



A graph based superpixel generation algorithm

Xiang Wu¹ · Xianhui Liu¹ · Yufei Chen¹ · Jianan Shen¹ · Weidong Zhao¹

Published online: 29 June 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

In recent years, superpixels have become a prevailing tool in computer vision and many methods have been proposed. However, due to the problems such as high time complexity, low object boundary adherence and irregular shape, only a few methods are widely used. To improve these issues, we propose a novel general superpixel segmentation method called minstpixel, which relies on energy functional minimization. Minstpixel introduces an energy functional based on minimal spanning tree and designs a strategy to gain the global optimum. It never needs sophisticated optimization scheme, complicated mathematical deduction or fussy iteration process. At the same time, the time complexity of minstpixel is approximately linear with respect to the number of image pixels. The benchmark on Berkeley segmentation database shows that minstpixel could rival state-of-the-art in every aspect.

Keywords Superpixels · Energy minimization · Minimization spanning tree

1 Introduction

Superpixels are usually defined as perceptually meaningful atomic regions [1] that can not only effectively capture image features but also greatly reduce the number of entities to be processed in subsequent image processing tasks, such as image segmentation, saliency detection [2, 3], contour closure [4], sketches extraction [5], object location [6], object tracking [7], 3D reconstruction [8], and many others [9].

According to previous work [10–12], we sum up five important properties of superpixel.

1. **Connectivity:** Each superpixel is an individual connected region, all of the superpixels constitute the whole image, and superpixels do not overlap each other.
2. **Homogeneity:** Pixels in the same superpixel should present same or similar visual features, especially in color and texture.
3. **Adhesion:** The boundary of superpixels should strongly adhere to the boundary of objects in an image. In other words, the set of objects' boundary is a proper subset of the set of superpixels' boundary.

4. **Regularity:** The regular arrangement, similar size, and the same shape are necessary, which provides more convenience for subsequent analysis. In the past, regularity is always ignored and devalued.
5. **Complexity:** Algorithm complexity includes time complexity and space complexity. As an image preprocessing technique, linear time complexity and linear space complexity are necessary.

Among the above properties, connectivity is the essential qualification for image segmentation; homogeneity and adhesion are the premise and foundation for the accuracy of subsequent image analysis tasks; regularity is a convenience to application developers and helps reduce the trouble of development. Unfortunately, adhesion and regularity are a pair of contradictions, while homogeneity and efficiency are another pair of contradictions. Most superpixel segmentation methods attempt to strike a balance among these contradictions. Besides these four properties, computation efficiency is also an important qualification obviously. Superpixels are typically used to accelerate complex image understanding and analysis process. Therefore, the superpixel generation algorithm itself should be simple, straightforward to implement and of linear complexity, otherwise it will not be useful.

Each superpixel segmentation method is a compromise, and no method is perfect. For example, ERS is good at adhesion but weak in regularity, while SLIC is good at regularity but weak in adhesion [13–15]. Minstpixel devotes

✉ Xianhui Liu
Liu_xian_hui@126.com
Xiang Wu
saddock@139.com

¹ College of Electronic and Information Engineering,
Tongji University, Shanghai, China

itself to strike a balance among the five properties. To meet this goal, we regard the superpixel segmentation problem as an energy minimization problem then design an energy functional based on the minimal spanning tree, and finally develop a greedy optimization strategy to optimize it. Unlike previous work, minstpixel never needs sophisticated optimization schemes, complicated mathematical deductions or a fussy iteration process. In benchmark, we implement minstpixel with approximately linear time complexity, which has achieved state-of-the-art performance on homogeneity, adhesion, regularity, and efficiency. Additionally, the [Appendix](#) mathematically proves that the greedy optimization strategy can exactly achieve the optimal value of the proposed energy functional. The drawback of minstpixel we have noted is that an individual pixel has the potential to be an independent superpixel, especially when there are too many superpixels. To overcome the drawback minstpixel merges the individual pixel into neighboring superpixels in a post-processing. Both theoretical analysis and experimental results demonstrate that minstpixel is effective and practical (Fig. 1).

2 Related works

Scientists have developed more than twenty distinctive superpixel generation methods. We selectively pick out seven methods for the contrastive experiments, and each chosen method is recognized as an excellent one [10]. They are SLIC(2012) [10], Ncut(2000) [16], ERS(2012) [17], Quick shift(2008) [18], Watersheds(1991) [19], Waterpixel(2015) [20], SEEDS(2012) [21], LSC(2017) [22], Manifold SLIC(2016) [1] and FH(2004) [23]. We

will make a comparison between minstpixel and these seven methods in the experiment section of this article. Watershed transformation [19] proposed by Vincent and Soille is a pioneer of superpixel segmentation algorithms. It automatically seeks minimal gradient pixels as seed points and gradually extends these points to superpixels. The latest improved watershed transformation is waterpixel which achieves a tunable tradeoff between the superpixel regularity and the adherence to object boundary. Simple linear iterative clustering (SLIC) clusters pixels of input image to k superpixels using modified K-means algorithm. Due to its simplicity and high performance, SLIC is widely used. LSC and MSLIC are the last improved scheme of SLIC. MSLIC extends SLIC to content sensitive superpixels which means small superpixels are in dense regions and large superpixels are in sparse regions [1]. LSC further enhances the performance of SLIC using the conclusion that the objective functions of the weighted K-means and the normalized cuts share the same optimum points by mapping each point to specific feature space [22]. To generate superpixels, Ncut [16] recursively partitions a given graph by minimizing a cost function defined on the edges between partition boundary [10]. Mean-shift [24] and Quick-shift [18] are mode-seeking algorithms that generate superpixels by recursively moving to the kernel smoothed centroid for every data point in the pixel feature space [10]. SEEDS [21] extracts superpixels by minimizing a cost function defined by color distribution term and boundary term using hill-climbing method. Except the chosen methods above, ERGC [25], ETPS [26], Vcell [27], VCCS [28], PF [29], TPS [30], DASP [31], superpixel lattice [12] and Turbopixels [32] are meaningful methods as well.

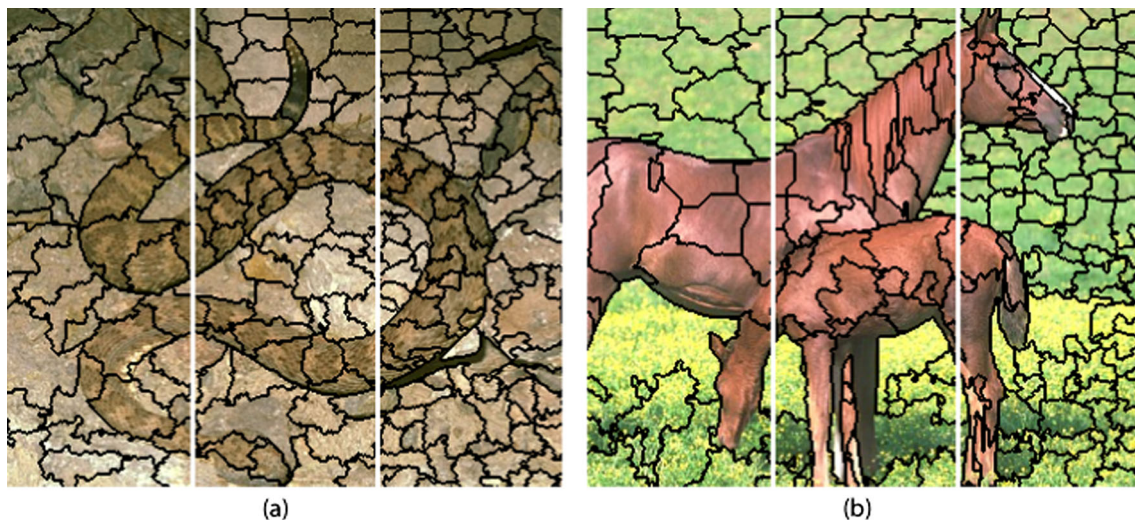


Fig. 1 Superpixel illustration. The original image comes from the famous Berkeley segmentation database. From left to right each superpixel contains about 1000, 500, 250 pixels respectively

3 Minstpixel

This section provides a detailed description of minstpixel. In this section, we first show a method to represent a superpixel via minimal spanning tree (MST), next propose an approach to construct energy functional based on MST, then introduce a strategy to get the global optimal solution of the energy functional, and finally analyze the time complexity.

