Content-Based Image Retrieval Based on Rectangular Segmentation

Chan-Fong Wong and Chi-Man Pun

Department of Computer and Information Science
University of Macau
Macau SAR, China

Abstract: - In this paper, we present a new framework for effective content-based image retrieval (CBIR) based on rectangular segmentation. In image segmentation, speed is more important than accuracy in CBIR. We propose a new rectangular approximate image segmentation to solve the problem. We also develop a significance function to reflect the importance of different position in image, and improve the segmentation and retrieval performance. Finally, we present a similarity measure between images with multi-objects. Experimental results show that the proposed method is more efficient and achieves higher precision on image retrieval of a large dataset.

Key-Words: - Color quantization, color similarity, image retrieval, image segmentation, rectangular segmentation.

1 Introduction

With the rapid development of multimedia database, the application of digital library and image search engine become more and more widely used. Traditional text-based data retrieval approach is not effective and efficient in image retrieval. This is because the perceptual meaning of an image usually hard to be described by a few keywords accurately. Also it is very inefficient to manually extract keywords from images.As the size of image database growing larger, we need a methodology for querying images by its content. Recent years, content-based image retrieval (CBIR) emerged to be a quite active field. In CBIR each image is represented by image features such as color, texture and shape. Images are retrieved by computing the relevance based on the similarity of features to the querying image. Color is one of the most important features in general images. A color quantization method known as color histogram [1] is widely used in comparing images with their colors. Color histogram is derived by first quantize colors in the image into a number of bins in a specific color space, and counting the number of image pixels in each bin. Texture is also an important image feature in CBIR. The texture feature of image is extracted by first applying the wavelet transform [2, 3]. This process transforms the image into some wavelet coefficients, which represent the high frequency information of the image in multi resolution and direction. Afterward the wavelet energy [4, 5] is extracted from the wavelet coefficient as texture feature.

However, images with similar global features not always perceptually similar. This is because global features lack of spatial information about images. The simplest way to extract local image features is by partitioning image into fix blocks, and extract features block by block [6, 7]. Another way to extract local features is by image segmentation, which partitions image into visually homogeneous regions that can describe semantic meaning of the image. One approach to image segmentation is by clustering [8, 9], which categorize image pixels into a few clusters. Each cluster forms an object, and the local features can be extracted from these objects. Normalized cuts [10] is also a well-known image segmentation approach. In this approach, the nodes of the graph are the points in feature space, and edge is formed between pairs of nodes. This approach is closely related to graph theory. Other image segmentation methods include: Edge detection [11], Region growing [12], Watershed transform [13] etc. In CBIR application, QBIC [14] is one of the earliest commercial CBIR systems. Additional systems include imgSeek [15], visualSeek [16], SIMPLIcity [9], and Cortina [17] etc. Despite various approaches in CBIR, low level features such as color and texture do not strongly correlated to human semantic judgment. This often led to a semantic gap between user and machine [18, 19], results in low retrieval rate in current CBIR system.

ISBN: 978-960-6766-66-4

ISSN: 1790-5117

In this paper, we present an efficient rectangular segmentation framework for content-based image retrieval.

2 Image Feature Extraction

The common ground of CBIR system is to extract a feature to represent the image, since the image itself is too large to compare. Also using image feature can be more concentrate on useful information of image. After feature extraction, the extracted features are stored in database for future user querying.

In common image segmentation method, image segmentation based on clustering tends to produce scatter objects when segmenting roughly textured image [9]. Normalized cuts is good but computationally complex, thus not very suitable in CBIR. Moreover, due to the semantic gap problem in CBIR area, a very accurate segmentation usually not improves the retrieval rate a lot, but requires huge time complexity. Therefore, the aim of image rectangular segmentation is to provide an approximate segmentation of image objects while preserves the objects texture.

The rectangular segmentation first searches all possible vertical and horizontal cuts in order to find the best cut. Then decide whether the partition should proceed. Once the image is partitioned into two regions, then recursively partition it until every segmented region is similar to pure texture. Afterwards, a clustering method is applied on segmented regions in order to create the objects. The details of rectangular segmentation algorithm are given below.

