Boundary-preserving Superpixel Segmentation

Yuejia Lin, Zhao Li, Chenxun Yuan, and Yi Liu*

School of Software, Shandong University, Jinan 250101, China

Corresponding author. E-mail: liuyi@sdu.edu.cn

Received: March. 23, 2023; Accepted: May. 09, 2023

In recent years, superpixel segmentation has been widely used in image processing tasks as a preprocessing step. Superpixel segmentation aims to group pixels into homogeneous regions while maintaining edges. This paper proposes a superpixel segmentation algorithm based on boundary preservation. In the algorithm, the side window filtering is first used to smooth the image texture area, so that the superpixel shape generated at the texture is regular. Different from other superpixel clustering algorithms, the algorithm in this paper uses a new distance measurement function for distance measurement, which can assign different weights to its color distance items and spatial distance items according to different pixels, so that the superpixel fit in the image boundary area. The boundary is regular in the flat area. The distance measurement function also takes into account the pixel information of the linear path from the pixel to the cluster center, and avoids the category error division caused by only the local information of the pixel for clustering. Finally, this paper designs a new cluster center update strategy, which uses only the weighted average of some reliable pixels in the superpixel as the new cluster center, thereby reducing the update of the cluster center of pixels that are not very similar to the cluster center. The interference makes the cluster center update more accurate. Experimental results show that our algorithm can get better results in visual effects and BR,UE,ASA indicators compared with existing algorithms.

Keywords: Superpixel segmentation; Clustering; superpixels; Image boundaries; Image segmentation

http://dx.doi.org/10.6180/jase.202404_27(4).0002

1. Introduction

In 2003, Ren and Malik et al. first proposed the concept of superpixel segmentation [1]. Superpixel segmentation refers to dividing an image into several sub-areas with regular shapes and the same size, and making the same area have features such as grayscale, color, and texture. Similar to the above, these sub-regions are called superpixel. Replacing the original pixels in the image with superpixel to represent the image can effectively reduce the redundant information of the image, thereby shortening the running time of subsequent image processing algorithms, so superpixel segmentation is often used as an image preprocessing step to be widely used in various computer vision tasks. For example, in image segmentation [2, 3], image denoising [4], target tracking [5], object recognition [6], saliency detection [7], indoor scene segmentation [8], quality inspection [9] ,classification [10, 11] and other computational vision tasks [12–15] has the application of superpixel segmentation.

In various applications of superpixel segmentation, the quality of superpixel will greatly affect the effect of subsequent algorithms. Generally, a good superpixel segmentation algorithm should meet the following requirements: First, the generated superpixel should be close to the edge of the object in the image. The closer the edge of the superpixel is to the edge of the object, the more accurate the segmentation result will be. The second is that the pixels inside the superpixel should have similar characteristics such as color, texture, brightness, etc., and only pixels with the same or similar properties can be represented by a superpixel. The third is to ensure that the generated superpixel have regular shapes, and regular and dense superpixel are conducive to the subsequent processing of the algorithm. In fact, boundary regularization and boundary overlap themselves are often contradictory. If the generated superpixel are more regular, the boundary of the superpixel is difficult to keep fit with the image boundary, and the higher the degree of fit between the generated superpixel boundary and the image boundary, Then superpixel must be difficult to ensure regularity.

This paper proposes a superpixel segmentation algorithm based on clustering. The algorithm first preprocesses the image with side window filtering, smooths the texture area in the image, and makes the superpixel generated by the texture area more regular. In order to make the generated superpixel fit the image boundary more closely, this paper proposes a new distance measurement formula, which consists of two terms. The first term is the distance term based on the boundary probability. The boundary probability in this term is calculated by the image gradient. The boundary probability reflects the probability that the pixel is located on the image boundary. When the boundary probability is large, the color of the pixel is the main basis for superpixel division, thus ensuring that the superpixel generated in the boundary area can be Closely fit the image boundary; when the boundary probability is small, the spatial distance term becomes the main division basis in the clustering process, so that the superpixel generated in the non-boundary area are more regular. The second term of the distance term is a penalty term based on the linear path. This term takes into account the boundary probability information on the linear path from the pixel to the cluster center, and imposes a penalty on the cross-boundary clustering, thereby further improving the superpixel's performance. Uniformity within the class. Finally, this paper designs a new cluster center update strategy, which can reduce the interference of unreliable pixels in the cluster to the update of cluster centers and improve the accuracy of the update of cluster centers. The main points of this article are the following three innovations:

This paper introduces side window filtering as a preprocessing step, so that the generated superpixel can keep a certain degree of regularity in the texture area while keeping close to the edge of the object.

- The algorithm in this paper designs a new distance measurement formula. This formula can adaptively assign different weights to the color distance items and spatial distance items of pixels in different regions, so that the generated superpixel can be closer to the image boundary. In addition, the formula also takes the boundary information from the pixel point to the cluster center into consideration, which further improves the boundary fitting performance of the superpixel.
- This paper proposes a new cluster center update strategy. This strategy first eliminates the abnormal pixels in the superpixel according to the three sigma principle, and then assigns different weights to the remaining pixels based on their similarity to the cluster centers, and updates the cluster centers in a weighted average manner to make the cluster centers The update is more accurate.

2. Related work

Superpixel segmentation has received a lot of attention in recent years. Scholars have proposed many effective superpixel segmentation algorithms. Superpixel segmentation algorithms can be roughly divided into two categories: graph theory-based superpixel segmentation algorithms and cluster-based superpixel Segmentation algorithm.

2.1. Graph theory-based algorithms

Among the superpixel segmentation algorithms based on graph theory, Ncuts [16] is the most typical one. The algorithm treats the pixels in the image as nodes of the graph, and uses the brightness and texture information of the image to perform iterative segmentation; however, the computational complexity of the algorithm is high, and the contour fit of the segmentation result is poor. In addition, Liu et al. proposed an entropy rate-based superpixel segmentation algorithm ERS (entropy rate superpixel) [17] in 2011, which is another classic algorithm in this field. This algorithm uses an entropy rate that contains a random walk of an image and An energy function that guarantees that superpixel have similarly sized balance terms to produce superpixel. The algorithm has high execution efficiency and the generated superpixel boundary and the image boundary can be well overlapped, but the generated superpixel is very irregular.

