Regularity Preserved Superpixels and Supervoxels

Huazhu Fu, Xiaochun Cao, Dai Tang, Yahong Han, and Dong Xu, *Senior Member, IEEE*

***Abstract*—Most existing superpixel algorithms ignore the spatial structure and regularity properties, which result in undesirable sizes and location relationships for the subsequent processing. In this paper, we introduce a new method to generate the regularity preserved superpixels. Starting from the lattice seeds, our method relocates them to the pixel with locally maximal edge magnitudes and treats them as the superpixel junctions. Then, the shortest path algorithm is employed to find the local optimal boundary connecting each adjacent junction pair. Thanks to the local constraints, our method obtains homogeneous superpixels with adjacency in lowly textured and uniform regions and simultaneously preserves the boundary adherence in the high contrast contents. Our method preserves the regularity property without significantly sacrificing the segmentation accuracy. Moreover, we extend this regular constraint for generating the supervoxels. Our method obtains the regular supervoxels, which preserves the structural relation on both spatial and temporal spaces of the video. Quantitative and qualitative experimental results on benchmark datasets demonstrate that our simple but effective method outperforms the existing regular superpixel methods.**

***Index Terms*—Over-segmentation, spatial structure, superpixels, supervoxels.**

I. Introduction

SUPERPIXEL is originally from the over-segmentation. Many existing superpixel algorithms are widely employed due to the following advantages. First, superpixels can appropriately preserve the boundary of different objects in the image by keeping the region homogeneity. Second, subsequent processes on the superpixels are more efficient than those on individual pixels. Therefore, superpixel generation has become a popular preprocessing technique for many multimedia applications, such as saliency detection [1]–[3], object recognition [4], [5], and image segmentation [6], [7].

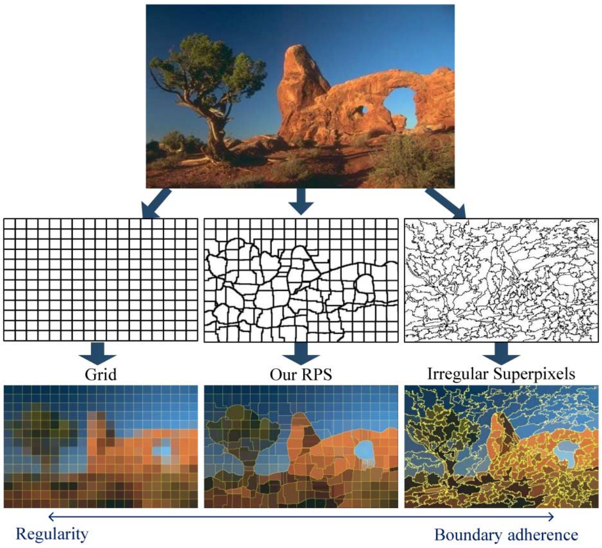


Fig. 1. Grid sampling maintains the original spatial structure of the pixel, but it lacks of the boundary accuracy. Existing superpixel algorithms [8], [9] perform well for clustering the pixels with similar appearances, but they ignore the spatial relation in adjacent superpixels. Our method preserves the regularity as well as the boundaries. From the top to the bottom: the input image, the superpixel boundary maps, and the images with the superpixels shown with the mean color of pixels.

Traditional superpixel algorithms group pixels under irregular sizes and positions [13]–[16]. Those algorithms perform well for clustering pixels with similar appearances, but they neglect the uniformity of the atomic regions and the spatial structure. For example, the work in [8] proposes a graph-based approach to generate the superpixels. This method preserves image boundaries well but can not explicitly represent the structural relationships (see Fig. 2(a)). Moreover, this irregular superpixel method generates a lot of scattered tiny superpixels. The method called ERS [9] employs the entropy rate of a random walk to obtain the superpixels with high boundary precision. ERS favors the superpixels formation with homogeneity. However, the spatial structures of neighboring superpixels are not considered in [9], as shown in Fig. 2(b).

In this paper, we revisit the notion of regular superpixels and summarize the following two observations. First, the superpixels represent the information of the pixels, preserving appearance coherence of the generated region for the given resolution. Second, the typical properties of the pixels, such as spatial topology, structure, homogeneity, and isometric information, should be preserved by the superpixels as much as possible. To this end, we propose a simple yet effective method,

Manuscript received December 12, 2012; revised June 17, 2013 and October 12, 2013; accepted December 16, 2013. Date of publication February 10, 2014; date of current version May 13, 2014. This work was supported by NSFC (No. 61332012; 61003200), National Basic Research Program of China (2013CB329305), 100 Talents Programme of The Chinese Academy of Sciences, and Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030601). The work of Y. Han was supported in part by the NSFC (under Grant 61202166). This paper is an extended version of the original paper which appeared in the Proceedings of IEEE ICME 2012 Conference and was among the top-rated 4% of ICME'12 submissions. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sen-Ching Cheung.

H. Fu is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China, and also with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: hzfu@ntu.edu.sg).

X. Cao is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: caoxiaochun@iie.ac.cn).

D. Tang is with the School of Computer Software, Tianjin University, Tianjin 300072, China (e-mail: debelletang@gmail.com).

Y. Han is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: yahong@tju.edu.cn).

D. Xu is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: dongxu@ntu.edu.sg).

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/TMM.2014.2305571

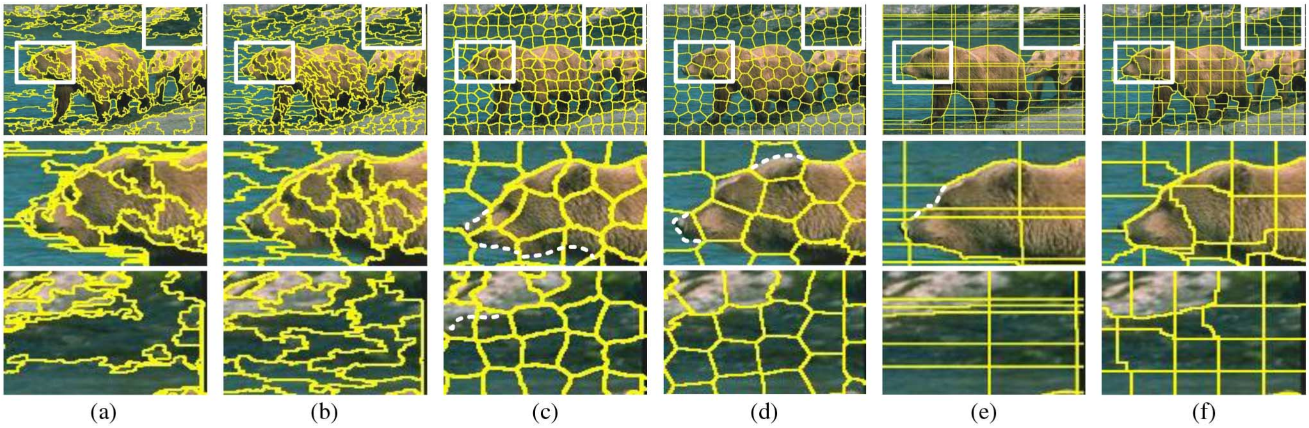


Fig. 2. Superpixels obtained from six methods: (a) GraphBased [8], (b) ERS [9], (c) TurboPixel [10], (d) SLIC [11], (e) Lattice [12], and (f) Our RPS. The superpixel resolution is $N = 200$. The zoom-in results for the two white boxes in the first row are shown in the last two rows. Our RPS method achieves a good balance between the regularity and the segmentation accuracy.

called regularity preserved superpixels (RPS). Intuitively, the superpixels in our RPS appropriately express the contours of physical objects. In particular, the proposed local optimal criterion ensures that, in the absence of boundary cues (e.g. the lowly textured regions), superpixels maintain the homogeneous shape. Fig. 1 gives an illustration between the regular and irregular superpixels, where our RPS method provides a good balance by preserving spatial structure while following object boundaries. Last but not least, our approach can also be generalized for the supervoxel segmentation for videos, while preserving the structural relation on both spatial and temporal spaces.

