

## An Improved SLIC Superpixels using Reciprocal Nearest Neighbor Clustering

Junrui Lv

*School of mathematics and computer science, Panzhihua University,  
Si chuan Panzhihua 617000, China  
junrui003@126.com*

### **Abstract**

*An improved SLIC method using uniform segmentation and reciprocal nearest neighbor (RNN) clustering is presented in this paper. This approach is made of two steps. First, image is segmented to a large number of regular homogeneous and small regions which are similar to cell. Second, instead of the original image pixels, small regions segmented are regarded as input of RNN clustering. A new similarity criterion is decided by regional diversity of average value normalized and variance. Regional constraint filter limited the large size of superpixel guarantees the uniformity and compactness of superpixel. Finally, the regions in a small range of distance are merged by RNN clustering. Results of experiment on BSDS 500 dataset of natural images show the proposed method has advantages of high boundary recall and low under-segmentation error over SLIC on small numbers superpixel.*

**Keywords:** Superpixel, Multiscale, SLIC, K-means Clustering

### **1. Introduction**

Superpixels provide a convenient primitive for use in computer vision applications. Superpixels algorithm is the process that neighboring pixels with the same or similar features in image are grouped into perceptually meaningful homogeneous regions. Ren and Malik, *et al.*, first proposed superpixel concept that a large number of regions in image are segmented according to adjacent homogeneous pixels as a unit of process [1]. Regions obtained by superpixel algorithm might be a more natural representation of reality. Meanwhile, superpixels provide a compact representation of an original image, which have a great improvement in computational efficiency, lower running time and save memory cost because of transition from hundreds of thousands of pixels to only a few hundred superpixels. Therefore, it is an essential tool to image preprocessing in the field of image processing and computer vision. In the past few years, superpixel is applied to preprocessing in more effective approach from a mass of literatures increasingly, such as image segmentation [2], image parsing [3], object localization [4]. A good approach for superpixel defined for major applications is difficult; we believe the following properties are generally desirable.

1) Superpixel algorithm as a preprocessing step should be computationally efficient. It reduces the complexity of images and should be fast to compute, memory efficient, less to set parameters by user.

2) Superpixels resulting of an over-segmentation should conserve most structures in the image and its boundaries adhere well to image boundaries. Loss in moving from the pixel-grid to the superpixel map is as little as possible.

3) All applications based on superpixel should obtain benefits of speed and quality of results.

In recent years, because of the needs of massive applications, some traditional superpixel algorithms can't satisfy high processing of images in computer vision on

efficiency of computation and quality of the results. High computational complexity and too many parameters are the main factors that limit the development of superpixels. Therefore, superpixel algorithm has become a new research focus.

In general, there are mainly two schools of thought on generating superpixel, The respective representative of which are graph cuts from the whole to locality and clustering methods according to the characteristic of color, texture and boundary. Methods based on the graph cuts are realized by defining a minimum cost function on graph and treating each pixel of image as a node in a graph, similarity of which is edge weight between neighboring nodes. Superpixels are generated by segmenting nodes into two parts iteratively according to globally minimizing a cost function. Its drawback is large amounts of short boundary and leading to split out many small regions.

In order to solve the problem, the method, called Normalized cuts (N-cuts), recursively partitions a graph of all pixels normalized in the image using contour and texture cues. It is high cost to find the global optimal and high computational complexity. It is not satisfied to the effective and practical requirements. Another method based on clustering aggregates pixels to regions between neighboring pixels by calculation of local structure and boundary information, which is a kind of fast generation method. Jie Wang and Xiaoqiang Wang proposed an Edge-Weighted centroidal voronoi Tessellations clustering energy algorithm to generate superpixels [6]. Its main idea is k-means clustering, and it is efficient with low computational cost and nicely preserving local image boundaries. But final segmentation effect relies mainly on the consistency of partitioning polygon region boundary and image edge in the first step. Alex Levinshtein, *et al.*, [7] proposed a method of Turbopixel to generate superpixels. It progressively dilates a set of seed locations using level set based on geometric flow. Geometric flow relies on local image gradients, aiming to regularly distribute superpixels on the image plane. It is better than N-cuts on the efficiency and effect but computational cost of the algorithm is proportional to image size.

