Using SLIC Method for Photo Stylized Rendering

Sanyuan Zhao[1,a] , Fengxia Li[1], Qingyun Wang[2], Bo Liu[1]

[1]Department of Computer Science, Beijing Institute of Technology, Beijing 100081, China

[2]Beijing Simulation Center，Beijing 100854, China

[a]email: zhaosanyuan@bit.edu.cn

Keywords: Photo Stylized Rendering; SLIC

Abstract. Our work present a novel approach to deal with photo stylized rendering. Using the SLIC method, the stylized process will be faster and is able to keep the effect. The regions are constructed using SLIC, which is used in image segmentation field. After that we extract the color value for each region and render the image. The method shows satisfying experimental result.

Introduction

photo stylized rendering, especially watercolor, also aquarelle from French, is a painting method in which the paints are made of pigments suspended in a water-soluble vehicle. Its painting technique is special in the realm of painting. It is more accurate to say that watercolor techniques are unique to watercolor. Water is an active and complex partner in the watercolor painting process, changing both the absorbency and shape of the paper when it is wet and the outlines and appearance of the paint as it dries. When the paint is finished, its color is rich in sense of gradation and it looks like that the object is separated into pieces.

Owing to the qualities of watercolor mentioned above, a novel idea is proposed for processing images into watercolor. Superpixel has often been defined as a popular method in computer vision such as image segmentation and Saliency Detection. However, considering that its algorithm can output regular and compact superpixels, this kind of method could be ingeniously applied in changing images into aquarelle.

We propose a practical superpixel algorithm for aquarelle-style transform which provided four qualities:

1. generate compact superpixels;
2. generate regular superpixels;
3. fast computation;
4. low computational overhead;

The simple linear iterative clustering (SLIC)[1], which can clusters pixels in the combined five-dimensional color and image plane space to achieve compact and uniform superpixels, exhibits all of these desired qualities.

Based on this algorithm, we propose an efficient method. Given a user-defined number as parameter, one can quickly transforming images into aquarelle. Experiments show that our approach have excellent performance on images with nature objects, for instance, plants, fruits, animals, etc.

The SLIC Formulation

Not all of the segmentation algorithms we proved in section 2 were focus on the qualities our novel purpose need, but we still seek the SLIC out to generate aquarelle nonetheless.

1.1 Cluster center

Supposing that we need M Superpixels with approximately equal size in an input image with N pixels, obviously the size of each superpixel is S = N/M. Therefore, at the onset of the algorithm, we initialize M cluster centers in every grid interval $\sqrt{S}$:

$$C_m = \begin{bmatrix} L_m \\ A_m \\ B_m \\ x_m \\ y_m \end{bmatrix}, \text{m} = [1,\text{M}] \quad (1)$$

Where L,A,B is the three channel of the CIELAB color space and x, y is the pixel coordinates of each cluster center. We choose 4×S as the searching area for each center's neighbor pixels. Since each superpixel's size is approximately S, a 4×S searching area is enough safe to search all the pixels which are related to the corresponding cluster center.

3.2 Distance

Since the 5-D space proposed by SLIC includes color space and spatial space, simply using Euclidean distance as the method to measure the distance between normal pixels and cluster centers is obviously inadvisable. We consequently define D as the distance:

$$\text{D} = D_{lab} + \frac{m}{\sqrt{S}} D_{xy} \quad (2)$$

The value of m can control the compactness of superpixels. It can be ranged in [1, 20], and we choose m = 10 D_{lab} and D_{xy} is the distance in the color space and spatial space. Here we choose Euclidean distance in this two spaces:

$$D_{lab} = \sqrt{(L_m - L_i)^2 + (A_m - A_i)^2 + (B_m - B_i)^2} \quad (3)$$

$$D_{xy} = \sqrt{(x_m - x_i)^2 + (y_m - y_i)^2} \quad (4)$$

3.3 Gradient

To avoid placing noisy pixels and choosing the cluster centers at an edge, moving them to seed location corresponding to the lowest gradient position is necessary. We define V(x, y) as the LAB vector of the pixel at (x, y), then the gradients can be computed as:

$$\text{Grad} = ||V(x+1, y) - V(x-1, y)||^2 + ||V(x, y+1) - V(x, y-1)||^2 \quad (5)$$

Where ||.|| is the L_2 norm.

3.4 Iterative computation

After defining the cluster centers and moving them to the lowest gradient position. A iterative process is executed as show in table 1:

Table 1 The iterative process in SLIC generation

do
for every cluster center C_m
do
search the pixels which has the best matching D in the 4S search area of the corresponding cluster center.
end for
Replace the cluster center and obtain the residual error E.
While E > threshold
Connectivity.
Where E is the L_1 distance between new center and previous center.