3.1 Requisite knowledge

Graph representation An undirected graph is a tuple $G(\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} are vertex set and edge set, respectively. In edge set \mathcal{E} , each edge $e(v_i, v_j)$ consists of a pair of vertices v_i, v_j and a weight $w(v_i, v_j)$.

Connected component A connected component is a subgraph in which any two vertices are connected to each other by a path which only consists of vertices in this subgraph. Besides, vertices in two distinct subgraphs are not connected. Any undirected graph consists of one connected component at least.

Minimal spanning tree A spanning tree T of an undirected graph G is a tree that includes all of the vertices of G . A minimal spanning tree (MST) of a graph is a spanning tree whose weight (the sum of the weights of its edges) is no larger than any other spanning tree. Figure 2 shows

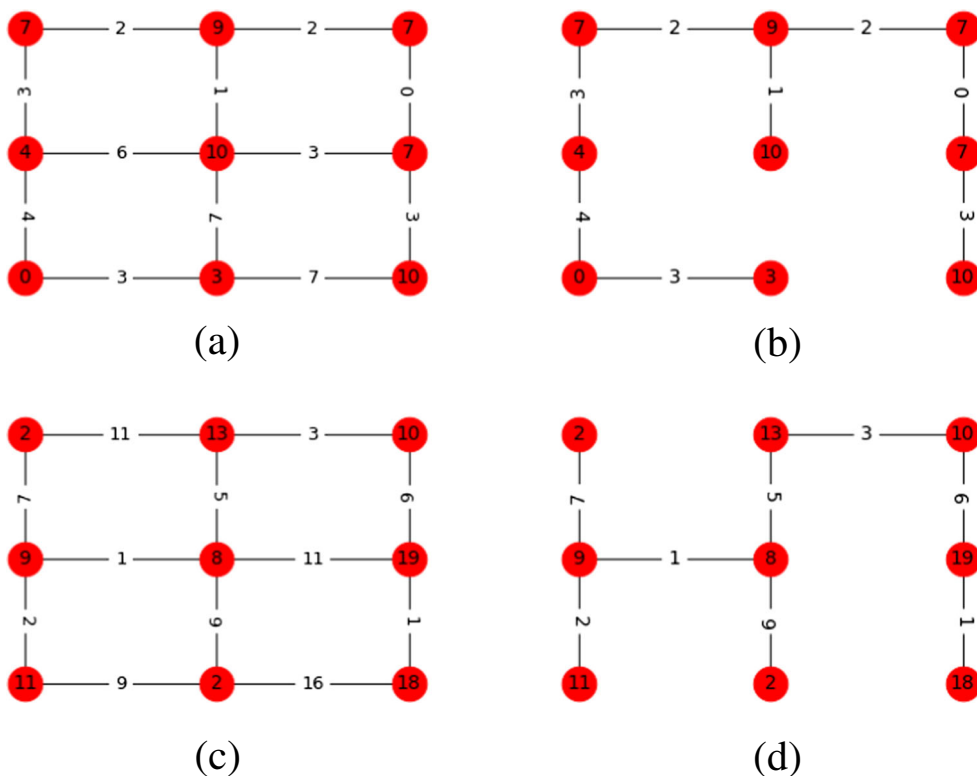
an example of minimal spanning tree. Figure 2a and c are undirected connected graphs where each vertex has a pixel value and each edge has a weight representing the difference between two adjacent vertices. Figure 2b and d are the minimal spanning trees of Fig. 2a and c respectively. As we can see, the weight of minimal spanning tree in Fig. 2c is smaller than that in Fig. 2d. This means vertices in Fig. 2a are more homogeneous than these in Fig. 2c.

Graph partition A graph partition \mathcal{S} refers to a division of the vertex set \mathcal{V} into disjoint connected components $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$ such that $S_i \cap S_j = \emptyset$ for $i \neq j$ and $\bigcup_i S_i = \mathcal{V}$. In other words, the graph partition aims to turn a graph $G(\mathcal{V}, \mathcal{E})$ into a new graph $G(\mathcal{V}, \mathcal{A})$ which consists of k connected components by removing some edges from edge set \mathcal{E} . It can be restated as a subset selection problem. The goal of subset selection problem is to select a subset of edges $\mathcal{A} \in \mathcal{E}$ such that the resulting graph $(\mathcal{V}, \mathcal{A})$ consists of k connected components.

3.2 Construct graph

We now associate a weighted undirected graph $(\mathcal{V}, \mathcal{E})$ with an image I . Each vertex $v_i \in \mathcal{V}$ corresponds to one pixel $I(x_i, y_i)$ in image I . If pixel $I(x_i, y_i)$ and pixel $I(x_j, y_j)$ are neighbors in image I , there is an edge $(v_i, v_j) \in \mathcal{E}$ between vertex v_i and vertex v_j . Weight $w(v_i, v_j)$ is a non-negative measure of the distance between pixel $I(x_i, y_i)$ and

Fig. 2 minimal spanning tree



$I(x_j, y_j)$, usually based on the differences in intensity, color, texture or some other local attributes. In our experiment, distance between pixels is determined in CIElab color space which is widely considered as perceptually uniform for small color distances [10], as shown in (1). From the viewpoint of graph theory, superpixels generation problem can be restated as the graph partition problem. In graph partition problem, people manage to divide graph $(\mathcal{V}, \mathcal{E})$ into a group of smaller separated connected components, all the connected components form a new graph $(\mathcal{V}, \mathcal{A})$. The graph $(\mathcal{V}, \mathcal{A})$ corresponds to a segmentation \mathcal{S} , and each connected component c_i corresponds to a superpixel \mathbf{s}_i in segmentation \mathcal{S} . There are various criteria to describe the quality of a segmentation but almost all criteria support that the pixels in the same superpixel should share as many of the same features as possible, while the pixels in the distinct superpixels should share as less of the same features as possible. This means that the lower the weight of edges in \mathcal{A} and the higher weight of edges in $\mathcal{E} - \mathcal{A}$, the better the segmentation result. Any partition of a graph can be determined by the edge set \mathcal{A} or $\mathcal{E} - \mathcal{A}$. So, there are two ways to get the partition. One method, such as Ncut, is to manage to get the set $\mathcal{E} - \mathcal{A}$, that is to say, some chosen edges will be removed from the graph $(\mathcal{V}, \mathcal{E})$ [33]. The other method, such as minstpixel, is to manage to get the edge set \mathcal{A} .

$$w(v_i, v_j) = \sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \quad (1)$$

3.3 Energy functional

From the viewpoint of graph theory, minstpixel regards a superpixel as an individual connected component and defines the internal similarity measure as the weight of minimal spanning tree. It is

$$D(\mathbf{s}_i) = \sum_{e \in MST(\mathbf{s}_i)} w(e) \quad (2)$$

In principle, this measure is problematic since it depends on the size of c_i . It leads to that $D(c_i)$ can not compare to each other directly because their size may be unequal. However, we have found this measure works quite well in practice because minstpixel tends to generate superpixels with little differences in size. Although some definitions are more robust to outliers, they make the optimization strategy NP hard.