Achanta, *et al.*, introduce an superpixel algorithm, simple linear iterative clustering (SLIC), which adapts a k-means clustering approach to generate superpixels in the 5-D space defined by weighing of the value  $L$ ,  $a$ ,  $b$  of the CIELAB color space and the distance similarity of the  $xy$  plane [7]. In the initial stages clustering centers of  $k$  which is the desirable superpixel numbers of user are arranged uniformly in the image, the surrounding pixels of clustering center belongs to recent clustering. Then pixels belonging to clustering center are moved according to the distance similarity and clustering centers are re-orientated by recount. After several iterations, algorithm converges to the global optima, namely, the best position of clustering center.

SLIC algorithm had compared with Turbopixel, Normalized cuts, mean drift, watershed segmentation algorithm on the Berkeley benchmark dataset [9]. Experiments show SLIC is significantly more efficient than competing methods by measure of standard boundary recall and under-segmentation error. But SLIC algorithm did not consider the characteristic of plenty of edge information of regions of interest in image, and only  $k$  initial cluster centers are simply sampled on a regular grid by uniform distribution, which leads to loss of abundant weak edge information. Due to high quality, compactness, nearly uniform superpixels playing an important role in performance of the subsequent segmentation, an improved SLIC method by combining both uniform segmentation and RNN clustering to generate superpixels is proposed in this paper and ensures that the performance of time and accuracy has nothing with  $k$ .

## 2. The Related Techniques

This section firstly analyses the advantages and disadvantages of SLIC superpixels, then and represents RNN clustering algorithm.

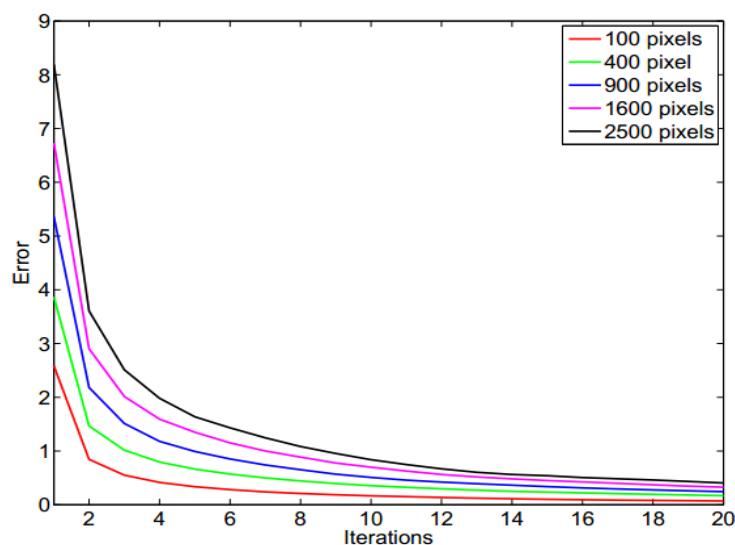
### 2.1. SLIC

Given an image of  $N$  pixels, by default, the only parameter of SLIC is  $k$ , the desired number of approximately equally-sized superpixels in SLIC algorithm. K-means clustering in SLIC was applied to cluster by a similarity measure of the distance weighting between color similarity normalizing in the CIELAB color space and spatial proximity normalizing in  $xy$  coordinates in a region of  $2S \times 2S$ , where  $S = \sqrt{N/K}$ . Pixels are only fallen in one of adjacent eight clusters.

