Multimed Tools Appl (2017) 76:21787–21809
DOI 10.1007/s11042-016-4096-5

A fast and effective image retrieval scheme using color-, texture-, and shape-based histograms

Amandeep Khokher[1] **· Rajneesh Talwar**[2]

Received: 21 August 2015 / Revised: 2 September 2016 / Accepted: 27 October 2016 /
Published online: 9 November 2016

Abstract The rapid growth of digital image collections has prompted the need for development of software tools that facilitate efficient searching and retrieval of images from large image databases. Towards this goal, we propose a content-based image retrieval scheme for retrieval of images via their color, texture, and shape features. Using three specialized histograms (i.e. color, wavelet, and edge histograms), we show that a more accurate representation of the underlying distribution of the image features improves the retrieval quality. Furthermore, in an attempt to better represent the user's information needs, our system provides an interactive search mechanism through the user interface. Users searching through the database can select the visual features and adjust the associated weights according to the aspects they wish to emphasize. The proposed histogram-based scheme has been thoroughly evaluated using two general-purpose image datasets consisting of 1000 and 3000 images, respectively. Experimental results show that this scheme not only improves the effectiveness of the CBIR system, but also improves the efficiency of the overall process.

Keywords Content-based image retrieval · Feature extraction · Color histogram · Discrete wavelet transform · Robinson compass masks · Graphical user interface

1 Introduction

In recent years, rapid advances in multimedia technology have produced a large amount of image data in diverse areas, such as medicine, journalism, military, architectural and

✉ Amandeep Khokher
amandeep.khokher@gmail.com

1 I.K. Gujral Punjab Technical University, Kapurthala, India

2 CGC Technical Campus, Jhanjeri, India

engineering design, crime prevention, art galleries, remote sensing systems, etc. As the popularity of digital images grows, the need to store and retrieve images in an intuitive and efficient manner arises. Therefore, there is an urgent need to develop efficient and automatic tools to solve the management problem for the growing image databases.

Depending on the query formats, there are two different types of methods usually adopted in image retrieval: text- and content-based [22]. In text-based retrieval systems, text descriptors, such as keywords and captions are used to annotate and retrieve images. However, there are several difficulties with this seemingly attractive approach [33, 52]. First, since an image usually embodies rich information, it is very difficult to describe image contents with a small set of keywords. Second, vast amount of labour is required in manual annotation of images. This process becomes tedious, cumbersome and expensive for large image databases. Third, the task of describing image content is quite subjective. That is, due to perception subjectivity, an annotator and a user may use different words to describe the same image content, leading to a deteriorated image retrieval performance.

To alleviate the difficulties of text-based systems, an alternative approach, the so-called content-based image retrieval (CBIR) [8, 14], has been proposed in the research community. In CBIR, instead of keywords, images are represented by numerical features directly extracted from the image pixels. Generally, low-level visual features such as color, texture, and shape are extracted and represented in the form of feature vectors. These vectors are stored in a different database called feature database. When a user submits a query by providing an existing image (or creating one by sketching), its feature vectors are similarly constructed and matched with those in the feature database. The system ranks the database images in a decreasing order of similarity to the query image and retrieves a given number of most similar target images from the database. Recent retrieval systems have incorporated users' relevance feedback in the retrieval process. Relevance feedback is a query modification technique which attempts to capture the user's precise needs through iterative feedback and query refinement [8]. By considering user's feedback, the retrieval system automatically adjusts the query and provides refined results more in line with what the user wants. Since the inception of CBIR, many techniques have been proposed and several CBIR systems have been developed to index and retrieve images based on their content. Examples of CBIR systems include QBIC [11], Photobook [44], VisualSEEk [57], NeTra [36], Blobworld [4], PicHunter [7], PicToSeek [13], and SIMPLIcity [64].

In early stages of the development of CBIR, research was primarily focused on developing a single concise feature like color, shape, or texture. However, it is hard to attain satisfactory retrieval effectiveness by using just one feature because, in general, an image contains various visual characteristics. Therefore, attempts have been made by researchers to combine different features for effective image retrieval. However, it is a challenging problem to use multiple features for image retrieval. Since a particular visual feature tends to capture only one aspect of the image properties, different weights needs to be assigned to each feature in accordance with the amount of discriminatory information they carry. A general-purpose image retrieval system should be able to automatically decide on what weights should be chosen for good retrieval performance. Alternately, the system can involve a user to give feedback to the system by assigning the weights for each of the features in accordance with his/her judgement of significant features.

In this paper, we propose an efficient and effective CBIR system using color-, texture- and shape-based histograms. For representing color information of an image, color histogram in HSV color space is used. Wavelet histogram generated from 3-level wavelet decomposed image is used for texture information and edge histogram obtained from Laplacian filtered image is used for representing shape. Each histogram leads to a similarity value and a weighted linear combination of the three similarity values is used for retrieving relevant images from the database. The objective of using this histogram-based querying approach is to improve the retrieval effectiveness and efficiency simultaneously. The main advantages of the proposed system can be listed as follows.

- The computational complexity of the system is low, hence, it may be employed in time-critical systems efficiently.
- The proposed system is robust since all the included histograms are normalized before similarity matching.
- The system is flexible since a variety of parameters, including feature weights can be adjusted for achieving retrieval refinement according to user's need.
- The system is simple because of the ease with which it can be operated through the graphical user interface (GUI).
- The system's algorithm can be applied to virtually all kinds of color image databases rather than specific databases.

The rest of this paper is organized as follows. Section 2 provides a review about the related works in CBIR. Section 3 presents our proposed image retrieval system along with a description of the considered color, texture, and shape features. Similarity measures are explained in Section 4 followed by experimental results in Section 5. Finally, concluding remarks are given in Section 6.

2 Related work

Since its advent, CBIR has been actively investigated by researchers from a wide range of disciplines. Comprehensive surveys exist on the different techniques used in this area [31, 33, 54]. Also, there are some literatures that survey the important CBIR systems [52, 61]. Early systems mostly adopted simple features such as color, texture, and shape for image retrieval, while more effective features such as GIST [41], SIFT [34], CNN [49], VLAD [23], and BoW [70] have been popular recently. Since the work in this paper is related to search using color, texture, and shape features, this section mainly reviews existing works based on these features.

Amongst the primitive visual features, color is most extensively used in CBIR. As conventional color features used in CBIR, there are color histogram [60], color moments [58], color coherence vector [43], and color correlogram [20] under a certain color space. The color histogram is often used to represent the color information, but it does not take into account the spatial distribution of color across different areas of the image. Hence, color representation techniques that incorporate spatial information have been investigated for more accurate retrieval. In [27], Li presented a novel algorithm based on running sub-blocks with different similarity weights for object-based image retrieval. By splitting the entire image into certain sub-blocks, color region information and similarity matrix analysis are used to retrieve images under the query of special object. Yoo et al. [69] proposed a new system using a signature-based color-spatial image retrieval method. Color and its spatial distribution within the image are used for features. For the purpose of effectively retrieving more similar images from the digital image databases, Lu and Chang [35] used the color distributions, the mean value and the standard deviation, to represent the global characteristics of the image, and the image bitmap to represent the local characteristics of the image.

Texture is another important property of images that has been intensively studied in pattern recognition and computer vision. As conventional texture features used in CBIR, there are gray-level co-occurrence matrices [18], Gabor filters [37], Wold features [30], wavelet transform [63], and Markov random fields [32]. In [5], Chun et al. proposed two new texture features, block difference of inverse probabilities (BDIP) and block variation of local correlation coefficients (BVLC), for CBIR. Pi et al. [46] presented a novel, effective, and efficient characterization of wavelet subbands by bit-plane extractions in texture image retrieval. In [16], Han and Ma proposed rotation-invariant and a scale-invariant Gabor representations, where each representation only requires few summations on the conventional Gabor filter impulse responses.

Shape is yet another important visual feature that identifies objects in images. It has been used in CBIR in conjunction with color and other features for efficient and robust image retrieval. Shape matching is a well-explored research area, and many shape representation methods and description techniques exists in the literature. In general, the shape representations are classified in two groups [52]: boundary-based (e.g. Fourier descriptors [45]) and region-based (e.g. moment invariants [19]). The former is based on the outer boundary of the shape while the latter is based on the entire shape region. In [17], Han and Guo presented a novel five-stage image retrieval method based on salient edges. In [68], Xu et al. presented a partial shape matching technique using dynamic programming for the retrieval of spine X-ray images. Wei et al. [66] proposed a content-based trademark retrieval system with a feasible set of feature descriptors, which is capable of depicting global shapes and interior/local features of the trademarks.

