Fast indexing and searching strategies for feature-based image database systems

Li-Wei Kang  
Jin-Jang Leou  
National Chung Cheng University  
Department of Computer Science and Information Engineering  
Chiayi, Taiwan 621  
E-mail: jtleou@cs.ccu.edu.tw

Abstract. Because visual data require a large amount of memory and computing power for storage and processing, it is greatly desired to efficiently index and retrieve the visual information from multimedia database systems. We propose efficient indexing and searching strategies for feature-based image database systems, in which uncompressesd and compressed domain image features are employed. Each query or stored image is represented by a set of features extracted from the image. The weighted square sum error distance is employed to evaluate the ranks of retrieved images. Many fast clustering and searching techniques exist for the square sum error distance used in vector quantization (VQ), in which different features have different dynamic ranges and different importances, i.e., different features may have different weighting coefficients. We derive a set of inequalities based on the weighted squared sum error distance and employ it to speed up the indexing (clustering) and searching procedures for feature-based image database systems.

Good simulation results show the feasibility of the proposed approaches. © 2005 SPIE and IS&T. [DOI: 10.1117/1.1866148]

1 Introduction

Multimedia database systems become more and more popular with the advent of high-powered PCs, low-cost computer storage devices, broadband networks, and visual/audio compression standards. Because visual data require a large amount of memory and computing power for storage and processing, it is greatly desired to efficiently index and retrieve (search) the visual information from multimedia database systems. Some existing systems include the famous QBIC system from IBM, the Safe system from Columbia University, the VIR Image Engine from Virage, and the Photobook from Massachusetts Institute of Technology.1–3 Traditionally, database systems are usually accessed by text-based queries,4,5 such as keywords, captions, and filenames, which are not suitable for a modern image database system.

Image retrieval based on image content is greatly desirable in a number of applications.6 As shown in Fig. 1, feature vectors representing images are usually extracted, organized, and stored properly in a multimedia database system when it is created. In the query process, a feature vector (or a set of features) representing a query image is extracted from the query image, and then the similarity (or distance) measures between the query image and stored images in the image database system are evaluated. Finally, the most similar image(s) will be presented to the user.

For a feature-based image database system, each image can be represented by a k-dimensional feature vector, i.e., an image is converted into a k-attribute numeric data in a k-dimensional Euclidean space. To make the image database scalable to a very large size, efficient multidimensional indexing and searching techniques must be explored. Traditional multidimensional data structures, such as R-tree, kDB-tree, and grid files, are not suitable for image feature indexing due to inability to scale to high dimensionality.7 A kind of famous multidimensional indexing structure is the tree-based indexing technique that can be classified into data partitioning (DP)-based and space partitioning (SP)-based index structures. A DP-based index structure consists of bounding regions arranged in a spatial containment hierarchy, such as R-tree, X-tree,8 and SR-tree.9 An SP-based index structure consists of space recursively partitioned into mutually disjoint subspaces. The hierarchy of partitions forms the tree structures, such as kd-tree,10 kDB-tree,11 and hB-tree.12

Berchtold et al.13 proposed an indexing method, called the pyramid-technique, for high-dimensional data spaces. It is claimed that the pyramid technique outperforms the X-tree and the Hilbert R-tree. Kim et al.14 proposed an enhanced version of the R*-tree. Chakrabarti and Mehrotra7 proposed a multidimensional data structure called the hybrid tree for indexing high-dimensional feature spaces. It is claimed that the hybrid tree outperforms both purely DP-based and SP-based indexing structure. Wu15 presented a context-based indexing technique, called ContIndex, which is formally defined by adapting a classification tree concept. A special neural network model was developed to create node categories by fusing multimodal feature measures. Gong et al.16 presented an indexing scheme using color histogram, in which a B*-tree is used to store the numerical keys of color histograms in the database system. Vazirgiannis et al.17 proposed two indexing schemes using the R-tree indexing structure for large multimedia applications.
fast algorithms is to use some inequalities to reject many unlike vectors, without detailed calculating of the distance functions between vectors. Several fast algorithms for VQ are introduced briefly as follows.

