Journal of Computers Vol. 29 No. 5, 2018, pp. 80-93
doi:10.3966/199115992018102905008

Toward Path-based Content Perception for Superpixels Generation

Qi Wang[1], Fan Wang[1], Nan Ding[1], Xiao-Peng Hu[1*]

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian, China
wangqi@mail.dlut.edu.cn, {wangfan, dingnan, xphu}@dlut.edu.cn

Received 17 June 2017; Revised 28 October 2017; Accepted 1 December 2017

Abstract. Superpixels partition an image into homogeneous regions with regular shapes and adherence to object edges. A critical factor of superpixels generation is the distance measure of pixels. Euclidean distance is usually utilized to compute the spatial and color difference between pixels. As a pre-processing procedure, superpixels should provide image information as much as possible for further analysis. But image content is ignored as Euclidean distance can hardly deal with. In this paper, we focus on the distance computation of superpixel methods. We propose the adaptive k distance method for image content analysis. We design a computational method of adaptive k distance which is applied to images specially. Combining Euclidean distance and the proposed distance, we present a superpixel method which provides tunable parameters to compromise between superpixel compactness and image edge adherence. We evaluate our method on BSD500 dataset comparing with several state of the art superpixel methods. We discuss the experimental results qualitatively and quantitatively, and the referred evaluation criteria that have not been analyzed in detail. Experimental results show that our method achieves favorable performance against the state of the art.

Keywords: minimum spanning tree, path analysis, superpixel

1 Introduction

Superpixels aim to provide an over-segmentation of images, which is usually used as a preprocessing procedure [1]. In human vision system, people tends to group elements together if they are proximal, similar or connected (or continuous) to each other [2]. At the same manner, superpixels group pixels of different objects into perceptually homogeneous regions while adherence to object edges. Superpixel algorithm is a rational method because pixels of natural images are not independent to each other especially in the local scope. Superpixels with regular-ity lattice are able to represent an image with only a couple of hundred segments instead of tens of thousands of pixels, which heavily reduce the number of elements to handle. Superpixels are beneficial for a wide range of application domains in computer vision: salience detection [3-4] and tracking [5-6] etc.

To provide suitable representation of images, superpixels should have the following properties.

(1) Homogeneity: pixels of each superpixel should have similar colors and be connected.

(2) Boundary Adherence: superpixels should preserve object edges of images.

(3) Compactness: superpixels should have regular shapes and form a lattice structure.

(4) Practical Applicability: superpixels should be computational efficient, memory saving, and easy to use.

(5) Controllable Numbers of Superpixels: the number of superpixels could be specified by users.

Note that it is negative correlation between compactness and boundary adherence [7]. A good superpixel algorithm should provide tunable parameters to control the generation of superpixels according to the requirements on these two properties.

The published superpixel methods have two principal steps; distance computation and iteration

* Corresponding Author

procedure, while detail discussion about these methods is described in next section of related work. In this paper, distance metric between image pixels is mainly considered as the baseline. Distance metric measures the proximity, similarity and connectivity of pixels which determines the content and the shape of a superpixel. Euclidean distance is the most frequently used criterion to judge the proximity and similarity of pixels. The connectivity of pixels needs to be calculated in the large scope. Euclidean distance is sensitive to noise and is based on the assumption that the underlying data distribution is Gaussian [8]. But these two conditions can't be satisfied in a large scope on images. As a result, the connectivity of pixels that reflects the content of images can't be measured by Euclidean distance.

To accommodate the non-Gaussian character of images, geodesic distance is recommended which is computed by finding shortest paths in a connected graph [9]. Geodesic distance [10] and manifold [11] have also been used to design superpixel algorithms. However geodesic distance is not robust enough against noise and outliers as well which widely exist in natural images [12]. When there are variations caused by noises or outliers, all gaps between pixels along the path will be accumulated to geodesic distance.

The min-max distance is proposed which is also a path-based dissimilarity measure [13]. According to this definition, two image pixels are located on the same object if they are either similar or there exists a path that the two consecutive pixels on the path are similar. This is the essential property of superpixel algorithms which need to preserve image edges. In the meantime, this path-based distance can eliminate the effect of accumulative error because only one edge is involved in the final distance value. However, this distance metric may loss some important spatial relations and feature variations of image contents because only the largest edge cost of the minimal path is taken into consideration.

In Fig. 1, three points are considered, the common start point and the two end points of two arrows which are red color. The two end points have the same distance from the start point in image space, and the three points possess similar appearance because they are all located in the glass. Obviously Euclidean distance gives the same values in spatial and feature space. Geodesic distance accumulates the small gaps produced by illumination variation and noise. Meanwhile min-max distance captures only one edge while there are two extra edges along the upwards arrow actually.

Fig. 1. Illustration of different distance methods

To capture the image content that is ignored by the distance methods aforementioned, we propose a new distance method for superpixels generation. The contributions of this paper are listed briefly. Firstly, we propose a new path based distance, adaptive k distance to analyze the image content. Secondly, we give a computational method of this path based distance for image pixels. Thirdly, we present a superpixel method that integrates the spatial relation, the color similarity, and the image content between two pixels. Actually, the proposed distance metric can be applied to other superpixel methods as well.

The organization of this paper is described as follows. In Section 2, we summarize the recent related work of superpixel methods. In Section 3, we present our method, content perception superpixel. In Section 4, we evaluate our superpixel method on Berkeley Segmentation Dataset (BSD500) against several state of the art methods. Finally, we conclude our paper and further work in Section 5.