To facilitate a more accurate retrieval process, a number of researches using a combination of multiple features have been carried out. In [28], Liapis and Tziritas explored image retrieval mechanisms based on a combination of texture and color features. Texture features are extracted using Discrete Wavelet Frames analysis. Two- or one-dimensional histograms of the CIE Lab chromaticity coordinates are used as color features. Chun et al. [6] proposed a CBIR method which uses the effective combination of color autocorrelograms of the hue and saturation component images in HSV color space, and BDIP and BVLC moments of the value component image. In [65], Wang et al. proposed a new and effective color image retrieval scheme which uses the combination of dynamic dominant color, steerable filter texture feature, and pseudo-Zernike moments shape descriptor.

Since low-level features do not necessarily represent the high-level semantics of an image, a few research works in CBIR have developed interactive mechanisms that involve a human as part of the retrieval process. Rui et al. [51] introduced a Human-Computer Interaction approach to CBIR based on relevance feedback. During the retrieval process, the user's high-level query and perception subjectivity are captured by dynamically updated weights based on the user's feedback. In [67], Wu and Zhang presented a feature re-weighting approach using the standard deviation of feature values from relevant images as well as the distribution pattern of irrelevant images on the axis of each feature component. In [15], Guldogan and Gabbouj presented a relevance feedback method for CBIR systems based on dynamic feature weights.

3 Proposed histogram-based approach

In developing a CBIR system, the first critical decision to be made is to determine what image feature, or combination of image features is to be used for image indexing and retrieval purposes. In this section, we first present an overview of our proposed image retrieval system and then review the considered low-level features in our approach.

3.1 Overview of the proposed system

Figure 1 shows the block diagram of the proposed CBIR system. It operates in five steps:

1. *Feature extraction*: In this step, color, texture, and shape features of the database images are extracted and saved in a feature database in the form of feature vectors.
2. *Querying*: When a user provides a query image, its feature vectors are constructed using the same feature extraction algorithm as implemented in step 1.
3. *Similarity measurement*: The system computes the similarities between the color, texture, and shape feature vectors of query image and the database images using the respective similarity measures. The overall similarity between two images is calculated by linearly combining the similarity results of individual feature-based queries.
4. *Retrieval*: The system retrieves and displays 20 top-ranked target images in decreasing order of similarity in the GUI. As a result, the user is able to visualize the best matches to his/her query image.
5. *Refining search*: Using the initial result set, the user can give some feedback to the system by assigning weights to different features, depending on his/her interests. The system then processes this feedback and provides refined results according to the user's criteria.

3.2 Color feature based on color histogram

Color is simple, straightforward, and most widely used visual content in image retrieval. Selecting a powerful yet economic color-based descriptor is important in the design of the CBIR system. In our experiments, the development of color feature extraction algorithm follow a progression: (1) selection of a color space, (2) quantization of the color space, (3) computation of a histogram, and (4) normalization of the histogram.

To represent color, a color space must first be selected. There are a number of different color spaces currently used in image processing. In order to choose a better color space for our color feature extraction algorithm, three different color spaces, viz., RGB, HSV, and YCbCr were tested for comparison. Ultimately, HSV (hue, saturation, value) [55] turned

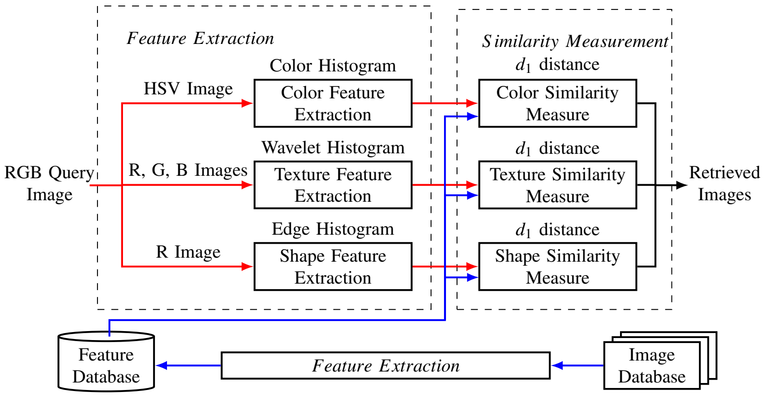


Fig. 1 The block diagram of the proposed CBIR system. The *red* and *blue* lines indicate the path of query image and database image, respectively

out to be the most effective of the three for our experiments. In this space, hue is used to distinguish colors, saturation describes the amount of white light present in a color and value corresponds to the brightness of color. The HSV color space is developed to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color [38].

Color quantization is useful for reducing the storage space and the computational complexity. Furthermore, it improves the system performance because it eliminates the insignificant colors and emphasizes the prominent colors. In HSV color space, quantization of hue requires the most attention. The human visual system is more sensitive to hue than saturation and value, and therefore, hue should be quantized finer than saturation and value [62]. In our algorithm, hue is quantized to 12 levels, saturation to 5 levels, and value to 5 levels. This quantization method has been successfully applied in the research of CBIR [2, 62]. For a detailed description on this method, interested readers may directly refer to them. As a result of quantization, we obtain 300 ($12 \times 5 \times 5$) distinct colors.

To extract the color features, a color histogram in the quantized HSV color space is used. Color histogram [60] is one of the most widely used visual feature in color-based image retrieval [24]. The advantages of color histogram include simple procedure, quick calculation, and its effectiveness in characterizing the distribution of colors in an image. Given a discrete color space, the color histogram of an image is constructed by counting the number of pixels of each color. Accordingly, the color histogram $H(I)$ for a given image I is defined as a vector: $H(I) = [h_{c_1}, h_{c_2}, \ldots, h_{c_i}, \ldots, h_{c_k}]$, where each element h_{c_i} represents the number of pixels of color c_i in the image and k is the number of bins in the histogram.

Often we work with normalized histograms while comparing images. A normalized color histogram $H_N(I)$ of image I is calculated as: $H_N(I) = \frac{H(I)}{p}$, where $H(I)$ is the starting histogram and p is the sum of values of $H(I)$.

3.3 Texture feature based on wavelet histogram

Texture refers to a visual pattern that have properties of homogeneity that do not result from the presence of only a single color or intensity [56]. Because of its importance and usefulness, numerous varieties of texture features have been used in image retrieval applications. The discrete wavelet transform (DWT) is a popular technique for extracting texture features from images, and has been successfully used for image retrieval [26]. In this paper, an efficient wavelet based multiresolution histogram generated from 3-level DWT decomposed images is used for texture characterization. In the DWT decomposition procedure, Haar wavelets are used, since they are fastest to compute and easiest to implement [40].

The first step consists in computing the 1-level wavelet decomposition of the red (R) component of the image. As a result, the image is decomposed into four sub-bands, namely, the approximation sub-band A_1, the horizontal detail sub-band H_1, the vertical detail sub-band V_1, and the diagonal detail sub-band D_1. Applying the same procedure to the sub-band A_1 generates the 2-level wavelet transform consisting of four sub-bands of A_1: A_2, H_2, V_2, and D_2. A 3-level pyramid wavelet transformation is then obtained by applying the same procedure to the approximation sub-band A_3 resulting in A_3, H_3, V_3, and D_3 sub-bands. Figure 2 shows the 3-level DWT decomposition using the Haar wavelet.

After applying DWT, 2-bin normalized histograms of approximation, horizontal, and vertical sub-bands of 3-level decomposed image (i.e., A_k, H_k, V_k; for $k = 1, 2, 3$) are computed. Normalized histograms are similarly computed from the green (G) and blue (B) component images. Finally, an integrated histogram (termed as wavelet histogram, since it

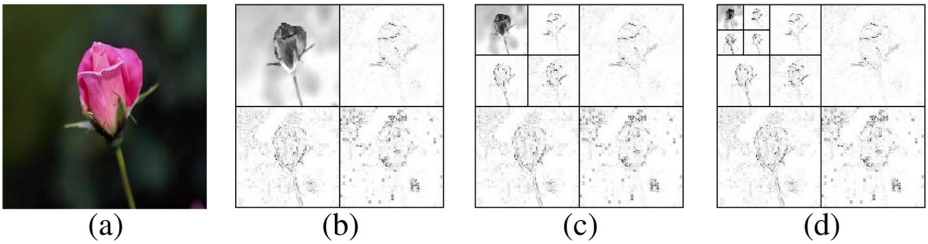


Fig. 2 A 3-level DWT decomposition using Haar wavelet: **(a)** original RGB image (256 × 256), **(b)** decomposition at level 1, **(c)** decomposition at level 2, and **(d)** decomposition at level 3

is generated in the wavelet domain) is formed by concatenating the histograms obtained from all three component images, giving $3 \times 2 \times 3 \times 3 = 54$ features, where the factors indicate, the number of sub-bands for each level, the number of histogram bins for each sub-band, the number of decomposition levels, and the number of component images.