A. Related Works

There are some approaches to generate the regular superpixels. However, most of them do not maintain the spatial structure. For example, TurboPixel [10] generates the regular superpixels by evolving the geometric flow from the seeds placed uniformly in the image. However, TurboPixel can not provide the explicit spatial relations of superpixels. Moreover, TurboPixel exhibits relatively poor boundary adherence. This is highlighted by the dashed white lines in Fig. 2(d). SLIC [11] employs k-means clustering to generate superpixels. However, for the low superpixel resolution, the regular constraint can be harmful for the boundary adherence leading to the low segmentation accuracy.

The method called Lattice [12], [17] provides an alternative solution for obtaining the regular superpixels by constructing the vertical and horizontal superpixel boundaries. The greedy algorithm [12] or global GraphCut optimization technique [17] seeks the optimal paths within the predefined strips, and the future path costs are modified around the previous paths, in order to avoid multiple crossings. This constraint can easily lead to under-segmentation on the highly textured regions. In the second row of Fig. 2(e), Lattice under-segments the head of the bear, as highlighted in the dashed white lines. Furthermore, this constraint also generates the irregular superpixels in lowly textured regions, such as in the last row of Fig. 2(e), where the superpixels near the river bank are highly irregular. These irregular superpixels produce many narrow and empty regions.

In contrast to the above methods, we utilize the superpixel junctions as the geometric constraint to produce regular superpixels with spatial structure. The boundaries between superpixels are generated via the shortest path algorithm in the regional graph, which preserves the boundary adherence of the image. Fig. 2(f) shows that our superpixels are more uniform in the size and with more regular shape. Another important advantage of our method is that it maintains topological relation between the adjacent patches.

In the preliminary conference version of this work, our method is only employed to obtain the regular superpixels on the 2D image [18]. In this paper, we extend our method into the 3D video to obtain the regularity preserved supervoxels. Similar to the superpixels, the supervoxels are a set of spatio-temporal contiguous voxels in the 3D space with similar appearances. In contrast to 2D images, there is a temporal dimension in the 3D videos. Most existing supervoxel methods [19], [20] or spatio-temporal segmentation methods [21]–[23] focus on the long trajectory clustering. The basic property of spatio-temporal uniformity, which encourages the compact superpixels with regular shape in the space-time domain [20], is ignored in these supervoxel methods. For some cases, the compact supervoxels perform better than those with varying sizes for the high-level tasks, such as salient object segmentation [15]. Furthermore, most methods directly group the superpixels using the similarity appearances [11], [13], [21]. The work in [19] forms a superpixel affinity matrix using the spatio-temporal edges in the superpixel graph, and optimally cuts this graph using the parametric maxflow method. The work in [24] uses a graph-based hierarchical segmentation approach to segment the video. These methods partly lose the topological inter-frame connection, especially the neighboring relationship of supervoxels in the temporal space. To address these problems, we extend our regularity idea to the videos. Our generated supervoxels appear as the cube-like 3D segmentation, preserving the structural relation on both spatial and temporal spaces of the video.

B. Regularity and Spatial Structure

In the past several years, many regular superpixel methods have been proposed [10]–[12], [17]. However, how to use the structural regularity to benefit the vision applications is not very clear. In this section, we discuss the potential advantages of the

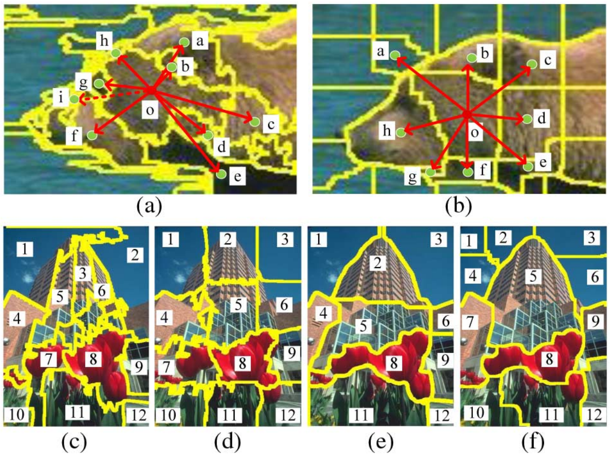


Fig. 3. Top: (a) The irregular superpixels [8]. (b) Our RPS method preserves the distance and spatial structure. Bottom: Superpixel results ($N = 12$) with the labeled grid index: (c) ERS [9], (d) SLIC [11], (e) Lattice [12], and (f) Our RPS. The index is labeled for each superpixel. Our RPS method achieves the proper tradeoff between the regularity and the segmentation accuracy of superpixels.

regularity and spatial structure properties. Although we do not address these specific advantages in this paper, it is worth to mentioning them. The spatial relation of each superpixel provides the geometric correspondence, which allows us to learn the high-order structural relation of the image.

Regular distance. When the superpixels are employed as the atomic regions, the distance of two superpixels is often defined as the distance between their centers [6], [25], [26]. Fig. 3(a) illustrates the irregular superpixels, where the red point denotes the target superpixel and the green points denote the surrounding superpixels. For example, the superpixels with the index numbers b and e are the neighbors of the target superpixel o, and the distance $|ob|$ is smaller than $|oe|$. The superpixel i is not the neighbor of o, but the distance $|oi|$ is smaller than $|oe|$. It means that the distance prefers the superpixels with smaller sizes, and the neighboring relationship does not depend on the distance for the irregular superpixel. In contrast, the regular superpixels have the uniform or similar sizes, which provide the meaningful distances corresponding the spatial relationship, as shown in Fig. 3(b).

Spatial structure. The structural relation is an important spatial prior for many applications [27]–[30]. The common spatial structure are the 4/8-connected neighboring relation [22], [31], [32] and the up/down context relation [33], [34]. The spatial structure of irregular superpixels is often invalid, such as in Fig. 3(a). The superpixels have no clear vertical and horizontal relationship. In [35], the authors introduce the dummy nodes to reconstruct the superpixel spatial grid from the regular or near-regular superpixels. And then, they show that the grid-like structure is useful for rapid object localization. However, our RPS method can directly define the spatial relation, as shown in Fig. 3(b), which can be used to obtain more semantically reliable grid-like superpixel.

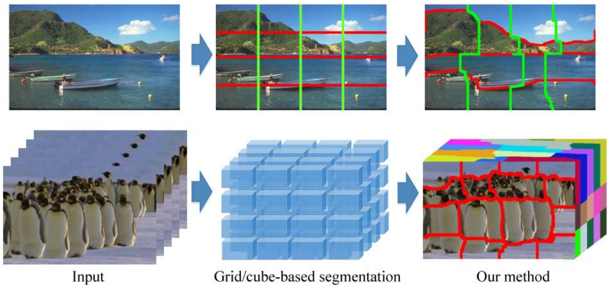


Fig. 4. Spatial structure provides the global geometric correspondence, which is useful in the grid-based representation, such as spatial pyramid matching [36]. Our superpixels preserve the spatial structure and also achieve accurate segmentation of the images, when compared with the grid-based representation. Moreover, our supervoxels also inherit the structural property of superpixels, leads to the regularity preserved segmentation results in the 3D spatio-temporal domain [37].