Shen et al. proposed a random walk superpixel segmentation algorithm LRW (lazy random walks) in 2014 [18]. The algorithm is improved on the basis of the traditional random walk algorithm, and proposes a lazy random walk algorithm with compact term constraints. The algorithm takes into account the relationship between the pixel and all cluster centers, and solves the traditional problem. The random walk algorithm only considers the problem of irregular shape of superpixel caused by the local relationship between pixels and cluster centers. The algorithm can generate regular superpixel in the texture area and has strong robustness, but its complexity is squared with the number of pixels in the image, which makes the algorithm more time-consuming to execute.

In 2018, Gong et al. proposed a differential evolution-based superpixel generation algorithm DES (Differential evolutionary superpixel) [19]. The objective function proposed by the algorithm is composed of superpixel internal errors, image boundary gradients, and global variance regularization terms. The evolutionary global optimization realizes the over-segmentation of the image and obtains the super-pixel segmentation result. The time complexity of this algorithm has a linear relationship with the number of pixels in the image, which can meet the real-time requirements of superpixel segmentation, but the superpixel generated by the segmentation have poor regularity, which is not conducive to the application of subsequent algorithms.

2.2. Cluster-based algorithm

In the cluster-based superpixel segmentation algorithm, the superpixel generation process is regarded as the pixel clustering process. Therefore, many clustering algorithms are used to generate superpixel. Achanta et al. [20] proposed the SLIC algorithm in 2012, which uses the K-means clustering method to generate superpixel. The algorithm can generate superpixel with regular shapes and similar sizes, and the algorithm has a high time execution efficiency. Therefore, the algorithm is widely used in many image preprocessing. The main disadvantage of this algorithm is poor boundary fit, especially in complex areas. Shen [21] and others will use the DBSCAN clustering algorithm to apply superpixel segmentation tasks. The DBSCAN algorithm can obtain regular superpixel very quickly, but it cannot maintain the boundary well, and the number of superpixel finally obtained is not controlled.

In 2018, Ban et al. proposed a superpixel generation algorithm GMM (gaussian mixture model)[[22] based on a Gaussian mixture model. The algorithm is based on the Gaussian mixture clustering algorithm, and each superpixel block corresponds to a Gaussian function. And then use the EM algorithm to solve the parameters of the Gaussian function through the maximum likelihood estimation method, and map the corresponding pixels in the image to the Gaussian model to obtain the superpixel segmentation result. The superpixel boundary fitting performance generated by this algorithm is better, but the shape of the superpixel is irregular.

Zhang [23] and others also proposed an algorithm that includes pixel boundary probability. Unlike us, the pixel boundary probability function is only used to ensure the regularity of superpixel. This algorithm can generate regular superpixel and superpixel boundaries can be very good. The ground fits the image boundary, but this algorithm is more time-consuming.

Giraud et al. and 2018 proposed a linear path-based superpixel generation algorithm (scalp) [24], which is based on the SLIC algorithm, which adds a linear path from the pixel to the cluster center in the distance measurement formula. The above color information improves the intra-class uniformity of superpixel.

Xu et al. [3] proposed a robust divide-and-conquer superpixel segmentation method to divide the input image into flat and non-flat regions. Among them, the former targets shape regularity and the latter emphasizes boundary observance. And the algorithm also adopts adaptive parameters. This algorithm can quickly generate superpixels with stronger boundary adhesion and keep regularity.

3. The method

The algorithm in this paper is based on the K-means clustering algorithm. The algorithm mainly includes three parts: firstly, the input image is processed by side window filtering, and then K superpixel seed points are uniformly initialized in the image in a regular hexagon manner as the cluster center. Then use the newly designed distance measurement formula to cluster the pixels, and use the new cluster center update scheme to update the cluster centers, and repeat the above process until convergence. Finally, the small superpixel blocks are merged. The superpixel generated by the algorithm in this paper can fit the image boundary as low as possible, and the shape can be regular in the flat area.

3.1. Side window filtering

There are extensive texture regions in natural images. The sharp changes in the gradient of the texture region of the image will result in irregular shapes of the generated superpixel. Therefore, it is necessary to perform proper filtering preprocessing on the original image to smooth the texture regions and improve the quality of superpixel generation. In the filtering preprocessing step, if traditional filtering methods such as Gaussian filtering are used, it will cause edge blur, which will affect the quality of subsequent superpixel generation. Here we use side window filtering to solve this problem.

Side window filtering [25] is a local linear filtering that combines edge preservation and texture smoothing. The core idea of this algorithm is to treat all pixels to be processed as potential edge pixels, and to compare the edges of the filter sub-window with the pixels to be processed. Align to protect the edges of the image.If a given pixel i, p_i represents the pixel value, the side window filter will filter by enumerating eight possible directions, and by minimizing the cost function of the difference between the input p_i and the output $I_{ij}(j = \{1, 2, 3, \ldots, 8\})$ and adaptively select the best direction for filtering. The filtered image can be expressed as Equation 1:

$$I_{SWF} = \arg \min_{\forall I_{ij}, j \in [1,8]} \left\| p_i - I_{ij} \right\| \tag{1}$$

Among them, I_{SWF} represents the filtering result of the side window of the pixel i, and I_{ij} represents the filtering result of the j direction of the pixel i. In the experiment, the filter window radius is set to 4 , as shown in Figure 1(b), because the gradient of the texture area at the ripples of the water surface changes greatly, and the superpixel algorithm is sensitive to the gradient change when measuring the distance, so it is generated in the texture area. The superpixel are extremely irregular. Figure 1(c) is the result of the superpixel generated after the side window filtering process. Compared with the superpixel generated without the side window filtering process, it can be seen that the superpixel generated after the side window filtering process can guarantee the fit the edge can be regular in the image texture area.

