A CONTEXT-BASED APPROACH FOR COLOR IMAGE RETRIEVAL

JAU-LING SHIH* and LING-HWEI CHEN†

*Department of Computer and Information Science,
National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu, 30050 Taiwan, R.O.C.*
*sjl@chu.edu.tw
†lhchen@cc.nctu.edu.tw

In this paper, a color image retrieval method based on the primitives of images will be proposed. First, the context of each pixel in an image will be defined. Then, the contexts in the image are clustered into several classes based on the algorithm of fast noniterative clustering. The mean of the context in the same class is considered as a primitive of the image. The primitives are used as feature vectors. Since the numbers of primitives between images are different, a specially designed similarity measure is then proposed to do color image retrieval. To better adapt to the preferences of users, a relevance feedback algorithm is provided to automatically determine the weight of each primitive according to the user's response. To demonstrate the effectiveness of the proposed system, several test databases from Corel are used to compare the performances of the proposed system with other methods. The experimental results show that the proposed system is superior to others.

Keywords: Content-based image retrieval; primitives; clustering; relevance feedback algorithm.

1. Introduction

In recent years, with the construction of digital libraries, the management of the large multimedia databases such as color photographs, trademarks, stamps and paintings has become an important issue. Thus, the demand for an automatic and user-friendly image retrieval system based on image content has become urgent.

QBIC[5,11] proposed by IBM is such a system, in which users can choose one kind of feature: color histogram, color layout, texture and shape to do image retrieval. The color histogram is invariant to rotation, translation and scaling. However, two images with similar color histograms may look very different, due to which the locations of the color regions in an image are ignored. This can be solved by using a color layout. Color layout is concerned more on the rough color positions, but the detailed context of an image is not considered. Texture features derived from the second-order statistics are suitable only for images full of textures, such as cloud, sand, grass and cloth, etc. However, in the real world, only a few images have texture spreading over the whole image. Thus, for a natural image, if users only

take the texture features to do retrieval, a good result will not be expected. Shape features consist of shape area, circularity, eccentricity, major axis orientation and invariant moments. For some special type of images such as trademarks and Chinese antiques, the shape features are very important. However, there does not exist a characteristic shape in many natural images. In summary, if users take only one kind of feature, the retrieval result will not be satisfied. Although, QBIC provides four kinds of features, users can choose only one kind of features to do retrieval. Thus, how to combine various features to get better results is a significant research topic.

Virage system[2,15] provides four features: color histogram, color layout, texture and structure (object boundary information). Users can arbitrarily combine the above four features to do similar image retrieval. In addition, the weight of each feature can be adjusted by users to get better retrieval result. For example, for some images, such as cars and balloons, with special shapes, the feature of object boundary should be emphasized. However, it is difficult for a user to properly select the weights for the above four features. Thus, how to find a good method that can automatically adjust the weights is important.

MARS system[9,10] uses a relevant feedback algorithm[12] to automatically adjust the weights among color, texture,[17] and shape features interactively with users. Moreover, the weight of each component among a feature vector can also be adjusted. In the above-mentioned systems, the spatial relationship in each image is not taken into consideration during the process of retrieval. The color layout can roughly describe the color positions in an image, but not fully specify the relationship among the color regions.

The WebSEEk system[13,16] considers this type of relationship. Each image is first mapped onto a color set with 166 colors in the HSV color space and then segmented into several regions. A 2D string is constructed to represent the spatial relationships among these regions and used for querying a similar image. However, similar images may have different segmentation results, which will produce very different 2D strings and thus affect the retrieval results.

For reducing the influence of improper segmentation, Blobworld system,[3] which is developed under the digital library project of University of California-Berkeley, transforms the raw image data to a small set of image regions, called blobs. Each blob, which roughly corresponds to an object or a part of an object, is coherent in color and texture. For images, such as the sunset, zebra, or sky, proper blobs can be easily extracted. However, it is hard to guarantee that the proper representative blobs could be found in each color image.

All the above-mentioned color features only contain the color information of each pixel in an image or the global spatial relation among regions, the local relationship among neighboring pixels is not involved. Huang *et al.*[7] proposed a kind of feature, called color correlograms, and used it for image retrieval. A color correlogram expresses how the spatial correlation of pairs of colors changes with distance. In order to reduce the size of the feature set, each image is quantized into 64 colors in the

RGB color space. However, if an image contains various colors, such as stain-glass images, 64 colors are not enough to represent the color information.

Maron and Ratan[8] also proposed a kind of feature, called instance, to consider the local relationship among neighboring pixels for image classification. An instance consists of the color information of a subimage, not just a pixel. Some representative instances are calculated for each image class (for example: waterfall, sunset, or mountain) by a semi-automatically learning algorithm. Each image in the testing database can be classified correctly by using the representative instances. However, it is not appropriate to apply the algorithm to do similar image retrieval, since there are two problems. The first problem is that the representative instances for each image class must be found in advance. Therefore, what kinds of classes existing in an image database and the training images for each class must be manually decided. The second and most serious problem is that we can merely decide what class an image is, but cannot extract those similar images.

