Image Retrieval by Content Using Segmentation Approach

Bhogeswar Borah and Dhruba K. Bhattacharyya

Tezpur University, Tezpur-784028, India
(bgb, dkb)@tezu.ernet.in

Abstract. This paper presents an efficient image retrieval technique based on content using segmentation approach and by considering global distribution of color. To cope with significant appearance changes, the method uses a global size and shape histogram to represent the image regions obtained after segmenting the image based on color similarity. The indexing technique can be found to be significant in comparison to its other counterparts, such as moment based method [12], due to its transformation invariance and effective retrieval performance over several application domains.

1 Introduction

Content-Based Image Retrieval (CBIR) is the retrieval of relevant images from an image database based on automatically derived features. Unlike keyword based databases content driven image databases are mostly queried by providing example images. It is the task of the search algorithm to find result images, which are most similar to the query image by computing the similarity, based upon extracted features. Primitive features such as color, texture, shape or a combination of these features are very useful for image description and retrieval. The need for efficient content-based image retrieval has increased tremendously in many application areas such as bio-medicine, crime prevention, the military, commerce, culture, education, entertainment, advertising, journalism, design, web-image classification and searching, etc .

As color plays an important role in image composition, many color indexing techniques have been developed including the original work by Swain and Ballard [1]. Although global color histograms and moments have been proven to be very useful for image indexing [2], they do not take color based spatial information into account resulting in many false hits as the image database size increases. The retrieval becomes more efficient when spatial arrangement of color is taken into account or when one or more low-level features, such as texture or shape, is added to the system. Several schemes [3,4,5,6,7] including spatial information have been proposed. An image can be partitioned into regions for the representation of spatial color distribution. A region-based retrieval system [8,9,10] applies image segmentation to decompose an image into regions, which correspond to objects if the decomposition is ideal. Color indexing can be obtained

using global distribution or local distribution of color. Global color indexing enables whole image matching while regional indexing enables matching localized regions within images. However, if the image partition depends on the pixel positions, the approach may not tolerate significant appearance changes [11].

A compact and effective way of capturing the spectral content (i.e., chrominance) of an image is presented in [12]. This method is based on the *CIE xyY* color space. Each pixel in a given image yields a pair of (x, y) chromaticity values, thus forming a set of chromaticities for the entire image. This set of (x, y) pairs is called the *chromaticity set* of the specific image, while the corresponding xy Euclidean space is called the *chromaticity space*. Given an image, its chromaticities are characterized by two attributes, namely, their two-dimensional shape (*chromaticity set*) and their two-dimensional distribution (*histogram*). A set of moments are subsequently computed to capture the essence of these two attributes. These moments are called *chromaticity moments* and form a feature set for the corresponding image.

In this paper, we focus on region based retrieval of images considering global distribution of color. To cope with significant appearance changes, our method uses a global size and shape histogram to represent the image regions obtained after segmenting the image based on color similarity. The indexing technique is significant due to the following features: (1) the index has the property of translation, rotation and scaling invariance; it also has been found to be robust to significant appearance changes; (2) it is compact from the storage point of view; (3) it is comparatively effective in terms of retrieval performance and (4) it provides color indirect matching, i.e., similar objects with different colors can be matched;

Rest of the paper is organized as follows: In Section 2 the proposed image index is described. Section 3 presents experimental results. Finally, we conclude in Section 4.

2 Proposed Spatial Color Descriptor

The image is segmented into different color regions. The segment sizes are represented in an unnormalized size histogram. Similarly, for each segment a shape measure is computed and corresponding shape histogram is prepared. Both the histograms are concatenated into a single histogram. We define a similarity measure to compute the similarity of two images in the histogram space.

