Mathematical and Computer Modelling 54 (2011) 1121–1127

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Content-based image retrieval using color and texture fused features

Jun Yue [a,b], Zhenbo Li [b,1], Lu Liu [b], Zetian Fu [b,*]

[a] School of Information Science and Engineering, LUDONG University, YanTai, 264025, China

[b] College of Information and Electrical Engineering, China Agricultural University, Beijing, 100083, China

ARTICLE INFO

Article history:
Received 16 September 2010
Accepted 4 November 2010

Keywords:
Content based image retrieval
HSV
Color
Texture

ABSTRACT

This paper presents a method to extract color and texture features of an image quickly for content-based image retrieval (CBIR). First, HSV color space is quantified rationally. Color histogram and texture features based on a co-occurrence matrix are extracted to form feature vectors. Then the characteristics of the global color histogram, local color histogram and texture features are compared and analyzed for CBIR. Based on these works, a CBIR system is designed using color and texture fused features by constructing weights of feature vectors. The relevant retrieval experiments show that the fused features retrieval brings better visual feeling than the single feature retrieval, which means better retrieval results.

1. Introduction

Images are widely used nowadays. It has the advantage of visual representation and it is usually adopted to express other mediums. With the rapid development of computers and networks, the storage and transmission of a large number of images become possible. Instead of text retrieval, image retrieval is wildly required in recent decades. Content-based image retrieval (CBIR) is regarded as one of the most effective ways of accessing visual data [1]. It deals with the image content itself such as color, shape and image structure instead of annotated text. Huge amounts of data retrieval challenge the traditional database technology, but the traditional text-object database cannot satisfy the requirements of an image database. The traditional way of an annotated image using text, lacks the automatic and effective description of the image. In order to implement CBIR, the system need to understand and interpret the content of managed images. The retrieval index should be produced automatically, which provides more a visual retrieval interface to users.

CBIR refers to image content that is retrieved directly, by which the images with certain features or containing certain content will be searched in an image database. The main idea of CBIR is to analyze image information by low level features of an image [2], which include color, texture, shape and space relationship of objects etc., and to set up feature vectors of an image as its index. Retrieval methods focus on similar retrieval and are mainly carried out according to the multi-dimensional features of an image.

As images are rich in content and without language restrictions to facilitate international exchanges, etc., CBIR has very broad and important applications in many areas including military affairs, medical science, education, architectural design, the justice department and agriculture, etc. Many CBIR systems have been developed gradually. Typical examples of the CBIR retrieval systems include QBIC [3], Virage [4], Photobook [5], VisualSEEk [6], Netra [7] and SIMPLIcity [8] etc.

The progress of CBIR research was lucidly summarized at a high level in [2,9]. Features are the basis for CBIR, which are certain visual properties of an image. The features are either global for the entire image or local for a small group of pixels. According to the methods used for CBIR, features can be classified into low-level features and high-level features.

* Corresponding author. Tel.: +86 13562559603.
E-mail address: yuejun509@gmail.com (Z. Fu).

1 Zhenbo Li contributed equally with Jun Yue, and is the co-first authors for this paper.

0895-7177/$ – see front matter
doi:10.1016/j.mcm.2010.11.044

The low-level features are used to eliminate the sensory gap between the object in the world and the information in a description derived from a recording of that scene. The high-level features are used to eliminate the semantic gap between the information that one can extract from the visual data and the interpretation that the same data has for a user in a given situation.

The most commonly used low-level features include those reflecting color, texture, shape, and salient points in an image [2]. Because of the robustness, effectiveness, implementation simplicity and low storage requirements advantages, color has been the most effective feature and almost all CBIR systems employ colors. HSV or CIE Lab and LUV spaces are used to represent color instead of the RGB space as they are much better with respect to human perception [10]. Generally, the distribution of color was represented by color histograms and formed the images' feature vectors. Shape [11], texture [12] and spatial features [13] etc. were adopted to improve the color based CBIR as different images may have similar or identical color histograms and images taken under different ambient lighting may produce different histograms. The texture feature is another wildly used feature in CBIR, which intended to capture the granularity and repetitive patterns of surfaces within an image [2]. In the MPEG-7 standard, a set of color and texture descriptors including histogram-based descriptors, spatial color descriptors and texture descriptors were defined to interpret natural images [14].

