Integrating Region Distribution and Edge Detection for Color Image Segmentation

Yong-Ren Huang, Chung-Ming Kuo, Chaur-Heh Hsieh, Chia-Yu Pai
Department of Information Engineering I-Shou University
E-Mail: D890302D@stmail.isu.edu.tw

Abstract

In this paper, we propose a new approach which integrates the region distribution and edge detection to achieve ideal results of color image segmentation. Firstly, we develop a histogram classification algorithm to classify color vectors and create binary trees for color image quantization, and we generate seeds for region growing from histogram classification automatically and obtain the initial regions. And further, the fast entropic thresholding algorithm and our multi-entropic-thresholding algorithm provide mechanisms for global and local edges detection respectively. Then, we integrate the information of regions and edges for eliminating texture regions, as well as small regions are merged by the size ratio of regions to overall regions with the same color vectors. Finally, the experimental results will show the results of color image segmentation.

Keywords: *automatic seed generation, color vectors, seed region growing, region merging, fast entropic thresholding.*

1. Introduction

Image segmentation is an essential and fundamental technique to various image analysis tasks, which splits an image into several regions with inner pixels of same homogeneous features. It is important to locate the regions of objects and texture areas in color images for some further applications. The proposed techniques for image segmentation could be classified into several categories, like region-based, edged-based and hybrid.

The seeded region growing approach controlled the formations of regions by given seeds which are individual pixels or regions [1][2]. In [7], an unsupervised texture segmentation method which utilized the distribution of local patterns, started with region-based segmentation and improved by pixel-wise classification scheme. The method which manipulated novel average contrast and peripheral contrast discontinuity measures to improve region growing had proposed in [12].

The edge occurs while adjacent regions are with different gray levels [3]. A new boundary detection approach, edge flow technique [4], introduced the process of directional color gradient to detect the boundary for homogenous regions. G. Iannizzotto and L. Vita [5] proposed an edge-based segmentation algorithm built on active contour with low computational complexity. A hierarchy configuration of the watersheds provided multiscale analysis of intensity minima in the gradient magnitude image [6]. Boundary decision by marker extraction and color measurement method had been proposed for color image segmentation in [13].

Pavlidis et al. presented a method which combined region growing and edge detection for image segmentation [8]. J. Fan *et al.* proposed a fast entropic thresholding algorithm for edge detection and integrated SRG for color image segmentation [9]. M. Tabb et al. proposed multiscale image segmentation by integrating region and edge detection in [10]. The segmented objects can be decrypted by a three dimension graph from segmentation results of integrating edges and regions [11].

In this paper, we propose a new approach which integrates the region distribution and edge information to improve the accuracy of segmentation results. We start with histogram classification and automatic seeds generation. Moreover, we employ fast entropic thresholding algorithm [9] for detecting global edges and develop a multi-entropic-thresholding algorithm for detecting local edges. Then, we introduce a new algorithm which integrates the information of edges and regions for elimination of texture regions. Finally, the reasonable results of image segmentation are achieved by our approach. Section 2 will describe our proposed methods in detail. The experimental results of our new approach will show in section 3, and we describe our conclusions in section 4.

2. Proposed Methods

Generally, it is hard to get a good segmentation result by either region-based or edge-based method. So, we utilize the distribution of region growing and information of edge detection for elimination of texture regions to obtain accurate segmentation results. The YUV components of images are employed in the segmentation processes. The detail algorithms are described in following sub sections.

2.1 Histogram Quantization and Automatic Seeds Generation

The histogram of gray levels of image is based on the probability. So, it is feasible to analysis the feature of histogram for gray-scale quantization. The SRG (seeded region-growing) approach performs image segmentation with a set of seeds and provides closed regions in image segmentation [1][2]. This approach is controlled by choosing the seeds which will influence the result of segmentation. In this sub-section, we will develop histogram classification for gray-scale and color images and automatic generation of seeds for SRG.