3.2. Definition of distance measurement function

In the process of superpixel clustering, a pixel should be classified into the category most similar to its nature. The similarity is measured between the pixels and each category through the distance measurement function. We define a new distance metric to calculate the similarity between the pixel point and the seed point. Formula 2 is the definition of the new distance function. The distance publicity is composed of two distance items, $d_W(x, k)$ and $d_L(x, k)$. $d_W(x, k)$ is the distance term from the pixel point x to the cluster center k based on the edge probability. This term maps the gradient of the pixel point x to the boundary probability through the boundary probability mapping function, and assigns weights to the spatial distance item and the color distance item according to the boundary probability. This item makes the generated superpixel fit the edges more closely. $d_L(x, k)$ is the penalty term based on the linear path from the pixel x to the cluster center k. This item is obtained by accumulating and summing the boundary probabilities on the linear path from the pixel x to the cluster center k. The distance measurement acts as a penalty and further strengthens the uniformity of colors within the superpixel. These two are introduced separately below.

$$\mathrm{D}(x, k) = d_W(x, k) + d_L(x, k) \tag{2}$$

3.2.1. *Distance term based on marginal probability*

Formula 3 is a distance term based on the edge probability, where $I(x, k)$ represents the color space distance between the pixel point x and the cluster center k, and this term is used to ensure that the internal color of the generated superpixel is uniform. The color space distance term is calculated by formula 4, where λ_1 is a constant, $d_c(x, k)$ is the Euclidean distance between the pixel point x and the cluster center k in the CIELAB color space, the closer the color of the pixel point x and the cluster center k is to $I(x, k)$ The smaller.

$w(x)$ is the boundary probability function, which represents the probability that the pixel x falls on the image boundary. This item assigns different weights to the color distance item and the space distance item for different pixels, and the weight is determined by $w(x)$. The greater the probability that the pixel falls on the boundary, the color distance between the pixel and the seed point should be considered more in the clustering and division process of this type of pixel, so as to ensure that the generated superpixel fits the edge of the image more closely.

Therefore, the color distance weight should be increased at this time, and the spatial distance weight should be reduced.

$$\begin{aligned} d_W(x, k) = \left(1 + \frac{w(x)}{2}\right) \cdot I(x, \mathrm{k}) + \\ \left(1 - \frac{w(x)}{2}\right) \cdot C(x, k) \end{aligned} \tag{3}$$

$$I(x, k) = e^{\frac{d_c(x,k)}{\lambda_1}} \tag{4}$$

$$C(x, k) = e^{\frac{d_p(x,k)}{\lambda_2}} \tag{5}$$

The aforementioned edge probability $w(x)$ is obtained by mapping the image gradient amplitude $g(x)$ through the edge probability function. The gradient magnitude of the pixel can reflect the probability of the pixel falling on the image boundary to a certain extent. Generally, the gradient is small in the flat area of the image, and the gradient is large at the edge of the image. Although the gradient amplitude can characterize the probability of a pixel falling on the image boundary, the gradient information between different images is very different, and the gradient amplitude has a large range, which makes it difficult to directly use the gradient amplitude to characterize the probability of a pixel on the edge. Therefore, we need a normalized function to reasonably map the gradient amplitude to obtain

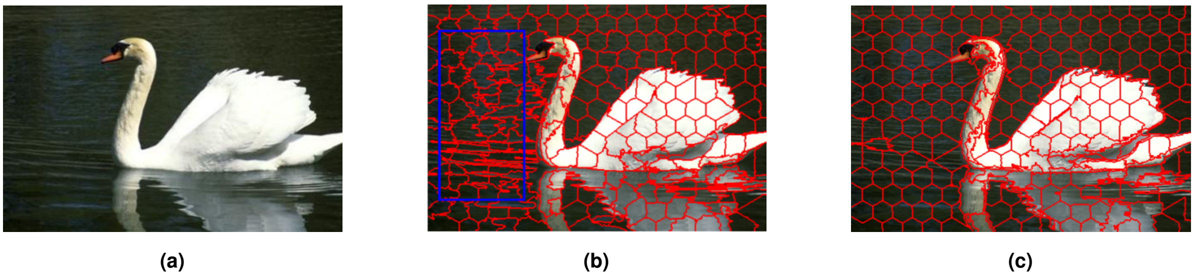


Fig. 1. (a) Original image (b) Original image superpixel generation result (c) Original image superpixel generation result.

the edge probability of the pixel.

$$w(x) = \frac{1}{1 + e^{-a[g(x)-b]}} \tag{6}$$

In this paper, the function shown in formula 8 is used as the boundary probability mapping function, which is obtained by shifting the sigmoid function by b units along the x axis and performing stretching changes. As shown in Figure 3, this function has good characteristics. It is symmetric about the center of point $(b, 0.5)$, and can map the domain of $[0, +\infty)$ to the range of $(0, 1)$

And this function has dual threshold characteristics. When the gradient amplitude of the pixel is in the range of $[0, t)$, it means that the gradient amplitude of the pixel is too small. It can be considered that the pixel is not the pixel at the edge of the image. At this time, the boundary probability mapping function will change the gradient amplitude. The value is mapped to a value close to 0 ; when the gradient amplitude of the pixel is within the range of $(T, +\infty)$, it means that the gradient amplitude of the pixel is large enough to be considered as a pixel at the edge of the image, and the boundary probability function will map it to close the value of 1 . In Equation 5, a and b are hyperparameters, and we can adjust the values of the two thresholds a and b by adjusting the values of t and T.

