Multiscale Superpixels and Supervoxels Based on Hierarchical Edge-Weighted Centroidal Voronoi Tessellation

Youjie Zhou, *Student Member, IEEE*, Lili Ju, and Song Wang, *Senior Member, IEEE*

***Abstract*— Superpixels and supervoxels play an important role in many computer vision applications, such as image segmentation, object recognition, and video analysis. In this paper, we propose a new hierarchical edge-weighted centroidal Voronoi tessellation (HEWCVT) method for generating superpixels/supervoxels in multiple scales. In this method, we model the problem as a multilevel clustering process: superpixels/supervoxels in one level are clustered to obtain larger size superpixels/supervoxels in the next level. In the finest scale, the initial clustering is directly conducted on pixels/voxels. The clustering energy involves both color similarities and boundary smoothness of superpixels/supervoxels. The resulting superpixels/supervoxels can be easily represented by a hierarchical tree which describes the nesting relation of superpixels/supervoxels across different scales. We first investigate the performance of obtained superpixels/supervoxels under different parameter settings, then we evaluate and compare the proposed method with several state-of-the-art superpixel/supervoxel methods on standard image and video data sets. Both quantitative and qualitative results show that the proposed HEWCVT method achieves superior or comparable performances with other methods.**

***Index Terms*— Superpixel, supervoxel, image segmentation, hierarchical image segmentation, edge-weighted centroidal Voronoi tessellation.**

I. Introduction

BY COMPACTLY representing 2D/3D images using a collection of perceptually meaningful atomic regions/volumes, superpixels/supervoxels have become a standard tool in many vision applications [1]–[4]. Good superpixels/supervoxels usually have several desired properties [5], [6]: 1) all the pixels/voxels in a superpixel/supervoxel share similar features, such as color and/or texture; 2) all the generated superpixels/supervoxels are compact and uniformly distributed; 3) the superpixel/supervoxel boundaries well align with structural boundaries in the original image.

In addition, many vision applications require the use of multiscale superpixels/supervoxels with different coarse levels to better infer the high-level structural information [3], [4], [7]–[11]. Multiscale superpixels/supervoxels can usually be obtained by varying certain configurations, such as the number of or the average size of superpixels/supervoxels. However, simply varying these configurations may not generate multiscale superpixels/supervoxels with boundary consistency, i.e., the boundaries in a coarser level may not be drawn from the boundaries in a finer level. This way, the superpixels/supervoxels in different scales may not show a hierarchical nested relations, which is important for inferring high-level structural information [3], [4], [7], [8], [12].

In this paper, we develop a *hierarchical edge-weighted centroidal Voronoi tessellation* (HEWCVT) method for generating multiscale superpixels/supervoxels. In this method, superpixels/supervoxels in a finer level is clustered to achieve superpixels/supervoxels in a new coarser level. In the finest level, all the image pixels/voxels are taken as the entities for HEWCVT clustering. This iterative clustering process guarantees the hierarchical nested relations across different levels. In HEWCVT method, the clustering energy consists of not only a term that measures the color/feature similarity between superpixels/supervoxels, but also an edge term that measures the boundary smoothness of the obtained superpixels/supervoxels. With this edge term, the proposed HEWCVT method is able to produce superpixel/supervoxel boundaries better aligned with the underlying structural boundaries in each level. Examples of superpixels resulting from the proposed method are shown in Fig. 1. In the experiments, we justify the proposed method by qualitatively and quantitatively comparing its performance against several other state-of-the-art superpixels/supervoxels methods on three standard image/video datasets.

A. Related Work

In [13], the K-means algorithm is extended to a Simple Linear Iterative Clustering (SLIC) algorithm for generating superpixels, by restricting the search space inside each superpixel and combining color and spatial proximity into a

Manuscript received October 24, 2014; revised February 16, 2015 and April 30, 2015; accepted June 8, 2015. Date of publication June 24, 2015; date of current version July 30, 2015. This work was supported in part by the National Science Foundation under Grant IIS-1017199, in part by the Air Force Office of Scientific Research, Arlington, VA, USA, under Grant FA9550-11-1-0327, and in part by the Open Project Program through the State Key Laboratory of CAD and CG, Zhejiang University, Hangzhou, China, under Grant A1420. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Olivier Bernard.

Y. Zhou and S. Wang are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: zhou42@email.sc.edu; songwang@cec.sc.edu).

L. Ju is with the Department of Mathematics, University of South Carolina, Columbia, SC 29208 USA (e-mail: ju@math.sc.edu).

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/TIP.2015.2449552

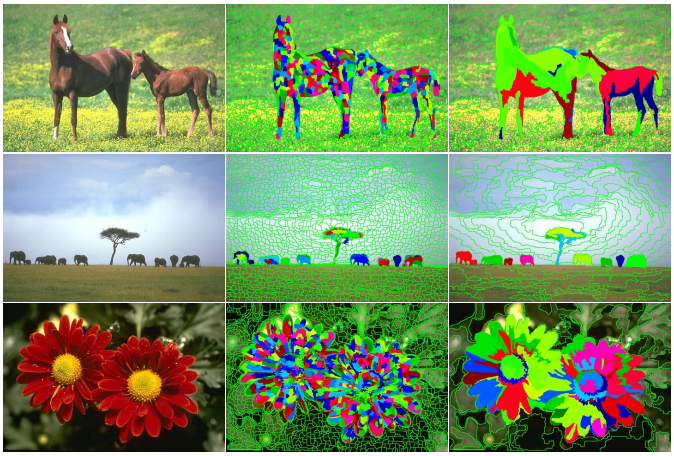


Fig. 1. Sample results of the proposed HEWCVT method on 2D images: the original image (left column), superpixels constructed on the finest level of hierarchy (middle column) and superpixels constructed on the highest level of hierarchy (right column). Superpixels of objects are visualized by colorful patches. The hierarchical nested relations are guaranteed across scales, e.g. between middle and right columns.

weighted distance function. SLIC is conducted only on pixels and may not guarantee the hierarchical nested relations across different scales. In [14], the Lazy Random Walk (LRW) algorithm, incorporated with a new segmentation energy function, was proposed for superpixel construction. Due to its high complexity, it is difficult to apply LRW on large size images or to be extended for supervoxel construction on videos. In [15], superpixels/supervoxels are generated by stitching overlapping patches such that each pixel belongs to only one of patches. This algorithm is built upon the GraphCut-based MRF energy optimization framework. However, our later experiments show that it is often difficult to catch structural boundaries when the number of superpixels is small.

More related to the proposed HEWCVT method is the VCells algorithm developed in [16]. VCells generates superpixels by using a modified edge-weighted centroidal Voronoi tessellation (EWCVT) model [17]–[19] on pixels. Starting from a spatially uniform tessellation of the image space, VCells iteratively moves the superpixel boundaries to better align with the structural boundaries. Except for minimizing an energy involving color/intensity similarity and boundary smoothness, VCells also enforces the connectivity of each superpixel in the optimization. However, as many other methods, VCells is only conducted on pixels and cannot produce multiscale superpixels with hierarchical nested relations. In the HEWCVT method, we propose a multiscale clustering algorithm, and a boundary smoothness measurement for superpixels/supervoxels.

Multiscale superpixels/supervoxels are usually obtained directly from multiscale 2D/3D image segmentation algorithms by varying certain configurations as discussed previously. Several multiscale image/video segmentation algorithms have been used for superpixel/supervoxel construction recently [1], [6], [20]. All of them are based on graph aggregation using color/feature similarity and therefore the obtained superpixels/supervoxels are nested across different scale levels. However, without specific constraints on the superpixel/supervoxel connectivity and boundary smoothness, the resulting superpixels/supervoxels could be quite fragmented and scattered (see Fig. 14). We justify the performance of the proposed HEWCVT method by comparing with many of these state-of-the-art superpixels/supervoxels algorithms.

The remaining part of the paper is organized as follows. First we review the EWCVT model and algorithms in Section II. We then propose the new HEWCVT model and develop algorithms for HEWCVT construction in Section III. Quantitative and qualitative evaluations and discussions are presented in Section IV. Finally we give concluding remarks in Section V.

II. Edge-Weighted Centroidal Voronoi Tessellation

Let $\mathbb{U} = \{\vec{u}(i,j) \mid (i,j) \in \mathbb{I}\}$ denote the set of color or feature vectors of a 2D image $\mathbb{I}$ (extension to 3D images will be discussed in Section III-F), where $\vec{u}$ is the color/feature function associated with $\mathbb{I}$. In the experiments, we use the `Lab` color feature. For L arbitrary color vectors $\mathcal{W} = \{\vec{w}_l\}_{l=1}^{L}$ (called *generators*), the corresponding *Voronoi tessellation* of $\mathbb{U}$ is defined as $\mathcal{V} = \{V_l\}_{l=1}^{L}$ such that $V_l = \{\vec{u}(i,j) \in \mathbb{U} \mid \|\vec{u}(i,j) - \vec{w}_l\| < \|\vec{u}(i,j) - \vec{w}_m\|,\ m = 1,\ldots,L \text{ and } m \neq l\}$, where $\|\cdot\|$ is a distance function defined on $\mathbb{U}$. Given a weight or density function ρ defined on each pixel of $\mathbb{I}$, we can further define the centroid (i.e., the center of mass) of each Voronoi region V_l as $\vec{w}_l^*$ such that $\vec{w}_l^* = \arg\min_{\vec{w} \in V_l} \sum_{\vec{u}(i,j) \in V_l} \rho(i,j)\|\vec{u}(i,j) - \vec{w}\|^2$.