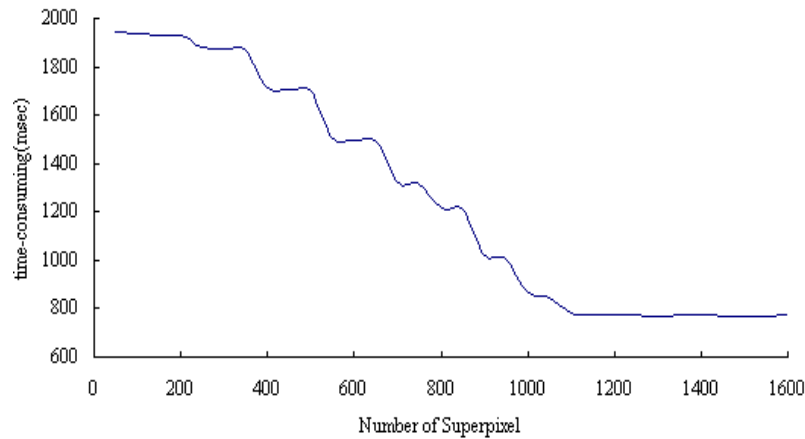
K-means clustering is the frequent practical method. K-means algorithm is an effective feature space clustering analysis tool, which has a wide range of applications in many areas of computer vision and shows good robustness and high processing efficiency. But the clustering results are very sensitive to the choice of initial cluster, computational complexity is proportional to clustering point distribution amount, and it is apt to ultimately converge to a local optimal solution.

Experiments found that the search scope of each k-means clustering center is inversely proportional to  $k$  for the same image through analysis of SLIC. When the value of  $k$  increases, it reduces each clustering calculation and quickly converges to the global optimal. In other words, the bigger the value  $k$  is, the more accurate edge segmentation is. The value  $k$  has an influence on computational complexity and reserves boundary integrity. However, if the value  $k$  is exceedingly large, it will lead to serious over-segmentation of smooth region in image. Therefore, the reasonable selection of superpixel number is an important factor of SLIC algorithm performance.

Figure 1(a) shows that the convergence speed with residual error is different from different size of the image in the same value of  $k$ , and the image of 100 pixels only need 6 times iteration, and that of 2500 pixels need 20 times iteration to converge to the stable states. So, images of the same size and residual error can effectively reduce iterations by increasing  $k$ . The value of residual error being 0.8 in Figure 1(b) demonstrates the spending time curve that SLIC generates superpixels of image of  $800 \times 600$  pixels with different values of  $k$ . the practical time complexity is  $O(NKI)$ , where  $I$  is the number of iterations and is irrespective of  $k$ . Although segmented regions increase with the value of  $k$ , its iteration times decrease. Therefore, from the overall point of view, computation time does not increase. Iterative k-means was implemented by CUDA framework of GPU by Carl Yu [10]. Therefore, the analysis finds that the overall computing time to measure through residual error should be more reasonable, which still can effectively reduce iterations on condition of guaranteed performance.



(a) Residual Error and Iterative Times with Different Pixels



(b) Running Time by Superpixels Number  $k$  in SLIC

Figure 1. SLIC Analysis

## 2.2 Reciprocal nearest Neighbor Clustering

A clustering algorithm based on reciprocal nearest neighbors (RNN), which is an agglomerate hierarchical clustering algorithm, is introduced by B. Leibe [11]. It is applied to cluster image features using SIFT, SURF or PCA-SIFT descriptors and recognize object class. Recognition performance is better than that of  $k$ -means clusters. In the meanwhile, it has a good clustering effect on low dimensional or high dimensional data.

An overview of reciprocal nearest neighbors is that it is based on the construction of RNN pairs of vectors  $x_i$  and  $x_j$ , so that  $x_i$  is the NN to  $x_j$ , and vice versa. As soon as a RNN pair is found, it can be agglomerated. It is an efficient implementation that ensures that RNN can be found with recomputation as little as possible. This is achieved by building a NN chain, which consists of an arbitrary vector, followed by its NN, which is again followed by its NN among the remaining points and so on. Hence, a NN chain of length  $L$  can be defined as the sequence of vectors  $\{x_1, x_2=NN(x_1), \dots, x_{l-1}=NN(x_l), x_l=NN(x_l)\}$ , where  $NN(x_i)$  is the NN of  $x_i$ . It is worth noting that the distances between adjacent vectors in the NN chain are monotonically decreasing, and that the last pair of nodes is RNN.