In this paper, a color image retrieval method based on the primitives of images will be proposed. In the proposed method, for each image, some primitives are first extracted to represent the most important components in the image. Using the primitives as feature vectors, a similarity measure is proposed to perform color image retrieval. Besides, a relevance feedback algorithm is given to automatically adjust the weight of each primitive according to the user's response. In order to show the effectiveness of the proposed method, some comparisons among the proposed method and others are also provided. The rest of the paper is organized as follows. In Sec. 2, we will introduce the extracting method for the proper primitives of each image. In Sec. 3, the similarity measure and the relevance feedback algorithm are described. The experimental results are described in Sec. 4. Finally, conclusions will be given in Sec. 5.

2. Primitive Extraction

In this section, we will describe a kind of color feature, called primitive, which will be used for color image retrieval. Before introducing the primitive, the context will be first defined since the primitives will be extracted based on contexts.

2.1. *Contexts*

For a color image, the RGB model is the well-known color model. However, it is not appropriate for color image retrieval. The major reason is that the three components (red, green, and blue) have very high dependency and the luminance of each pixel is embedded in these three components simultaneously. Thus, two colors with small distance in the RGB space may look totally different. Contrarily, the YIQ model is more related to the human perception, since it can isolate luminance and chrominance. Hence, in this paper, we take the YIQ model as the color model. To reduce the influence of noise, each image is first down-sampled. Let

$f(x, y) = (Y_{x,y}, I_{x,y}, Q_{x,y})$ denote the color value of the down-sampled image located at (x, y). The context of a point a, located at (x, y), is defined by a 15-tuple vector:

$$C_a[c_{a,1}, c_{a,2}, \ldots, c_{a,15}]$$

where

$$c_{a,1} = \alpha_1 Y_{x,y}, \ c_{a,2} = \alpha_2 I_{x,y}, \ c_{a,3} = \alpha_3 Q_{x,y},$$

$$c_{a,4} = \alpha_1 (Y_{x,y} - Y_{x,y-1}), \ c_{1,5} = \alpha_2 (I_{x,y} - I_{x,y-1}), \ c_{a,6} = \alpha_3 (Q_{x,y} - Q_{x,y-1}),$$

$$c_{a,7} = \alpha_1 (Y_{x,y} - Y_{x+1,y}), \ c_{a,8} = \alpha_2 (I_{x,y} - I_{x+1,y}), \ c_{a,9} = \alpha_3 (Q_{x,y} - Q_{x+1,y}),$$

$$c_{a,10} = \alpha_1 (Y_{x,y} - Y_{x,y+1}), \ c_{a,11} = \alpha_2 (I_{x,y} - I_{x,y+1}), \ c_{a,12} = \alpha_3 (Q_{x,y} - Q_{x,y+1}),$$

$$c_{a,13} = \alpha_1 (Y_{x,y} - Y_{x-1,y}), \ c_{a,14} = \alpha_2 (I_{x,y} - I_{x-1,y}), \ c_{a,15} = \alpha_3 (Q_{x,y} - Q_{x-1,y}),$$

and $\alpha_1, \alpha_2, \alpha_3$ are the weights of the color components, respectively. $c_{a,4}$ to $c_{a,15}$ denote the weighted color differences between a and its four neighbors, as shown in Fig. 1.

From the above definition, we can see that two points with the same color values and their corresponding neighbors with different color values will have different contexts. For example, the two points a and b shown in Fig. 2 will be classified as

	$(x\text{-}1, y)$	
$(x, y\text{-}1)$	(x, y)	$(x, y+1)$
	$(x+1, y)$	

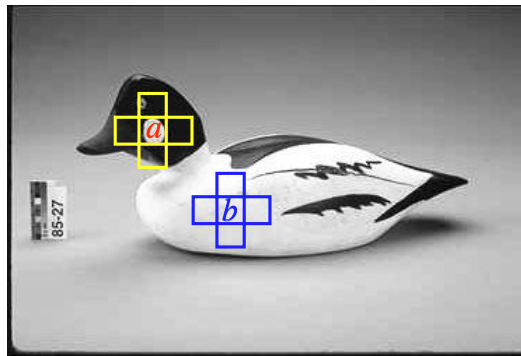Fig. 1.   The four neighbors of the pixel (x, y).

Fig. 2.   An example of two points a and b with the same color values and different contexts.

the same group if only color histogram is used. However, they will be regarded as different classes if the context is used as a feature. From the above definition, for a down-sampled image, a clustering method[1] is applied to find several representative contexts, called primitives. Then these primitives are used to represent the image.

2.2. *Primitives*

In this section, a progressive constructive clustering algorithm[6] is used to find the primitives of an image. The algorithm will classify all contexts in an image into several clusters. The central vector of each cluster is regarded as a primitive of the image.