Some researches aim to reducing the semantic gap between the visual features and the richness of human semantics [15]. In order to derive high-level semantic features for CBIR, object-ontology [16] was used to define high-level concepts. Supervised or unsupervised learning methods were used to associate low-level features with query concepts [17]. Relevance feedback was introduced into the retrieval loop for learning of users' intentions [18] and semantic templates [19] were generated to support high-level image retrieval.

As there is inconsistency in understanding visual data for different users, the semantic gap is difficult to eliminate. The most practical CBIR system constructed is based on the color, shape, texture and other low-level features [20]. We experiment using color and texture fused features for a CBIR in the paper. The remainder of the paper is organized as follows. In Section 2, we compare the block color histogram and global color histogram used in color based CBIR. Section 3 describes the work on co-occurrence matrix based texture feature extracting and comparing retrieval results with color-based CBIR. Section 4 introduces our CBIR system based on color and texture fused features, especially the weights of color and texture features. Conclusions and future works are discussed in Section 5.

2. Comparing two features of color based CBIR

Color based image retrieval is the most basic and most important method for CBIR. Color features are the most intuitive and most obvious image features. It is also an important feature of perception. Comparing with other image features such as texture and shape etc., color features are very stable and robust. It is not sensitive to rotation, translation and scale changes. Moreover, the color feature calculation is relatively simple. A color histogram is the most used method to extract color features.

A color histogram is a frequency statistic for different colors in a certain color space. The advantage is that it describes the global color distribution for images. It is especially suited for those images difficult to segment and neglect spatial locations. However, its drawback is that it cannot describe the local distribution of the image in color space and the spatial position of each color. It means that the color histogram cannot describe specific objects or things in the image.

The color space needs to be divided into several small ranges in order to calculate the color histogram. Each interval is regarded as a bin. Thus, the color is quantized. The color histogram can be calculated through counting pixels where the colors fall into each interval. Color features include global color histogram and block color histogram.

2.1. Global color histogram based CBIR

For an image selected from an image database, because an RGB color space does not meet the visual requirements of people, for image retrieval, the image normally converts from an RGB space to other color spaces. The HSV space is used in the paper as it is a more common color space. The global color histogram can be calculated as follows:

Step 1. Convert the images from RGB space to HSV space.

Step 2. Quantify the images using formula 1.

$$H = \begin{cases} 0 & h \in [316, 360] \\ 1 & h \in [1, 25] \\ 2 & h \in [26, 40] \\ 3 & h \in [41, 120] \\ 4 & h \in [121, 190] \\ 5 & h \in [191, 270] \\ 6 & h \in [271, 295] \\ 7 & h \in [295, 315] \end{cases} \tag{1}$$

$$S = \begin{cases} 0 & s \in [0, 0.2) \\ 1 & s \in [0.2, 0.7) \\ 2 & s \in [0.7, 1] \end{cases}$$

Fig. 1. Color based CBIR retrieval results. (a) Input image for retrieval. (b) Using global color histogram. (c) Using 3 × 3 block color histogram.

$$V = \begin{cases} 0 & v \in [0, 0.2) \\ 1 & v \in [0.2, 0.7) \\ 2 & v \in [0.7, 1]. \end{cases}$$

Step 3. Count each feature value.
Step 4. Calculate similarity by Euclidean distance using formula 2.

$$D = \sum_{i=1}^{n} (A_i - B_i)^2. \tag{2}$$

Here, A and B are two feature vectors. n is the dimension of feature vectors.

The global color histogram is a simple way of extracting image features. High effective calculation and matching are its main advantage. The feature is invariable to rotation and translation. The drawback is that the global color histogram only calculates the frequency of colors. The spatial distribution of color information is lost. Two completely different images can get the same global color histogram, which will cause retrieval errors.

