International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.9 (2016), pp.93-100
http://dx.doi.org/10.14257/ijmue.2016.11.9.10

Automatic Segmentation for Textured Object Images

Chang-Min Park[1] and Chang-Geun Kim[2*]

[1]*School of Undeclared Majors, YoungSan University, Busan, 612-743 Korea*
Email: cmpark@ysu.ac.kr
[2]*Department of Computer Engineering, Gyeongnam National University of Science and Technology, Jinju, 660-758 Korea*
Email: cgkim@gntech.ac.kr

Abstract

In this paper, we proposed an automatic segmentation method of object color images with irregular texture. Recently segmentation often used for the image retrieval and in the application. It is more important to approximate the regions than to decide precise region boundary. A color image is divided into blocks, and edge strength for each block is computed by using the modified color histogram intersection method that has been developed to differentiate object boundaries from irregular texture boundaries effectively. The edge strength is defined to have high values at the object boundaries, while it is designed to have relatively low values at the texture boundaries or in the interior of a region. The proposed method works based on small-size blocks, the color histogram of each of which is computed preliminarily once. Thus it works fast but provides rough segmentation. A hybrid color quantization method is used to select a small number of appropriately quantized colors quickly. The proposed method can be applicable for the segmentation in object based image retrieval.

Keywords: *Segmentation, edge, irregular texture, histogram intersection, quantization*

1. Introduction

Many segmentation methods can be classified roughly into three groups, which are local method, area-based method, and global optimization method [1]. And the local method relies on local information in an image for finding contours and edges of regions. In the image database, users usually want to retrieve images based on objects that are of interest. Therefore, the performance of the image database depends on whether or not objects in the image can be segmented. However, the objects often consist of several segments, so it is more important to segment an image into objects by ignoring the details in the image retrieval application.

Several approaches have been attempted to segment an image. The Blobworld [2] proposed new image segmentation methods. The [2] segmented the pixels into regions by modeling the joint distribution of color, texture, and position features with a mixture of Gaussians using EM algorithm and MDL. Both systems are designed to extract accurate edges by a pixel-based processing method. Thus, they have the drawbacks of over-segmentation by texture and an exponential increase of computation time. A, Dimai [3] proposed an region extraction algorithm based on local and area-based information of multidimensional features, such as luminance color and texture. Such algorithm detects automatically and robustly salient region-descriptors in image scenes. However, the extraction of more meaningful salient region-descriptors depends on the result of segmentation.

The Segmentation method for CBIR has to segment an image into the regions of interest, and it is more important to segment an image into the regions of the objects as well. Additionally, fast computation time is required. Therefore, it is necessary to

ISSN: 1975-0080 IJMUE

develop an efficient segmentation method for salient region. So we have defined a new measure based on the modified Histogram Intersection (HI) [4]. HI reflects the features of color distribution and texture in a region, and HI can be used to calculate the edge strength. But HI has the meaningful information for block above 3 x 3 pixels, we need to divide image by the suitable size of blocks and obtain the HI for each blocks.

A watershed transformation technique [5] is used to segment the edge strength image, which can generate closed boundary for each region. And watershed is applied to create the new image (M image) from the block image that is a gray scale image. However, the technique results in over-segmentation, so a region-merging method is developed in this paper. The method uses the common boundary strength between adjacent regions, which is defined as Sum of Edge strength of blocks on the common boundary. Two regions that are adjacent to a high-valued common boundary are prevented from being merged, because the common boundary can be an object boundary with high probability. The region-merging method also adopts the color histogram intersection technique in order to use information of color distribution in each region. This information is useful for preventing similarly color-textured regions from being merged into a new region.

In Section 2, we describe the quantization method of the color image. Section 3 describes the new method for computing block-based edge and texture similarity of block and *M*-image. The watershed transformation and region merging techniques are presented in Section 4. In Section 5, some experimental results are presented followed by our conclusions in Section 6.

2. Quantization of the Textured Color Image

First, an input image is quantized to a small number of colors. In general, color quantization methods aim to decrease the number of colors in the image without significantly degrading the image quality, but in the quantization for image segmentation, it is more important to make it easy to segment images. We will use less than or equal to twenty quantized colors while distortion of the region boundaries is minimized. Color quantization methods can be classified into two classes, which are uniform quantization and adaptive quantization. The adaptive quantization provides better results than the uniform quantization in the several aspects like as quantization error. There are some popular adaptive quantization methods; popularity algorithm, median cut algorithm, K-mean clustering algorithm, variance-based algorithm, and octree algorithm. The proposed quantization method is suggested in this paper, which first used the octree algorithm to select 30 quantized colors and then the K-means clustering algorithm to obtain less than or equal to 20 quantized colors. In the K-means clustering, the counts of pixels belonging to each quantized color by octree are used to calculate the center of each cluster in the every iteration. Before the K-means clustering method is applied, merging of similar quantized colors is performed. If the color distance between any two quantized colors in CIELAB color space is less than 6, they are merged into a new quantized color because they cannot be discriminated nearly. The K-means clustering algorithm works very fast because there are only 30 quantized colors. Therefore, the hybrid method can produce a good color quantized result with no much increasing processing time of the octree algorithm [6].