Topological correspondence. One advantage of the grid-based sampling is to preserve the global structural relation of the image, especially for the low superpixel resolution. This topological structure provides the global geometric correspondence of the image, which is demonstrated to be useful for significantly improving the performance on the state-of-the-art categorization method, such as spatial pyramid matching [36], and shape model representation [38], as shown in Fig. 4. Most irregular superpixels can not preserve the grid structure without any structural constraint (see Fig. 3(c)). Some near-regular superpixel methods (e.g. TurboPixel [10], SLIC [11]) start from a set of uniformly selected seeds, but the final results often lead to poor segmentations for the low superpixel resolution as shown in Fig. 3(d). Lattice [12] produces the exact grid structure regions under the structural constraints. However, its greedy strategy for finding the optimal paths may easily produce many fragments. Fig. 3(e) shows the results of Lattice [12] with the resolution $N = 12$, where the 7th and 10th superpixels are empty regions. In contrast, our RPS method generates the regular superpixels based on the grid junction constraint, which seeks for the proper tradeoff between the topological structure and the segmentation accuracy. Moreover, the topological correspondence also contributes to represent the relation of superpixels in the video, and obtain the cube-like supervoxels, which is helpful for other related tasks like 3D spatio-temporal pyramid matching [37], video action descriptor [39]–[41], and 3D medical applications [42].

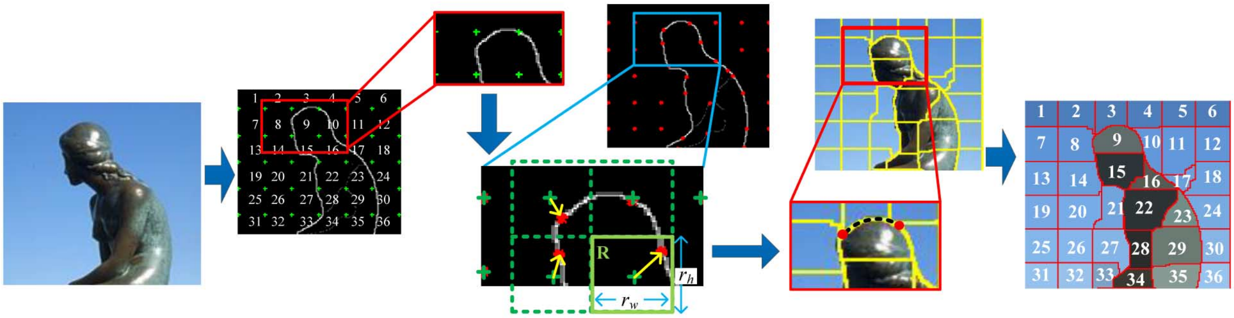
II. Regularity Preserved Superpixels

Fig. 5 illuminates the framework of our RPS method, which includes three main steps. First, we arrange initial junctions based on a lattice grid. Then, we relocate each initial junction to the pixel based on the locally maximal edge magnitude constraints, which includes both distance term and magnitude term. Third, we generate the local optimal path between the pairs of the vertical and horizontal neighboring junctions, and output the superpixel. We now describe each step below.

A. Junction Initialization

For an input image, we need to obtain the edge magnitude map. In our method, we can use any boundary map. But we prefer to employ the gPb map [43], because it obtains the higher values on the object boundary than the background region. Moreover, for the boundary map, the value in the range of $(0, 1]$ is more suitable to our subsequent processes.

By default, the original junctions are sampled on a regular grid. Here the density of the junctions depends on the superpixel resolution N, which is an input parameter for most image



(a) (b) (c) (d) (e)

Fig. 5. The framework of our RPS method includes three steps. (b) Uniformly place initial junctions (green points) based on the regular grid on the edge magnitude map. The numbers are denoted the index of the grid structure. (c) Relocate each initial junction to the appropriate boundary pixel obeying maximal edge magnitude constraints. The relocated junctions are marked as red stars. The green dashed rectangle denotes the searching region for each initial junction. (d) Generate the optimal path between each pair of relocated junctions to obtain the regular superpixels. The zoom-in view shows a pair of relocated junctions (red dots) and the optimal path (dashed black line) linking them. (e) The index of superpixels shows that our RPS method preserves the structural relationships similar to the grid structure in (b).

processing methods. The sampling intervals N_h and N_w along the horizontal and vertical directions are given by:

$$\begin{cases} W/N_w \approx H/N_h \\ N = N_w N_h \end{cases} \Rightarrow \begin{cases} N_h = \left\lfloor \sqrt{NH/W} \right\rfloor \\ N_w = \lfloor N/N_h \rfloor \end{cases}, \tag{1}$$

where H and W denote the height and the width of the input image, N denotes the superpixel resolution. Apparently, these two parameters (N_h, N_w) depend on the superpixel resolution N.

B. Junction Relocation

In the second step, the junctions are moved to the optimal positions according to maximal magnitudes of the boundary map. For each displacement, the searching region for each junction is defined as:

$$\begin{cases} r_w = \lfloor W/N_w \rfloor \\ r_h = \lfloor H/N_h \rfloor \end{cases}. \tag{2}$$

The searching region $\mathbf{R}$ is given by r_w and r_h, as shown as the green rectangle in Fig. 5(c). The searching region aims to limit potential collisions of the adjacent junction, and force the junctions to be relocated within a local searching region $\mathbf{R}$.

Then the question is how to define the optimal position for relocating each junction. We consider two constraints for regular superpixels: one is that the superpixel junctions should appear on the boundaries of the object. The second constraint is that the spatial structure of superpixels should be preserved, which encourages the junctions to be close to their initial grid positions. To address these two constraints, we define the optimally relocated position $\hat{p}$ of the initial junction p_0 in the region $\mathbf{R}$ as follows:

$$\hat{p} = \arg\max_{p_i \in \mathbf{R}} \left[f_b(p_i) \cdot f_d(p_i, p_0) \right], \tag{3}$$

where $f_b(p_i)$ is the boundary magnitude of pixel p_i, and f_d is Euclidean distance between the initial junction p_0 and the candidate position p_i. In this paper, the Gaussian kernel is used to compute f_d. This equation considers two terms related to spatial constraint and boundary magnitude. The junctions adhere to the proper boundaries appropriately. We employ the multiplication operation to integrate the two terms. The weighted sum can be also used, however, it requires an extra parameter to balance these two terms.

C. Junction Connection

Followed by the junction relocation, the third step is to generate the local optimal boundary, connecting the adjacent junctions vertically and horizontally. We formulate the superpixel boundary generation task as a shortest path problem. We generate an undirected graph $G = \langle V, E \rangle$ by using the two searched regions of the junction pair. This path generation strategy is similar to [12], [17]. However, in our method, we make full use of the edge map, and provide a simpler graph. In our graph, the nodes V denote the pixels (including the junctions), and the edge is defined as:

$$E_{ij} = \frac{1}{P_b(p_i) + P_b(p_j)}, \tag{4}$$

where $P_b(p)$ is the boundary magnitude of the pixel p, and the weighted edge E_{ij} provides relevance information. It is clear that the optimal link between junction pairs is composed by several selected nodes, which are used to identify the shortest path when generating the graph structure. Thus, the shortest path problem can be efficiently solved by using the dynamic programming method [44]–[46]. Actually, the grid-like junction layout guarantees the structural relationships of the superpixels, and the local shortest path method preserves the boundary adherence. Thanks to the constraints of the grid structure and the limited searching area, the obtained superpixels are considerably homogeneous (Fig. 5(e)).