If the generators $\{\vec{w}_l\}_{l=1}^{L}$ of the Voronoi regions $\{V_l\}_{l=1}^{L}$ of $\mathbb{U}$ are the same as their corresponding centroids, i.e.,

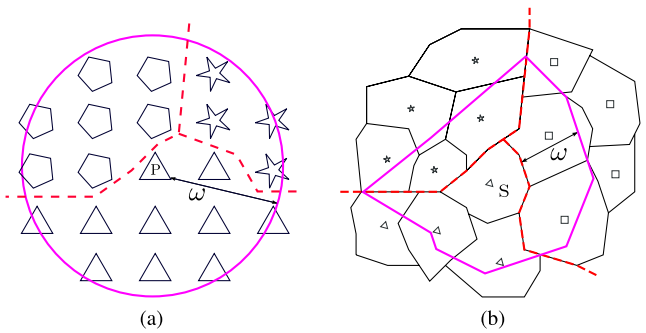
$$\vec{w}_l = \vec{w}_l^*,\ l = 1,\ldots L,$$

then we call the Voronoi tessellation $\{V_l\}_{l=1}^{L}$ a *centroidal Voronoi tessellation* (CVT) of $\mathbb{U}$. Since each Voronoi region V_l stands for a cluster in the color space we can easily construct a corresponding partition of the 2D image $\mathbb{I}$ using the correspondence between pixel indices and color vectors through $\vec{u}$. Let $\mathcal{C} = \{C_l\}_{l=1}^{L}$ denote a clustering of the physical space of the image $\mathbb{I}$, then the CVT clustering energy can be defined as

$$E_{cvt}(\mathcal{C}, \mathcal{W}) = \sum_{l=1}^{L} \sum_{(i,j) \in C_l} \rho(i,j)\|\vec{u}(i,j) - \vec{w}_l\|^2. \qquad (1)$$

The construction of CVTs often can be viewed as a clustering energy minimization problem, i.e., solving $\min_{(\mathcal{C},\mathcal{W})} E_{cvt}(\mathcal{C}, \mathcal{W})$. The Lloyd method [21], [22] (equivalent to the weighted K-means) has been widely used to compute CVTs, which is basically iterations between constructing Voronoi regions and centroids. Assume that the Euclidean distance is used for the color space, then we simply have the centroid of the cluster C_l as $\vec{w}_l^* = \sum_{(i,j) \in C_l} \rho(i,j)\vec{u}(i,j) / \sum_{(i,j) \in C_l} \rho(i,j)$.

In order to enforce the smoothness of segment boundaries, a special edge energy was proposed and added into the clustering energy [16], [17]. Specifically, let us define an indicator function $\chi(i,j) : \mathbb{N}_\omega(i,j) \to \{0,1\}$ on the neighborhood of



(a) (b)

Fig. 2. An illustration of boundary smoothness measurement. Dash lines are cluster boundaries. Pink curve indicates the local neighborhood area for smoothness measurement. (a) Boundary smoothness measurement for a pixel P. Each pixel is visualized as a polygon and its shape stands for the pixel's current cluster assignment. (b) Boundary smoothness measurement for a superpixel S. Each polygon represents a superpixel and the shape of its center marker stands for the superpixel's current cluster assignment.

pixel (i, j) with radius ω as

$$\chi_{(i,j)}(i', j') = \begin{cases} 1 & \text{if } \pi(i', j') \neq \pi(i, j) \\ 0 & \text{otherwise} \end{cases}$$

where $\pi(i, j)$ tells the cluster index that (i, j) belongs to. Then the edge energy is defined as

$$E_{edge}(\mathcal{C}) = \sum_{(i,j)\in\mathbb{I}} \sum_{(i',j')\in\mathbb{N}_\omega(i,j)} \chi_{(i,j)}(i', j'). \tag{2}$$

Figure 2a illustrates the boundary smoothness measurement on a single pixel. It has been shown in [17] that $E_{edge}(\mathcal{C})$ is proportional to the total length of boundaries in $\mathcal{C}$ in the limit. Finally, the edge-weighted CVT clustering energy can be defined as

$$E_{ewcvt}(\mathcal{C}, \mathcal{W}) = E_{cvt}(\mathcal{C}, \mathcal{W}) + \lambda E_{edge}(\mathcal{C}) \tag{3}$$

where λ is a weight parameter balancing the clustering energy and the edge energy. Construction of EWCVTs is equivalent to solving the minimization problem $\min_{(\mathcal{C},\mathcal{W})} E_{ewcvt}(\mathcal{C}, \mathcal{W})$. An edge-weighted distance function from a pixel (i, j) to a cluster center (generator) $\vec{w}_k$ was derived for the energy E_{ewcvt} as

$$dist((i, j), \vec{w}_k) = \sqrt{\rho(i, j)\|\vec{u}(i, j) - \vec{w}_k\|^2 + 2\lambda\tilde{n}_k(i, j)} \tag{4}$$

where $\tilde{n}_k(i, j) = |\mathbb{N}_\omega(i, j)| - n_k(i, j) - 1$ with $n_k(i, j) = \sum_{(i',j')\in\mathbb{N}_\omega(i,j)} \pi(i', j') \neq k$. Based on the above distance function, some efficient algorithms for constructing EWCVTs are suggested in [17] based on the K-means type techniques. We specially note that a cluster C_l produced by the EWCVT method may consist of physically disconnected regions in the image $\mathbb{I}$, i.e., multiple segments.

III. Hierarchical Edge-Weighted Centroidal Voronoi Tessellation

The proposed hierarchical method begins with an over-segmentation on pixels using the a modified EWCVT algorithm that strictly enforces the simple-connectivity of superpixels [16]. This oversegmentation is taken as the finest level of superpixels in the hierarchy. For the higher levels, we merge finer level superpixels with similar color features, meanwhile preserve superpixel connectivity and enforce the boundary smoothness of superpixels.

Algorithm 1 (Finest Level Superpixel Algorithm)

Input: The target 2D image $\mathbb{I}$ and the color function $\vec{u}$
M_1: Number of desired superpixels
$niter$: Maximum number of iterations
0 **Initialization**: Construct the initial superpixels of $\mathbb{I}$, $\{C_l\}_{l=1}^{M_1}$ using the k-means with the Euclidean distance and the feature function $\vec{u}^+$.
1 **FOR** each $C_k \in \{C_l\}_{l=1}^{M_1}$
2 Compute centroid $\vec{w}_k = \frac{1}{|C_k|} \sum_{(i,j)\in C_k} \vec{u}(i, j)$
3 **FOR** $iter = 1$ to $niter$
4 Create the set of boundary pixels $\mathcal{B}$
5 **FOR** each $(i, j) \in \mathcal{B}$
6 Find the closest centroid to the pixel (i, j)
$\vec{w}_k \in \{\vec{w}_l \mid l \in \pi(\mathcal{N}_4(i, j))\}$
w.r.t. the edge-weighted distance (Eq. (4))
7 **IF** $\pi(i, j) \neq k$
8 Set $\tilde{k} = \pi(i, j)$ and $\pi(i, j) = k$
9 Update $\vec{w}_k$, $\vec{w}_{\tilde{k}}$
10 **IF** there is no cluster index change
11 Break
12 Perform the simple-connectivity filtering
Output: The cluster/superpixel index function π

A. Finest Level

At the finest level we deal with the generation of superpixels directly from the image pixels. Let M_1 be the desired number of superpixels. Similar to the VCells algorithm proposed in [16], we first use the classic K-means with the Euclidean norm on pixel coordinates $\mathbb{I}$, to generate M_1 simply-connected and quasi-uniformly distributed superpixels on the input image. We also set $\rho \equiv 1$ here. Next we apply the VCells algorithm to the initial superpixel configuration where we only allow transferring of boundary pixels between neighbor clusters at each iteration. The whole algorithm is described in Algorithm 1. If $\pi(i, j)$ is different from the label of at least one of its 4 neighbors, i.e., $(i \pm 1, j)$ or $(i, j \pm 1)$, we say (i, j) is a boundary pixel, and denote $\mathcal{B}$ as the set of all boundary pixels. We remark that each pixel moving between neighbor clusters in Algorithm 1 will decrease the energy E_{ewcvt}, thus Algorithm 1 guarantees monotonic decreasing of E_{ewcvt} along the iterations till it terminates, see [17] for detailed discussions.

