SSGD: SUPERPIXELS USING THE SHORTEST GRADIENT DISTANCE

Ning Zhang and Lin Zhang[*]

School of Software Engineering, Tongji University, Shanghai, China

ABSTRACT

As a pre-processing step for many problems in the field of computer vision, superpixel algorithms aim to over-segment the image by grouping homogenous pixels. In this paper, we propose a novel superpixel segmentation algorithm, namely *S*uperpixel using the *S*hortest *G*radient *D*istance (SSGD for short) in a k-means clustering framework. Starting from initializing the superpixel seeds, bilateral filtering is applied to the texture-rich regions centered at initial seeds. Then, a novel distance function taking the shortest gradient distance into account is computed to enforce adherence to boundaries. Unlike using the simple Euclidean distance, the proposed combined distance function increases the accuracy of associating a pixel to a cluster. The experimental results demonstrate that our algorithm outperforms the state-of-the-art methods in this field. Source codes of SSGD are publicly available at http://sse.tongji.edu.cn/linzhang/ssgd/index.htm.

***Index Terms*—** Superpixels, segmentation, shortest gradient distance

1. INTRODUCTION

Superpixels are regarded as perceptually meaningful atomic regions [1]. Recently, superpixel methods have been applied to many computer vision tasks, such as semantic segmentation [2], tracking [3], gesture recognition [4], etc.

Research on superpixels has drawn great attention during the past decade. Most of the superpixel algorithms are graph-based or clustering-based. Graph-based approaches consider pixels as nodes of a graph, and usually propose a cost function. By optimizing the cost function, graph-based methods are able to enforce color homogeneity. Shi *et al.* proposed N-Cuts [5] and minimized the cost function by formulating it as a generalized eigenvalue problem. GraphCut [6] proposed by Veksler *et al.* can generate compact superpixels in an energy minimization framework.

In contrast to graph-based methods, clustering-based algorithms group pixels into clusters and refine them until the specific conditions are satisfied. Some representative papers belonging to this category are reviewed here. Turbopixels [7] uses geometric flow to yield compact and regular superpixels efficiently. The *Simple Linear Iterative Clustering* [8] algorithm (SLIC) introduces a local k-means approach to organize pixels in CIELAB color space. This method may fail to cling to object contours. Recently, several variants of SLIC have been proposed to improve its performance of clinging to the contours. Zhang *et al.* [9] introduced a boundary term to enhance boundary constraint. A structure-sensitive algorithm [10] based on geodesic distance can generate adaptive superpixels according to the region density. SCALP [11] includes a contour prior and computes the linear path to the cluster barycenters to ensure both the regular size and the color homogeneity; however, several parameters need to be tuned.

Inspired by the simplicity and high performance of SLIC [8], in this paper, we propose a novel effective superpixel algorithm, namely Superpixels using the Shortest Gradient Distance (SSGD), which is actually an extension to SLIC. The main contributions of this paper are: 1) When the initial seeds are distributed, we apply bilateral filtering [12] to texture-rich regions centered at seeds. This operation is able to reduce the interference of non-edged texture-rich regions. 2) A new distance function taking the shortest gradient distance into account is computed during each iteration. This combined distance function prevents a pixel from being associated to a cluster if the pixel is surrounded by boundary. What's more, only two parameters need to be set in our newly proposed distance function.

We evaluated SSGD on the Berkeley Segmentation Data Set [13], and also compared it to other state-of-the-art algorithms. The results show that our approach could outperform those competitors. To make the results fully reproducible, the source code of SSGD are publicly available at http://sse.tongji.edu.cn/linzhang/ssgd/index.htm.

2. SSGD: SUPERPIXELS USING THE SHORTEST GRADIENT DISTANCE

Our approach SSGD extends SLIC to yield superpixels clinging to boundaries. In this section, we describe the details of SSGD. We first describe the SLIC framework briefly, and then, introduce the bilateral filtering operation. Next, we present the shortest gradient distance and design a distance function incorporating this new spatial distance term. After that, we adopt the strategy proposed in [9] to update the cluster centers, and finally, the overall flowchart of SSGD is given.

