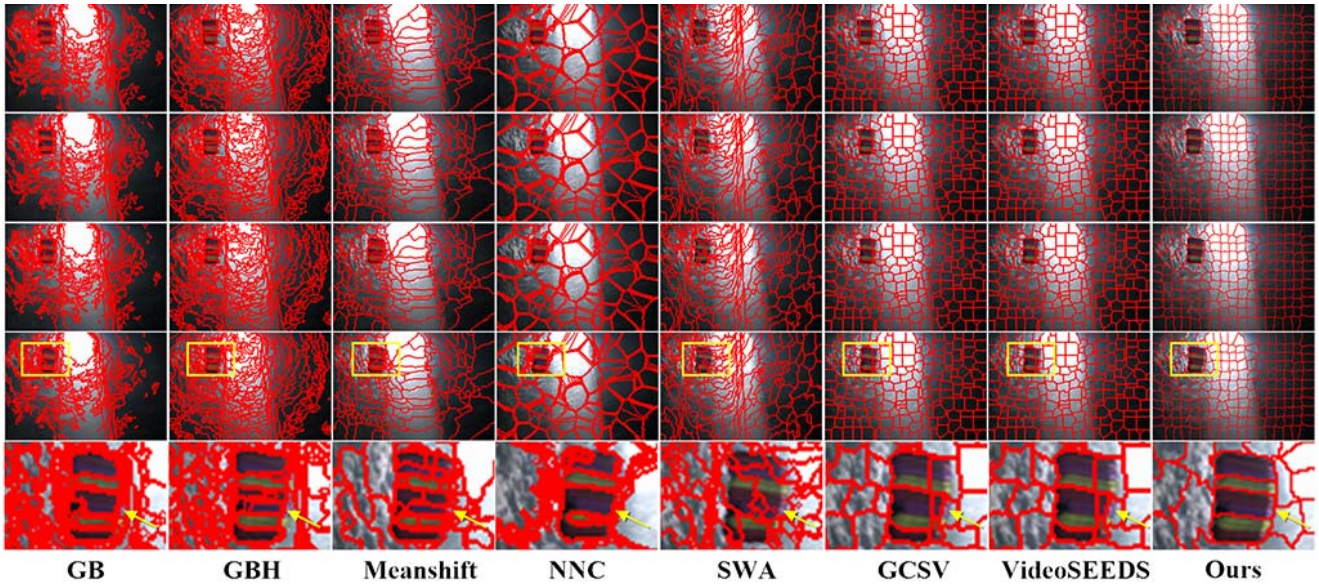


Fig. 8. Qualitative comparison with other seven well-known algorithms: GB [4], GBH [14], MS [8], NNCs [3], SWA [1], GCSV [15], and VideoSEEDS [29]. The last row is the close-up of the yellow rectangles in the fourth row. The initial number of our supervoxels is 200.

2) Segmentation Accuracy: The segmentation accuracy (ACCU) measures whether a fragment g_i of ground truth is correctly classified by the proposed algorithm, i.e., each supervoxel should cover only one object or a piece of fragment. For a fragment g_i of ground truth, if the main part of a supervoxel s_j covers g_i, then this supervoxel is as correct. Therefore, the 3-D segmentation accuracy $\mathrm{ACCU}(g_i)$ [24] is defined as

$$\mathrm{ACCU}(g_i) = \frac{\sum_{j=1}^{k} \mathrm{Vol}(s_j \bigcap g_i)}{\mathrm{Vol}(g_i)}. \tag{16}$$

Fig. 7(c), (d), (g), and (h) shows the segmentation accuracy of 2-D-superpixel and 3-D-supervoxel results, respectively. Apparently, GBH and our method perform significantly better than the other seven methods in terms of the 2-D and 3-D segmentation accuracy. Moreover, the 2-D segmentation accuracy reaches 85% when the number of supervoxels increases to about 600. Meanwhile, our method also achieves a 3-D segmentation accuracy of 81% when the number of supervoxels decreases to 200. It indicates that our algorithm has better performance than previous supervoxel algorithms.

C. Qualitative Comparison With Other Algorithms

We will provide a qualitative comparison with the state-of-the-art algorithms: GB [4], NNCs [3], SWA [1] and MS [8], GBH [14], GCSV [15], SLIC [21], and VideoSEEDS [29] in Fig. 8. Then we analyze the performance in terms of two measurements: boundary adherence and size uniformity. The comparison result shows that our approach outperforms the other seven competing methods as desired.