A. Histogram Classification

For classifying the representative gray-scale images of histogram, we will exploit the statistic characterization of histogram to obtain the classifications with dominant features. Let H is a set of gray-scale histogram and each subset can be denoted as H_i^j , where $j = 0,1,2,...,n$, i=$0,1,2,...,2^j - 1$ and $H_i^j \subseteq H$. The j and i are the times of divisions and the indexing number of subsets respectively. Let μ_i^j and σ_i^j are the mean value and standard deviation of each subset H_i^j, which are given by

$$\mu_i^j = \frac{1}{N_i^j} \sum_{h \in H_i^j} f_h \cdot h \tag{1}$$

$$\sigma_i^j = \left[\frac{1}{N_i^j} \sum_{h \in H_i^j} \left(f_h - \mu_i^j \right)^2 \cdot h \right]^{\frac{1}{2}} \tag{2}$$

where N_i^j is sum of gray-level values of histogram in each subset, f_h is the histogram value of gray-level h and $h \in H_i^j$. When the standard deviation σ_i^j is greater than a predefined threshold value T_h, the subset H_i^j will be partitioned into two subsets H_{2i}^{j+1} and H_{2i+1}^{j+1} by mean value μ_i^j automatically, where $H_{2i}^{j+1} = \{h \mid h \le \mu_i^j, h \in H_i^j\}$ and $H_{2i+1}^{j+1} = \{h \mid h > \mu_i^j, h \in H_i^j\}$. At the same time, we build a binary tree for remaining the mean value in each node. The value of binary tree root is μ_0^0. And the binary tree will extend two new nodes which values are μ_{2i}^{j+1} and μ_{2i+1}^{j+1}, where left node μ_{2i}^{j+1} < right node μ_{2i+1}^{j+1} . Fig.1 shows the division processes of histogram. The creation of binary tree shows in Fig.2. The steps of histogram classification will iterate automatically until the standard deviation of each subset is not greater than the predefined threshold value.

For the color image, we develop a new algorithm for the classification of hybrid components. We exploit the classification of gray-scale histogram for each component of YUV to construct three binary trees individually, and the values of terminal nodes of each binary tree can be grouped into three sets denoted as $G_y = \{g_y^0, g_y^1, \cdots, g_y^n\}$, $G_u = \{g_u^0, g_u^1, \cdots, g_u^m\}$, $G_v = \{g_v^0, g_v^1, \cdots, g_v^p\}$ respectively, where n, m and p represent the number of classified gray levels in each component. Afterward, we combine the elements of G_y, G_u and G_v to obtain a set of color vectors V=$\{v_0, v_1, \ldots, v_{nxmxp}\}$, where the color vector $v_{n \times m \times p} = \{g_y^n, g_u^m, g_v^p\}$ and $g_y^n \in G_y, g_u^m \in G_u, g_v^p \in G_v$. Then, we sort the $\{g_y^n, g_u^m, g_v^p\}$ by keys of g_y^n, g_u^m and g_v^p orderly, and map to the set of indexing numbers of colors vectors denoted as S=$\{0,1,2,\ldots, n \times m \times p\}$. Because the set S is still too large, we have to classify S similarly in gray-scale histogram classification. The iteration of color classification will stop while the standard deviation is not greater than a predefined threshold T_c. Fig.3 shows the construction of color vectors classification using binary trees. In the implementation of color vectors classification, we predefine four threshold values for classifying YUV components and color vectors individually.

B. Quantization and Automatic Seeds Generation

For splitting the image into closed regions, we will employ SRG which evolves inductively from the seeds. After the automatic classification of histogram, the deviation of each classified histogram subset is under the threshold, and the binary tree, which nodes contain the mean values of classified histogram subsets, has been constructed. We can quantize original gray levels by the binary tree. Each original gray level of image pixel will compare to the values of nodes of binary tree. If the value of original gray level is not greater than the value of node, the comparison will toward to left sub tree. On the other hand, the process move to right sub tree. For color vectors quantization, the original YUV values of color pixels will be quantized by binary trees of Y, U and V individually, and match color vectors of set V for obtaining the indexing numbers of set S. Then, the indexing numbers search the classified values in the binary tree for acquiring the color vectors of quantization.

Seeds generation is essential and important in SRG. The values of nodes in binary tree are the mean values of subsets after histogram divisions. And the mean value of subset can be chosen to represent the gray levels of the subset. So, the set of seeds can be generated automatically by the terminal nodes of the binary tree without any extra computation.