To quantize the image, *k*-means clustering algorithm can be used but takes much processing time. As 20~30 colors are sufficient for the image segmentations, we first quantize the image into 30 colors by using the octree quantization method, and then the *k*-means clustering algorithm is applied to obtain the final quantized image of 20 colors without much more processing time(Figure 1).

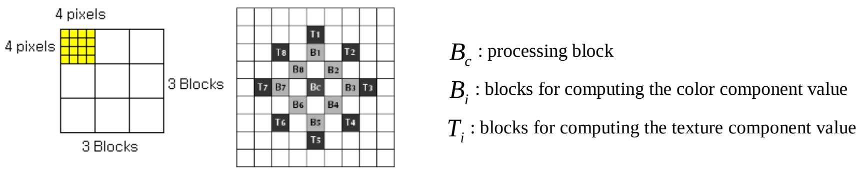
(a) Original Image (b) Quantized Image

Figure 1. Examples Images Using the Proposed Quantization Method

3. Component Value of the Block

The component value of local color distribution is computed using a modified color histogram intersection technique that enables us to measure the discontinuity of colors at boundaries of objects efficiently and to lessen the effect of discontinuity at finely textured regions. We divide an image into sub-blocks to extract the object's region boundary only excluding the region boundary of the texture pattern. The block size can change along by the texture pattern and the object size to be extracted. But, we have to fix the generalized block size. If the block is small, the color histogram does not have the meaningful information. Otherwise, if the block is large, the extracted edge is excessively blurred. Therefore, we determine the block size to 4 x 4 pixels irrelevant to the object size and texture pattern. The super-block consists of 3 by 3 blocks (Figure 2(a)).

The component value computing masks are composed of two kinds of the neighbor blocks. The one is B_i and the other is T_i. The B_i is used to compute the color component value of the block, and the T_i is used to compute the texture component value. Figure 2(b) shows the shape of the component value computing mask. The value of color component value represents discontinuities between the super-blocks of the B_c and the B_i. On the other hand, the value of texture component value depends on that the block is in the texture region. The component value to generate a component value image is computed by weighed summation of the color component value and the texture component value. The computed component value is assigned equally to the 16 pixels of the block (B_c).



(a) (b)

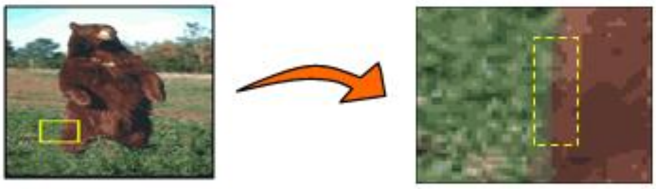
Figure 2. A Block (4×4 Pixels) and its Super-Block (a) and the Component Value Computing Mask (b)

3.2. Color Component Value

The color component value $\boldsymbol{D_C}$ for the block $\boldsymbol{B_c}$ is computed by using the modified color histogram intersection method as Eq. (1) between the processing block Bc and eight neighbor blocks *Bi*.

$$D_C = M - \sum_{k=1}^{N} H_C(k) = M - \sum_{k=1}^{N} \min\{ H_{I1}(k), \ldots, H_{I8}(k)\},$$
$$where\ H_{Ii}(k) = \min\{ H_{SBc}(k), H_{SBi}(k)\} \ ---(1)$$

The M and N are total number of pixels in a super-block and the number of quantized colors, respectively. The color component value represents discontinuity of color distribution at B_c in eight directions. Even if only one of neighbors of B_c has different color distribution with B_c, the color component value at B_c will be increased. Blocks with the higher color component values can be located near the region boundaries with a strong probability.



(a) Original Image (b) D_C Image

Figure 3. Example Image Using Color Component Value

Figure 3(b) is a part of Figure 3(a) using color component value. The dotted area in Figure 3(b) contains the boundary of the object and background. This boundary appears as vertical line. Therefore, we consider only left and right-neighbor blocks.