The comparison in Fig. 3 illustrates the effect of this item. As shown in Fig. 3(a), when the distance item without weight function is used for measurement, the spatial distance between the green pixel x and the seed point k_2 is much smaller than that to the seed point. The spatial distance of k_1, that is $C(x, k_2) < C(x, k_1)$, although the pixel x is closer to k_1 in the color space, that is $I(x, k_1) < I(x, k_2)$, the spatial distance gap is much larger than the color distance gap, which leads to the superpixel segmentation process, the pixel point is too far away from the seed point The problem that the color distance item is invalid at close time. The weighted distance term mentioned in Equation 3 can solve this problem. Equation 3 shows that when the pixel x is located at the edge of the image, $w(x)$ is larger, so that the color distance term has a greater impact on the distance

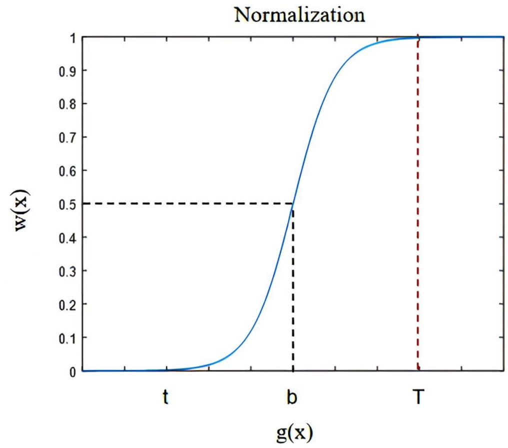


Fig. 2. Normalization function.

measurement. The color distance term $I(x, k)$ is the main basis for the division of pixel x, so that the pixel x is divided into more color In the similar class k_1, the superpixel boundary and the image boundary are further ensured.

3.2.2. Penalty term based on linear path

The linear path was introduced into superpixel segmentation by Remi Giraud [20]. It refers to the path taken by the connection between two pixels. As shown in Figure 4, the connection between the pixel point x and the cluster center k passes through The pixels constitute a linear path P_x^k from the pixel point x to the cluster center k.

$\sum_{x \in P_x^k} w(x)$ in formula 7 represents the sum of boundary probabilities accumulated on the linear path from the pixel x to the cluster center k, and λ_3 is a constant. As shown in Fig. 4, when passing the edge pixel x of the image, since $w(x)$ is larger, $d_L(x, k)$ is increased. That is, if the line from the pixel to be divided to the cluster center crosses the edge of the image, the distance term will increase. Therefore, this item can restrict the line from the seed point to the cluster center to cross the boundary of the object, and this item further ensures the uniformity within the superpixel

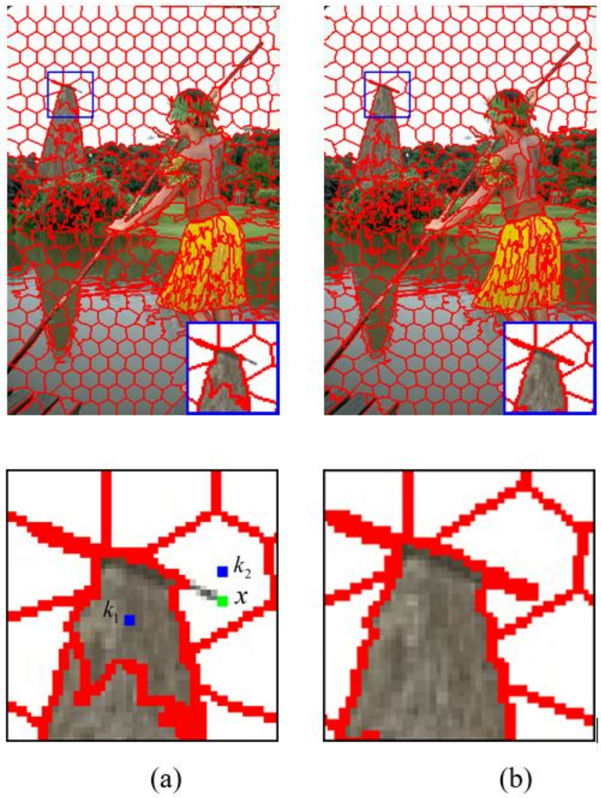


Fig. 3. Description of the boundary probability distance term. (a) Superpixel generated without the boundary probability distance term; (b) Superpixel generated using the boundary probability distance term.

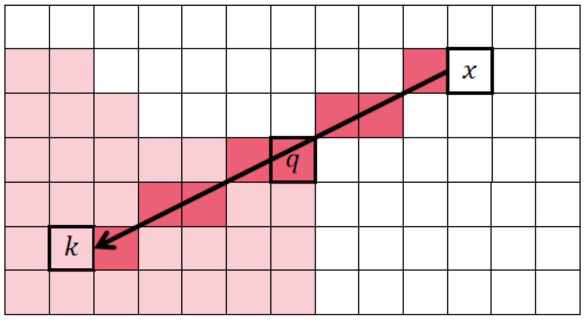


Fig. 4. Schematic diagram of linear path.

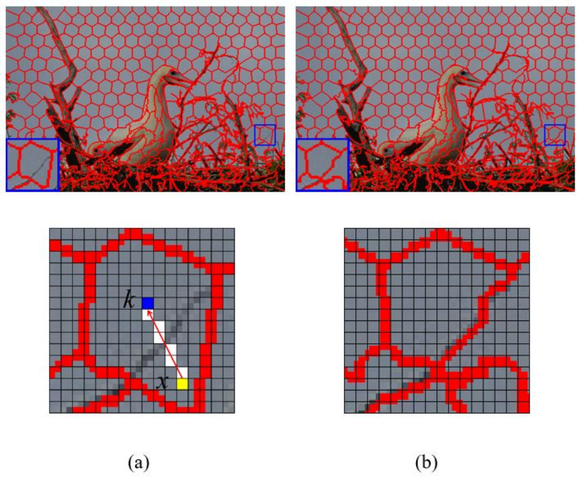


Fig. 5. Description of linear path penalty terms. (a) Superpixel generated without adding this item; (b) Superpixel generated by adding this item.

class.

$$d_L(x,k) = e^{\frac{\sum_{x\in p_x^k} w(x)}{\lambda_3}} \tag{7}$$

The ideal superpixel edge should be close to the edge of the object in the image. However, in most cluster-based superpixel generation algorithms, only considering the relationship between a single pixel and the cluster center in color space and location space, it may lead to The pixels are incorrectly classified. As shown in Figure5, if the distance metric formula 4 based on linear path penalty is not added for clustering, then for the yellow pixel x, the blue cluster center k is adjacent to it in spatial distance. The distance is also similar to it. If only the distance term in formula 4 is used for distance measurement, the pixel x will be erroneously classified into the class to which the cluster center k belongs.