2.1 Image Segmentation

The objective of segmentation is to partition an image into non overlapped regions based on similarity of color. Let the color of a pixel be represented by the *RGB* vector $\mathbf{a}$. Let $\mathbf{z}$ denote a neighboring pixel of the pixel $\mathbf{a}$. We say that $\mathbf{z}$ is similar to $\mathbf{a}$ if the distance between them is less than a specified threshold D_0 . The Euclidean distance between $\mathbf{z}$ and $\mathbf{a}$ is given by [13] :

$$D(\mathbf{z}, \mathbf{a}) = [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{1/2}$$

where, the subscripts R , G and B denote the RGB components of vectors $\mathbf{a}$ and $\mathbf{z}$. If $D(\mathbf{z}, \mathbf{a}) \leq D_0$ pixels $\mathbf{z}$ and $\mathbf{a}$ belong to the same color segment. Beginning with an arbitrary pixel at location (r, c) a region can be grown by examining the nearest neighbors ($8 - neighbours$) of (r, c) one by one and accepting a neighboring pixel to belong to the same region as (r, c) if the distance between them is less than D_0 . Once a new pixel is accepted as member of the current region the nearest neighbor of this new pixel is examined. This process goes on recursively until no more pixel is accepted. All the pixels of the current region are marked with a unique label. Then another unlabeled pixel is picked up and the same procedure is repeated. Scanning the image from left to right, top to bottom we go on labeling the regions until every pixel is assigned to some region or the other.

2.2 Color Segment Size and Shape Histograms

Let the total number of color segments in an image obtained by the segmentation process be t . The size of a segment, denoted by *segment_size* is the number of pixels present in the segment. The size histogram S is a list of $n + 1$ integers:

$$S = \{s_0, s_1, s_2, \dots, s_n | s_i \geq 0, \sum_{i=0}^n s_i = t\}$$

where, s_i is the number of segments such that $2^i \leq \text{segment_size} < 2^{i+1}$.

The size histogram S is rotation invariant. With minor adjustment it becomes scale invariant. A nice property of the histogram is that for every horizontal or vertical scaling of the image by a factor of 2, the histogram is shifted by one location to the right. If the image is scaled up both horizontally and vertically by a factor of 2, the histogram is shifted to the right by 2 locations. In other words, if the overall image size is increased by 2^i ($i = 1, 2, 3 \dots$) times then the histogram is shifted to the right i locations. Whether an image is scaled up one and if so, the number of bins the histogram should be shifted to the left to make it scale invariant can be obtained during segmentation.

A shape measure of a segment is a fraction in the closed interval $[0, 1]$. It is defined as

$$\text{Shape_measure} = \frac{\text{segment_size}}{\text{bounding_rectangle_size}}$$

The *bounding_rectangle_size* is the area of the minimum rectangle whose sides are parallel to the axes and that completely encloses the segment. The shape histogram H is a list of $m + 1$ integers :

$$H = \{h_0, h_1, h_2, \dots, h_m | h_i \geq 0, \sum_{i=0}^m h_i = t\}$$

where, h_i is the number of segments such that $\frac{i}{m} \leq \text{Shape_measure} < \frac{i+1}{m}$.

The shape histogram is invariant to scaling of the image. It is also invariant to rotation of the image by 90 degrees. In fact the size histogram and shape histogram can be concatenated into a single histogram, where the first n bins represent the size histogram and the last m bins represent the shape histogram.

2.3 Similarity Function for Image Comparison

Let, the combined histogram of size and shape be represented by Q for query image and D for any database image. Then the similarity measure between Q and D is a positive fraction ≤ 1 . It is defined as

$$Similarity(Q, D) = \frac{1}{n+1} \sum_{i=0}^h \frac{|q_i - d_i|}{\max(q_i, d_i) + w_i}$$

where, h represents the total number of bins in the combined histogram and w_i represents a non-zero positive weight assigned to bin i . Typical value for w_i is 1.

3 Experimental Results

We have implemented our approach in JAVA on an Intel workstation (1 Ghz CPU with 256 MB SDRAM). To test the technique we used an image collection of 4500 GIF, JPG and PNG images of size varying from 100x50 to 800x600 collected from various sources including *ETH* – 80 database [14]. Images are segmented with distance threshold $D_0 = 4$. The combined histogram consists of 22 bins. First, the size histogram is computed and stored in bin numbers 0 to 10. Occurrence of segments of size bigger than $2^{10} = 1024$ pixels are very few. If such segments are found they are also counted in bin number 10. The shape histogram is stored starting from bin number 11 up to bin number 21. In this study, we use the efficiency of retrieval for the formal evaluation of retrieval effectiveness [3]. The efficiency of retrieval from a given output list of size T is defined as

$$E_T = \begin{cases} \frac{n}{N}, & \text{if } N \leq T; \\ \frac{n}{T}, & \text{otherwise.} \end{cases}$$

where n is the number of similar images retrieved and N is the total number of similar images in the database. We note that if $N \leq T$, then E_T becomes the traditional recall value of information retrieval, while if $N > T$, then E_T becomes the traditional precision value. Performance comparison of content-based image retrieval systems is a non-trivial task since it is very difficult to determine the relevant sets. Even if we do not exactly know the relevant sets, the observer's feel for relevant images in the retrieved set is what can be used as a measure of