[*]Corresponding author: cslinzhang@tongji.edu.cn

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

Fig. 1. Image reconstruction with 300 superpixels using mean color of clusters. (a) original image, (b) SLIC[8], (c) SCALP[11], (d) SSGD without bilateral filtering operation, (e) SSGD. Regions of interest enclosed by white rectangle are provided with high resolution for zoom-in examination.

2.1. SLIC: Simple linear iterative clustering

The input image with *N* pixels is denoted by *I*. For any pixel $p \in I$, its color $c(p)$ is represented in the CIELAB color space, *i.e.,* $c(p) = [l_p,\ a_p,\ b_p]$, and the coordinates of *p* are $X_p = [x_p\ ,\ y_p]$. The iterative clustering procedure starts from an initialization of *K* cluster centers $\{C_k = [l_k, a_k, b_k, X_k]\}_{k=1}^{K}$, and the grid step is $S = \sqrt{N/K}$.

The color distance d_c and the spatial distance d_s between a pixel *p* and the cluster center C_k are defined by SLIC as:

$$d_c(C_k, p) = \sqrt{(l_p - l_k)^2 + (a_p - a_k)^2 + (b_p - b_k)^2}, \quad (1)$$

$$d_s(C_k, p) = \sqrt{(x_p - x_k)^2 + (y_p - y_k)^2}. \quad (2)$$

Then, the *5D* Euclidean distance is simply combined as:

$$D(C_k, p) = d_c(C_k, p) + d_s(C_k, p). \quad (3)$$

The cluster centers search for pixels in a $2S \times 2S$ region, which is called a searching window, and each pixel is assigned to the nearest cluster by calculating Eq. 3. After this process, the barycenters of the clusters are adjusted as the mean of all the pixels belonging to the same cluster.

2.2. Incorporating bilateral filtering

During the iterations that clusters group pixels, the texture-rich regions, such as grass, wood, fur, *etc.,* could have an negative effect on segmentation due to dramatic color variation. In our SSGD approach, we introduce the bilateral filtering operation [12] for edge-preserving smoothing to ensure a better segmentation result. In Fig. 1, we reconstruct one image from [13] with 300 superpixels using mean color of clusters. The result shows that with the bilateral filtering operation, the texture-rich regions are blurred while the edges can be protected. Bilateral filtering can be expressed as [12]:

$$h(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi, \quad (4)$$

with the normalization:

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi. \quad (5)$$

In Eq. 4, $c(\xi, x)$ and $s(f(\xi), f(x))$ are two weight functions related to the geometric closeness and the photometric similarity between a central pixel *x* and its nearby pixel ξ, respectively. We simply set these two weight functions as both Gaussian functions, i.e.,

$$c(\xi, x) = e^{-\frac{1}{2}(\frac{d(\xi, x)}{\sigma_d})^2}, \quad (6)$$

$$s(\xi, x) = e^{-\frac{1}{2}(\frac{\delta(f(\xi), f(x))}{\sigma_r})^2}, \quad (7)$$

$d(\xi, x)$ denotes the Euclidean distance between ξ and x, while $\delta(\phi, f)$ measures the color difference between ϕ and f.

In SSGD, bilateral filtering will only be applied to texture-rich regions. When the initial seeds are dispersed, we measure the texture-richness of the $2S \times 2S$ region R_i centered at the seed s_i by counting the number of different intensity values (represented by integers) in R_i. If the region's texture-richness is greater than a predefined threshold, bilateral filtering will be performed in this region.

2.3. Incorporating the shortest gradient distance

Given the coordinates of two points, SLIC uses Eq. 2 to acquire the simple spatial distance. In order to achieve better boundaries adherence, we thus propose to take the shortest gradient distance into consideration.