The RNN algorithm starts with a vector at random. A NN chain is then built. When a RNN pair is found, *i.e.*, no more vectors can be added to the current chain, the corresponding clusters are merged if their similarity is above a fixed cut-off threshold  $T$ , otherwise the algorithm discards the whole chain. This way of merging clusters can be applied whenever the distance matrix  $D$  satisfies the reducibility property,  $dist(C_i, C_k) \leq \min\{dist(C_i, C_j), dist(C_j, C_k)\} \leq dist(C_{i \cup j}, C_k)$ , where  $dist(C_i, C_k)$  is the distance between clusters  $C_i$  and  $C_k$ , and  $C_{i \cup j}$  is the cluster after merging  $C_i$  and  $C_j$ . This property guarantees that when RNN are merged, the NN relations for the remaining chain members are unaltered; therefore they can be used for the next iteration. When the current chain is empty or has been discarded, a new arbitrary point is selected, and a new NN chain is started.

## 3. The improved SLIC Approach using RNN clustering

The advantages and disadvantages of the SLIC superpixels are presented in above section. Our proposed method is that image is uniformly subdivided into an increasing number of homogeneous small regions by virtue of  $k$ -means clustering, and then RNN region merge only computes distances from a cluster center to another within a  $S \times S$  region. It is not until the region number reduced to  $k$ .

### 3.1. Uniform Segmentation

The centers of  $r$  for  $k$ -means clustering are uniformly allocated without overlap on image plane, and the number of  $r$  far outweigh the number of  $k$ . After several iterations, image is evenly divided into  $r$  regions of homogenization. Similarity criterion which is weighted both the distance of Lab values in CIELAB color space and the distance of image plane between clusters is inspired by SLIC algorithm. Number of  $r$  is dependent on both  $k$  and  $\varepsilon$  which is coefficient of the final segmentation number and the number of initial segmentation of  $k$ -means clustering. Setting the value of  $\varepsilon$  is generally between 3 and 10. Through a lot of cluster centers, image is polymerization into the number of small homogeneous regions  $r$ . According to the characteristic of  $k$ -means clustering of small region converged optimally, this approach not only realizes the consistency of segmented region boundary and image edge but also reduces under-segmentation.

### 3.2. RNN Region Merging

It is necessary to merge parts of smooth regions after initial segmentation. Region merging in the agglomeration hierarchical clustering image segmentation is indispensable. It is some traditional methods that firstly build similarity matrix weighted between color similarity and boundary distance, and then search regional neighbors from similarity matrix to merge. The establishment of similarity matrix and the merge update need time complexity  $O(n^3)$ , where  $n$  is the number of clustering centers. Its time complexity is high. Therefore, this paper puts forward a RNN region merging method that random select starting center and look for a pair of clusters meeting RNN of superpixel constraint to combine. This method can guarantee local optimization of combination.

After uniform segmentation, the issue on the generation of superpixels is converted into the process of set  $X=\{x_i, i=1, \dots, r\}$  combined into  $\mathfrak{R}=\{C_j, j=1, \dots, m\}$ , where  $x_i$  is a six dimensional features vector of extraction received from uniform segmentation,  $x_i=[\mu_L, \mu_a, \mu_b, \sigma, a, y]$ ,  $\mu_L, \mu_a, \mu_b$  is respectively the mean value of pixel  $[L \ a \ b]^T$ ,  $\sigma$  is variance of region,  $a$  is the number of pixels and  $y$  is a vector of two dimensional space coordinates.

Regional similarity criterion is the basis of region merging. Color feature is an important way to measure similarity of regions. We propose a novel similarity measure that fuses both differences of mean value of regional color and variance of regional color, then and multiplies by sigmoid function of plane distance between regions. The equation is defined as follows.

$$g^{lab}(R_i, R_j) = (\mu_{Li} - \mu_{Lj})^2 + (\mu_{ai} - \mu_{aj})^2 + (\mu_{bi} - \mu_{bj})^2$$

$$g^d = \frac{1}{(1 + \exp(-c \|d_i - d_j\|))}$$

$$g(R_i, R_j) = g^d \times \sqrt{((\sigma_i^2 + \sigma_j^2) + g^{lab}(R_i, R_j))} \quad (1)$$

The values of  $\mu_L, \mu_a, \mu_b$  are respectively normalized to the scope of 0 and 1 before applying similarity measure.  $g^d$  is a sigmoid function of fine-tuned distance similarity, the lower its value is, the higher similarity of both regions is.  $c$  is the slope of the sigmoid function. The metric standards can not only distinguish between regional differences but also can reduce computation cost.

