A Hierarchical Approach to Color Image Segmentation Using Homogeneity

H. D. Cheng and Ying Sun
Dept. of Computer Science
Utah State University
Logan, UT 84322-4205

Abstract

In this paper, a novel hierarchical approach to color image segmentation is studied. We extend the general idea of a histogram to homogeneity domain. In the first phase of the segmentation, uniform regions are identified via multi-level thresholding on homogeneity histogram. While we process homogeneity histogram, both local and global information is taken into consideration. This is particularly helpful in taking care of small objects and local variation of color images. An efficient peak-finding algorithm is employed to identify the most significant peaks of the histogram. In the second phase, we perform histogram analysis on the color feature hue for each uniform region obtained in the first phase. We successfully remove about 99.7% singularity off the original images by redefining the hue values for the unstable points according to the local information. After the hierarchical segmentation is performed, a region merging process is employed to avoid over-segmentation. CIE($L^*a^*b^*$) color space is used to measure the color difference. Experimental results have demonstrated the effectiveness and superiority of the proposed method after an extensive set of color images was tested.

I. Introduction

Image segmentation serves as the key of image analysis and pattern recognition. It's a process of dividing an image into different regions such that each region is homogeneous, but the union of any two regions is not [1, 2].

Color of an image can carry much more information than gray level [1]. In many pattern recognition and computer vision applications, the additional information provided by color can help the image analysis process and yield better results than approaches using only gray scale information [3]. More research has focused on color image segmentation due to its demanding need. At present, color image segmentation methods are mainly extended from monochrome segmentation approaches by being implemented in different color spaces [1]. Gray level segmentation methods are directly applied to each component of a color space, then the results are combined to obtain the final segmentation result [4]. Generally, color image segmentation approaches can be divided into the following categories: statistical approaches, edge detection approaches, region splitting and merging approaches, methods based on physical reflectance models, methods based on human color perception, and the approaches using fuzzy set theory [1, 2].

Histogram thresholding is one of the widely used techniques for monochrome image segmentation [5]. As for color images, the situation is different due to the multi-features [6]. Since the color information is represented by tristimulus R, G and B or some linear/non-linear transformation of RGB, representing the histogram of a color image in a 3-dimensional array and selecting threshold in the histogram is not a trivial job [7]. One way to solve this problem is to develop efficient methods for storing and processing the information of the image in the 3D color space. [8] used a binary tree to store the 3D histogram of a color image, where each node of the tree includes RGB values as the key and the number of points whose RGB values are within a range centered by the key value. [9] also utilized the same data structure and similar method to detect clusters in the 3D normalized color space (X, Y, I). Another way is to project the 3D space onto a lower dimensional space, such as 2D or even

1D. [10] used projections of 3D normalized color space (X, Y, I) space onto the 2D planes (X-Y, X-I, and Y-I) to interactively detect insect infestations in citrus orchards from aerial color infrared photographs. [11] provided segmentation approaches using 2D projection of color space. [12] suggested a multidimensional histogram thresholding scheme using threshold values obtained from three different color spaces (RGB, YIQ, and HSI). This method used a mask for region splitting and the initial mask included all pixels in the image. For any mask, histograms of the nine redundant features (R, G, B, Y, I, Q, H, S, and I) of the masked image are computed, all peaks in these histograms are located, the histogram with the best peak is selected and a threshold is determined to split the masked image into two sub-regions for which two new masks are generated for further splitting. This operation is repeated until no mask left unprocessed, which means none of the nine histograms of existing regions can be further thresholded and each region is homogeneous.

This paper proposes a novel approach to color image segmentation. We extend the general concept of the histogram and define ahomogeneity histogram. The histogram analysis is applied to both the homogeneity domain and the color feature domain. While calculating the homogeneity feature, both local information and global information are taken into account. We employ a hierarchical histogram analysis method based on homogeneity and color features.

II. Homogeneity Histogram Analysis

2.1. Homogeneity and the histogram based on homogeneity

Homogeneity is largely related to the local information extracted from an image and reflects how uniform a region is [13]. It plays an important role in image segmentation since the result of image segmentation would be several homogeneous regions. We define homogeneity as a composition of two components: standard deviation and discontinuity of the intensities I, $I = (R + G + B)/3$. For color images with RGB representation, the color of a pixel is a mixture of the three primitive colors red, green and blue. We use the average luminance of the three primitive colors as the intensity of the target pixel. This is based on the essential rules of colorimetry: the luminance of any color is equal to the sum of the luminance of each primitive color, and a linear transformation of the primitive colors does not change the basis of the representation for a color [14]. Standard deviation describes the contrast within a local region [13]. Discontinuity is a measure of abrupt changes in gray levels and could be obtained by applying edge detectors to the corresponding region.

Suppose g_{ij} is the intensity of a pixel P_{ij} at the location (i, j) in an M×N image, $w^{(1)}{}_{ij}$ is a size d×d window centered at (i, j) for the computation of variation, $w^{(2)}{}_{ij}$ is a size t×t window centered at (i, j) for the computation of discontinuity, d and t are odd integers greater than 1. We regard $w^{(1)}{}_{ij}$ and $w^{(2)}{}_{ij}$ as the local regions while we calculate the homogeneity features for pixel P_{ij}.

The standard deviation of pixel P_{ij} is calculated:

$$v_{ij} = \sqrt{\frac{1}{d^2} \sum_{p=i-\frac{d-1}{2}}^{i+\frac{d-1}{2}} \sum_{q=j-\frac{d-1}{2}}^{j+\frac{d-1}{2}} (g_{pq} - \mu_{ij})^2} \tag{1}$$

where $0 \leq i, p \leq M-1$, $0 \leq j, q \leq N-1$.

μ_{ij} is the mean of the gray levels within window w_{ij}, and calculated as:

$$\mu_{ij} = \frac{1}{d^2} \sum_{p=i-\frac{d-1}{2}}^{i+\frac{d-1}{2}} \sum_{q=j-\frac{d-1}{2}}^{j+\frac{d-1}{2}} g_{pq} \tag{2}$$

The discontinuity for pixel P_{ij} is described by edge value. There are many different edge operators: Sobel, Laplace, Canny, etc [15]. Since we do not need to find the exact locations of the edges, and due to its simplicity, we employ Sobel operator to calculate the discontinuity and use the magnitude of the gradient at location (i, j) as the measurement [13]:

$$e_{ij} = \sqrt{G_x^2 + G_y^2} \tag{3}$$

where G_x and G_y are the components of the gradient in the x and y directions, respectively.

The standard deviation and discontinuity values are normalized in order to achieve computational consistence:

$$V(g_{ij}, w^{(1)}{}_{ij}) = \frac{v_{ij}}{v_{max}} \tag{4}$$

$$E(g_{ij}, w^{(2)}{}_{ij}) = \frac{e_{ij}}{e_{max}} \tag{5}$$

where $v_{max} = max\{v_{ij}\}$, $e_{max} = max\{e_{ij}\}$ $(0 \le i \le M-1,\ 0 \le j \le N-1)$.

The homogeneity is represented as:

$$H(g_{ij}, w^{(1)}{}_{ij}, w^{(2)}{}_{ij}) = 1 - E(g_{ij}, w^{(2)}{}_{ij}) \times V(g_{ij}, w^{(1)}{}_{ij}) \tag{6}$$

where $0 \le i \le M-1$, $0 \le j \le N-1$.

