Available online at www.sciencedirect.com
SCIENCE DIRECT
J. Vis. Commun. Image R. 17 (2006) 42–56
JOURNAL OF VISUAL Communication & IMAGE Representation
www.elsevier.com/locate/jvci

Fast indexing method for image retrieval using k nearest neighbors searches by principal axis analysis

Shyi-Chyi Cheng [a,*], Tian-Luu Wu [a,b]

[a] *Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, 1 University Road, Yenchao, Kaohsiung 824, Taiwan*
[b] *Department of Electronic Engineering, Yung Ta Institute of Technology and Commerce, Ping Tung 909, Taiwan*

Received 22 October 2003; accepted 11 August 2005
Available online 25 October 2005

Abstract

This paper presents a fast indexing scheme for content-based image retrieval based on the principal axis analysis. Image databases often represent the image objects as high-dimensional feature vectors and access them via the feature vectors and similarity measure. A similarity measure similar to the quadratic histogram distance measure is defined for this indexing method. The computational complexity of similarity measure in high-dimensional image database is very huge and hence the applications of image retrieval are restricted to certain areas. In this work, feature vectors in a given image are ordered by the principal axis analysis to speed up the similarity search in a high-dimensional image database using k nearest neighbor searches. To demonstrate the effectiveness of the proposed algorithm, we conducted extensive experiments and compared the performance with the IBM's query by image content (QBIC) method, Jain and Vailay's method, and the LPC-file method. The experimental results demonstrate that the proposed method outperforms the compared methods in retrieval accuracy and execution speed. The execution speed of the proposed method is much faster than that of QBIC method and it can achieve good results in terms of retrieval accuracy compared with Jain's method and QBIC method.
© 2005 Elsevier Inc. All rights reserved.

Keywords: High-dimensional image database; Content-based image retrieval; Principal axis analysis; Vector ordering; k-NN search

1. Introduction

As the advances of the Internet, the demand of storing multimedia information (such as text, image, audio, and video) has increased. With such databases comes the need for a richer set of search facilities that include keywords, sounds, examples, shape, color, texture, spatial structure, and motion. Traditionally, textual features such as filenames, captions, and keywords have been used to annotate and retrieve images. As it is applied to a large database, the use of keywords becomes not only cumbersome but also inadequate to represent the image content. Many content-based image retrieval systems have been proposed in the literature [1]. Although content-based image retrieval is extremely desirable in many applications, it suffers from several

* Corresponding author. Fax: +886 7 6011012.
E-mail address: csc@ccms.nkfust.edu.tw (S.-C. Cheng).

1047-3203/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2005.08.006

problems including image segmentation, extracting features from the images that capture the perceptual and semantic meanings, and matching the images in a database with a query image based on the extracted features. The problem of large-scale image databases with effective indexing and searching remains to be solved for content-based image retrieval (CBIR).

Existing general-purpose CBIR systems can roughly classified into three categories depending on the approach to extract features: histogram, color layout, and the region-based search [2]. These approaches have a common characteristic: the image objects are often represented as vectors of d numeric features and access them via the feature vectors and similarity measure. The feature vector dimensions of typical vector-based descriptors are quite large. The high dimensionality of the feature vectors leads to high computational complexity in distance calculation for similarity retrieval, and inefficiency in indexing and search. As an example, color histogram is the most simple and usual color descriptor in color-based image retrieval. However, color histograms result in large feature vectors that suffer from high index and retrieval cost [1,2]. The number of bins in a typical color histogram ranges from few tens to a few hundreds. To make the content-based image retrieval truly scalable to large size image databases, efficient multidimensional indexing techniques need to be explored.

In recent years, several methods have been proposed to solve these problems. The techniques can be roughly categorized into the following classes [3,4]: (1) the dimensionality reduction (DR) approach, (2) the multidimensional indexing approach, and (3) the filter-based approach. An indexing algorithm may be a combination of two or more of the mentioned classes. For example, one promising approach is to first perform dimension reduction and then use appropriate multidimensional indexing techniques.

Singular value decomposition (SVD) [5] and Hilbert curve fitting [6] are used to reduce the dimensionality of the feature vectors. However, these methods have their own drawbacks. In [5], SVD is performed on the quadratic matrix of correlations between the color histogram bins. The resulting eigenvectors are not related to the feature data, and may result in significant errors when lower-dimensional transformed feature vectors are used to approximate the original feature vectors. The results of Hilbert curve fitting depend on the data distributions. Points that are close to each other in the original feature space might be far apart on the Hilbert curve. The distances in the original space might not be preserved well in the curve approximation. In [7], dominant colors in a given region are clustered into a small number of representative colors and work as the color descriptor for image retrieval. A region in an image is essentially homogeneous and hence the number of colors to characterize the color information needs not be large.

The existing popular multidimensional indexing techniques include the bucketing algorithm, k-d tree, priority k-d tree, quad-tree, K-D-B tree, hB-tree, R-tree and its variants RC-tree, and R*-tree. In addition to the above approaches, clustering and neural nets, widely used in pattern recognition, are also promising indexing techniques. Very good reviews and comparisons of various indexing techniques in image retrieval can be found in [8]. Multidimensional indexing methods treat d-dimensional feature vectors as points in a d-dimensional vector space and the similarity measure can be viewed as a measure of distance within that space. The multidimensional indexing approach receives a challenge to access image databases: the performance of existing multidimensional indexing schemes degrades dramatically as the dimensionality increases [2,9].

The filter-based approach searches the nearest k neighbors of a query by filtering the vectors so that only a small portion of them must be visited. The percentage of vectors visited during a search depends on the strategy used to design the filter. As an example, the LPC-file [3] partitions the vector space into rectangular cells and these cells are used to generate bit-encoded approximates for each vector. The k-NN queries are processed by first scanning the entire approximation file and by filtering the vast majority of vectors from the search based only on these approximations. The drawbacks of the filter-based approach are: (1) the design of an approximation is not a trivial work and the precision of the approximation is not good while applying to image data of good locality; (2) additional information should be added to the approximate in order to enhance the filtering rate when the database is getting larger and larger.

The main ideas of the previous works to speed up the similarity searches in high-dimensional image databases are perhaps best summarized in two aspects: (1) the dimensionality of feature vectors should be carefully reduced in order to speed up the computation of similarity measurement; and (2) the irrelevant images in a large database with respect to a query image should be skip to visit from similarity searching. For the former case, traditional dimension-reduction techniques (i.e., SVD) can serve well except for their high computational

complexity. For the latter case, as mentioned above, many existing multidimensional data structures and filter-based methods can work for this purpose except for the problem of dimension curse in indexing very large image databases. In this work, we suggest a novel scheme to reduce the dimensions from high-dimensional feature vectors for similarity measurement by using the principal axis analysis. The proposed principal axis analysis is analytical, and hence, the computation of similarity measurement is fast. Following the same concept, the principal axis analysis can be used to design an efficient filter to approach the demand of a scalable image retrieval system. As compared to previous works, the amount of extra indexing data of the proposed filter is relatively small.