2.2. Block color histogram based CBIR

For the block color histogram, an image is separated into $n \times n$ blocks. Each block will have less meaning if the block is too large, while computation of retrieval process will be increased if the block is too small. Through our comparative analysis, a two-dimensional space divided into 3 × 3 will be more effective.

For each block, we carry out a calculation of the color space converting and color quantisation. The normalized color features for each block can be calculated.

Usually in order to highlight the specific weight of different blocks, a weight coefficient is distributed to each block. And the weight of middle block is often larger.

2.3. Comparing two color features based CBIR

We compared the two color features in our CBIR system. Given an input image, two color based image retrieval approaches are adopted respectively, and the retrieval results are shown as Fig. 1.

From the results, it is obviously to see that the global color histogram counts the overall image color information without spatial information. So a great characteristic is rotational invariance. It is reflected well in the first four images of Fig. 1(b). But a non-car image appears in Fig. 1(b), which is only because the overall color is relatively similar, but they are completely different images actually. However, since the block color histogram contains certain position information, the results as in Fig. 1(c) will be better than the global color histogram. Experiments show that the block color histogram is better than the global color histogram from the human visual perception.

3. Comparing texture and color based CBIR

We introduce our work on texture-based CBIR, and compare the method with color-based CBIR in this section. A co-occurrence matrix is adopted to extract texture features as it requires less computation and has a faster feature extraction speed. Moreover, the feature vector dimensions are also smaller.

3.1. Calculating the co-occurrence matrix

Suppose the input image has Nc and Nr pixels in the horizontal and vertical directions respectively. Assume $Zc = \{1, 2, \dots, Nc\}$ is a horizontal space domain and $Zr = \{1, 2, \dots, Nr\}$ is a vertical space domain. When the direction θ and distance d are given, the matrix element $P(i, j/d, \theta)$ can be expressed by calculating the pixel logarithm of co-occurrence grey level i and j. Assume the distance is 1, θ equals 0°, 45°, 90°, 135° respectively, the formulae are:

$$P(i,j/1,0) = \#\left\{\begin{matrix}[(k,l),(m,n)] \in (Zr \times Zc) \\ |k-m| = 0, |l-n| = 1, f(k,l) = i, f(m,n) = j\end{matrix}\right\} \tag{3}$$

$$P(i,j/1,90) = \#\left\{\begin{matrix}[(k,l),(m,n)] \in (Zr \times Zc) \\ |k-m| = 1, |l-n| = 0, f(k,l) = i, f(m,n) = j\end{matrix}\right\} \tag{4}$$

$$P(i,j/1,45) = \#\left\{\begin{matrix}[(k,l),(m,n)] \in (Zr \times Zc)(k-m) = 1, (l-n) = -1 \\ \text{or} \quad (k-m) = -1, (l-n) = 1, f(k,l) = i, f(m,n) = j\end{matrix}\right\} \tag{5}$$

$$P(i,j/1,135) = \#\left\{\begin{matrix}[(k,l),(m,n)] \in (Zr \times Zc)(k-m) = 1, (l-n) = 1 \\ \text{or} \quad (k-m) = -1, (l-n) = -1, f(k,l) = i, f(m,n) = j\end{matrix}\right\} \tag{6}$$

where, k, m and l, n represent changes of selected calculating windows, # represents pixel logarithm which establishes brackets.

3.2. Extracting texture features

The texture features are extracted in 5 steps in our system.

Step 1. Image color conversion.

The color image will be converted to a grey-scale image by formula 7 [21], the number of the grey-scale is 256.

$$Y = 0.29 \times R + 0.587 \times G + 0.114 \times B \tag{7}$$

where Y is the grey-scale value. R, G, B represent red, green and blue component values respectively.

Step 2. Grey-scale quantification.

Because the grey-scale is 256, the corresponding co-occurrence matrix is 256 × 256. According to the human visual feature, the similarity of most images is mainly distinguished by the relative coarse texture features. The grey-scale of the initial image will be compressed to reduce calculations before the co-occurrence matrix is formed. 16 compression levels were chosen in the paper to improve the texture feature extracting speed.

Step 3. Feature value calculation.