For a graph $G(\mathcal{V}, \mathcal{A})$ (or corresponding superpixel segmentation \mathcal{S}), we define the similarity measure as the sum of all superpixels. It is,

$$E(\mathcal{S}) = \sum_{\mathbf{s}_i \in \mathcal{S}} D(\mathbf{s}_i) = \sum_{\mathbf{s}_i \in \mathcal{S}} \sum_{e \in MST(\mathbf{s}_i)} w(e) \quad (3)$$

3.4 Constraint condition

Energy functional (3) has at least two obvious defects.

- It can not provide any way to control the number of superpixels explicitly.
- It can not allow us to explicitly adjust the distribution of superpixels.

In order to ameliorate these defects, we define a seed point set \mathcal{P} containing k seed points. Meanwhile, we impose that each superpixel must have one and only one seed point. If seed points are evenly distributed across the whole image, the superpixels will be distributed across the whole image evenly. More formally,

$$E(\mathcal{S}) = \sum_{c_i \in \mathcal{S}} \sum_{e \in MST(c_i, E)} w(e) \quad wst. \quad (4)$$

$$\forall i, \quad | \mathbf{s}_i \cup \mathcal{P} | = 1$$

$$\forall i \neq j, \quad \mathbf{s}_i \cup \mathcal{P} \neq \mathbf{s}_j \cup \mathcal{P}$$

Obviously, the minimal value of (4) corresponds to the optimal superpixel segmentation.

3.5 Initial seed points placement

In minstpixel each superpixel grows up from a seed pixel, thus it is important to start from a good initial seed point set. SLIC [10], Turbo Pixels [32] and some other methods use regular grid points as seed points, but minstpixel considers that the center of each homogeneous region is the optimum position for seed. This seed point set makes the initial energy lower than regular grid points, so minstpixel can achieve better segmentation result.

To locate these points heuristically, minstpixel employs (6) to describe edge distribution because it performs well when searching for edges of the image [34].

$$E(x, y) = \frac{\|\nabla I\|}{G_{\sigma_1} * \|\nabla I\| + \gamma} \quad (5)$$

$$D(x, y) = \frac{1}{Z} (G_{\sigma_2} * (E(x, y) + \lambda \overline{E(x, y)})) \quad (6)$$

Eq.5 describes the calculation of the normalized gradient, where G_{σ} , $\|\nabla I\|$ and γ are the Gaussian function with standard deviation σ , the gradient magnitude of color image I and the parameter to reduce the noise disturbance respectively. In (6), the $\overline{E(x, y)}$ is the mean of $E(x, y)$, and the Z is the regularized factor making sure that the integral of $D(x, y)$ is 1. $D(x, y)$ can be regarded as a probability that (x, y) lies on edges of objects.

Furthermore, we suppose that point set $\mathcal{P}' = \{(x', y')\}$ contains k grid points which have the same span $\sqrt{m/k}$, where m is the total pixel number of the image and k is the number of superpixels. Our initialization method

can be represented as a map $\mathcal{P}' \rightarrow \mathcal{P}$. Equation (7a,b) quantitatively describes the relationship between \mathcal{P}' and \mathcal{P} .

$$\int_0^{x_i} D(x, y)dx = x'_i \times \frac{\int_0^w D(x, y)dx}{w} \tag{7a}$$

$$\int_0^{y_i} D(x, y)dy = y'_i \times \frac{\int_0^h D(x, y)dy}{h} \tag{7b}$$

From (7a), x_i and y are in one-to-one relationship, then (7a) is a curve extending in direction y . Similarly, (7b) is a curve extending in direction x . The intersection of the two curves is the seed point (x_i, y_i) . If there are multiple points of intersection, minstpixel will choose the one with the lowest $D(x, y)$.

3.6 Energy minimization

Mathematically, the problem that we are facing is a typical combination optimal problem. It's almost impossible to find any analytical solution with current technologies. Instead, we craft a greedy method to gain reasonable superpixel segmentation. Our method begins by sampling k seed pixels and then gradually extend to all pixels based on a greedy rule. Then, we have proved that the optimum point can be reached by this method using mathematical induction in the Appendix.

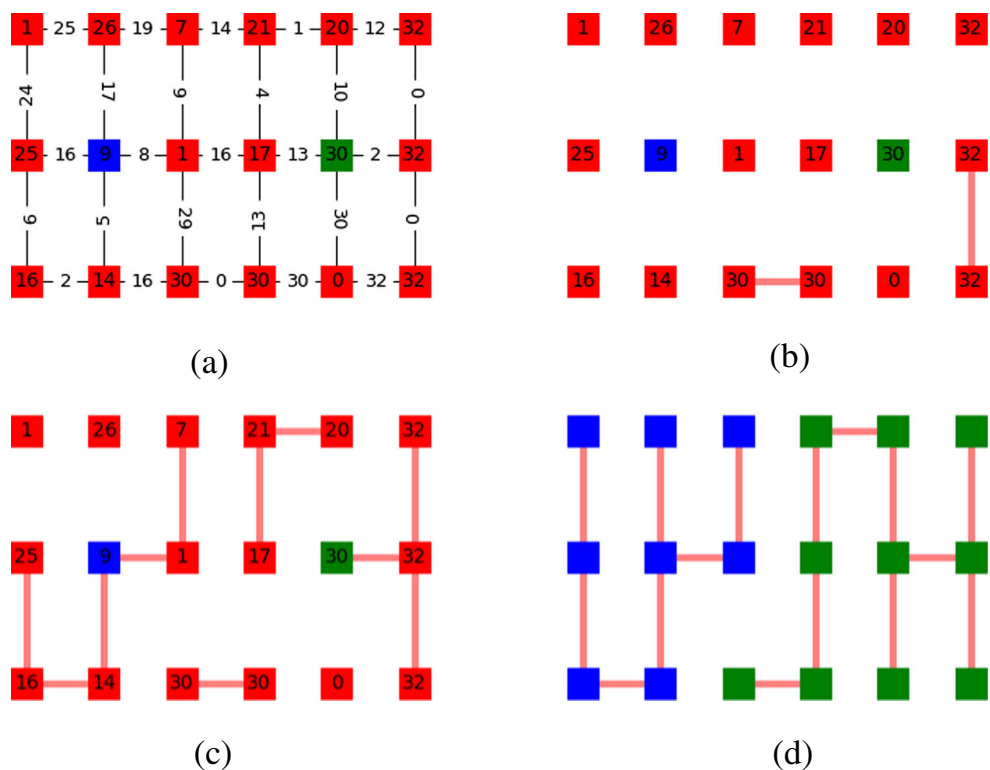
Algorithm 1

The input is a graph $G(\mathcal{V}, \mathcal{E})$ with n vertices, m edges and a seed point set $\mathcal{P} = \{(x_i, y_i)\}$. The output is a segmentation of \mathcal{V} with k components in $\mathcal{S} = (c_1, \dots, c_k)$.