There is no guarantee to preserve the simple-connectivity property of each segment in the algorithm above. Thus in the end we perform a filtering step to further enforce the simple-connectivity of superpixels, which is widely used in other superpixel algorithms [5], [13], [15], [16] and will be described in Section III-C.

B. Higher Levels

At a higher level q ($q > 1$), we already have a superpixel from the previous level $q-1$, $\mathbb{S} = \{S_m\}_{m=1}^{M_{q-1}}$. Given the desired

number of superpixels M_q ($M_q < M_{q-1}$) in Level q, we will merge adjacent superpixels to reach that goal according to minimization of certain energy function. Each superpixel is treated as a point and we will cluster them into M_q simply-connected parts, where $M_q < M_{q-1}$ is the desired number of superpixels in Level q. This way we can easily build a tree structure for superpixels between these two levels.

The initialization step is different from that in the finest level. The most intuitive idea is to apply the K-means clustering on the set of average coordinates of all superpixels constructed in the previous level. However, the merged superpixels may not be simply-connected. Instead, we first build a superpixel graph $G = (V, E, \mathcal{E})$, where V consists of all the previous level's superpixels $\{S_m\}_{m=1}^{M_{q-1}}$ and E is the set of all pairs of neighbor superpixels. The edge weight for $(S_a, S_b) \in E$ is defined as

$$\mathcal{E}(S_a, S_b) = \frac{\|\vec{u}(S_a) - \vec{u}(S_b)\|}{\max_{(S_i,S_j)\in E} \|\vec{u}(S_i) - \vec{u}(S_j)\|} \tag{5}$$

where $\vec{u}(S) = \frac{1}{|S|}\sum_{(i,j)\in S} \vec{u}(i, j)$ denotes the average color vector of all the pixels belonging to the superpixel S. Then the superpixel graph G will be partitioned into M_q subgraphs which are considered as initialized superpixels at level q. The proposed HEWCVT method will refine initialized superpixels later. Therefore, any graph partition algorithm can be used for this initialization. Based on algorithm efficiency and code availability, we choose the METIS algorithm [23] here.

We define the density function ρ on $\mathbb{S}$ as $\rho(S_m) = |S_m|$, i.e., the number of pixels contained in the superpixel $S_m \in \mathbb{S}$. Let $\mathcal{C}^{sp} = \{C_l^{sp}\}_{l=1}^{M_q}$ be a clustering of $\mathbb{S}$ and $\mathcal{W} = \{\vec{w}_l\}_{l=1}^{M_q}$ be an arbitrary set of color vectors. Then we define the new CVT clustering energy as

$$E_{cvt-sp}(\mathcal{C}^{sp}, \mathcal{W}) = \sum_{l=1}^{M_q} \sum_{S\in C_l^{sp}} \rho(S)\|\vec{u}(S) - \vec{w}_l\|^2. \tag{6}$$

In order to measure the boundary length (or the smoothness) of superpixels, we propose an edge energy for the superpixel image. As illustrated in Fig. 2b, we define the local neighborhood $\mathbb{N}_\omega(S)$ for a superpixel $S \in \mathbb{S}$ as

$$\mathbb{N}_\omega(S) = \bigcup_{(i,j)\in\mathcal{B}(S)} \mathbb{N}_\omega(i, j) - S$$

where $\mathcal{B}(S)$ denotes the set of all boundary pixels of the superpixel S. Then we define the edge energy as

$$E_{edge-sp}(\mathcal{C}^{sp}) = \sum_{S\in\mathbb{S}} \sum_{(i,j)\in\mathbb{N}_\omega(S)} \Gamma_S(i, j) \tag{7}$$

where $\Gamma_S(i, j) : \mathbb{N}_\omega(S) \to \{0, 1\}$ is an indicator function, similar as $\chi(i, j)$ in Eq. (2), and is defined by

$$\Gamma_S(i, j) = \begin{cases} 1 & \text{if } \pi(i, j) \neq \pi(S) \\ 0 & \text{otherwise} \end{cases}$$

where $\pi(S)$ returns the cluster index of the superpixel S in $\mathcal{C}^{sp}$.

Finally, the edge-weighted CVT clustering energy for superpixels can be defined as

$$E_{ewcvt-sp}(\mathcal{C}^{sp}, \mathcal{W}) = E_{cvt-sp}(\mathcal{C}^{sp}, \mathcal{W}) + \lambda E_{edge-sp}(\mathcal{C}^{sp}). \tag{8}$$

Algorithm 2 (Higher Level Superpixel Merging Algorithm)

Input: The superpixel image $\mathbb{S}$ and the color function $\vec{u}$
M_q: Number of desired superpixels
$niter$: Maximum number of iterations
0 **Initialization**: Construct the superpixel graph G and partition $\mathbb{S}$ into M_q simply-connected regions $\{C_l^{sp}\}_{l=1}^{M_q}$ using METIS [23]
1 **FOR** each $C_k^{sp} \in \{C_l^{sp}\}_{l=1}^{M_q}$
2 Compute centroid $\vec{w}_k = \frac{\sum_{S\in C_k^{sp}} \rho(S)\vec{u}(S)}{\sum_{S\in C_k^{sp}} \rho(S)}$
3 **FOR** $iter = 1$ to $niter$
4 Create the set of boundary superpixels $\mathcal{B}(\mathbb{S})$
5 **FOR** each $S \in \mathcal{B}(\mathbb{S})$
6 Find the closest centroid to S
$\vec{w}_k \in \{\vec{w}_l \mid l \in \pi(\mathcal{N}(S)\}$
w.r.t. the edge-weighted distance (Eq. (9))
7 **IF** $\pi(S) \neq k$
8 Set $\tilde{k} = \pi(S)$ and $\pi(S) = k$
9 Update $\vec{w}_k$, $\vec{w}_{\tilde{k}}$
10 **IF** there is no cluster index change
11 Break
12 Perform the simple-connectivity filtering
Output: The cluster/superpixel index function π

We can derive the distance from a superpixel S to a cluster center $\vec{w}_k$ corresponding to the above energy as

$$dist(S, \vec{w}_k) = \sqrt{\rho(S) \parallel \vec{u}(S) - \vec{w}_k \parallel^2 + 2\lambda\tilde{n}_k(S)} \tag{9}$$

where $\tilde{n}_k(S)$ measures the number of inconsistent pixels in the neighborhood of the superpixel S: $\tilde{n}_k(S) = |\mathbb{N}_\omega(S)| - n_k(S)$ with $n_k(S) = \sum_{(i,j)\in\mathbb{N}_\omega(S)} \pi(i, j) \neq k$.

Furthermore, in order to keep superpixels simply connected, we follow the idea in the finest level (Section III-A), i.e., only superpixels located at cluster boundaries will be considered during the clustering, and we only allow cluster index change among adjacent clusters. The whole algorithm is described in Algorithm 2. We again remark that similar to Algorithm 1, Algorithm 2 guarantees monotonic decreasing of $E_{ewcvt-sp}$ along the iterations till it terminates.

C. Simple-Connectivity Filtering

Although we have enforced that the pixel/superpixel transferring can only occur among adjacent clusters, due to the image noises, few superpixels may still break into several disconnected parts and/or contain holes (especially in 3D cases). Thus after the HEWCVT clustering process, we merge small ($|S| \leq \varepsilon$) and isolated superpixels into their surroundings. Similar post-step has been applied in several state-of-the-art superpixel/supervoxel methods [5], [13], [15], [16].

Specifically, in the finest level, for each pixel p in a small or isolated superpixel S, we first locate its nearest neighbor pixel p' based on their coordinate distance in another superpixel S', and then simply merge pixel p into S'. In the higher levels, however, we need to consider each small or isolated superpixel as a whole. Otherwise, merging inside pixels individually

may produce new and inaccurate superpixel boundaries. Here, we first associate each inside pixel p in S with its nearest neighboring superpixel S' based on the coordinate distance between p and the center of S', and then we merge the small or isolated superpixel S with a neighboring superpixel that has the largest number of associated inside pixels.

D. Adaptive Determination of λ

The energy weight parameter λ defined in Eqs. (3) and (8) balances the ratio between the CVT clustering energy E_{cvt} (or E_{cvt-sp}) and the edge energy E_{edge} (or $E_{edge-sp}$). However, these energies are varying for different images/videos and/or changing along different scale levels. Especially in video segmentation, different videos also have variant number of frames and frame rates. Thus a fixed λ is obviously inappropriate. Instead we aim at controlling the ratio between E_{cvt} and λE_{edge}. Therefore, given an predetermined energy ratio θ that $\frac{E_{cvt}}{\lambda E_{edge}} = \frac{E_{cvt-sp}}{\lambda E_{edge-sp}} = \theta$, we can adjust λ adaptively by setting $\lambda^{(iter)} = \frac{E_{cvt}^{(iter-1)}}{\theta E_{edge}^{(iter-1)}}$ at each iteration in Algorithm 1 and $\lambda^{(iter)} = \frac{E_{cvt-sp}^{(iter-1)}}{\theta E_{edge-sp}^{(iter-1)}}$ in Algorithm 2.