III. Regularity Preserved Supervoxels

In this section, we extend our RPS method to generate supervoxels in the video. For supervoxels, there are two key challenges. The first is how to cope with the temporal relationship of the inter-frame superpixel, and the second is how to segment the video along the long temporal direction. As described in Section I-A, most existing supervoxel methods group the superpixels using the appearance similarity [11], [13], [21], or build a complex graph for connecting the superpixels from multiple

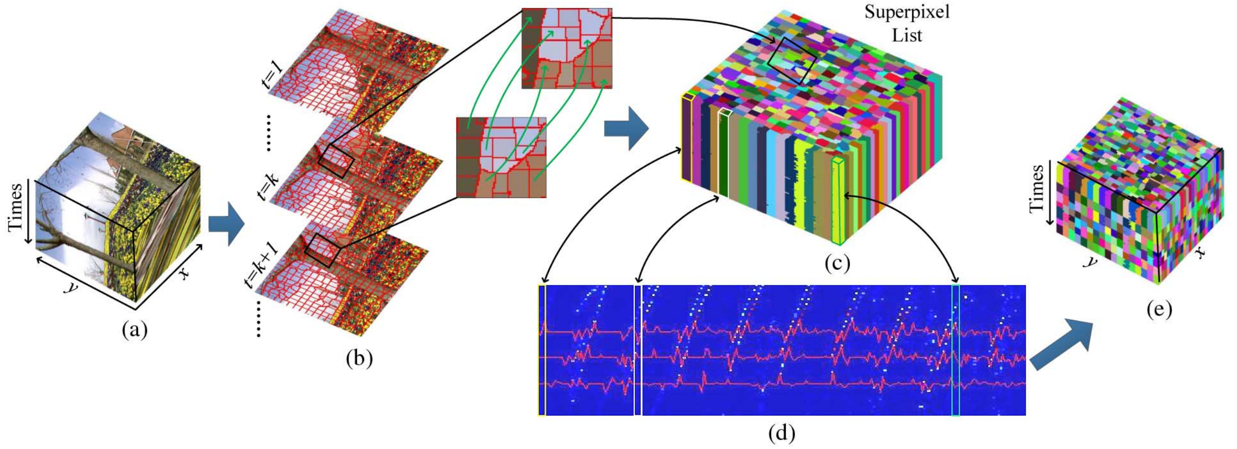


Fig. 6. The framework of our regular supervoxel method. (b) The index for each superpixel in our RPS method allows us to directly build the temporal relation of the inter-frame. (c) The superpixels with the same index are organized as a list of superpixels for the subsequent process. Then, the inter-frame distance matrix is computed between the temporally connected superpixels for each list of supervoxels. (d) After obtaining the distance matrix, the optimal cuts are computed to generate the regular supervoxels in the temporal dimension.

frames [19], [24]. These methods partly lose the structural relationship of supervoxels in the temporal space. In contrast, our supervoxel method aims to connecting the inter-frame superpixels, which has three advantages. First, it generates the homogeneous superpixels for each frame. Second, we divide the 3D video into two independent spaces: 2D image space and 1D temporal space, to avoid normalizing the spatial and temporal scales. Third, similarly as in our RPS method, it preserves the spatio-temporal structure.

Fig. 6 shows the flowchart of our supervoxel method. We first extract the regular superpixels for each video frame. Next, we construct an inter-frame distance matrix, which measures the appearance distance between any superpixel and its temporally connected superpixels. Then, the optimal cuts are decided to segment the regular supervoxels in the temporal dimension. As shown in Fig. 6(e), our regular supervoxels preserve the structural relation and regularity in both spatial and temporal spaces.

A. Inter-Frame Distance Matrix

The inter-frame distance matrix is obtained by using two steps, called temporal connection and distance measurement. The inter-frame superpixels can be directly connected via their structural indexes. In other words, the superpixels on the consecutive frames are organized together as a list of supervoxels with the same index. As shown in Fig. 6(c), the structural index is used the superpixel list index. For each supervoxel list, we compute the appearance distance between the temporally connected superpixels. We use the Euclidean distance to compute the distance based on the RGB color appearance for each pair of superpixels as follow:

$$\mathrm{A}(s^t) = \|z^t - z^{t+1}\|_2, \tag{5}$$

where s^t denotes the superpixel from the frame t, and z^t denotes the color appearance of the superpixel s^t. Fig. 7(a) shows an example of the inter-frame distance matrix, where each column and row denote the superpixel list and video frame respectively. Since the last frame of the video is naturally considered as the last cut in the temporal space.

B. Temporal Cutting

Once the inter-frame distance matrix is constructed, the optimal cuts are computed along the temporal space. As in our RPS method, we uniformly select superpixels along the temporal space as the initial cuts, as shown in Fig. 7(b). The temporal interval D is given by $D = \lfloor K/C \rfloor$, where K and C denote the number of video frames and the temporal resolution, respectively. For each list of superpixels with the same index, we refine each cut $\hat{s}_c$ in the local neighborhood along the temporal space:

$$\hat{s}_c = \mathop{\arg\max}_{s^t \in \mathbf{T}(s_c)} \left[\mathrm{A}(s^t) \cdot f_T\left(s^t, s_c\right)\right], \tag{6}$$

where s_c denotes the initial cut, s_t denotes the superpixel on the frame t, $\mathbf{T}(s_c)$ is the set of temporal neighboring frames around the initial cut within the radius $D/2$, with D being the temporal interval. $\mathrm{A}(s^t)$ denotes the appearance distance value in Eq. (5), and f_T is the distance between the initial cut s_c and the candidates s^t.

Though the temporal cut is defined within each independent superpixel list, we still employ the spatio-temporal local constraint to obtain the cube-like supervoxels. First, the regular superpixels on each frame provide the spatial structure. Then the initial uniform cuts guarantee the topological structure in the temporal spaces. Third, the cuts are limited in the local temporal spaces, so the final cuts can not change the neighboring structural relations. Moreover, we employ the similar local refinement strategy to obtain the regular supervoxels, which well preserves the topological structure in both spatial and temporal spaces of the video. Fig. 7(c) shows the final temporal segmentation of the image in Fig. 7(b), where the cuts move to the local distance maximum, while preserving the structural relation in the temporal space. Finally, the regular supervoxels are extracted, as shown in Fig. 6(e). Our regular supervoxel method is summarized in Algorithm 1.

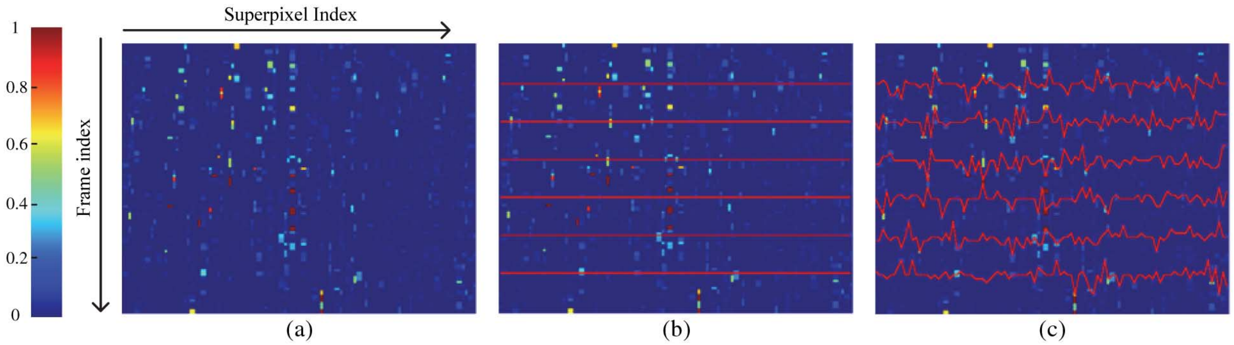


Fig. 7. (a) The inter-frame distance matrix, where each column denotes the superpixel list. (b) The initial cuts, where the red lines denote the temporal grid cuts. (c) The optimal cuts on the inter-frame distance matrix.