Before describing the algorithm, several definitions and measures are stated. The central context, P_k, of the kth context cluster is defined by

$$P_k = [p_{k,1}, p_{k,2}, \ldots, p_{k,15}] = \frac{\sum_{j=1}^{n_k} C_{a_j^k}}{n_k}$$

$$= \left[\frac{\sum_{j=1}^{n_k} c_{a_{j,1}^k}}{n_k}, \frac{\sum_{j=1}^{n_k} c_{a_{j,2}^k}}{n_k}, \ldots, \frac{\sum_{j=1}^{n_k} c_{a_{j,15}^k}}{n_k} \right], \tag{1}$$

where $C_{a_j^k}$, $j = 1, 2, \ldots, n_k$, belongs to the kth context cluster. In addition, the Euclidean distance between C_a and P_k is defined as follows:

$$d_{a,k} = \sqrt{\sum_{i=1}^{15} (c_{a,i} - p_{k,i})^2} \,.$$

According to the above definitions, the progressive constructive clustering algorithm is described as follows.

Step 1. For an image, randomly choose a point r in the image and take its context C_r as the central context, P_1, of Cluster 1.

Step 2. Take one context C_a, which is not processed, and find its nearest central vector, $P_{k'}$, from the existing clusters. If $d_{a,k'}$ is larger than a predefined distance threshold, T_d, go to Step 4.

Step 3. The context C_a is put in the k'th context cluster and $P_{k'}$ is updated according to Eq. (1). Go to Step 5.

Step 4. Create a new cluster and take C_a as the central context of the new cluster.

Step 5. If all contexts have been processed, exit; otherwise, go to Step 2.

The advantage of this algorithm is that the clustering process is fullfilled in one iteration. After all the contexts in an image have been classified, the central

contexts of all clusters are regarded as the primitives of the image. Note that during the construction of an image database, the primitives will also be attached to each image for retrieval purpose. Since the distance threshold, T_d, is fixed for all images, the number of primitives varies for different images. To treat this situation, a new method to evaluate the similarity between two images with different number of primitives will be proposed.

3. Color Image Retrieval

In this section, a similarity measure between two images with variant numbers of primitives is provided first. Each primitive represents one of the major components in an image and its importance will be expressed by the corresponding weight, which will be used in the similarity measure. Thus, a method to adjust the weights of all primitives in response to each user's subjective point of view will be proposed. Initially, each primitive is set the same weight. Then, a relevance feedback algorithm is provided to automatically adjust the weights of all primitives according to the retrieval result and the user's response.

3.1. *Similarity measure*

Before introducing the similarity measure, we will first describe several definitions. The kth primitive of a query image q is a 15-tuple vector: $P_k^q = [p_{k,1}^q, p_{k,2}^q, \ldots, p_{k,15}^q]$, where $k = 1, 2, \ldots, m$, and m is the number of primitives in the query image. The λth primitive for a matching image s is defined by $P_\lambda^s = [p_{\lambda,1}^s, p_{\lambda,2}^s, \ldots, p_{\lambda,15}^s]$. The distance between P_k^q and P_λ^s is defined as follows:

$$D_{k,\lambda}^{q,s} = \sqrt{\sum_{i=1}^{15} (p_{k,i}^q - p_{\lambda,i}^s)^2} \,.$$

The minimum distance between P_k^q and all the primitives of s is defined by

$$D_k^{q,s} = \min_\lambda (D_{k,\lambda}^{q,s}) \,.$$

The distance between the query image q and the matching image s is defined by

$$D^{q,s} = \sum_{k=1}^{m} \omega_k \times n_k^q \times D_k^{q,s} \,,$$

where n_k^q is the number of contexts in the kth cluster and ω_k is the weight for the kth primitive. Initially, ω_k is set to 1. The similarity between q and s is defined by the inverse of $D^{q,s}$ and is evaluated as

$$\text{Sim}^{q,s} = \frac{1}{D^{q,s}} \,.$$

Note that the larger $\text{Sim}^{q,s}$ a matching image has, the more similar it is to the query one. Thus, we find several top similar images to the query one based on

$\text{Sim}^{q,s}$. Next, a relevance feedback algorithm is provided to automatically adjust the weight of each primitive according to the current retrieval result and the user's response.

3.2. *Relevance feedback algorithm*

In this subsection, a relevance feedback algorithm is provided to automatically adjust the weight of each primitive according to the user's response. First, the initial retrieval result is derived using a set of fixed weights. To meet the user's subjective point of view, a user chooses u relevant images, $R_1, R_2, \ldots, R_u$, and t nonrelevant images, $N_1, N_2, \ldots, N_t$, from the current retrieval result. These relevant images are more similar to the query one from the user's viewpoint and the nonrelevant images are more dissimilar. Then, a more important primitive in the query image should occur more frequently in the set of relevant images and less frequently in the set of nonrelevant images. Thus, according to this phenomenon, a representative vector for the kth primitive of the query image is redefined by M_k:

$$M_k = \frac{1}{u+1}\left(P_k^q + \sum_{i=1}^{u} P_{\hat{k}}^{R_i}\right), \quad k = 1, 2, \ldots, m,$$

where

$$\hat{k} = \arg\min_{\lambda}(D_{k,\lambda}^{q,R_i}).$$

Then, the standard deviations for the kth primitive from relevant and nonrelevant images are defined by

$$\sigma_k^{\text{rel}} = \sqrt{\frac{(P_k^q - M_k)(P_k^q - M_k)^T + \sum_{i=1}^{u}[(P_{\hat{k}}^{R_i} - M_k)(P_{\hat{k}}^{R_i} - M_k)^T}{u+1}},$$
$$k = 1, 2, \ldots, m,$$

and

$$\sigma_k^{\text{nonrel}} = \sqrt{\frac{\sum_{i=1}^{t}[(P_{\hat{k}}^{N_i} - M_k)(P_{\hat{k}}^{N_i} - M_k)^T]}{t}}, \quad k = 1, 2, \ldots, m.$$

From the above definitions, for each P_k, if σ_k^{rel} is small and σ_k^{nonrel} is large, P_k should be an important primitive and its corresponding weight, ω_k, should be increased. Therefore, ω_k can be adjusted as

$$\omega_k = \frac{\sigma^{\text{nonrel}}}{\sigma_k^{\text{rel}}}.$$

After adjusting all of the weights, the retrieval result for the second time is obtained. In general, the new retrieval result will be closer to what the user really wants. Furthermore, a user can interactively repeat the search process until a satisfactory retrieval result is obtained.

4. Experimental Results

To evaluate the performance of the proposed method, experiments have been conducted based on Corel photo library, which is often used by image retrieval research groups.[7,8,17] There are two major test databases, D1 and D2, selected from Corel in our experiments. D1 is a small and well-classified test database. Based on D1, we also implement other methods using the color histogram,[5] color moment,[14] color set,[13] or color correlograms[7] as features to compare their performances with ours. On the other hand, Zhou *et al.*[17] have provided their retrieval results of using water-filling and wavelet moments as features on a large test database which are also obtained from Corel. Thus, we select images from Corel to establish a large database D2 with the same size as that used in Zhou's. Using D2, we apply our method to compare the performances with water-filling and wavelet moments without implementing their methods. Finally, we also compare our performance with color histogram, color moment, color set and color correlograms on D2.

4.1. *Experimental results on small database D*1

The small database D1 has 1300 images. These images are classified into 13 classes, including the flower, stained glass, woman, sunset, sports car, sailboat, ancient architecture, dinosaur, duck, waterfall, painting, underwater world and gong fu. Each class contains 100 images. Figure 3 shows several example images for each

| 01-001 | 01-010 | 02-031 | 02-026 | 03-001 | 03-005 | 04-001 | 04-099 |

| 05-001 | 05-074 | 06-020 | 06-092 | 07-008 | 07-021 | 08-002 |

| 08-087 | 09-018 | 09-078 | 10-025 | 10-061 | 11-021 | 11-038 |

| 12-008 | 12-077 | 13-001 | 13-100 |

Fig. 3.   Several example images from the database D1.

class of D1. The first term in the file number denotes the class number and the second term the image number. For example, "01-001" stands for the first image in Class 1 (flower). The performance is measured by recall and precision.[4] Note that the recall, Re, is defined by the following equation:

$$\text{Re} = \frac{N}{T}$$

where N is the number of relevant images retrieved and T is the total number of relevant images. The precision, Pr, is defined as follows:

$$\text{Pr} = \frac{N}{K}$$

where N is the number of relevant images retrieved and K is the total number of retrieved images.
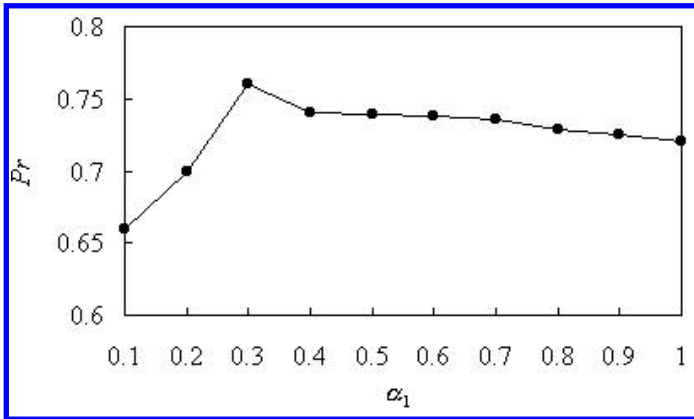
Fig. 4. The precision comparison on D1 using different weights of Y, α_1. The weights of I and Q are set to 1, $\alpha_2 = \alpha_3 = 1$.
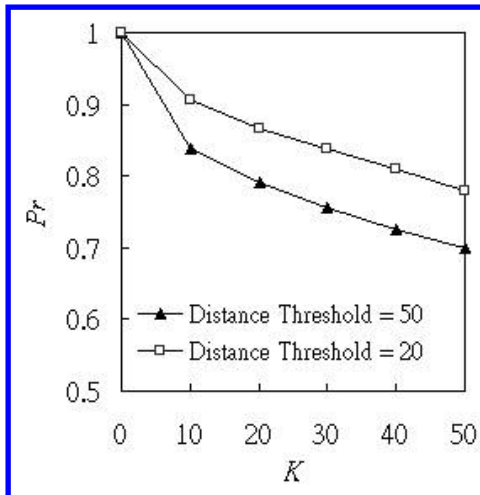

Fig. 5. The precision comparison in D1 using different distance thresholds, T_d.