3.4 Shape feature based on edge histogram

Shape is a key attribute for perceptual object recognition, and its efficient representation plays an important role in image retrieval. The main objective of shape description is to measure geometric attributes of an object, that can be used for classifying, matching and recognizing objects [12]. In image retrieval, it is usually required that the shape descriptor is invariant to scaling, rotation, and translation [53].

In our experiments, we represent the shape content of an image on the basis of its edge histogram (i.e., histogram of the edge pixels). Technically, edge detection is a process of finding places in an image where the intensity changes rapidly. This can be done by using edge detection algorithms such as Sobel, Roberts, Prewitt, Laplacian, etc. In our experiments, we use Robinson compass masks [50] for detecting the edges. The Robinson Compass edge operator provides better edge information with an advantage of being less sensitive to noise and extract explicit information about edges in any direction [3]. The masks are defined by taking a single mask and rotating it to eight major compass orientations: North, North-west, West, South-west, South, South-east, East and North-east as represented in Fig. 3.

$$\underset{r_0}{\begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix}}\quad \underset{r_1}{\begin{bmatrix}0&1&2\\-1&0&1\\-2&-1&0\end{bmatrix}}\quad \underset{r_2}{\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix}}\quad \underset{r_3}{\begin{bmatrix}2&1&0\\1&0&-1\\0&-1&-2\end{bmatrix}}$$

$$\underset{r_4}{\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix}}\quad \underset{r_5}{\begin{bmatrix}0&-1&-2\\1&0&-1\\2&1&0\end{bmatrix}}\quad \underset{r_6}{\begin{bmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{bmatrix}}\quad \underset{r_7}{\begin{bmatrix}-2&-1&0\\-1&0&1\\0&1&2\end{bmatrix}}$$

Fig. 3 Robinson compass masks

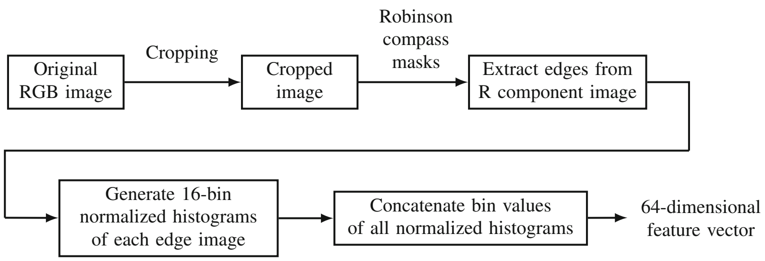


Fig. 4 Block diagram of the shape feature extraction method

The proposed shape feature extraction method consists of four steps. The block diagram of feature extraction process is shown in Fig. 4.

The original RGB image is cropped. The input $M \times N$ image is divided into two areas: the central area and the peripheral area as shown in Fig. 5a. Here, M and N are the row and column sizes of the image. Considering that it is very common to have the major object located in the central position in an image, we eliminate the background details by removing the narrow peripheral area and consider only the central area for image retrieval. Figure 5b and c, respectively show an example of the original RGB image and the resultant cropped image.

In the second step, edge detection is implemented by convolving the R component of cropped image with the eight masks resulting in respective edge images. In the third step, histograms are computed by uniformly quantizing the edge pixels into 8 bins and then normalizing the bins. Normalization is done by dividing each bin value by the total number of pixels in the edge image. Since there are eight edge images, corresponding to eight masks, therefore, eight quantized and normalized histograms are computed. In the last step, the bin values of all normalized histograms are concatenated to form an integrated edge histogram with 64 (8×8) bins.

4 The similarity measures of features

After the feature extraction process, the retrieval system calculates the similarity between the feature vectors of the query image and the previously computed vectors in the feature

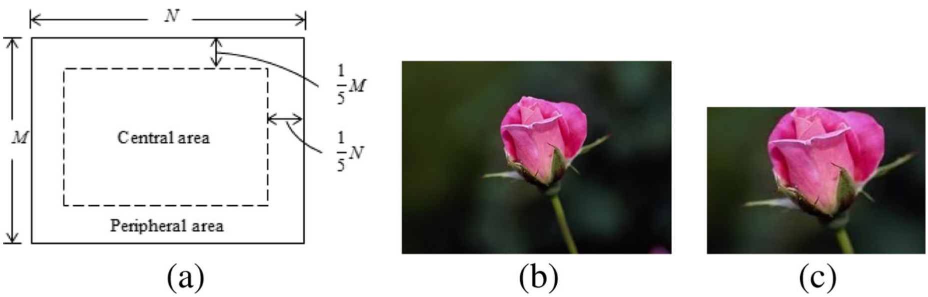


Fig. 5 **(a)** Image area division, **(b)** original image, and **(c)** cropped image

Table 1 Similarity measure comparisons (Corel dataset)

Similarity measure	Formula	Average precision	Average recall
Euclidean distance	$d(F^Q, F^I) = \sqrt{\sum_{i=1}^{n} (f_i^Q - f_i^I)^2}$	0.6538	0.1308
City block distance	$d(F^Q, F^I) = \sum_{i=1}^{n} \lvert f_i^Q - f_i^I \rvert$	0.7486	0.1497
χ^2-statistic [47]	$d(F^Q, F^I) = \sum_{i=1}^{n} \frac{(f_i^Q - m_i)^2}{m_i}$ where $m_i = \frac{f_i^Q + f_i^I}{2}$	0.7543	0.1509
d_1 distance [1]	$d(F^Q, F^I) = \sum_{i=1}^{n} \frac{\lvert f_i^Q - f_i^I \rvert}{\lvert 1 + f_i^Q + f_i^I \rvert}$	0.7559	0.1512

database. There are several distance formulas for measuring the similarity of features, so it is better to compare and select the one which demonstrates superior performance. In order to select a good similarity measure, we tested the retrieval effectiveness (described in Subsection 5.3) of the proposed system with respect to four different similarity measures: Euclidean, city block, χ^2-statistic [47], and d_1 distance [1]. Here, we used all images in the Corel dataset as query images. The average precision, which is defined as the mean of precisions of all individual queries, is calculated by evaluating the top 20 returned results. The average recall is calculated in the similar manner. Table 1 summarizes the results of these tests. We can see that d_1 distance yielded best retrieval effectiveness, and hence used as a similarity measure in our experiments.

Let $F_C^Q = [f_1^Q, f_2^Q, \ldots, f_{300}^Q]$ and $F_C^I = [f_1^I, f_2^I, \ldots, f_{300}^I]$, respectively, represent the color feature vector of the query image Q and a certain database image I. Then, the

(a)

(b)

Fig. 6 One sample image from each of the 10 categories of the **(a)** Corel dataset, and **(b)** MIR Flickr dataset color similarity distance $d_{color}(F_C^Q, F_C^I)$ between Q and I is calculated via the following equation [1]:

$$d_{color}(F_C^Q, F_C^I) = \sum_{i=1}^{300} \frac{\left|f_i^Q - f_i^I\right|}{\left|1 + f_i^Q + f_i^I\right|} \tag{1}$$

where f_i^Q and f_i^I denote the ith components of F_C^Q and F_C^I, respectively.

Considering texture feature vectors $F_T^Q = [g_1^Q, g_2^Q, \ldots, g_{54}^Q]$ and $F_T^I = [g_1^I, g_2^I, \ldots, g_{54}^I]$ of images Q and I, the texture similarity distance $d_{texture}(F_T^Q, F_T^I)$ between Q and I is formulated as the following:

$$d_{texture}(F_T^Q, F_T^I) = \sum_{i=1}^{54} \frac{\left|f_i^Q - f_i^I\right|}{\left|1 + f_i^Q + f_i^I\right|} \tag{2}$$

Considering shape feature vectors $F_S^Q = [t_1^Q, t_2^Q, \ldots, t_{64}^Q]$ and $F_S^I = [t_1^I, t_2^I, \ldots, t_{64}^I]$ of images Q and I, the shape similarity distance $d_{shape}(F_S^Q, F_S^I)$ between Q and I is determined as follows:

$$d_{shape}(F_S^Q, F_S^I) = \sum_{i=1}^{64} \frac{\left|f_i^Q - f_i^I\right|}{\left|1 + f_i^Q + f_i^I\right|} \tag{3}$$