When a penalty term based on a linear path is added to the distance measurement formula, since the linear path from x to k crosses the boundary of the image, the term will increase, so that x and k measured by formula 3 The distance between them increases, so that x will not be classified as k. Therefore, this item can make up for the defect that the general distance measurement formula only considers the information of a single pixel, and the linearity between the pixel point and the cluster center The information on the path is taken into account, which further ensures the uniformity of the pixel properties within the class.

3.2.3. Seed point update method

The superpixel generation algorithm based on K-means uses the mean value of all pixels in the cluster to update the cluster center when updating the cluster center. As a result, each cluster center update process will be interfered by pixels with low color space similarity to the cluster center, which makes the update of the cluster center inaccurate and affects the distance measurement in subsequent iterations, resulting in the gradual accumulation of errors, and ultimately resulting in inaccurate superpixel segmentation.

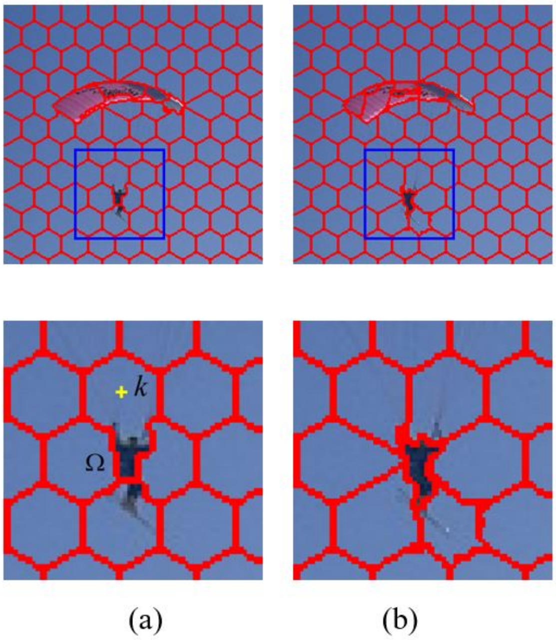


Fig. 6. Superpixel generated using different seed point update strategies (a) The result of using all pixels to update the seed point (b) The result of using the weighted average to update the seed point.

In order to solve this problem, the algorithm in this paper uses a weighted average strategy to update the cluster centers. First remove the pixels in the cluster that are too low in color space with the cluster center according to the three sigma principle, and then update the cluster center in a weighted average manner, giving the cluster an unreliable degree of similarity to the cluster center. Pixels have smaller weights to weaken their influence on the cluster centers during the iteration process. The superpixel cluster center can be obtained by formula 8.

$$C_{k,i} = \sum_{x \in \Phi(k,i)} w_{k,i}(x) p(x) \tag{8}$$

$$\Phi(k,i) = \{x \mid L(x) = k \text{ and } |h(x) - \mu_i| < 3\sigma_i\} \tag{9}$$

$$w_{k,i}(x) = \frac{e^{-[h(x)-h_k]^2}}{Z} \tag{10}$$

$$Z = \sum_{x \in \Phi(k,i)} e^{-[h(x)-h_k]^2} \tag{11}$$

In formula 8, $C_{k,i}$ represents the k superpixel seed point in the i iteration process, $p(x)$ represents the color vector (L_x, A_x, B_x) or position vector $(r(x), c(x))$ of the pixel point x, and $w_{k,i}(x)$ represents the pixel point x during the i th seed point x update process weights.

In formula 9, $h(x)$ represents the gray level of pixel x, μ_i and σ_i are the mean and variance of the gray values of pixels within the class in the i iteration, and $\Phi(k,i)$ is the cluster center after removing unreliable pixels according to the three sigma principle k pixel collection.

Formula 10 is the weight function of the weighted average of the pixels within the class, h_k represents the gray value of the cluster center k, $w_{k,i}(x)$ can measure the gray similarity of the pixel point x and the cluster center k, when the two gray levels are more similar, the larger $w_{k,i}(x)$.Z is the normalization parameter, so $\sum_x w_{k,i}(x) = 1$

According to statistical principles, 99.73% of the data that conform to the normal distribution are within the interval of three standard deviations from the average. In the iterative update process, the pixels within the class and the average gray value difference greater than three times the standard deviation are eliminated according to Equation 9, to ensure that more reliable pixels are used for seed point update. Then the remaining pixels are weighted and averaged using the weight function $w_{k,i}(x)$ based on gray-scale similarity. The pixels with low color space similarity to the seed points are given smaller weights, thereby reducing the effect caused by the iteration process. interference.

Figure 6 shows the results of superpixel generated using different seed point update strategies. Figure 6(a) is the result of using all pixel update methods. It can be seen that color unevenness occurs in this type of cluster. This is because in the initial iteration, the black area Ω is erroneously classified into the superpixel k. The color space of this area and the cluster center is quite different. When the cluster center is finer, the cluster center is in the color space. The value of is the mean value of the pixels in the black area Ω and other areas in the cluster, so the cluster center is disturbed by the black area Ω, causing the update result to be inaccurate. Furthermore, in the next iteration, the distance measurement from the pixel point to the cluster center is inaccurate. As the iteration error gradually accumulates, the resulting superpixel cannot be closely attached to the edge of the image. Figure 7a is the result of using the weighted average update method. In the update process of Figure 7b, according to formula 8, the pixels with low similarity to the cluster center are eliminated, and then according to the pixel point and the cluster center The similarity of the gray-scale features gives the pixels corresponding weights, so in the process of cluster center iteration, it can be less interfered by abnormal pixels, and the final generated superpixel can be closer to the edge of the object.