Algorithm 1: Regularity Preserved Supervoxels

```
Input: The input video sequences $\{I^1, I^2 \ldots I^K\}$, and supervoxel resolution $M$.
Extract regular superpixels for each frame;
Organize the superpixels with the same index as a list;
for each list of superpixels with the same index do
    for each frame do
        Compute the inter-frame distance by using Eq. (5);
    end
Uniformly Sample initial cuts;
    for each cut $s_c$do
        Optimize the cuts $\hat{s}_c$ by using Eq. (6);
    end
end
Output: The regular superpixels for each frame, and regular supervoxels for the video.
```

IV. Experiments

A. Quantitative Performance

We employ the Berkeley Segmentation Database Set (BSDS) [47], which contains 300 images with the resolution of 481×321 pixels or 321×481 pixels. We compare our RPS method with five state-of-the-art superpixel methods: GraphBased [8], Lattice [12], TurboPixel [10], entropy rate superpixel (ERS) [9], and SLIC [11], where the codes are obtained from the corresponding authors, and the parameters are set as the default values in their papers. The goal of superpixel segmentation is different with object segmentation. Therefore, the performance metrics are also different. We employ three human-independent evaluation metrics, called explained variation, compactness measure and size variation, to quantitatively evaluate our method on the benchmark dataset.

Explained Variation (EV): The explained variation in [12] is used to evaluate the overall difference between the original pixels and the superpixels as:

$$\mathrm{EV} = \frac{\sum_{k=1}^{N} (\mu_k - \mu)^2}{\sum_{i \in I} (x_i - \mu)^2}, \tag{7}$$

where x_i, μ_k and μ denote the actual pixel value in the image, the mean value of the superpixel k and the mean of all pixels in the entire image I, respectively. Explained variation measures the color distortion level caused by the superpixels. Fig. 8(a) shows the explained variation results of all superpixel methods. The explained variation scores of all the methods are expected to be higher when the superpixel resolution increases. The ERS obtains the best performance. It is not surprising that our method performs worse than ERS, because we preserve regularity property. However, our method outperforms Lattice in all resolutions, and it achieves about 5% higher EV results on the average. Our method is also comparable to some irregular superpixel methods [8], [10]. Note that the performance of our method is better than [8], [10] when using the low superpixel resolution (say less than 200 superpixels). For example, when the superpixel resolution is 50, the EV score of our RPS method is 0.66, which is higher than TurboPixel (0.57) and GraphBased method (0.58), and is even comparable with SLIC [11] and ERS [9]. Thanks to the local optimal superpixel boundary, our method can obtain reasonable segmentation results with certain semantic and topological properties, as shown in Fig. 3.

Moreover, the performance of our method using the canny edge map is better than that using gPb map when the superpixel resolution is high. One possible explanation is that for the high superpixel resolution, the canny edge detection method provides more dense boundaries than gPb map. The dense boundaries tend to have the strong edge magnitudes in the lowly textured regions. However, if the superpixel resolution is low, the gPb map offers more meaningful boundaries, so RPS_p obtains better segmentation results.

Compactness Measure (CM): The shape compactness is a common measure, which is taken from the isoperimetric inequality [48], [49]. The compactness measures the superpixel

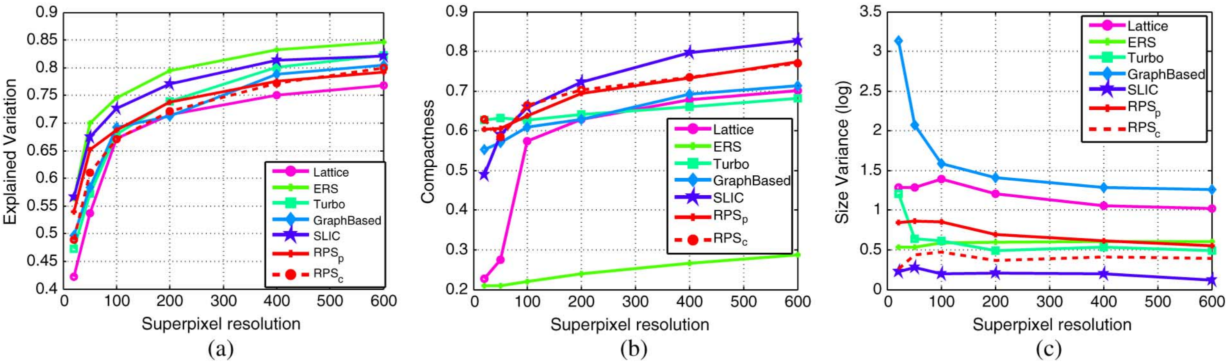


Fig. 8. The quantitative performances of our RPS method and other superpixel methods, GraphBased [8], Lattice [12], Turbo [10], SLIC [11] and ERS [9], on the BSDS dataset. For our RPS method, we report the results of RPS_c and RPS_p, when using the two edge methods Canny map and the gPb map [43]. (a) Explained Variation, (b) Compactness Measure, (c) Size Variation.

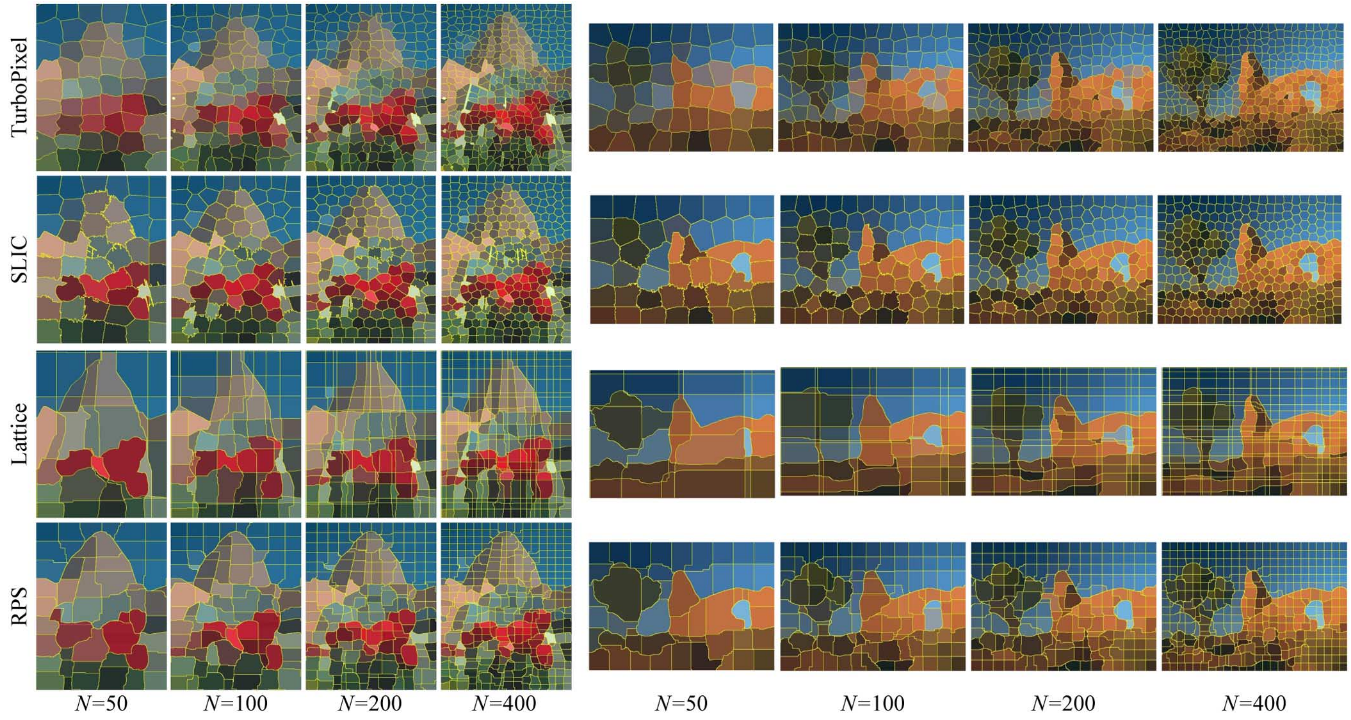


Fig. 9. Performance comparison of superpixels with the mean color of the pixels. The superpixel resolutions are $N = 50, 100, 200, 400$, respectively. Form top to bottom: TurboPixel [10], SLIC [11], Lattice [12], and our RPS method.