Four co-occurrence matrices are formed according formula 3 to formula 6 in four directions. The four texture parameters: capacity, entropy, moment of inertia and relevance are calculated. Finally, the means and standard deviations of each parameter are taken as each component of the texture features.

Step 4. Internal normalization.

For an image I^i and its corresponding feature vector $H^i = [h^{i,1}, h^{i,2}, \dots, h^{i,N}]$, assume the feature component value satisfies a Gaussian distribution. The Gaussian normalization approach is used to implement internal normalization in order to make each feature of the same weight.

$$h^{i,j'} = \frac{h^{i,j} - m_j}{\sigma_j} \tag{8}$$

where, m_j is the mean and σ_j is the standard deviation. $h^{i,j}$ will be unitized on range [−1,1].

Step 5. Texture feature comparison.

The texture feature of each image is calculated according to the above steps. The texture values are compared by Euclidean distance, the closer the distance the higher the similarity.

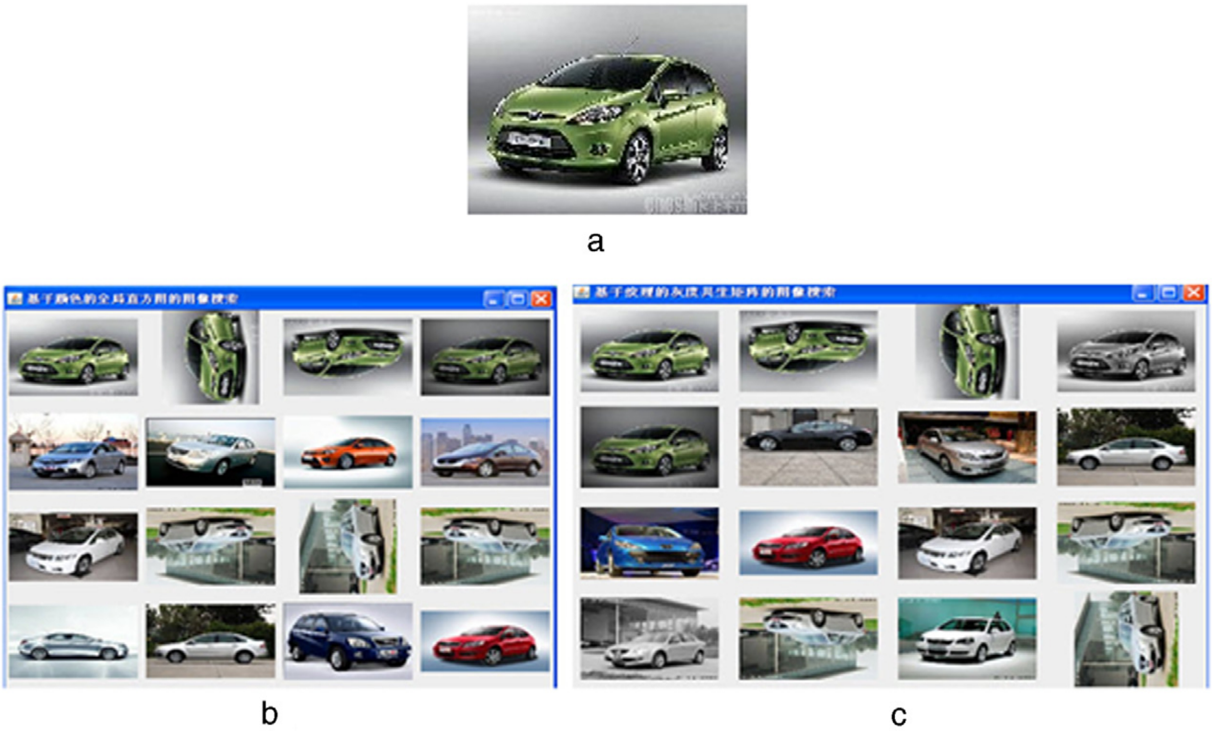

Fig. 2. Comparing color and texture CBIR results. (a) Input image for retrieval. (b) CBIR using global color histogram. (c) CBIR using texture features.