E. Complexity and Convergence Analysis

The Finest Level Superpixel Algorithm 1 is equivalent to the VCells, and it contains two major steps: 1) initializing boundary pixels $\mathcal{B}$ which takes $\mathcal{O}(N)$ where N is the total number of image pixels; 2) EWCVT algorithm only considering boundary pixels which takes $\mathcal{O}\left(K\sqrt{M_1 \cdot N}\right)$ where K is the total number of iterations and M_1 is desired number of superpixels in the finest level. We refer the reader to [16] for more details about the complexity analysis of VCells.

Excluding the cost of boundary pixels initialization, for the Higher Level Superpixel Merging Algorithm 2, as we only consider the boundary superpixels, thus the computational cost in each iteration is $\mathcal{O}\left(n_{\mathcal{B}(S)} \cdot n_{\mathcal{B}}\right)$, where $n_{\mathcal{B}(S)}$ is the number of boundary superpixels, and $n_{\mathcal{B}}$ is the number of boundary pixels utilized for measuring proposed superpixel boundary smoothness. Each merged superpixel should contain approximately $\frac{M_{q-1}}{M_q}$ superpixels from the previous level, where M_{q-1} is the number of superpixels in the previous level and M_q is desired number of merged superpixels in current level, thus there are $\sqrt{\frac{M_{q-1}}{M_q}}$ boundary superpixels. Similarly the number of boundary pixels in a superpixel can be approximated by $\sqrt{\frac{N}{M_{q-1}}}$. Therefore,

$$\mathcal{O}\left(n_{\mathcal{B}(S)} \cdot n_{\mathcal{B}}\right) \sim \mathcal{O}\left(M_q \cdot \sqrt{\frac{M_{q-1}}{M_q}} \cdot \sqrt{\frac{N}{M_{q-1}}}\right) \sim \mathcal{O}\left(\sqrt{M_q \cdot N}\right).$$

For K iterations, we have $\mathcal{O}\left(K\sqrt{M_q \cdot N}\right)$. Overall, the complexity of proposed HEWCVT method is $\mathcal{O}\left(N + K\sqrt{M \cdot N}\right) = \mathcal{O}(N)$ where M is desired number of superpixels in a hierarchy level.

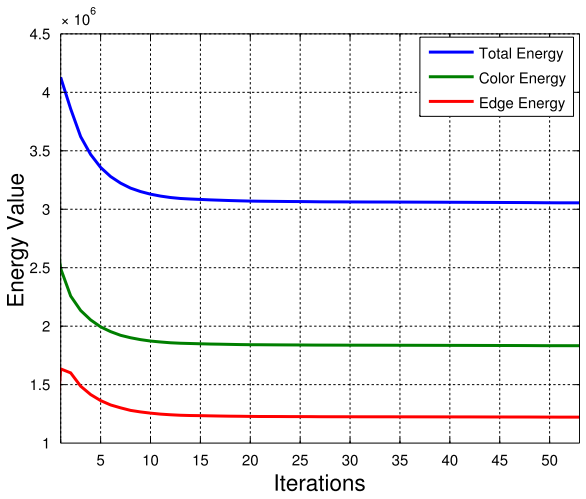


Fig. 3. An illustration of energy convergence of the proposed HEWCVT, with $\theta = 1.5$, when constructing superpixels on a sample image.

Both of Algorithm 1 and Algorithm 2 will converge to a local minimum of the defined HEWCVT energy. We illustrate the value change of the total energy, the color energy and the edge energy along iterations on a sample image in Fig. 3. In this example, we set the desired ratio between the color energy and the edge energy, i.e., $\theta = 1.5$. We can see that, all three energies decrease quickly and converge to local minimal values, while the ratio between the color energy and the edge energy always remains the same as the desired value. For mathematical proofs on EWCVT-based energy convergence, please see [17] for details.

F. Extension to Supervoxels

We can easily extend the proposed hierarchical method into 3D case. The major difference is the neighbor system among voxels and supervoxels. Instead of 4-neighbor system in 2D case, we use 6-neighborhood for the voxel level oversegmentation. We note that more complex neighbor systems also can be used.

Another issue is that in the 2D case we assume the units of all coordinate directions are the same. For 3D images this assumption is still valid in most situations. However, for video data, the unit of the temporal direction could be different from those of spatial axises. Therefore for video data, one could use $\mathbb{I}_{3D} = H * (i, j, k)^T$ where $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \gamma_k \end{bmatrix}$ is a scaling matrix and γ_k is data dependent. In the video experiments, we just simply used $H = I_{3\times 3}$ and it worked fine for the test video data.

IV. Experiments

We tested the proposed HEWCVT method on three standard image/video segmentation benchmarks which have been widely used for evaluating the performance of superpixels/supervoxels:

- the Berkeley Segmentation Dataset and Benchmark (BSDS300) [24], which consists of 300 color images of

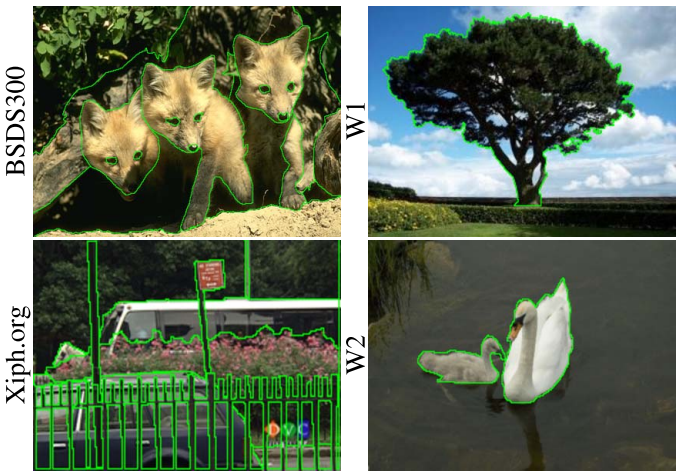


Fig. 4. Sample images and their human annotated boundaries for superpixel and supervoxel evaluations.

dimensions 481 × 321 or 321 × 481. Each image has been annotated by different subjects, thus obtained ground truth segments are at varying levels of granularity.

- the Weizmann image dataset [25], which consists of 200 color images of size approximately 300 × 225. Different from the BSDS300 dataset, subjects only annotated contours of the foreground objects. Based on the number of objects in an image, the whole dataset consists of two parts: images with single object (W1) and images with two objects (W2).
- the Xiph.org video dataset [26], which consists of 8 color videos of approximately 85 frames (240 × 160) each. The videos have been labeled frame by frame with temporal consistency taken into consideration.

Sample images overlaid with corresponding ground-truth boundaries are shown in Fig. 4.

A. Evaluation Metrics

In order to quantitatively evaluate the performance of superpixels/supervoxels, we used human labeled segmentation as the ground truth because the superpixel/supervoxel boundaries should well align with the structural boundaries. Based on the ground truth, we applied three standard superpixel/supervoxel measurements: boundary recall, under-segmentation error and segmentation accuracy [6], [13], [15]. Note that in this paper we propose a superpixel/supervoxel method. Therefore, we use superpixel/supervoxel metrics instead of image segmentation metrics, e.x., metrics from the Berkeley segmentation dataset. For each of the three metrics we report the average values on each dataset.

1) Boundary Recall: This metric measures the fraction of ground truth boundaries that fall within a certain distance t of at least one superpixel/supervoxel boundary. It is formulated as

$$BR = \frac{\sum_{p\in\mathcal{B}(g)} \mathcal{I}\left[\min_{q\in\mathcal{B}(s)} \parallel p-q \parallel < t\right]}{|\mathcal{B}(g)|} \tag{10}$$

where $\mathcal{B}(g)$ is the union set of ground truth boundaries, $\mathcal{B}(s)$ is the union set of superpixel/supervoxel boundaries and $\mathcal{I}$ is an indicator function that returns 1 if a superpixel/supervoxel boundary pixel is close enough to the ground truth boundaries. We set $t = 2$ for both images and videos as in [6] and [13]. In general the larger the number of superpixels/supervoxels, the more boundaries, and the better the boundary recall.