In this paper, an efficient indexing and searching method for image retrieval from large databases using the principal axis analysis is presented. The image objects are represented by high-dimensional feature vectors, which are obtained by scanning the segmented regions in a top-to-bottom and left-to-right fashion and characterizing these regions as their mean colors and shape parameters. The size of the feature vector corresponding to an input image depends on the number of regions used to represent the content of the image. For illustration convenience, the feature vector is treated as a feature histogram with each bin being a color vector or a shape vector in this paper.

In general, the quadratic feature histogram distance measure [10] is a precise way to compute the similarity between two images, in which one is a query image and the other is a database image. If the number of bins in representing a feature histogram is large enough, only a small size of neighborhood of a bin vector is needed to compute the quadratic feature histogram distance between two feature histograms. And hence, the problem of image retrieval based on the quadratic feature histogram distance measure could be solved by the k nearest neighbors (k-NN) searching techniques. In this work, features in a given image are ordered by the principal axis analysis to speed up the computation of the quadratic feature histogram distance. To demonstrate the effectiveness of the proposed algorithm, we conducted extensive experiments and compared the performance with the IBM's query by image content (QBIC) method [10], Jain and Vailay's method [11], and the LPC-file method [3]. The quadratic feature histogram distance is used in the QBIC method; Jain and Vailay presented a similarity measurement different from the quadratic feature histogram distance using hybrid color and shape features; and the LPC-file method is a filter-based method. The experimental results demonstrate that the proposed method outperforms all the methods in retrieval accuracy and execution speed. The execution speed of the proposed method is much faster than that of QBIC method and that of the LPC-file method, and it can achieve good results in terms of retrieval accuracy compared with Jain's method and QBIC method. Basically, the LPC-file method can be treated as the QBIC method with a filtering scheme. And hence, the retrieved accuracy of the LPC-file method is the same as that of the QBIC method.

The remainder of this paper is organized as follows. Section 2 describes the method to compute the similarity between two images by the quadratic feature histogram distance measure using k-NN searches. Section 3 presents the method to order vectors in a feature histogram by the principal axis analysis. Section 4 presents the indexing method and the proposed image retrieval strategy. Some experimental tests to illustrate the effectiveness of the proposed image retrieval method are shown in Section 5. Finally, conclusions are drawn in Section 6.

2. Similarity measure by quadratic color histogram distance using k-NN searches

Consider a database DB consisting of a large number of images, where each of which is represented as a d-dimensional feature sequence $F = \{\{f_i, p_i\},\ i = 1, \ldots, d\}$, where f_i and p_i are the ith feature and the corresponding percentage of pixels, respectively. Depending on the types of features used in the system, each f_i could be a scalar, a vector, or a set of vectors. For example, if we separate an image I into several regions, where each of which is represented by its mean color, then each f_i of F corresponding to a region i of I is a color vector. The Euclidean distance is commonly used as a dissimilarity measure of two images.

Fig. 1 shows an example to represent an image as a feature histogram by the following process: (1) separate the image into a set of disjoint regions by a segmentation algorithm [12]; (2) scan the regions in the left-to-right and top-to-down fashion to construct a region sequence; and (3) construct the histogram by representing each region as its characteristic values such as the mean color. In this work, two kinds of histogram, namely color histogram and shape histogram, are used to represent the content of an image. The color histogram is

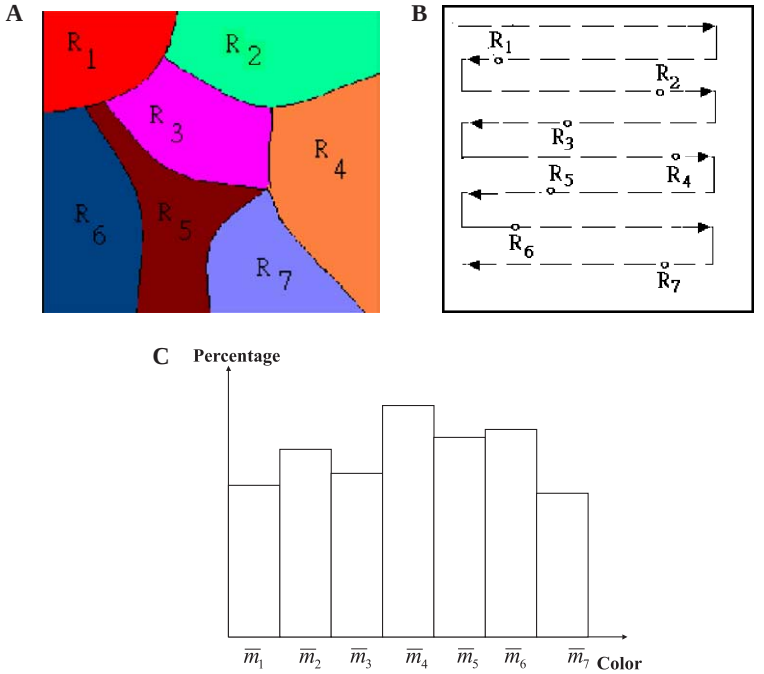

Fig. 1. An example of representing an image by a color histogram: (A) separate the image into a set of disjoint regions by a segmentation algorithm [12]; (B) scan the regions in the left-to-right and top-to-down fashion to construct a region sequence (R_1, R_2, R_3, R_4, R_5, R_6, R_7); (C) the color histogram of the image, where $\bar{m}_i,\ i = 1, \ldots, 7$ is the mean color of R_i.

obtained by representing each region as its mean color, on the other hand, the bin of each shape histogram is defined as the shape vector of the tri-stimulus values of the region in question: perch ratio, perimeter, and density. The perch ratio is defined as the ratio of the long axis l_l and the shortest axis l_s of a region and computed by

$$\mathrm{PR} = \frac{l_s}{l_l}, \quad 0 < \mathrm{PR} \leqslant 1. \tag{1}$$

The perimeter N_i reveals the size of region i and can be easily computed by

$$N_i = |E_i|, \tag{2}$$

where E_i is the set of edge pixels of region i. The region density is used to measure the diversity of a region. Given a region R, the region density D_R can be computed by

$$D_R = \sum_{p \in R} \sum_{q \in \mathrm{neighbors}(p)} L_{ab(p,q)} \Big/ \sum_{p \in R} 1, \tag{3}$$

where p is a pixel of R and q is one of the neighboring pixels of p. The function $L_{ab}(p,q)$ in Eq. (3) return 1 if both p and q belong to the same region, otherwise it returns 0. Note that the values of shape features should be normalized to be within 0 and 1 before applying to retrieve images. Note that the method to measure the similarity between two images on the basis of shape feature is the same as the one using color feature, except for replacing the color-feature vectors with the shape-feature vectors.

