WATERSHED SUPERPIXEL

分水岭

Zhongwen Hu [1] *Qin Zou*[2] *Qingquan Li*[⋆,1]

[1]Shenzhen Key Laboratory of Spatial Smart Sensing and Service,
Shenzhen University, P.R. China
[2]School of Computer Science, Wuhan University, P.R. China
{zwhoo@szu.edu.cn, qzou@whu.edu.cn, liqq@szu.edu.cn}

ABSTRACT

As a pre-processing tool, superpixel algorithms have been popular used in many computer-vision applications. High efficiency is a desired property of superpixel algorithms, especially in real-time vision systems. In this paper, a novel high-efficient superpixel algorithm is developed based on the watershed algorithm, namely the spatial-constrained watershed (SCoW). SCoW performs watersheding in a marker-controlled manner, with a set of evenly placed markers. To align superpixel boundaries to image edges, an edge-preserving scheme is embedded into the SCoW which makes a balance between the homogeneity and the compactness. Without any complex computing, the proposed superpixel algorithm is found to produce high quality superpixels as traditional superpixel algorithms, while holding much higher efficiency.

Index Terms— superpixel, watershed, spatial constraint, image segmentation

1. INTRODUCTION

Superpixel segmentation, which over-segments an image into a number of superpixel regions, is a widely used pre-processing step for image- and vision-based systems. A superpixel region is commonly defined as a perceptually uniform and homogenous region in the image [1, 2]. With the development of computer vision and multimedia techniques, superpixel algorithms have becoming increasingly popular for many applications, e.g., object detection and tracking [3], image segmentation and modeling [4, 5, 6], salience detection [7, 8, 9, 10] and image and object classification [11], etc.

(a) (b) (c)

Fig. 1. Watershed Superpixels. (a) Traditional watershed, (b) watershed with uniform markers but without spatial constrain, (c) the proposed SCoW.

One main advantage of using superpixel is to improve the computational efficiency in the corresponding systems. With superpixels, the later-stage processing would be speedup since the number of image primitives has been dramatically reduced as compared to the original pixel representations. Another advantage is that, the superpixels provide to investigate the image in a larger scale, which can give better spatial support to compute regional features.

In the past two decades, important efforts have been made in the field of image segmentation, e.g., watershed segmentation [12], Normalized Cut [13], graph-based segmentation [14], statistic region merging [15], etc. Generally, image segmentation is used as a pre-processing step for vision computing applications. However, with increasing demand of high-performance pre-processing, the above segmentation methods gradually show their limitations, and superpixel algorithms are found to be more suitable as a pre-processing tool. Exactly, a set of famous superpixel algorithms have been proposed in the past decade [16, 17, 1], In [16], a method named Turbopixels introduces spatial constraint to level-set evolution to get dense over-segmented and compact superpixels. While in SLIC [1], spatial constraint is embedded into a k-mean clustering method to achieve superpixel segmentation. These two methods can produce superpixels effectively, however both requires a complex iterative process to refine the results. Some newly developed algorithms can be found in [18, 19, 20]. In [19], superpixels are gained by a structure sensitive over-segmentation technique which exploits Lloyd's k-means algorithm with a geodesic distance.

This study is jointly supported by grants from Postdoctoral Science Foundation of China under grants No. 2014M562206 and2012M521472, Geographic National Condition Monitoring Engineering Research Center of Sichuan Province under grant No. GC201513, National Natural Science Foundation of China under grant No.61301277, Shenzhen Scientific Research and Development Funding Program under grants No. ZDSY20121019111146499 and No. JSGG20121026111056204, Shenzhen Dedicated Funding of Strategic Emerging Industry Development Program under grant No. JCYJ20121019111128765 and National Basic Research Program of China (973 Program) under grant No. 2012CB725303. *Corresponding author: Dr. Q.Q. Li.

In [18], superpixels are produced in a energy-driven way, where a hill-climbing algorithm is used to continuously refine the superpixels by modifying the boundaries. In [20], superpixel segmentation is achieved in a lazy random walk framework, which iteratively optimized the energy function. Most of the above superpixel algorithms are computational expensive, e.g., it may take a few seconds even minutes to segment an image of a moderate size, e.g., 481×321 pixels. Since the superpixel segmentation serves as a basis for later-stage processing, higher efficient algorithms are always desired.