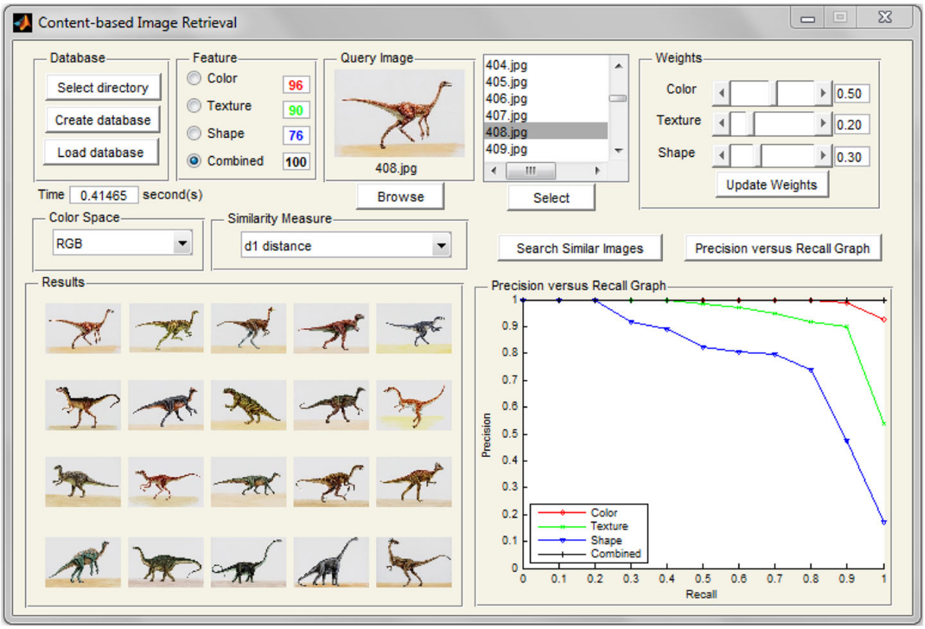


Fig. 7 The GUI of our implemented system showing the results of a search on a query image from the 'Dinosaurs' category of the Corel dataset. The query image is shown on *top* of the window and the thumbnails of the retrieved images are displayed in the *bottom-left* hand panel according to their decreasing similarity to the query image in raster scan order

4.1 Integration of color, texture, and shape features

Experience shows that effective CBIR system cannot be developed with only one type of visual feature. We need multiple heterogeneous features in order to increase the retrieval effectiveness. In such a case, different weights need to be assigned appropriately to different features, since different features have different discriminating capabilities. In our experiments, we integrate the similarity results of color-based retrieval, texture-based retrieval, and shape-based retrieval by linearly combining the associated similarity values. Accordingly, the overall similarity between Q and I is determined as follows:

$$d(F^Q, F^I) = w_C d_{color}(F_C^Q, F_C^I) + w_T d_{texture}(F_T^Q, F_T^I) + w_S d_{shape}(F_S^Q, F_S^I) \quad (4)$$

where w_C, w_T, and w_S are the weights assigned to the color-, texture-, and shape-based similarity values, respectively, and are subject to the condition $w_C + w_T + w_S = 1$. These weights are set by the user through the GUI. In the experiments, we have assigned the following weights: $w_C = 0.5$, $w_T = 0.2$, and $w_S = 0.3$.

Since similarity values for different features may vary within a wide range, therefore, before implementing Eq. 4, values $d_{color}(F_C^Q, F_C^I)$, $d_{texture}(F_T^Q, F_T^I)$, and $d_{shape}(F_S^Q, F_S^I)$ are normalized so that each value receives equal emphasis when the the overall similarity between two images is calculated. Normalization is done using the following formula [25]:

$$d_i^N = \frac{d_i - \min}{\max - \min}, \qquad i = 1, 2, \ldots, v \quad (5)$$

where d_i is the similarity value between the query image and the ith database image, min and max refer to the smallest and the biggest values in the sequence d_i, and v is the length of d_i.

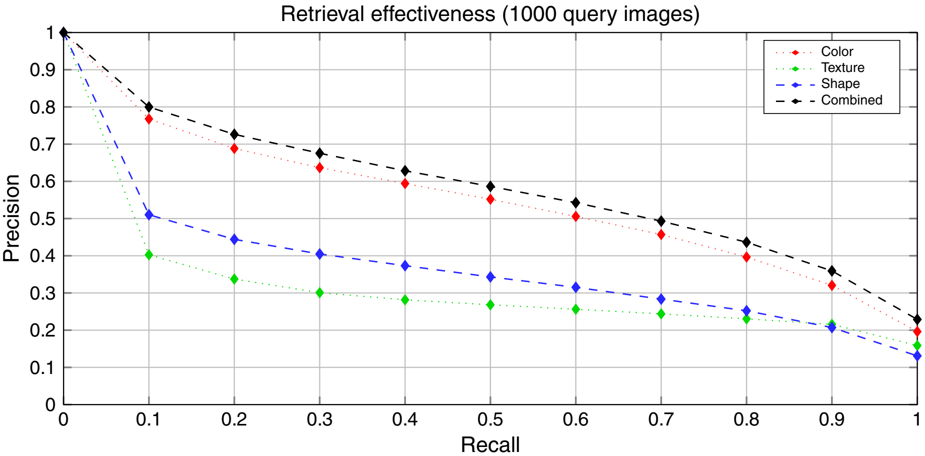


Fig. 8 Interpolated precision averages at 11 standard recall levels for the various methods mentioned before (Corel dataset)

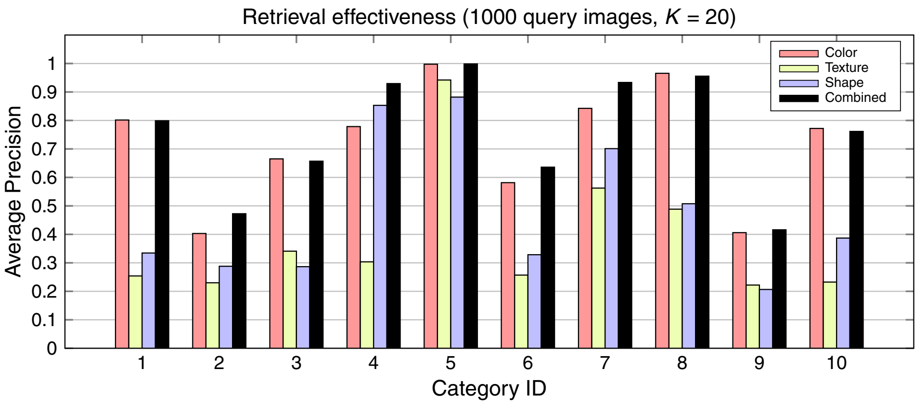
5 Experimental results

The performance of the CBIR system depends upon two main parameters [10, 42]. One is the retrieval effectiveness which focuses on the accuracy of the retrievals. The other is the retrieval efficiency which is concerned with the speed of the retrievals. In this section, we will demonstrate the performance of our proposed histogram-based approach through a number of experiments.

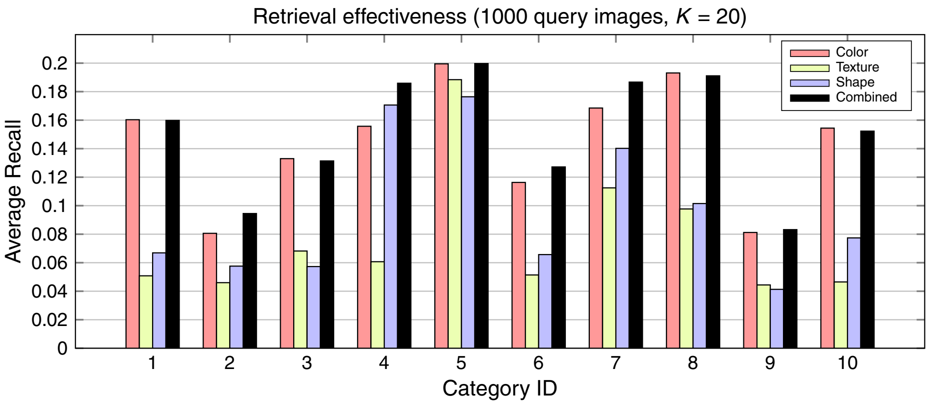
5.1 Image datasets and implementation environment

Two general-purpose image datasets, consisting of 1000 and 3000 images, respectively, are used to evaluate the performance of the proposed system.

The first image dataset used in this work is that of Wang et al. [64]. It is a subset of the Corel photo collection and is composed of 1000 color images, divided into 10 semantic



(a)



(b)

Fig. 9 The retrieval effectiveness for each category of the Corel dataset using the various methods mentioned before: **(a)** average precision, and **(b)** average recall categories, each containing 100 images. The categories are ‘African people and village’, ‘Beach’, ‘Buildings’, ‘Buses’, ‘Dinosaurs’, ‘Elephants’, ‘Flowers’, ‘Horses’, ‘Mountains and glaciers’, and ‘Food’, with corresponding category ID’s denoted by integers from 1 to 10, respectively. This category information availability is an advantage of this dataset, since it makes evaluation of retrieval results easier. Ideally, the goal is to retrieve images belonging to the same category as the query image. In this dataset, each image is stored in JPEG format with size 384 × 256 or 256 × 384. Figure 6a shows one sample image from each of the 10 categories of the Corel dataset.