In image retrieval, color components (I and Q) are more important than luminance component (Y). Hence, the weights of I and Q components must be larger than that of the Y component. To show the phenomenon, Fig. 4 compares the precision using a set of weight assignments to α_1, α_2 and α_3 which denote the weight of Y, I and Q, respectively. As Fig. 4 shows, if the influence of luminance is appropriately reduced, a better retrieval result can be expected. Hence, in our experiments α_1, α_2 and α_3 are set to 0.3, 1, and 1, respectively.

The number of primitives depends on the distance threshold, T_d. If T_d is larger, the average number of primitives in each image will be reduced and the retrieval speed will be fast. However, the contexts within each class will have higher variance, and thus the corresponding primitive would not represent a certain component well. This will slightly reduced the retrieval efficiency. In Fig. 5, the comparison of precision based on the same down-sampled image sizes, 50×50, and different distance thresholds is shown.

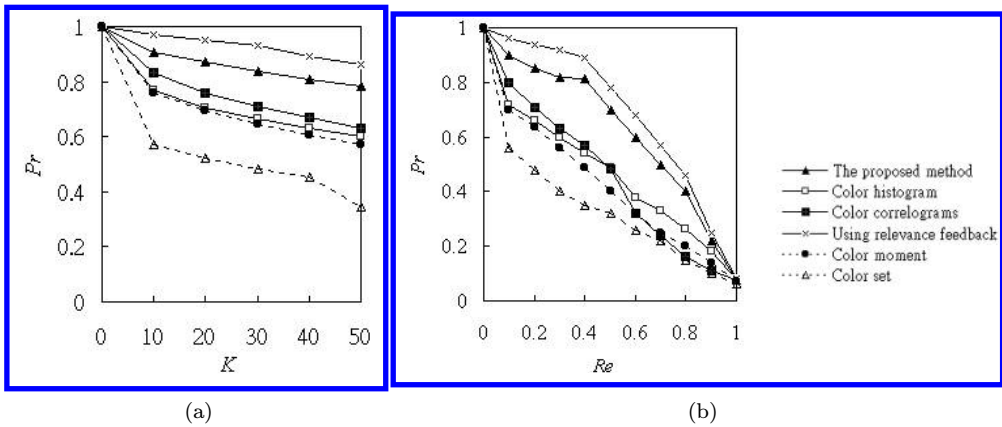(a)   (b)

Fig. 6.   The performance comparison among the proposed method with or without using the relevance feedback algorithm and other methods on D1. (a) The precision curves. (b) The precision versus recall curves ($T = 100$).
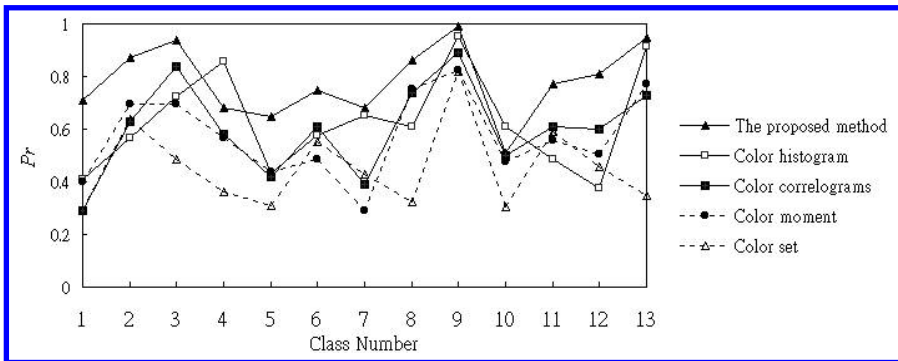

Fig. 7.   The precision comparison for each class of D1 among the proposed method and other methods ($K = 50$).

To show the performance of the proposed method, the retrieval results are compared with those retrieved by color histogram, color correlograms, color moment or color set on D1. As shown in Fig. 6, the proposed method is much better than other methods. In all of the above experiments, the relevance feedback algorithm is not applied. To make the retrieval results meet the user's need, a user chooses a set of relevant images he really wants and he does not concern with nonrelevant images from the retrieval result. Then, the proposed relevance feedback algorithm is applied to obtain the new retrieval result that in general will be closer to what the user really wants. The result is also shown in Fig. 6. The detailed comparison of performances for each class is shown in Fig. 7. Since color histogram, color set and color moment do not consider the color information of neighboring pixels and the color correlograms quantize the images into fixed number of colors. Thus, our method has much better performance on the classes with more complex color layout, such as the stain-glass, painting, and under-water images, as shown in Fig. 7.