3.3. Texture Component Value

Texture component value is designed to weaken the strength of false boundaries at sparse and coarse texture patterns. The texture component value D_T at a block B_c is defined as described in Eq. (2) by subtracting the averaged sum of usual color histogram intersection between the super-block histogram of the B_c and one of each T_i from total number of pixels in a super-block.

$$D_T = M - \frac{1}{8}\sum_{i=1}^{8}\left[\sum_{k=1}^{N}\min\{H_{SB_c}(k), H_{ST_i}(k)\}\right] \qquad (2)$$

The neighbor blocks T_i's for texture component value are determined as shown in Figure 2(b). Note that they are located farther from the B_c than the B_i's are. Actually, the second term in Eq. (2) is the averaged sum of color similarity between nearly overlapped super-blocks, so the texture component value represents color component value in relatively large-scale. Thus the texture component value weakens the component value at coarsely textured regions than the color component value. In the Figure 4(b), we can see that boundaries of regions with sparse and coarse texture have low texture component value.

(a) Original Image (b) D_T Image

Figure 4. Example Image Using Texture Component Value

3.4. Generation of Block Component Value Image

The total block component value is computed by the summation of the color component value and the texture component value of the block (Figure 5). According to applications, the weight of each measure can be changed, but by default each measure is treated equally. The lower the total block component value is, the nearer the block is to object boundaries.

Figure 5(d) shows the K - image for the original image in Figure 5(a), when the weight factor is set to 0.5. We can see that component values at the boundaries of the four textured regions are greater than those at the texture boundaries. Thus the boundaries of the regions can be easily discriminated from the texture boundaries.

$$K = 0.5\ m_B + 0.5\ m_T \tag{3}$$

(a) Original Image (b) Quantized Image (c) D_C Image (d) K - Image

Figure 5. Result Image Using Block Component Value

4. Segmentation Using the Watershed Transformation

Even though component value images have relatively high values at boundaries of textured regions, there are still many noise peaks and broken boundaries. Thus the edge-based segmentation approach cannot be well applied to K-images because of difficulty of edge localization and edge linkage. The watershed transformation is used for segmenting K-images in this paper, which is one of the region-based segmentation methods.

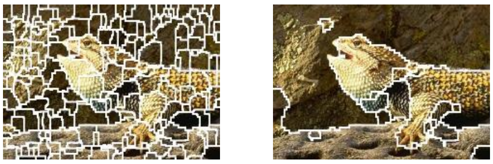
4.1. The Watershed Transformation and Region Merging

There are three types of watershed transformations: immersion, rain-falling and hill climbing. In this paper, we apply the immersion method [7] to K-images. At first, in the sorting step, pixels are sorted by their component value. In the flooding step, from the seed pixels that have the lowest component value, every pixel adjacent to the each seed is labeled or is set to watershed depending on the predefined rules. If any adjacent pixel of the current location has a different label from the current pixel, or is marked as watershed, then the next seed is selected. The flooding step is continued until all pixels are labeled or are set to the watershed. Figure 6(a) shows the watershed transformation result of the K-image in Figure 5(d). The boundaries of the four textured regions are well localized. However, we can see that the transformation tends to over-segment the K-image. Thus, a region merging technique should be applied to.

To merge the over-segmented regions, first of all, we make a region adjacency graph and then define common boundary strength (BS). The strength is defined in the common boundaries between two adjacent regions as Eq. (4). A common boundary of two regions is defined as the boundary that two regions share. In Eq. (4), Nt is the number of total pixels in the common boundary and Ns is the number of pixels that have a higher component value in the common boundary than a threshold Ts of the component value.