In order to guarantee uniform and compactness superpixels and prevent large region combined, candidate regions satisfying the constraint conditions of Eq.(2) is eliminated,  $N/K$  for the average value of region area.

$$(a - N / K) > 0 \quad (2)$$

To simplify similarity computation, a new cluster of both clusters meeting merging conditions and combined updates mean value of regional color and variance of regional color. The definition of which is as follows Eq.(3-6).

$$\mu_{new} = \frac{a_i \mu_i + a_j \mu_j}{a_i + a_j} \quad (3)$$

$$\sigma_{new}^2 = \frac{1}{a_i + a_j} (a_i \sigma_i^2 + a_j \sigma_j^2 + \frac{a_i \times a_j}{a_i + a_j} (\mu_i - \mu_j)^2) \quad (4)$$

$$y_{new} = (y_i + y_j) / 2 \quad (5)$$

$$a_{new} = a_i + a_j \quad (6)$$

It is greatly improved on efficiency, as the computation of  $\mu_{new}$  and  $\sigma_{new}^2$  only uses the mean value and variance of region on color.

The below is the concrete steps of region merging.

Step 1: Initialization RNN clustering parameters. Small regions are regarded as nodes forming clustering data set container called  $X$ . Information on the mean and variance of regional color, regional area, regional two-dimensional coordinate from uniform segmentation stage are saved in own cluster structure.  $R$  is a temporary storage of clustering data. Chain  $L$  is a container for nearest neighbor clusters ( $L[last]$  is the nearest neighbor of  $L[last-1]$ ),  $lastsim$  indicates the nearest neighbor similarity value. container  $C$  stores the last result of aggregation, and initialized to  $NULL$ .  $last$  is a parameter and initially equal to zero,  $lastsim[0]$  is equal to zero,  $R$  is set to  $NULL$ .

Step 2: randomly select a cluster  $v$  from  $X$ ,  $L[last]$  is equal to  $v$ , and delete  $v$  from  $X$ , move rest clusters of  $X$  to  $R$ .

Step 3: if  $|R|+|C|+|L|>m$  is true, a cluster  $L[last]$  from  $R$  searches the RNN cluster by (1),  $s$  is the RNN cluster, and  $sim$  is the similarity of a pair of RNN cluster, otherwise, move clusters from  $R$  and  $L$  to  $C$ , then go to step 7.

Step 4: if  $sim$  is greater than  $lastsim [last]$ , let last add one and  $L[last]$  be equal to  $s$ , remove  $s$  from  $R$ , let  $lastsim[last]$  be equal to  $sim$ , otherwise go to step 5. Then redirecting step 3.

Step 5: combine the last two clusters of  $L$ , i.e.,  $L[last]$  and  $L[last-1]$  into a new cluster  $s$ , then and update mean value, variance and regional area of cluster  $s$  by (3)-(6). If  $a_s$  contents the condition of  $(a_s - N/k) > 0$ , join it to  $C$ , or insert it into  $R$ .

Step 6: let  $last$  be equal to zero, the remain in  $L$  will put back  $R$ , then let  $L[last]$  be equal to a arbitrary cluster  $v$  from  $R$ , and the cluster  $v$  deleted from  $R$ , then go to step 3.

Step 7: the algorithm ends. Cluster container  $C$  stores the final clusters.

Region merging method directly ruled out some candidate merging regions through the region area constraint conditions, and the sigmoid function of parameter using Euclidean distance between regions is regarded as similarity coefficient, then RNN region cluster is applied to merge small regions according to similarity criteria. The algorithm has advantages of no arguments, the initial cluster point choice optional. RNN algorithm can guarantee local optimization merging.