2.1 Mean Difference Method
Within the mean difference method (MDM) proposed by Lee and Chen,25 let \( \mathbf{x} = (x_1, x_2, \ldots, x_k) \) be an input \( k \)-dimensional vector and \( \mathbf{y} = (y_1, y_2, \ldots, y_k) \) be a \( k \)-dimensional vector (code word) in the codebook. Define the mean values of \( \mathbf{x} \) and \( \mathbf{y} \) as

\[
\begin{align*}
m_x &= \frac{1}{k} \sum_{j=1}^{k} x_j, \\
m_y &= \frac{1}{k} \sum_{j=1}^{k} y_j.
\end{align*}
\]

For the square sum error distance function between \( \mathbf{x} \) and \( \mathbf{y} \), i.e.,

\[d^2(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{k} (x_j - y_j)^2,
\]

we have

\[d^2(\mathbf{x}, \mathbf{y}) \geq k(m_x - m_y)^2.
\]

That is, for a vector (code word) \( \mathbf{y} \) in the codebook, if \( k(m_x - m_y)^2 \) is larger than the current minimum distortion \( d^2_{\text{min}} \), \( \mathbf{y} \) will not be the closest vector (code word) to the input vector \( \mathbf{x} \) and thus can be rejected.

2.2 Equal-Average Equal-Variance Nearest-Neighbor Search Algorithm
For the equal-average equal-variance nearest-neighbor search (EENNS) algorithm proposed by Lee and Chen,26 the variance \( V_x^2 \) of \( \mathbf{x} \) is defined as

\[V_x^2 = \frac{1}{k} \sum_{j=1}^{k} (x_j - m_x)^2.
\]

It can be proved that

\[d(\mathbf{x}, \mathbf{y}) \geq |V_x - V_y|.
\]

Within the EENNS algorithm, the vectors (code words) whose means are similar to \( m_x \), but whose variances are very different from \( V_x^2 \) will be rejected accordingly.

2.3 Baek et al.'s Method
Within the algorithm proposed by Baek et al.,27 we have

\[d^2(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{k} (x_j - y_j)^2,
\]

then

\[d^2(\mathbf{x}, \mathbf{y}) \geq k(m_x - m_y)^2 + (V_x - V_y)^2.
\]

2.4 Integral Projection Mean-Sorted Partial Search Algorithm
For the integral projection mean-sorted partial search (IP-MPS) algorithm proposed by Lin and Tai,28 for an input \( k \)-dimensional vector \( \mathbf{x}(i,j) \) arranged by two indices \((i,j)\) with \( k=n \times n \), where \( i, j = 1, 2, \ldots, n \), the three kinds of integral projections are defined as follows:
Fast indexing and searching strategies...

1. vertical projection: 
   \[ v_{P_x}(j) = \sum_{i=1}^{n} x(i,j), \quad 1 \leq j \leq n, \]  
   \[ (7) \]

2. horizontal projection: 
   \[ h_{P_y}(i) = \sum_{j=1}^{n} x(i,j), \quad 1 \leq i \leq n, \]  
   \[ (8) \]

3. massive projection: 
   \[ m_{P_x} = \sum_{i=1}^{n} \sum_{j=1}^{n} x(i,j). \]  
   \[ (9) \]

For the three simple distortion measures given by
\[ d^2_{p}(x,y) = (m_{P_x} - m_{P_y})^2, \]  
\[ (10) \]
\[ d^2_{v}(x,y) = \sum_{i=1}^{n} [v_{P_x}(i) - v_{P_y}(i)]^2, \]  
\[ (11) \]
\[ d^2_{h}(x,y) = \sum_{i=1}^{n} [h_{P_y}(i) - h_{P_x}(i)]^2, \]  
\[ (12) \]
we have
\[ d^2_{p}(x,y) \leq n^2 d^2(x,y), \]  
\[ (13) \]
\[ d^2_{v}(x,y) \leq n d^2(x,y), \]  
\[ (14) \]
and
\[ d^2_{h}(x,y) \leq n d^2(x,y). \]  
\[ (15) \]