2 Related Work

In this section, some typical superpixel methods and superpixel-like segmentation methods are shortly summarized to give a brief introduction of this field. At the same time, algorithms adopted by these papers are discussed to show their emphases on superpixel generation.

A graph-based image segmentation method is proposed in [14]. This method is initialized by defining each pixel as a subgraph. And then similar subgraphs are merged by evaluating their similarity until convergence. Quick shift is another popular image segmentation method [15] which originates from Mean Shift algorithm. By connecting each point to the nearest neighbor, a tree is constructed and then splitted to generate superpixels. Because these two methods don't consider the requirements of superpixels, quite cluttered superpixels with varying sizes and irregular shapes are produced.

As a consequence, superpixel methods are designed specifically to overcome these drawbacks. The published superpixel methods have two principal steps: distance computation and iteration procedure. Many methods like clustering, morphological processing and optimization algorithms have been introduced to implement the iteration procedure. Distance computation is another research direction which attempt to reflect the image content.

SLIC algorithm leverages k-means clustering approach to generate superpixels [16]. Despite its simplicity, SLIC provides superior performance like boundary adherence and computational efficient. LSC maps the similarity metric of pixels to the high dimensional feature space using a kernel function [17]. Then simple K-means clustering method is applied to generated superpixels. SCSP builds the connectivity-constrained probabilistic model based on Gaussian Mixture Model [18]. Connectivity constraints ensure that each superpixel is simply connected.

GSM-TM computes the geographic element distribution and image edges of topographic map firstly [19]. Then, both elements and edges are input to guided watershed transform to obtain superpixels. Waterpixels method leverage the marker controlled watershed transformation to generate superpixels [20]. The spatial regularized gradient image is introduced to comprise between boundary adherence and regularity.

SEEDs method builds an energy driven function based on the color distribution and boundary of superpixels [21]. The hill climbing optimization algorithm is utilized to solve this function. LRW method produces initial boundaries of superpixels by lazy random walk firstly [22]. Further on an energy optimization framework is introduced, which is composed of the positions and the homogeneous of superpixels. Initial superpixels are re-fined during calculating process for these two methods.

Another critical component of superpixel algorithms is the distance metric between image pixels. SSS exploits geodesic distance to sense image structures [10]. The density of superpixels can be automatically adjusted according to image content. M-SLIC maps the 5-dimensional image features to a 2-dimensional manifold [11]. The content density of images can be computed by area elements of the manifold. Similar to [10], M-SLIC generates small superpixels in content-dense regions and large superpixels in content-sparse regions. A more detailed and comprehensive review of superpixels can be found in literature [1, 23].

3 Content Perception Superpixel

In this section, we describe the details of our method, including the image content analysis encountered in super-pixels, the path based distance computation method and the procedure of our superpixel algorithm. For image content analysis issue, we present a new path based distance, adaptive k distance method and the corresponding computing method for image pixels. Then we give the total work flow of our algorithm which combines Euclidean distance and the proposed path based distance. To bring down the computational burden, we use several strategies to simplify the distance computation.

3.1 Image Content Analysis

In this section, we discuss the content analysis for superpixels generation. According to the discussion aforementioned, the relation of two pixels is determined by three factors: spatial location, feature similarity and image content in-between. The first two factors are usually measured by Euclidean distance. But when the third factor is neglected, the measurement value remains the same even if the image content in-between varies dramatically. This may lead to disconnection and irregularity on superpixels' shape and size. The image content between two pixels needs to be taken into account as well. To solve this problem, we propose an adaptive k path based distance method for image content analysis. The proposed path based distance method can be synthesized into the computation of Euclidean distance easily.

Firstly we give the spatial and color difference computation method for pixels. Euclidean distance is utilized to calculate spatial location and feature similarity of pixels. The distance of spatial locations of two pixels p_i and p_j is calculated by their image coordinates

$$d_s(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\ , \tag{1}$$

where (x, y) is the image coordinate of a pixel. The distance of feature values of two pixels is calculated by their pixel value in CIELAB color space

$$d_c(p_i, p_j) = \sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}\ , \tag{2}$$

where (L, a, b) is the color of a pixel.

Our path based distance is based on geodesic distance and min-max distance. Geodesic distance and min-max distance are two commonly used methods to define path based distance. Geodesic distance is defined by

$$d_G = \min_{P \in R} \sum_{i=1}^{|P|-1} d_c(P[i], P[i+1]), \tag{3}$$

where R is the set of all paths connecting two specific pixels and symbol | | is the set size. Min-max distance is defined by

$$d_{MM} = \min_{P \in R} \left\{ \max_{i \in \{1,\ldots,|P|-1\}} \left\{ d_c(P[i], P[i+1]) \right\} \right\}, \tag{4}$$

where only the maximum cost edge is considered in contrast to geodesic distance. These two distances can be obtained efficiently by finding the minimal path of two pixels. Min-max distance can alleviate the negative impact of noise and outliers. But some spatial relations and feature variations may be neglected as illustrated in Fig. 1. As a result, we need a well-designed method to refrain from these drawbacks.

We propose an adaptive k path based distance method formulated as

$$d_k = L \cdot Num_k\left(d_{MM}\right), \tag{5}$$

where parameter L is the length of the minimal path d_{MM} and Num is the amount of maximum value along this path. Actually subscript k reflects the amount of edges crossing the path, which is shown in the next section. This method can alleviate the flaws of min-max distance which misses some important image contents like the edges in Fig. 1. Meanwhile this distance method inherits the robustness of min-max distance. Parameter L can be viewed as a penalty term. We utilize the length of a path as another metric in Equation 5. Under the condition of the same k, the longer the path length, and the larger value this distance has. This term is intuitive because the correlation of two pixels is inversely proportional to the length of the minimal path between them. Under this assumption, paths even encountering few edges will be penalized due to its too long length.