Note that, the watershed transform can be high-efficiently performed. In this paper, we propose to develop a new superpixel algorithm based on the watershed. However, without spatial constraint, it produces superpixels of irregular shapes and sizes, as shown in Fig. 1(a) and (b). Therefore, we present to use spatial constrain to the watershed, which leads to a spatial-constrained watershed superpixel algorithm (SCoW). The proposed algorithm introduces spatial constraints and uniform markers to obtain compact and evenly distributed superpixels, as shown in Fig. 1(c), while offering a significant boost in the speed. Moreover, an edge-preserving scheme is used to align superpixel boundaries to the image edges.

The remainder of this paper is organized as follows. Section 2 describes details of the proposed spatial-constrained watershed algorithm. Section 3 reports the experiment results, followed by a brief conclusion in Section 4.

2. SPATIAL-CONSTRAINED WATERSHED

2.1. Marker-controlled Watershed

The basic idea of marker-controlled watershed transform is that flooding the topographic surface from a previously defined set of markers. The gradient image is viewed as topographic surface, thus, pixels with low gradient values will be flooded with high priority. Meyer proposed the marker-controlled watershed by utilizing a set of ordered queues (Fig.2) with different priorities [12]. In this method, marker points are firstly put into the ordered queue, depending on their priority; and then pixels with highest priorities are pulled out and labeled; the unlabeled neighbors of this pixels are then pushed into the queue. When the ordered queue is empty, each pixel of the image is labeled. A watershed transform is usually applied on gradient image, and the priority inverse to the gradient magnitude. The details of implementation of this method can be seen in [12].

The pixels are labeled according to their priorities, which is defined as follows:

$$p(x, y) = g_{max} - g(x, y) \tag{1}$$

where $p(x, y)$ denotes the priority of pixel located at (x, y) in the image and $g(x, y)$ is the gradient magnitude value; g_{max}

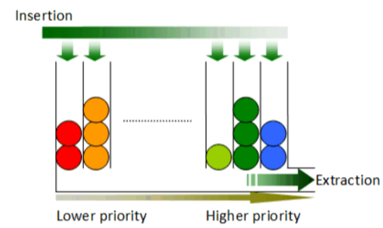

Fig. 2. An illustration of ordered queue with different priorities.

stands for the maximum possible gradient magnitude in the image; its value is equal to the number of the ordered queues.

The marker-controlled watershed starts with selecting a set of markers. To make the superpixels have uniform sizes and even distribution, the proposed method starts with placing evenly distributed marker points in the image with a step $w \times w$, where w is the grid interval. To avoid the markers located on the image edges, the marker points are perturbed in an 3×3 neighborhood to the lowest gradient positions.

2.2. Spatial Constraint and Edge Preserving

Meyer's method [12] processes pixels only depending on their gradient values, which often leads to irregular shapes and sizes. Thus, spatial constraint is introduced in this paper to force the superpixels compact. The basic idea of spatial constraint is that the priority should not only depend on its gradient value, but also on the spatial pattern. The basic assumption is that under similar conditions, the pixels closer to their corresponding markers should be processed with higher priorities. On the other hand, when the distances from the markers are the same, pixels with lower gradient values should be given higher priorities. Then we define the pixel priority as follows:

$$p'(x, y) = p_g(x, y) + \lambda p_s(x, y) \tag{2}$$

where $p_g(x, y)$ denotes pixel priority derived from the gradient; and $p_s(x, y)$ denotes the pixel priority derived from spatial constraint; λ is the user input parameter that balancing two measurements.

The spatial constraint ensures that pixels near marker points have higher priorities. Moreover, the superpixel boundaries should align to the image edges. Thus, the edge-preserving scheme should be considered to eliminate the over-smoothed effect due to strong spatial constraint near the edges. The pixel priority derived from spatial constraint considering edge-preserving scheme is defined as :

$$p_s(x, y) = -d_{x,y} \cdot E(x, y) \tag{3}$$

where $d_{x,y}$ denotes the Euclidean distance from pixel (x, y) to its corresponding marker point; $E(x, y)$ is the edge-preserving function, which balances the spatial constraint

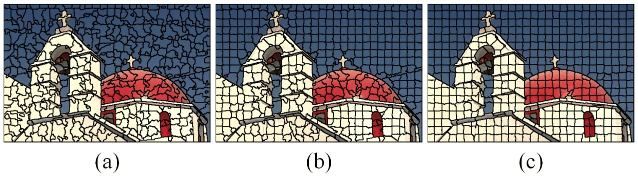

Fig. 3. SCoW superpixels with different spatial constraint ($\alpha = 5, w = 17$). (a) $\lambda = 0$; (b) $\lambda = 0.25$; (c) $\lambda = 0.5$.