The value of the homogeneity at each location of an image has a range from 0 to 1. The more uniform the local region surrounding a pixel is, the larger the homogeneity value the pixel has. The size of the windows has influence on the calculation of the homogeneity value. The window should be big enough to allow enough local information to be involved in the computation of the homogeneity for the pixel. Furthermore, using a larger window in the computation of the homogeneity increases smoothing effect, and makes the derivative operations less sensitive to noise [13]. However, smoothing the local area might hide some abrupt changes of the local region. Also, a large window causes significant processing time. Weighing the pros and cons, we choose a 5×5 window for computing the standard deviation of the pixel P_{ij}, and a 3×3 window for computing the edge.

A classical histogram is a statistical graph counting the frequency of occurrence of each gray level in an image or in part of an image [15]. We extend this idea and define a histogram in homogeneity domain. First, homogeneity value for each pixel is calculated. Second, for each intensity value from 0 to 255, add up the homogeneity values of all the uniform pixels with this intensity. Here two issues are concerned in computing the homogeneity for each intensity value. One is that we need to find pixels that uniform for a given intensity, and only uniform pixels are counted. The other is the number of uniform pixels since we should identify substantial uniform regions, not small ones. Experimentally, we set the homogeneity threshold to be 0.95, which means the pixels that have homogeneity equal to or greater than 0.95 take part in the computation of the homogeneous characteristics. Last, we have the homogeneity value for each intensity value normalized and plotted against that intensity. The homogeneity histogram gives us a global description of the distribution of the uniform regions across intensity levels. Each peak in this histogram represents a uniform region.

2.2 A peak-finding histogram analysis algorithm

A histogram of the analyzing features of an image could produce a global description of the image's information and is utilized as an important basis of statistical approaches in image processing [15]. The basis of histogram analysis approach is that the regions of interest tend to form modes (a dominating peak that could represent a region) in the corresponding histogram. For example, a light object in a dark background might produce two modes in the image's gray level histogram, one is at the bright intensity side, and the other is at the dark intensity side [13]. Then, a typical image segmentation approach based on histogram analysis generally carries out three steps: First, recognize the modes of the histogram. Second, find the valleys between different modes. Third, apply thresholds to the image. Locating the modes of an image is the most important and difficult task among the three steps.

The key of partitioning modes in a histogram is a process of finding and removing peaks in a histogram curve. Some widely used methods choose significant peaks by examining a peak's sharpness or area [16]. If a peak is not sharp or big enough, then it is ignored. Experimental results showed that this approach does not work well sometimes, especially for the images that have some noise or radical variation. For instance, if a small peak resides on the top of a big peak (as shown in Fig. 1(a)), then removing the small peak would also remove the big peak. In another occasion, if a sizable peak is just a branch of a huge peak (as shown in Fig. 1(b)), it should not be distinguished from the main peak.

We propose a new peak-finding histogram analysis method. This algorithm has been proved to be efficient by testing on more than one hundred images. An example showing the process of this algorithm is in Figs. 1(c)-(h).

Suppose a homogeneity histogram of an image is represented by a function h(i), where i is an integer, $0 \leq i \leq 255$.

Peak Finding Algorithm

(1) Find all peaks:

Find the set of points corresponding to the local maximums of the histogram:

$$P_0 = \{(\mathrm{i}, \mathrm{h(i)})|\mathrm{h(i)} > \mathrm{h(i-1)} \ \& \ \mathrm{h(i)} > \mathrm{h(i+1)}, 1 \leq \mathrm{i} \leq 254\} \tag{7}$$

(2) Find significant peaks: The points in set P_0 form a new curve. On this new curve, repeat the operation of step (1). The result forms set P_1:

$$P_1 = \{(\mathrm{p_i}, \mathrm{h(p_i)})|\mathrm{h(p_i)} > \mathrm{h(p_{i-1})} \ \& \ \mathrm{h(p_i)} > \mathrm{h(p_{i+1})}, \mathrm{p_i} \in \mathrm{P_0}\} \tag{8}$$

All the points in set P_1 are much more significant than the points in set P_0 in determination of the peaks of the histogram.

(3) Thresholding: it includes three steps.

The first step is to remove small peaks. If a peak is too small compared to the biggest peak, then it is removed. Suppose i_{max} is the value of the highest peak satisfying $h_{max} = h(i_{max})$. For any peak j, if $\frac{h(j)}{h_{max}} < 0.05$, then peak j is removed. Since the values have been normalized to the range of 0 to 1, h_{max} is equal to 1. Therefore, the points with h(j)<0.05 will be removed.

The second step is to choose one peak if two peaks are too close. For two peaks h(p_1) and h(p_2), $p_2 > p_1$, if $p_2 - p_1 \leq 15$, then $h = max\{h(p_1), h(p_2)\}$. Thus the peak with the bigger value is chosen.

The third step is to remove a peak if the valley between two peaks is not obvious. We examine the obviousness of the valley by calculating the average value for the horizontal axis value between the two peaks. We consider the valley between the two peaks is not obvious if the average value is too big compared to the peaks. Suppose h_{avg} is the average value among the points between peaks p_1 and p_2:

$$h_{avg} = \frac{\sum_{p_i=p_1}^{p_i=p_2} h(p_i)}{p_2 - p_1 + 1} \tag{9}$$

Then, if $h_{avg}/\frac{(h(p_1)+h(p_2))}{2} > 0.75$, we say the valley is not deep enough to separate the two peaks. We will remove the peak with the smaller value from the candidates. The threshold 0.75 is based on the experiments on more than one hundred images.

This peak-finding algorithm locates the globally significant peaks of the histogram. After the peaks are selected, the minimum values between any adjacent peaks are the valleys. The valleys are the boundaries for the segmentation in homogeneity domain.

2.3 Segmentation in homogeneity domain

The homogeneity histogram represents the homogeneity distribution across intensities of the image. Using the proposed peak-finding algorithm, we may find a series of valleys that could separate the most significant modes in the homogeneity histogram.

Suppose the intensity values of the corresponding valleys of the histogram are v_1, v_2, ..., v_{m-1}, then the original image could be divided into uniform regions with the intensity boundaries: 0, v_1, v_2, ..., v_{m-1}, 255. Set $v_0 = 0$ and $v_m = 255$. All the pixels of the ith region have the intensity value between v_{i-1} and v_i ($1 \leq i \leq m$).

An advantage of segmentation in homogeneity domain is that local information and global information are both taken into account in determining the segmentation criteria, whereas in traditional histogram approaches only global information is considered.

III. Hierarchical Segmentation Using Color Feature Hue

3.1 Hue

Hue can be obtained by a non-linear transformation from R, G and B color features [17]:

$$Hue = arctan(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}) \tag{10}$$

Hue reflects the predominant color of an object and has a great capability in subjective color perception [18]. Hue is also the most useful attribute in color segmentation since it is less influenced by the non-uniform illumination such as shade, shadow or reflect lights [16].