TABLE I
AVERAGE RUNNING TIME OF DIFFERENT SUPERPIXEL/SUPERVOXEL ALGORITHMS ON SEVERAL IMAGE/VIDEO DATASETS

	HEWCVT	VCells[16]	GraphCut[15]	SLIC[13]	LRW[14]
BSDS300	1.42s	1.32s	5.39s	0.27s	1090.55s
W1	0.24s	0.62s	1.97s	0.13s	1160.83s
W2	0.23s	0.55s	2.16s	0.12s	854.57s

	HEWCVT	GBH[1]	SWA[20]
Xiph.org	0.54s	0.47s	0.13s

2) Undersegmentation Error: This metric measures the fraction of superpixels/supervoxels that is leaked across the boundary of the ground-truth segments. For each ground truth segment g_i, we calculate the "bleeding" area of superpixels/supervoxels that overlap with g_i. It is formulated as

$$UE = \frac{\sum_{i=1}^{G}\left[\left(\sum_{s_j:s_j\cap g_i>r} \mid s_j \mid\right) - \mid g_i \mid\right]}{\sum_{i=1}^{G} \mid g_i \mid} \tag{11}$$

where $s_j \bigcap g_i$ is the overlapping between a superpixel/supervoxel s_j and a ground truth segment g_i. r is set to be 5% as in [13]. In general superpixels/supervoxels that tightly fit the ground truth segments result in a lower value of UE.

3) Segmentation Accuracy: This metric measures the fraction of a ground truth segment that is correctly classified by the superpixels/supervoxels, and we report the average fraction over all the ground truth segments. It is formulated as

$$SA = \frac{1}{G}\sum_{i=1}^{G} \frac{\sum_{s_j:s_j\cap g_i>c} \mid s_j \mid}{\mid g_i \mid} \tag{12}$$

where the overlapping ratio c specifies whether a ground truth segment is correctly classified or not and we set $c = 95\%$ as in [13] and [15].

In the following, we evaluate the proposed method under different parameter settings, discuss the principles of determining parameters, and compare the performance with 6 well known superpixel/supervoxel algorithms quantitatively and qualitatively. We do not include comparisons with other superpixel/supervoxel algorithms because according to the recent superpixel/supervoxel benchmark surveys [6], [13] the algorithms we have compared with have achieved the state-of-the-art performance and they have been widely used in different applications already.

We implemented the proposed method and the benchmark evaluation algorithm in C/C++. For the comparison algorithms, we used implementations published by their authors. All experiments were conducted on a Linux workstation with 8 GB memory and an Intel processor clocked at 2.4GHz with 8 cores. Average running time of evaluated superpixel/supervoxel algorithms on all image/video datasets is shown in Table I. Proposed HEWCVT method achieved comparable time efficiency among other algorithms in both superpixel and supervoxel constructions.

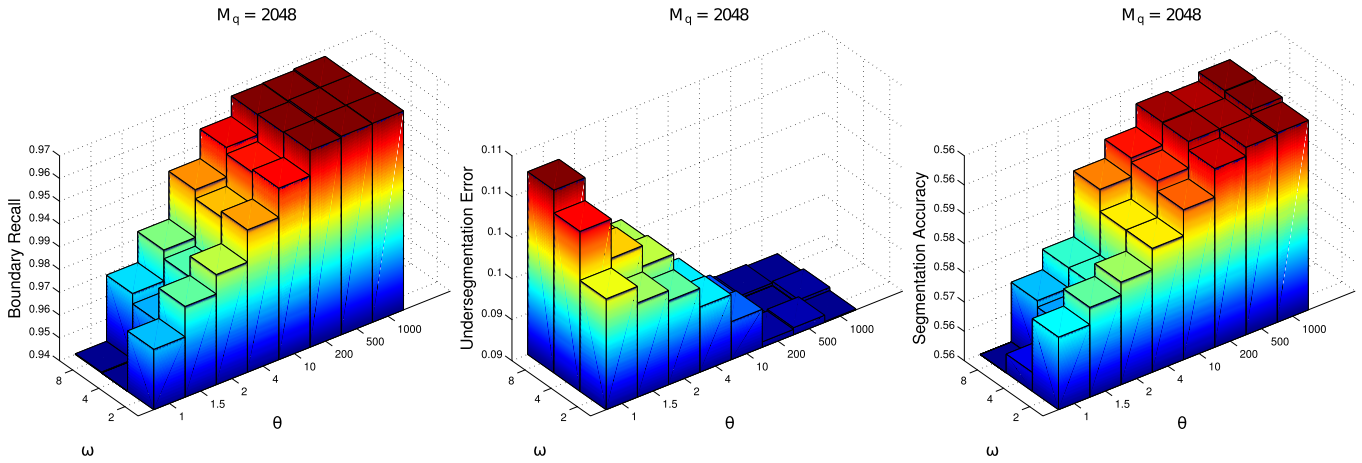


Fig. 5. Superpixel performance, in terms of boundary recall, undersegmentation error and segmentation accuracy, of the proposed HEWCVT under different values of ω and θ in one hierarchy level on the image validation set. Better view in color.

B. Parameters

There are two major parameters that can be tuned in the proposed HEWCVT method:

1) ω, which defines the radius of the 2D/3D local neighborhood region of a superpixel/supervoxel, as illustrated in Fig. 2;
2) θ, the ratio between the CVT clustering energy and the edge energy, as defined in Section III-D.

For the METIS algorithm utilized for initializing superpixel/supervoxel graphs in higher levels, we follow its default parameter settings, i.e., performing a k-way graph clustering with at most 10 iterations. We stop both Algorithm 1 and Algorithm 2 after $niter = 30$ and $niter = 200$ iterations for superpixel and supervoxel constructions respectively, since in practice we are already able to achieve converged HEWCVT energies.

To determine the values of ω and θ, we perform a grid search on validation sets, which consist of around 30% images and videos randomly selected among three datasets, and later we choose the parameter settings that achieve the highest performance across the three evaluation metrics. For the comparison algorithms, we did the same grid search on the validation sets and reported performance on testing sets using obtained parameter settings.

It's impossible to enumerate all possible values and combinations of these two major parameters, thus we only tested values within certain ranges: $\omega \in \{2, 4, 8\}$, $\theta \in \{1, 1.5, 2, 4, 10, 200, 500, 1000\}$. For the hierarchy structure, we set $M_q \in \{4096, 2048, 1024, 512, 256, 128\}$ for the superpixel construction on images and $M_q \in \{1000, 900, \ldots, 100\}$ for the supervoxel construction on videos. For each hierarchy level, we pick the results with the highest performance in the previous level as the initialization. We list the parameter settings that achieve the highest performance on the validation sets in Table II, and later we report performance of proposed method on testing sets using these fixed settings.

We also investigate the influence of different parameter values on the superpixel/supervoxel construction in the proposed HEWCVT method. Figure 5 illustrates the superpixel performance of the proposed HEWCVT under different (ω, θ) on the image validation set. From where we can see that, larger ω decreases the performance while larger θ always leads to better performance. The reason is that, compared with the whole image, the average size of constructed superpixels is quite small. Thus given a large local neighborhood region (large ω), the edge energy will dominant the total HEWCVT energy. Without balancing the energy ratio between the color energy E_{cvt} and the weighted edge energy λE_{edge} (requiring large θ), over-smoothed superpixel boundaries are not well adherent to the structural boundaries. Similar phenomenon has been observed on the video validation set as well.

TABLE II
PARAMETER SETTINGS (ω, θ) THAT ACHIEVE THE HIGHEST PERFORMANCE ON EACH HIERARCHY LEVEL IN THE GRID SEARCH ON THE VALIDATION SETS

M_q	4096	2048	1024	512	256	128
Image	$(2, 1.5)$	$(2, 1000)$	$(2, 1000)$	$(2, 1000)$	$(2, 1000)$	$(2, 1000)$

M_q	1000	...	100
Video	$(2, 1)$	$(2, 1)$	$(2, 1)$

C. Superpixel Evaluation

In order to evaluate the proposed HEWCVT method on the superpixel construction, we further compare it with other 4 state-of-the-art superpixel algorithms, including the VCells algorithm [16], a MRF model based algorithm (GraphCut) [15], the SLIC algorithm [13] and the LRW algorithm [14]. We do not include other superpixel algorithms such as the Turbopixel algorithm [5], and other segmentation based methods such as the NormalizedCut algorithm [2], Meanshift [27] and Quickshift [28] algorithms, into our comparison because: according to [13] SLIC outperforms many state-of-the-art superpixel and segmentation algorithms on the Berkeley dataset and we have included SLIC into our comparison.