Given the three distance measures, an evaluation criterion is needed to judge how similar two pixels are. To combine the three terms into a uniform criterion, it is necessary that they need to be normalized. Location proximity and content connectivity are normalized by their respective maximum values, the initial superpixel interval S and the max path length M. Because the size of images is known, these two parameters can be estimated in advance. Normalization of color similarity can be implemented by divided a constant C. The total score is then formulated as

$$Dis(p_i, p_j) = \alpha \frac{d_s}{S} + \beta \frac{d_c}{C} + (1 - \alpha - \beta) \frac{d_k}{M}, \tag{6}$$

where compactness parameters control the proportion of the three terms. These two compactness parameters directly affect the superpixel shape and size.

The evaluation criterion has two types of parameters, the normalization parameters and the compactness parameters. Though 5 parameters are needed, they are friendly for users. The setting of normalization parameters and the effect of compactness parameters will be discussed in the next sections.

3.2 Distance Computation

In this section, we depict the adaptive *k* distance computation method for content perception superpixel. The adaptive *k* distance needs minimal paths between any two pixels as a preprocessing stage. The initial weight graph is constructed by the 4-neighbor of pixels. The minimal paths can then be calculated by Minimum Spanning Tree (MST). Prim and Kruskal algorithms can be applied to solve MST problem. A minimal path is denoted as $mp(p_1, p_2, L)$ where p_1 and p_2 are endpoints and L is the number of pixels on the path. After minimal paths are acquired, the image content between two specific pixels can be analyzed.

If the minimal path between two pixels has been obtained, the next step is the computation of adaptive *k* distance. Edges reflect the primary content of an image, by which superpixels should be separated. The goal of adaptive *k* distance is to capture the number of edges crossing the minimal path between the given two pixels. An intractable problem is the determination of parameter *k*. It isn't a constant because images have different content and prior information is unknown.

It can be regarded that the content of images comprises homogeneous regions and edges. In homogeneous regions, pixels present slight different values caused by noise and illumination etc.; in edges, pixels present significant variations in contrast to homogeneous regions. As a result, the maximum variations of a path correspond to the edges encountered by this path. Based on this observation, zero-crossings of second derivatives of pixels can be used to capture the maximum variations. Note that values of nodes along a path can be regarded as magnitudes of the first gradient of pixels. Parameter *k* can be identified by the count of pixels which satisfy

$$Num_k(mp) = \left|\left\{p_i \mid \left|\nabla I(p_i)\right| > T, \nabla^2 I(p_i) = 0, p_i \in mp\right\}\right|, \tag{7}$$

where symbol | | also denotes the magnitude of vectors, and T is the pre-specified threshold.

To reduce the computation burden, several strategies are leveraged. A too long path indicates that two pixels have weak relations. Because superpixels are suggested regular shapes for subsequent analysis steps, long paths should be ignored. To implement this operation, a further step is adopted directly

$$d_k = M \text{ } if \text{ } L > l, \tag{8}$$

where parameter l depends on the size of superpixels which is set as $2S$ in this paper and parameter M is the max path length.

Given two pixels p_1 and p_2, the value of adaptive *k* distance is also related to their spatial locations and color similarity. According to this observation, two other strategies are introduced. In view of homogeneity of superpixels, a minimal path meets

$$d_k = M \text{ } if \text{ } d_c(p_i, p_j) > \mu, \tag{9}$$

where parameter μ is a specified threshold. In view of compactness of superpixels, a minimal path meets

$$d_k = M \text{ } if \text{ } d_s(p_i, p_j) > \nu, \tag{10}$$

where parameter ν is a specified threshold as well. In this paper, parameters μ and ν are set as the constants C and $2S$ respectively. These two equations show that if two pixels are away from each other in color space or image space, the minimal path between them is set at the maximum value M. Under these three conditions, the number of needed minimal paths can be reduced dramatically.

3.3 Algorithm Procedure

In this section, we present the implementation details of our superpixel algorithm. Firstly the block diagram of the major work flow is illustrated in Fig. 2. Our method comprises four steps: a pre-processing step and a main loop that is composed of three steps. The generation of MST is a preprocessing stage before the main loop procedure. The loops should be executed until the error between two iterations converges. Generally after about 20 loops, shapes and locations of superpixels trends to be stable. The number of loops can be controlled by users on account of their requirement on accuracy. According to our experiments, 20-30 loops are recommended as reference.

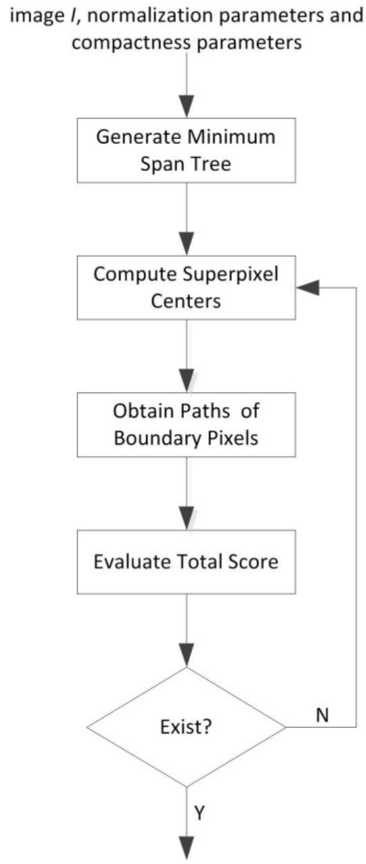


Fig. 2. Work flow of our method

Parameters of the proposed algorithm include compactness parameters α, β and normalization parameters S, M, C. Suppose that the number of pixels is N and the desired number of superpixels is K. Parameter S is the initial interval of superpixel centers that $S = \sqrt{N/K}$. Parameter M is set as the length of perimeter of images and parameter C can be in the range [10, 40]. Parameter C isn't a decisive parameter because it can be adjusted by compactness parameters further. Compactness parameters should be set according to users' requirements on shapes of superpixels.

