

LETTER

Fast Superpixel Segmentation via Boundary Sampling and Interpolation

Li XU[†], Bing LUO^{††a)}, *Members*, Mingming KONG^{††}, Bo LI^{†††}, and Zheng PEI^{††}, *Nonmembers*

SUMMARY This letter proposes a fast superpixel segmentation method based on boundary sampling and interpolation. The basic idea is as follow: instead of labeling local region pixels, we estimate superpixel boundary by interpolating candidate boundary pixel from a down-sampling image segmentation. On the one hand, there exists high spatial redundancy within each local region, which could be discarded. On the other hand, we estimate the labels of candidate boundary pixels via sampling superpixel boundary within corresponding neighbour. Benefiting from the reduction of candidate pixel distance calculation, the proposed method significantly accelerates superpixel segmentation. Experiments on BSD500 benchmark demonstrate that our method needs half the time compared with the state-of-the-arts while almost no accuracy reduction.

key words: superpixel, acceleration, sampling, interpolation

1. Introduction

Superpixel segmentation aims at obtaining local regions with appearance and location consistency. It is used to extract perceptually meaningful element regions, which significantly reduces the computation complexity for other computer vision applications, such as object segmentation [1], [2], object detection [3] and recognition [4].

In recent years, existing superpixel segmentation methods have made a great progress in terms of high segmentation accuracy and computation efficiency. For example, it evolves from the polynomial time, i.e., Ncut [5], $O(N^{\frac{3}{2}})$ and Meanshift [6] ($O(N^2)$) to linear time complexity $O(N)$, such as SLIC [7], Turbopixel [8], SSS [9] and LSC [10]. Although the latter ones obtain high performance with linear time, they still require multiple iteration and update until convergence, which handles realtime applications. Recently, Achanta et.al propose a simple non-iteration clustering-based superpixel to overcome the limitation of multiple iteration and disjoint region generation [11]. However, it also need calculate the distance between each pixel to its corresponding connected seed. Intuitively, superpixel accuracy is not determined by inner region in it, but its bound-

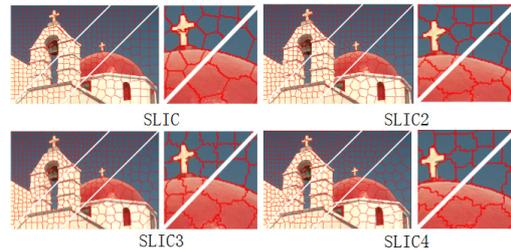


Fig. 1 The subjective results for original SLIC, step $s = 2$ (SLIC2), $s = 3$ (SLIC3) and $s = 4$ (SLIC4).

ary pixels. Therefore, reducing the distance calculation of superpixel's interior is critical for improving computational efficiency.

In this letter, we propose a novel fast and efficient superpixel algorithm by reducing the distance calculation of redundant pixels. To obtain accurate segmentation, a boundary interpolation method is proposed based on the sampling boundary pixels. Specifically, we first down-sample the original image evenly to obtain the superpixel segmentation of small scale image. Then, after identifying the candidate boundary pixel around the sampled points, we obtain accurate superpixel boundary via a novel boundary interpolation method. Benefited from reducing the distance calculation of inner redundant pixels, it significantly improves the computation efficiency. In Fig. 1, we show some subjective results for original SLIC and three versions of our method, i.e., step $s = 2$ (SLIC2), $s = 3$ (SLIC3) and $s = 4$ (SLIC4). It shows that our methods almost have the same segmentation results with original SLIC.

Feng et al. [12] also propose a down-sampling strategy to cluster pixels on the alternating points of an image. The distance computation is only reduced half of image pixels by their 4-labeled neighbors. In contrast, the proposed method estimates only superpixel boundary by interpolating candidate boundary pixel from a down-sampling image segmentation. Hence, high spatial redundancy within each local region are discarded, which results into fewer pixel distance calculation than method [12].