Table 1. Comparison Results in terms of E_T

Size-of-list	Moment_based method [12]	Proposed method
5	0.58	0.70
10	0.43	0.52
15	0.40	0.45
20	0.35	0.43

precision. In each of the experiments performed, we have calculated the metric of precision for 50 randomly selected images and taken the average precision value. Precision is calculated for the first 5, 10, 15 and 20 nearest neighbor images of the result set. Also, for the same set of queries we have computed the precision values with the chromaticity moments based method proposed in [12]. The results are presented in Table 1.

Fig. 1. Examples of Query results

Two query results are presented in Fig. 1. In all the results, the image at the top left corner is the query image. The retrieved list is ranked in order of decreasing similarity from left to right, top to bottom.

4 Conclusions

This paper has presented an efficient spatial color indexing technique designed using the segmentation approach. The technique has the property of translation, rotation and scaling invariance. The retrieval is tolerant to significant appearance changes and provides the facility of color indirect matching. The retrieval performance of the technique also has been established to be superior in comparison to its other counterparts, such as [12].

References

1. M.J. Swain and D.H. Ballard, "Color Indexing", International Journal of Computer Vision, Volume 7, Number 1, 1991, pp. 11-32.
2. M. Stricker and M. Orengo, "Similarity of Color Images", Proc. of SPIE Storage and Retrieval for Image and Video Databases, Volume 2420, February 1995, pp. 381-392.
3. Y. Tao, W.I. Grosky, "Spatial Color Indexing: A Novel Approach for Content-Based Image Retrieval", IEEE Intl. Conf. on Multimedia and Systems, Volume 1, June 1999, pp. 530-535.
4. W. Hsu, T.S. Chua and H.K. Pung, "An Integrated Color-Spatial Approach to Content-based Image Retrieval", Proc. of ACM Multimedia, San Francisco, California, November 1995, pp. 305-313.
5. H.Y. Lee, H.K. Lee and Y.H. Ha, "Spatial Color Descriptor for Image Retrieval and Video Segmentation", IEEE Transaction on Multimedia, Volume 5, Number 3, September, 2003.
6. G. Pass and R. Zabih, "Histogram Refinement for Content-Based Image Retrieval", IEEE Workshop on Applications of Computer Vision, 1996, pp. 96-102.
7. I.K. Park, I.D. Yun and S.W. Lee, "Color image retrieval using hybrid graph representation", Image and Vision Computing, Volume 17, Number 7, May 1999, pp. 465-474.
8. I.K. Sethi, I Coman, et al., "Color-WISE: A system for Image Similarity Retrieval Using Color", Proc. of SPIE Storage and Retrieval for Image and Video Databases, Volume 3312, February 1998, pp. 140-149.
9. S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color and texture-Based Image Segmentation Using EM and Its Application to Content-Based Image Retrieval", Proc. of the Intl. Conf. on Computer Vision (ICCV'98), 1998.
10. B.M. Mehtre, M.S. Kankanhalli, and W.F. Lee, "Content-Based Image Retrieval Using A Composite Color-Shape Approach", Information Processing and Management, Volume 34, Number 1, January 1998, pp. 109-120.
11. J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih, "Image indexing Using Color Correlograms", Proc. of the 16th IEEE Conf. on Computer Vision and Pattern Recognition, June 1997, pp. 762-768.
12. G. Paschos, I. Radev, and N. Prabakar, "Image Content-Based Retrieval Using Chromaticity Moments", IEEE Ttransaction on Knowledge and Data Engineering", Volume 15. Number 5, Sept./Oct. 2003, pp. 1069-1072.
13. R.C. Gonzalez and R.E. Woods, "Digital Image Processing", 2nd ed., Pearson Education, 2003
14. <http://www.vision.ethz.ch/projects/categorization/download.html>