1) Boundary Adherence: The problem of boundary adherence is to find the correct edges of objects when they are parts of a consistent boundary. Fig. 8 shows that most of previous methods fail to segment out the week boundaries. As shown in Fig. 8, they are unable to seperate the correct edges in the region indicated by a yellow arrow. In contrast, our approach generates more consistent supervoxel results, which have smooth boundaries to adhere better to the object edges. This is because the PARW algorithm fully considers the global information which can greatly eliminate the misleading problem that exists in most RW methods (e.g., [7], [22], [23]).

2) Size Uniformity: The supervoxel results in the first five columns of Fig. 8 show that these five methods fail to generate regular and uniform supervoxels. Some supervoxels are narrow, some are small, and some are big. This may be due to the fact that these methods only consider the boundary adherence but do not consider the size uniformity. The remaining methods (GCSV and VideoSEEDS) generate more regular supervoxels than the aforementioned five methods; however, the size is still not very uniform. The reason in the case of GCSV may be that it uses the minimum cut criterion by imposing a ceiling bound of supervoxel size. For VideoSEEDS, this is because this method aims to provide a fast supervoxel method which may sacrifice some performance such as regularity with uniform size. In contrast, our approach enforces PARW segmentation to optimize the supervoxels iteratively, which generates more regular and uniform supervoxels in both shape and size.

To ensure the temporal continuity of supervoxels, we add the spatial–temporal information to our PARW optimization framework. As mentioned before, this information includes two important aspects. One is the motion consistency between two adjacent frames, and the other is the seeds propagation from the previous frame to the current frame. To demonstrate the effectiveness of this spatial–temporal information, we run our algorithm without and with this information on an input video, and the results of six consecutive frames are shown in Fig. 9. It appears that the algorithm without spatial–temporal information provides many nonuniform supervoxels [shown in Fig. 9(Top)], and loses the temporal continuity

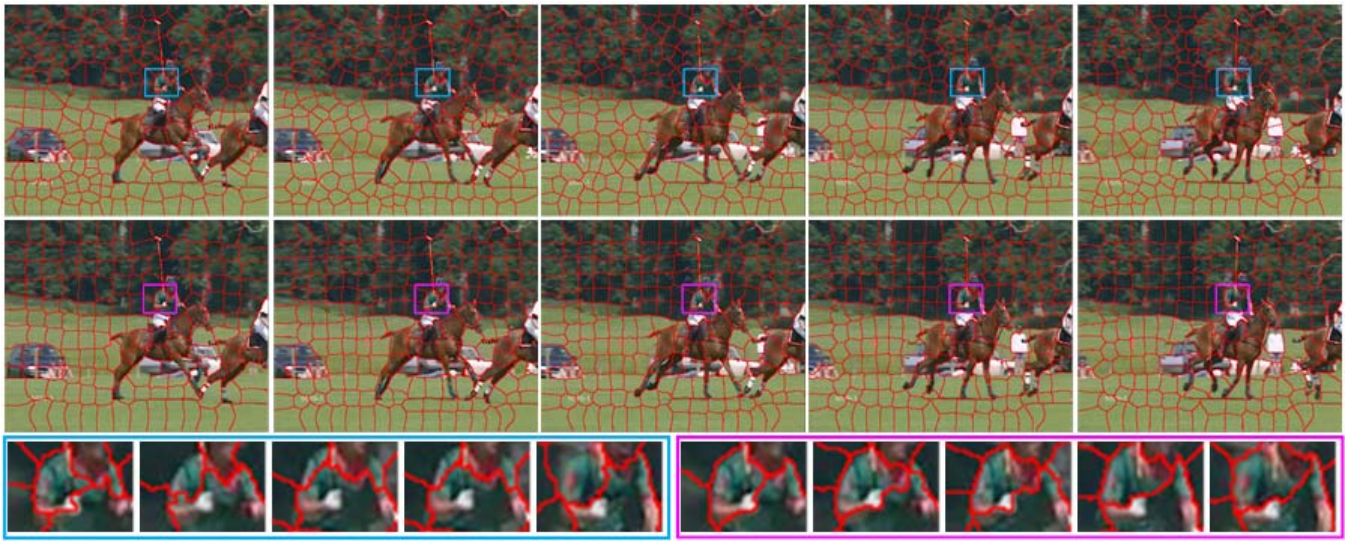


Fig. 9. Superpixel results with/without spatial–temporal information. Top: results by our full approach. Middle: results by our method without spatial–temporal information. Bottom: the blue and pink rectangles are the close-ups of blue rectangles and pink rectangles. The initial number of supervoxels is 200.