Algorithm 1: Image rectangular segmentation

1. Based on image color histogram matrix, for every vertical and horizontal cut compute the histogram distance between the mean histogram of two partitions. The larger the histogram distance means the more perceptually different between two partitions. In order to have a better result, a weight is added to the histogram distance. Find the cut having the largest weighted histogram distance as the best cut. That is, for any cut *i* of two partitions of blocks $P_i(h_1,\ldots,h_k)$ and $Q_i(h_{k+1},\ldots,h_K)$, the best cut is the cut having

$$\max\left(D\left(mean(P_i), mean(Q_i)\right)\cdot w_i\right) \quad (7)$$

   where w_i denotes the weight of cut *i*.

2. If the weighted histogram distance of the best cut is above a threshold, partition the image along the best cut, otherwise do not partition it. In our implementation this threshold is 1.5.
3. For each newly segmented region, executes step 1 recursively. Until no any new segmented regions created.
4. Calculate the color histogram of all segmented regions by averaging the color histograms of all blocks in that region.
5. Suppose all regions belong to one cluster. Calculate the within-cluster sum of color histogram distance. That is, for a cluster with regions $C(R_1,\ldots,R_M)$, the within-cluster sum is

$$sumd(C) = \sum_{m=1}^{M} D(R_m, cent_C) \quad (8)$$

   where $cent_C$ is the mean of histogram in cluster *C*. If the within-cluster sum is below a threshold, all regions form one object. This threshold is 1.5 in our experimentation.

6. If step 5 does not form an object, begin k-means clustering algorithm with k=2. The total within-cluster sum is given by

$$tsumd(k) = \sum_{n=1}^{k} sumd(C_n) \quad (9)$$

   The algorithm gradually increase k until the total within-cluster sum $tsumd(k)$ is below the threshold, or $tsumd(k) - tsumd(k-1)$ is below the threshold. This is to make sure that k-means clustering is converged. For a fast clustering, the colors of dominant k color bins of image are used as the initial cluster center. After the regions are clustered, each cluster forms an object.

A weight is introduced in (7) which is intended to improve the segmentation performance. The definition of w_i is

$$w_i = 1 + w_i^A + w_i^S \tag{10}$$

where weight w_i^A encourage the algorithm segment large regions than small one, while weight w_i^S encourage segment in the image central part than outer part. w_i^A is calculated by

$$w_i^A = \min(area(P_i), area(Q_i)) \cdot f_A \tag{11}$$

where P_i and Q_i is two partitions of cut *i*, *area*(.) denotes the area percentage of the partition over the whole image, f_A is a parameter which is 40 according to our experiment. As shown in (11), w_i^A is proportional to the area of the smaller partition, which encourages the segmentation of large regions.

In (10) w_i^S is defined as

$$w_i^S = S(dc(x_i, y_i)) \cdot f_S \tag{12}$$

where x_i, y_i is the coordinate of the cut line center in *i*th cut, *dc*(.) denotes the normalized distance to image center of the point, *S*(.) is a significance function describe the significance of that position, f_S is a parameter of 2 in our system. The definition of *dc*(.) is

$$dc(x,y) = \sqrt{\left(\frac{x}{width/2} - 1\right)^2 + \left(\frac{y}{height/2} - 1\right)^2} \Big/ \sqrt{2} \tag{13}$$

where *width* and *height* is the size of image.

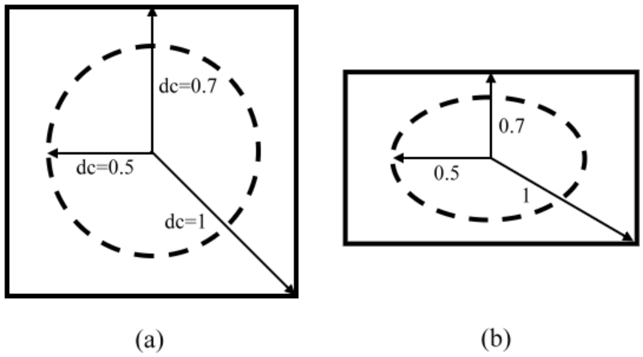


Fig. 1. Normalized distance to center and interest region. (a) In square image. (b) In rectangular image.