2.2 Global Edge Detection and Local Edge Detection

Edge is important information of analysis of image objects. In this work, we utilize four directional Sobel operators to compute the gradient of image, denoted in Fig.4, and then employ the fast entropic thresholding algorithm [9] to obtain an optimal threshold value for detecting the global edges of image objects. The global edges are important information which figures out the boundaries and locations of objects. But the process of single thresholding algorithm ignores too much edge information for further analysis of texture regions. So, we develop a block multi-thresholding algorithm for local edges detection in this sub section and manipulate both global and local edges in the following process.

A. Global Edge Detection

The fast entropic thresholding algorithm has been proposed in [9]. The maximum range of edge strength is [0, M] and there are f_i pixels with edge strength, where $i \in [0, M]$. The given threshold value is denoted as T. The probabilities of edge and non-edge are $P_e(i)$ and $P_n(i)$ respectively. And the entropies of threshold are $H_n(T)$ and $H_e(T)$ have derived in [9].

The recursive algorithm of entropic thresholding is given by

$$H_n(T+1) = \frac{P_n(T)}{P_n(T+1)} H_n(T) - \frac{f_{T+1}}{P_n(T+1)} \log \frac{f_{T+1}}{P_n(T+1)} - \frac{P_n(T)}{P_n(T+1)} \log \frac{P_n(T)}{P_n(T+1)} \tag{3}$$

$$H_e(T+1) = \frac{P_e(T)}{P_e(T+1)} H_e(T) + \frac{f_{T+1}}{P_e(T+1)} \log \frac{f_{T+1}}{P_e(T+1)} - \frac{P_e(T)}{P_e(T+1)} \log \frac{P_e(T)}{P_e(T+1)} \tag{4}$$

The computational complexity is reduced significantly because the recursive calculations are processed only by adding the increment part in the iterative steps. We can obtain a global threshold value and achieve the result of global edge detection.

B. Local Edges Detection

For detecting more local edges, we would like to modify the single thresholding algorithm. First, the original image is partitioned into blocks with two pixels extension for connectivity of object edges denoted in Fig.5. Then, we detect edges in each block by local fast entropic thresholding algorithm independently. Due to processing in blocks, there will be multi-entropic-thresholding values for fast entropic algorithm. Let the maximum value of the m'th block gradient is denoted as M_m and T_m is a given threshold value of the current block. There are f_{Bi} pixels of the block with edge strength, where $Bi \in [0, M_m]$. The probabilities of edge pixels $P_e(Bi)$ is given by

$$P_n(Bi) = \frac{f_{Bi}}{\sum_{h=0}^{T_m} f_h}, \quad 0 \le Bi \le T_m \tag{5}$$

The probabilities of non-edge pixels in the block are denoted as $P_n(Bi)$.

$$P_e(Bi) = \frac{f_{Bi}}{\sum_{h=T+1}^{M_m} f_h}, \quad T_m + 1 \le Bi \le M_m \tag{6}$$

The entropies of $P_e(Bi)$ and $P_n(Bi)$ in the current block are

$$H_n(T_m) = -\sum_{Bi=0}^{T_m} P_n(Bi) \log P_n(Bi) \tag{7}$$

$$H_e(T_m) = -\sum_{Bi=T+1}^{M_m} P_e(Bi) \log P_e(Bi) \tag{8}$$

The optimal threshold value $\bar{T}_m$ can be obtained by

$$H(\bar{T}_m) = \max_{T_m = 0,1,2,\ldots,M_m} \{H_n(T_m) + H_e(T_m)\} \tag{9}$$

The similar recursive algorithm of fast entropic thresholding is processes as (4)-(5).

For obtaining complete edges, we process the global local edge detection of color image in each Y, U and V component. Let I is the original color image. The detected edges of Y, U and V images denoted as $E_Y(I)$, $E_U(I)$ and $E_V(I)$ respectively. Then, we union the edges to acquire the result of edge detection, e.g., $E(I)$, as

$$E(I) = \{E_Y(I) \cup E_U(I) \cup E_V(I)\} \tag{10}$$

2.3 Elimination of Texture Regions and Region Merging

After SRG, the scattered regions, which could not represent objects, might be small regions or texture regions. So we have to eliminate the texture and small regions in order to obtain meaningful and semantic regions or objects for further manipulations.