Two examples are further presented to illustrate the effectiveness of using the relevance feedback algorithm (see Figs. 8–11). In Fig. 8, the first big red flower is used as the query image and the others are the nineteen most similar images searched by applying the proposed method without user's feedback. Although the component in the query image is a big flower, we can retrieve both big and smaller flower images. Note that the thirteenth underwater-world image, which looks

Fig. 8. The initial retrieval result of the big red flower of D1 shown in the first image without using the relevance feedback algorithm.

Fig. 9.   The retrieval result of the big red flower of D1 after using the relevance feedback algorithm.
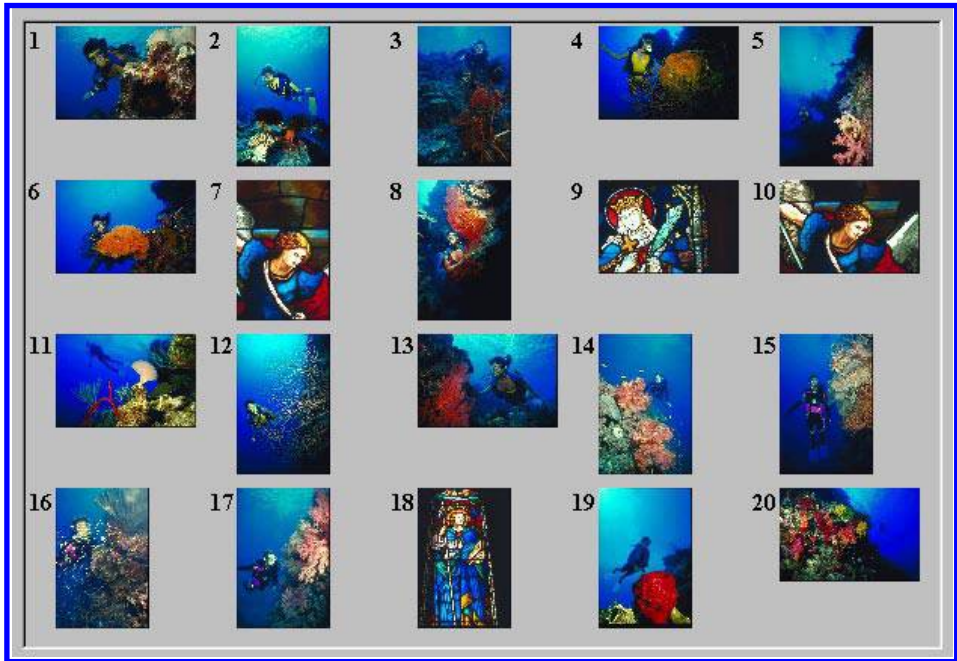

Fig. 10.   The initial retrieval result of the underwater world image of D1 shown in the first image without using the relevance feedback algorithm.
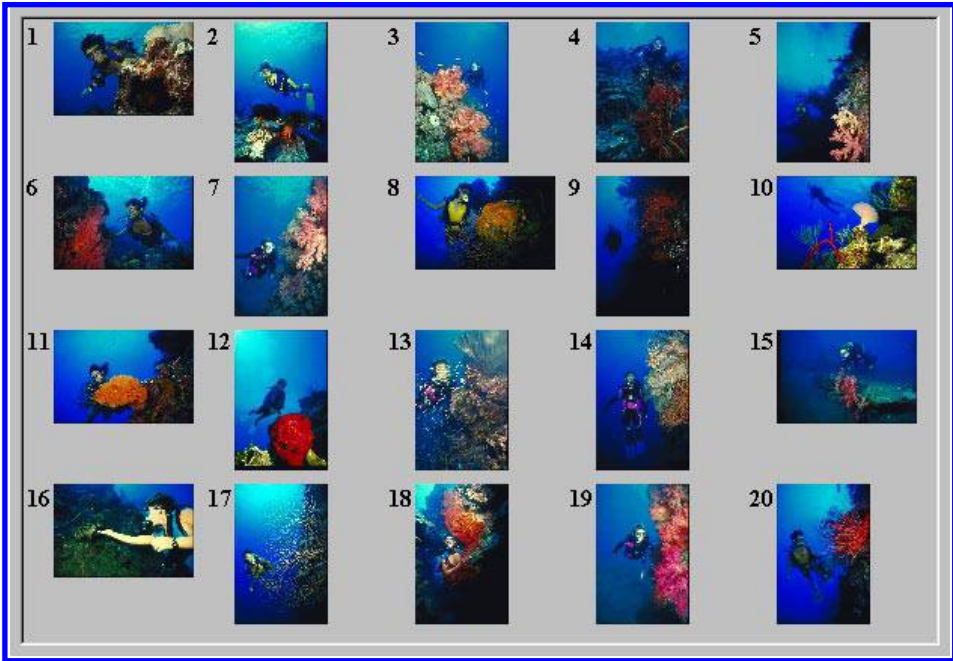
Fig. 11. The retrieval result of the underwater world image of D1 after using the relevance feedback algorithm.