The processing steps are shown in detail in Program 1. After minimum spanning tree of image pixels has been obtained or total scores have been evaluated, superpixel centers need to be figured out again. The initial superpixel centers are located on a regular grid with S pixels interval. After each loop, the superpixel center will be transferred to the average location of all the pixels belonging to the same superpixel. The color term d_c is the Euclidean distance between colors of pixels and the mean color of the corresponding superpixel. The spatial term d_s is the Euclidean distance between locations of pixels and the center of the corresponding superpixel. The content term d_k is the image structure variations along the path from pixels to the center of the corresponding superpixel. Finally, each pixel is assigned to a label that is the serial number of superpixels.

Program 1: Superpixel Generation.

```
Input: image I, number of superpixels K and compactness parameters α, β
Output: labels of image pixels
/* Initialization */
Initialize superpixel centers S_k on a regular grid with S pixels
interval
Generate MST for image I
```

```
Label(p_i) = -1 for each pixel
Dis(p_i) = ∞ for each pixel
Repeat 20-30 times
/* Assignment */
For each superpixel with center S_k
  For each pixel p_i
    If p_i is boundary
      Compute the distance D between p_i and S_k
      If D < Dis(p_i)
        Dis(p_i) = D
        Label(p_i) = k
  End For
End For
```

To reduce the amount of path re-calculation, two extra principles are applied during the loop procedure:
(1) Only paths with two endpoints at the center and boundary of superfixels are considered;
(2) If centers of superpixels don't move dramatically, paths maintain unchanged.

According to 1, only the distance of boundary pixels of superpixels is computed in each iteration. As a byproduct, this operation guarantees the connectedness of superpixels which provides better compactness. According to 2, if the center of a superpixel satisfies

$$\left|d_s^{t+1} - d_s^t\right| < \varepsilon \, , \tag{11}$$

adaptive *k* distance of boundary pixels stays the same as the previous round. If most boundary pixels of a superpixel remain unchanged and the location of the corresponding center doesn't move more than the threshold ε, the adaptive *k* distance of these boundary pixels maintains unchanged as well. This case occurs especially several iterations later.

4 Experiment and Discussions

In this section, we set up the experiments to show the performance of the proposed superpixel method. The Berkeley Segmentation Dataset (BSD500) is utilized to evaluate the performance of our method. BSD500 is a widely used public dataset in the field of image segmentation [24]. BSD500 dataset contains three parts: train set, validation set and test set. This dataset comprises 500 color images totally. At least 5 human annotated ground truth segmentations are given for each image. Superpixel methods are treated as a kind of pre-processing tasks in which parameters are usually appointed by users to acquire interesting information. In section 4.3, we can see how the specified parameters depend on the need of different requirements. So no parameter learning procedure is introduced in our experiments. All the three parts are used for performance evaluation. Our method is compared with several recent state of the art superpixel methods, including SEEDs [21], WP [20], LSC [17] and LRW [22].

In the remaining part of this section, the detailed content is presented. Firstly, evaluation metrics involved in our paper are discussed in Section 4.1. Secondly, qualitative results and quantitative results are given out in Section 4.2 and 4.3 respectively.

4.1 Evaluation Metrics

We utilize the standard metrics to evaluate the performance of superpixel methods, which are Boundary Recall (BR), COmpactness (CO) and three types of under-segmentation error, Levin's Under-segmentation Error (LUE), Neubert's Under-segmentation Error (NUE) and Michael's Under-segmentation Error (MUE). The generated superpixels are denoted as $S = \{s_i\}$ and the segmentations of the corresponding ground truth are denoted as $G = \{g_i\}$. Evaluation results are computed based on these two values. For BR and CO, higher is better; for LUE, NUE and MUE, lower is better.

4.1.1 Boundary Recall

Boundary recall is a commonly used metric to assess boundary adherence [24]. Boundary recall calculates the percentage of borders from the ground truth that are captured by the borders of superpixels. It can be formulated as

$$BR(S,G)=\frac{\left|\left\{p \mid p\in G, D\left(p,S\right)<2\right\}\right|}{|G|}, \tag{22}$$

where $D(p, S)$ is the minimal distance between p and pixels in S.

4.1.2 Compactness

Compactness is firstly proposed by Alexander Schick [7]. This metric is utilized to evaluate the shape of superpixels especially the boundary length. Compactness is defined as follows

$$CO(S,G)=\frac{1}{K}\sum_{s_i}|s_i|\frac{4\pi A\left(s_i\right)}{P(s_i)}, \tag{33}$$

where A is the area of superpixel s_i and P is the area of a circle that has the same perimeter with s_i. Compactness compares the area of each superpixel with the area of a circle with the same perimeter. The larger the area of a region for a given boundary length, the higher is its compactness. As a result, this metric favors regular superpixels.