2.5 Chang and Hu’s Method

In the generalized integral projection (GIP) model developed by Chang and Hu, a k-dimensional vector can be partitioned into any p segments and each segment has q components (pixels), where \( k = p \times q \). For each of the \( p \) segments of a codevector \( x \), the \( q \) components (pixel values) can be summed to obtain a projection \( P_{x}(l) \), for \( l = 1, 2, \ldots, p \). If the distortion measure between two codevectors is given by
\[ d^2_{(p,q)}(x,y) = \sum_{i=1}^{p} [P_{x}(i) - P_{y}(i)]^2, \]  
\[ (16) \]
where \( P_{x}(l) \) and \( P_{y}(l), l = 1, 2, \ldots, p \), are projections of the codevectors \( x \) and \( y \), respectively, we have
\[ d^2_{(p,q)}(x,y) \leq q \times d^2(x,y). \]  
\[ (17) \]

3 Proposed Fast Indexing and Searching Strategies for Image Database Systems

3.1 Features Extracted from Uncompressed and Compressed Images

In this study, the first type of features extracted from uncompressed images is the color coherence vector \( CCV \), which is a type of histogram-based features incorporating the spatial information. Within the CCV, each pixel in a given color bucket is classified into either “coherent” or “incoherent.” A pixel is coherent if the size of its connected component exceeds a predefined threshold \( \tau \); otherwise, it is incoherent. A CCV stores the number of coherent and incoherent pixels for each color bucket. The CCV can take spatial information into account and outperform the traditional color histogram method. In the CCV method, the number \( n \) of color buckets is set to 64 and the threshold \( \tau \) is set to be 1% of the size of an image, which are followed in this study. That is, \( n = 64 \) and the threshold \( \tau \) is set to be 655 (the size of an image is 65,536). Hence the CCV for an image can be represented as a 128-dimensional vector \( [(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_{64}, \beta_{64})] \).

In this study, the second type of features extracted from uncompressed images is the five (among 14) statistical texture measures derived by Haralick et al., which can be computed based on the gray-scale coocurrence matrices. Haralick et al. derived the 14 measures, but only five of them are found to be truly useful. The gray-scale coocurrence matrices, which describe spatial dependence of pixels in a gray-scale image, are often used in statistical analysis of textures. Figure 2(a) shows a 4×4 image block with gray values 0 to 3, and Fig. 2(b) shows the corresponding general form of a coocurrence matrix, where each element \((i,j)\) denotes the occurrence frequency of two neighboring pixels with gray-values \( i \) and \( j \). If \( S_{\theta}(d) \) denotes a coocurrence matrix with distance \( d \) and direction \( \theta \), Fig. 2(c) shows the neighboring gray-value pairs for computing \( S_{\theta}(d) \) and Fig. 2(d) shows the contents of \( S_{\theta}(d) \), where “e” denotes a neighboring gray-value pair. The other coocurrence matrices can be similarly calculated.

If \( N \) denotes the number of gray values and \( S_{\theta}(i,j|d) \) denotes the \((i,j)\)’th element in \( S_{\theta}(d) \), the five texture measures are given by
\[ f_1 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [S_{\theta}(i,j|d)]^2, \]  
\[ (18) \]
2. entropy: \[ f_2 = -\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} S_d(i,j,d) \log S_d(i,j,d), \]

(19)

3. correlation: \[ f_3 = \frac{1}{\sigma_x \sigma_y} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-u_x)(j-u_y) \times S_d(i,j,d), \]

(20)

4. inverse different moment (IDM): \[ f_4 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{S_d(i,j,d)}{1+(i-j)^2}, \]

(21)

5. inertia: \[ f_5 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-j)^2 S_d(i,j,d). \]

(22)

In Eq. (20), \( u_x, u_y, \sigma_x, \) and \( \sigma_y \) are the means and standard deviations of \( S_d(i,j,d) \) along the horizontal and vertical directions, respectively, which are given by

\[ u_x = \sum_{i=0}^{N-1} i \sum_{j=0}^{N-1} S_d(i,j,d), \]

(23)

\[ u_y = \sum_{j=0}^{N-1} j \sum_{i=0}^{N-1} S_d(i,j,d), \]

(24)

\[ \sigma_x^2 = \sum_{i=0}^{N-1} (i-u_x)^2 \sum_{j=0}^{N-1} S_d(i,j,d), \]