Figure 1(a) shows what normalized distance to center *dc* means. It is a measure of closeness from any points to image center. The range of *dc* is from 0 to 1. We define the region *dc* < 0.5 as interest region, because this region is usually the most attracted to human's attention. Since *dc* is normalized by image width and height, the interest region in a rectangular image is an ellipse as in figure 1(b).

The significance function *S*(.) in (12) is defined as

$$S(dc) = \left(\sqrt[3]{1-2dc} + 1\right)/2 \tag{14}$$

This function reflects the significance of different position in image. Figure 2 shows the graph of significance function. Where a point is inside the interest region, its significance is much higher than a point outside the interest region.

ISBN: 978-960-6766-66-4

ISSN: 1790-5117

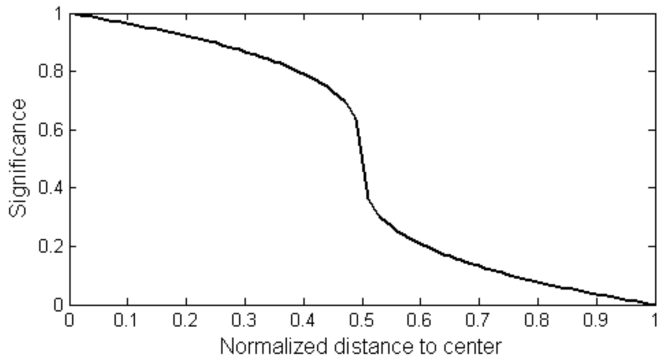
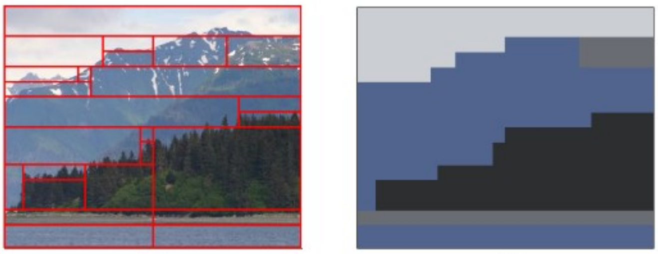


Fig. 2. Graph of significance function.

With the weight w_i^S in (12), a cut near the image center has a higher weight, which increases the number of partitions in image central part and reduces the number of partitions in image background. An example segmentation result is shown in figure 3.

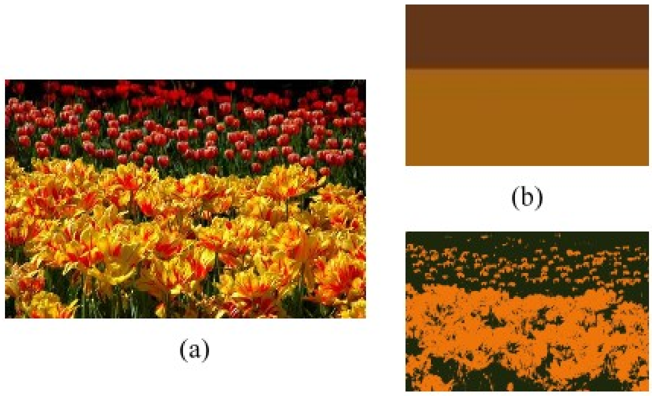


(a) (b)

Fig. 3. Image rectangular approximate segmentation. (a) Segmented regions. (b) 4 objects segmented.

3 Experimental Results

The segmentation is very fast comparing to high resolution segmentation. Moreover, the textures in objects are preserved in rectangular segmentation. Figure 4 is a comparison between rectangular segmentation and segmentation based on clustering. Rectangular segmentation in figure 4(b) successfully partition two kinds of flower beds. On the other hand, with segmentation by clustering the texture of two flower beds is lost in figure 4(c).