4.1.3 Under-segmentation Error

Under-segmentation error is another commonly used metric to assess boundary adherence. In ideal conditions, each superpixel s_i should be completely covered by only one segmentation g_j from the ground truth. Under-segmentation error measures how many pixels from s_i that has an intersection with g_j "escape" across the boundary of g_j. Owing to different viewpoints, different formulations about under-segmentation error are proposed. For comprehensive comparisons, we leverage three definitions of under-segmentation error.

Levin's Under-segmentation Error (LUE). This is the first definition of under-segmentation error [25]. It can be formulated as

$$LUE=\frac{1}{|G|}\sum_{g_i}\frac{\left(\sum_{|s_i\cap g_i|\neq 0}|s_i|\right)-|g_i|}{|g_i|}, \tag{44}$$

where $|g_i|$ is the number of pixels in the segmentation g_i and $|G|$ is the number of human-annotated segmentations. For each ground truth segmentation, the number of pixels in the superpixels that are overlap with this ground truth segmentation minus the number of pixels in the segmentation. Then these pixels are summarized and divided by the number of segmentations in the ground truth. This metric has two main drawbacks. Firstly LUE penalizes those superpixels that are only slightly overlapping with neighboring ground truth segmentations. For example, if some pixels of a superpixel locate on the boundary of a ground truth segmentation, LUE penalizes this superpixel on at least two ground truth segmentations with this boundary. Secondly the result of LUE doesn't lie in [0, 1].

Neubert's Under-segmentation Error (NUE). Neubert et al propose a new definition of under-segmentation error [26] which can be formulated as

$$NUE=\frac{1}{N}\sum_{g_i}\sum_{s_i\cap g_i\neq\phi}\min\left\{\left|s_i\cap g_i\right|,\left|s_i-g_i\right|\right\}, \tag{55}$$

where $|s_j \cap g_i|$ is the number of pixels that belong to s_j and g_i simultaneously, and $|s_j-g_i|$ is the number of pixels that belong to s_j but not g_i. For each ground truth segmentation, the minimal value between the intersection set and the difference set is adopted from the superpixels that are overlap with this ground truth segmentation. Then these values are summarized and divided by the number of pixels in the image. The value of NUE lies in the interval [0, 1] because the inner sum is less than or equal to the size of g_i.

Michael's Under-segmentation Error (MUE). Michael et al propose another new definition of under-segmentation error [21] which can be formulated as

$$MUE = \frac{1}{N}\sum_{s_j}\left|s_j - \max_{g_i}\left|s_j \cap g_i\right|\right|, \quad \textbf{(66)}$$

where the operations are the same as those in NUE. For MUE, each superpixel is assigned to only one ground truth segmentation that has the largest overlap. Pixels of each superpixel that don't belong to the corresponding ground truth segmentation are taken into account. Finally these pixels are summarized and divided by the number of pixels of the image. The value of MUE also lies in the interval [0, 1] because each superpixel is counted only once.

An interest phenomenon is that NUE and MUE are relevant even if their definitions seem quite different according to Equation 15 and 16. The detailed explanation is as follows. First of all, a simple sketch of NUE and MUE is shown in Fig. 3. For NUE, the smaller parts of sp1 are counted for gt1 and gt2; for MUE, the smaller parts of sp1 are counted as well but only for gt1. The difference is that NUE traverses all the ground truth segmentations, so some superpixels are counted more than once. Meanwhile MUE traverses all the superpixels only once. As a result, NUE is at least two times as much as MUE because the concerned superpixels are covered by at least two ground truth segments.

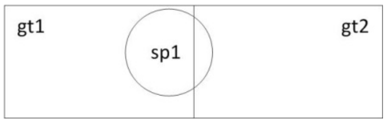


Fig. 3. A sketch of NUE and MUE

4.2 Qualitative Results

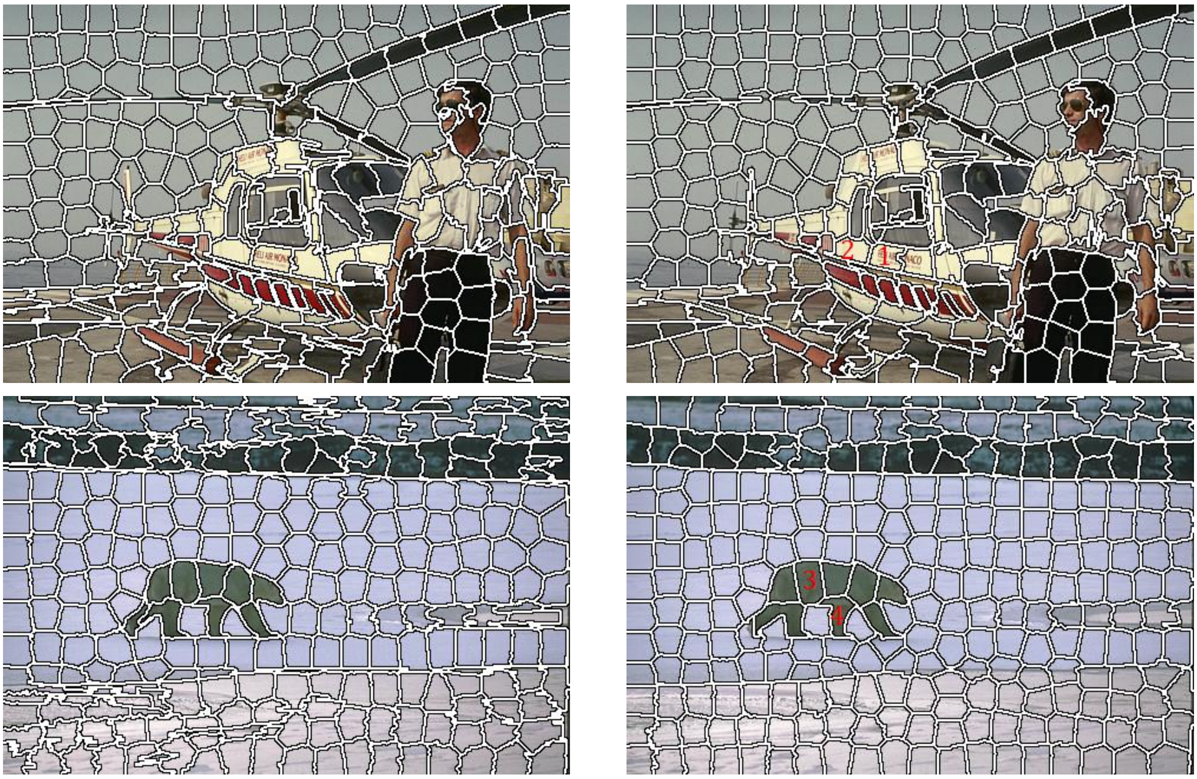
In this section, we present the qualitative results of our superpixel method. First of all, we give the parameter setting of our method in Table 1 where *N* is the number of pixels and *K* is the number of superpixels. Two groups of compactness parameters are adopted to show the performance of our method, which are (0.50, 0.45) and (0.75, 0.20).