Let $S = \{\{s_i, p_i\},\ i = 1, \ldots, d_1\}$ and $T = \{\{t_i, q_i\},\ i = 1, \ldots, d_2\}$ be two feature histograms. Then the distance between S and T based on the concept of the QBIC method [7] is computed as

$$D^2(S,T) = \sum_{i=1}^{d_1} p_i^2 + \sum_{i=1}^{d_2} q_j^2 - \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} 2a_{i,j} p_i q_j, \tag{4}$$

where $a_{i,j}$ is the similarity coefficient between features s_i and t_j,

$$a_{i,j} = 1 - \frac{d_{i,j}}{d_{\max}}, \tag{5}$$

where $d_{i,j}$ is the Euclidean distance between feature values s_i and t_j and $d_{\max}$ is the maximum distance between any two feature values. This metric gives the weighted length of the difference vector between S and T, weighted by $a_{i,j}$, $i = 1, \ldots, d_1$; $j = 1, \ldots, d_2$, which accounts for the perceptual distance between different pairs of features.

The computational complexity of Eq. (4) is very high, especially when the bin numbers of histograms are large. Fortunately, if the number of bins in representing feature histograms is large enough, only a small size of neighborhood of a bin vector is needed to compute the quadratic feature histogram distance between two feature histograms. And hence, the problem of image retrieval on the basis of feature histograms could be solved by the k-NN searching techniques. Moreover, the value of a_{ij} is set to zero if the jth bin vector of the database histogram is not one of the k nearest neighbors of the ith bin vector of the query histogram. The bin vectors of a database histogram outside the k nearest neighbors of a query vector are not considered similar. Based on the k-NN searches, Eq. (5) can be re-written as

$$a_{i,j} = \begin{cases} 1 - \frac{d_{i,j}}{d_{\max}} & \text{if } s_i \in k\text{-NN}(t_j) \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where k-NN(t_j) returns the set of the k-NN from the histogram S. In this work, a fast k-NN searching procedure is proposed by the principal axis analysis to reduce the computational burden of Eq. (4). The idea underlying Eq. (6) is that the bin vectors in S outside the k nearest neighbors of t_j are considered being less contributed to compute the value of similarity between the two histograms S and T due to the fact that the corresponding similarity coefficients are small. The actual problem is the histograms of two images taken from the same image with different conditions (i.e., lighting condition) are slightly different. Some pixels of a histogram bin might locate on the neighboring bins of the other histogram in terms of feature values, and hence, to compute the similarity coefficients by Eq. (6) in a sense eliminates such kind of noise.

3. Vector ordering using principal axis analysis

Finding out the vector order statistics of feature vectors is a feasible step toward fast vector similarity searches. There has been a number of ways proposed to perform multivariate data ordering that are generally classified into marginal ordering (M-ordering), reduced or aggregate ordering (R-ordering), partial ordering (P-ordering), and conditional ordering (C-ordering) [13]. In M-ordering, the multivariate samples are ordered along each one of the multidimensions independently. In R-ordering, each vector is reduced to a scalar value according to a distance criterion. The vectors are then arranged in ascending order of magnitude of the associated metric values. In P-ordering, the objective is to partition the data into groups or sets of vectors, such that the groups can be distinguished with respect to order, rank, or extremeness. In C-ordering, the vectors are ordered conditional on one of the marginal sets of observations. This has the disadvantage in feature vector processing that only the information in one component is used.

From the above we can conclude that R-ordering is more appropriate for feature vector ordering than the other vector ordering methods. The actual problem in R-ordering is how to reduce such vectors to scalar values so that a feature vector can be discriminated easily. An optimum approach for R-ordering is to project vectors on the principal axis, which is defined as the line of the least moment of inertia. The arrangement of the projection scores $p_{(i)}s$ in ascending order ($p_{(1)} \leqslant p_{(2)} \leqslant \cdots \leqslant p_{(N)}$), we associate the same ordering to vectors $\vec{v}^{(i)}s$

$$\vec{v}^{(1)} \leqslant \vec{v}^{(2)} \leqslant \cdots \leqslant \vec{v}^{(N)}, \tag{7}$$

where N is the total number of vectors in the test histogram.

The principal axis can be conveniently represented in terms of moments. In the case of three-dimensional feature space S^3, the principal axis L can be represented as

$$\frac{x}{\cos\alpha} = \frac{y}{\cos\beta} = \frac{z}{\cos\gamma}, \tag{8}$$

where α, β, and γ are the angles between the principal axis L and the axes x, y, and z, respectively. $\cos\alpha$, $\cos\beta$, and $\cos\gamma$ are the three direction numbers of L and satisfy the following relationship:

$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1. \tag{9}$$

Let $\vec{z} = (\cos\alpha, \cos\beta, \cos\gamma)$ be a vector aligned with L. Then, as shown in Fig. 2, the distance from a feature vector $\vec{c}$ to L is given by

$$d = \frac{\|\vec{z} \times (\vec{c} - \bar{c})\|}{\|\vec{z}\|}, \tag{10}$$

where $\bar{c}$ is the centroid of the feature space and is defined as

$$(\bar{x}, \bar{y}, \bar{z}) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i,\ \frac{1}{N}\sum_{i=1}^{N} y_i,\ \frac{1}{N}\sum_{i=1}^{N} z_i \right), \tag{11}$$

where x_i, y_i, and z_i are the tri-stimulus values of the ith vector. For processing convenience, we subtract the centroid from all the feature vectors in order to translate the origin of the feature space to the centroid.