The output of the first phase of the proposed approach is several uniform regions based on homogeneity. The second phase of the segmentation is to apply histogram analysis on the color feature hue. That is, in each uniform region obtained from the first phase, the pixels are divided into several groups with each group having similar colors. In this sense, the segmentation approach is performed hierarchically.

3.2 Singularity

Hue has been proven to be an efficient color feature in color image segmentation. But a disadvantage of hue is its singularity and numerical instability at low saturation [1]. A commonly used method is to treat the unstable pixels separately, since RGB color space and its linear transformations do not have singularity problems. For instance, chromatic and achromatic regions were defined for this purpose [16]. [19] utilized a first-order membrane type stabilizer based on Markov random fields to smooth the unstable hue regions.

However, separating an image into chromatic regions and achromatic regions might lose local information for the determination of segmentation criteria. The smoothing algorithms might have high computational complexity and blur the detail of the image. In our approach, we re-define the hue value for a singularity pixel by averaging the hue values of its eight neighbors. The hypothesis entertained here is that the color of a pixel is similar to its neighbors. If the pixel happens to be near an edge, then at least half of its neighbors are in the same region with it. Therefore, taking the average value would be reasonable. If some of its neighbors are also undefined pixels, then we just take the average value of the rest neighbors having valid hues. If all its eight neighbors are undefined pixels, the target pixel is left undefined until the color region merging is conducted.

Among the 122 images in our experiments, the average singularity rate is 0.35%. We calculate an image's singularity rate as the ratio of the number of singularity points to the number of all the pixels of the image. After the above mentioned re-definition of the hue values for the singularity points, the singularity rates for 110 images become to zero. For the other 12 images, their average singularity rate reduces to 0.01%. Altogether, we could remove about 99.7% singularity off the original images. Table 1 lists the changing of the singularity rates for some of the images employed in the experiments. Thus, this method is proved to be efficient to remove the singularity in HSI representation of a color image. Re-defining the singularity points will make those pixels take part in the segmentation process and retain the local information that is useful in calculating the homogeneity of the image. After the re-defining process, only few singularity points are left. A point with the non-removable singularity will be merged into a region that is most similar to the pixel by the region merging process described below.

3.3 Hierarchical segmentation on Hue

In our proposed method, the second phase is to divide the pixels in a uniform region produced in the first phase into one or more sub-regions with each sub-region has the similar visualized color. For the pixels in the same uniform region, we produce another histogram as the number of pixels versus hue values. Hue is computed from a non-linear transformation of RGB color space and normalized to a range of 0 to 255. This is consistent with the histogram representation in intensity with the values from 0 to 255. After the histogram on hue is plotted, we apply the previously described peak-finding algorithm to identify the significant peaks. Finally, we find valleys between peaks and divide the hue range into several segments, and each segment represents a sub-region that is similar in color.

After all the uniform regions are processed and all the sub-regions are obtained, the average color for each sub-region is calculated and assigned to each pixel of the region. RGB representation of the color is used at this point since we need to display the segmented color image. Once the pixels get the new RGB values, they are ready to be displayed directly [18].

3.4 Color Region Merging

3.4.1 CIE color difference description

After the first and second phases of the proposed hierarchical segmentation approach, several sub-regions with similar color and similar homogeneity have been generated. However, over-segmentation may happen when the pixels are in different homogeneity regions but posses similar colors. A color region merging phase is very important to combine those pixels together and produce a more concise set of regions.

CIE (Commission International de l'Eclairage) color system defines three primary colors, denoted as X, Y, and Z. XYZ coordinates come from a linear transformation of RGB space, indicated by Eq.(11):

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{11}$$

CIE($L^*a^*b^*$) appears to possess more uniform perceptual properties than another CIE space, CIE($L^*u^*v^*$) [20]. It is obtained through a nonlinear transformation on XYZ:

$$\begin{aligned} L^* &= 116(\sqrt[3]{\frac{Y}{Y_0}}) - 16 \\ a^* &= 500[\sqrt[3]{\frac{X}{X_0}} - \sqrt[3]{\frac{Y}{Y_0}}] \\ b^* &= 200[\sqrt[3]{\frac{Y}{Y_0}} - \sqrt[3]{\frac{Z}{Z_0}}] \end{aligned} \tag{12}$$

where (X_0, Y_0, Z_0) are XYZ values for the standard white [1, 3].

CIE spaces have metric color difference sensitivity to a good approximation and are very convenient to measure small color difference, while the RGB space does not [21].

If the CIE($L^*a^*b^*$) representations for two color points are (L_1, a_1, b_1) and (L_2, a_2, b_2), respectively, the color difference between them is:

$$\Delta E = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \tag{13}$$

The ability to express color difference of human perception by Euclidean distance is a nice matching from the sensitivity of human eyes for color to computer image processing.

3.4.2 Merging criteria

Suppose k regions were generated from the previous phases of the segmentation method, there are k colors each represents one region. The color difference is computed for any two out of these k colors according to Eq.(13). Totally, there are $\frac{k(k-1)}{2}$ color differences produced. The standard deviation of the list of color differences are calculated as:

$$\sigma_v = \sqrt{\frac{1}{n}\sum_{i=1}^{i=n}(D_i - \mu_d)^2} \tag{14}$$

where n is the total number of color differences, and $n = \frac{k(k-1)}{2}$, μ_d is the mean of the differences and D_i is the *ith* color difference in the list.

We set the threshold for the color merging as:

$$T_d = \mu_d - \sigma_v \tag{15}$$

In the merging algorithm, the two regions having the smallest color difference are first located. If this minimum difference is smaller than T_d, the two regions will be merged. The color of the new region is calculated again and the mean value of the colors is assigned to the pixels within this region. This operation is repeatedly performed until no color difference is smaller than the threshold T_d.

The proposed approach can be described by a flowchart as shown in Fig. 2.

IV. Experiments and Discussions

A large variety of color images is employed in our experiments. Some experimental results are shown in Figs. 3-13. The images originally are stored in RGB format. Each of the primitive colors (red, green and blue) takes 8 bits and has the intensity range from 0 to 255.

Figs. 3-7 demonstrate the results of the proposed approach. It is very obvious that the resulting images with much smaller numbers of colors can preserve the main features of the objects. In Fig. 3, the image "flowers" consists of mainly three kinds of objects: red flowers, yellow flowers and white background. The RGB description of the images is $256 \times 227 \times 256$, which implies that the red component has 256 gray levels, the green component has 227 gray levels, and the blue component has 256 gray levels. After applying the proposed segmentation approach, the resulting image is divided into only three regions, with their colors to be red, yellow and white, respectively. In Fig. 4, the image "beans" is represented by five colors: red, yellow, green, black jelly beans and light cyan background, respectively. Figs. (d) and (e) are very similar, however, (d) has 8 colors and (e) has only 5 colors. In Fig. 5, the resulting image of "kayak" has eight colors. Comparing Figs. (d) and (e), we can find that the 'hand' and the upper-left corner of the image are much better segmented in (e) than those in (d). It demonstrates the necessity of the merging process. In Figs. 6(a)-(b), only five colors are needed to represent the color image "lake". In Figs. 6(c)-(d), the resulting image of "sail" has only seven colors. In Figs. 6(e)-(f), three colors are enough to express the shape, material, and color of the splash and its background. In Figs. 7(a)-(d), the resulting images "river" and "fashion" have only six colors and four colors, respectively. In brief, the experimental results are quite consistent with the visualized color distribution in the objects of the original images. Table 2 lists the experimental results for each segmentation phase of the proposed approach.