regularity, which can provide the regular superpixel distance. The compactness measure is defined as:

$$\mathrm{CM} = \frac{1}{N}\sum_{i=1}^{N}\frac{4\pi S(s_i)}{L^2(s_i)}, \tag{8}$$

where $S(s_i)$ and $L(s_i)$ denote the area and perimeter of the superpixel s_i. Fig. 8(b) shows the compactness measure results. Based on this measure, we have the following observations: 1) The compactness measure is dependent on the superpixel resolution, when the resolution is high, it produces smaller superpixels, which have less distortions. So if the resolution is higher, the CM scores are better. 2) SLIC obtains the best score, when the resolution is high. And our RPS method produces the square-like superpixels, so our score is lower than SLIC. However, our RPS method outperforms other methods, when the resolution is low. 3) As a cluster-based method, ERS favors the feature coherence of superpixels rather than the regular shape. It achieves the lowest CM score. 4) The lattice superpixel method does not obtain better CM score than the other methods. One possible explanation is that this method produces many narrow superpixels, as shown in Fig. 2(e), which is worse when using the compactness measure, especially when the resolution is low.

Size Variation (SV): We also use the superpixel size as another measure, which describes the size distribution of the superpixels. With the same superpixel resolution, the isometric superpixels can better preserve the spatial and topological structure. If the variance of the superpixel size is smaller, the superpixels are more regular. Fig. 8(c) shows the superpixel

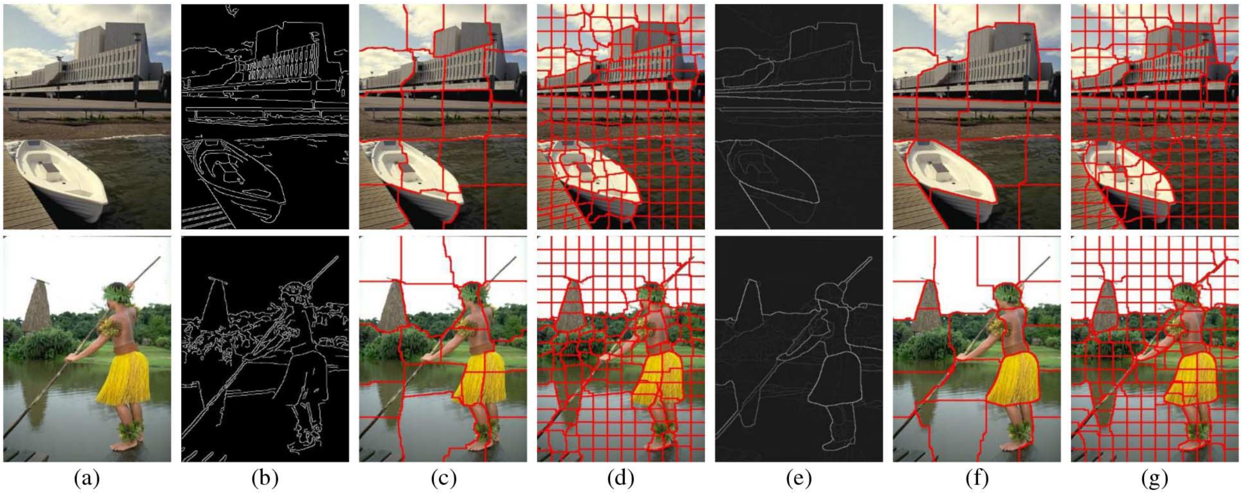


Fig. 10. Our RPS results by using different edge maps and different superpixel resolutions $N = 50, 200$. (a) The input image. (b) The canny edge map. (c-d) The superpixels using the canny edge map. (e) The gPb edge map [43]. (f-g) The superpixels using the gPb edge map.

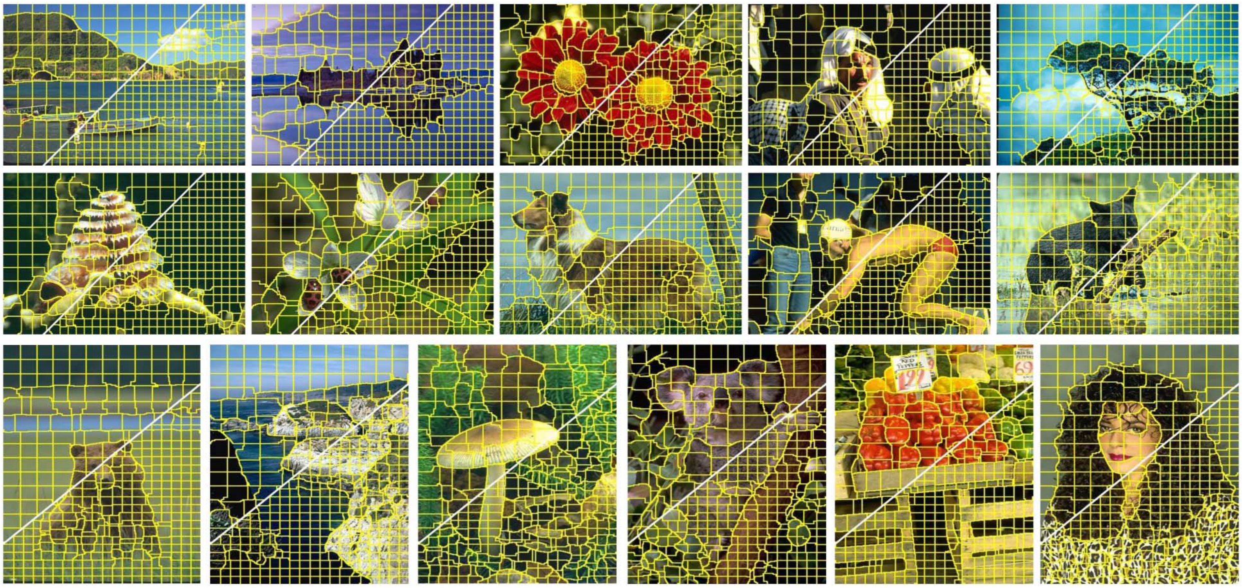


Fig. 11. Visual examples of our RPS method with the superpixel resolutions $N = 200$ (upper left of each image) and $N = 600$ (lower right of each image) on the BSDS dataset.

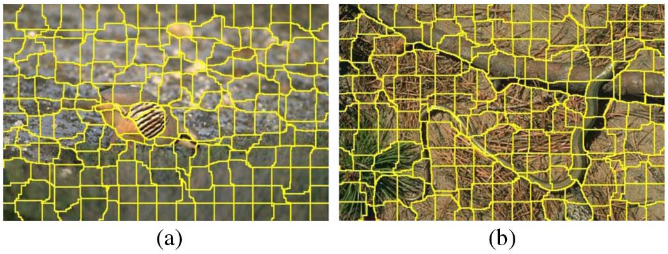


Fig. 12. Some challenging examples for our RPS method. (a) Low contrast foreground. (b) The thin object. The superpixel resolution is $N = 200$.

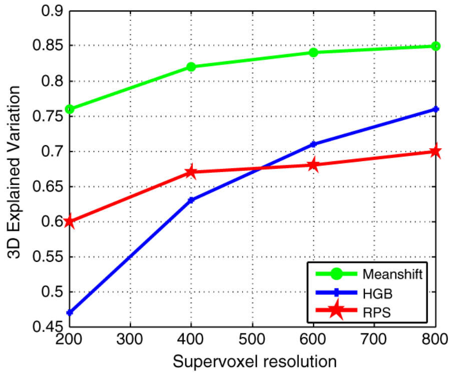


Fig. 13. 3D Explained Variation comparison of our method and other supervoxel methods, Meanshift [21] and HGB [23], on the video dataset.

size variation curves. The SLIC method obtains the best performance with the smallest SV score. One possible explanation is that the clustering process in SLIC guarantees the superpixels are with equivalent sizes. Our RPS method achieves similar results to ERS, but our method preserves the meaningful topological structure. The GraphBased and Lattice methods are worst, because they generate many fragments with various sizes.