We start from converting the input image *I* into a gray-scale image and extracting its gradient magnitude (GM) map

I_G by $Sobel$ operator. Boundary pixels in I correspond to large GM values in I_G. Therefore, we treat the GM map as a weighted undirected graph and when a node point p_1 arrives at another adjacent node point p_2, the cost is calculated by $\frac{1}{2}I_G(p_1) + \frac{1}{2}I_G(p_2)$. Similarly, given a node point p_3 which is adjacent to p_2, the cost that p_1 goes across p_2 to p_3 can be calculated by $\frac{1}{2}I_G(p_1) + \frac{1}{2}I_G(p_2) + \frac{1}{2}I_G(p_2) + \frac{1}{2}I_G(p_3)$.

For any given point p, its shortest minimum-cost path to the cluster center C_k can be obtained using the Dijkstra shortest-path algorithm and we denote it by $SP_{C_k \to p}$. We regard the sum of the costs along the path $SP_{C_k \to p}$ as the shortest gradient distance between C_k and p and denote it by $COST(C_k, p)$. The spatial distance between p and C_k combined with the shortest gradient distance is defined as:

$$\begin{aligned} &d_{sc}(C_k, p) = \lambda \cdot d_s(C_k, p) + (1-\lambda) \cdot \eta \cdot d_s(C_k, p), \\ &\eta = \exp\left(norm(COST(C_k, p)) + \tfrac{|g(p) - g(C_k)|}{255}\right), \end{aligned} \tag{8}$$

where $\lambda \in [0, 1]$ weights the influence of the traditional Euclidean distance and λ is fixed to 0.3 in this paper. $norm(\cdot)$ is the normalization operation, making each element of the cost matrix between 0 and 1, and $\frac{|g(p)-g(C_k)|}{255}$ represents the similarity of gradient magnitudes between two points p and C_k. If C_k and p are very similar and both lie in the same flat region, then η is small. On the contrary, when they are similar and both locate on the boundary, η is medium owing to the large $norm(COST(C_k, p))$ and the small $\frac{|g(p)-g(C_k)|}{255}$. In addition, if p is surrounded by boundaries while C_k is on the smooth region, C_k must go across the boundary to reach p, so even though they are alike, η is large. High value η will prevent C_k from absorbing p into the homogeneous superpixel.

Finally, the proposed distance function $D(C_k, p)$ used in SSGD is defined as:

$$D(C_k, p) = d_c(C_k, p) + \frac{m^2}{S^2} d_{sc}(C_k, p), \tag{9}$$

where m is a compactness parameter and it is set to 15 in this paper.

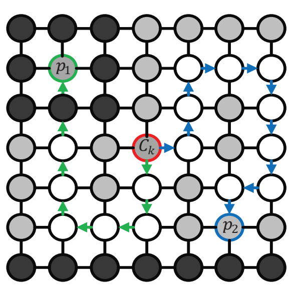


Fig. 2. The black color indicates those pixels having the largest gradient magnitude values, and the white pixels have the least gradient magnitude values.

Fig. 2 depicts a searching window in the GM map. The cluster center C_k (red border) searches the shortest path to point p_1 (green border) and point p_2 (blue border). p_1 is encompassed with edges, so C_k must consume energy to cross the large gradient point to reach p_1. Although p_1 and p_2 have the same Euclidean distance to C_k, the minimum-cost path is totally different.

Table 1. SSGD algorithm

Input: image I, expected number of superpixels K
Output: superpixel labels S
1: Place K initial seeds $\{C\}_{k=1}^{K}$.
2: Perform bilateral filtering in texture-rich regions.
3: Initialize labels of all pixels $S \leftarrow 0$.
4: Update the labels of all pixels using Eq. 9.
5: Update cluster centers using Eq. 10.
6: Repeat 4 and 5 till reach the termination condition.
7: Assign the orphaned pixels to the adjacent cluster to enforce connectivity.