3.3. The method steps

The superpixel segmentation method in this paper is mainly divided into the following three steps:

Fig. 7. Comparison of superpixel vision generated by different superpixel algorithms.

1. In order to improve the quality of the superpixel generation in the texture area, the image is preprocessed by side window filtering to smooth the texture area while maintaining the edges.

2. K initial seed points are uniformly generated on the image in a regular hexagonal manner. In order to avoid the seed points from falling on the boundary, they are moved to the pixel point position with the smallest gradient value in the eight neighborhoods. These seed points are used as the initial Superpixel cluster center.

3. The process of superpixel generation is the process of marking all pixels in the image. In this paper, the distance formula defined by formula 2 is used to measure the distance from the seed point to the cluster center, and the pixel is marked as the label of the superpixel with the shortest distance from it. After the marking is completed, formula 8 is used to update the cluster centers. Repeat the above process several times until the superpixel converge. After the iteration is over, the superpixel area adjacent to the isolated pixel is merged.

4. Experimental results and analysis

We will use this algorithm and the current mainstream superpixel algorithm to conduct a comparative experiment to verify its effectiveness. This section compares the algorithms proposed in this paper with ERS [13], SLIC [20], DBSCAN [21], SSBC [23], Turbopixel [26], LSC [27], WSS [28] and other algorithms. The results are calculated using open source codes online and using their default configuration parameters. Among them, SLIC, LSC, DBSCAN, SSBC, and WSS are all superpixel segmentation algorithms based on clustering, while ERS and Turbopixel are superpixel algorithms based on graph cutting. The data set used in this section is the Berkeley Segmentation Database [29], which contains the test set, training set and val set. The three sets contain 200 images, 200 images and 100 images respectively. The data set contains two resolution images, 481 x 321 and 321 x 481, each image has a corresponding 6 artificially labeled segmentation results (ground truth).

This section first compares the visual effects of several superpixel algorithm segmentation results to point out their respective advantages and disadvantages, and then introduces several commonly used indicators for judging the performance of superpixel segmentation algorithms, and proves the algorithm by analyzing objective experimental data and various indicators. Finally, the algorithm in this paper is compared with the other algorithms mentioned above in the task of saliency detection to further verify the accuracy of the algorithm in this paper. All tests in this section are completed under the following parameter configuration, a=3, b=8, $\lambda_1 = 0.4$, $\lambda_2 = 2$, $\lambda_3 = 0.25$.

4.1. Subjective visual comparative analysis

Figure 7 shows the superpixel results of DBSCAN, ERS, LSC, SLIC and our algorithm. It can be seen from Figure 7 that compared with LSC, ERS, SLIC and DBSCAN using fixed weights, the superpixels generated by our algorithm can better fit the image boundary and maintain regular. This is because our algorithm can assign different weights to its attributes according to the size of the boundary probability at the pixel when calculating the distance between the pixel and the seed point. Moreover, the pixel point information passing through the line connecting the pixel point and the cluster center is considered comprehensively when calculating the distance measure. Due to the fixed weights, these algorithms make a trade-off between bounds fitting and shape regular. The more regular the generated superpixels are, the harder it is for the boundaries of the superpixels to fit the image boundaries. The higher the degree of fit between the generated superpixel boundary and the image boundary, then the superpixel must be difficult to ensure regularity. For example, although the superpixels generated by the entropy rate-based superpixel segmentation algorithm ERS can also closely fit the edge of the image, the shape of the superpixels generated by this algorithm is the most irregular compared with other algorithms. The superpixels generated by the DBSCAN and SLIC algorithms have the most regular shape, but their boundaries fit poorly. LSC has made improvements on the basis of SLIC, taking into account both boundary fit and regular shape, but there is still room for improvement in its boundary fit. Compared with these algorithms, our method achieves better results in boundary fitting and shape regularization.

4.2. Numerical comparison

In order to objectively quantify the quality of superpixel generated by the superpixel generation algorithm, scholars have proposed several evaluation criteria, including Boundary recall (BR, edge recall rate), under segmentation error (UE, under segmentation rate), and achieveable segmentation accuracy (ASA, Segmentation accuracy) is the evaluation index of three commonly used superpixel generation algorithms. This section uses these three indicators to evaluate the method in this paper and other superpixel methods.

The boundary recall rate (BR) is an important indicator used to measure the degree of fit between the superpixel and the image boundary. It calculates the proportion of the artificially labeled ground truth that falls within the λ pixel width area around the superpixel boundary, and λ is usually taken as 2. The higher the value of BR indicates that the superpixel generated by the algorithm can be closer to

Algorithm 1: Basic steps of superpixel generation algorithm
Input: Image I, The expected number of superpixels to generate K Output: All pixel markers
1. Side-window filtering is performed on the input image. 2. K initial seed points are evenly generated on the image in the form of a regular hexagon. 3. Repeat 4. According to formula 2, each pixel is divided into the category closest to it. 5. Update the superpixel clustering center according to formula 8. 6. Until Superpixel convergence 7. Merge smaller superpixels according to the principle of closest color.

the edge of the image.

The under-segmentation ratio (UE) can be used to measure the proportion of pixels that are not accurately identified in the segmentation result of the super-pixel generation algorithm relative to the ground truth. The standard segmentation of an image is area $\{G_i \mid i = 1, 2, \cdots, M\}$, and a superpixel is segmented into superpixel $\{S_l \mid l = 1, 2, \cdots, K\}$, the USE calculation formula of the whole image is shown in formula 12 :

$$USE = \frac{1}{N}\left[\sum_{i=1}^{M}\left(\sum_{\{S_i \| S_i - G_i \| > B\}} \text{Area } (S_l)\right) - N\right] \quad (12)$$

Where N is the number of all pixels, Area (S_l) is the area of the superpixel S_l, and B is the area of the smallest number of overlapping regions. In this article, set B to 5% of Area (S_l). The smaller the USE value means that more objects are recognized in an image.