#### 4. Experiments and Analysis

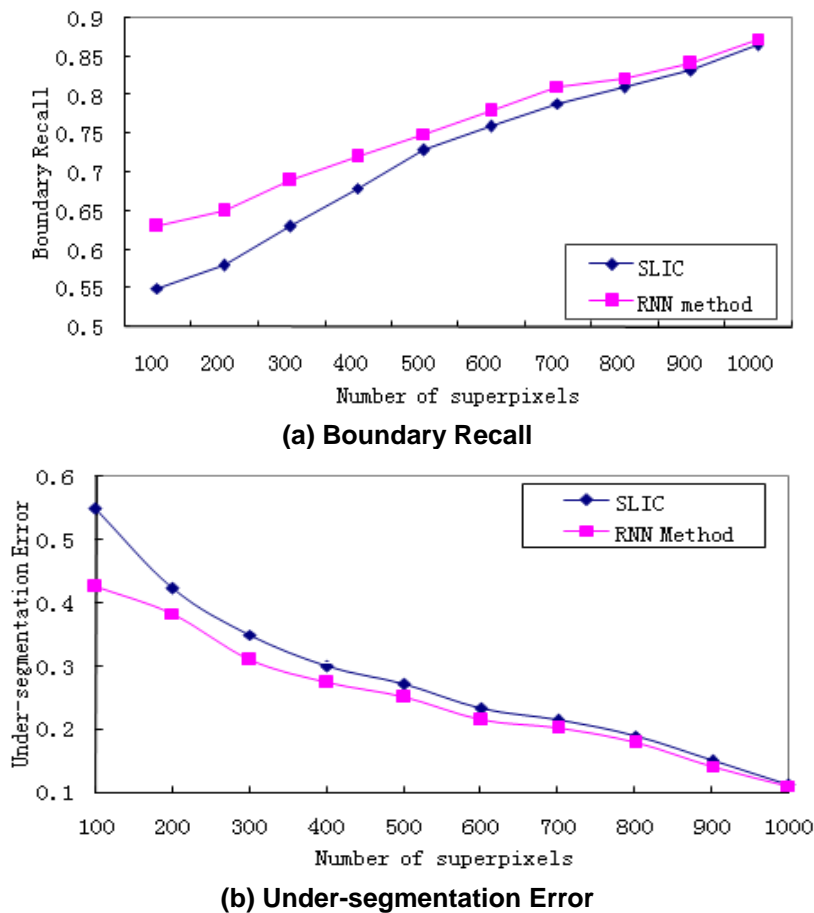
In order to examine and verify the effects of the proposed method, it is realized by C++ language, and we experiment on Berkeley segmentation data set 500 (BSDS500) and ground truth benchmark data set. It is respectively compared with SLIC algorithm by measure of segmentation accuracy and efficiency.

Both boundary recall and under-segmentation error are two measure standards of image boundary reserved edge accuracy and segmentation error. The boundary recall is calculated by using the standard measure, *i.e.*, the fraction of the ground truth boundaries which match with the superpixels boundaries. Under-segmentation error is calculated by using Eq.(7). The equation is defined as follows.

$$E = \frac{1}{Area(G_m)} \left[ \sum_{m=1}^M \left( \sum_{\{R_i | R_i \cap G_m > 0\}} |R_i| \right) - Area(G_m) \right] \quad (7)$$

where  $\{G_m\}_{m=1}^M$  represents the segmentation of the ground truth image and  $\{R_i\}_{i=1}^L$  denotes superpixels produced by the algorithm. Eq. (4) refers to a single ground truth image segment. Under-segmentation error of an image is calculated by averaging this quantity over all ground truth segments.

In the experiment, set  $\varepsilon$  be equal to 5, as shown in Figure 2, it shows that boundary recall and under-segmentation error performance comparison of the proposed method and the SLIC method. When a desired number of superpixel is small, the performance of our method is more excellent than the SLIC method, and boundary recall is obviously higher than that of the SLIC method and under-segmentation error is low. With the increasing number, it is the same performance of two methods basically.



