A Color Image Segmentation Algorithm and Its Application

Zhenjiang Miao, Baozong Yuan

Institute of Information Science,
Beijing Jiaotong University, Beijing 100044, P.R.China
Email : zjmiao@center.njtu.edu.cn

Abstract: This paper proposes an adaptive and robust color image segmentation scheme. It is efficient and flexible for color image segmentation. It can work with a given precision or in a way which gives final segmented color region number beforehand. Its application to rose petal analysis is presented.

Keywords: Image Procession, Color Image Segmentation, Pattern Recognition, Intelligent Systems

1. Introduction

Image segmentation is an important issue for image analysis. Researchers have developed many different color segmentation techniques [1-5] so far. The basic idea for segmentation is to use a clustering procedure to find clusters of pixel values in color space and then assigns each image pixel to one of the clusters. We can apply this method to the whole color image to carry out segmentation, which is a frequently used approach [6]. This approach is simple but efficient. Another approach to color segmentation is region splitting. Region splitting involves recursively breaking the input image into smaller and smaller pieces until each piece is uniform in some property.

Histograms can also be used to guide region splitting. For each region this algorithm computes histograms from the input RGB values. The best peak in one of the histograms is used to threshold the region to separate out the pixels corresponding to the peak. Since histogram analysis is a global process, this method tends to miss small image regions which will not produce strong peaks in any histogram. The above approaches are histogram based method. Another type of segmentation techniques is the neighborhood based approach which employs Markov Random Field [7]. One crucial problem for this approach is that it requires *a priori* information of the image. Physically-based segmentation techniques use the underlying physical models of the color image formation process in developing color difference metrics [8]. One common problem in most segmentation is that the boundaries of the regions do not necessarily follow material boundaries. Instead, they follow color or intensity variations, such as highlight and shadow boundaries. To alleviate this problem, a model called the Dichromatic Reflection Model (DRM) [8] is proposed. Healey [9]] uses this model in color image segmentation. But there are many rigid assumptions of the DRM, e.g., the illumination conditions, the type of materials. For most realistic scenes, these assumptions do not hold. Therefore, the DRM model can only be used in a controlled environment. This paper proposes an adaptive and robust color image segmentation scheme and applies it to petal analysis.

2. Color Distribution Analysis of Rose Petal

Flower consists mainly of petals. The colors of petals specify the flower complexion. On the surface of a petal, there are a variety of colors distributing differently. What is interested us for our rose variety recognition project is the color itself and its distribution on the surface of petal.

Normally the color distribution on petal is segmented as many color zones based on our requirements. Since our purpose for image segmentation is to study color distributions of flower petals and what we are concerned about is not the color region shapes on the petal images, the measurement space guided spatial clustering is more suitable for our problem.

There are two different cases for our segmentation. The first case is comparatively simple as we have known what color and how many kinds of color are included in the petal/scene before processing. The remaining problem is to use classification criteria to count the amount of pixels for each color. For the second case, we must analyze the image and decide how many colors are included in the image and get their color space characteristics for each color. In order to fulfill this process, we should analyze the image histogram. For color image, there are three images such as RGB for each. How to comprehensively analyze these three images is a crucial issue.

3. Algorithm Description

(1) General Description

The basic idea for the color image segmentation method is clustering, i.e., grouping the similar pixels and separating the dissimilar ones. Suppose we are given a set of patterns (pixels) without any a priori information such as the number of classes presented in the set. The clustering problem in such a case is to identify the number of classes according to a certain criterion, and assign the membership of the patterns in these classes. So, for a given color image there are two basic tasks for segmentation, one is to decide how many colors in the image or how many colors we expect to cluster the image, the other is to assign each pixel to its most similar color according to a certain similarity measure, i.e., classification.

For the first problem, we should give not only the color number but also the corresponding position of each color in the color space. Then we should choose decision functions required to identify possible clusters. To define a cluster, we need to establish a basis for assigning patterns to the domain of a particular cluster. The most common similarity rule is the Euclidean distance between two patterns x_i and x_j defined as:

$$\left\| x_i - x_j \right\| = \sqrt{(x_i - x_j)^t (x_i - x_j)}$$

But this similarity measurement does not always work well. For example, for the color similarity measurement, this measure often makes mistakes. The following measurement is more useful**:**

$$\cos\varphi = \frac{x_i^t x_j}{\left\| x_i \right\| \left\| x_j \right\|}$$

We can choose different measurement rule based on the practical situations.

In general, the operation of the method we propose may be generally described by the following steps:

Step 1 cluster center detection, that is, to detect all the possible color/grey classes of R, G, B images respectively
Step 2 cluster merging, that is, to merge the similar classes as one and repeat step 1 until finding the proper class centers for each R, G, B image respectively
Step 3 comprehensive cluster merging

(2) Histogram Analysis

To deal with the segmentation problem, we can first find highest peak and lowest valley in histogram. Between the valley and its nearest peak, we think all the pixels have the same grey level as that of the peak if the distance between the peak and valley is less than a threshold value T_1. If the distance between peak and valley is greater than the threshold value T_1, for example, 2.4 times T_1, we can separate this peak-valley area into three parts and we think each part have the same grey level.