Experiments on the segmentation based on traditional histogram thresholding are conducted for comparison. In this approach, only global information is considered in the histogram analysis. The experimental results indicate that the proposed approach, which utilizes both local and global information, is better than the traditional method. As shown in Fig. 8, the sign above the door and the window is better recognized by the proposed approach. The difference of the segmentation result lies in the first phase segmentation. The pixels near the edge of the characters of the sign are grouped into the same region with the light yellow background in the segmentation based on ordinary histogram due to the similarity of the colors. But using homogeneity histogram, the sign pixels do not belong to the same region with the sign's background, since they are not as uniform as the background pixels, and are separated in the thresholding process of the homogeneity histogram. In Fig. 9, the green color at the bottom of the food pile is identified by the proposed approach (Fig. 9(c)), but is not signified by the traditional approach (Fig. 9(b)). In Fig. 10, obviously better result is obtained by the proposed method due to the consideration of the local information in addition to the globally histogram thresholding, as shown in Fig. 10(c). The spattering water, the color and the shape of the canoe, the quality of the skin of the man's face and arm, the sports wearing, and even the texture of the long oar held by the man, are clearly recognized, whereas in Fig. 10(b), even the color of the skin is messed up with the color of the canoe, the splashed water near the man's arm and waist is not identified either.

For comparison, in the second phase of the hierarchical segmentation approach, we test the performance of the proposed approach in RGB color space by applying histogram analysis to Red, Green and Blue color features respectively for each uniform region obtained in the first phase segmentation. The major problem that RGB space suffers is a strong degree of correlation among the three components. The three values change dependently and are highly sensitive to the variation of lightness [22]. This could be observed in our experimental results. In Fig. 11, the reflection of the mountain and the clouds in the water has some violet color that does not exist in the original image. This is an outcome of the inconsistency of the segmentation in Red, Green and Blue three color features independently. In this case, a region is recognized in Red and Blue components but is not identified by Green component. The problem can also be observed in Fig. 13, such as the floor of the "door" image and the face of the "panda" image. In Fig. 12, the yellow flower at the left-bottom corner is under the shadow of an adjacent red flower (Fig. 12(a)), then, the yellow flower is misclassified as red during the segmentation using RGB space (Fig. 12(b)). The same flower is segmented correctly using hue (Fig. 12(c)). This proves that hue is less influenced by shadow and highlight in an image [1]. Using RGB may produce results with much more colors than those using hue, but still can misclassify some regions. However, RGB space can distinguish small changes in color with a much larger computational complexity. As shown in Fig. 11, the trees show more detailed color information using RGB space than using only hue, and therefore bear more likeliness to the original image. Using hue and RGB as the color features in segmentation both have their pros and cons. Table 3 summarizes their advantages and disadvantages.

V. Conclusions

In this paper we propose a novel hierarchical approach to color image segmentation. In the first phase, uniform regions are identified via a thresholding operation on a newly defined homogeneity histogram. While the homogeneity is calculated for an image pixel, both local information and global information are considered. This is pragmatically helpful in recognizing small objects and local standard deviation of color images. The output regions of the color segmentation tend to include more detailed local information important to distinguish different objects in a color image. The quality of the segmentation result is much improved by identifying significant local information more efficiently. While performing histogram thresholding, an effective peak-finding algorithm is employed to identify most significant peaks in a histogram. The color feature hue is proved to be more efficient than RGB color features by this research. RGB requires more computational time. The advantages and disadvantages of different color spaces, hue and RGB, are also given. The proposed approach can be useful for color image segmentation.

References

[1] H. D. Cheng, X. H. Jiang, and Y. Sun, "A Survey on Color Image Segmentation," *The First International Workshop on CVPRIP*, Triangle Park, North Carolina, 1998.

[2] N. Pal and S. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, Vol. 26, No. 9, pp.1277-1294, 1993.

[3] J. Gauch and C. Hsia, "A Comparison of Three Color Images Segmentation Algorithms in Four Color Spaces," *Visual Communications and Image Processing '92*, SPIE Vol. 1818.

[4] C. K. Yang and W. H. Tsai, "Reduction of Color Space Dimensionality by Moment-preserving Thresholding and Its Application for Edge Detection in Color Images," *Pattern Recognition Letters*, Vol. 17, 481-490, 1996

[5] E. Littmann and H. Ritter, "Adaptive Color Segmentation - a Comparison of Neural and Statistical Methods," *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, January, 1997.

[6] T. Uchiyama and M. A. Arbib, "Color Image Segmentation Using Competitive Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 12, 1197-1206, 1994

[7] R.M. Haralick, L.G. Shapiro, "Image Segmentation Techniques", CVGIP 29, 100-132(1985)

[8] B. Schacter, L. Davis, and A. Rosenfeld, "Scene segmentation by cluster detection in color space", Department of Computer Science, University of Maryland, College Park, MD, November, 1975

[9] A. Sarabi, J. K. Aggarwal, "Segmentation of Chromatic Images", *Pattern Recognition* Vol. 13, No. 6, 417-427, 1981

[10] S. A. Underwood, J. K. Aggarwal, "Interactive Computer Analysis of Aerial Color Infrared Photographs", *Computer Graphics and Image Processing*, 6, 1-24, 1977

[11] J. M. Tenenbaum, T.D. Garvey, S. Weyl, and H. C. Wolf, "An Interactive Facility for Scene Analysis Research", Technical Note 87, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, CA, 1974

[12] R. Ohlander, K. Price, and D. R. Reddy, "Picture Segmentation Using A Recursive Region Splitting Method", *Computer Graphics and Image Processing, 8,* 313-333, 1978

[13] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley Publishing Company, 1987.

[14] M. Chapron, "A New Chromatic Edge Detector Used for Color Image Segmentation," *IEEE International Conference on Image Processing A*, 311-314, 1992.

[15] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, John Wiley & Sons, 1997.

[16] D. C. Tseng and C. H. Chang, "Color Segmentation Using Perceptual Attributes," *IEEE International Conference on Image Processing A*, 228-231, 1992.

[17] D. Hoy, "On the Use of Color Imaging in Experimental Applications," Experimental Techniques, July/August, 1997.

[18] W. Kim and R. Park, "Color Image Palette Construction Based on the HSI Color System for Minimizing the Reconstruction Error," *IEEE, International Conference on Image Processing C*, 1041-1044, 1996.

[19] F. Perez and C. Koch, "Toward Color Image Segmentation in Analog VLSI: Algorithm and Hardware", *International Journal of Computer Vision*, 12:1, 17-42, 1994

[20] Y. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," *Computer Graphics and Image Processing*, 13, 222-241, 1980.

[21] G. Robinson, "Color Edge Detection," *Optical Engineering*, Sep/Oct, 1977, Vol. 16, No. 5.

[22] Naoko Ito et al., "The Combination of Edge Detection and Region Extraction in Nonparametric Color Image Segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6): 679-698, 1986.