Table 1. Parameters

Parameters	Values
α	0.50/0.75
β	0.45/0.20
S	$\sqrt{N/K}$
M	$4N$
C	10

Fig. 4 shows a few screenshots generated by our method. 300 superpixels are generated for each image. Part (a) is generated under the condition that α and β are 0.50 and 0.45; Part (b) is generated under the condition that α and β are 0.75 and 0.20. It can be found that the compactness parameters control the shape of superpixels directly. Higher α produces more regular superpixels and lower α produces superpixels more adherence to edges.

As indicated by numbers in red in the first row of Fig. 4, these two superpixels have similar shape and size. But from their appearance, these superpixels have different contents. On the contrary, other two superpixels indicated by numbers in the second row have similar contents even though they have different shape and size. Our method presents a powerful tool to describe these cases. Normalized adaptive *k* distance of pixels locating in the same superpixel is computed and summed which is shown in Table 2. Superpixel *1* and superpixel *2* have quite different values; superpixel *3* and superpixel *4* have almost the same value. These values depict the contents of these superpixels accurately.



(a) α and β are (0.50, 0.45) (b) α and β are (0.75, 0.20)

Fig. 4. Screenshots of our method

Table 2. Contents of superpixels

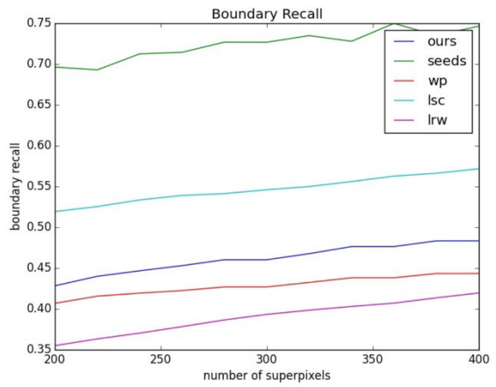
Superpixels	adaptive *k* distance
1	0.733097
2	0.055770
3	0.000731
4	0.001266

4.3 Quantitative Results

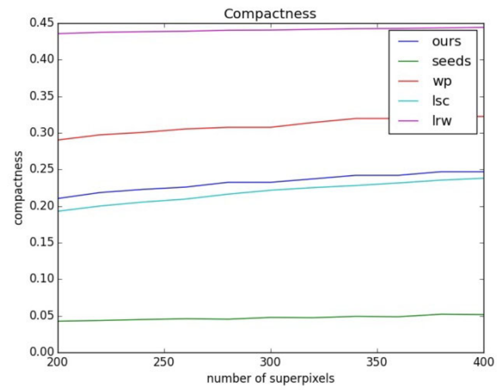
In this section, we present the quantitative results of our superpixel method comparing with the state of the art methods based on the metrics described in Section 4.1. The evaluation metrics are divided into two groups according to their emphasis on superpixels. One group is boundary recall and compactness which focuses on the shape of superpixels; the other is the three types of under-segmentation error which focuses on the size and location of superpixels.

Boundary recall and compactness curve of our method and the other methods are shown in Fig. 5. From this figure, we acquire the completely coincident results as Alexander Schick et al [7]. The higher the boundary recall, the lower the compactness is and vice versa. This is apparent because for a high boundary recall, superpixels have to capture all minor details of an image which results in the more irregular shapes. Boundary recall and compactness conflict to each other so they can't be concerned about simultaneously. This conflict comes from the inherent nature of images because edges of images aren't horizontal alignment or vertical alignment in general.

The conclusion from Fig. 5 doesn't depend on the superpixel methods no matter how the parameters are tuned. As a result, the desire to improve the performance of these two metrics can't be implemented. But from another point of view, these two metrics are mutually complementary. Until now, none of the published methods achieve a satisfactory trade-off between the two metrics.