3.3. Experiment results and comparing with color based CBIR

In image retrieval, the feature vector of each image in the image database will subtract the feature vector of target image separately as feature values. And the sum of the square of the result feature value is calculated to get the Euclidean distance. The images are queued according to the distance ordered from small to large.

In order to test the retrieval effect by extracting texture features through a co-occurrence matrix, the performance of the texture feature algorithm based on the grey-scale co-occurrence matrix is taken to compare with the color-based retrieval results, which are shown in Fig. 2.

From the results, it can be see that the retrieval results are closed to the input image from the point of view of color when just using the color histogram. When only using texture features, the black–white image emerges. Though they are with quite different colors, the textures are very similar. But in Fig. 2(b), the black–white car image does not emerge.

From Fig. 2(c), it is shown that the texture feature extracted by the grey-scale co-occurrence matrix belongs to the whole image domain. It stresses more on the dependence of grey space, which reflects the pixels grey-scale space relationship under the same texture mode. Only using one low-level retrieval feature cannot achieve a good result.

4. CBIR system using color and texture fused features

In order to implement a fast and robust CBIR system, color and texture features are fused in the paper. The system was developed using Java under the Eclipse development environment. SQL server 2005 served as the system database.

We extract N colors which can reproduce the initial image dominant hue well in the HSV color space as the color feature. The contrast, capacity, entropy and relevance of an image are calculated by the grey-level co-occurrence matrix approach as the texture feature. Based on the above image feature vectors, the color and texture features are fused by a linear weighted mode. The similarities are calculated by Euclidean distance in our work. The process can be described using three steps.

Step 1. The HSV color space is selected, which can reproduce human vision color features well. Under the color space, the dominant color of the initial image is extracted as color feature vectors through a clustering–analyzing algorithm considering the color resolution and color feature dimensions.

Step 2. Considering that only color features cannot express the semantic information of an image sufficiently, capacity, entropy and relevance of an image are calculated by the grey-level co-occurrence matrix approach as the texture features.

Step 3. Color feature and texture feature are fused to realize comprehensive image retrieval. The weight of color and texture features is determined through lots of experiments. Linear weighted mode combining with similar distances of color and texture features are adopted to retrieve images comprehensively.

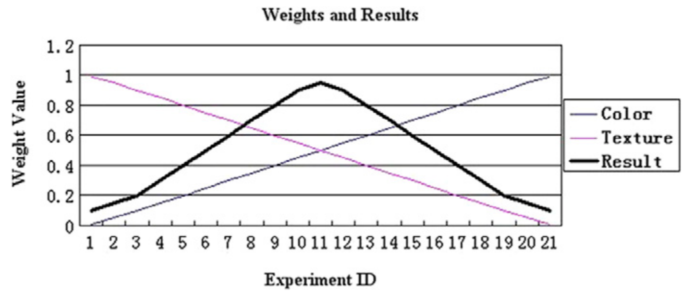

Fig. 3. Experiment results of color and texture feature weight.