**Figure 2. The Comparison of Performance with SLIC**

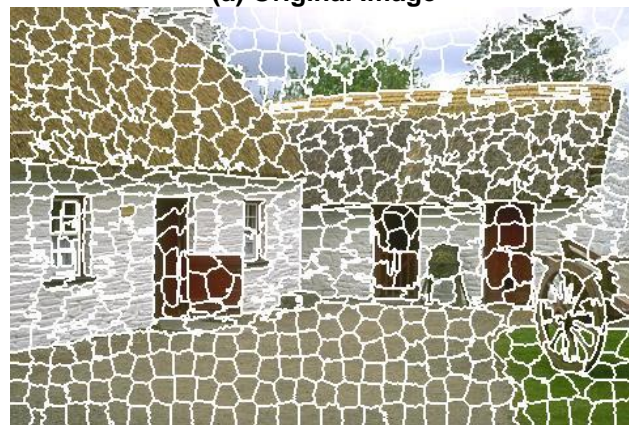
It is as shown in Figure 3 and Figure 4 that visual comparison of superpixels produced by both methods. The superpixel number of each image is approximately 600. The top images are original from BSDS500 dataset. The second line results obtained using SLIC



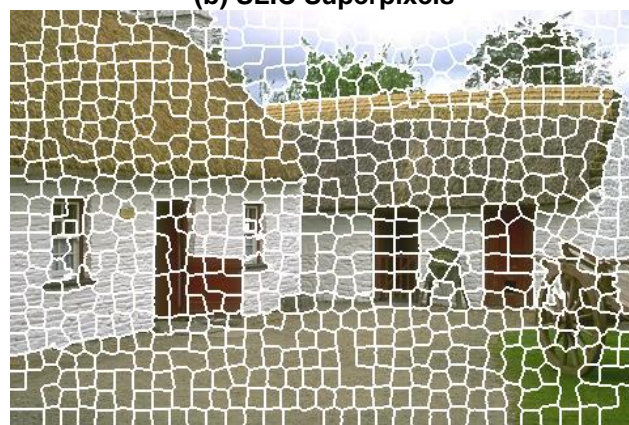
superpixels, and following results obtained from the proposed RNN method. The ability of our method to control the size, number, and compactness of the segments is more than SLIC. Through superpixels size constraint and RNN combined region algorithm, boundary recall and under-segmentation error performance is immune to the number of superpixels.