(25)

\[ \sigma_y^2 = \sum_{j=0}^{N-1} (j-u_y)^2 \sum_{i=0}^{N-1} S_d(i,j,d). \]

(26)

To reduce the time complexity for computing these five texture measures, the gray levels of an image are requantized to contain only 16 gray levels by using a simple uniform quantization scheme\(^{32}\) and only \( d = 1, \theta = 0, 45, 90, \) and 135 deg are used.\(^ {33} \) For each of the five texture measures, the means and the variances of the four corresponding values for \( \theta = 0, 45, 90, \) and 135 deg are computed. Then the means and the variances of the five texture measures form a 10-D feature vector for representing an image.

The seven invariant moments developed by Hu\(^ {34} \) are used as shape features in this study, which are invariant to translation, rotation, and scaling.\(^ {34,35} \) The 128 color features, 7 shape features, and 10 texture features form a 145-D feature vector for an uncompressed image.

On the other hand, the image features extracted from compressed images are the discrete cosine transform (DCT)-based coefficients extracted from block-based DCT-compressed images.\(^ {24} \) The two-dimensional (2-D) DCT and inverse DCT (IDCT) are given by

\[ C(u,v) = \alpha(u) \alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \times \cos \left( \frac{(2x+1)u \pi}{2N} \right) \cos \left( \frac{(2y+1)v \pi}{2N} \right) \]

for \( u,v = 0,1,2,...,N-1, \)

(27)

\[ f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u) \alpha(v) C(u,v) \times \cos \left( \frac{(2x+1)u \pi}{2N} \right) \cos \left( \frac{(2y+1)v \pi}{2N} \right) \]

for \( x,y = 0,1,2,...,N-1, \)

(28)

where

\[ \alpha(u) = \begin{cases} \left( \frac{1}{N} \right)^{1/2} & \text{for } u = 0, \\ \left( \frac{2}{N} \right)^{1/2} & \text{for } u = 1,2,...,N-1. \end{cases} \]

(29)

In this study, a set of nine features \( (f_1,f_2,...,f_9) \) is determined as follows:

\[ f_1 = C(0,0) \text{ (the dc coefficient).} \]

(30)

\[ f_2 = C(0,1)+C(0,2)+C(1,0)+C(1,1)+C(2,0), \]

(31)

\[ f_3 = C(0,3)+C(0,4)+C(1,2)+C(1,3)+C(2,1)+C(2,2) + C(3,0)+C(3,1)+C(4,0), \]

(32)

\[ f_4 = C(0,5)+C(0,6)+C(1,4)+C(1,5)+C(2,3)+C(2,4) + C(3,2)+C(3,3)+C(4,1)+C(4,2)+C(5,0) + C(5,1)+C(6,0), \]

(33)

\[ f_5 = C(0,0)+C(1,0)+C(2,0)+C(3,0)+C(4,0)+C(5,0) + C(6,0)+C(7,0), \]

(34)

\[ f_6 = C(1,1)+C(2,1)+C(3,1)+C(4,1)+C(5,1)+C(6,1) + C(7,1), \]

(35)

\[ f_7 = C(0,1)+C(0,2)+C(0,3)+C(0,4)+C(0,5)+C(0,6) + C(0,7), \]

(36)

\[ f_8 = C(1,2)+C(1,3)+C(1,4)+C(1,5)+C(1,6)+C(1,7), \]

(37)

\[ f_9 = C(2,2)+C(2,3)+C(3,2)+C(3,3)+C(3,4)+C(4,3) + C(4,4)+C(4,5)+C(5,4)+C(5,5). \]

(38)

Extracting features from DCT-compressed images directly is efficient for indexing an image database system. The DCT coefficients represent some gray-level variations.
and dominant directions in the original image. Here \( f_1 \) is the dc coefficient representing the average energy of the corresponding image block; \( f_2, f_3, \) and \( f_4 \) represent frequency band characteristics; and \( f_5, f_6, f_7, f_8, \) and \( f_9 \) represent spatial characteristics. Each \( 8 \times 8 \) block produces a 9-D feature vector \((f_1, f_2, \ldots, f_9)\) and the means and the variances of all the vectors within an image are calculated. Then the corresponding nine means and nine variances form an 18-D feature vector, which is the feature vector representing a compressed image.