To find the direction numbers of the three-dimensional principal axis L, take the origin at the centroid; then the moment of inertia of the points in S^3 about the line L is

$$I(\alpha, \beta, \gamma) = \sum\sum\sum_{(x,y,z)\in S^3} [(z\cos\beta - y\cos\gamma)^2 + (z\cos\alpha - x\cos\gamma)^2 + (y\cos\alpha - x\cos\beta)^2]. \tag{12}$$

Differentiating this with respect to $\cos\alpha$, $\cos\beta$, and $\cos\gamma$, and equating to zero gives

$$2\sum\sum\sum_{(x,y,z)\in S^3} z(z\cos\alpha - x\cos\gamma) + y(y\cos\alpha - x\cos\beta) = 0$$

$$2\sum\sum\sum_{(x,y,z)\in S^3} z(z\cos\beta - y\cos\gamma) - x(y\cos\alpha - x\cos\beta) = 0.$$

$$2\sum\sum\sum_{(x,y,z)\in S^3} -y(z\cos\beta - y\cos\gamma) - x(z\cos\alpha - x\cos\gamma) = 0$$

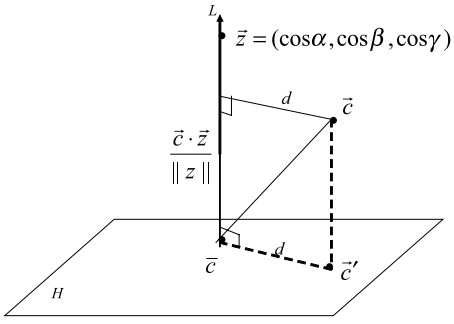

Fig. 2. The distance from a vector to a line L in the three-dimensional space S^3.

And hence,

$$\begin{aligned}
&(m_{0,2,0}+m_{0,0,2})\cos\alpha - m_{1,1,0}\cos\beta - m_{1,0,1}\cos\gamma = 0,\\
&-m_{1,1,0}\cos\alpha + (m_{2,0,0}+m_{0,0,2})\cos\beta - m_{0,1,1}\cos\gamma = 0,\\
&-m_{1,0,1}\cos\alpha - m_{0,1,1}\cos\beta + (m_{2,0,0}+m_{0,2,0})\cos\gamma = 0,
\end{aligned} \tag{13}$$

where $m_{s,t,u}$ is a 3-D moment given by

$$m_{s,t,u} = \sum\sum\sum_{(x,y,z)inS^3} x^s y^t z^u. \tag{14}$$

To solve the set of linear system in (13), we get

$$\frac{\cos\alpha}{\cos\beta} = \frac{(m_{2,0,0}+m_{0,0,2})m_{1,0,1} + m_{1,1,0}m_{0,1,1}}{(m_{0,2,0}+m_{0,0,2})m_{0,1,1} + m_{1,1,0}m_{1,0,1}} = k_1, \tag{15}$$

$$\frac{\cos\beta}{\cos\gamma} = \frac{(m_{2,0,0}+m_{0,2,0})m_{1,1,0} + m_{1,0,1}m_{0,1,1}}{(m_{2,0,0}+m_{0,0,2})m_{1,0,1} + m_{1,1,0}m_{0,1,1}} = k_2. \tag{16}$$

The values of k_1 and k_2 can be obtained directly from the values of 3-D moments which could be computed in advance according to (14). Combine Eqs. (9), (15), and (16), it is simple to obtain

$$(\cos\alpha, \cos\beta, \cos\gamma) = \left(\frac{k_1}{\sqrt{1+k_1^2+k_2^2}}, \frac{1}{\sqrt{1+k_1^2+k_2^2}}, \frac{k_2}{\sqrt{1+k_1^2+k_2^2}}\right). \tag{17}$$

Note that we do not need to compute the values of the three angles α, β, and γ to define the principal axis. Once the principal axis L is obtained, for each feature vector $\vec{c} = (x, y, z)$ we can project it onto L to compute the projection score of the vector by the following equation:

$$p_{\vec{c}} = \|\vec{c}\| \times \frac{\vec{c}\cdot\vec{z}}{\|\vec{c}\| \times \|\vec{z}\|} = x\cos\alpha + y\cos\beta + z\cos\gamma. \tag{18}$$

Although the discussion of finding the principal axis is limited to a three-dimensional space in this section, the same analysis process can be extended to find the principal axis in higher-dimensional space.

4. Proposed indexing method by principal axis analysis

The proposed indexing scheme takes a filter-based approach, which skips irrelevant vectors based on the projection scores on the principal axis for a query process. The design goal of the proposed indexing method is to maximize the filtering power of the filter-based method by using minimal information. Without loss of generality, the feature histogram is supposed to be a color histogram for the following discussion.

Given a database color histogram $F = \{\{c_i, p_i\}, i = 1, \ldots, n\}$, where c_i and p_i are the ith bin color and the corresponding percentage of pixels of the vector, respectively. The bin vectors can be considered as points in a three-dimensional space. Let L be the principal axis of the space. To speed up the computation of the quadratic color histogram distance between two input histograms, the bin color $\vec{c}$ is represented by the approximation

$$\vec{a} = (x_{\vec{c}}, |\vec{c}'|), \tag{19}$$

where $x_{\vec{c}}$ is the projection score of the vector projected on the line L and $|\vec{c}'|$ is the norm of the projection of $\vec{c}$ on the plane H which is a three-dimensional plane perpendicular to L, shown in Fig. 3. The vector $\vec{c}'$ is simple to obtain by

$$\vec{c}' = \vec{c} - (\vec{c}\cdot\vec{z})\vec{z}. \tag{20}$$

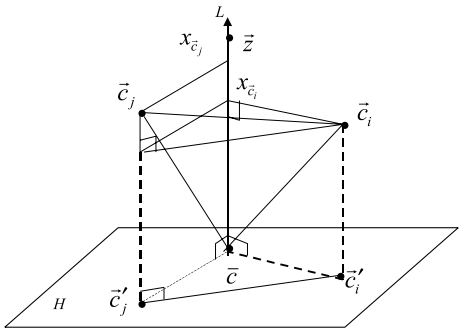

Fig. 3. Projection on a hyper-plane H, perpendicular to the principal axis L.

It is simple to prove that

$$|\vec{c}_i'|^2 = |\vec{c}_i|^2 - x_{\vec{c}_i}^2, \tag{21}$$

$$|\vec{c}_i' - \vec{c}_j'|^2 = |\vec{c}_i - \vec{c}_j|^2 - (x_{\vec{c}_i} - x_{\vec{c}_j})^2. \tag{22}$$

Given a color vector q of a query histogram, the bounds on the distance between the query vector and a bin vector p of a database histogram can be derived according to the values of projection scores to restrict the search space during the k-NN search. The bounds reflect the discrimination power of the indexing structure.