Our main contributions are concluded as follows:

- propose a new superpixel segmentation method based on boundary sampling and interpolation.
- propose an accurate nearest neighbour based interpolation method to fit different sampling scale.

Manuscript received August 12, 2018.

Manuscript revised December 4, 2018.

Manuscript publicized January 22, 2019.

[†]The author is with School of Science, Xihua University, Chengdu 610039, China.

^{††}The authors are with the Center for Radio Administration Technology Development, Xihua University, Chengdu 610039, China.

^{†††}The author is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China.

a) E-mail: mathild1987@163.com

DOI: 10.1587/transinf.2018EDL8168

2. Proposed Method

2.1 Sampling and Interpolation-Based Superpixel

Our basic idea is that there exists large redundancy in superpixel interior with regional consistency. In other words, adjacent pixels in superpixel interior have the same labels and ones in superpixel borders have different labels. We could only label the boundary pixels to improve computational efficiency. Then an interpolation-based superpixel method is proposed, which consists of two key steps:

- Extracting superpixel boundary on subsampled image.
- Interpolating candidate pixels on original image.

Specifically, given step s , we subsample original image along horizontally and vertically every s pixels. Then, we perform existing superpixel segmentation such as SLIC [7], LSC [10] and SNIC [11], etc, to obtain the rough superpixel segmentation. Let L as the labeling assignment for subsampling image. Two shift and difference operations are applied to extract superpixel boundaries. Intuitively, interior region in superpixel has same label and shifts 1 pixel will not change the labels. However, boundary pixels have different labels with their neighbour pixels, which are non-zero after shifting and differential operation.

As shown in Fig. 2, given subsampled image, the superpixel boundary is mapping to the original image, which could be defined as B . We utilize nearest neighbor interpolation to avoid the redundant computation in interior regions. However, this method will result in the sawtooth edge due to the inaccurate pixel labeling, as show in Fig. 3. In order to obtain the accurate and smooth boundary, we slide an $K \times K$ patch, where $K = 2s + 1$, on sampling boundary set B to cover the whole candidate superpixel boundary set. Then, the candidate boundary set is defined as B^c , which is the trajectory pixels on sliding patch except B . We define the set $B \cup B^c$ as the external region and other pixels constitute the interior region, as shown in Fig. 2 (c). Due to the consistency of interior pixels, we could discard the large label redundancy to improve the complexity.

For each $p \in B$, the candidate pixels associated with

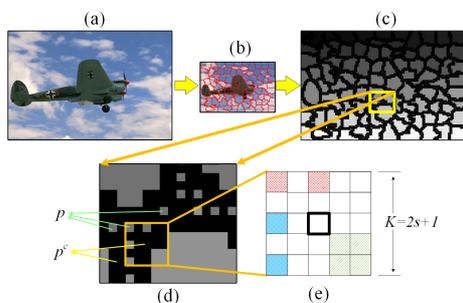


Fig. 2 (a) Input image. (b) Superpixel segmentation for subsampling image. (c) Labels mapping to original scale. (d) and (e) Label assigning for candidate boundary pixels. In the $K \times K$ square, pixel will be assigned as the label of adjacent cluster center with small distance.

Algorithm 1 Superpixel segmentation algorithm.

Require:

Input image I ;
Sampling scale s ;

Ensure:

Pixel labels L ;
1: Perform SLIC on subsampling image I ;
2: Obtain the superpixel sampling boundary B ;
3: Obtain candidate boundary set B^c ;
4:
5: **for** $i \in B^c$ **do**
6: **if** $|V_i| > 1$ **then**
7: $L(i) \leftarrow \arg \min_p D(I_i, I_p), I_p \in V_i$;
8: **else**
9: $L(i) \leftarrow L(p), I_p \in V_i$;
10: **end if**
11: **end for**

p are defined as $p^c \in B^c$, which is the adjacent region in $K \times K$ square. It makes sure that unsampled boundary pixel could be assigned correct labels. In details, $p^c \in B^c$ will be associated with multiple subsampled pixels and labeled as the label of pixel p with minimum distance by formulation:

$$l_i = \arg \min_p D(I_i, I_p), I_p \in V_i \quad (1)$$

where $V_i = \{L(p) | p \in B \cap K \times K\}$ and is the boundary label set associated with pixel I_i . Each candidate pixel is assigned as the label of associated boundary pixel I_p with minimum distance. Furthermore, pixels are inspected in a square candidate region. We set p^c as unsampled pixels. Then, $L(p^c) = L(p)$, if $|V| = 1$. Otherwise, the label is assigned by Eq. (1). The algorithm is summarized as Algorithm 1.

2.2 Time Complexity

In original SLIC, computational cost depends on the number of candidate clusters for each pixel. Except the pixels along image boundary, each pixel calculates the distance from itself to the four closed cluster centers. The expected number of candidate clusters per pixels is defined as

$$E_{SLIC} = \sum_c p(|w| = c) \cdot c \quad (2)$$

where c is the number of candidate cluster and w is the set of pixels whose candidate clusters are with the same number. E_{SLIC} indicates the expected number of distance calculation.

In contrast, our method has following expression:

$$E = \frac{1}{s^2} E_{SLIC} + \sum_{w \in B^c, c} p(|w| = c) \cdot c \quad (3)$$

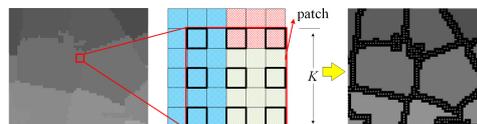


Fig. 3 Illustration for generation of external region. The red sliding window is used to cover the unsampled pixels.

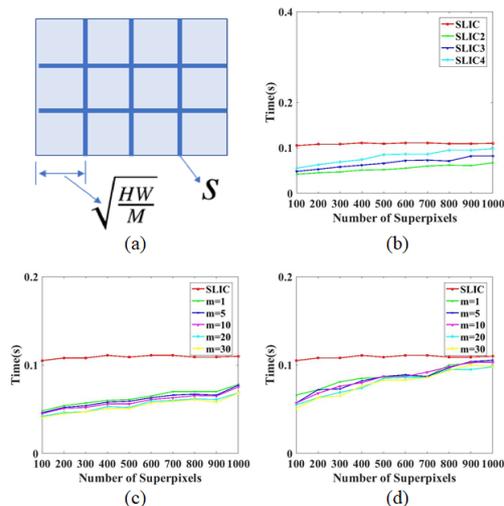


Fig. 4 (a) The illustration for boundary pixel. (b) The computational time for different sampling versions compared with the original SLIC. (c-d) The computational time for different compactness parameters of SLIC2 (c) and SLIC4 (d), respectively.

where the second term is the number of distance calculations for boundary pixels. The pixels who participate in the distance calculation will just locate along the superpixel boundaries, as shown in Fig. 4 (a). For simplicity, we segment $H \times W$ image with M superpixels. Then, each patch contains a square with side $b = \sqrt{\frac{HW}{M}}$. The image approximately contains $\frac{H}{b} - 1$ rows, $\frac{W}{b} - 1$ columns boundary ties, which contain $(\frac{H}{b} - 1) \cdot W \cdot s$ and $(\frac{W}{b} - 1) \cdot H \cdot s$ pixels, respectively, where s is stride length. Hence, the proportion of the candidate boundary set B^c can be approximately expressed as $(2\sqrt{\frac{M}{HW}} - \frac{H+W}{HW}) \cdot s$. Each unlabeled pixel will inspect the distance to no more than 8 adjacent boundary pixels. Then, Eq. (3) can be calculated as

$$E = \frac{E_{SLIC}}{s^2} + (2\sqrt{\frac{M}{HW}} - \frac{H+W}{HW}) \cdot s \cdot \sum_{c=4}^8 p(|w|=c) \cdot c \quad (4)$$

On the one hand, it can be seen that the computation complexity E will raise along with the increasing superpixel number M . On the other hand, increasing the sampling scale will not always reduce the computational time, especially for the equilibrium of the first and second term. In the experiments, we give the corresponding time comparison.