TABLE I
AVERAGE RUNNING TIMES FOR GENERATING THE SUPERPIXELS

Method	GB [8]	ERS [9]	TurboPixel [10]	Lattice [12]	SLIC [11]	Our RPS method
Code type	C++	C++	Matlab	C++	C++	Matlab
Time (second)	0.3	1.5	9	10	0.36	0.6

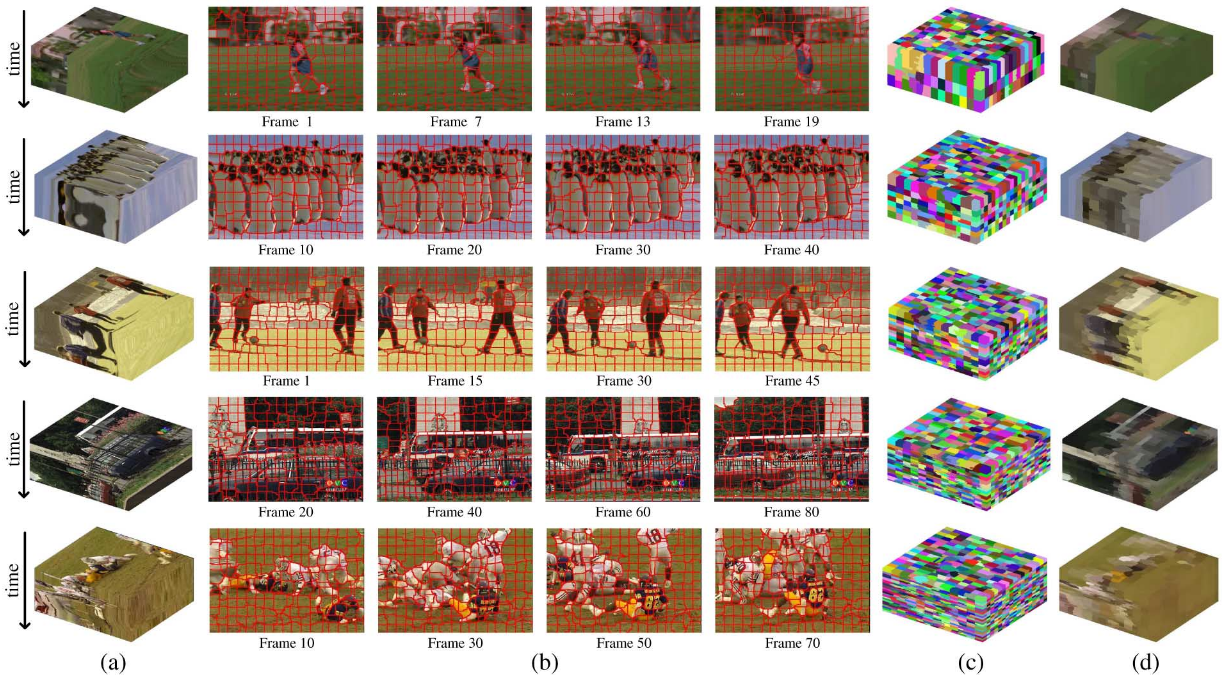


Fig. 14. Visual examples of our regular supervoxels. (a) The input videos. (b) The superpixels for each frame. (c) The supervoxels filled with the random color. (d) The supervoxels filled with the mean color of the pixels.

B. Qualitative Performance

Fig. 9 illustrates the visual example of the superpixels with various resolutions ($N = 50, 100, 200, 400$), which are represented by the mean value of the pixels. We compare our RPS method (bottom row) with three regular superpixel methods: TurboPixel [10] (first row), SLIC [11] (second row), and Lattice [12] (third row). On one hand, for any method, more details can be better extracted, when the resolutions increase. On the other hand, our method achieves the best results when the resolutions are lower, because the principal object can be well segmented. For example, when the resolution is $N = 50$, TurboPixel and SLIC can not well represent the shape of the building in the left image. And the Lattice method misses the boundary of the tree in the right image. In contrast, our RPS method clearly preserves the shape of these objects. Our method is more suitable when the superpixel resolution is low.

Fig. 10 shows the results of our RPS method when using different edge maps. When the superpixel resolution is high, the results with the canny edge maps are similar to those with the gPb map. However, when the superpixel resolution is low, the superpixels with the gPb map obtain more meaningful segmentation results. This observation is consistent with the quantitative performance. Fig. 11 gives more examples of our RPS method with the superpixel resolutions as 200 and 600, where our RPS method preserves most contours of the objects. Furthermore, thanks to the local junction constraints, our method also obtains homogeneous superpixels with explicit adjacency in lowly textured regions.

Our RPS method is based on the edge map. Therefore, the results of our method degrades when the edge detection method fails. One case is that the object has low contrast with the background. This often appears when the animal is with the protective color, such as the back of the bird in Fig. 12(a). Our method can not work well for the thin object, such as the snake in Fig. 12(b). Our RPS method produces the regular superpixels with uniform shapes (sizes), which constrain the minimum sizes of superpixels. Hence, our RPS method can not well cope with the tiny or thin objects. However, this case can be better handled by increasing the superpixel resolution.

C. Running Time

The main time-consuming part of our method is to find the shortest path. In our work, Dijkstra algorithm [44] is employed to search for the shortest path. Its time complexity is $O(\log(n) * e)$, where n and e are the numbers of nodes and edges, respectively. The superpixel resolution is N, and the junctions are connected by 4-neighborhood. The time complexity for the superpixel boundary generation step is $O(N * \log(n) * e)$. Moreover, our method generates the superpixel boundaries only based on the junction pairs and their local regions. So the regions are independent, and our algorithm can be further accelerated by using the parallel computing technique.

We carry out our experiment on a laptop with the Dual Core 2.8 GHz processor and 4 GB RAM. The code is implemented in Matlab without any optimization. We evaluate the running time on the BSDS dataset. Table I shows the average running time of the different superpixel methods. Our RPS method takes about 0.6 seconds per image using the Canny map. In fact, when the superpixel resolution is high, our method is not sensitive to the edge map, and any edge detector can be employed. When the superpixel resolution is low, the probability boundary map is recommended. However, most existing algorithms have similar performances for the "strong boundaries" [50]. It is better to select other efficient probability boundary detectors, such as PB [51] and BEL [52], to reduce the computational time.

D. Performance Comparison for Supervoxels

Fig. 14 shows some visual examples of our supervoxels, where the videos are obtained from [23]. The total numbers of frames for five videos are $K = 21, 42, 69, 85, 125$, respectively. We set the temporal interval $D = 6$ in Algorithm 1. Fig. 14(b) shows the superpixels, where we use our RPS method for each frame. And Figs. 14(c-d) show the supervoxels filled with the random color and the mean color of the pixels. Compared with other existing supervoxel methods, our method obtains the regular supervoxels, which preserving the structural relations on both spatial and temporal spaces of the video. We evaluate our method on the video dataset [53]. We compare our method with two state-of-the-art supervoxel methods: Meanshift [21] and HGB [23]. Fig. 13 shows the explained variation performances on the entire video. Our regular supervoxel method does not obtain the best 3D explained variation score, because our supervoxel method prefers to preserve the spatio-temporal relationship. Note that when the supervoxel resolution is low, our method outperforms HGB [23], because it can achieve the good tradeoff between the regularity and the segmentation accuracy.