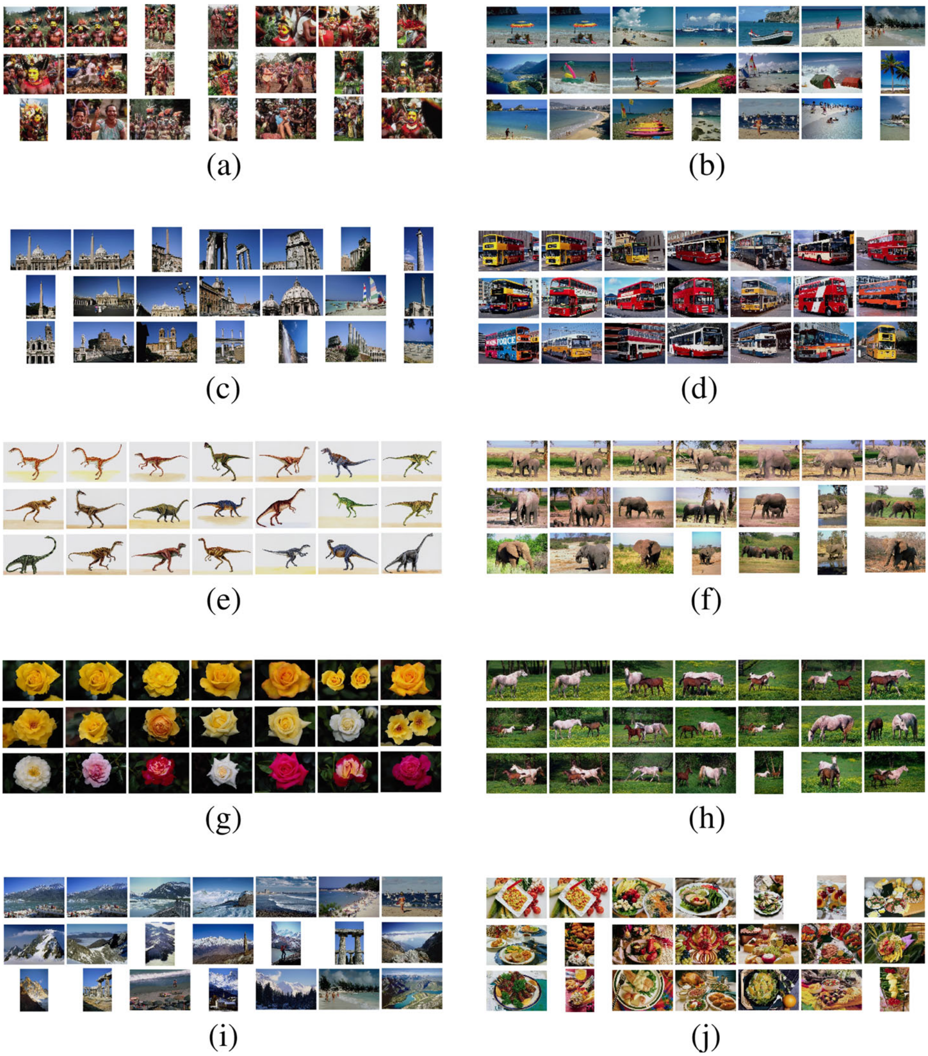


Fig. 10 Results for example query from: **(a)** ‘African people and village’, **(b)** ‘Beach’, **(c)** ‘Buildings’, **(d)** ‘Buses’, **(e)** ‘Dinosaurs’, **(f)** ‘Elephants’, **(g)** ‘Flowers’, **(h)** ‘Horses’, **(i)** ‘Mountains and glaciers’, and **(j)** ‘Food’ category of the Corel dataset using the proposed approach. The retrieved images are raster scan ordered by their similarities to the query image in the upper left corner

We also perform experiments over 3000 images from another image collection, called MIR Flickr [21]. This collection consists of 25000 images that were downloaded from the Flickr website. The color images are representative of a generic domain and are of high quality. MIR Flickr test set has been designed to address the four main requirements: representative of an area; accurate ground truth; freely redistributable; and standardized tests [21]. In the collection, there are 1386 tags which occur in at least 20 images. For our research purpose, we used a subset of this collection, formed by 10 image categories, each containing 300 images. The categories are 'Night', 'Clouds', 'Portrait', 'Dog', 'Sky', 'Flower', 'Trees', 'Sunset', 'People', and 'Lights'. For computational convenience, every image in the dataset is scaled to the size of 128×128 pixels using bilinear interpolation. In Fig. 6b, one sample image from each of the 10 categories of the MIR Flickr dataset is shown.

The proposed image retrieval system has been implemented on MATLAB on an Intel Core i3, 2.10 GHz processor with 4 GB RAM under the environment of Microsoft Windows 7.

5.2 User interface for proposed system

A user interface is a key component of the CBIR system, since the user-system interactions are carried out through it. It is used for query formulation, results presentation and query tuning. The user interface must be designed to let users easily select content-based properties, allow these properties to be combined with each other and with text or parametric data, and let users reformulate queries and generally navigate the database [11]. We build a simple interactive GUI with GUIDE (GUI Development Environment) tool in MATLAB (see Fig. 7). It incorporates all the key features to facilitate testing and evaluation of our multi-feature algorithm. The six panels of the GUI are explained briefly as follows:

1. The *Database panel* allows the entry of a directory containing images, which are processed to produce a feature database prior to query process.

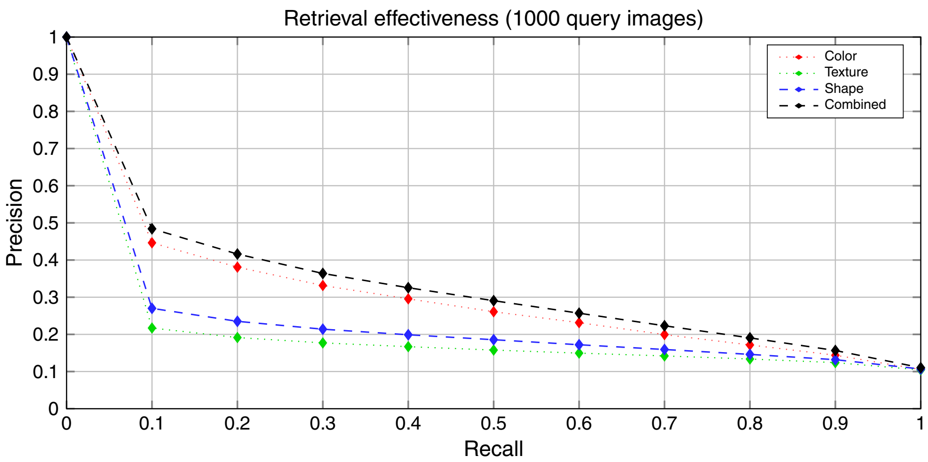


Fig. 11 Interpolated precision averages at 11 standard recall levels for the various methods mentioned before (MIR Flickr dataset)

2. For formulating a query, the user may select a random image from the list box to start with. Or, the user may load the query image by browsing through thumbnail images. The selected image is displayed in the *Query Image panel* positioned at top of the GUI.
3. Using the *Feature panel*, the user can perform selective retrieval by using just one of the three features (color, texture, and shape) or using a combination of them.
4. The *Weights panel* allows the user to assign weights to different features, depending on his/her interest in the query. For example, if a user is interested in color appearance of an image more than its texture and shape appearances, then the weight of color feature is set higher than texture and shape features. This provides flexibility in searching desired images.