We apply these superpixel algorithms on the two testing datasets: the BSDS300 dataset and the Weizmann dataset (including two parts: W1 and W2), and discuss both quantitative and qualitative results. Other than the standard metrics,

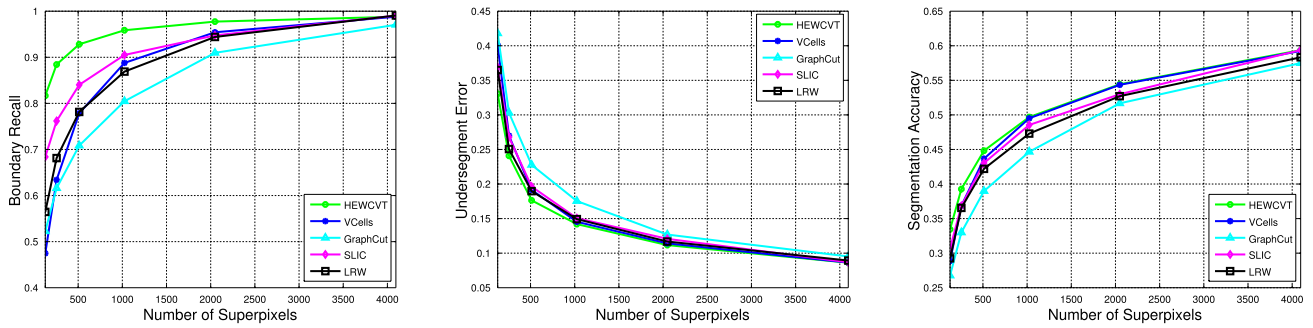


Fig. 6. Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the BSDS300 dataset.

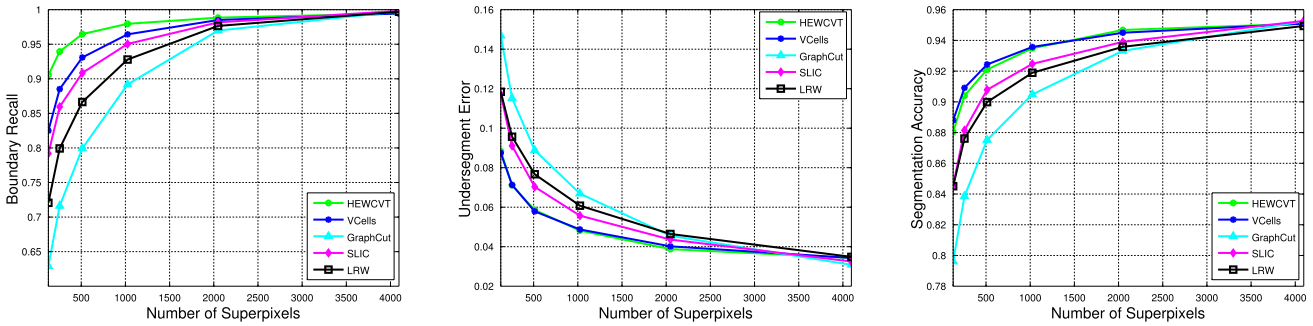


Fig. 7. Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the W1 dataset.

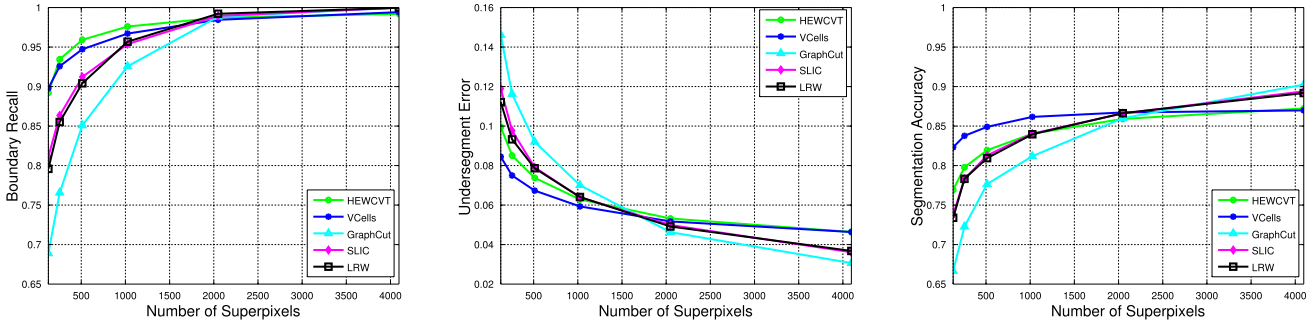


Fig. 8. Superpixel evaluation of HEWCVT, VCells, GraphCut, SLIC and LRW on the W2 dataset.

we also evaluate the constructed superpixels in term of semantic image segmentation accuracy using the algorithm described in [7], where the constructed superpixels are utilized as an initialization for a CRF based pixel labeling algorithm. Similar evaluation approach has been used in [13] as well.

1) Quantitative Results: Quantitative results of superpixel construction for all the datasets are shown in Fig. 6, 7, and 8 respectively. For the BSDS300 dataset, we can see that, proposed HEWCVT clearly achieves better performance in terms of both three metrics compared with other state-of-the-art methods. For both W1 and W2 datasets, EWCVT based methods: HEWCVT and VCells, outperforms other comparison algorithms, and HEWCVT achieves comparable performance with VCells on the W1 dataset. But for the W2 dataset, when the number of superpixels is small, VCells achieves better performance than HEWCVT in terms of undersegment error and segmentation accuracy. The major reason is that, in the Weizmann dataset, only object's external contours have been annotated as the ground-truth, as shown in Fig. 4, thus it favors superpixels constructed directly on a coarse scale without considering object's internal structures in finer scales. The proposed HEWCVT, however, achieves coarse scale superpixels using superpixels in finer scales, which leads to uneven external contours and lower performance than VCells that produces superpixels directly on the coarse scales.

2) Qualitative Results: Sample results of constructed superpixels from all the datasets are shown in Fig. 9 and 10. We can see that, compared with the four comparison methods, HEWCVT produces more uniform superpixels in the finest scale while catches structural boundaries more accurately in the coarsest scale.

3) Application to Semantic Image Segmentation: Different from previous boundary based segmentation datasets, the MSRC image dataset [29], which consists of 591 color images of size approximately 320×213, provides each pixel a semantic object class label. Sample images with ground truth

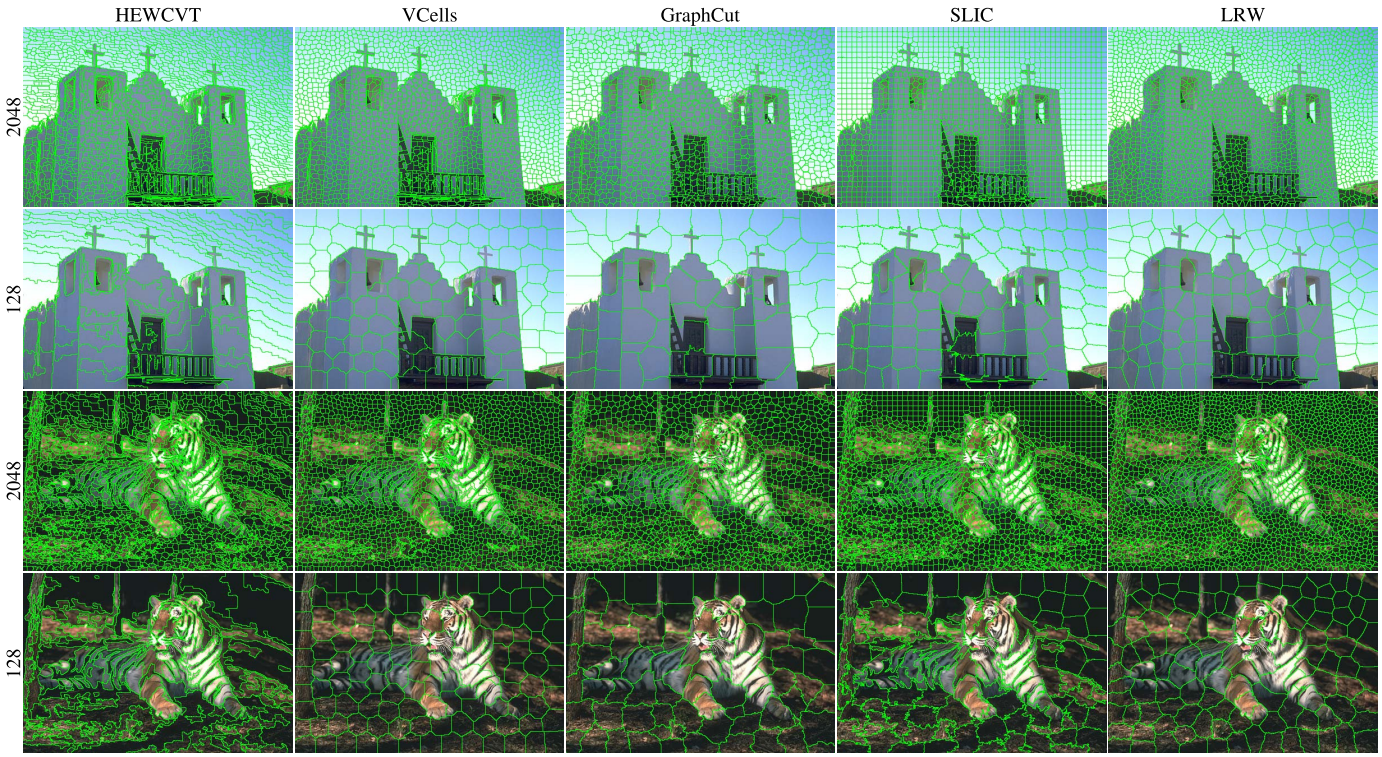


Fig. 9. Qualitative comparisons of the four superpixel methods (HEWCVT, VCells, GraphCut, SLIC, LRW) on two images from the BSDS300 dataset. The numbers at the left indicate the desired number of superpixels. Better view in color.