different from the query one, is also retrieved because its right part has red corals. If we select the third image as the relevant one and the thirteenth image as the nonrelevant one, and then apply the relevance feedback algorithm, a better retrieval result is obtained (see Fig. 9). As shown in Fig. 10, the first underwater-world image is used as the query image and the others are nineteen most similar images. As Fig. 10 shows, we can see that the seventh, ninth, tenth and eighteenth retrieved images do not belong to the underwater-world class. After applying the relevance feedback algorithm, we obtain a good result (see Fig. 11).

4.2. *Experimental results on large database D2*

Water-filling and wavelet moments[17] have been used to do color image retrieval on a large database which consists of 17695 images from Corel with 400 airplanes and 100 American eagles. In order to do an objective comparison, we conduct two test databases, D2-1 and D2-2, from Corel. D2-1 has 17695 images with 400 airplanes and 100 eagles. In D2-1, we did not take images with blue sky (almost all of the airplanes and eagles with blue sky). In D2-2, we take the first 17195 images from Corel according to Corel's serial numbers plus 400 airplanes and 100 eagles. That is, those images of D2-2 are more varied and not classified. Table 1 shows the comparison in terms of averaged number of relevant images that are retrieved for 100 airplanes and 100 eagles as query images on D2-1 and D2-2. From Table 1, we

Table 1.   Average number of relevant images that are retrieved for 100
airplanes and 100 eagles as query images on D2-1 and D2-2. The numbers
of retrieved images are 10, 20, 40 and 80.

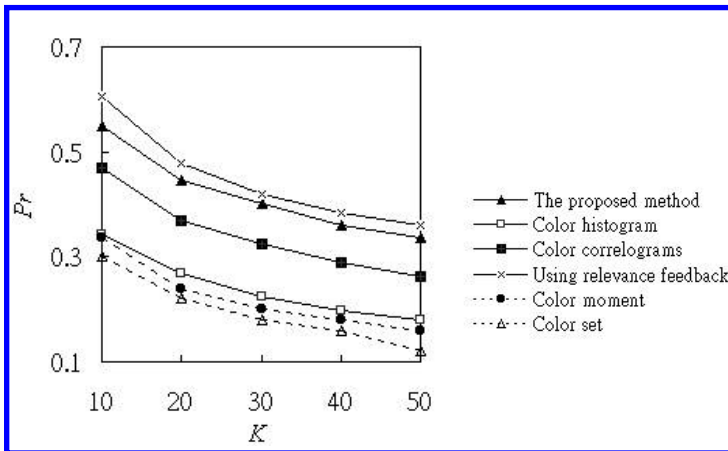|		10	20	40	80
Airplanes	Our method on D2-1	6.2000	11.130	19.680	32.170
	Our method on D2-2	4.1500	6.9200	11.420	18.170
	Water-filling	3.5600	6.2900	10.920	18.030
	Wavelet moments	3.3200	5.7500	9.9400	17.070
Eagles	Our method on D2-1	5.8700	9.8100	16.010	24.140
	Our method on D2-2	3.3100	4.9500	7.0400	9.4800
	Water-filling	2.6538	3.3269	4.9135	6.7885
	Wavelet moments	1.9808	2.8173	4.4327	6.5769

Fig. 12.   The comparison of precision among the proposed method with or without using the
feedback algorithm and other methods on D2-2.

can see that our method is superior to water-filling and wavelet moments in the
large test database.

Finally, based on D2-2, we also implement several other methods using color his-
togram, color moment, color set or color correlograms as features to compare their
performance with ours in the large test database. We randomly select 40 classes and
5 images for each class from D2-2 as query images. Those 200 query images include
the tiger, boat, underwater world, sunset, eagle, lion, penguin, elephant, horse,
flower, mountain, swim pool, fox, waterfall, goat, cat, dog, architecture, pyramid,
duck, monkey, etc. Since it is difficult to decide the total number of relevant images
of each class in the large test database in advance, we just compare the precision
(see Fig. 12). We can see that our method is also superior to other methods on the
large test database.

5. Conclusions

Content-based image retrieval (CBIR) has emerged as one of the most active research areas in recent years. Most of the early research use the color features, including color histogram, color set, color moment and color layout, to do image retrieval. These features only contain the color information of each pixel in an image or the global spatial relations among regions, but the local relationship among neighboring pixels is not involved. Hence, in this paper, a new color image retrieval method based on the color information of the neighboring pixels has been proposed. First, the primitives of each image are extracted. Then, using the primitives, a similarity measure between two images is then proposed for color image retrieval. The experimental results have shown that the proposed method is very efficient and outperforms the methods using color histogram, color moment, color set, color correlograms, water-filling, wavelet moments as features. Furthermore, to make the retrieval result more close to the user's need, a relevance feedback algorithm is proposed to automatically determine the importance of each feature vector. The experimental results also show that the retrieval results after using the relevance feedback algorithm are more appropriate to what the users really want. The future research direction is to combine the primitive and other kinds of feature vectors, such as shape and texture, to do image retrieval.