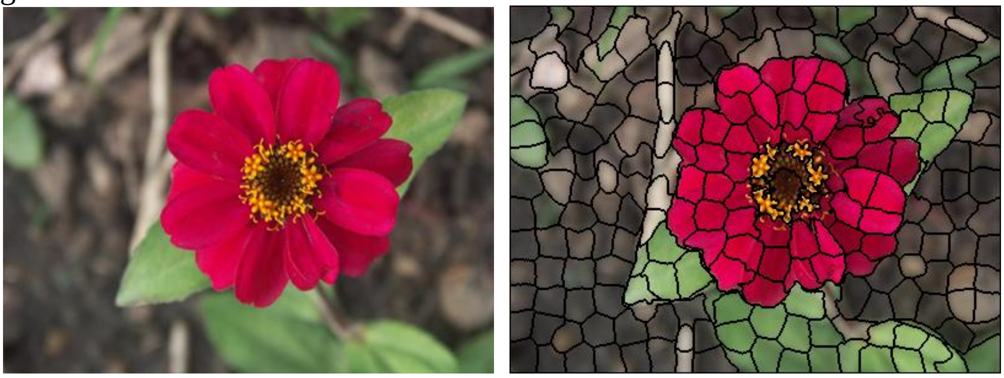
Achieve the watercolor using SLIC

Although white-black watercolor belongs to aquarelle, its painting technology and effect is quite different from common watercolor. Therefore, we merely consider the general watercolor. Based on the algorithm we provide in above section, a grayscale image like the image in Fig.1. could be produced. In this section, we show how to put color on this image to achieve the watercolor.

After clustering, we use number m to sign the pixels in the m^{th} region, then a tagged map is generated. Here we choose RGB color space to obtain the color information. Through the label map, we can directly determine the i^{th} pixel's value:

$$\begin{cases} R_i = \overline{RI}_m \\ G_i = \overline{GI}_m \\ B_i = \overline{BI}_m \end{cases} \quad i \in [1,N], m \in [1,M] \tag{6}$$

Where $\overline{RI}_m$, $\overline{GI}_m$, $\overline{BI}_m$ is the mean intensity of the pixels in the m^{th} region for each channel. According to this method, we compute every pixel's value to obtain the image with watercolor effect in Fig.1.



(a) (b)

Fig.1. a. original image b. SLIC segmentation

Test results

On the basis of painting objects, artistic work is classified into static-object paintings and human paintings. The former select some natural scenery such as mountains, rivers, flowers, animals as the objects. In comparison, human paintings pay more attention on human's behavior and expression. In our experiment, we evaluated the result of our approach on both paintings. The test image dataset is MSRA dataset[9]. We first test the former and results are extremely excellent:

We also notice that the effect is better when the difference between object and background is huge. After all, superpixel algorithm is designed for segmentation and saliency detection.

Meanwhile, considering the time complexity, we record the time of processing all the image in the selected database on each value of m and calculate the mean time using $\overline{T}$ in table 2.

Table 2.Time using of processing one image into watercolor on variable m.

m	125	250	375	500
$\overline{T}$/s	0.2069	0.3020	0.4032	0.4953

It is obvious that the result is unsatisfied when m is too small, in contrast, the algorithm will use so much time when the value of m is huge. To keep a balance between effect and time, we propose that the m can range from 250 to 375.

Conclusion

We provide a novel idea of automatically producing watercolor. Thanks to the support of algorithms in the area of computer vision, we process images into aquarelle. Comparing the existing superpixel algorithm, we select SLIC as our foundation. Based on this method, we compute each pixel’s value to realize the transform. Experiment results demonstrate that our approach has a good performance on processing images with natural objects.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. Technical Report, EPFL, 2010.

[2] R. Achanta, F. Estrada, P. Wils, and S. S¨usstrunk. Salient region detection and segmentation. In ICVS, 2008.

[3] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequencytuned salient region detection. In CVPR, 2009.

[4] Zou YaoBin, Gao Hui. Watercolor effect image painting [J]. Computer Programming Skills & Maintenance, 2008,11:61-64.

[5] Wang M Y, Wang Bin, Yong J H. Real-time watercolor illustrations and animation on GPU [J]. Journal of Graphics, 2012,3:73-80.

[6] Felzenszwalb, P., Huttenlocher, D.: Ecient graph-based image segmentation. IJCV (2004) 167-181.

[7] Shi, J., Malik., J.: Normalized cuts and image segmentation. PAMI (2000) 888-905

[8] Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. ECCV(2008)

[9] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. IEEE Conference on Computer Vision and Pattern Recognition, 2007.