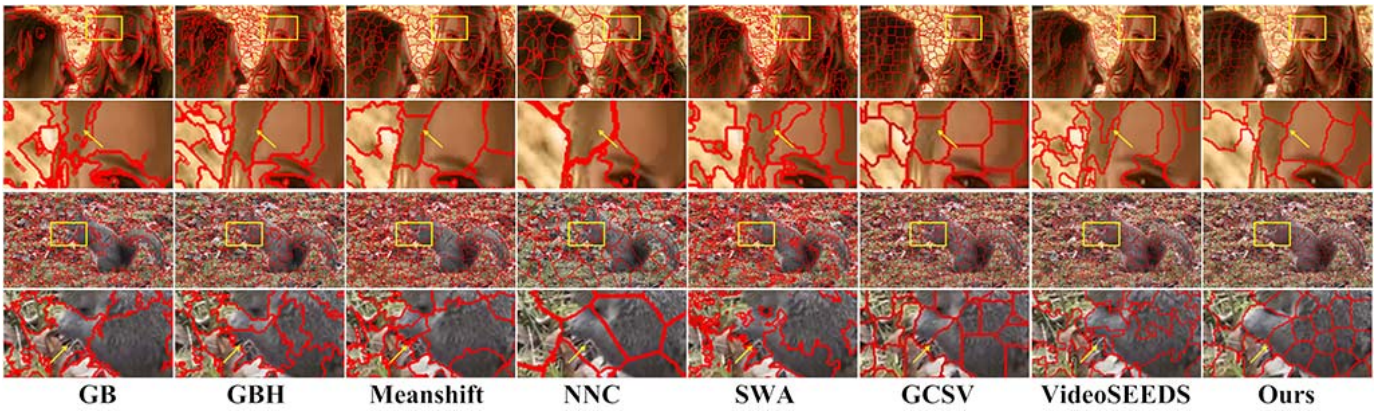


Fig. 10. Comparison results on videos with weak boundaries (top row) or complex texture (third row) between our method and other seven well-known methods: GB [4], GBH [14], MS [8], NNCs [3], SWA [1], GCSV [15], and VideoSEEDS [29]. The corresponding initial numbers of our supervoxels are 200 (top row) and 380 (third row). Both the second row and the fourth row are the zoomed-in results for the corresponding regions in the yellow rectangles.

[shown in the left of Fig. 9(Bottom)]. After adding this spatial–temporal information, our algorithm performs well for preserving the regularity and edge coherence in adjacent frames, as shown in Fig. 9(Middle) and (Bottom). This performance improvement may be because the strategy of seed propagation offers the appropriate seeds to maintain the uniformity of supervoxels, where the motion consistency ensures temporal continuity.

Furthermore, we evaluated our approach in some challenging videos with more weak boundaries or more complex texture compared with the seven well-known supervoxel methods: GB [4], GBH [14], MS [8], NNCs [3], SWA [1], GCSV [15], and VideoSEEDS [29]. Two sample video results are shown in Fig. 10. It is obvious that our approach performs the best in these videos. Our supervoxels keep the regular shape and uniform size, and adhere well to most boundaries of object. Both GBH and VideoSEEDS methods perform well, except that the sizes of their results are less uniform. In regions with complex texture, the MS approach may produce incorrect boundaries. For example, the boundaries between the mouse leg and the grass are complex boundaries [fourth row in Fig. 10], which cannot be segmented correctly. The GB and SWA methods may generate irregular boundaries both in regions with weak boundaries and in regions with a complex texture. The GCSV and NNC methods do not perfom well in these videos. For example, the boundaries between hair and face of the woman are very weak [second row in Fig. 10], so that these two methods cannot find them correctly.