Acknowledgment

References

1. N. Akrout, R. Prost and R. Goutte, "Image compression by vector quantization: a review focused on codebook generation," *Imag. Vis. Comput.* **12**, 10 (1994) 627–637.
2. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain and C. F. Shu, "The Virage image search engine: an open framework for image management," *Proc. SPIE Storage and Retrieval for Still Image and Video Databases IV*, San Jose, CA, USA, February 1996, pp. 76–87.
3. C. Carson, S. Belongie, H. Greenspan and J. Malik, "Region-based image querying," *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, *in Conjunction with CVPR'97*, 1997, pp. 762–768.
4. Y. Deng and B. S. Manjunath, "An efficient low-dimensional color indexing scheme for region-based image reitrieval," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing* **6** (1999) 3017–3020.
5. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by image and video content: the QBIC system," *IEEE Comput.* **28**, 9 (1995) 23–32.
6. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.
7. J. Huang, S. K. Kumar, M. Mitra, W. Zhu and R. Zabih, "Image indexing using color correlograms," *Proc. CVPR Int. Conf.*, 1997, pp. 762–768.

8. O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," *Proc. Machine Learning Int. Conf.*, 1998.

9. MARs, the Multimedia analysis and retrieval system developed at University of Illinois at Urbana-Champaign. Demo: http://jadzia.ifp.uiuc.edu:8000/.

10. S. Mehrotra, Y. Rui, O. B. Michael and T. S. Huang, "Supporting content-based queries over images in MARS," *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, 1997, pp. 632–633.

11. QBIC, the IBM QBIC project, Demo: http://wwwqbic.almaden.ibm.com/.

12. Y. Rui, T. S. Huang, M. Ortega and S. Mehrotra, "Relevance feedback: a power tool in interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.* (*Special Issue on Interactive Multimedia Systems for the internet*) **8**, 5 (1998) 644–655.

13. J. R. Smith and S. F. Chang, "Visually searching the web for content," *IEEE Trans. Multimed.* **4**, 3 (1997) 12–20.

14. M. Stricker and M. Orengo, "Similarity of color images," *Proc. SPIE Storage and Retrieval for Still Image and Video Databases III*, San Jose, CA, USA, February 1995, pp. 381–392.

15. Virage, the project of Virage Inc., Demo: http://virage.com/.

16. WebSEEk, the World Wide Web oriented text/image search engine, Demo: http://www.ee.columbia.edu/∼sfchang/demos.html/.

17. X. S. Zhou and T. S. Huang, "Image retrieval: feature primitives, feature representation, and relevance feedback," *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries, in Conjunction with CVPR'00*, 2000, pp. 10–14.

Jau-Ling Shih received the B.S. degree in electrical engineering from National Sun Yat-Sen University, Kaohsiung, Taiwan in 1992 and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan in 1994. She is now a Ph.D. candidate at the Department of Computer and Information Science at the National Chiao Tung University. She is also a lecturer at the Department of Computer Science and Information Engineering at the Chung Hua University.

Her current research interests include image processing and image retrieval.

Ling-Hwei Chen received the B.S. degree in mathematics and M.S. degree in applied mathematics from National Tsing Hua University, Hsinchu, Taiwan in 1975 and 1977, respectively, and the Ph.D. in computer engineering from National Chiao Tung University, Hsinchu, Taiwan in 1987.

From August 1977 to April 1979, she worked as a research assistant in the Chung-Shan Institute of Science and Technology, Taoyan, Taiwan, and then as a research associate in the Electronic Research and Service Organization, Industry Technology Research Institute, Hsinchu, Taiwan. From March 1981 to August 1983, she worked as an engineer in the Institute of Information Industry, Taipei, Taiwan. She is now a Professor in the Department of Computer and Information Science at the National Chiao Tung University.

Her current research interests include image processing, pattern recognition, document processing, image compression and image cryptography.

This article has been cited by:

1. Jun Zhang, Lei Ye. 2010. Series feature aggregation for content-based image retrieval. *Computers & Electrical Engineering* **36**:4, 691-701. [CrossRef]

2. Kuo-Liang Chung, Yi-Luen Huang, Yau-Wen Liu. 2007. Efficient algorithms for coding Hilbert curve of arbitrary-sized image and application to window query. *Information Sciences* **177**:10, 2130-2151. [CrossRef]

3. A. Kushki, P. Androutsos, K.N. Plataniotis, A.N. Venetsanopoulos. 2004. Retrieval of Images From Artistic Repositories Using a Decision Fusion Framework. *IEEE Transactions on Image Processing* **13**:3, 277-292. [CrossRef]