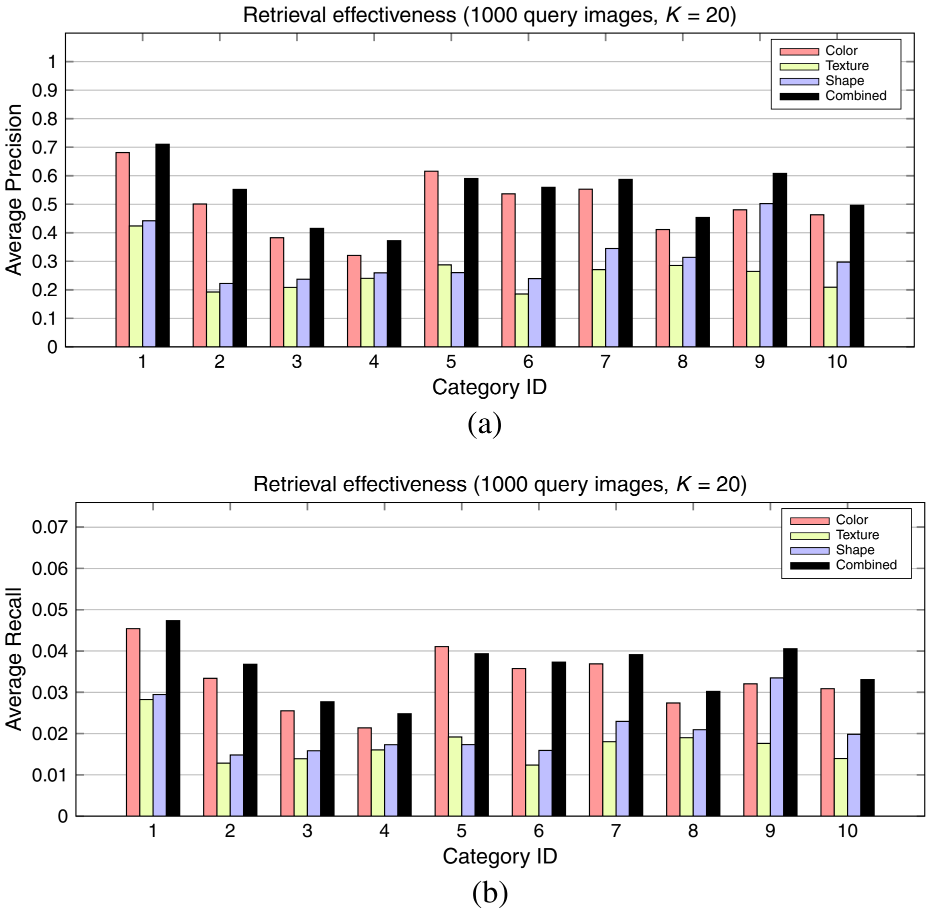


Fig. 12 The retrieval effectiveness for each category of the MIR Flickr dataset using the various methods mentioned before: **(a)** average precision, and **(b)** average recall

5. Once a query has been formulated, the system returns a set of results. These results are displayed in the *Results panel* in a 4×5 matrix according to their decreasing similarity to the query image in raster scan order.
6. The *Precision versus Recall Graph panel* is used to plot a precision versus recall graph for the query image. It helps the user to make a good inspection of the quality of the results; important for assigning weights to features.

In addition, the GUI provides multiple choices for color spaces and similarity measures. As a starting point, we select one set of system parameters and test the performance of the system under this set.

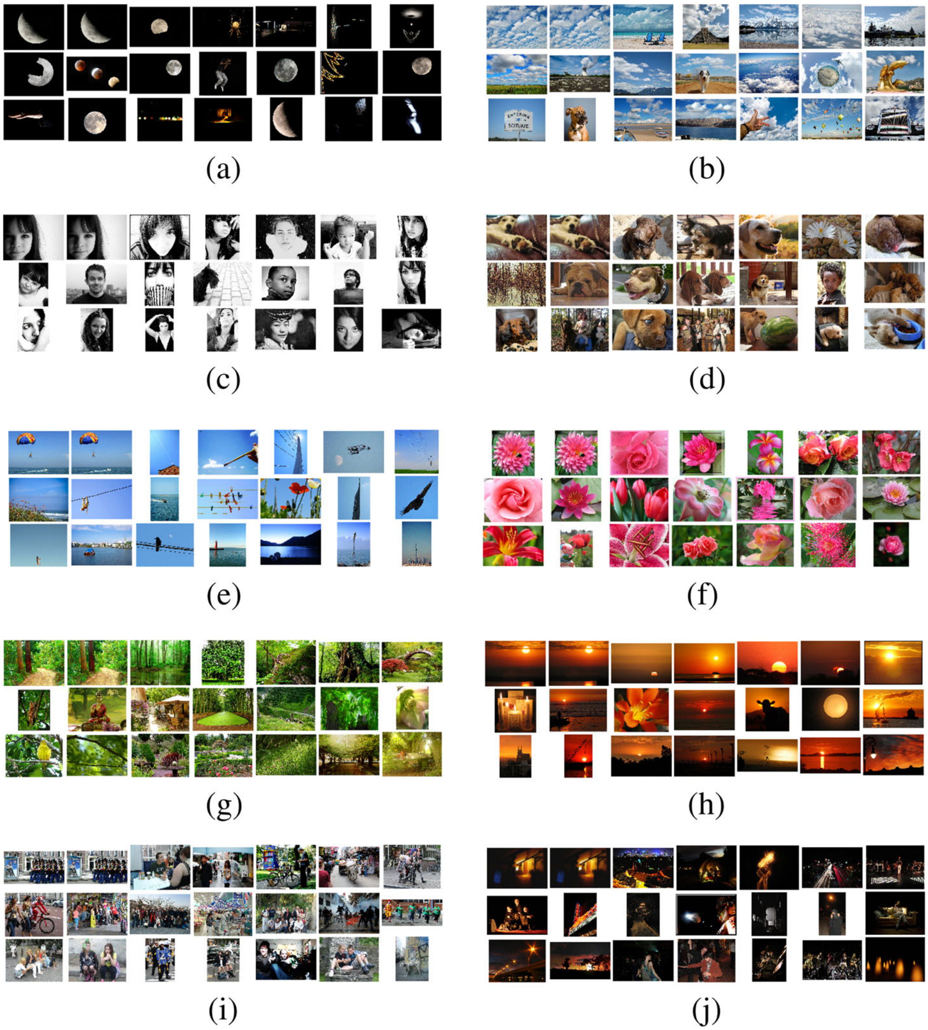


Fig. 13 Results for example query from: **(a)** 'Night', **(b)** 'Clouds', **(c)** 'Portrait', **(d)** 'Dog', **(e)** 'Sky', **(f)** 'Flower', **(g)** 'Trees', **(h)** 'Sunset', **(i)** 'People', and **(j)** 'Lights' category of the MIR Flickr dataset using the proposed approach. The retrieved images are raster scan ordered by their similarities to the query image in the upper left corner

5.3 Retrieval effectiveness measures

Two commonly used measures, namely precision (P) and recall (R) are used to evaluate the retrieval effectiveness of the proposed system. These are defined as [9]:

$$P = \frac{N_r}{K} \tag{6}$$

$$R = \frac{N_r}{N_t} \tag{7}$$

where K is the number of images retrieved by the system in response to the query image, N_r is the number of relevant images in the retrieved images, and N_t is the total number of relevant images available in the database. As indicated by the formulas, precision is the fraction of retrieved images that are relevant to the query image. In contrast, recall is the fraction of relevant images in the database that are retrieved in response to the query image. Generally, precision and recall values are plotted together in the form of precision versus recall graph. An ideal precision versus recall graph has precision = 1 for all values of recall, which indicates that the system retrieves all relevant images before any irrelevant ones. The closer the precision stays to 1, the better the effectiveness of the system.

5.4 Results

In the experiments, we test the proposed system for four cases: (1) solely using the color feature, (2) solely using the texture feature, (3) solely using the shape feature, and (4) using the combination of color, texture, and shape features. Here, we use all images in the Corel dataset as query images in turn. The retrieval response of the system is assessed with the precision versus recall graph. The individual precision values are interpolated to a set of 11 standard recall levels $(0, 0.1, 0.2, \ldots, 1)$ [39]. Figure 8 shows the average values of 1000 precision versus recall graphs. It clearly shows that integrating the results of color-, texture-, and shape-based queries provides better retrieval effectiveness than either of the individual feature based queries. This is obvious since single feature can capture only one aspect of image properties, and therefore, tends to give unsatisfactory results. However, it should be noted that while our proposed method is better on average, there are some kinds of images in the dataset for which one of the individual feature-based methods may be more suited. We address this concern by allowing the user to specify the weights of the included features through the GUI. Fine-tuning the feature weights can lead to more accurate results. We admit that fine-tuning requires human expertise, and therefore, to assist the user, a function is included in the GUI which plots precision versus recall graph for the query image. Also, the user has the option of performing selective retrieval by using just

Table 2 Retrieval efficiency of various histogram-based features for the images in Corel dataset

	Color histogram	Wavelet histogram	Edge histogram	Color histogram + Wavelet histogram + Edge histogram
Feature vector length	300	54	64	–
Indexing time (in seconds)	159.5204	46.4404	24.4123	–
Average response time (in second)	0.2314	0.0882	0.0758	0.4367

Table 3 Retrieval efficiency of various histogram-based features for the images in MIR Flickr dataset

	Color histogram	Wavelet histogram	Edge histogram	Color histogram + Wavelet histogram + Edge histogram
Feature vector length	300	54	64	–
Indexing time (in seconds)	112.4836	111.1453	72.6315	–
Average response time (in second)	0.2562	0.1746	0.2376	0.7566

one of the three features. The accuracy of these individual feature-based retrievals can help the user in determining the weights.