Based on the proposed approximation scheme and according to Eq. (22), the distance between the two vectors p and q in Fig. 4 can be obtained. It is simple to prove that $|p' - q'|$ can be bound as

$$\begin{cases} |q'| - |p'| \leqslant |p' - q'| \leqslant 2|p'| - |q'| & \text{if } |q'| > |p'| \\ |p'| - |q'| \leqslant |p' - q'| \leqslant |p'| + |q'| & \text{otherwise} \end{cases} \tag{23}$$

Combine Eqs. (22) and (23), we can compute the lower bound $|p' - q'|_{\min}$ and the upper bound $|p' - q'|_{\max}$ guaranteeing $|p - q|_{\min} \leqslant |p - q| \leqslant |p - q|_{\max}$. That is

$$|p - q|_{\max}^2 = \begin{cases} (x_p - x_q)^2 + (2|p'| - |q'|)^2 & \text{if } |q'| > |p'| \\ (x_p - x_q)^2 + (|p'| + |q'|)^2 & \text{otherwise} \end{cases} \tag{24}$$

and

$$|p - q|_{\min}^2 = (x_p - x_q)^2 + (|q'| - |p'|)^2. \tag{25}$$

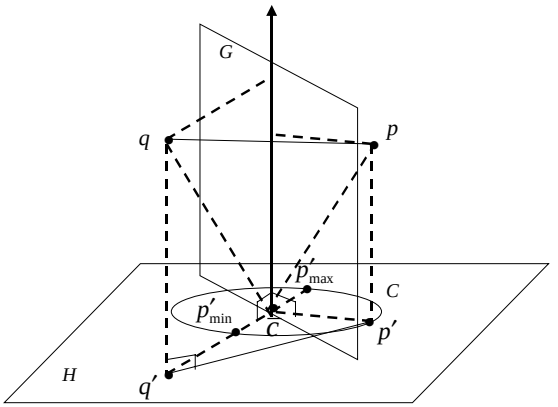

Fig. 4. Lower bound $|p - q|_{\min}$ and upper bound $|p - q|_{\max}$ for $|p - q|$ guaranteeing $|p - q|_{\min} \leqslant |p - q| \leqslant |p - q|_{\max}$.

Note that the values of $|p'|$ and $|q'|$ would be similar and small due to the fact that the principal axes are obtained by minimizing the moment of inertia, and hence the discrimination power of the proposed approximation scheme is excellent.

To complete the discussion, the proposed search algorithm based on the indexing structure is summarized below.

Algorithm Proposed k-NN search algorithm by principal axis analysis (K-NNPAA).

Input: A query color vector q and a number k which represents the desired number of nearest neighbors of q to output.

Output: A set of vectors corresponding to k nearest neighbors of q.

Method:

(1) Let S denote the collection of bin colors of a database histogram, where each of which is represented by an approximation. S is sorted in advance in ascending order in terms of the values of the approximations.

(2) Project q onto the projection axis and form the approximation $a_q = (x - q, |q'_{\mathrm{H}}|)$ using Eqs. (18) and (21).

(3) Set the values of thresholds ε_i, $i = 1, \ldots, k$ to be the maximum Integer.

(4) Let Q be a min-heap consists of the candidate vectors and empty initially. The root of Q contains the vector, which is the nearest neighbor of q encountered so far.

(5) Let the parameter j indicate the jth nearest neighbor to be found currently and initialize j as one.

(6) While $j \leqslant k$ or S is not empty do the following steps:

(6.1) Find the nearest vector p from S according to the values of the approximation of q by the binary search method. Compute $|p - q|_{\min}$ using Eq. (25).

(6.2) $S = S - \{p\}$.

(6.3) If $\varepsilon_j > |p - q|_{\min}$, then do following steps:

(6.3.1) compute $|p - q|$ according to their real vectors and insert p into Q;

(6.3.2) if $\varepsilon_j > |p - q|$, then, $\varepsilon_j = |p - q|$ and go to Step (6);

(6.4) Remove the root of Q and add it into the result set R.

(6.5) Let r be the new root of Q, increase j by 1 and $\varepsilon_j = |r - q|$.

(7) Output the vectors in R.

We need not check the remainder of the set S if Step (7) of the above algorithm is executed. The value of ε_k in the above algorithm denotes the minimal distance of the kth nearest neighbor of the query vector q. For each vector p in the remainder of S, we have $|p - q| \geqslant |p - q|_{\min} \geqslant \varepsilon_k$. And hence the vector p is impossible to be the one of the k nearest neighbors of q. Only a fraction of the bin vectors need to be visited for processing a query vector, and hence the performance of the proposed method is excellent as compared with other exhaustive nearest-neighbor search algorithms.

Applying the K-NNPAA procedure on each bin vector of a query histogram to obtain the values of similarity coefficient a_{ij}, defined in Eq. (6), the distance between the query histogram and the database histogram is obtained according to (4). The value of a_{ij} is set to zero if the jth bin color of the database histogram is not one of the k nearest neighbors of the ith color vector of the query histogram.

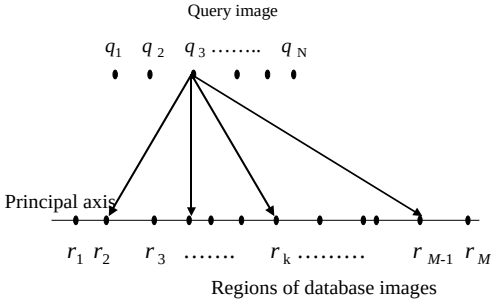

Fig. 5. The extended indexing structure.

The process to calculate the similarity on the basis of shape feature is the same as the one using color feature, except for replacing the color-feature vectors with the shape-feature vectors. The perch ratio, the perimeter and the density of a region, defined in Eqs. (1)–(3), respectively, are used to construct the three-dimensional shape-feature vector for indexing the region. Combine the color and shape features, the hybrid dissimilarity measure can be defined as

$$D^2_{\text{Total}} = w_c \cdot D^2_c(Q,I) + w_s \cdot D^2_s(Q,I), \tag{26}$$

where D^2_{Total} is the value of the hybrid distance between the query image Q and the database image I, $D^2_c(Q,I)$, and $D^2_s(Q,I)$ are the distances obtained by Eq. (4) on the basis of color and shape features, respectively, and w_c and w_s represent the weighting of $D^2_c(Q,I)$ and $D^2_s(Q,I)$, respectively. We should also set $w_c + w_s = 1$.

Basically, the proposed indexing method is a mechanism to speed up the computation of the quadratic feature distance between two high-dimensional histograms. The actual problem of the indexing method is we still need compare each database histogram with the query histogram one by one. That is the performance of the proposed system linearly depends on the size of the database. A further extension of the proposed indexing structure is necessary with the aim of designing a truly scalable CBIR system, whose performance will depend on the number of relevant images in the database for a query but not on the size of the database.