Table 1: Singularity rates before and after redefining process.

images	size (width×height)	singularity rate (original) (%)	singularity rate (after redefining) (%)
cafe	450×345	0.040	0.002
door	266×348	0.158	0
fashion	200×350	0.347	0.347
beans	179×167	0	0
kayak	372×243	0.608	0
lake	498×335	0.493	0
lemon	269×219	0.002	0
mount	480×326	0.056	0
panda	230×175	1.215	0
flowers	494×363	0.001	0
river	504×333	0.191	0
sail	245×358	0.029	0
splash	213×190	0.007	0

Table 2: Experimental results for each segmentation phase.

image	RGB description (R×G×B)	percentage of uniform pixels (%)	regions (after 1st phase)	regions (after 2nd phase)	regions (after merging)
cafe	256×256×256	87	5	17	7
door	241×230×217	91	4	19	8
fashion	256×256×256	89	4	4	4
beans	151×172×194	91	4	8	5
kayak	256×256×256	88	3	16	8
lake	256×256×256	76	3	28	5
lemon	256×241×256	89	5	40	9
mount	256×255×254	96	3	29	7
panda	256×256×256	93	3	16	5
flowers	256×227×256	92	3	9	3
river	256×256×256	83	4	23	6
sail	190×254×256	96	5	32	7
splash	195×130×243	94	4	11	3

Table 3: A comparison: Hue and RGB as the color features in image segmentation.

Color feature	Advantages	Disadvantages
Hue	(1) Can represent subjective color suitable for human observers. The result does not have correlation. (2) Ability to identify objects under shadow, shade, and highlights [1]. (3) The segmentation is performed on only one dimension, using less processing time. (4) The final result has less segments (or regions) than using RGB.	(1) Cannot remove large blocks of unstable points where saturation is low. (2) Sometimes, cannot distinguish small color changes.
RGB	(1) Perform better in small color changes since the segmentation is operated on three color features.	(1) There might be spurious result due to the correlation of the three color-features. (2) The segmentation is performed on three dimensions, needs triple processing time. (3) Segmentation result might be influenced with shadows and highlights on objects because all three parameters depend on light intensity [11]. (4) More segments (or regions) are produced.

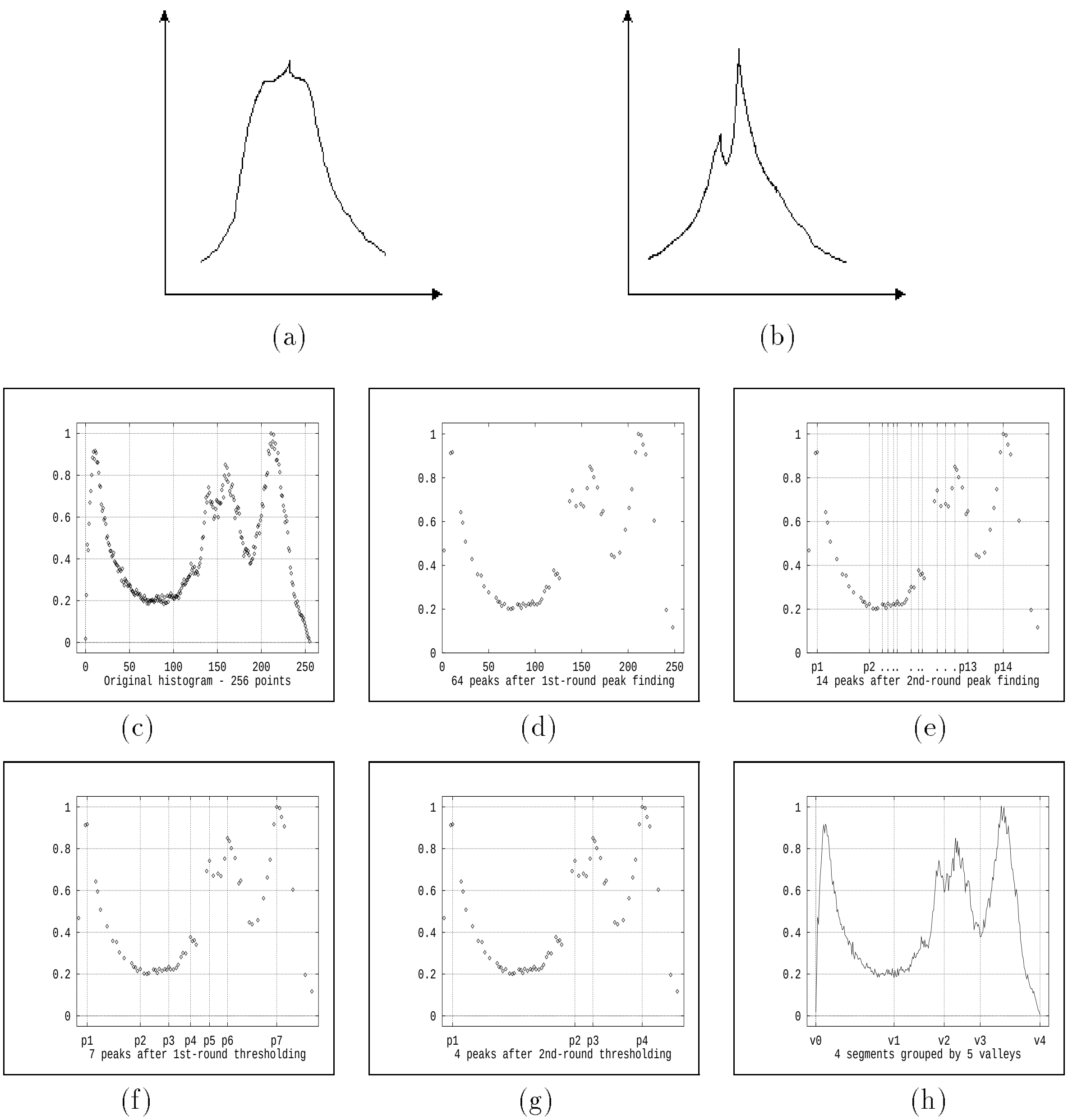


Fig. 1. Peak-finding algorithm for histogram analysis: (a) The small peak at the top of the huge peak should not be removed, (b) The small peak as a branch of the huge peak should be removed. (c) Original histogram, (d)-(g) Results of the four steps, respectively. (h) Final result by the proposed method.