(a) (b) (c)

Fig. 4. Comparison between rectangular segmentation and clustering. (a) Image to be segmented. (b) Rectangular segmentation with 2 objects. (c) Segmentation by k-means clustering with k=2.

To evaluate the speed of various image segmentation algorithm, the same 100 images is tested with rectangular segmentation, k-means clustering segmentation, and normalized cuts segmentation in the same resolution. Since the number of segmented objects has to be chosen before applying k-means clustering and normalized cuts, we gradually increase the number of objects from 2 to 5, and record the total time used. The results of time efficiency of various segmentation algorithms are shown in Table I. Rectangular segmentation is almost as efficient as k-means clustering. On the other hand, Normalized cuts is much more slower than rectangular segmentation.

Table I. Time efficiency of various image segmentation algorithms

	Rectangular segmentation	K-means clustering	Normalized cuts
Total time used on 100 images	46.48 seconds	43.52 seconds	525.2 seconds

To evaluate the performance of the proposed framework, a 2,360 images dataset of COREL collection is used[1]. Each image is manually annotated, which consist of 1 to 5 tags include flowers, trees, buildings, animals, natural scene, etc. For any query, if the retrieved image has any tag match the query image, it is considered as a relevant result. Figure 5 shows an example of image retrieval, where the upper left image is the querying image.

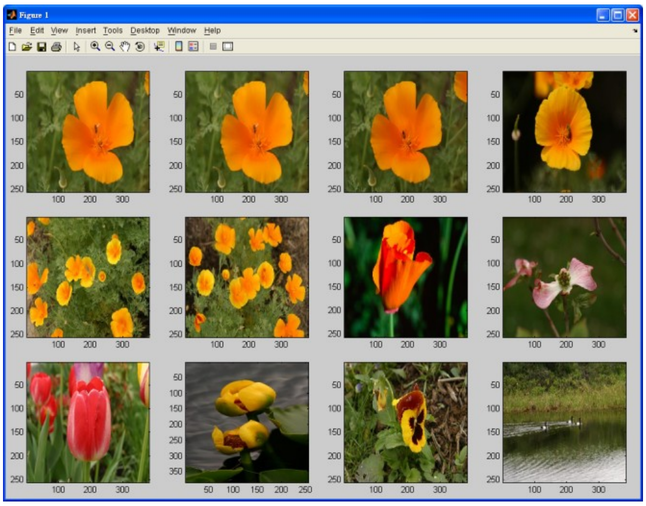


Fig. 5. An example of image retrieval

We have tested every image in the dataset, with Euclidean distance and quadratic distance respectively. For other CBIR system, since we only access to imgSeek, the results of imgSeek with the same dataset is included for comparison.

Table II. Precision on different scope

	Top 10	Top 20	Top 30
Quadratic distance	0.7377	0.6545	0.6081
Euclidean distance	0.6683	0.5857	0.5425
imgSeek	0.6379	0.5461	0.5019

[1] The dataset can be downloaded at http://www.stat.psu.edu/~jiali

Precision is the ratio of the number of retrieved relevant images to the number of total retrieved images. The average precision of our experiment is shown in table II and figure 6. Here we can see that the quadratic distance has the best result, but the computational complexity is also very high. Therefore, with the fast retrieval requirement, using Euclidean distance is more practical.

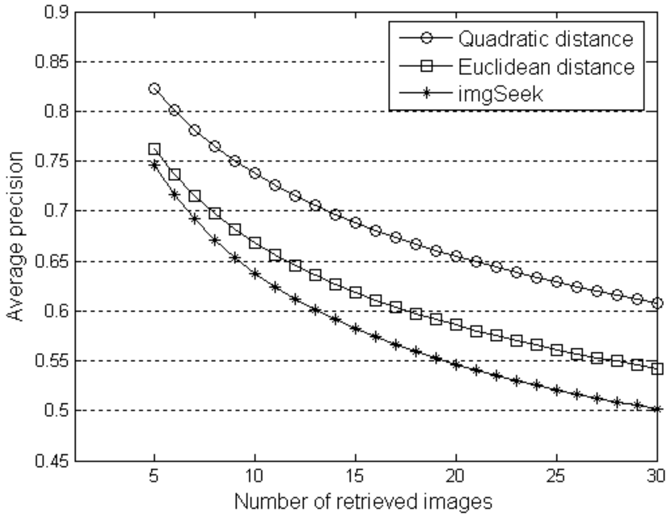


Fig. 6. Comparison between different methods.