Table 1
The results of the average number of vector distance computing by applying the query image 'sunset' to the test database with and without using the proposed k-NN search algorithm

Histogram size	Proposed k-NN method (A)				Exhaustive search method (B)				Ratio (B/A)			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
64	27.57	29.94	28.20	30.30	64	3*64	5*64	7*64	2.31	6.41	11.35	14.79
128	36.52	40.29	38.25	40.99	128	3*128	5*128	7*128	3.50	9.53	16.73	21.86
256	36.38	38.84	43.84	47.01	256	3*256	5*256	7*256	7.04	19.77	29.20	38.12
512	18.24	20.77	27.23	30.57	512	3*512	5*512	7*512	28.07	73.95	94.01	117.2
1024	8.81	13.44	19.04	21.91	1024	3*1024	5*1024	7*1024	116.2	228.5	268.9	327.2

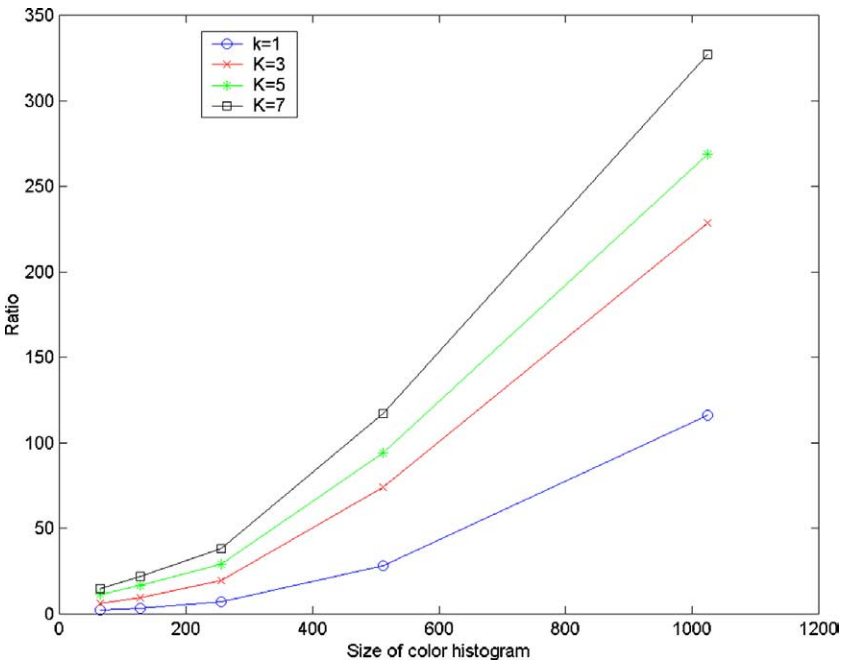

Fig. 6. The ratio of the number of vector distances computing between the exhaustive and proposed k-NN search algorithms under the situation of different values of k.

As shown in Fig. 5, all the database regions are collected as a region corpus C and their feature vectors are projected onto the principal axis of the corpus. The database indexing structure will then be built on the basis of the projection values. For each bin vector q of a query histogram, perform the K-NNPAA procedure on the database index file to find the k nearest regions of q in the region corpus C. Note that the regions belong to the set R_{q_i} consists of the nearest neighbors of the ith vector of the query histogram might come from different database images. A database image is considered similar by the decision rule: at least some of its regions belong to the region collection $R_q = \cup_{i=1}^{N} R_{q_i}$. That is a database image is significant if the number of its regions, which are members of R_q, is larger than a threshold τ. In this implementation, the value of τ is set to be 1. Finally, those significant images form a candidate set whose member is compared with the query image using the K-NNPAA procedure to compute the final quadratic feature vector distance.

5. Experimental results

In order to evaluate the proposed approach, a series of experiments were conducted on an Intel PENTIUM-III 500 MHz and four databases of 627, 1866, 8380, and 10456 natural images were used. Each image in the database is first tailored to the size of 256×256 for testing the retrieval approach.

Table 2
Performance comparison in terms of the average retrieval time (in seconds) for the proposed method, exhaustive search method, and LPC-file method [3] by applying a query image to an image database. The number of bits to encode each dimension of the LPC-file method is 6

Histogram size	Proposed k-NN method				Exhaustive search method				LPC-file
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	
64	0.0051	0.0121	0.0205	0.0269	0.0047	0.0141	0.0235	0.0329	0.0115
128	0.0077	0.0139	0.0224	0.0291	0.0073	0.0219	0.0365	0.0511	0.0315
256	0.0101	0.0172	0.0263	0.0343	0.0134	0.0402	0.0670	0.0938	0.0625
512	0.0113	0.0188	0.0292	0.0384	0.0231	0.0693	0.1155	0.1617	0.1134
1024	0.0132	0.0221	0.0323	0.0424	0.0773	0.2319	0.3865	0.5421	0.2453

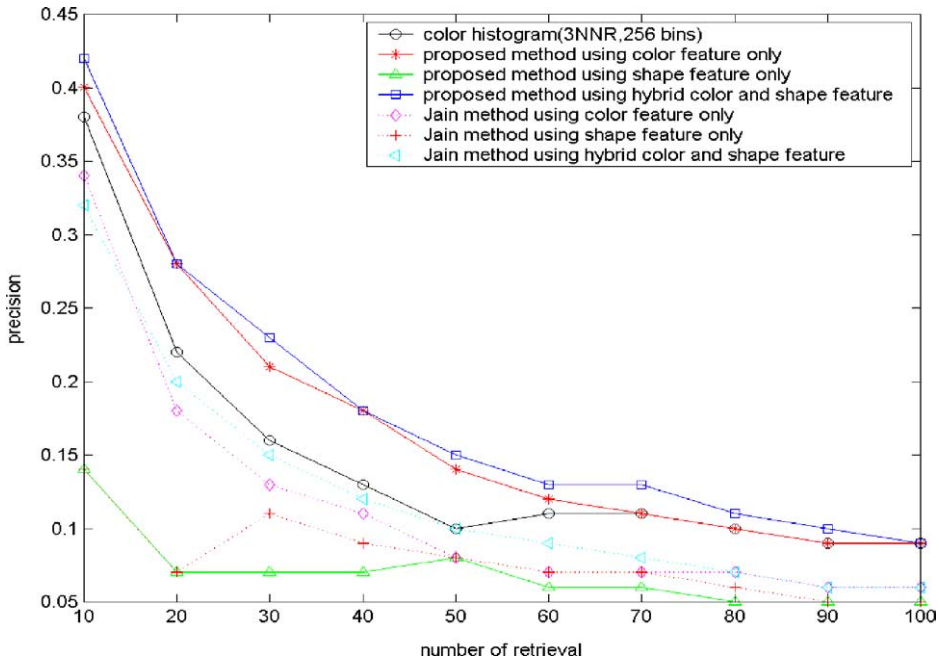

Fig. 7. Average precision versus number of retrievals.