Segmentation accuracy (ASA) is used to judge whether the object in the image is correctly identified. In other words, ASA calculates the highest recognition rate by labeling each superpixel with the standard segmentation with the largest overlap area. This metric is defined as shown in Equation 13. The larger the ASA value, the more correctly identified objects in an image.

$$ASA = \frac{\sum_{l_l} \arg_i \max |S_l \cap G_i|}{\sum_i |G_i|} \quad (13)$$

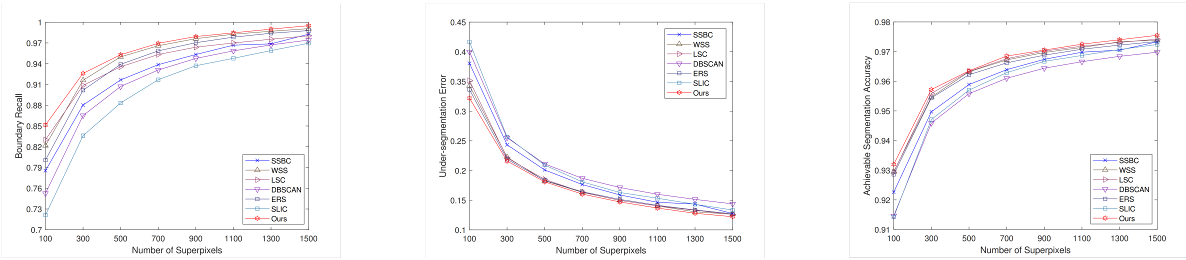
Figure 8 shows the numerical comparison of different algorithms. As shown in Figure 8(a), it can be seen that the algorithm proposed in this paper has obvious advantages in edge recall rate compared with other algorithms. The reason is that the distance measurement formula designed in this paper passes Add a weight function item to the original distance measurement formula to assign different weights to its color information and spatial information according to different pixels, so that the pixels at the edge use the color distance as the main classification basis when clustering, so the generated Superpixel can closely fit the image boundary. In addition, in the distance formula, the accumulation of boundary probabilities on the line from the pixel to the cluster center can punish the behavior of clustering across the boundary of the object, which further improves the boundary fitting rate.

Table 1 shows the numerical results of BR, ASA, and UE obtained by the algorithm of SLIC, LSC, Turbopixel, ERS, DBSCAN, SSBC, and WSS on the BSD500 data set when the number of superpixel is 500. When the number of superpixel is 500, the algorithm proposed in this paper can achieve the best results on the three indicators of BR, ASA, and UE. This shows that compared with the superpixel generated by other algorithms, the superpixel generated by the algorithm in this paper are more able to paste. Integrating the image boundary, the superpixel generated by this algorithm can be correctly identified more objects in the image.

Figure 9 shows the running efficiency of the different algorithm. It can be observed that the running time of our algorithm is better than that of LSC, TP, ERS, and SSBC methods, and under the same order of time, our algorithm achieves better results.

5. Conclusion

In this paper, we propose a boundary-preserving algorithm for superpixel segmentation. The core idea is to use a new distance metric function to assign different weights to the color distance items and spatial distance items of different pixels, so that the superpixels fit the boundary in the image boundary area and have a regular shape in the flat area. In addition, we also designed a new cluster center update strategy to reduce the interference of abnormal pixels on the cluster center update, thereby improving the accuracy of the cluster center update. Compared with current methods, the superpixels generated by our algorithm can better fit the image boundary and keep the regular shape. In the future, we consider using a new generation strategy to further improve the boundary fitting effect of superpixels and enhance the regularity of superpixels. For example, the whole image is adaptively divided into different regions, and different strategies are adopted for different regions.



(a) BR **(b)** UE **(c)** ASA

Fig. 8. Numerical comparison of different algorithms. (a) BR index comparison; (b) UE index comparison; (c) ASA index comparison.

Table 1. Experimental data of different algorithms on the BSD500 data set when the number of superpixel is 500.

	SLIC	LSC	TP	ERS	DBSCAN	SSBC	WSS	Ours
BR	0.883	0.935	0.790	0.939	0.907	0.916	0.949	0.953
ASA	0.957	0.963	0.951	0.962	0.955	0.958	0.961	0.963
UE	0.209	0.185	0.225	0.183	0.211	0.201	0.183	0.181
Time(S)	0.0874	0.301	3.9671	0.8988	0.0564	2.1675	0.2358	0.2642

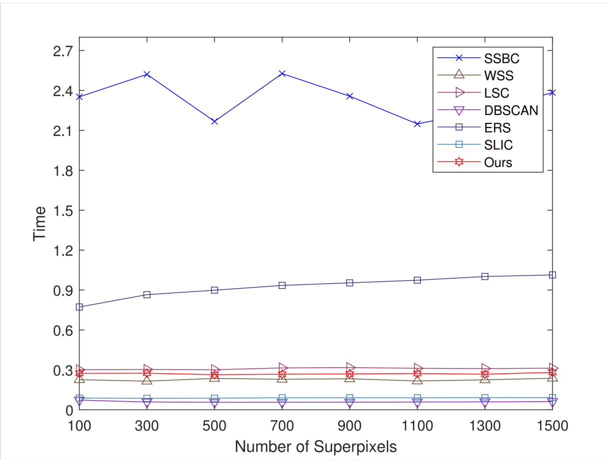


Fig. 9. Running efficiency of the different algorithm.

Acknowledgements

We thank the reviewers and editors for their constructive comments. This work was supported by the Natural Science Foundation of Shandong Province under Grant No. ZR2017MF03.

References

[1] X. Ren and J. Malik. "Learning a classification model for segmentation". In: *Computer Vision, IEEE International Conference on*. **2**. IEEE Computer Society. 2003, 10–10. DOI: 10.1109/ICCV.2003.1238308.

[2] B. Peng, L. Zhang, and D. Zhang, (2011) *"Automatic image segmentation by dynamic region merging"* **IEEE Transactions on image processing** ***20***(12): 3592–3605. DOI: 10.1109/TIP.2011.2157512.