(a) Original Image



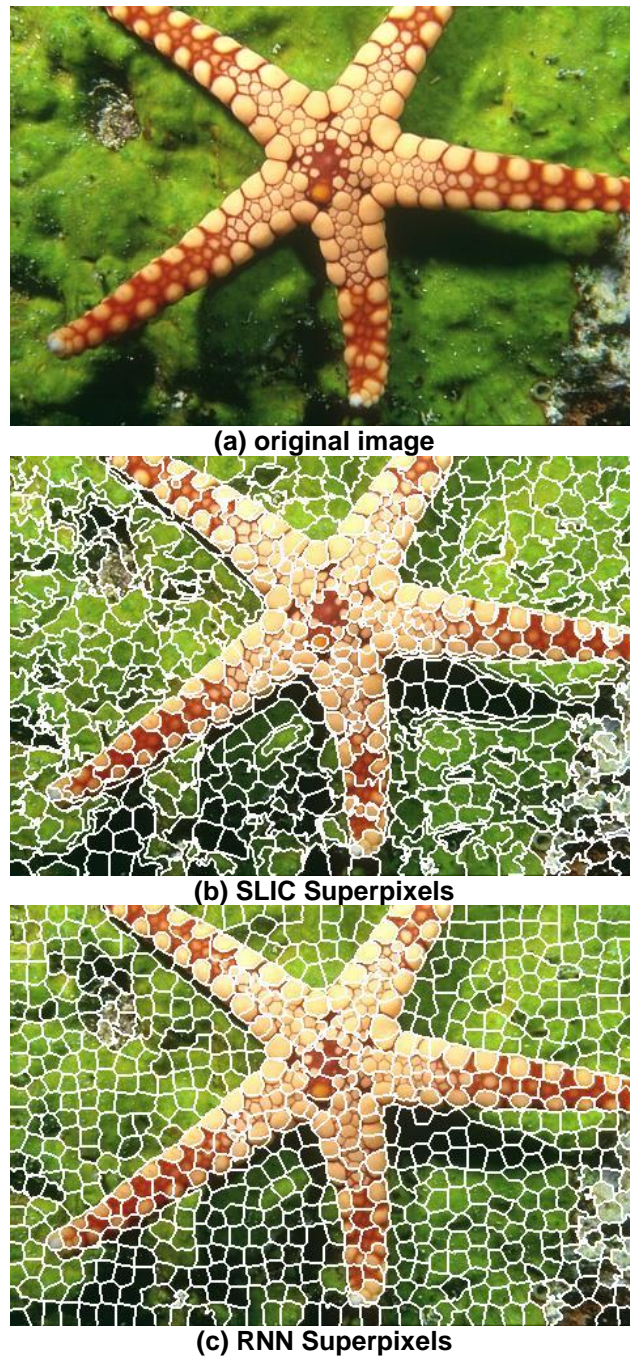
(b) SLIC Superpixels



(c) RNN Superpixels

**Figure 3. Visual Comparison of RNN Superpixels with SLIC Methods (House Image)**





**Figure 4. Visual Comparison of RNN Superpixels with SLIC Methods (Rays Image)**

## 5. Conclusion

Applications based on superpixels in the field of computer vision are becoming increasingly popular, the quality and computation efficiency of vision assignment based on superpixel are obviously better than the method based on pixel. Therefore, performance requirement of generating superpixels is more and more high. Our proposed method generated superpixel using RNN clustering region merging has an advantage of performance be independent of the number of superpixel. Our further work is that a parallel implementation of  $k$ -means cluster uniform segmentation using GPU and NVIDIA CUDA framework will guarantee computation efficiency.

## ACKNOWLEDGEMENTS

This research was partially supported by the science and technology foundation project of Panzhihua University in 2014 (No.2014YB41).

## References

- [1] X. Ren and J. Malik, "Learning a classification model for segmentation", Proceedings of the IEEE International Conference on Computer Vision, (2003) October 10-17, Washington, DC, USA.
- [2] X. He, R. Zemel and D. Ray, "Learning and incorporating top-down cues in image segmentation", Proceedings of 9th European Conference on Computer Vision, (2006) May 338-351, Graz, Austria.
- [3] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels", Proceedings of the 11th European conference on Computer vision: Part V, (2010) September 05-11, Heraklion, Crete, Greece.
- [4] B. Fulkerson, A. Vedaldi and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods", Proceedings of the International Conference on Computer Vision, IEEE Computer Society, (2009) September 670-677, Kyoto, Japan.
- [5] R. Achanta and A. Shaji, "SLIC superpixels compared to state-of-the-art superpixel methods", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, (2012), pp. 2274-2282.
- [6] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, "SLIC Superpixels", EPFL Technical Report no.149300, (2010) June.
- [7] C. Y. Ren and I. Reid, "gSLIC: a real-time implementation of SLIC superpixel segmentation", Technical Report University of Oxford, Department of Engineering, (2011).
- [8] B. Leibe, K. Mikolajczyk and B. Schiele, "Efficient Clustering and Matching for Object Class Recognition", Proceedings of the 17th British Machine Vision Conference (BMVC), (2006) September 789-798, Edinburgh, England.
- [9] S. A. Nene and S. K. Nayar, "A simple algorithm for nearest neighbor search in high dimensions", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 9, (1997), pp. 989-1003.
- [10] R. J. Lopez-Sastre, D. O. Rubio, P. Gil-Jimenez and S. M. Bascon, "Fast reciprocal nearest neighbors clustering", Signal Process, vol. 92, no. 1, (2012), pp. 270-275.
- [11] A. Levinshstein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 12, (2009), pp. 2290-2297.
- [12] J. Shi and J. Malik, "Normalized cuts and image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, (2000), pp. 888-905.
- [13] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal Voronoi tessellations", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 6, (2012), pp. 1241-1247.