The performance of the proposed image retrieval method is evaluated in terms of execution speed and retrieval accuracy. Although the two metrics are conflicted in some systems, both of them are important for a method to be practical. Table 1 shows the results of the average number of vector distance computing by applying the query image 'sunset' to the test databases with and without using the proposed k-NN search algorithm. Fig. 6 shows the ratio of the number of vector distance computing between the exhaustive and the proposed k-NN search algorithms under the situations of different values of k. According to the simulation results, the proposed k-NN search algorithm provides a dramatic improvement on the performance of an image retrieval system especially when the value of k and the size of feature histograms are large. As shown in

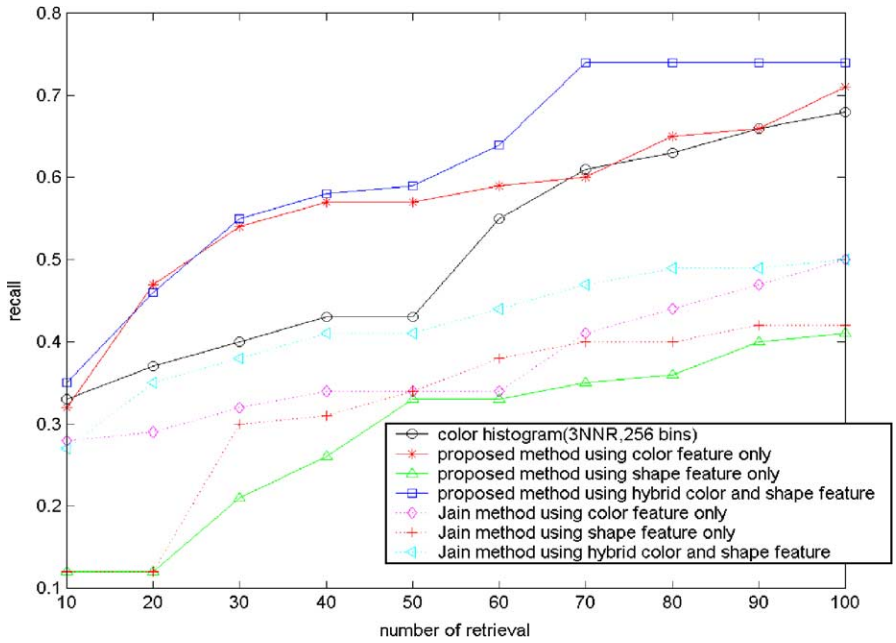

Fig. 8. Average recall versus number of retrievals.

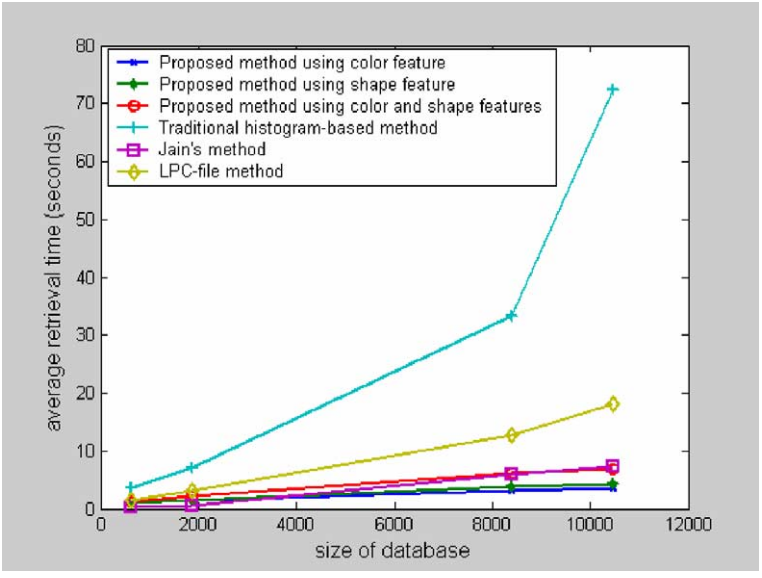

Fig. 9. Average retrieval time (in seconds) versus the size of database.

Table 2, the average retrieval time by ranging the size of histogram from 64 to 1024 in matching a query image with the whole test database is nearly constant using the proposed k-NN search method. On the contrary, the performance of image retrieval with exhaustive search method has considerable variation in execution speed from small to large histograms. The LPC-file method [3] is also simulated for comparison purposed. Although the LPC-file method improves the performance of traditional histogram-based methods, our method is still better that that of the LPC-file method, shown in Table 2. The execution speed of the proposed image retrieval method is much faster than that of the compared methods, and hence, our method is practical to construct a very large image database retrieval system.

It is difficult to derive a formal method in evaluating the retrieval accuracy of an image database system. Traditional metrics for evaluating performance are recall and precision. They are functions of both correct matches and the relevance of database images to a query. The retrieval accuracy measured by precision and recall is computed as the following. Recall measures the ability of the system to retrieval all the images that are relevant and defined as:

$$\text{Recall} = \frac{\text{relevances correctly retrieved}}{\text{all relevances}}.$$