Finally, we give one example to illustrate 2-D slices within the X-T and Y-T coordinates on two video clips in Fig. 11. In Fig. 11, each closed region in the X-T or Y-T plane represents the temporal slice of a supervoxel at this plane. And the longer length of this region at the T-axis represents a stronger temporal consistency. It is clear that the regular shape of one supervoxel is consistent in the time domain by slicing over the X-T and Y-T 2-D planes, which demonstrates the real supervoxel shape in the 3-D XYT space. The main

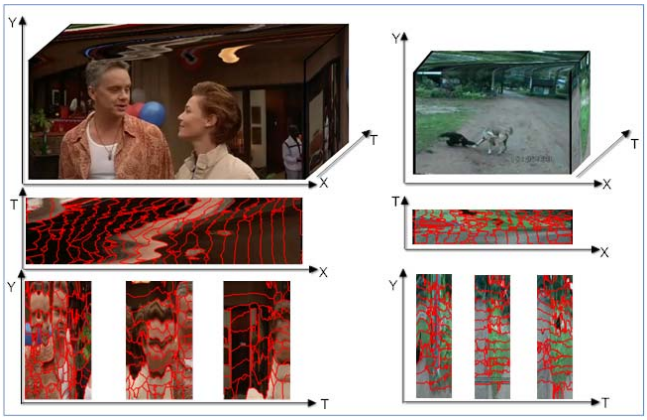


Fig. 11. 2-D slices with the X-T and Y-T coordinates on two video clips. The first row shows the original video clips. The second (third) row shows the consistent shape of supervoxel in X-T (Y-T) 2-D plane.

memory cost of our algorithm is the building of Laplacian graph matrix on two consecutive frames. Our approach only needs the constant memory to save the two consistent frames, where the memory cost is $O(2N)$ and N is the number of pixels in each frame. This makes our approach independent of the video clip length, and can segment the videos with arbitrarily long in theory. The prefixed number K of initial seeds drives the whole supervoxel process, and our PARW optimization method still obtains consistent spatiotemporal supervoxel segmentation with different values of K.

IV. Conclusion

In this paper, we have developed a novel spatial–temporal supervoxel generation algorithm for videos using PARWs. To the best of the authors' knowledge, our approach is the first work to perform supervoxel segmentation using the PARW algorithm. The proposed spatial–temporal framework is the key factor to obtain consistent yet accurate supervoxel results. A novel Laplacian optimization structure is introduced and designed by fully considering appearance features and motion information, which makes the supervoxel consist in spatial and temporal domains. Our method can handle long videos with weak boundaries and complex textures. We have implemented and evaluated our method on the video data sets under differently quantitative segmentation quality metrics. The proposed method is suitable for both long and short videos that contain motion changes or scene changes. Experimental results have shown that our approach outperforms the state-of-the-art supervoxel methods in terms of quantitative metrics as well as qualitative visual results.

References

[1] E. Sharon, A. Brandt, and R. Basri, "Segmentation and boundary detection using multiscale intensity measurements," in *Proc. IEEE CVPR*, Jun. 2001, pp. 469–476.

[2] J. Shen, Y. Du, and X. Li, "Interactive segmentation using constrained Laplacian optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1088–1100, Jul. 2014.

[3] C. Fowlkes, S. Belongie, and J. Malik, "Efficient spatio-temporal grouping using the Nyström method," in *Proc. IEEE CVPR*, Jun. 2001, pp. 231–238.

[4] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[5] X. Zhu, J. Kandola, Z. Ghahramani, and J. D. Lafferty, "Nonparametric transforms of graph kernels for semi-supervised learning," in *Proc. NIPS*, 2004, pp. 1641–1648.

[6] S. Baker and I. Matthews, "Lucas–Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[7] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[8] S. Paris and F. Durand, "A topological approach to hierarchical segmentation using mean shift," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[9] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch, "Broad expertise retrieval in sparse data environments," in *Proc. 30th ACM SIGIR*, 2007, pp. 551–558.

[10] P. Serdyukov, H. Rode, and D. Hiemstra, "Modeling expert finding as an absorbing random walk," in *Proc. 31st ACM SIGIR*, 2008, pp. 797–798.

[11] B. Nadler, N. Srebro, and X. Zhou, "Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data," in *Proc. NIPS*, 2009, pp. 1330–1338.

[12] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.

[13] J. Shen, H. Sun, J. Jia, H. Zhao, X. Jin, and S. Fang, "A unified framework for designing textures using energy optimization," *Pattern Recognit.*, vol. 43, no. 2, pp. 457–469, 2010.

[14] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE CVPR*, Jun. 2010, pp. 2141–2148.

[15] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. ECCV*, 2010, pp. 211–224.

[16] A. Y. C. Chen and J. J. Corso, "Propagating multi-class pixel labels throughout video frames," in *Proc. Western New York Image Process. Workshop (WNYIPW)*, Nov. 2010, pp. 14–17.

[17] B. Kveton, M. Valko, A. Rahimi, and L. Huang, "Semi-supervised learning with max-margin graph cuts," in *Proc. AISTATS*, 2010, pp. 421–428.