1. Sort \mathcal{E} into queue $\pi = (e_1, \dots, e_m)$, by non-decreasing edge weight.
2. Start with a segmentation S^0 , where each vertex v_i is in its own component.
3. Given S^{q-1} construct S^q . Let v_i and v_j denote the vertices connected by the first edge in the queue, c_i^{q-1} and c_j^{q-1} denote the connected components containing v_i and v_j respectively. If c_i^{q-1} and c_j^{q-1} are disjoint components of S^{q-1} and neither c_i^{q-1} or c_j^{q-1} contains seed points, then merge the two components. More formally, if $(c_i^{q-1} \neq c_j^{q-1}) \cap ((c_i^{q-1} \cap \mathcal{P} = \emptyset) \cup (c_j^{q-1} \cap \mathcal{P} = \emptyset))$ is true, then S^q is obtained from S^{q-1} by merging c_i^{q-1} and c_j^{q-1} , otherwise $S^q = S^{q-1}$.
4. Repeat step 3 until S^q remains k independent connected components i.e. $|S^q| = k$.
5. Return $\mathcal{S} = S^q$.

We now have established a segmentation S obeying energy functional (4). Figure 3 shows the complete process of algorithm 1. It is inescapably clear that the result graph includes m vertices, $m - k$ edges and k connected components, and each component corresponds

Fig. 3 An example of algorithm 1. **a** Each vertex corresponds to a pixel of input image. The figures on the vertex are pixel values. The figures on the edges are the distances between neighboring pixels. Blue vertex and green vertex are both seed points. **b** After two iterations, two edges are chosen. **c** After ten iterations, ten edges are chosen. **d** The graph is divided into two components. Each component is a minimal spanning tree and each minimal spanning tree corresponds to a superpixel



to a superpixel. In the sequel we show that algorithm 1 converges exactly to the global optimal solution of (4), although it makes greedy decisions in each iteration.

3.7 Implementation and complexity

Algorithm 1 maintains the segmentation \mathcal{S} using a disjoint set with path compression and union by rank. This disjoint set can finish n merge operations and m search operations in $O(m\alpha(n))$ time, where $m\alpha(n)$ is inverse Ackerman's function which is an approximate linear function [35]. Step 1 sorts m weights into nondecreasing order. In general, it can consume $O(m \log m)$ time using Quick Sort [35]. Step 2 and Step 3 find and merge eligible components. There are $m - k$ merge operations and up to m search operations, so it consumes $O((m - k)\alpha(m))$ time. All in all, the time complexity of algorithm 1 is $O(m \log m + (m - k)\alpha(m))$.

4 Experiments

We compare minstpixel with ten state-of-the-art algorithms, namely SLIC(2012) [10], Ncut(2000) [16], ERS(2012)

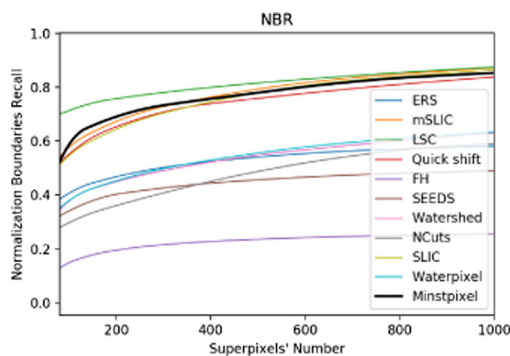
[17], Quick shift(2008) [18], Watersheds(1991) [19], Waterpixel(2015) [20], SEEDS(2012) [21], LSC(2017) [22], Manifold SLIC(2016) [1] and FH(2004) [23], using publicly available source codes on the Berkeley Segmentation Database [36]. All benchmarks are conducted on a desktop PC equipped with two intel xeon E5-2620 v4 processors and 16GB of memory.

4.1 Adhesion

The adhesion refers to the ability to adhere object boundary. Superpixels generally serve as the foundation of object detection, so adhesion is a key evaluation criterion. In order to evaluate adhesion, predecessors propose boundary recall against the number of superpixels (BR). BR is the fraction of ground truth boundary that fall in the superpixel boundary as (8). A ground truth boundary pixel is counted if it falls within strictly less than 2 pixels from superpixel boundary.

$$BR = \frac{TP}{TP + FN} \quad (8)$$

Where TP (True Positives) is the number of boundary pixels that fall in the superpixel boundary, TN is the number of



(a)

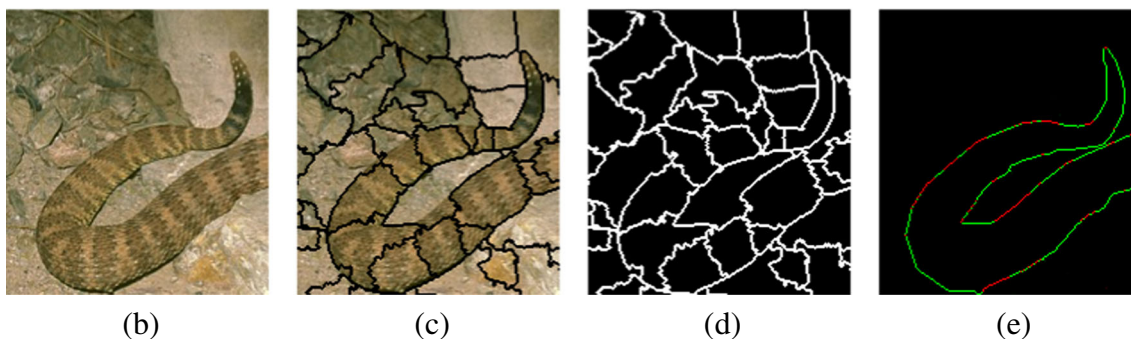


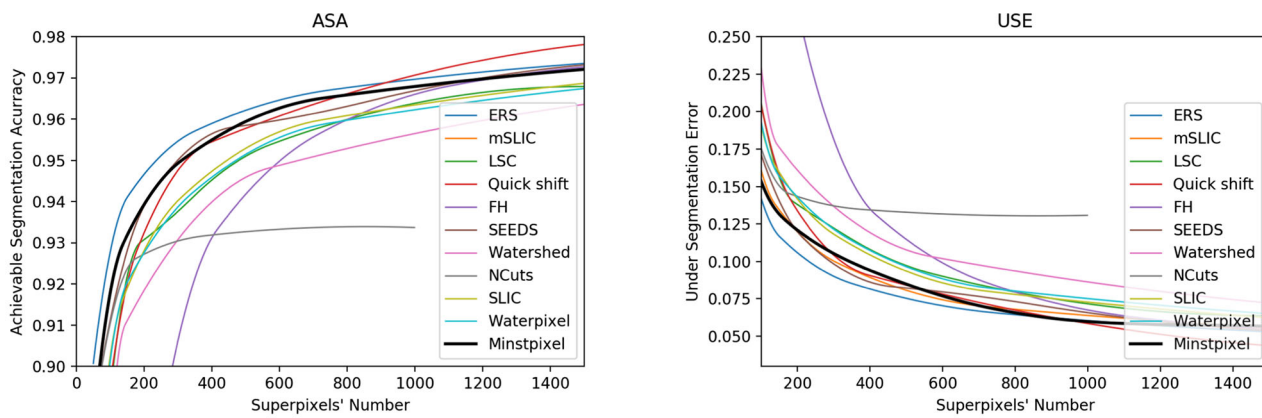
Fig. 4 Illustrations of boundary recall benchmark. **a** shows the comparison between minstpixel and eight fine methods, and minstpixel reaches upper-middle level. **b–e** demonstrate the boundary missed by minstpixel. **b** is an image slice from the Berkeley segmentation database in which the color and texture are very similar between foreground and background, so it's very difficult to accurately distinguish