(3) Cluster Detection

For each of RGB images, the following operations are carried out:

- to pre-process the image such as to filter and remove noise[10]
- to calculate its histogram
- to find the peak and valley points of the histogram (if the value of the histogram is greater than the non-zero ones of its both nearest sides, we think it is a peak; if less than its both nearest side non-zero ones, it is valley; otherwise, neither peak nor valley, it is the slope.)

(4) Cluster Merging

- fusion (from valley point to the nearest peak point if the valley is not too far from the peak)

In this process, some criteria are applies:
- if the peak-valley distance less than T_1 (such as 20 for the 256 grey levels), carry out fusion, i.e., we think that all the grey levels of the pixels between the peak and valley have the same grey level as that of the peak;
- if the distances from the valley to its two-side peaks are the same, fuse the valley to the higher peak ;
- if there is the peak whose both-side peak-peak distances greater than T_2 and its pixel number greater than T_3, take it out as a class;
- if there is the peak whose pixel number is very big, let us say, greater than T_4 (for example, row*col/3), take it out as a class;
- if there is the peak and the distance between its nearest two-side valleys is very big, let us say, greater than T_5 (for example, 40), take it out as a class.
- If the distance between peak and valley is too big and there are non-zero histogram values between the peak and valley, i.e., greater than the threshold value T_1, for example, 1.6 times T_1, we can separate this peak-valley area into two parts according to the non-zero slope histogram values and carry out fusion.

After many time iterations, if the class number is not stable, we can choose one of the following two methods to terminate the operation: one is the iteration times K, the other is the obtained cluster number (non-zero histogram number) C. We give the limits for both values in advance. If K or C is greater than its given limit, we stop fusion process. We can also apply the other criteria to terminate the operation, for example, if all the distances between the merged clusters are greater than a limit (such as 20 for 256 grey levels) which is given in advance, we can finish the operation.

(5) Comprehensive Cluster Merging

In the phases of cluster detection and cluster merging, we deal respectively with R, G, and B image (or other represented images) as it does for grey image segmentation and get the final segmented images. Of course, the segmented sets for each R, G, and B image are not the same but intersected. So it is necessary to comprehensively use R,G and B image to carry out clustering. One example is shown in Fig.1 which can demonstrate the indispensability of the comprehensive clustering. The original color image includes four color zones evidently. Its R, G and B images are with only three color zones for each. It is impossible to get four zones by segmenting each its R,G or B image. But if we comprehensively consider its R,G and B images, it is easy to get the four color zones. However, if we simply put the segmented R, G and B image together as a color image, in most cases we obtain tremendous color regions whose number is more than the region number of each segmented R, G, or B image even is their region number multiplication. Usually, the color region number is quite enormous but the color difference is not very evident among a lot of color regions. So it is necessary for us to carry our further comprehensive cluster merging in the three dimensional color space.

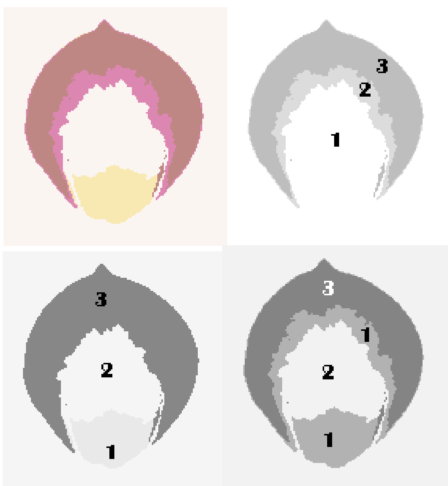

Fig.1 Original 4 Color Zones Image and Its R,G,B Images with only 3 Color Zones

Here we take RGB image as an example to explain the process:

- to find the image total color number

We carried out Cluster Detection and Cluster Merging for each R,G and B image and get their cluster numbers respectively. We also know the pixels that each cluster corresponds to. The total color number of the image is the combination number of the three R, G, B images. For example, the cluster numbers are 3 for each RGB image and the image total color number is 4.

- for the color classes whose pixel numbers are too small, for example, less than T_6, to find their nearest/most similar color class whose pixel number is greater than T_6 and put them to the bigger color class;

For the color distance/similarity, there are several criteria. The most useful is the Euclidean distance and cosine similarity rule. After some experiments we find that the cosine similarity rule is better for RGB, $L^*a^*b^*$ and $L^*H^\circ C^*$ image segmentation for our flower petal color segmentation:

- to calculate the distances between each color classes, find the smallest distance, make these two color class to one
- there are three criteria to terminate the classification operation: one is to give the expected image color number, the second is to give the smallest color distance, the third is to give the limited iteration number.

For other images such as $L^*a^*b^*$, $U^*V^*W^*$, $S\theta W^*$ and so on, the process is the same as that of RGB which we have explained above in detail.