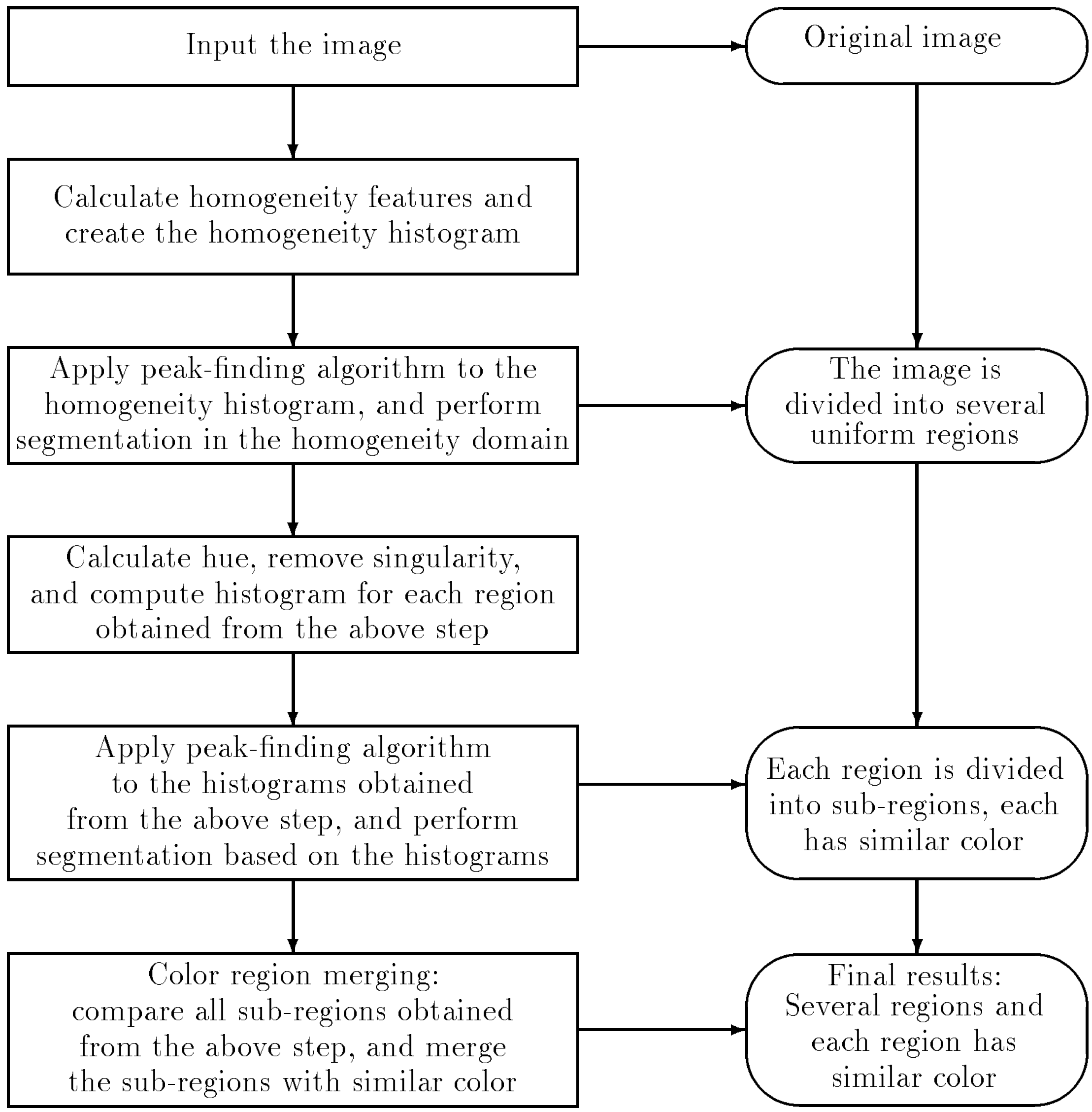


Fig. 2. The flowchart of the proposed approach.

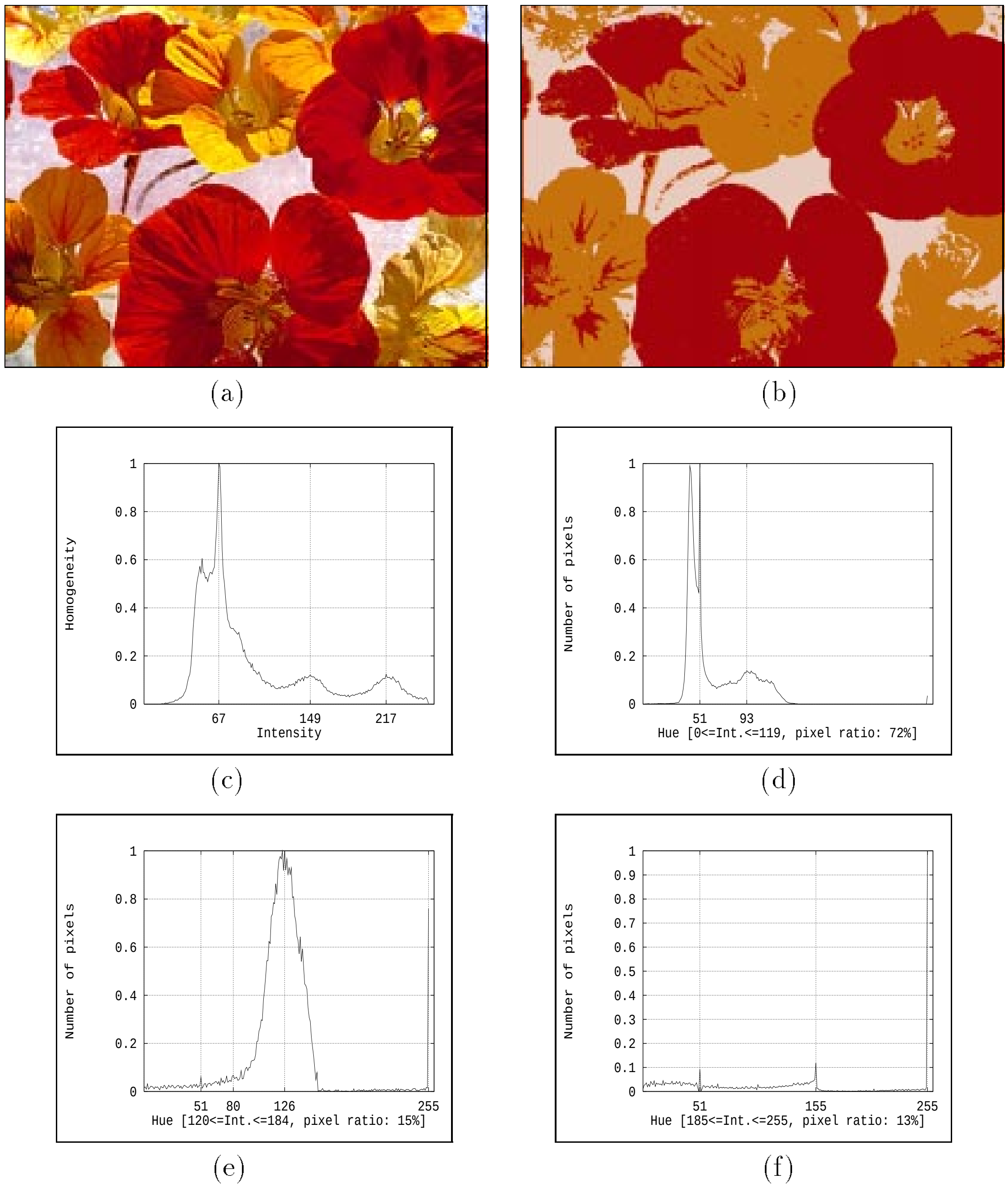


Fig. 3. The results of the proposed approach: (a) Original image "flowers" with RGB description ($256 \times 227 \times 256$), (b) Resulting image with 3 colors, (c) Histogram on homogeneity, (d)-(f) Histograms on hue for each uniform region.