[18] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Spatiotemporal closure," in *Proc. ACCV*, Nov. 2010, pp. 369–382.

[19] J. Shen, X. Yang, Y. Jia, and X. Li, "Intrinsic images using optimization," in *Proc. IEEE CVPR*, Jun. 2011, pp. 3481–3487.

[20] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *Proc. IEEE ECCV*, Sep. 2010, pp. 268–281.

[21] 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.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[22] F. Kawaf and T. Stephen, "Online shopping environments in fashion shopping: An S-O-R based review," *Marketing Rev.*, vol. 12, no. 2, pp. 161–180, 2012.

[23] M. D. Collins, J. Xu, L. Grady, and V. Singh, "Random walks based multi-image segmentation: Quasiconvexity results and GPU-based solutions," in *Proc. IEEE CVPR*, Jun. 2012, pp. 1656–1663.

[24] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Proc. IEEE CVPR*, Jun. 2012, pp. 1202–1209.

[25] X.-M. Wu, Z. Li, A. M. So, J. Wright, and S.-F. Chang, "Learning with partially absorbing random walks," in *Proc. NIPS*, 2012, pp. 3077–3085.

[26] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proc. IEEE ECCV*, Oct. 2012, pp. 626–639.

[27] F. Galasso, R. Cipolla, and B. Schiele, "Video segmentation with superpixels," in *Proc. ACCV*, Nov. 2012, pp. 760–774.

[28] J. Shen, D. Wang, and X. Li, "Depth-aware image seam carving," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1453–1461, Oct. 2013.

[29] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool, "Online video SEEDS for temporal window objectness," in *Proc. IEEE ICCV*, Dec. 2013, pp. 377–384.

[30] R. Trichet and R. Nevatia, "Video segmentation with spatio-temporal tubes," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2013, pp. 330–335.

[31] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 817–827, Jun. 2014.

[32] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, nos. 1–2, pp. 42–59, 2014.

[33] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.
[34] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3241–3253, Aug. 2014.
[35] J. Chang, D. Wei, and J. W. Fisher, "A video representation using temporal superpixels," in *Proc. IEEE CVPR*, Jun. 2013, pp. 2051–2058.
[36] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann, "Temporally consistent superpixels," in *Proc. IEEE ICCV*, Dec. 2013, pp. 385–392.
[37] T. Luo, J. Shen, and X. Li, "Accurate normal and reflectance recovery using energy optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 212–224, Feb. 2015.
[38] X. Dong, J. Shen, and L. Van Gool, "Segmentation using subMarkov random walk," in *Energy Minimization Methods in Computer Vision and Pattern Recognition* (Lecture Notes in Computer Science), vol. 8932. Berlin, Germany: Springer-Verlag, 2015, pp. 237–248.

Yuling Liang is working toward the M.S. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

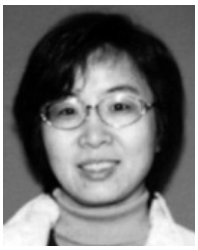
Her research interests include random walk and video segmentation algorithms.

Jianbing Shen (M'11–SM'12) is currently a Full Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. He has authored about 50 journals and conference papers, which have been published in such as IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE Computer Vision and Pattern Recognition, IEEE International Conference on Computer Vision, and IEEE International Conference on Multimedia and Expo. His research interests include computer vision and multimedia processing.

Mr. Shen received many flagship honors, including the Fok Ying Tung Education Foundation from the Ministry of Education, the Program for Beijing Excellent Youth Talents from the Beijing Municipal Education Commission, and the Program for New Century Excellent Talents in University from Ministry of Education.

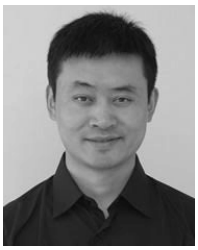
Xingping Dong is working toward the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His research interests include random walk and machine learning algorithms.



Hanqiu Sun (M'98) received the M.S. degree in electrical engineering from University of British Columbia, Vancouver, BC, Canada, and the Ph.D. degree in computer science from University of Alberta, Alberta, ON, Canada.

She is an Associate Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. Her research interests include virtual reality, interactive graphics/animation, real-time hypermedia, virtual surgery, mobile image/video synopsis and navigation, and touch-enhanced simulations.



Xuelong Li (M'02–SM'07–F'12) is a Full Professor with the State Key Laboratory of Transient Optics and Photonics, Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.