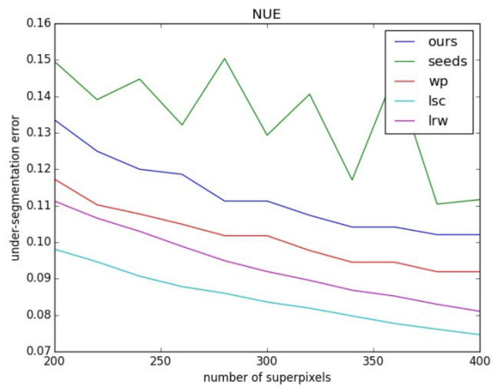
(a) Boundary recall



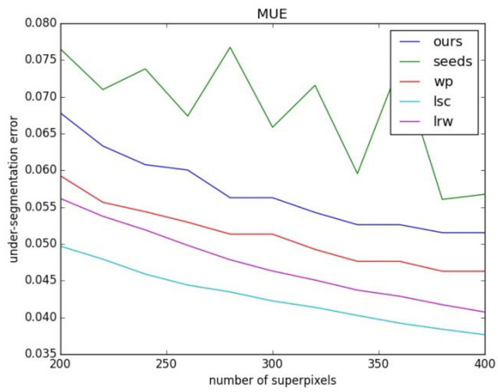
(b) Compactness

Fig. 5. Boundary recall and compactness curves

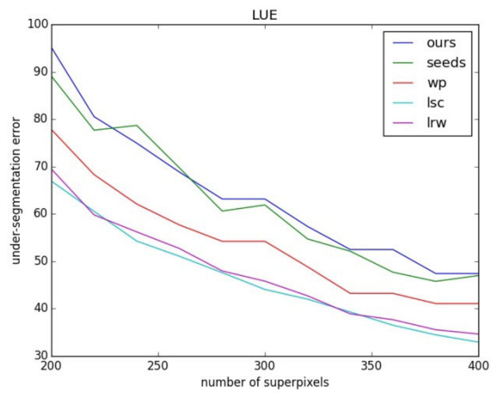
In real applications, superpixel methods are usually utilized as a pre-processing stage to provide operation primitives for higher level vision tasks such as image segmentation and object tracking. This means that superpixel methods need to be adapted to various sorts of requirements in different application scenarios. For example, boundary recall is more important for salience segmentation; compactness is more important for image pyramid representation. As a result, it is necessary for superpixel methods that they should provide a mechanism to compromise between boundary recall and compactness. Our method provides this kind of mechanism as the other methods involved in this section. The compactness parameters α, β can be tuned so that our method can handle the demand on boundary recall and compactness. The effect of compactness parameters is listed as follows: the bigger the α, the more compact superpixels are; the bigger the β, the more tightly superpixels are adherence to image edges.



(a) NUE



(b) MUE



(c) LUE

Fig. 6. Under-segmentation error curves

Comparisons of our method and the other methods on the three type under-segmentation errors are shown in Fig. 6. First of all, it is clearly that as the increasing of the number of superpixels, values of the three metrics trend to decrease gradually for all methods. Even though seeds method fluctuates within a certain range on NUE and MUE, the downward tendency has not changed. It can be concluded that one determining factor of UE is the number of superpixels which controls the superpixel size. Part (a) and (b) of Fig. 6 show that results of these methods on NUE and MUE agree with the analysis in Section 4.1. Graph representations of NUE and MUE are almost the same as each other. Besides, NUE is about twice as much as MUE. This consequence implies that most superpixels cover no more than two ground truth segmentations.

Part (a) and (b) of Fig. 6 shows that our method achieves the second best performance on NUE and MUE comparing with the state of the art methods. Although seeds method achieves the best performance among these methods especially on NUE and MUE, there exists a difficult issue to tackle for seeds method. Its performance on NUE and MUE doesn't depend on the number of superpixel monotonously. If we want to reduce its NUE and MUE, it may not be successful by increasing the number of superpixels purely. This makes seeds method not so kindly to use in practical applications. Part (c) shows that our method achieves the best performance on LUE comparing with the state of the art methods when the number of superpixels is larger than about 270. On the whole, our method is a satisfactory tool for image over-segmentation according to the results of experiments. Especially our method provides extra image content comparing with the other methods referred in this paper.

5 Conclusions and Future Work

Superpixels aim to provide low level image segmentations as a pre-processing stage to alleviate the complexity of subsequent operations. In this paper, our research concentrates on the distance calculation of superpixel algorithms. Existing superpixel methods focus on the spatial and color difference of pixels where Euclidean distance is widely used. But it is difficult for Euclidean distance to depict the image content. We propose a path based distance, adaptive k distance, to make up this defect. Following the definition of adaptive k distance, we give a computation method of adaptive k distance on image pixels. Finally we present the whole algorithm procedure of our method.

In the experiment section, we leverage the principal evaluation criteria which are divided into two groups according to their emphasis on superpixels. One group is boundary recall and compactness, the other group is the three types of under-segmentation errors. Comparisons between our method and the other methods are presented on the basis of these evaluation criteria. We discuss the results on the public dataset BSD500 in detail and analyses the availability of these methods in real applications. In the meantime, we investigate these evaluation criteria to give some instructions about their effect on evaluating superpixels. As a conclusion, our method achieves favorable performance comparing with the state of the art methods. In addition, the proposed distance method can be used in the other superpixel algorithms easily.

In future, we will improve our method continuously. First we will reduce the computational complexity of our method in which the generation of MST is a time-consuming operation. We will attempt to bring in fast algorithms of MST. Second we will study how the image contents sensitive superpixels enhance subsequent operations. Though image contents of superpixels are known, it is still an issue to leverage this information in practical applications. Finally we will utilize image pyramid to speed up our method based on the characteristic of natural images.