objects from background. **c** and **d** are minstpixel and the boundary recalled by minstpixel respectively, and boundaries are marked out by double pixels. **e** shows groundtruth marked out by single pixel, green parts represent the boundary minstpixel hit and red parts represent the boundary minstpixel miss

boundary pixels that fail to fall in the superpixel boundary. A high BR indicates that the majority of true boundary are preserved. However some previous algorithms also achieved high BR in benchmark, because these algorithms tend to generate tortuous boundary which result in high boundary density. For example, boundary density of FH significantly exceeds minstpixel, so FH has more chances to hit true boundary than minstpixel, as shown in Fig. 6. It is unfair to the method tending to generate smooth boundary. Machairas and Kalinin have noticed this problem and propose boundary density against boundary recall to evaluate adhesion of superpixels [11]. However, boundary density against boundary recall still ignores the influence of the quantity of superpixels. In order to take both the quantity of superpixels and the density of boundary into account, we define normalization boundary recall as $NBR = BR/C$, by introducing the penalty factor C which is proportional to boundary density. Figure 4a demonstrates the results of benchmark.

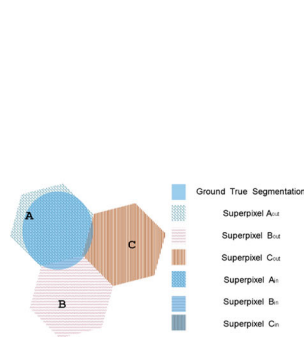
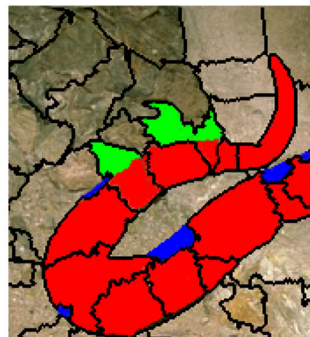
4.2 Homogeneity

Homogeneity refers to the similarity of the pixels from the same superpixel. Superior homogeneity means that a superpixel can only cover one object. The homogeneity benchmark consists of two common evaluation criteria: under segmentation error (USE) and achievable segmentation accuracy (ASA). USE measures what extent superpixels stretch over the ground truth segment boundary. A lower USE indicates that fewer superpixels cross the boundary of objects. USE is defined as $\left[\sum_S \left(\sum_{P: P \cap S \neq \emptyset} \min(|P_{in}|, |P_{out}|) \right) \right] / N$, where N is the number of pixels in image, $|P_{in}|$ is the number of pixels in part P_{in} and $|P_{out}|$ is the number of pixels in part P_{out} . A factitious case of USE is shown in Fig. 5f, in this case N is $(|A_{out}| + |B_{in}| + |C_{in}|) / |S|$. ASA measures the upper bound of accuracy when image objects are reconstructed using a given set of superpixels as units. Given a ground truth segmentation G and a



(a)

(b)



(c)

(d)

(e)

(f)

Fig. 5 Homogeneity illustrations. **a** Achievable segmentation accuracy. **b** Under segmentation error. **c** Original image. **d** Minstpixel. **e** Object detection by minstpixel. **a** and **b** are line charts smoothed by second order B spline function for the sake of beauty, and it can be seen

from these charts that minstpixel reaches the top three in these two tests. **c–e** show an example of object detection by minstpixel. Red represents right detection pixels, blue represents missed detection pixels and green represents over detection pixels

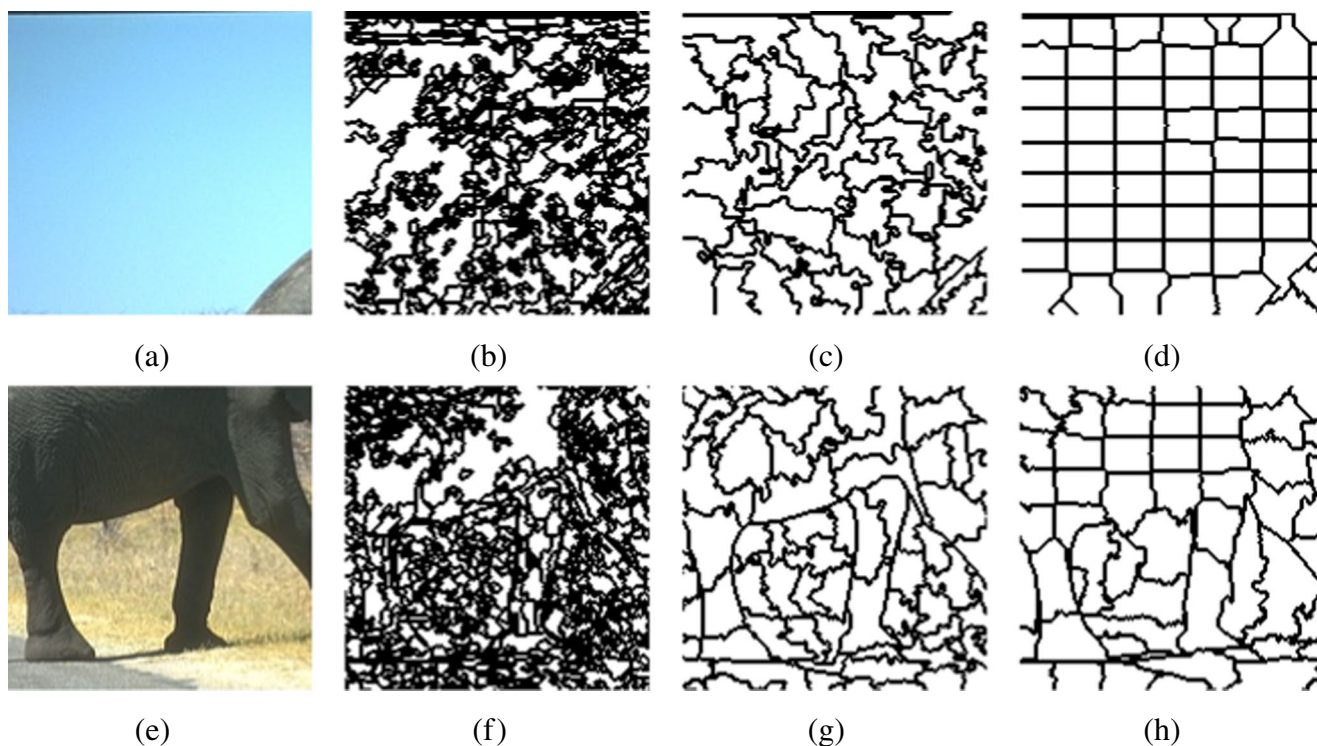


Fig. 6 The shape of superpixels. **a,e**: Origin image. **b,f**: FH. **c,g**: ERS. **d,h** Minstpixel. **a,e** are both 150×150 image slices from BSD500. **a** comes from background region while **(e)** comes from foreground region. It can be seen easily that the shape and boundary density of

distinct superpixels are various; for instance, the boundary of FH is very twisted, while the boundary of ERS is smooth. Minstpixel has regular shape and well maintains the border of objects in input image

superpixel segmentation S , achievable segmentation accuracy is defined as $(\sum_k \max_i |S_k \cap G_i|) / \sum_i |G_i|$.