A. Elimination of Texture Regions

The texture regions might be not meaningful but only parts of background or real objects. Although the texture regions are tight and chaotic, there still is some similarity characterization in a texture area, such as the distance of color vectors or gray levels in adjacent regions. By observing the results of edge detection, many unclosed boundaries always appear in the texture regions. So, for eliminating texture regions, we both consider the similarity of adjacent regions and the information of regions and edges. First, the similarity distance $S_d(r_i, r_j)$ of adjacent regions r_i and r_j is calculated by

$$S_d(r_i, r_j) = ((QY_{r_i} - QY_{r_j})^2 + (QU_{r_i} - QU_{r_j})^2 + (QV_{r_i} - QV_{r_j})^2)^{\frac{1}{2}} \tag{11}$$

Where $(QY_{r_i}, QU_{r_i}, QV_{r_i})$ and $(QY_{r_j}, QU_{r_j}, QV_{r_j})$ are the values of quantized color vector in region r_i and r_j respectively. If $S_d(r_i, r_j)$ is not greater than 100, the adjacent regions would be considered to group into a same texture region by information of edges and regions, and we execute the following step.

Let the edge pixels of two adjacent regions r_i and r_j be $RE(r_i, r_j)$, and $E(r_i, r_j)$ be the pixels from edge detection. We project edges onto to regions and calculate the ratio R_r

$$R_r = \frac{E\left(r_i, r_j\right) \cap RE\left(r_i, r_j\right)}{RE\left(r_i, r_j\right)} \quad (12)$$

While R_r is not greater than threshold T_r, the adjacent regions will be merged. Fig.6 shows the integration of region and edge. We eliminate the texture regions by steps from Eq.(11) to (12). Because we just consider the similarity feature of color vectors in the process of texture regions elimination, so some small regions with independent color features still exist. We have to implement small regions merging to reach the reasonable segmentation results.

B. Region Merging using Color Vectors

For merging small regions, we consider the ratio of region size to total regions with same color vector. After elimination of texture regions, we obtain a set of regions with quantized color vectors formed as $R_{V_i} = \left\{r_{V_i}^1, r_{Vi}^2, \cdots, r_{V_i}^{k_i}\right\}$, where $i = 1, 2, ...n$, V_i is color vector by color histogram quantization, $r_{V_i}^{k_i}$ is the k_i'th region with color vector V_i. Let $A(r_{V_i}^{k_i})$ is the region size of $r_{V_i}^{k_i}$ and $A(R_{V_i})$ is the size of all regions with color vector V_i in whole image. The ratio is given by

$$Ar = \frac{A(r_{V_i}^{k_i})}{A(R_{V_i})} \quad (13)$$

While Ar is smaller than a predefined threshold T_a, the region $r_{V_i}^{k_i}$ will be merge into a adjacent region which is with minimum color distance d, described in (11). The steps of region merging iterate until no region size is less than the predefined threshold.

3. Experimental Results

In our experiments, there are three kinds of images, e.g., human, animal and scenery, in our test. The frame sizes of animal images are 384x256 and 256x384. Fig.7 shows the result of histogram quantization and SRG by our proposed classification of color vectors. The local edge and global edge detections present in Fig.8. More edge information is obtained by our multi-entropic-thresholding method.

In Fig.9 and Fig.10, we represent the results of image segmentation, and describe the predefined threshold values and number of seeds and regions in Table I. The images of index (a) are original color images. The images of index (b) show the results SRG by our color histogram quantization and automatic seeds generation. The images, index (c) and (d), show the results of integrating regions and edges from local and global edge detection respectively.

In practical implementation, we can execute the segmentation by integrating local and global edges sequentially. Fig.11 shows the results which integrate the local and global edges orderly. The boundary of salient object is obtained efficiently, and the texture regions are split more accurately.