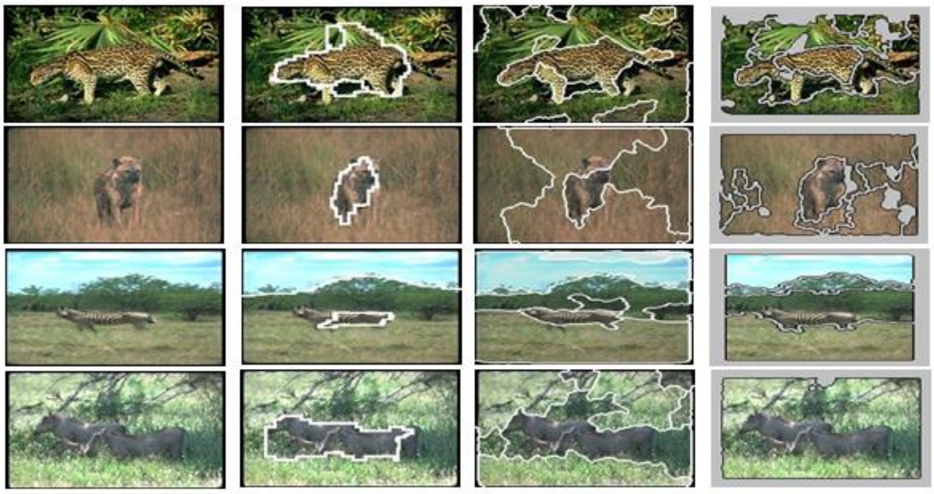
$$BS = \frac{Ns}{Nt} \tag{4}$$

The second procedure of region merging is to merge adjacent regions that have similar color distribution. We used the histogram intersection to merge those adjacent regions. If small regions remain without being merged to adjacent regions, these regions are also merged to the adjacent region that has the largest BS. Figure 6(b) shows the final merging result for the watershed transformation result in Figure 6(a). We can see that the four textured regions are well segmented.



(a) (b)

Figure 6. Result Image of Watershed Transformation (a) for Figure 1 (a) and the Merging Result of the Image in Figure 1 (a)



(a) Original Image (b) Proposed method (c) Netra (d) Blobworld

Figure 7. Original Images and Result Images

5. Experimental Results

The proposed method was applied to segment variety test images that included the objects with various colors and color-textures. Figure 7 shows the segmentation results using the proposed method. The results of Netra, Blobworld and the proposed method for test images are compared. The threshold values of common boundary strength and the similarity of color distribution were set to 0.8 of maximum value in BS and histogram intersection respectively. Figure 7 (c),(d) shows an over-segmentation region due to texture in the interior of region. Sometimes our method makes false merge. According to experimental results, the proposed method can extract the meaningful object boundary well. And it does not produce over segmented regions. Our method, therefore, is believed to be more suitable for CBIR than the Netra or Blobworld.

6. Conclusion

We proposed an automatic color image segmentation method in this paper, which can classify textured color patterns as a region instead of separating them several regions. For this purpose, we proposed the block component value and applied the watershed method to extract the closed boundary of regions as possible. Over-segmentation incurred by watershed method can be reduced and the final segmentation can be obtained through the proper use of the common boundary strength and the color distribution property of interior region of adjacent regions.

Experiments have confirmed that the proposed method can process the closed region segmentation regardless of the texture pattern faster than existing methods. Still, the method remains to be improved to be able to separate objects completely from the background when objects and background are similar.

Acknowledgements

This paper was supports by research funding in 2016 from Youngsan University.

References

[1] A. H. Kam, T. T. Ng, N. G. Kingsbury and W. J. Fitzgerald, "Content based image retrieval through object extraction and querying", IEEE Workshop on 2000, Content-based Access of Image and Video Libraries, Proceedings, **(2000)**, pp. 91-95.
[2] C. Carson, S. Belongie, H. Greenspan and J. Malik, "Blobworld: Image segmentation using Expectation-Maximization and its application to image querying", IEEE Transactions on Pattern Analysis and Machine Intelligence, to be Published.
[3] A. Dimai, "Unsupervised Extraction of Salient Region-Descriptors for Content Based Image Retrieval", International Conference on Image Analysis and processing, **(1999)**, pp. 686-691.
[4] C. G. Kim, S. G. Kim and H. J. Kim, "A Method for identification of Harmful Video Images Using a 2-Dimensional Projection Map", Journal of information and communication convergence engineering(JICCE), **(2013)**, pp. 62-68.
[5] J. Y. Herdeberg, F. Scmitt, H. Brettel, J. P. Crettez and H. Maitre, "Spectrophotometric imaging analysis of fine art paintings", Proceeding IS&T and SID's 4th color imaging conference: color science, Systems and Applications, **(1996)**, pp. 50-53.
[6] M. Gervautz and W. Purgathofer, "A Simple Method for Color Quantization: Octree Quantization", in Glassner, A.S. (Ed.), Graphics Gems, Academic Press, **(1990)**, pp. 287-293.
[7] L. Vincent and P. Soile, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", IEEE Trans. PAMI, vol. 13, no. 6, **(1999)**, pp. 583-597.

Authors

Chang Min Park, received the BS., M.S., and Ph.D. degrees in computer science from Dortmund University , Germany and Busan National University, Korea in 1990, 1996, and 2006 respectively. Since 1997, he has been with Youngsan University of School of Undeclared Majors, Busan, Korea. His research interests include image processing, image search and image representation.

Chang Geun Kim, received the BS., M.S., and Ph.D. degrees in computer science from Gyeongsang National University and Gyeongnam University, Korea in 1985, 1991, and respectively. Since 1995, he has been with Gyeongnam National University of Science and Technology, Korea, Where he is currently professor in the Department of Computer Science & Engineering. His research interests include computer networking, data communication, and e-business.