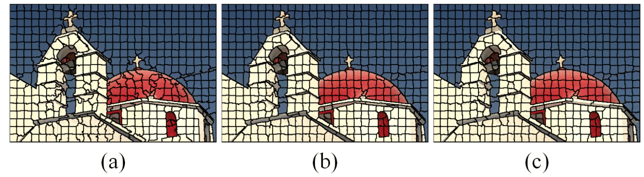

Fig. 4. SCoW superpixels with different spatial constraint ($\lambda = 0.5, w = 17$). (a) $\alpha = 5$; (b) $\alpha = 15$; (c) $\alpha = 30$.

on image edges. It is defined as:

$$E\left(x, y\right) = e^{-g(x,y)/\alpha} \tag{4}$$

where α is the parameter that balances the spatial constraint of different gradient values. The function $E(x, y)$ decreases monotonically with the gradient value. It means that the spatial constraint should be decreased as it reach to image edges. The parameter α influences the decreasing speed of the function. It's easy understanding from Equation.4 that the bigger the α, the slower the speed and stronger spatial constraint.

As is stated in Equation 3, pixels far from its corresponding markers will have low priorities. Moreover, as the edge-preserving function decreases monotonically with the gradient value, it alleviates strong spatial constraint around image edges, and makes the superpixel boundaries align well to the image edges.

3. RESULTS AND COMPARISONS

In this section, we firstly demonstrate the effectiveness and analysis the effect of parameters of the proposed SCoW[1], and then we compare SCoW with state-of-the-art superpixel algorithms: SLIC[1] and Turbopixles [16].

3.1. Analysis of Parameters

In this section, we evaluate the effectiveness of spatial constraint, and analysis the impact of parameters. In the SCoW, the parameter λ controls the weight of spatial constraint, and α controls the balance of spatial constraint and image edges. A set of superpixels are obtained using $\lambda = (0, 0.25, 0.5)$ and $\alpha = (5, 15, 30)$. The results are shown in Fig.3 and Fig.4.

As shown in Fig.3, when there's no spatial constraint ($\lambda = 0$), the watershed transform generates superpixels with irregular shapes and sizes, despite the markers are evenly distributed. With the increase of spatial constraint, the superpixels become more compact, and the sizes become more uniform.

Fig.4 presents the results obtained using different α. As shown in the figures, the parameter does not affect a lot over flat areas. But it affects the shape of boundaries near image edges. When α is low, the boundaries align better to weak and strong image edges, but the superpixels are less compact; as α increases, the superpixels become more compact, and the boundaries can only align well to strong edges. It can be seen from Fig.4 (b) and (c), the superpixels do not differ a lot. For most of the cases, we suggest $\alpha = 15$, from our experiments.

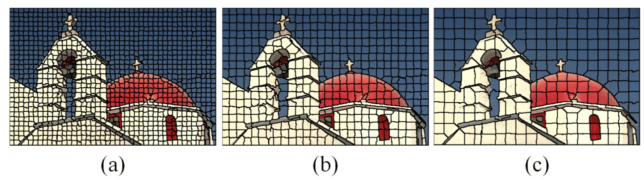

Fig. 5. SCoW superpixels of different sizes (rgSize). (a) rgSize=169 (w=13); (b) rgSize=289 (w=17); (c) rgSize=441 (w = 21).

The parameter w controls the size of superpixel. The average size of superpixels is about $w \times w$. The results are shown in Fig. 5. It is shown in the figures, although the sizes are different, the superpixels are compact and uniform, and the boundaries align well to image edges.