Figure 9a and b show the average precision and the average recall for l0 categories of Corel dataset with respect to the selected feature(s) (only color, only texture, only shape, and combination of color, texture, and shape). Looking at these Figures, there is a great amount of variations in effectiveness across different categories. For the images of ‘Dinosaurs’ that consist of a single object with clear background, the proposed approach using either individual or combined features produces near perfect results. Similar effectiveness is also observed on categories having prominent objects such as ‘Buses’, ‘Flowers’ and ‘Horses’. For heterogeneous images consisting of a variety of patterns and colors such as ‘African people and village’ and ‘Food’, the retrieval effectiveness is quite reasonable. Our method gives only a little worse performance for ‘Beach’ and ‘Mountains and glaciers’. The main reason for this discrepancy is that most images of these categories have a similar blue sky background occupying a large area. In this sense, these images were found to be similar because of the background, not the object themselves. This is a typical problem with global features and is an important research issue in CBIR. An obvious way to tackle this problem is to segment the image into regions and use local image descriptors. But, this is out of the scope of this paper.

Figure 10 shows ten examples of retrieval results using the proposed approach for Corel dataset. In Fig. 11, average values of 1000 precision versus recall graphs for MIR Flickr dataset (Here, we selected 100 images from each of the 10 categories, i.e., 1000 query images in total) are shown. Figure 12a and b show the average precision and the average recall for l0 categories of MIR Flickr dataset with respect to the selected feature(s) (only color, only texture, only shape, and combination of color, texture, and shape). In Fig. 13, ten examples of retrieval results using the proposed approach for MIR Flickr dataset are shown.

As stated earlier, retrieval efficiency is another parameter to measure the performance of the CBIR system. Efficiency is closely related with the storage requirements and the responsiveness of the system. We examine retrieval efficiency by measuring the *indexing time* (time taken to extract and store feature vectors from all images in the database) and the *response time*[1] (time taken by the system to respond to user’s query) of the proposed system. Indexing time mainly relies on the size of image database, the software and hardware conditions that the system runs on, and the feature vector length. To see how fast our proposed method is for CBIR, we calculate the average of response times of 10 randomly selected query images. Tables 2 and 3 show the retrieval efficiency of the various histogram-based features with respect to the Corel and the MIR Flickr datasets. It can be observed that the proposed method lends well to efficient indexing which leads to fast image retrieval.

[1]The *response time* is displayed in the GUI every time the user clicks the ‘*Search Similar Images*’ button.

Table 4 Comparison of average precision values from four methods (Corel dataset)

Category	Average precision ($K = 20$)			
	Lin et al. [29]	Subrahmanyam et al. [59]	Rao and Rao [48]	Proposed method
African people and village	0.6830	0.6975	0.7515	0.7990
Beach	0.5400	0.5425	0.5765	0.4725
Buildings	0.5615	0.6395	0.747	0.6570
Buses	0.8880	0.8965	0.943	0.9295
Dinosaurs	0.9925	0.9870	0.9895	0.9985
Elephants	0.6580	0.4880	0.5655	0.6360
Flowers	0.8910	0.9230	0.9545	0.9335
Horses	0.8025	0.8945	0.8665	0.9555
Mountains and glaciers	0.5215	0.4730	0.459	0.4160
Food	0.7325	0.7090	0.82	0.7615
Average	0.7270	0.725	0.7673	0.7559

5.5 Comparison with other methods

To see how good our proposed method is for CBIR, we compare the results of our method against the results reported by Lin et al. [29], Subrahmanyam et al. [59], and Rao and Rao [48]. The results of comparative methods are obtained from the original research work reported by the corresponding authors. To ensure a fair comparison against these methods, we used the same 1000 images from Corel collection, the same number of query images (i.e., each image in each category as a query image) and retrieved the same number of images, i.e., 20, to calculate the precision.

Table 4 reports the comparisons among the proposed method and the former methods in terms of average precision. As it can be seen in this Table, the average retrieval precision of our proposed method (0.7559) is more as compared to Lin et al. (0.7270), Subrahmanyam et al. (0.725), and close to Rao and Rao (0.7673). Thus, our proposed querying scheme is effective in retrieval quality, even though the image descriptors are simply derived from low level visual features, i.e., color distribution, texture, and edge characteristics.

6 Conclusion and future work

The main contribution of this paper is the selection of proper features that are complementary to each other so as to yield an improved retrieval performance and to combine chosen features effectively. More important, all features used in the proposed system, no matter color, texture, or shape, are represented in the simple form of histogram, yet leading to impressive results. The use of three normalized histograms, (i.e. color, wavelet, and edge histograms) provides a robust feature set and ensures that the retrieval system produces results which are highly relevant to the content of the query image, by taking into account the three distinct features of the image. Furthermore, our proposed system includes a friendly GUI, which offers options for feature selection and extraction, feature combination and weighting, and similarity measures.

The proposed system can retrieve images ranging from purely objects, such as an image of a dinosaur, a horse, an elephant, and so on, to images containing a mixture of structure, such as images of architecture, buildings, and mountains. Experiments conducted on Corel dataset showed that in four image categories, viz., ‘Buses’,‘Dinosaurs’, ‘Flowers’, and ‘Horses’, average precisions were more than 92 %, while with MIR Flickr dataset, six image categories, viz., ‘Night’, ‘Clouds’, ‘Sky’, ‘Flower’, ‘Trees’, and ‘People’ resulted in average precisions of more than 55 %. In addition, the average response times for Corel and MIR Flickr datasets, respectively, were less than 0.44 and 0.76 second. Hence, our system can be used in many application areas requiring not only high precision, but also speed. In our future research, we will consider local image descriptors and add new algorithms into our current system to extract more accurate semantics. We expect the retrieval effectiveness of our system to be further improved.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Funding This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

1. Agarwal M, Maheshwari R (2012) Á Trous gradient structure descriptor for content based image retrieval. International Journal of Multimedia Information Retrieval 1(2):129–138
2. Androutsos D, Plataniotis KN, Venetsanopoulos AN (1999) A novel vector-based approach to color image retrieval using a vector angular-based distance measure. Comput Vis Image Underst 75(1-2):46–58
3. Babu CR, Reddy ES, Rao BP (2015) Age group classification of facial images using rank based edge texture unit (retu). In: Information Systems Design and Intelligent Applications. Springer, pp 483–496
4. Carson C, Thomas M, Belongie S, Hellerstein JM, Malik J (1999) Blobworld: A system for region-based image indexing and retrieval. In: Visual Information and Information Systems. Springer, pp 509–517
5. Chun YD, Seo SY, Kim NC (2003) Image retrieval using BDIP and BVLC moments. IEEE Trans Circuits Syst Video Technol 13(9):951–957
6. Chun YD, Kim NC, Jang IH (2008) Content-based image retrieval using multiresolution color and texture features. IEEE Trans Multimedia 10(6):1073–1084
7. Cox IJ, Miller ML, Minka TP, Papathomas TV, Yianilos PN (2000) The bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Trans Image Process 9(1):20–37
8. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2):5:1–5:60
9. Deng Y, Manjunath B (1999) An efficient low-dimensional color indexing scheme for region-based image retrieval. In: 1999 IEEE International conference on acoustics, speech, and signal processing, vol 6. IEEE, pp 3017-3020
10. Faloutsos C, Barber R, Flickner M, Hafner J, Niblack W, Petkovic D, Equitz W (1994) Efficient and effective querying by image content. J Intell Inf Syst 3(3-4):231–262
11. Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D, Yanker P (1995) Query by image and video content: The QBIC system. Computer 28(9):23–32
12. Fonseca MJ, Ferreira A, Jorge JA (2006) Generic shape classification for retrieval. In: Graphics Recognition. Ten Years Review and Future Perspectives, vol 3926. Springer, pp 291–299
13. Gevers T, Smeulders AWM (2000) PicToSeek: Combining color and shape invariant features for image retrieval. IEEE Trans Image Process 9(1):102–119