4.3 Regularity

The regularity is always devalued but it is an important indicator. On the one hand, irregular boundary will contribute to the false high performance in benchmark.

On the other hand, irregular shape will cause trouble for subsequent image analysis (Fig. 6).

The regularity contains two items [20]: (1) the smoothness of boundary (2) the shape and size similarity of superpixels. The smoothness is evaluated by the boundary density which is defined as a ratio of the number of superpixel boundary pixels to the total number of pixels in the image. The shape and size similarity is evaluated by average

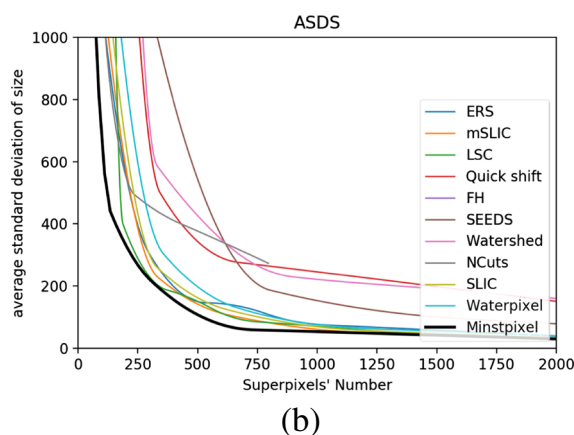
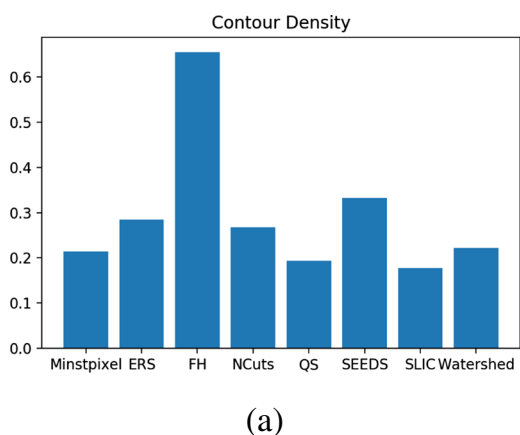


Fig. 7 Benchmark: contour density and average standard deviation of size. **a**: a bar graph of average contour density on Berkeley segmentation database and each method is set to generate 400 superpixels. **d**: average standard deviation of size against the number of superpixels

standard deviation of size. Figure 7 shows the benchmark result in regularity.

4.4 Efficiency

Obviously, time complexity is also a crucial evaluation criterion for superpixel segmentation algorithm. Table 1 and Fig. 8 shows the theoretical time complexity and actual run time on the Berkeley database.

4.5 Application

In this section, we display one typical computer vision application that benefit from superpixels: object segmentation. Minstpixel reduces the number of entities to manipulate and improve the result of existing algorithm. The task is to recognize 21 classes of objects from MSRC database using Ada-Boost classifiers and the appearance features are based on the work of Gould et al [9]. For each superpixel s_i , we construct an 83 dimensions feature vector $\phi(s_i)$ to describe region's size, location, color, shape and texture feature.

It specifically includes:

- Size and location (1 + 2)
- The ratio of area to circumference and inertia moment (1 + 2)
- The mean, standard deviation, skewness and kurtosis statistics of RGB and Lab color space components (4 × (3 + 3))
- The mean, standard deviation, skewness and kurtosis statistics of texture feature drawn from 13 filters responses (4 × 13)

In addition, we add the weighted mean of adjacent superpixels' feature vector to each superpixel as (9),

$$\frac{\sum_{S_j \in \mathcal{N}(S_j)} |S_j| \cdot \phi(S_j)}{\sum_{S_j \in \mathcal{N}(s_i)} |S_j|} \tag{9}$$

where $\mathcal{N}(s_i) = \{S_j | (s_i, S_j) \in \mathcal{E}(\mathcal{I})\}$ is the set of superpixels which are adjacent to s_i and $|S_j|$ is the area of superpixel S_j . We train a one-vs-all Ada-Boost classifier $\sigma_{c'}$ (c' is the class of object) for each object and then normalize

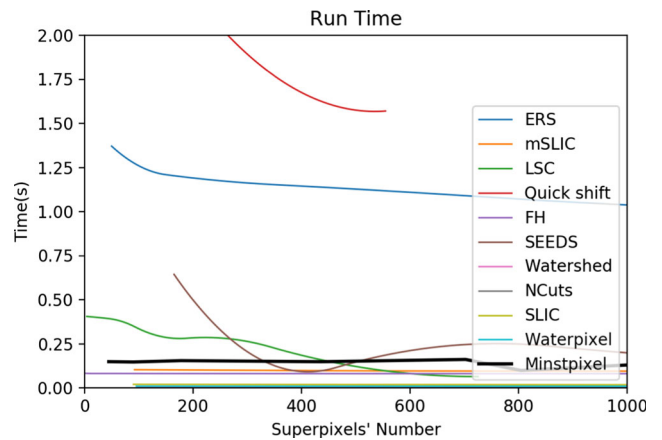


Fig. 8 Time consumption per image of eleven superpixel segmentation algorithms against number of superpixels on Berkeley image segmentation database. Minstpixel takes less than 0.5s at any number of superpixels

over all classifiers to get the probability $P(c_i = c' | s_i, \mathcal{I})$ as (10)

$$P(c_i = c' | s_i, \mathcal{I}) = \frac{\exp\{\sigma_{c'}(s_i)\}}{\sum_c \exp\{\sigma_c(s_i)\}} \tag{10}$$

where σ_c is the Ada-Boost classifier for class c . Finally, let $c_i = \arg \max_{c'} P(c_i = c' | s_i, \mathcal{I})$, then c_i is the class of superpixel s_i .

Compared with the original method, the accuracy is increased by 1.8% and the running time is reduced by 99%, as show in Table 2.

4.6 Discussion and analysis

We can draw many meaningful conclusions from experiment results. Firstly, none of superpixel segmentation algorithms can really defeat all other algorithms comprehensively. Each algorithm is a compromise between performance and time consumption. The increase in performance is usually accompanied by decrease in efficiency or other properties. When performance reaches a certain level, time consumption will increase exponentially. For instance, ERS leads minstpixel by 1% in performance but ERS spends 10 times than minstpixel. Secondly, whether a superpixel is

Table 1 Comparison of runtime among state-of-art superpixels

Superpixel	ERS	SLIC	Quick Shift	FH	SEEDS	Watershed	Ncuts	Minstpixel
Complexity	$O(m^2 \log m)$	$O(m)$	$O(m^2)$	$O(m \log m)$	$O(m)$	$O(m)$	$O(m^{\frac{2}{3}})$	$O(m \log m)$
Average time	1.37s	0.361s	4.67s	0.103s	0.526s	0.0167s	105.3s	0.205s

The numbers in first line show the theoretical computational complexity and m is the number of pixels in the image. The second line is the actual average time per image on Berkeley image segmentation database for $k = 400$

Table 2 Object class recognition for various superpixel methods

	ERS	SLIC	Quick Shift	FH	SEEDS	Watershed	Origin	Minstpixel
Accuracy	79.8%	77.1%	62.0%	74.6%	72.%	60.3%	75.5%	77.3%
Average time	3.06s	1.05s	5.59s	0.912s	1.23s	0.727s	106s	0.915s

good or bad should be judged based on a specific scene. For example, ERS is good at dealing with the demand for high accuracy, SLIC is good at dealing with the demand for regular shape and WS is skilled at dealing with real time scenes. Finally, all things considered, minstpixel has reached the level of state-of-art algorithms, although minstpixel can not comprehensively defect all other algorithms.