3.2. Comparison with the state-of-the-art superpixel

The parameters $\lambda = 0.4$,$\alpha = 15$ and $w = 13$ are used. Moreover, two state-of-art superpixel segmentation methods Turbopixels [16] and SLIC [1] are used in comparison.

3.2.1. Visual Comparison

Figure 6 provides two results for visual comparison of our output against the TurboPixels and SLIC. As is shown, all of the three algorithms produce compact and uniform superpixels. However, since the edge-preserving scheme is taken into account in the SCoW algorithm, the spatial constrained should be alleviated to achieve consistency of boundaries. It is clearly illustrated in the zoomed areas that the proposed SCoW performs better than SLIC and TurboPixel.

3.2.2. Quantitative Metrics

Three metrics are employed to evaluate the performance of this method: boundary recall (tolerance is 1 pixel in this experiment), under-segmentation error and achievable overall accuracy [21].

[1]Codes are available at https://sites.google.com/site/qinzoucn/documents

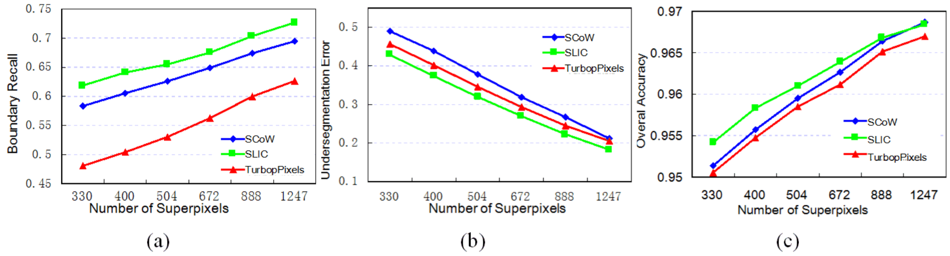

Fig. 7. Comparisons in terms of (a) Boundary Recall, (b) Under-segmentation Error, and (c) Overall Accuracy.

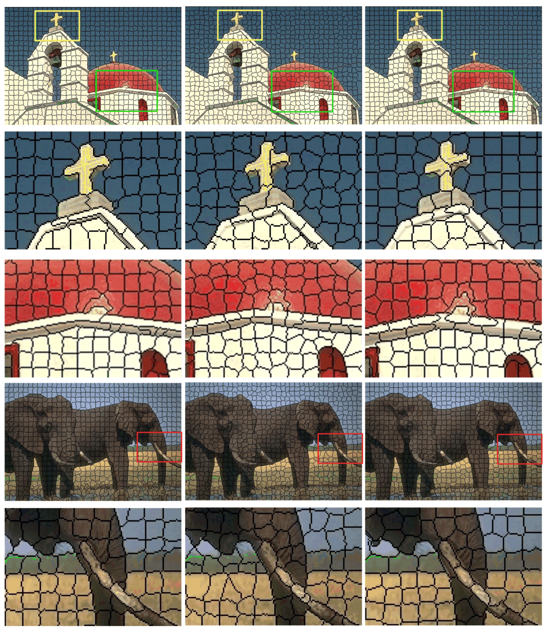

Fig. 6. Visual comparison of the superpixels. Column a: the proposed SCoW algorithm. Column b: TurboPixels [16]. Column c: SLIC [1].

The results are presented in Fig.4 a,b and c. They are plotted against the number of superpixels of the results.

As shown in Fig.7 a, b and c, the SCoW algorithm offers better boundary recall and overall accuracy than the TurboPixels. The main limitation is that the SCoW algorithm may risk slightly higher under-segmentation error. The main reason is that, in some cases, the sizes of superpixels near image edges are larger than SLIC and Turbopixel, such as the results shown in the third row of Fig. 7. However, as the number of superpixels increases, the SCoW algorithm performs comparable to the SLIC, especially the overall accuracy.

3.2.3. Algorithm Complexity and Timing Efficiency

Since one of the major advantages of using superpixels is its acceleration of later-stage processing, it should be fast and efficient. In the computation of gradient image and the flooding process, every pixel will be processed only once separately and no iteration is need, thus, the SCoW achieves O(N) complexity, the same as the SLIC and TurboPixels.