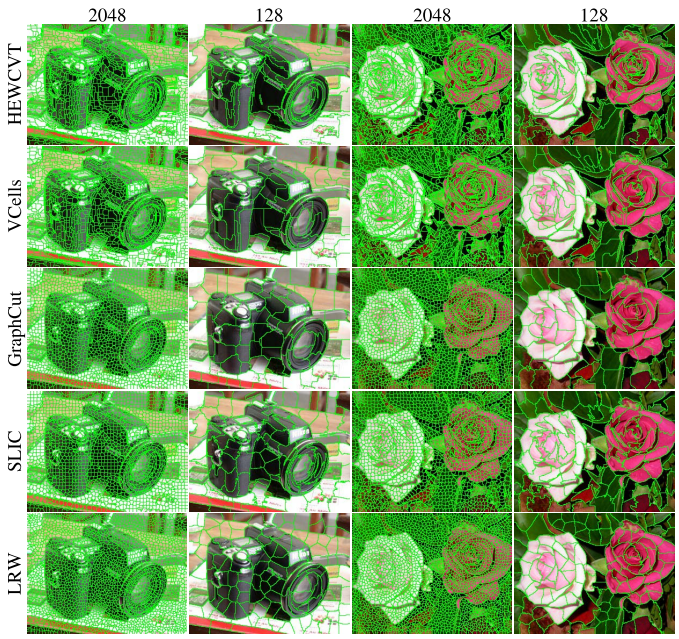


Fig. 10. Qualitative comparisons of the four superpixel methods (HEWCVT, VCells, GraphCut, SLIC, LRW) on two images from the W1 and W2 datasets. The numbers at the top indicate the desired number of superpixels. Better view in color.

Fig. 11. Sample images and their semantic pixel labels from the MSRC dataset.

TABLE III
CLASS-AVERAGE SEGMENTATION ACCURACY ON THE MSRC DATASET USING SUPERPIXELS CONSTRUCTED BY LISTED ALGORITHMS

	HEWCVT	VCells[16]	GraphCut[15]	SLIC[13]	LRW[14]
Accuracy	76.2%	75.4%	73.2%	76.9%	74.6%

semantic labels are shown in Fig. 11. In order to evaluate the performance of the obtained superpixels, we use the method described in [7] where superpixels are used as input for a CRF based semantic image segmentation approach, and report the final class-averaged segmentation accuracy in Table III. We can see that the proposed HEWCVT achieves comparable class-average segmentation accuracy as the state-of-the-art method SLIC, and outperforms other four superpixel methods.

D. Supervoxel Evaluation

Similar to the superpixel evaluation, we compare the supervoxel construction performance of the proposed HEWCVT against two state-of-the-art supervoxel algorithms: the graph

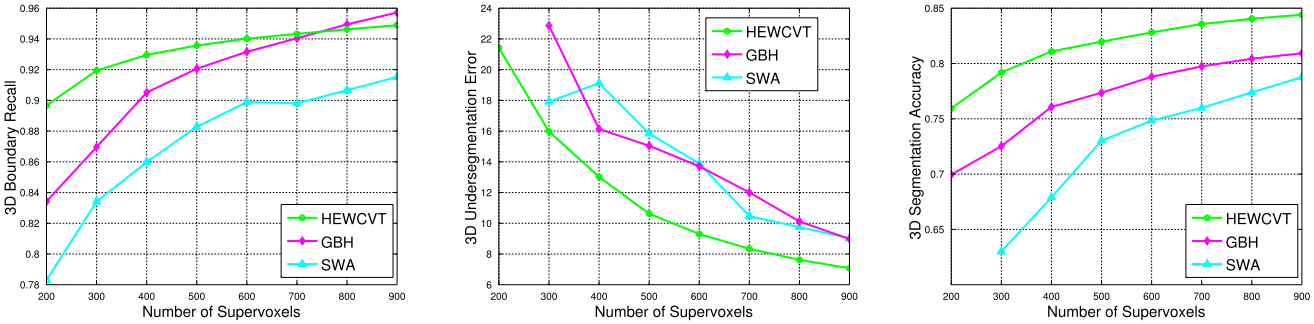


Fig. 12. Supervoxel evaluation (w/ connectivity enforcement) of GBH, SWA, and HEWCVT on the Xiph.org dataset

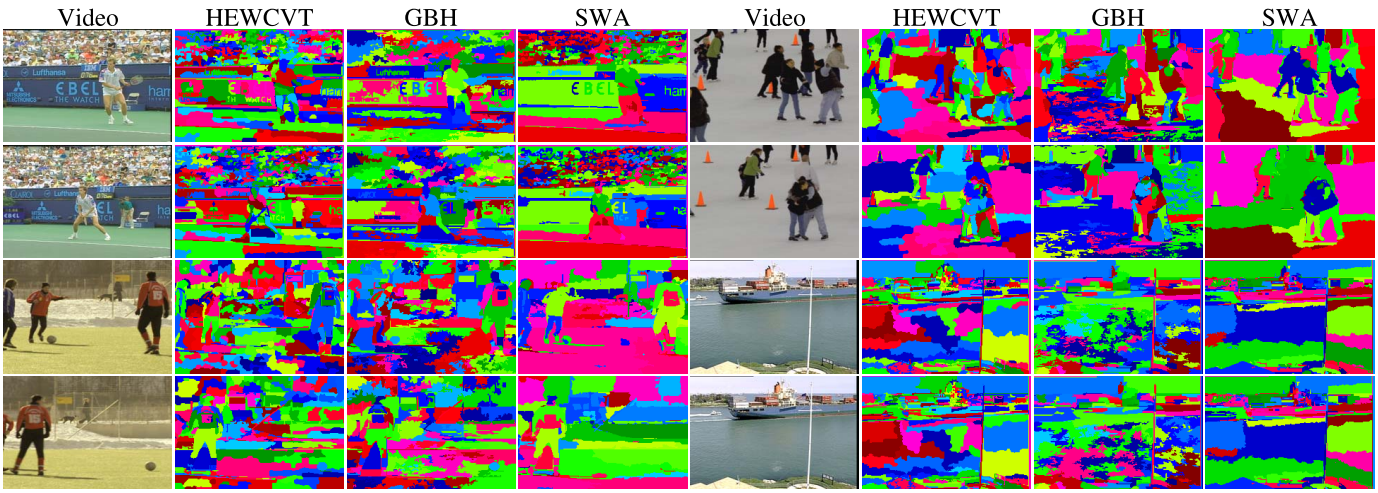


Fig. 13. Qualitative comparisons of the three supervoxel methods (HEWCVT, GBH, SWA) on four videos. On each video, the number of supervoxels generated by these three methods are similar for fairer comparison. Neighboring supervoxels are shown in different color.

based hierarchical algorithm (GBH) and the weighted aggregation algorithm (SWA). All three algorithms consider a video as an entire 3D volume. We quantitatively and qualitatively evaluate these supervoxel algorithms on the Xiph.org video dataset. However, two comparison algorithms, GBH and SWA, do not enforce the connectivity of each supervoxel, which results in supervoxel fragments in the 3D space. For a fairer comparison, we apply the same connectivity enforcement ($\epsilon = 15$) to remove such fragments and then count each connected component as a separate supervoxel in evaluating GBH and SWA in this paper. Later we will still present the evaluation results without applying the connectivity enforcement and discuss the supervoxel fragment problem.

1) Quantitative Results: Quantitative results on the Xiph.org video dataset are shown in Fig. 12. In terms of 3D boundary recall, proposed HEWCVT achieves comparable performance to GBH and better performance than SWA. When the number of supervoxels is very large, supervoxels generated by GBH become highly scattered with a large number of disconnected supervoxel fragments. Therefore GBH achieves better boundary recall. However, highly scattered supervoxels lead to lower accuracy in catching structural boundaries, which is measured by other two metrics. For the other two metrics, 3D undersegmentation error and 3D segmentation accuracy, HEWCVT clearly performs better than both GBH and SWA.

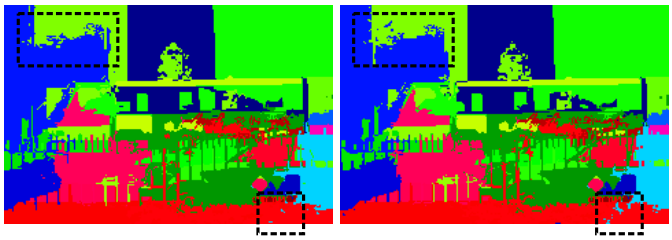


Fig. 14. An illustration of the dis-connectivity issue in GBH. Constructed supervoxels in two adjacent video frames are visualized with specific colors. Each supervoxel from GBH actually contains many disjoint fragments. Highlighted by black bounding boxes. Better view in color.