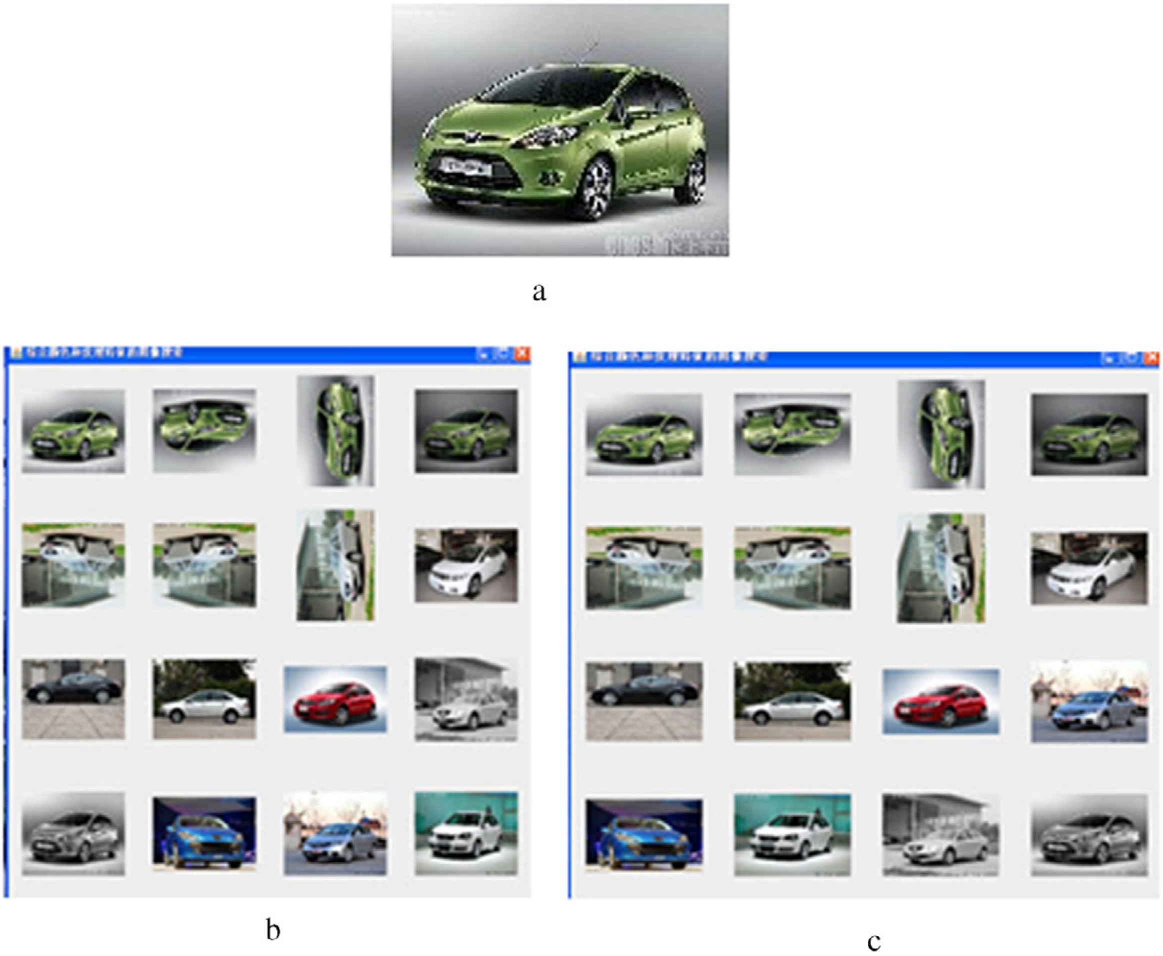

Fig. 4. CBIR using color and texture fused features. (a) Input image for retrieval. (b) Color, texture weights as 0.5, 0.5. (c) Color, texture weights as 0.6, 0.4.

In order to find the proper weight values of color and texture features, experiments have been done 21 times when the retrieval approach combining the two features is adopted, the results are shown in Fig. 3.

One weight value changes to smaller from near 1 in the range of (0,1) gradually, meantime the other weight value will be larger from near 0 in the range of (0,1) correspondingly keeping the sum of the two weight values as 1. With the difference between the two weight values approximating 0, retrieval results approach the input image which is the expected result. When the two weight values are approximately equal, the retrieval results approximate to be the best visual effect. Therefore, the default weights of color and texture features in the system are assumed as 0.5 and 0.5.

In Fig. 4(b), the weights of color and texture are 0.5 and 0.5 respectively, while the weights of color and texture are 0.6 and 0.4 respectively in Fig. 4(c). Observe black–white images among the retrieval results in Fig. 4(b) and (c), i.e. the fourth image from right-bottom in Fig. 4(b) and the right-bottom image in Fig. 4(b). Obviously, the result image in Fig. 4(b) is closer to the expected result as its black–white texture is the same with the expected image, the only difference being the color.

The weight of the color and texture are 0.5 and 0.5 which shows more reasonable, by which the image retrieval accuracy is improved.