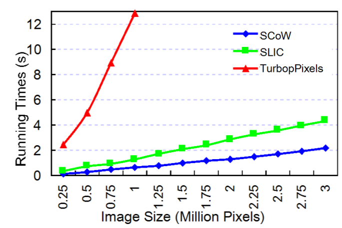

Fig. 8. Comparisons of running efficiency.

We compare SCoW with SLIC and TurboPixels using a set of images with different sizes. In each experiment, the numbers of superpixels for SLIC and TurboPixels are set ensuring the average superpixel encompasses about 170 pixels. The parameters for SCoW are $\lambda = 0.4$, $\alpha = 15$ and $w = 13$. The curves are plotted against the image sizes in Fig. 8.

As shown in Fig.8, the SCoW and the SLIC perform much faster than the TurboPixels. What surprising us is that the running time of SCoW is about 50% of the SLIC algorithm. Our algorithm takes about 0.1 second for segmenting an image of 500×500 pixels and about 2 seconds for an image of 2000×1500 pixels on a notebook computer with a 2.4 GHz CPU and 2G RAM.

4. CONCLUSION

A highly efficient spatial-constrained watershed algorithm (SCoW) was proposed for superpixel segmentation in this paper. The proposed SCoW introduced spatial constraint in the marker-controlled watershed to obtain compact and evenly distributed superpixels. Experiments demonstrated the effectiveness and the outstanding efficiency of SCoW. As such, SCoW is very suitable for pre-processing in real-time computer vision and engineering applications. Moreover, the proposed SCoW can be easily implemented in parallel to further improve the efficiency.

5. REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[2] David Stutz, Alexander Hermans, and Bastian Leibe, "Superpixel segmentation using depth information," September 2014.

[3] Christopher Rasmussen, "Superpixel analysis for object detection and tracking with application to uav imagery," in *ISVC*, pp. 46–55. Springer, 2007.

[4] Xiaofeng Ren and Jitendra Malik, "Learning a classification model for segmentation," in *ICCV*. 2003, pp. 10–17, IEEE.

[5] B Micusik and J Kosecka, "Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry," in *ICCV Workshops*. IEEE, 2009, pp. 625–632.

[6] Xiyu Song, Lili Zhou, Zhongguo Li, Jian Chen, Bin Yan, and Lei Zeng, "Interactive image segmentation based on hierarchical superpixels initialization and region merging," in *CISP*. IEEE, 2014, pp. 410–414.

[7] Na Tong, Huchuan Lu, Lihe Zhang, and Xiang Ruan, "Saliency detection with multi-scale superpixels," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1035–1039, 2014.

[8] Federico Perazzi, Philipp Krahenbuhl, Yael Pritch, and Alexander Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *CVPR*, 2012, pp. 733–740.

[9] Zhixiang Ren, Yiqun Hu, Liang-Tien Chia, and Deepu Rajan, "Improved saliency detection based on superpixel clustering and saliency propagation," in *ACM MM*, 2010, pp. 1099–1102.

[10] Na Tong, Huchuan Lu, Lihe Zhang, and Xiang Ruan, "Saliency detection with multi-scale superpixels," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1035–1039, 2014.

[11] Bin Liu, Hao Hu, Huanyu Wang, Kaizhi Wang, Xingzhao Liu, and Wenxian Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 907–924, 2013.

[12] Fernand Meyer, "Color image segmentation," in *International Conference on Image Processing and its Applications*, 1992, pp. 303–306.

[13] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[14] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[15] Richard Nock and Frank Nielsen, "Statistical region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[16] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.

[17] Alastair P Moore, Simon J D Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones, "Superpixel Lattices," in *CVPR*, 2008.

[18] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool, "SEEDS: Superpixels extracted via energy-driven sampling," in *ECCV*, pp. 13–26. 2012.

[19] Peng Wang, Gang Zeng, Rui Gan, Jingdong Wang, and Hongbin Zha, "Structure-sensitive superpixels via geodesic distance," *International journal of computer vision*, vol. 103, no. 1, pp. 1–21, 2013.

[20] Jianbing Shen, Yunfan Du, Wenguan Wang, and Xuelong Li, "Lazy random walks for superpixel segmentation.," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1451–1462, 2014.

[21] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa, "Entropy rate superpixel segmentation," in *CVPR*, 2011, pp. 2097–2104.