Acknowledgements

This research is supported by "National Natural Science Foundation of China" (No. 61272523), "the National Key Project of Science and Technology of China" (No. 2011ZX05039-003-4) and "the Fundamental Research Funds for the Central Universities".

References

[1] D. Stutz, A. Hermans, B. Leibe, Superpixels: an evaluation of the state-of-the-art. <https://arxiv.org/abs/1612.01601>, 2016.

[2] L. Chen, The topological approach to perceptual organization, Visual Cognition 12(4)(2005) 553-637.

[3] K. Fu, C. Gong, J. Yang, Y. Zhou, I.Y.-H. Gu, Superpixel based color contrast and color distribution driven salient object detection, Signal Processing-Image Communication 28(10)(2013) 1448-1463.

[4] F. Perazzi, P. Kraehenbuehl, Y. Pritch, A. Hornung, Saliency filters: contrast based filtering for salient region detection, in: R. Chellappa, B. Kimia, S.-Ch. Zhu (Eds.), Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR2012, Rhode Island, USA, June 16-21, 2012, IEEE, New York, USA, 2012, pp. 733-740.

[5] Y. Fan, L. Huchuan, Y. Ming-Hsuan, Robust superpixel tracking, IEEE Transactions on Image Processing 23(4)(2014) 1639-1651.

[6] A. Mazzu, P. Morerio, L. Marcenaro, C.S. Regazzoni, A cognitive control-inspired approach to object tracking, IEEE Transactions on Image Processing 25(6)(2016) 2697-2711.

[7] A. Schick, M. Fischer, R. Stiefelhagen, An evaluation of the compactness of superpixels, Pattern Recognition Letters 43(2014) 71-80.

[8] Y. Jie, T. Qi, J. Amores, N. Sebe, Toward robust distance metric analysis for similarity estimation, in: A. Fitzgibbon, Camillo J. Taylor, Y. LeCun (Eds.), Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition, CVPR2006, New York, USA, June 17-22, 2006, IEEE, New York, USA, 2006, pp. 316-322.

[9] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290(5500)(2000) 2319-2323.

[10] P. Wang, G. Zeng, R. Gan, J. Wang, H. Zha, Structure-sensitive superpixels via geodesic distance, International Journal of Computer Vision 103(1)(2013) 1-21.

[11] Y.-J. Liu, M. Yu, B.-J. Li, Y. He, Intrinsic manifold SLIC: a simple and efficient method for computing content-sensitive superpixels, IEEE Transactions on Pattern Analysis and Machine Intelligence 40(3)(2017) 653-666.

[12] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering, Pattern Recognition 41(1)(2008) 191-203.

[13] B. Fischer, J.M. Buhmann, Path-based clustering for grouping of smooth curves and texture segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25(4)(2003) 513-518.

[14] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59(2)(2004) 167-181.

[15] A. Vedaldi, S. Soatto, Quick shift and kernel methods for mode seeking, in: D. Forsyth, P. Torr, A. Zisserman (Eds.), Proceedings of the 10th European Conference on Computer Vision, ECCV2008, Marseille, France, October 12-18, 2008, Springer, Berlin, German, 2008, pp. 705-718.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Suesstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34(11)(2012) 2274-2281.

[17] J. Chen, Z. Li, B. Huang, Linear spectral clustering superpixel, IEEE Transactions on Image Processing 26(7)(2017) 3317-3330.

[18] O. Freifeld, Y. Li, J.W. Fisher, A fast method for inferring high-quality simply-connected superpixels, in: J.-L. Dugelay, A.M. Morin (Eds.), Proceedings of 2015 IEEE International Conference on Image Processing, ICIP2015, Quebec City, Canada, September 27-30, 2015, IEEE, New York, USA, 2015, pp. 2184-2188.

[19] Q. Miao, T. Liu, J. Song, M. Gong, Y. Yang, Guided superpixel method for topographic map processing, IEEE Transactions on Geoscience and Remote Sensing 54(11)(2016) 6265-6279.

[20] V. Machairas, M. Faessel, D. Cardenas-Pena, T. Chabardes, T. Walter, E. Decenciere, Waterpixels, IEEE Transactions on Image Processing 24(11)(2015) 3707-3716.

[21] M. Van den Bergh, X. Boix, G. Roig, L. Van Gool, SEEDS: superpixels extracted via energy-driven sampling, International Journal of Computer Vision 111(3)(2015) 298-314.

[22] S. Jianbing, D. Yunfan, W. Wenguan, L. Xuelong, Lazy random walks for superpixel segmentation, IEEE Transactions on Image Processing 23(4)(2014) 1451-1462.

[23] M. Wang, X. Liu, Y. Gao, X. Ma, N.Q. Soomro, Superpixel segmentation: a benchmark, Signal Processing-Image Communication 56(2017) 28-39.

[24] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5)(2011) 898-916.

[25] A. Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, K. Siddiqi, TurboPixels: fast superpixels using geometric flows, IEEE Transactions on Pattern Analysis and Machine Intelligence 31(12)(2009) 2290-2297.

[26] P. Neubert, P. Protzel, Evaluating superpixels in video: metrics beyond figure-ground segmentation, in: T. Burghardt, D. Damen, W. Mayol-Cuevas, M. Mirmehdi (Eds.), Proceedings of 2013 British Machine Vision Conference, BMVC2013, Bristol, UK, September 9-13, 2013, BMVA Press, Durham, UK, 2013, pp. 54.1-54.11.