V. Conclusion

In this paper, we have presented a new method to generate regularity preserved superpixels, in which the regularity and the boundary adherence are both represented. Our method obtains regular, homogeneous, and isometric superpixels. It also maintains the segmentation accuracy in the high contrast contents and object contents. Moreover, our approach can be also used to obtain the regular supervoxels for the videos, by preserving the spatio-temporal structure. The experiments demonstrate the effectiveness of our approach.

References

[1] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proc. CVPR*, 2011, pp. 409–416.

[2] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. CVPR*, 2012, pp. 853–860.

[3] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3766–3778, Oct. 2013.

[4] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in *Proc. CVPR*, 2004, pp. 326–333.

[5] B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. ICCV*, 2009, pp. 670–677.

[6] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Optimal contour closure by superpixel grouping," in *Proc. ECCV*, 2010, pp. 480–493.

[7] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proc. CVPR*, 2012, pp. 789–796.

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

[9] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. CVPR*, 2011, pp. 2097–2104.

[10] 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.

[11] 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.

[12] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *Proc. CVPR*, 2008, pp. 1–8.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[14] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. ICCV*, 2003, pp. 10–17.

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

[16] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha, "Structure-sensitive superpixels via geodesic distance," *Int. J. Comput. Vision*, vol. 103, no. 1, pp. 1–21, 2013.

[17] A. P. Moore, S. J. D. Prince, and J. Warrell, ""lattice cut" - constructing superpixels using layer constraints," in *Proc. CVPR*, 2010, pp. 2117–2124.

[18] D. Tang, H. Fu, and X. Cao, "Topology preserved regular superpixel," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2012, pp. 765–768.

[19] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Spatiotemporal closure," in *Proc. Asian Conf. Computer Vision*, 2011, pp. 369–382.

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

[21] S. Paris, "Edge-preserving smoothing and mean-shift segmentation of video streams," in *Proc. ECCV*, 2008, pp. 460–473.

[22] M. Sargin, L. Bertelli, B. Manjunath, and K. Rose, "Probabilistic occlusion boundary detection on spatio-temporal lattices," in *Proc. ICCV*, 2009, pp. 560–567.

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

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

[25] X. Wang and X. Zhang, "A new localized superpixel markov random field for image segmentation," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2009, pp. 642–645.

[26] J. Cheng, J. Liu, Y. Xu, F. Yin, D. Wong, N. Tan, D. Tao, C. Cheng, T. Aung, and T. Wong, "Superpixel classification based optic disc and optic cup segmentation for glaucoma screening," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 1019–1032, 2013.

[27] Z. Tu, X. Chen, A. L. Yuille, and S. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *Int. J. Comput. Vision*, vol. 63, no. 2, pp. 113–140, 2005.

[28] P. Kohli, L. Ladicky, and P. Torr, "Robust higher order potentials for enforcing label consistency," in *Proc. CVPR*, 2008, pp. 1–8.

[29] K. Zeng, N. Zhao, C. Xiong, and S. Zhu, "From image parsing to painterly rendering," *ACM Trans. Graph.*, vol. 29, no. 1, pp. 1–11, 2009.

[30] S. Liu, S. Yan, T. Zhang, C. Xu, J. Liu, and H. Lu, "Weakly supervised graph propagation towards collective image parsing," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 361–373, 2012.

[31] F. Perronnin, J.-L. Dugelay, and K. Rose, "Iterative decoding of two-dimensional hidden Markov models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2003, vol. 3, pp. 329–323.

[32] P. Carr and R. Hartley, "Minimizing energy functions on 4-connected lattices using elimination," in *Proc. CVPR*, 2009, pp. 2042–2049.

[33] Y. Lee and K. Grauman, "Object-graphs for context-aware visual category discovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 346–358, 2012.

[34] M. J. Choi, A. Torralba, and A. Willsky, "A tree-based context model for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 240–252, 2012.

[35] L. Liang, W. Feng, L. Wan, and J. Zhang, "Maximum cohesive grid of superpixels for fast object localization," in *Proc. CVPR*, 2013.

[36] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006, pp. 2169–2178.

[37] J. Choi, Z. Wang, S. Lee, and W. Jeon, "A spatio-temporal pyramid matching for sports video retrieval," *Comput. Vision Image Understand.*, vol. 117, no. 6, pp. 660–669, 2013.

[38] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. ACM Int. Conf. Image and Video Retrieval*, 2007, pp. 401–408.

[39] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proc. British Machine Vision Conf.*, Sep. 2008, pp. 995–1004.

[40] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008, pp. 1–8.

[41] C. Chen and K. Grauman, "Efficient activity detection with max-subgraph search," in *Proc. CVPR*, 2012, pp. 1274–1281.

[42] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua, "Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features," *IEEE Trans. Med. Imag.*, vol. 31, no. 2, pp. 474–486, 2012.

[43] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[44] E. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, no. 1, pp. 269–271, 1959.

[45] P. F. Felzenszwalb and R. Zabih, "Dynamic programming and graph algorithms in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 721–740, 2011.

[46] F. Meng, H. Li, G. Liu, and K. N. Ngan, "Object co-segmentation based on shortest path algorithm and saliency model," *IEEE Trans. Multimedia*, vol. 14, no. 5, pp. 1429–1441, 2012.

[47] 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. ICCV*, 2001, pp. 416–423.

[48] L. Payne, "Isoperimetric inequalities and their applications," *SIAM Rev.*, vol. 9, no. 3, pp. 453–488, 1967.

[49] R. Montero and E. Bribiesca, "State of the art of compactness and circularity measures," in *Proc. Int. Mathematical Forum*, 2009, vol. 4, no. 25-28, pp. 1305–1335.

[50] X. Hou, A. Yuille, and C. Koch, "Boundary detection benchmarking: Beyond f-measures," in *Proc. CVPR*, 2013.

[51] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, 2004.

[52] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proc. CVPR*, 2006, pp. 1964–1971.

[53] D. Tsai, M. Flagg, A. Nakazawa, and J. Rehg, "Motion coherent tracking using multi-label MRF optimization," *Int. J. Comput. Vision*, pp. 1–13, 2012.

Huazhu Fu is a Research Fellow in School of Computer Engineering at Nanyang Technological University (NTU), Singapore. He received the B.S. degree from Nankai University in 2006, the M.E. degree from Tianjin University of Technology in 2010, and the Ph.D. degree from Tianjin University, China, in 2013. His current research interests include computer vision, medical image processing, multiple object correspondence, and image segmentation.

Xiaochun Cao is a professor at the Institute of Information Engineering, Chinese Academy of Sciences since 2012. He received the B.E. and M.E. degrees both in computer science from Beihang University (BUAA), China, and the Ph.D. degree in computer science from the University of Central Florida, USA, with his dissertation nominated for the university-level Outstanding Dissertation Award. After graduation, he spent about three years at ObjectVideo Inc. as a Research Scientist. From 2008 to 2012, he was a professor at Tianjin University. He has authored and coauthored over 50 journal and conference papers. In 2004 and 2010, Dr. Cao was the recipients of the Piero Zamperoni best student paper award at the International Conference on Pattern Recognition.

Dai Tang received her B.S. degree in Software Engineering from Tianjin University in 2012. Her research work includes: superpixel segmentation and supervoxel segmentation.

Yahong Han received the Ph.D. degree from Zhejiang University, Hangzhou, China in 2012. He is currently an Associate Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His current research interests include multimedia analysis, retrieval, and machine learning.

Dong Xu (M'07–SM'13) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, in 2001 and 2005, respectively. While pursuing his Ph.D., he was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years. He was a Post-Doctoral Research Scientist with Columbia University, New York, NY, for one year. In May 2007, he joined Nanyang Technological University, Singapore, where he is currently an Associate Professor with the School of Computer Engineering. His current research interests include computer vision, machine learning, and multimedia content analysis. Dr. Xu was the co-author of a paper that won the Best Student Paper Award in the IEEE International Conference on Computer Vision and Pattern Recognition in 2010.