### 3.2 Proposed Inequalities for the Weighted Square Sum Error Distance

Assume that \( x = (x_1, x_2, \ldots, x_k) \) and \( y = (y_1, y_2, \ldots, y_k) \) are two \( k \)-dimensional feature vectors representing two images, \( x \) and \( y \), respectively, and \( w_1, w_2, \ldots, w_k \) are the weighting coefficients for feature vectors with \( w_j \geq 0, j = 1, 2, \ldots, k \) and \( \sum_{j=1}^{k} w_j = 1 \). The weighted square sum error distance \( d^2(x, y) \) between two images or feature vectors \( x \) and \( y \) is given by

\[
d^2(x, y) = \sum_{j=1}^{k} w_j (x_j - y_j)^2.
\]

If the weighted sum \( m_x \) and variance \( V_x^2 \) of \( x = (x_1, x_2, \ldots, x_k) \) are defined, respectively, as

\[
m_x = \sum_{j=1}^{k} w_j x_j,
\]

\[
V_x^2 = \sum_{j=1}^{k} w_j (x_j - m_x)^2,
\]

a lemma can be given as follows.

**Lemma 1.** If \( x = (x_1, x_2, \ldots, x_k) \) and \( y = (y_1, y_2, \ldots, y_k) \) are two \( k \)-dimensional feature vectors, \( M_x = (m_1, m_2, \ldots, m_k) \) and \( M_y = (m_1, m_2, \ldots, m_k) \), then

\[
d^2(x, M_x) \leq d^2(x, M_y).
\]

Based on Eqs. (39) to (42), the second lemma can be derived as follows.

**Lemma 2.** If \( x = (x_1, x_2, \ldots, x_k) \) and \( y = (y_1, y_2, \ldots, y_k) \) are two \( k \)-dimensional feature vectors, then

\[
d^2(x, y) \geq (m_x - m_y)^2,
\]

\[
d^2(x, y) \geq (V_x - V_y)^2,
\]

\[
d^2(x, y) \geq (m_x - m_y)^2 + (V_x - V_y)^2.
\]

Similar to the IPMPS algorithm\(^{28,29}\) for VQ, if \( x \) is a \( k \)-dimensional feature vector with \( k = n \times n \), i.e., \( x = (x_{11}, x_{12}, \ldots, x_{nn}) \). Using two indices, \( i \) and \( j \), two weighted projections can be defined as

1. weighted vertical projection: 
\[
vp_x(j) = \sum_{i=1}^{n} w_{ij} x_{ij},
\]

\[
\leq j \leq n,
\]

2. weighted horizontal projection: 
\[
hp_x(i) = \sum_{j=1}^{n} w_{ij} x_{ij},
\]

\[
\leq i \leq n.
\]

If \( x = (x_{11}, x_{12}, \ldots, x_{nn}) \) and \( y = (y_{11}, y_{12}, \ldots, y_{nn}) \) are two \( k \)-dimensional feature vectors with \( k = n \times n \), two simple distance measures can be defined, respectively, as

1. \( d^2_w(x, y) = \sum_{i=1}^{n} [vp_x(i) - vp_y(i)]^2, \)

2. \( d^2_w(x, y) = \sum_{i=1}^{n} [hp_x(i) - hp_y(i)]^2, \)

**Lemma 3.** If \( x \) and \( y \) denote two \( k \)-dimensional feature vectors using two indices with \( k = n \times n \), then

1. \( d^2(x, y) \geq d^2_w(x, y), \)

2. \( d^2(x, y) \geq d^2_h(x, y). \)

Similar to the GIP model\(^{29}\) for VQ, a \( k \)-dimensional feature vector can be segmented into any projection pair \((p, q)\) with \( k = p \times q \), where \( p \) and \( q \) are two positive integers. For each of the \( p \) segments of a feature vector \( x \), the weighted sum of \( q \) component values can be viewed as a weighted projection \( P_x(l) \), for \( l = 1, 2, \ldots, p \). A distance measure between two feature vectors is defined as

\[
d^2_{(p, q)}(x, y) = \sum_{i=1}^{p} [P_x(i) - P_y(i)]^2,
\]

where \( P_x(l) \) and \( P_y(l) \) are the weighted projections of feature vectors \( x \) and \( y \), respectively, for \( l = 1, 2, \ldots, p \).