5. Conclusions and future works

After a survey the previous CBIR works, the paper explored the low-level features of color and texture extraction for CBIR. After comparing the two color histogram features as well as comparing color and texture features, the paper implemented a CBIR system using color and texture fused features. Similar images can be retrieved quickly and accurately by inputting an image. More low-level features such as shape and spatial location features etc. will be fused to make the system more robust in the future. The image feature matching method and semantic based image retrieval are the other two important aspects for the CBIR system.

Acknowledgement

This work was supported by the National Nature Science Foundation of China (Grant No. 60875039).

References

[1] P. Muneesawang, L. Guan, An interactive approach for CBIR using a network of radial basis functions, IEEE Transactions on Multimedia 6 (2004) 703–716.
[2] R. Datta, D. Joshi, J. Li, J.Z. Wang, Image retrieval: ideas, influences, and trends of the new age, ACM Computing Surveys 40 (2) (2008) 1–60.
[3] W. Niblack, R. Barber, W. Equitz, et al., The QBIC project: querying images by content using color, texture, and shape, in: SPIE 1908, San Jose, CA, 1993, pp. 173–187.
[4] A. Pentland, R.W. Picard, S. Scarloff, Photobook: tools for content-based manipulation of image databases, in: SPIE 2185, San Jose, CA, 1994, pp. 34–47.
[5] S. Mehrotra, Y. Rui, M. Ortega, et al., Supporting content-based queries over images in MARS, in: Proc. of IEEE Int'l Conf. on Multimedia Computing and Systems'97, Ottawa, Ontario, Canada, 1997, pp. 632–633.
[6] J.R. Bach, C. Fuller, A. Gupta, et al., Virage image search engine: an open framework for image management, in: SPIE 2670, 23, San Jose, CA, 1996, pp. 76–87.
[7] J.R. Smith, Integrated spatial and feature image systems: retrieval, analysis and compression, Ph.D. Dissertation, Columbia University, New York City, 1997.
[8] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (9) (2001) 947–963.
[9] A.W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.
[10] H. Muller, N. Michoux, D. Bandon, A. Geissbuhler, A review of content-based image retrieval systems in medical applications—clinical benefits and future directions, International Journal of Medical Informatics 73 (2004) 1–23.
[11] G. Gagaudakis, P.L. Rosin, Incorporating shape into histograms for CBIR, Pattern Recognition 35 (2002) 81–91.
[12] P.S. Hiremath, Jagadeesh Pujari, Content based image retrieval using color, texture and shape features, in: 15th International Conference on Advanced Computing and Communications, ADCOM 2007, 2007, pp. 780–784.
[13] Y.K. Chan, C.Y. Chen, Image retrieval system based on color-complexity and color-spatial features, Journal of Systems and Software 71 (2004) 65–70.
[14] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Transactions on Circuits and Systems for Video Technology 11 (6) (2001) 703–715.
[15] Y. Lu, C. Hu, X. Zhu, H. Zhang, Q. Yang, A unified framework for semantics and feature based relevance feedback in image retrieval systems, in: Proceedings of the ACM International Conference on Multimedia, 2000.
[16] V. Mezaris, I. Kompatsiaris, M.G. Strintzis, An ontology approach to object-based image retrieval, in: Proceedings of the ICIP, vol. II, 2003, pp. 511–514.
[17] I.K. Sethi, I.L. Coman, Mining association rules between low-level image features and high-level concepts, in: Proceedings of the SPIE Data Mining and Knowledge Discovery, vol. III, 2001, pp. 279–290.
[18] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Transactions on Circuits and Systems for Video Technology 8 (5) (1998) 644–655.
[19] Y. Zhuang, X. Liu, Y. Pan, Apply semantic template to support content-based image retrieval, in: Proceedings of the SPIE, Storage and Retrieval for Media Databases, vol. 3972, 1999, pp. 442–449.
[20] X. Yuan, C. Wu, M. Zhu, D. Wang, CBIR retrieval based on dominant color selection, Journal of Computer Research and Development 39 (9) (2002) 1120–1125.
[21] H. Wang, P. Shi, An approach for feature extraction of texture images, Journal of Communication University of China (Science and Technology) 13 (1) (2006) 49–52.