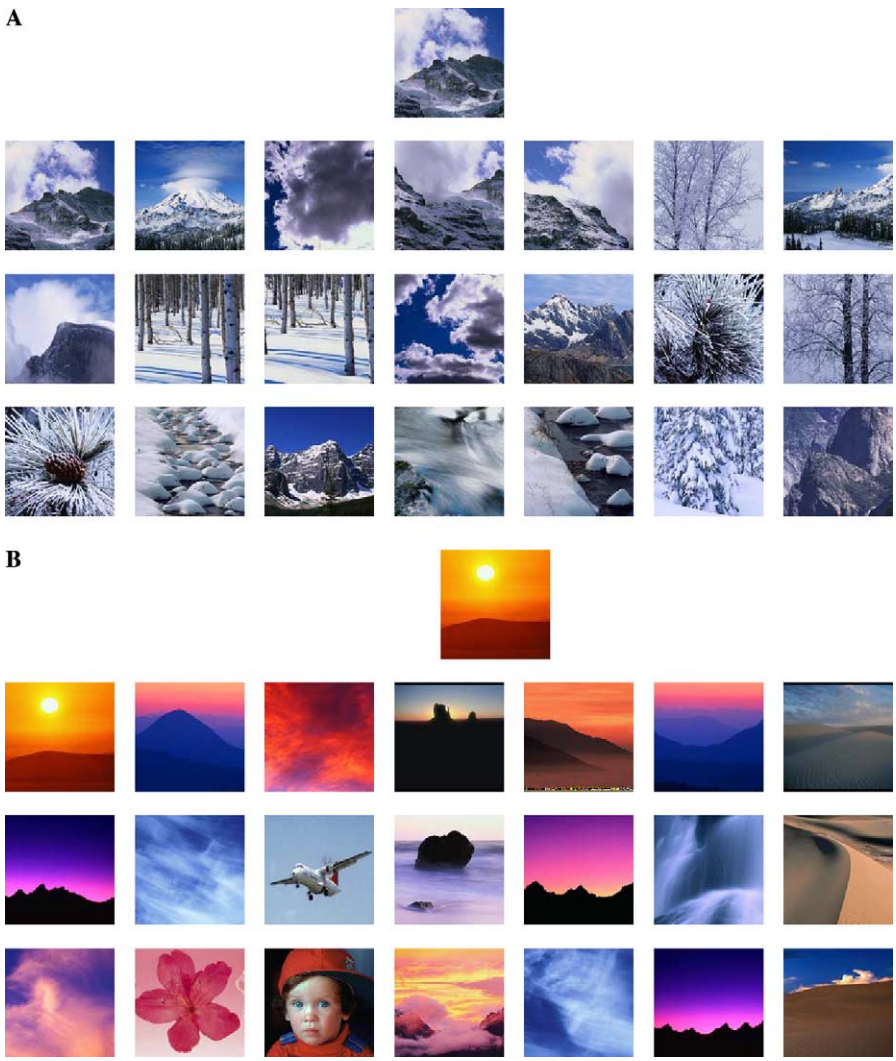

Fig. 10. Examples of retrieval results of the proposed method using: (A) color feature and (B) shape feature. The top image is the query image, the retrieval results are ranked from left-to-right and top-to-bottom according their similarity measurements.

Precision measures the ability of the system to retrieve only images that are relevant and can be computed by:

$$\text{Precision} = \frac{\text{relevances correctly retrieved}}{\text{all retrieved}}.$$

Recall and precision require a ground truth to assess the relevance of images for a set of significant queries. We compare the retrieval performance of the proposed method to the conventional histogram method (QBIC) [10] and Jain and Vailaya's method [11]. The proposed method can be viewed as a variation of histogram approach in terms of the fast k-NN search method and distance measure used. A variety of color and shape histograms with different sizes are extracted from each image in the database for the purpose of performance comparison. The average precision and recall curves are plotted in Figs. 7 and 8. It can be seen that from the figures the proposed method achieves good results in terms of the retrieval accuracy compared to the conventional histogram method [10] and Jain and Vailaya's method [11]. As mentioned in the previous sections, the retrieval accuracy depends on the size of histogram used. High-dimensional histograms achieve good performance in general.

Fig. 11. Examples of retrieval results of the compared methods: (A) the query image; (B) the results of the proposed method using shape and color features; (C) the results of Jain's method; and (D) the results of LPC-file method with 1024-dimensional space. For each figure, the retrieval results are ranked from left-to-right and top-to-bottom according their similarity measurements.

The performance of the proposed method is close to that of the high-dimensional histogram methods even when the number of bins used in our method is small. Moreover, the retrieval speed of the proposed method is much faster than that of the high-dimensional histogram method, shown in Fig. 9. The bin number of histogram can be further decreased without sacrificing the retrieval accuracy in terms of precision and recall if we represent an image as the proposed region-based color histogram according to Section 5.

Figs. 10A and B are retrieval examples of the proposed method using color feature and using shape feature, respectively. Figs. 11B–D are the three examples of retrieval under the same query image (Fig. 11A) using the proposed method, Jain and Valiaya's method, and the LPC-file method, respectively. The performance of the proposed method is superior to that of Jain and Vailaya's method and that of the LPC-file method subjectively.

6. Conclusion

In this paper, a fast k-NN search method using the principal axis analysis for image retrieval is proposed. The problem of high-dimensional feature histograms, which result in a high-dimensional feature space and suffer from high index and retrieval costs, is also solved by the proposed method. That is the proposed method has good performance in terms of the retrieval accuracy and retrieval speed. Experimental results show that the proposed method is fast and effective.

There are some deficiencies of the proposed method. First, the proposed method is based on a single feature type. The index structure of hybrid features such as shape, texture, spatial relationship, and semantic information should also be included in the proposed retrieval system. Another limitation of the system is that it does not handle the problem of indexing high-dimensional feature histograms for relevance feedback, which is known to be a feasible way for capturing users' semantic. However, the execution speed of a CBIR system would degrade dramatically to include a relevance feedback mechanism in the system. More work is needed to address these issues.

References

[1] W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.
[2] J.Z. Wang, J. Li, G. Wiederhold, SIMIPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (9) (2001) 947–963.
[3] G.H. Cha, Xiaoming Zhu, Dragutin Petkovic, C.W. Chung, An efficient indexing method for nearest neighbor searches in high-dimensional image databases, IEEE Trans. Multimedia 4 (1) (2002) 76–87.
[4] G. Lu, Techniques and data structures for efficient multimedia retrieval based on similarity, IEEE Trans. Multimedia 4 (3) (2002) 372–384.
[5] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, W. Niblack, Efficient color histogram indexing for quadratic form distance functions, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 729–736.
[6] G. Cha, C. Chung, Multi-mode indices for effective image retrieval in multimedia systems, in: Proc. IEEE Multimedia Computing Systems, 1998, pp. 152–159.
[7] Y. Deng, B.S. Manjunath, C. Kenney, M.S. Moore, H. Shin, An efficient color representation for image retrieval, IEEE Trans. Image Process. 10 (1) (2001) 140–147.
[8] R. Ng, A. Sedighian, Evaluating multi-dimensional indexing structures for images transformed by principal component analysis, in: Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.
[9] R. Webber, H.-J. Schek, S. Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in: Proc. 24th Int. Conf. VLDB, 1998, pp. 194–205.
[10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, Query by image and video content: the QBIC system, IEEE Comput. 28 (9) (1995) 23–32.
[11] K.A. JAIN, A. Vailaya, Image retrieval using color and shape, Pattern Recognit. 29 (8) (1996) 1233–1243.
[12] S.C. Cheng, A region-growing approach to color segmentation using 3-D clustering and relaxation labeling, IEE Proc.—Image Vis. Signal Process. 150 (4) (2003) 270–276.
[13] H.A. David, Order Statistics, Wiley, New York, 1980.