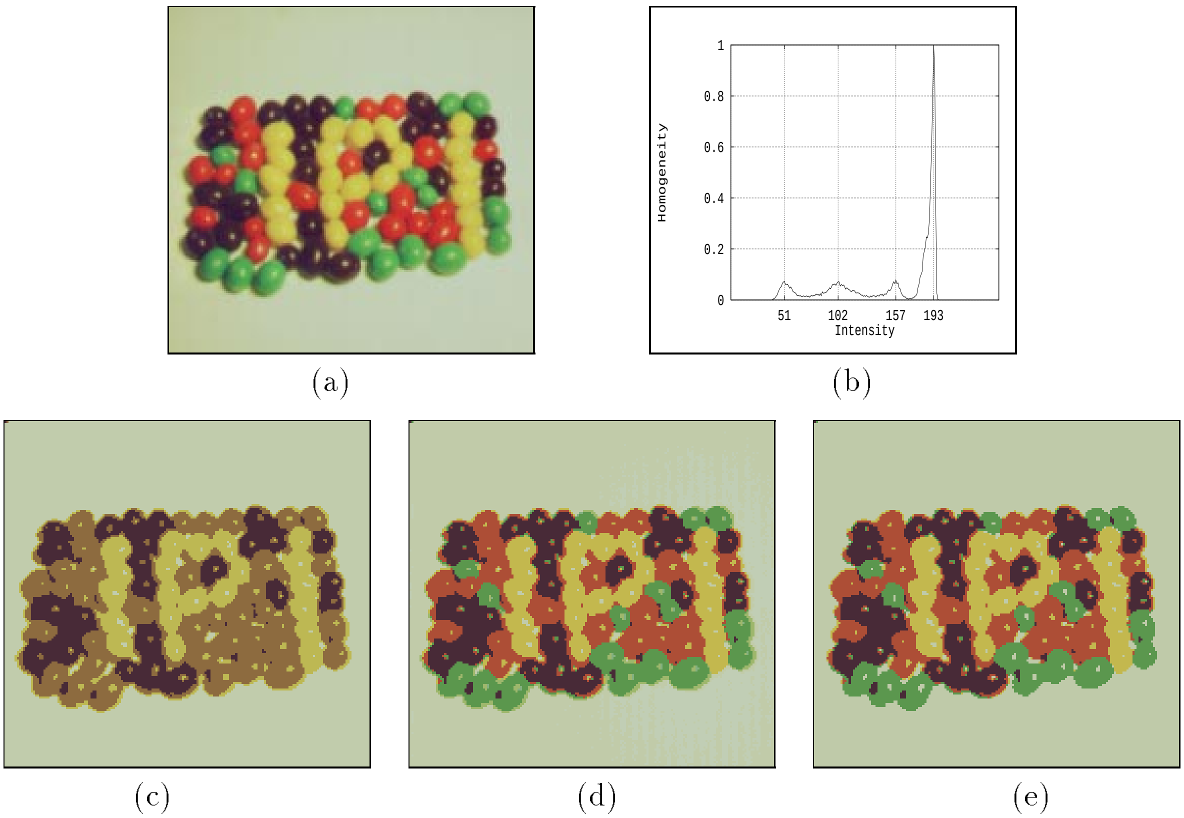


Fig. 4. The results of the proposed approach: (a) Original image "jlybeans" with RGB description (151 × 172 × 194), (b) Histogram on homogeneity, (c) The result of the first phase has 4 segments (colors), (d) The result before merging has 8 segments (colors), (e) The final result with 5 colors.

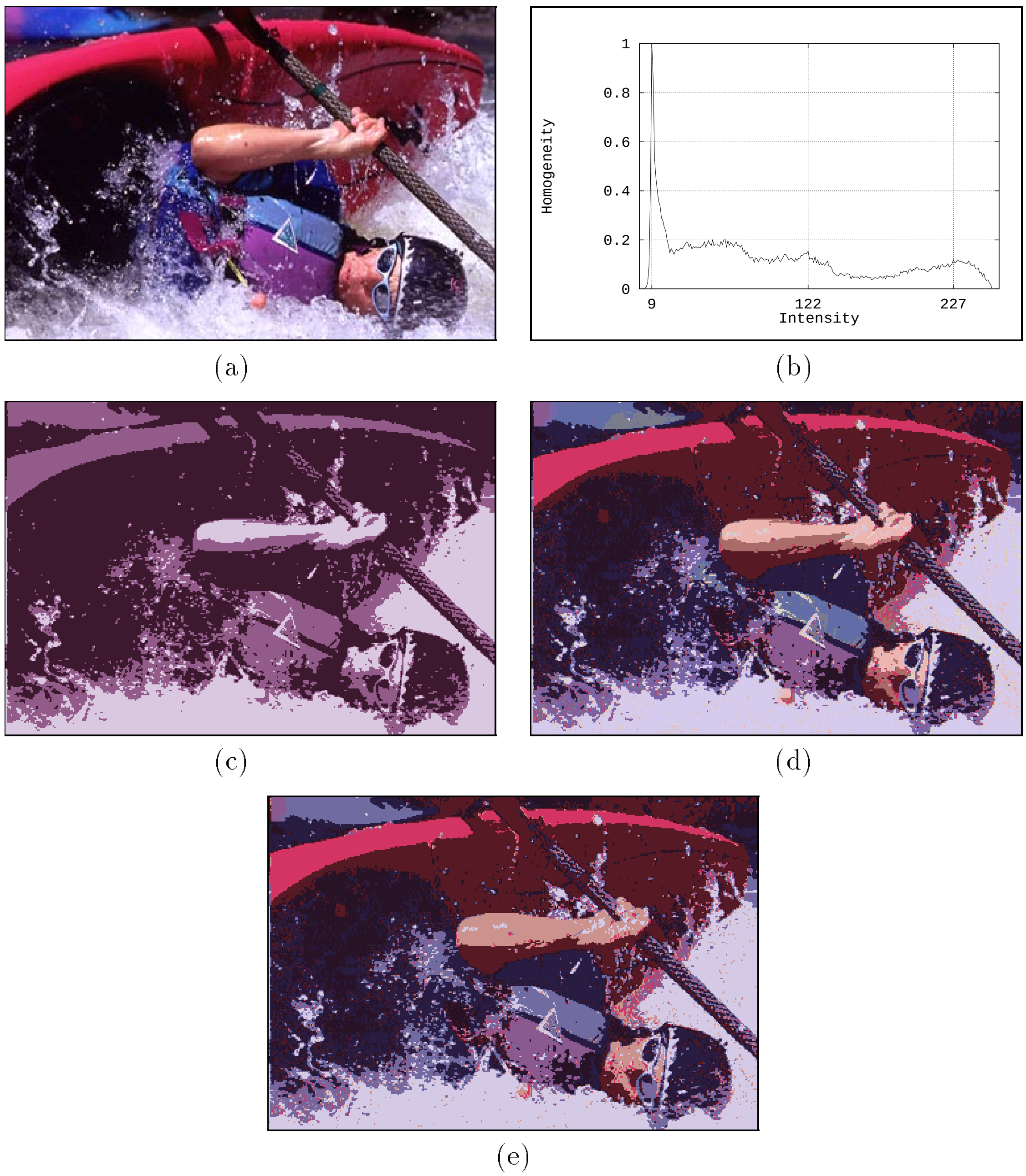
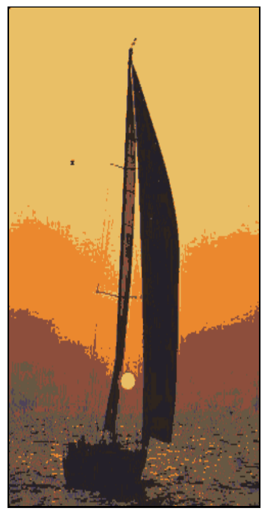