4 Conclusion

In order to have fast image segmentation for CBIR, we have developed the image rectangular segmentation, which extracts objects in image. To improve the segmentation and retrieval quality, we have developed a methodology to enhance the importance of image central part, which is closer to human judgment. We have also developed an image similarity evaluation method between images with multi-objects. Experimental results show that the proposed method is more efficient and achieves higher precision on image retrieval of a large dataset.

References

[1] Swain, M.J. and D.H. Ballard, Color Indexing. International Journal of Computer Vision, 1991. 7(1).

[2] Stollnitz, E.J., T.D. DeRose, and D.H. Salesin, Wavelets for computer graphics: a primer.1. Computer Graphics and Applications, IEEE, 1995. 15(3).

[3] Stollnitz, E.J., T.D. DeRose, and D.H. Salesin, Wavelets for computer graphics: a primer. 2. Computer Graphics and Applications, IEEE, 1995. 15(4).

[4] Chang, T. and C.-C.J. Kuo, Texture Analysis and Classification with Tree-Structured Wavelet Transform. IEEE Transactions on image processing, 1993. 2(4).

[5] Unser, M., Texture Classification and Segmentation Using Wavelet Frames. IEEE Transactions on image processing, 1995. 4(11).

[6] Rahman, M.M., B.C. Desai, and P. Bhattacharya. A Feature Level Fusion in Similarity Matching to Content-Based Image Retrieval. in Information Fusion, 2006. ICIF '06. 9th International Conference. 2006.

[7] Lei, Z., L. Fuzong, and Z. Bo, A CBIR method based on color-spatial feature, in TENCON 99. Proceedings of the IEEE Region 10 Conference. 1999. p. 166-169.

[8] Pappas, T.N., An Adaptive Clustering Algorithm for Image Segmentation. IEEE Transactions on signal processing, 1992. 40(4).

[9] Wang, J.Z., J. Li, and G. Wiederhold, SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries. IEEE Transactions on pattern analysis and machine intelligence, 2001. 23(9).

[10] Shi, J. and J. Malik, Normalized Cuts and Image Segmentation. IEEE Transactions on pattern analysis and machine intelligence, 2000. 22(8).

[11] Ma, W.-Y. and B.S. Manjunath, EdgeFlow: A Technique for Boundary Detection and Image Segmentation. IEEE TRANSACTIONS ON IMAGE PROCESSING, 2000. 9(8).

[12] Deng, Y. and B.S. Manjunath, Unsupervised Segmentation of Color-Texture Regions in Images and Video. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2001. 23(8).

[13] Pratikakis, I., et al., Unsupervised watershed-driven region-based image retrieval. Digital Object Identifier, 2006. 135(3): p. 313-322.

[14] Flickner, M., et al., Query by Image and Video Content: The QBIC System. IEEE Computer, 1995. 28(9).

[15] Jacobs, C.E., A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. 1995.

[16] Smith, J.R. and S.-F. Chang. VisualSEEk: A Fully Automated Content-Based Image Query System. in ACM Multimedia. 1996.

[17] Quack, T., et al. Cortina: a system for large-scale, content-based web image retrieval in Proceedings of the 12th annual ACM international conference on Multimedia. 2004.

[18] Vasconcelos, N., From Pixels to Semantic Spaces: Advances in Content-Based Image Retrieval. Computer, 2007. 40(7).

[19] Datta, R., et al., Image Retrieval: Ideas, Influences, and Trends of the New Age, in ACM Transactions on Computing Surveys. 2007.

[20] Vasconcelos, N. and A. Lippman. A Unifying View of Image Similarity. in Proceedings 15th International Conference on Pattern Recognition. 2000.

ISBN: 978-960-6766-66-4

ISSN: 1790-5117