2.4. Update cluster centers

We only use the most reliable pixels to update the cluster centers. The position of the l^{th} cluster center is updated using the 3-sigma rule which is proposed in [9], i.e.,

$$\begin{aligned} &C_{l,i} = \frac{\sum_{x \in \Phi_{l,i-1}} q(x)}{|\Phi_{l,i-1}|}, i \geq 1 \\ &\Phi_{l,i-1} = \{x | L(x) = l \; and \; |p(x) - C_{l,i-1}| \leq 3\xi_{l,i-1}\}, \end{aligned} \tag{10}$$

where $C_{l,i}$ is the cluster center of the l^{th} superpixel in the i^{th} iteration, $q(x)$ is distance function in Eq. 3. $p(x)$ is the intensity of pixel x in CIELAB space. And $\xi_{l,i-1}$ denotes the standard deviations of the mean color of the l^{th} cluster at the $(i-1)^{th}$ iteration. The overall flowchart of our SSGD approach is summarized in Table 1.

3. EXPERIMENTS

We compare SSGD with several state-of-the-art methods, including SLIC [8], SCALP [11], TP [7], NC [5], on BSDS500 [13]. BSDS500 consists of 500 natural images with human-annotated ground truth.

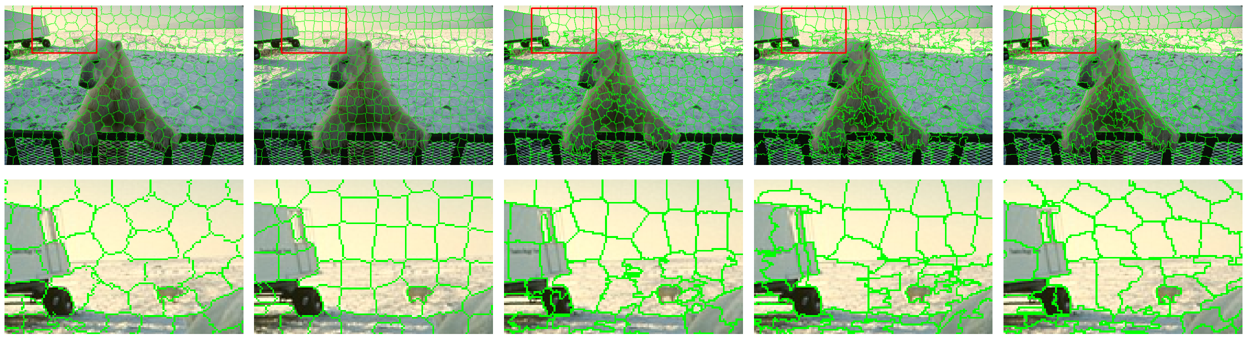
3.1. Performance metrics and parameters

We use three standard metrics to evaluate the performance of different algorithms, which are Boundary Recall (BR), Corrected Undersegmentation Error (CUE) and Achievable Segmentation Accuracy (ASA). For BR and ASA the higher the better, while for CUE the lower the better.

1) Boundary Recall (BR): BR evaluates the fraction of ground truth edges falls within a distance threshold ϵ. In our experiments, ϵ is fixed to 2.

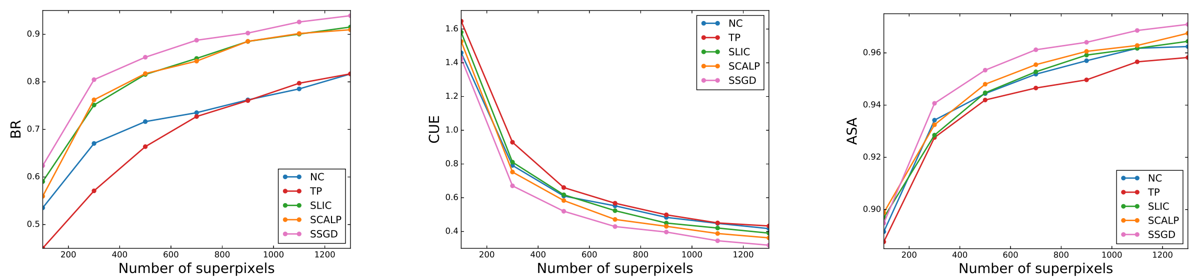
2) Corrected Undersegmentation Error (CUE): CUE is defined as[14]:

$$CUE = \frac{\sum_k |s_k - g_{max} s_k|}{\sum_i |g_i|}, \tag{11}$$



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

Fig. 3. Superpixel segmentation results on a sample image. (a) NC [5], (b) TP [7], (c) SLIC [8], (d) SCALP [11], (e) SSGD.



(a) (b) (c)

Fig. 4. Evaluation of representative algorithms and SSGD on BSDS500. (a) BR; (b) CUE; (c) ASA.

where s_k are the superpixels produced by algorithm, $g_{max}s_k$ denotes the matching ground truth segments of s_k with the largest overlap, g_i are the ground truth clusters, and $|\cdot|$ is the size of the segment.

3) Achievable Segmentation Accuracy (ASA): ASA measures the highest performance by labeling each superpixel with the label of ground truth segment with the largest overlap. ASA is defined as:

$$ASA = \frac{\sum_k max_i |s_k \cap g_i|}{\sum_i g_i}, \tag{12}$$

3.2. Results and discussion

Using a sample image, we first made a qualitative evaluation of the superpixel segmentation performance of the examined approaches and the results are shown in Fig. 3. It can be seen that NC [5] and TP [7] can generate the most compact superpixels but they perform worst in clinging to boundaries. As an improvement of SLIC, SCALP [11] enhances in boundaries detection, however, part of the skyline cannot be segmented correctly by SCALP [11]. In general, our SSGD can result in the most visually appealing results.

Then, we performed a quantitative evaluation using the curves of BRs against the number of superpixels, the curves of CUEs against the number of superpixels, and the curves of ASAs against the number of superpixels. The resultant plots are shown in Fig. 4, and we can have the following findings. Firstly, thanks to our novel distance function, higher BR can be achieved without sacrificing many regularities. Meanwhile, a vital reason contributes to this improvement is the adoption of bilateral filtering, which can effectively reduce the interference of texture-rich regions. Secondly, as shown in Fig. 4 (b), SSGD has the lowest CUE, this indicates that superpixels generated by SSGD overlap less ground truth segments. Finally, Fig. 4 (c) demonstrates that with the increasing of the numbers of superpixels, higher ASA can be obtained. Actually, no matter which performance criteria is used, SSGD always achieves the best results.

4. CONCLUSION

In this paper, a novel superpixel algorithm SSGD was proposed. SSGD is actually an effective and reasonable extension of SLIC by applying bilateral filtering to texture-rich regions and incorporating the shortest gradient distance term in the distance function used for clustering. The superiority of SSGD over the other state-of-the-art competitors was corroborated by extensive experiments. Our future work may focus on adapting SSGD to depth image inpainting.

References

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

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

[3] F. Yang, H. Lu, and M. H. Yang, "Robust superpixel tracking," *IEEE Trans. IP*, vol. 23, no. 4, pp. 1639–1651, 2014.

[4] C. Wang, Z. Liu, and S. C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, 2015.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

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

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

[8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, and P. Fua, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[9] Y. Zhang, X. Li, X. Gao, and C. Zhang, "A simple algorithm of superpixel segmentation with boundary constraint," *IEEE Trans. CSVT*, pp. 1–1, 2016.

[10] P. Wang, G. Zeng, R. Gan, and J. Wang, "Structure-sensitive superpixels via geodesic distance," *IJCV*, vol. 23, no. 1, pp. 447–454, 2013.

[11] R. Giraud, V. Ta, and N. Papadakis, "SCALP: Superpixels with contour adherence using linear path," in *ICPR*, 2016.

[12] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

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

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