Fig. 5. The results of the proposed approach: (a) Original image "kayak" with RGB description (256 $\times$ 256 $\times$ 256), (b) Histogram on homogeneity, (c) The result of the first phase has 3 segments (colors), (d) The results before merging has 16 segments (colors), (e) The final result with 8 colors

(a)

(b)

(c)



(d)

(e)

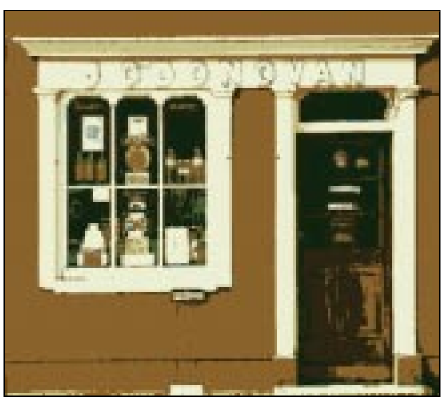
(f)

Fig. 6. The results of the proposed approach: (a) Original image "lake" with RGB description ($256 \times 256 \times 256$), (b) Resulting image with 5 colors, (c) Original image "sail" with RGB description ($190 \times 254 \times 256$), (d) Resulting image with 7 colors, (e) Original image "splash" with RGB description ($195 \times 130 \times 243$), (f) Resulting image with 3 colors.

Fig. 7. The results of the proposed approach: (a) Original image "river" with RGB description ($256 \times 256 \times 256$), (b) Resulting image with 6 colors, (c) Original image "fashion" with RGB description ($256 \times 256 \times 256$), (d) Resulting image with 4 colors.

(a)

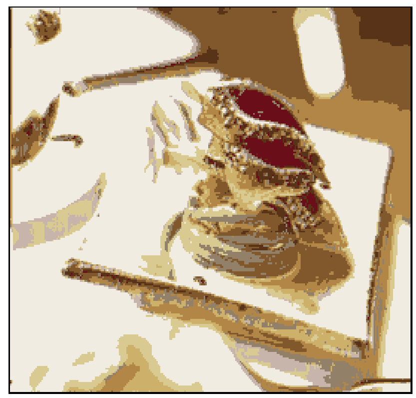
(b)



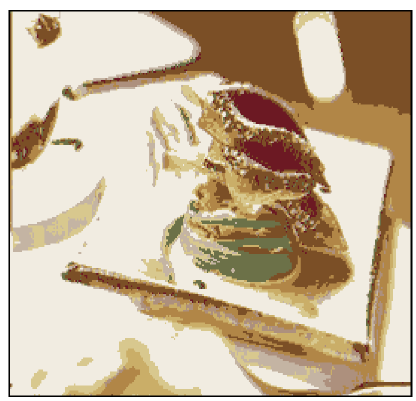
(c)

Fig. 8. Comparison with the results using only global information: (a) Original image "cafe" with RGB description (256 × 256 × 256), (b) Segmentation based on intensity and hue (5 colors), (c) Segmentation based on homogeneity and hue (7 colors).

(a)



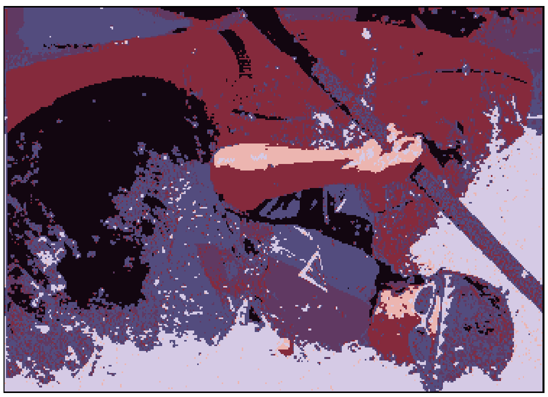
(b)



(c)

Fig. 9. Comparison with the results using only global information: (a) Original image "lemon" with RGB description (256 × 241 × 256), (b) Segmentation based on intensity and hue (9 colors), (c) Segmentation based on homogeneity and hue (9 colors).

(a)



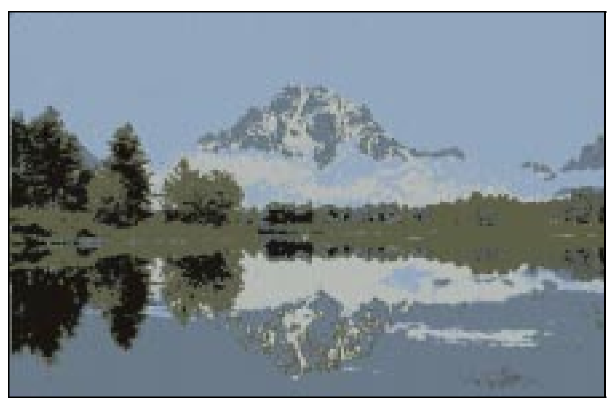
(b)

(c)

Fig. 10. Comparison with the results using only global information: (a) Original image "kayak" with RGB description (256 × 256 × 256), (b) Segmentation based on intensity and hue (6 colors), (c) Segmentation based on homogeneity and hue (8 colors).

(a)

(b)



(c)

Fig. 11. Comparison with the results using homogeneity and RGB: (a) Original image "mount" with RGB description (256 × 255 × 254), (b) Segmentation based on homogeneity and RGB (6 × 6 × 5), (c) Segmentation based on homogeneity and hue (7 colors).

(a)

(b)

(c)

Fig. 12. Comparison with the results using homogeneity and RGB: (a) Original image "flowers" with RGB description (256 × 227 × 256), (b) Segmentation based on homogeneity and RGB (7 × 4 × 4), (c) Segmentation based on homogeneity and hue (3 colors).

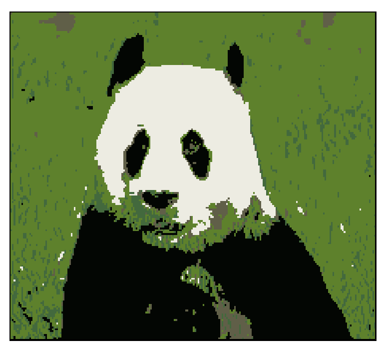
(a)

(b)

(c)

(d)

(e)



(f)

Fig. 13. Comparison with the results using homogeneity and RGB: (a) Original image "door" with RGB description ($214 \times 230 \times 217$), (b) Segmentation based on homogeneity and RGB ($8 \times 9 \times 7$), (c) Segmentation based on homogeneity and hue (8 colors), (d) Original image "panda" with RGB description ($256 \times 256 \times 256$), (e) Segmentation based on homogeneity and RGB ($6 \times 4 \times 6$), (f) Segmentation based on homogeneity and hue (5 colors).