5 Conclusions

We have presented a novel superpixel segmentation algorithm called minstpixel, which produces compact and regular shaped superpixels with high accuracy, low time complexity and simple implementation. To generate high quality superpixels, minstpixel firstly proposes an energy functional based on minimal spanning tree and the initialization strategy. Then, minstpixel optimizes the energy functional by picking out the specific edge at each iteration. Additionally, we prove that the optimization algorithm can produce exactly the global optimum of the energy functional. Both theory and practice suggest that minstpixel is reasonable and practicable. Certainly, minstpixel is not perfect and may be improved by multi-scale mutual information [37–39].

- Similarity measure for single superpixel $D(s_i)$ in energy functional (4) depends on the size of superpixel. It causes the superpixel size to be nonuniform. Usually it isn't a serious implication, since the constraint condition that each superpixel has one and only one seed point to fill this gap to a certain extent.
- Energy functional (4) has multiple optimal solutions but algorithm 1 can only find one. It means that there may be some better solutions which are neglected by algorithm 1.

Acknowledgments This work was supported by the National High Technology Research, Development Program of China (No.2017YFB1401600), the National Natural Science Foundation of China (No.61573235,61703260), the Shanghai Innovation Action Project of Science and Technology (No.16111105802), and the Fundamental Research Funds for the Central Universities.

Appendix

This section prepares an proof for the global optimality of the algorithm 1. Proposition 1 and proposition 2 demonstrate

that the segmentation \mathcal{S} generated by algorithm 1 is a valid superpixel segmentation. Proposition 3 demonstrates that algorithm 1 will always converge to the optimum of (4).

Proposition 1 *There is one and only one seed point in each connected component produced by algorithm 1.*

Proof On the one hand, the number of seed points in each component is less than 2 (1 or 0) according to the step 3 of algorithm 1. On the other hand, we suppose that in segmentation \mathcal{S} there exists one connected component c_i without seed points, then the number of connected components of segmentation is greater than k . There is a contradiction between assumption and termination condition of algorithm 1. So, the number of seed points in each component cannot be other except 1. \square

Proposition 2 *Each connected component produced by algorithm 1 is a minimum spanning tree.*

Proof Let $G(\mathcal{V}, \mathcal{E})$ be an input graph, $G(\mathcal{V}, \mathcal{A})$ is the segmentation result produced by algorithm 1, $c_i(\mathbf{v}_i, \mathbf{a}_i)$ is the i th connected component of graph $(\mathcal{V}, \mathcal{A})$ i.e. $\bigcup_i \mathbf{v}_i = \mathcal{V}$ and $\bigcup_i \mathbf{a}_i = \mathcal{A}$. Obviously, if we remove an edge $e \notin \mathcal{A}$ from graph $(\mathcal{V}, \mathcal{E})$, then the segmentation result $(\mathcal{V}, \mathcal{A})$ produced by algorithm 1 remains the same. Now, we build a graph $(\mathcal{V}', \mathcal{E}')$ from graph $(\mathcal{V}, \mathcal{E})$ by removing all edges $e(v_1, v_2)$ satisfying $v_1 \in c_i, v_2 \in c_j$ and $c_i \neq c_j$. Suppose that $(\mathcal{V}', \mathcal{A}')$ is the segmentation result produced by algorithm 1 on graph $(\mathcal{V}', \mathcal{E}')$, $c'_i(\mathbf{v}'_i, \mathbf{a}'_i)$ is the i th connected component of graph $(\mathcal{V}', \mathcal{A}')$ i.e. $\bigcup_i \mathbf{v}'_i = \mathcal{V}'$ and $\bigcup_i \mathbf{a}'_i = \mathcal{A}'$. Graph $(\mathcal{V}, \mathcal{A})$ and graph $(\mathcal{V}', \mathcal{A}')$ is identical because all removed edges don't belong to \mathcal{A} , and then each c_i is identical to the corresponding c'_i . On the other side, the process of algorithm 1 for graph $(\mathcal{V}', \mathcal{E}')$ is perfectly equivalent to the process of Kruskal algorithm for every c'_i [40]. So, each connected component c'_i is a minimum spanning tree, and then each c_i is also a minimum spanning tree. \square

Proposition 3 *The algorithm 1 can converge to the global minimum point.*

Proof We define proposition P : If \mathcal{A} is the set of edges chosen at any stage of the algorithm 1, then there are some optimal segmentations that contain \mathcal{A} . Clearly, proposition

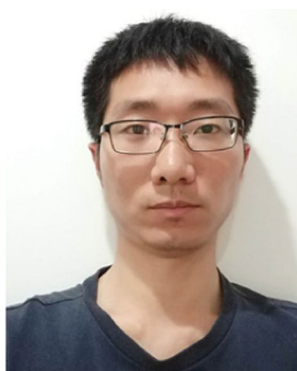
3 is equivalent to proposition P . We show that proposition P is true by induction.

1. Clearly, P is true at the beginning, when \mathcal{A} is empty: any optimal segmentation will do, and there exists one because a weighted connected graph always has some segmentation satisfying the constraint condition of energy functional (4).
2. Now assume that P is true for a non-final edge set \mathcal{A} and let \mathcal{S} be an optimal segmentation that contains \mathcal{A} . If the next chosen edge e is also in \mathcal{S} , then P is true for $\mathcal{A} + e$. Otherwise, $\mathcal{S} + e$ has a cycle C or leads to a superpixel which contains two seed points p_1 and p_2 .
 - If, $\mathcal{S} + e$ has a cycle C and there is another edge f that is in C but not \mathcal{A} . (If there were no such edge f , then e could not have been added to \mathcal{A} , since doing so would have created the cycle C .) Then $\mathcal{S} - f + e$ is a valid superpixel segmentation, and it has the same weight as \mathcal{S} , since \mathcal{S} has minimum weight and the weight of f cannot be less than the weight of e , otherwise the algorithm 1 would have chosen f instead of e . So $\mathcal{S} - f + e$ is a minimum spanning tree containing $\mathcal{A} + e$ and again, P holds.
 - If, $\mathcal{S} + e$ leads to a superpixel contains two seed points p_1 and p_2 . In other words, there is a path between p_1 and p_2 and there is another edge f that is in path $p_1 p_2$ but not \mathcal{A} . Then $\mathcal{S} - f + e$ is a valid superpixel segmentation, and it has the same weight as \mathcal{S} , since \mathcal{S} has minimum weight and the weight of f cannot be less than the weight of e , otherwise the algorithm 1 would have chosen f instead of e . So $\mathcal{S} - f + e$ is a minimum spanning tree containing $\mathcal{A} + e$ and again, P holds.
3. Therefore, by the principle of induction, P holds when \mathcal{A} has become a valid segmentation, which is possible only if \mathcal{A} is a optimal superpixel segmentation itself. □