4. Conclusions

In this paper, we present a new approach which integrates the region distribution and edge information to improve the accuracy of segmentation results for color images. For the quantization of color images, we employ the distribution of histogram to classify the Y, U and V components individually, and construct binary trees to create the set of color vectors. The index set of color vectors is classified to build a binary tree for histogram quantization. The seeds are generated automatically by histogram classification for SRG. Then, we utilize the fast entropic thresholding algorithm to obtain global edges, and we modify the algorithm to develop the multi-entropic-thresholding algorithm for local edges detection. Moreover, we propose a new method, which integrates the information of regions and edges, for elimination of texture regions, and then small regions are merged by the size ratio of regions which are in same color vectors. Finally, the experimental results show good performance in our approach.

Reference

[1]. R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 16, No. 6, pp.641-647, *June 1994.*

[2]. Y. L. Chang and X. Li "Adaptive Image Region-Growing," *IEEE Trans. on Image Processing,* Vol. 3, Issue: 6, pp. 868–872, Nov. 1994.

[3]. J. Canny, "A Computational approach to edge detection," *IEEE Trans. Patterm Anal. Machine Intel.l,* Vol. PAMI-8, pp.667-698, Nov. 1986.

[4]. W. Y. Ma and B. S. Manjunath, "Edge Flow: A Technique for Boundary Detection and Image Segmentation," *IEEE Trans. Image Processing,* Vol. 9, No. 8, pp.1375-1388, Aug. 2000.

[5]. G. Iannizzotto and L. Vita, "Fast and Accurate Edge-Based Segmentation with No Contour Smoothing in 2-D Real Image", *IEEE Trans. Image Processing,* Vol. 9, No. 7, pp.1232-1237, July 2000.

[6]. John M. Gauch, "Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies," *IEEE Trans. Image Processing,* Vol. 8, No. 1, pp. 69-79, Jan. 1999.

[7]. T. Ojala and M. Pietikainen, "Unsupervised Texture Segmentation Using Feature Distributions," *Pattern Recognition,* Vol.32, pp. 477-486, 1999.

[8]. T. Pavilidis and Y. T. Liow, "Integrating region growing and edge detection," *IEEE Trans. Pattern Anal. Machine Intell.,* Vol. 12, pp. 255-233, 1990.

[9]. J. Fan, David. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing," *IEEE Trans. on Image*

Processing, Vol. 10, No.10, pp.1454-1466, Oct. 2001.

[10]. M. Tabb and N. Ahuja, "Multiscale Image Segmentation by Integrated Edge and Region Detection," *IEEE Trans. on Image Processing*, Vol. 6, No. 5, pp.642-655, May 1997.

[11]. D. M. Zhao and X. T. Zhang, "Range-Data-Based Object Surface Segmentation via Edges and Critical-Points," *IEEE Trans. on Image Processing*, Vol. 6, No. 6, pp.826-830, Jun. 1997.

[12]. S. A. Hojjatoleslami and J. V. Kittler, "Region Growing: A New Approach," *IEEE Trans. on Image Processing*, Vol. 7, No. 7, pp. 1079-1084, Jul. 1998.

[13]. H. Gao, W. C. Siu, and C. H. Hou, "Improved Techniques for Automatic Image Segmentation," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 11, No. 12, pp.1273-1280, Dec. 2001

Table I

Images	Threshold values
Claire	σ of Y = 15, U = 5, V = 5, color vectors = 10 T_r = 0.5, T_a = 0.01
Foreman Sitting Orangutan	σ of Y = 15, U = 3, V = 3, color vectors = 5 T_r = 0.5, T_a = 0.01
Elephant	σ of Y = 15, U = 2, V = 2, color vectors = 5 T_r = 0.5, T_a = 0.01

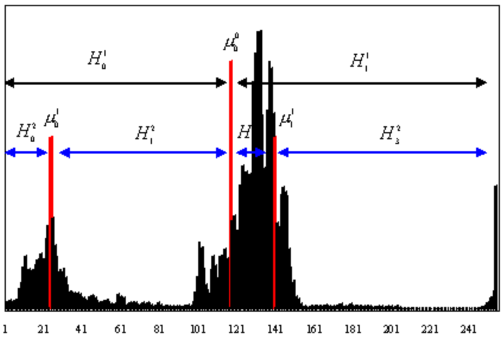

Fig.1 The division processes of histogram.