[3] Y. Xu, X. Gao, C. Zhang, J. Tan, and X. Li, (2022) *"High quality superpixel generation through regional decomposition"* **IEEE Transactions on Circuits and Systems for Video Technology**: DOI: 10.1109/TCSVT.2022.3216303.

[4] X. Ma, X. Li, Y. Zhou, and C. Zhang, (2021) *"Image smoothing based on global sparsity decomposition and a variable parameter"* **Computational Visual Media** ***7***: 483–497. DOI: 10.1007/s41095-021-0220-1.

[5] L. Wang, Y. Shoulin, H. Alyami, A. A. Laghari, M. Rashid, J. Almotiri, H. J. Alyamani, and F. Alturise. *A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images*. 2022. DOI: 10.1002/gdj3.162.

[6] S. Karim, Y. Zhang, S. Yin, A. A. Laghari, and A. A. Brohi, (2019) *"Impact of compressed and down-scaled training images on vehicle detection in remote sensing imagery"* **Multimedia Tools and Applications** ***78***: 32565–32583. DOI: 10.1007/s11042-019-08033-x.

[7] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. "Saliency filters: Contrast based filtering for salient region detection". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, 733–740. DOI: 10.1109/CVPR.2012.6247743.

[8] X. Pan, Y. Zhou, F. Li, and C. Zhang, (2016) *"Superpixels of RGB-D images for indoor scenes based on weighted geodesic driven metric"* **IEEE transactions on visual-**

ization and computer graphics 23(10): 2342–2356. DOI: 10.1109/TVCG.2016.2621763.

[9] A. A. Laghari, S. Yin, et al., (2022) *"How to Collect and Interpret Medical Pictures Captured in Highly Challenging Environments that Range from Nanoscale to Hyperspectral Imaging."* **Current Medical Imaging**: DOI: 10.2174/1573405619666221228094228.

[10] J. Cheng, J. Liu, Y. Xu, F. Yin, D. W. K. Wong, N.-M. Tan, D. Tao, C.-Y. Cheng, T. Aung, and T. Y. Wong, (2013) *"Superpixel classification based optic disc and optic cup segmentation for glaucoma screening"* **IEEE transactions on medical imaging 32**(6): 1019–1032. DOI: 10.1109/TMI.2013.2247770.

[11] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, (2012) *"Superpixel-based classification with an adaptive number of classes for polarimetric SAR images"* **IEEE Transactions on Geoscience and Remote Sensing 51**(2): 907–924. DOI: 10.1109/TGRS.2012.2203358.

[12] B. Alexe, T. Deselaers, and V. Ferrari, (2012) *"Measuring the objectness of image windows"* **IEEE transactions on pattern analysis and machine intelligence 34**(11): 2189–2202. DOI: 10.1109/TPAMI.2012.28.

[13] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool. "Superpixel meshes for fast edge-preserving surface reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 2011–2020.

[14] D. Hoiem, A. A. Efros, and M. Hebert. "Automatic photo pop-up". In: *ACM SIGGRAPH 2005 Papers*. 2005, 577–584. DOI: 10.1145/1186822.1073232.

[15] J. Lim and B. Han. "Generalized background subtraction using superpixels with label integrated motion estimation". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, 173–187. DOI: 10.1007/978-3-319-10602-1_12.

[16] J. Shi and J. Malik, (2000) *"Normalized cuts and image segmentation"* **IEEE Transactions on pattern analysis and machine intelligence 22**(8): 888–905. DOI: 10.1109/34.868688.

[17] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. "Entropy rate superpixel segmentation". In: *CVPR 2011*. IEEE. 2011, 2097–2104. DOI: 10.1109/CVPR.2011.5995323.

[18] J. Shen, Y. Du, W. Wang, and X. Li, (2014) *"Lazy random walks for superpixel segmentation"* **IEEE Transactions on Image Processing 23**(4): 1451–1462. DOI: 10.1109/TIP.2014.2302892.

[19] Y.-J. Gong and Y. Zhou, (2017) *"Differential evolutionary superpixel segmentation"* **IEEE Transactions on Image Processing 27**(3): 1390–1404. DOI: 10.1109/TIP.2017.2778569.

[20] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, (2012) *"SLIC superpixels compared to state-of-the-art superpixel methods"* **IEEE transactions on pattern analysis and machine intelligence 34**(11): 2274–2282. DOI: 10.1109/TPAMI.2012.120.

[21] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, (2016) *"Real-time superpixel segmentation by DBSCAN clustering algorithm"* **IEEE transactions on image processing 25**(12): 5933–5942. DOI: 10.1109/TIP.2016.2616302.

[22] Z. Ban, J. Liu, and L. Cao, (2018) *"Superpixel segmentation using Gaussian mixture model"* **IEEE Transactions on Image Processing 27**(8): 4105–4117. DOI: 10.1109/TIP.2018.2836306.

[23] Y. Zhang, X. Li, X. Gao, and C. Zhang, (2016) *"A simple algorithm of superpixel segmentation with boundary constraint"* **IEEE Transactions on Circuits and Systems for Video Technology 27**(7): 1502–1514. DOI: 10.1109/TCSVT.2016.2539839.

[24] R. Giraud, V.-T. Ta, and N. Papadakis, (2018) *"Robust superpixels using color and contour features along linear path"* **Computer Vision and Image Understanding 170**: 1–13. DOI: 10.1016/j.cviu.2018.01.006.

[25] H. Yin, Y. Gong, and G. Qiu. "Side window filtering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, 8758–8766.

[26] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, (2009) *"Turbopixels: Fast superpixels using geometric flows"* **IEEE transactions on pattern analysis and machine intelligence 31**(12): 2290–2297. DOI: 10.1109/TPAMI.2009.96.

[27] Z. Li and J. Chen. "Superpixel segmentation using linear spectral clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, 1356–1363.

[28] Z. Li and J. Chen. "Superpixel segmentation using linear spectral clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, 1356–1363.

[**29**] D. Martin, C. Fowlkes, D. Tal, and J. Malik. "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*. **2**. IEEE. 2001, 416–423. DOI: 10.1109/ICCV.2001.937655.