References

1. Liu Y-J, Cheng-Chi Y, Min-Jing Y, He Y (2016) Manifold SLIC: a fast method to compute content-sensitive superpixels. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 651–659
2. Cheng M-M, Zhang G-X, Mitra NJ, Huang X, Shi-Min H (2011) Global contrast based salient region detection. In: CVPR 2011, vol 37. IEEE, pp 409–416
3. Achanta R, Hemami S, Estrada F, Susstrunk S (2009) Frequency-tuned salient region detection. In: 2009 IEEE conference on computer vision and pattern recognition, number 1c. IEEE, pp 1597–1604
4. Levishtein A, Sminchisescu C, Dickinson S (2010) Optimal contour closure by superpixel grouping. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), volume 6312 LNCS. Springer, Berlin, pp 480–493
5. Yu C-C, Liu Y-J, Wu MT, Li K-Y, Fu X (2014) A global energy optimization framework for 2.1D sketch extraction from monocular images. *Graph Models* 76(5):507–521
6. Fulkerson B, Vedaldi A, Soatto S (2009) Class segmentation and object localization with superpixel neighborhoods. In: 2009 IEEE 12th international conference on computer vision. IEEE, pp 670–677
7. Shu W, Huchuan L, Fan Y, Ming-Hsuan Y (2011) Superpixel tracking. In: 2011 IEEE international conference on computer vision (ICCV). IEEE, pp 1323–1330
8. Mičušík B, Košecká J (2010) Multi-view superpixel stereo in urban environments. *Int J Comput Vis* 89(1):106–119
9. Gould S, Rodgers J, Cohen D, Elidan G, Koller D (2008) Multi-class segmentation with relative location prior. *Int J Comput Vis* 80(3):300–316
10. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans Pattern Anal Mach Intell* 34(11):2274–2282
11. Machairas V, Decenciere E, Walter T (2014) Waterpixels: superpixels based on the watershed transformation. In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 4343–4347
12. Moore AP, Prince SJD, Warrell J, Mohammed U, Jones G (2008) Superpixel lattices. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE, pp 1–8
13. Mathieu B, Crouzil A, Puel JB (2017) Oversegmentation methods: a new evaluation. *Pattern Recogn Image Anal* 10255:185–193
14. Wang M, Liu X, Gao Y, Ma X, Soomro NQ (2017) Superpixel segmentation: a benchmark. *Signal Process Image Commun* 56:28–39
15. Neubert P, Protzel P (2012) Superpixel benchmark and comparison. In: Forum bildverarbeitung, pp 1–12
16. Jianbo S, Malik J (2000) Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell* 22(8):888–905
17. Liu M-Y, Tuzel O, Ramalingam S, Chellappa R (2011) Entropy rate superpixel segmentation. In: CVPR 2011. IEEE, pp 2097–2104
18. Vedaldi A, Soatto S (2008) Quick shift and kernel methods for mode seeking. In: Computer vision – ECCV 2008, volume 5305 LNCS. Springer, Berlin, pp 705–718
19. Vincent L, Soille P (1991) Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans Pattern Anal Mach Intell* 13(6):583–598
20. Machairas V, Faessel M, Cardenas-Pena D, Chabardes T, Walter T, Decenciere E (2015) Waterpixels. *IEEE Trans Image Process* 24(11):3707–3716
21. Van den Bergh M, Boix X, Roig G, de Capitani B, Van Gool L (2012) SEEDS: superpixels extracted via energy-driven sampling. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), volume 7578 LNCS, pp 13–26
22. Chen J, Li Z, Bo H (2017) Linear spectral clustering superpixel. *IEEE Trans Image Process*, 1–1
23. Felzenszwalb PF, Huttenlocher DP (2004) Efficient graph-based image segmentation. *Int J Comput Vis* 59(2):167–181
24. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell* 24(5):603–619
25. Buysens P, Gardin I, Ruan S (2014) Eikonal based region growing for superpixels generation: application to semi-supervised real time organ segmentation in CT images. *IRBM* 35(1):20–26

26. Yao J, Boben M, Fidler S, Urtasun R (2015) Real-time coarse-to-fine topologically preserving segmentation. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2947–2955
27. Wang J, Wang X (2012) VCells: simple and efficient superpixels using edge-weighted centroidal voronoi tessellations. *IEEE Trans Pattern Anal Mach Intell* 34(6):1241–1247
28. Papon J, Abramov A, Schoeler M, Worgotter F (2013) Voxel cloud connectivity segmentation - supervoxels for point clouds. In: 2013 IEEE conference on computer vision and pattern recognition. IEEE, pp 2027–2034
29. Drucker F, MacCormick J (2009) Fast superpixels for video analysis. In: 2009 workshop on motion and video computing (WMVC). IEEE
30. Tang D, Huazhu F, Cao X (2012) Topology preserved regular superpixel. In: 2012 IEEE international conference on multimedia and expo. IEEE, pp 765–768
31. Weikersdorfer D (2012) Depth-adaptive superpixels
32. Levinshtein A, Stere A, Kutulakos KN, Fleet DJ, Dickinson SJ, Siddiqi K (2009) TurboPixels: fast superpixels using geometric flows. *IEEE Trans Pattern Anal Mach Intell* 31(12):2290–2297
33. Ren X, Malik J (2003) Learning a classification model for segmentation. *Proc Ninth IEEE Int Conf Comput Vis* 1(c):10–17, 1
34. Wang P, Zeng G, Gan R, Wang J, Zha H (2013) Structure-sensitive superpixels via geodesic distance. *Int J Comput Vis* 103(1):1–21
35. Cormen T, Rivest R, Leiserson C (1989) *Introduction to algorithms*. MIT Press
36. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings eighth IEEE international conference on computer vision. ICCV 2001, vol 2*. IEEE Comput. Soc, pp 416–423
37. Wei W, Fan X, Song H, Fan X, Yang J (2016) Imperfect information dynamic Stackelberg game based resource allocation using hidden Markov for cloud computing. *IEEE Trans Serv Comput*, 1–1
38. Wei W, Fan X, Song H, Wang H (2017) Video tamper detection based on multi-scale mutual information. *Multimed Tools Appl*, 1–18
39. Wei W, Yang X-L, Zhou B, Feng J, Shen P-Y (2012) Combined energy minimization for image reconstruction from few views. *Math Probl Eng* 2012:1–15
40. Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc Amer Math Soc* 7(1):48



Xiang Wu is currently pursuing a PhD degree in image processing at College of Electronics and Information Engineering, Tongji University.



Xianhui Liu received the Ph.D. degree from Tongji University in 2014. He is currently an associate professor with the College of Electronic and Information Engineering of Tongji University, Shanghai, China. He was a researcher of the Computer Aided Design Center of Tongji University from 2006 to 2013. His research topics include data analysis and image processing.



Yufei Chen received the Ph.D. degree from Tongji University in 2010. She is currently an associate professor with the College of Electronic and Information Engineering of Tongji University, Shanghai, China. She was a Post-Doctoral Researcher with the Control Science and Engineering of Tongji University from 2010 to 2012. She was also a Guest Researcher with the Fraunhofer Institute for Computer Graphics Research, Germany, from 2008 to 2009. Her research topics include image processing and data analysis.



Jianan Shen is a first year Master student at Tongji University. She graduated in computer science from the College of Electronics and Information Engineering, Tongji University. Her research topic is image processing.



Weidong Zhao Ph.D supervisor. He is a member of China National Technical Committee for Industrial Automation Systems and Integration Standardization, the chief expert on Information Technology in Sci-Tech Engineering of Manufacturing Industry during the eleventh five-year project of the Ministry of Science and Technology of China, and the team leader of Information Technology Manufacturing Engineering of Shanghai Municipal Science and Technology Commission.