The reasons why we do not use the original color image to carry out classification directly in the color space are: (1) it requires too much RAM to operate because nearly all the color pixels are not the same color in the real application; (2) the operation is too slow. But theoretically we can do this directly.

In general, taking RGB image as an example, the segmentation scheme can be summarized as follows:

(I) Image Pre-processing Phase
We input a color image into computer and carry out pre-processing such as filtering to remove image noise for its R, G, and B image respectively.

(II) Cluster Center Detection Phase
step 1 – to calculate histogram
step 2 – to find peak and valley points in the histogram
step 3 – to merge the pixels between valley and its nearest peak to the peak point
step 4 – to check. If it satisfies the criteria, stop; otherwise, repeat step 1,2,3

(III) Clustering Phase
step 1 – to find the possible color cluster centers and the possibly segmented total color number
step 2 – to find the very small color clusters and merge them absolutely to their nearest bigger color clusters
step 3 – to calculate all the distances/similarities between each clusters
step 4 – to find the smallest distance and merge the two clusters
step 5 – to set the new cluster center
step 6 – to check. If it satisfies the criteria, stop; otherwise, repeat step 3,4,5

4. Some Segment Results and Comparison Comments

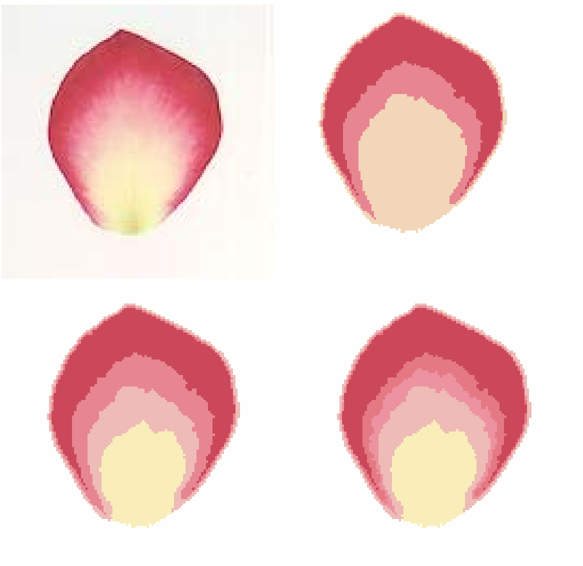

Fig.2 Original and Segmented Petal with Different Color Scale Zones

Fig.2 is one segmentation result. We can see the segmental results are quite good. We also used some other methods to segment petal color region, but the results are not good. They are difficult to segment petal 1 as we do now. They segment the upper edge folding shadow area as a color zone. In fact, this is not another color region but just the same as its neighbor color. For some specified petals, maybe we can adjust some threshold to deal with this kind of problem, but it is difficult to get a general adaptive segment algorithm. Our present algorithm is better than others at least for this point. In addition, most of other methods give the segmentation precision first, or to say threshold, and then carry out segmentation. This way does not always work well. It lacks of adaptability. In some cases, you can see that there are, let us say, five color regions on a petal. But the algorithm gives four color regions, if the threshold is too low, or six, if the threshold too high. In some cases, the given precision works well for one kind of petal but not well for another. Our method can work in both ways, i.e., the given precision way as the other methods do or the way of giving segment color region number as you want or you see. So we can say our method is adaptive and robust

Acknowledgement: This work is supported by University Fund 2004SZ002, 2004RC020, 2003SZ002, NSF60441002, 973 Program 2004CB318110

References

[1] Heng-Da Cheng, et al, , A Hierarchical Approach to Color Image Segmentation Using Homogeneity, IEEE Trans. on Image Processing Vol. 9, No.12, 2000
[2] S. Wesolkowski, et al, A Markov Random Fields Model For Hybrid Edge- and Region-Based Color Image Segmentation, IEEE Canadian Conference on Electrical and Computer Engineering, 2002
[3] Hsin-Chia Chen, et al, Contrast-Based Color Image Segmentation, IEEE Signal Processing Letters, Vol.11, No.7, 2004
[4] Qixiang Ye, Wen Gao, et al, Color Image Segmentation Using Density-Based Clustering, Proc of 2003 International Conference on Multimedia and Expo, 2003.
[5] Liu Rujie, Yuan Baozong, Unsupervised Color Image Segmentation, Proc. Of ICSP'02, 2002
[6] A. Sarabi, et al, Segmentation of chromatic Images, Pattern Recognition, 13 (6), 1981, pp.417-427
[7] J. Lin, et al, Multiresolution Color Image Segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence 16, 1994, pp.689-700
[8] G.J.Klinker, S.A.Shafer, and T.Kanade, A Physical Approach to Color Image Understanding, Int. J. Comput. Vision, Vol.4, 1990, pp.7-38
[9] G.Healey, Segmentation Images Using Normalized Color, IEEE Trans. Syst., Man, Cybern., Vol.22, 1992, pp.64-73
[10] R.C. Gonzalez, Digital Image Processing, (2nd edition), Pearson Education, Inc (Prentice Hall), 2002