14. Gudivada VN, Raghavan VV (1995) Content based image retrieval systems. Computer 28(9):18–22
15. Guldogan E, Gabbouj M (2009) Dynamic feature weights with relevance feedback in content-based image retrieval. In: 24th International Symposium on Computer and Information Sciences. IEEE, pp 56–59
16. Han J, Ma KK (2007) Rotation-invariant and scale-invariant gabor features for texture image retrieval. Image Vision Comput 25(9):1474–1481
17. Han JW, Guo L (2003) A shape-based image retrieval method using salient edges. Signal processing: Image communication 18(2):141–156
18. Haralick RM, Shanmugam K, Dinstein I (1973) Textural features for image classification. IEEE Trans Syst Man Cybern SMC 3(6):610–621
19. Hu MK (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8(2):179–187
20. Huang J, Kumar SR, Mitra M, Zhu WJ, Zabih R (1997) Image indexing using color correlograms. In: 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp 762–768
21. Huiskes MJ, Lew MS (2008) The MIR Flickr retrieval evaluation. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval. ACM, pp 39–43
22. Jain AK, Vailaya A (1996) Image retrieval using color and shape. Pattern Recogn 29(8):1233–1244
23. Jégou H, Douze M, Schmid C, Pérez P (2010) Aggregating local descriptors into a compact image representation. IEEE
24. Jeong S, Won CS, Gray RM (2004) Image retrieval using color histograms generated by gauss mixture vector quantization. Comput Vis Image Underst 94(1-3):44–66
25. Jiawei H, Kamber M (2006) Data mining: concepts and techniques, 2nd edn. Morgan Kaufmann, USA
26. Kokare M, Biswas PK, Chatterji BN (2007) Texture image retrieval using rotated wavelet filters. Pattern Recogn Lett 28(10):1240–1249
27. Li X (2003) Image retrieval based on perceptive weighted color blocks. Pattern Recogn Lett 24(12):1935–1941
28. Liapis S, Tziritas G (2004) Color and texture image retrieval using chromaticity histograms and wavelet frames. IEEE Trans Multimedia 6(5):676–686
29. Lin CH, Chen RT, Chan YK (2009) A smart content-based image retrieval system based on color and texture feature. Image Vision Comput 27(6):658–665
30. Liu F, Picard RW (1996) Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. IEEE Trans Pattern Anal Mach Intell 18(7):722–733
31. Liu Y, Zhang D, Lu G, Ma WY (2007) A survey of content-based image retrieval with high-level semantics. Pattern Recogn 40(1):262–282
32. Llorente A, Manmatha R, Rüger S (2010) Image retrieval using markov random fields and global image features. In: Proceedings of the ACM International Conference on Image and Video Retrieval. ACM, pp 243–250
33. Long F, Zhang H, Feng DD (2003) Fundamentals of content-based image retrieval. In: Multimedia Information Retrieval and Management. Springer, pp 1–26
34. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
35. Lu TC, Chang CC (2007) Color image retrieval technique based on color features and image bitmap. Inf Process Manag 43(2):461–472
36. Ma WY, Manjunath BS (1999) NeTra: A toolbox for navigating large image databases. Multimedia Syst 7(3):184–198
37. Manjunath BS, Ma WY (1996) Texture features for browsing and retrieval of image data. IEEE Trans Pattern Anal Mach Intell 18(8):837–842
38. Manjunath BS, Ohm JR, Vasudevan VV, Yamada A (2001) Color and texture descriptors. IEEE Trans Circuits Syst Video Technol 11(6):703–715
39. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge university press, New York
40. Matias Y, Vitter JS, Wang M (1998) Wavelet-based histograms for selectivity estimation. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. ACM, pp 448–459
41. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175
42. Ooi BC, Tan KL, Chua TS, Hsu W (1998) Fast image retrieval using color-spatial information. VLDB J 7(2):115–128

43. Pass G, Zabih R, Miller J (1996) Comparing images using color coherence vectors. In: Proceedings of the Fourth ACM International Conference on Multimedia. ACM, pp 65–73
44. Pentland A, Picard RW, Sclaroff S (1996) Photobook: Content-based manipulation of image databases. Int J Comput Vision 18(3):233–254
45. Persoon E, Fu KS (1977) Shape discrimination using fourier descriptors. IEEE Trans Syst Man Cybern 7(3):170–179
46. Pi MH, Tong CS, Choy SK, Zhang H (2006) A fast and effective model for wavelet subband histograms and its application in texture image retrieval. IEEE Trans Image Process 15(10):3078–3088
47. Puzicha J, Hofmann T, Buhmann JM (1997) Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In: 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp 267–272
48. Rao LK, Rao DV (2015) Local quantized extrema patterns for content-based natural and texture image retrieval. HCIS 5(1):1–24
49. Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) CNN features off-the-shelf: an astounding baseline for recognition. In: 2014 IEEE Conference on computer vision and pattern recognition workshops(CVPRW). IEEE, pp 512-519
50. Robinson GS (1977) Edge detection by compass gradient masks. Comput Graphics Image Process 6(5):492–501
51. Rui Y, Huang TS, Ortega M, Mehrotra S (1998) Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Trans Circuits Syst Video Technol 8(5):644–655
52. Rui Y, Huang TS, Chang SF (1999) Image retrieval: Current techniques, promising directions, and open issues. J Vis Commun Image R 10(1):39–62
53. Sikora T (2001) The mpeg-7 visual standard for content description an overview. IEEE Trans Circuits Syst Video Technol 11(6):696–702
54. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380
55. Smith AR (1978) Color gamut transform pairs. In: Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques, ACM, SIGGRAPH '78, pp 12–19
56. Smith J, Chang SF (1996a) Automated binary texture feature sets for image retrieval. In: 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol 4. IEEE, pp 2239–2242
57. Smith JR, Chang SF, 1996b VisualSEEk: a fully automated content-based image query system. In: Proceedings of the 4th ACM international conference on Multimedia. ACM, pp 87–98
58. Stricker MA, Orengo M (1995) Similarity of color images Proceedings of the SPIE 2420, storage and retrieval for image and video databases, vol III, pp 381-392
59. Subrahmanyam M, Wu QJ, Maheshwari R, Balasubramanian R (2013) Modified color motif co-occurrence matrix for image indexing and retrieval. Comput Electr Eng 39(3):762–774
60. Swain MJ, Ballard DH (1991) Color indexing. Int J Comput Vision 7(1):11–32
61. Veltkamp RC, Tanase M (2002) A survey of content-based image retrieval systems. In: Content-Based Image and Video Retrieval, vol 21. Springer, USA, pp 47–101
62. Wan X, Kuo CCJ (1996) Color distribution analysis and quantization for image retrieval Proceedings of the SPIE 2670, storage and retrieval for still image and video databases, vol IV, pp 8-16
63. Wang JZ, Wiederhold G, Firschein O, Wei SX (1998) Content-based image indexing and searching using daubechies' wavelets. Int J Digit Libr 1(4):311–328
64. Wang JZ, Li J, Wiederhold G (2001) SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. IEEE Trans Pattern Anal Mach Intell 23(9):947–963
65. Wang XY, Yu YJ, Yang HY (2011) An effective image retrieval scheme using color, texture and shape features. Comp Stand Inter 33(1):59–68
66. Wei CH, Li Y, Chau WY, Li CT (2009) Trademark image retrieval using synthetic features for describing global shape and interior structure. Pattern Recogn 42(3):386–394
67. Wu Y, Zhang A (2002) A feature re-weighting approach for relevance feedback in image retrieval. In: 2002 International Conference on Image Processing, vol 2. IEEE, pp II–581–II–584
68. Xu X, Lee DJ, Antani S, Long LR (2008) A spine X-ray image retrieval system using partial shape matching. IEEE Trans Inf Technol Biomed 12(1):100–108
69. Yoo HW, Park HS, Jang DS (2005) Expert system for color image retrieval. Expert Syst Appl 28(2):347–357
70. Zhang S, Tianm Q, Hua G, Huang Q, Li S (2009) Descriptive visual words and visual phrases for image applications. In: Proceedings of the 17th ACM international conference on Multimedia. ACM, pp 75–84

Amandeep Khokher received her B.Tech. and M.Tech. degrees in Electronics and Communication Engineering from Punjab Technical University, India, in 2004 and 2008 respectively. Currently, she is pursuing her Ph.D. degree in Electronics Engineering from I.K. Gujral Punjab Technical University, Kapurthala, India. She has more than 5 years of teaching experience. Her research interests include visual information retrieval, pattern recognition, and image processing.

Rajneesh Talwar did his Ph.D. in 2010 and M.Tech. in 2002 from Thapar University, Patiala, India. He is currently working as Principal, CGC Technical Campus, Jhanjeri, India. He has more than 15 years of professional experience of teaching, research, and administration. Dr. Talwar has a U.S. patent 'Fiber Optic Point Temperature Sensor' to his credit. He has been delivering expert talks on wireless communication, communication systems, etc., at various national level FDPs and invited talks.