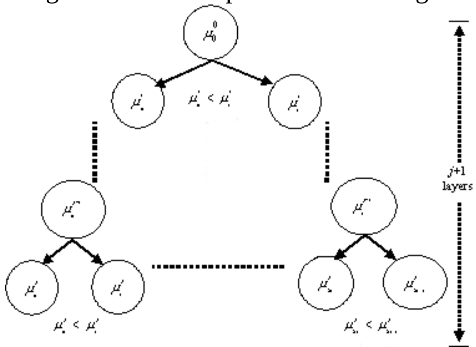

Fig.2 The creation of binary tree.

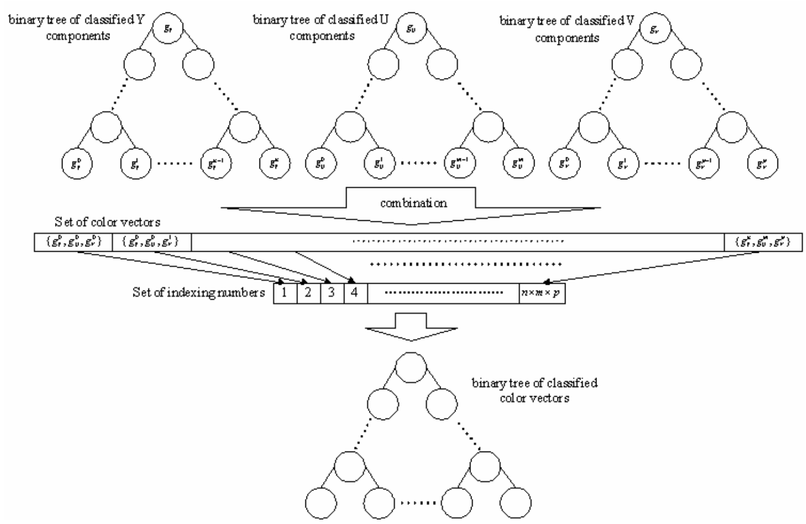

Fig.3 Classification of color vectors by binary tree of Y, U and V components.

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

0	1	2
-1	0	1
-2	-1	0

2	1	0
1	0	-1
0	-1	-2

Fig.4 Four directional mask coefficients of Sobel mask.

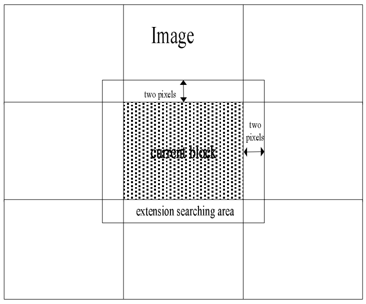

Fig.5 Block searching area for fast entropic thresholding algorithm.

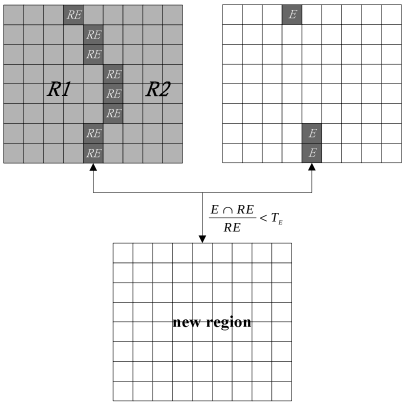

Fig.6 Region merging by integration of region and edge.

(a) (b)

Fig.7 The color vectors histogram classification of Foreman sequence. (a) original image. (b) quantized and segmented image.

(a) (b) (c)

Fig.8 Edge detection result. (a) original image. (b) local edge detection by block multi threshold. (c) global edge detection by single threshold.

(a) (b) (c) (d)

Fig.9 The results of Sitting Orangutan in size 256x384. (a) The original image. (b)Automatic seeds generation and SRG. (c) Texture elimination by local edge and region merging. (d) Texture elimination by global edge and region merging.

(a) (b) (c) (d)

Fig.10 The results of Elephant in size 384x256 (a) The original image. (b)Automatic seeds generation and SRG. (c) Texture elimination by local edge and region merging. (d) Texture elimination by global edge and region merging.

(a) (b)

Fig.11 The results of texture regions elimination of Sitting Orangutan in size 256x384. (a)original image. (b)integration of region and local /global edges