3. Experiment and Analysis

We perform experiments on BSD500 segmentation benchmark from two perspectives: computational efficiency and segmentation performance.

3.1 Computational Efficiency

We validate the computational efficiency with different scales on a 3.3GHz core i5 processor with 8GB of RAM. We

set step size as $s = 2, 3, 4$. Figure 4 (b) shows the computational time for different sampling versions compared with the original SLIC [7]. In experiments, we set the compactness parameters for all SLIC versions as $m = 20$. It can be seen that SLIC2, SLIC3, SLIC4 run faster than the original SLIC. Specifically, SLIC [7] takes more than 0.1s for a 481×321 image on average. However, the average running time of SLIC2 is close to 0.05s. Meanwhile, with the rise of superpixel number, the running cost of SLIC2 has a slight increase. The reason is that more boundary pixels participate into distance calculation.

Furthermore, the running time of different sampling versions increases with the extern step from $s = 2$ to $s = 4$. As the range K of candidate boundary regions enlarges, it needs more time to inspect the boundary pixel labels, which results into inferior computational efficiency. We also compare running time of different compactness parameters $m = 1, 5, 10, 20, 30$ for SLIC2 and SLIC4 with the original SLIC in Fig. 4 (b). It shows that the running time of SLIC is robust to the compactness parameters m . Our methods will increase the running time as the increasing superpixel number. Meanwhile, the lower the compactness parameter m is, the more boundary pixels participate into distance calculation. For the worst case, i.e., $m = 1$ and $s = 4$, SLIC4 is also faster than SLIC, which shows the effectiveness of our methods. In addition, we perform our methods based on matlab + C code without any optimization. In future, we will improve the code via C.

The limitation of our method is that the time complexity will raise along with the increasing superpixel number, especially for large sampling step, i.e., $s \geq 4$. The time cost for SLIC4 will be higher than the original SLIC over 1000 superpixels, as shown in Fig. 4 (b). Therefore, our method could only significantly accelerates superpixel segmentation with small sample step $s = 2, 3$.

3.2 Segmentation Performance

Some subjective results are shown in Fig. 5. It shows that our methods and the original version, i.e., SLIC, SLIC2, SLIC3 and SLIC4 almost have the same segmentation results. Moreover, SLIC2 obtains more accurate segmentation than SLIC for the girl's finger. It is difficult to find any difference among the compared methods for the fish's boundary, which demonstrates the efficiency of our method.

In terms of objective results, three metrics, boundary recall (**BR**), under segmentation error (**UE**) and achievable segmentation accuracy (**ASA**) [13] are used to measure segmentation performance. Figure 6 shows that our three versions, SLIC2, SLIC3 and SLIC4 all obtain the approximate segmentation performance with the original SLIC. Especially for **BR**, the proposed methods are all slightly higher than SLIC. The reason is that the labels of candidate boundary pixels is assigned as the minimum distance to the adjacent labeled pixels, which is more accurate than the minimum distance to the adjacent superpixel centers of the original SLIC. Meanwhile, we note that the larger sampling step

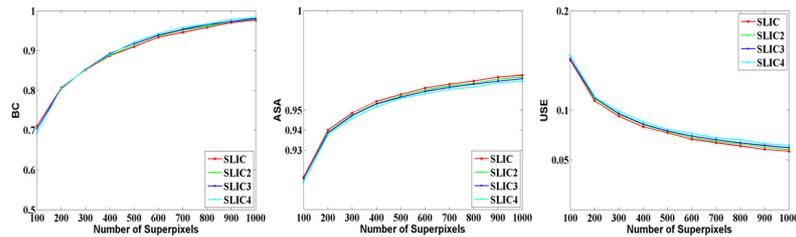


Fig. 6 The objective results for our methods and the original SLIC in terms of BR, UE and ASA.