2) Qualitative Results: Qualitative results of constructed supervoxels from different methods are illustrated in Fig. 13. We can see that, with a similar number of supervoxels, proposed HEWCVT can produce more uniform supervoxels to catch the structural boundaries, but without generating many small fragments, when compared with GBH and SWA.

3) Discuss on Connectivity Enforcement: Unlike the proposed HEWCVT method and previous superpixel/supervoxel algorithms, recent supervoxel algorithms, GBH and SWA, do not enforce the simple-connectivity among constructed supervoxels, which leads to many disjoint fragments. An example is shown in Fig. 14, where GBH generates 35 supervoxels on a video and these 35 supervoxels actually consist of

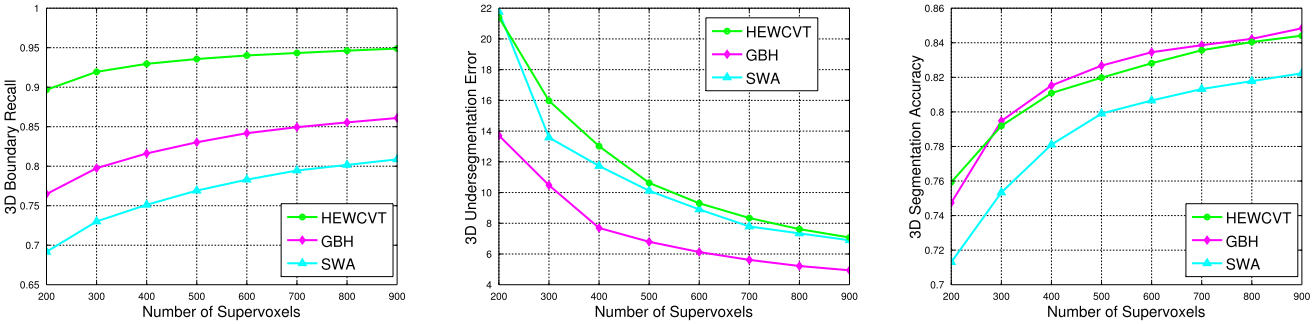


Fig. 15. Supervoxel evaluation (w/o connectivity enforcement) of GBH, SWA, and HEWCVT on the Xiph.org dataset.

15226 connected components. Given that the three evaluation metrics are dependent on the number of supervoxels, it is clearly unfair and inaccurate to count only 35 supervoxels when evaluating the results in Fig. 14. Therefore, in the previous evaluation, for GBH and SWA, we apply the same connectivity enforcement ($\epsilon = 15$) as for the proposed HEWCVT to merge such fragments, thus the performance curves of GBH and SWA reported in Fig. 12 are different from those reported in [6]. Specifically, as described in Section III-C, given small and/or isolated supervoxels constructed by GBH and SWA, we merge its voxels into neighboring supervoxels based on coordinate distances between voxel and supervoxel centers.

Here in Fig. 15, we also report the evaluation results without applying the simple-connectivity enforcement for both three supervoxel algorithms, which is the same setting as in [6] and [30]. By comparing them with the results illustrated previously in Fig. 12 (with connectivity enforcement), we can see that, after applying the connectivity filtering:

- The performance of the proposed HEWCVT does not change too much, which indicates that the proposed multiscale supervoxel clustering process already preserves very well the simple-connectivity property of constructed supervoxels. However the performance of other two methods varies a lot, which is caused by merging fragmenting supervoxels into their neighbors.
- Since the voxel based connectivity enforcement produces more boundaries, the 3D boundary recall of GBH and SWA increases after the filtering. However, after merging isolated supervoxels, the area of supervoxels that leak across the boundary of the ground-truth segments increases, and the number of supervoxels that have large portion overlapping with ground-truth segments decreases, thus the undersegment error increases and the segment accuracy decreases.

Based on above observations, we can conclude that the proposed HEWCVT supervoxel method is able to achieve better performance than GBH and SWA, and meanwhile it can also preserve the simple-connectivity property as much as possible, which is important for 3D image segmentation.

V. Conclusions

In this paper, we have proposed a hierarchical edge-weighted centroidal Voronoi tessellation method for generating multiscale superpixels/supervoxels. In the finest scale, superpixels/supervoxels are constructed directly from pixels/voxels. In the higher scales, larger size superpixels/supervoxels are obtained by clustering superpixels/supervoxels in the lower levels. The clustering energy involves both the color feature similarity and the boundary smoothness of superpixels/supervoxels. The obtained structural boundaries are consistent among superpixels/supervoxels in different scales. We have investigated the performance of the proposed method under different parameter settings, and discussed the principles of determining parameters. Quantitative and qualitative results from various experiments show that the HEWCVT method can achieve superior or comparable performances over several current state-of-the-art algorithms. In the future, we will further investigate utilization of motion based features in the clustering energy function for supervoxel construction and also consider extending the proposed method to handle streaming videos.

References

[1] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2141–2148.
[2] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 10–17.
[3] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 789–796.
[4] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 268–281.
[5] 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.
[6] C. Xu and J. J. Corso, "Evaluation of super-voxel methods for early video processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1202–1209.
[7] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vis.*, vol. 80, no. 3, pp. 300–316, 2008.
[8] P. Kohli, L. Ladický, and P. H. S. Torr, "Robust higher order potentials for enforcing label consistency," *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 302–324, 2009.
[9] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object detection for multi-class segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3113–3120.
[10] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1323–1330.
[11] X. He, R. S. Zemel, and D. Ray, "Learning and incorporating top-down cues in image segmentation," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 338–351.

[12] S. J. Dickinson, A. Levinshtein, and C. Sminchisescu, "Perceptual grouping using superpixels," in *Pattern Recognition*, vol. 7329. Berlin, Germany: Springer-Verlag, 2012, pp. 13–22.
[13] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "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.
[14] 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.
[15] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 211–224.
[16] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1241–1247, Jun. 2012.
[17] J. Wang, L. Ju, and X. Wang, "An edge-weighted centroidal Voronoi tessellation model for image segmentation," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1844–1858, Aug. 2009.
[18] J. Wang, L. Ju, and X. Wang, "Image segmentation using local variation and edge-weighted centroidal Voronoi tessellations," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3242–3256, Nov. 2011.
[19] J. Liu, X.-C. Tai, H. Huang, and Z. Huan, "A fast segmentation method based on constraint optimization and its applications: Intensity inhomogeneity and texture segmentation," *Pattern Recognit.*, vol. 44, no. 9, pp. 2093–2108, 2011.
[20] E. Sharon, A. Brandt, and R. Basri, "Fast multiscale image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 70–77.
[21] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999.
[22] Q. Du, M. Gunzburger, and L. Ju, "Advances in studies and applications of centroidal Voronoi tessellations," *Numer. Math., Theory, Methods Appl.*, vol. 3, no. 2, pp. 119–142, 2010.
[23] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.
[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jul. 2001, pp. 416–423.
[25] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[26] A. Y. C. Chen and J. J. Corso, "Propagating multi-class pixel labels throughout video frames," in *Proc. Western New York Image Process. Workshop*, 2010, pp. 14–17.
[27] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[28] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 705–718.
[29] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, 2009.
[30] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 626–639.

Youjie Zhou received the B.S. degree in software engineering from East China Normal University (ECNU), Shanghai, China, in 2010. He is currently pursuing the Ph.D. candidate in computer science and engineering with the University of South Carolina as a Research Assistant with the Computer Vision Laboratory. From 2007 to 2010, he was a Research Assistant with the Institute of Massive Computting, ECNU, where he worked on multimedia news exploration and retrieval. His main research interests include computer vision, machine learning, and large-scale multimedia analysis.

Lili Ju received the B.S. degree in mathematics from Wuhan University, China, in 1995, the M.S. degree in computational mathematics from the Chinese Academy of Sciences, in 1998, and the Ph.D. degree in applied mathematics from Iowa State University, in 2002. From 2002 to 2004, he was an Industrial Post-Doctoral Researcher with the Institute of Mathematics and Its Applications, University of Minnesota. He joined the University of South Carolina in 2004, where he is currently a Professor of Mathematics. He has been a Guest Professor in the Beijing Computational Science Research Center since 2012. He has served as an Associate Editor of the *SIAM Journal on Numerical Analysis* since 2012. His research interests include numerical analysis, image processing, ice sheet modeling and simulation, mesh generation and optimization, nonlocal modeling, and parallel computing.

Song Wang received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, USA, in 2002. From 1998 to 2002, he was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, medical image processing, and machine learning. He serves as the Publicity/Web Portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence and the IEEE Computer Society, and an Associate Editor of *Pattern Recognition Letters*. He is a member of the IEEE Computer Society.