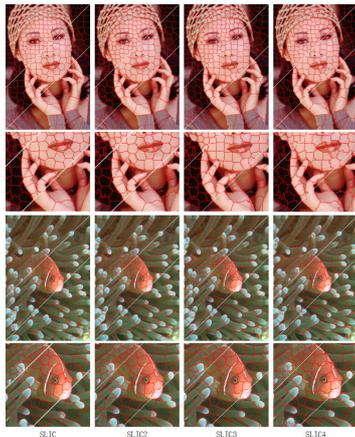


Fig. 5 Some subjective results of compared methods with 100, 300 and 600 superpixels, respectively.

obtains higher boundary recall, i.e., SLIC4 has higher BR value than SLIC3. The more candidate boundary pixels will fit object boundary more accurate, which leads higher BR. In contrast, the performance of the proposed methods are slightly lower than the original SLIC in terms of UE and ASA. This is because that these two metrics measure the regularity of superpixel regions. While, method based on pixel distance to adjacent superpixel centers could generate more regular superpixel regions than the proposed method, which is based on the distance between the candidate pixel to the associate subsampling pixel with minimum distance.

4. Conclusions

This letter proposes a fast superpixel segmentation method based on boundary sampling and interpolation. Instead of labeling local region pixels, we estimate superpixel boundary by interpolating candidate boundary pixel from a down-sampled image segmentation. Firstly, high spatial redundancy within each local region have been discarded. Then we estimate the labels of candidate boundary pixels via sampled superpixel boundary within corresponding neighbour. Due to the reduction of candidate pixel distance calculation, our method significantly accelerates the superpixel segmentation. Experiments on BSD500 benchmark demonstrate our method needs half the time compared with state-of-the-arts while almost no accuracy reduction.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61801398, 61372187, 61473239), the Scientific and Technological Project of Sichuan Province (No. 2018GZ0256).

References

- [1] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized prim's algorithm," Proceedings of the IEEE international conference on computer vision, pp.2536–2543, 2013.
- [2] B. Luo, H. Li, T. Song, and C. Huang, "Object segmentation from long video sequences," Proceedings of the 23rd ACM international conference on Multimedia, pp.1187–1190, ACM, 2015.
- [3] J.R. Uijlings, K.E. Van De Sande, T. Gevers, and A.W. Smeulders, "Selective search for object recognition," International journal of computer vision, vol.104, no.2, pp.154–171, 2013.
- [4] G. Mori, X. Ren, A.A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., pp.II–326–II–333 vol.2, 2004.
- [5] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.22, no.8, pp.888–905, 2000.
- [6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Transactions on pattern analysis and machine intelligence, vol.24, no.5, pp.603–619, 2002.
- [7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slıc 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.
- [8] A. Levinstein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.31, no.12, pp.2290–2297, 2009.
- [9] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha, "Structure-sensitive superpixels via geodesic distance," International journal of computer vision, vol.103, no.1, pp.1–21, 2013.
- [10] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," CVPR, pp.1356–1363, 2015.
- [11] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4895–4904, 2017.
- [12] H. Feng, F. Xiao, Q. Bu, F. Liu, L. Cui, and J. Feng, "An accelerated superpixel generation algorithm based on 4-labeled-neighbors," CCF Chinese Conference on Computer Vision, vol.771, pp.539–550, Springer, 2017.
- [13] M. Wang, X. Liu, Y. Gao, X. Ma, and N.Q. Soomro, "Superpixel segmentation: A benchmark," Signal Processing: Image Communication, vol.56, pp.28–39, 2017.