**Lemma 4.** If \( x \) and \( y \) are two \( k \)-dimensional feature vectors using two indices with \( k = p \times q \), where \( p \) and \( q \) are two positive integers, then

\[
d^2(x, y) \geq d^2_{(p, q)}(x, y).
\]

The set of inequalities (lemmas 2 to 4) based on the weighted square sum error distance is used to speed up the indexing (clustering) and searching phases of the proposed approach.

### 3.3 Proposed Fast Indexing and Searching Strategies

The proposed approach consists of two phases, namely, the indexing (clustering) phase and the searching phase. In the indexing phase, the feature vectors for all images are calculated and stored in the database system. And the images (feature vectors) in the database system are classified into a
A prespecified number \( K \) of clusters with \( K \) cluster centers by the proposed fast modified \( K \)-means clustering techniques. In the searching phase, a small number \( M \) of closest clusters (based on the cluster centers) for the query image is first determined and then the \( N \) most similar images (ranks 1, 2, \ldots, \( N \)) for the query image will be searched among the \( M \) closest clusters.

Within the indexing phase of the proposed approach, the feature vectors and some auxiliary information, such as weighted sums, variances, and weighted projections, and weighted generalized projections of feature vectors, for all images are calculated, sorted by variances, and stored in the database system. Within the proposed fast modified \( K \)-means clustering algorithm using the weighted square sum error distance, a set of derived inequalities [Eqs. (43) to (45), (50), (51), and (53)] is used to speed up the modified clustering algorithm.

On the other hand, the set of derived inequalities is also used to speed up the searching phase. The proposed searching approach is described as follows, where the input is a query image, an image database system containing \( K \) clusters, and a set of weighting coefficients, and the output is the \( N \) most similar images within the image database system of the query image.

1. Step 1. Extract the feature vector \( \mathbf{x} \) from the query image and calculate the auxiliary information.
2. Step 2. Find the \( M \) closest clusters, \( C_1, C_2, \ldots, C_M \), among the \( K \) clusters of \( \mathbf{x} \).
3. Step 3. For each \( C_i, i = 1, 2, \ldots, M \), of the \( M \) closest clusters, find the \( N \) closest feature vectors (images) among \( C_i \) of \( \mathbf{x} \).
4. Step 4. Determine the \( N \) closest feature vectors (images) of \( \mathbf{x} \) among the \( MN \) feature vectors (images) obtained in step 3.

The searching phase can be summarized in Fig. 3. Note that the six inequalities developed in this study can be “individually” or “together” (i.e., different sets of inequalities) used to speed up both the proposed indexing (clustering) and searching phase.

### Simulation Results

Within the proposed fast indexing and searching strategies, an image database system containing 10,000 256×256 uncompressed images and 10,000 256×256 compressed images, respectively, is used to evaluate the performance of the proposed approaches. The 10,000 uncompressed images consist of 1000 texture images with 256 gray levels, 70 color texture images, 600 binary shape images, 30 color shape images, and 8300 natural color images, whereas the 10,000 compressed images are the corresponding JPEG-compressed version of the 10,000 uncompressed images.

#### Table 1

<table>
<thead>
<tr>
<th>No. of Clusters</th>
<th>ES</th>
<th>Eq. (43)</th>
<th>Eq. (44)</th>
<th>Eq. (45)</th>
<th>Hor</th>
<th>Ver</th>
<th>Hybrid-I</th>
<th>Hybrid-II</th>
<th>Hybrid-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>156.10</td>
<td>144.29</td>
<td>70.03</td>
<td>72.18</td>
<td>150.50</td>
<td>174.05</td>
<td>70.75</td>
<td>173.62</td>
<td>80.41</td>
</tr>
<tr>
<td>50</td>
<td>725.24</td>
<td>675.09</td>
<td>224.87</td>
<td>235.69</td>
<td>715.02</td>
<td>836.07</td>
<td>225.28</td>
<td>839.59</td>
<td>278.86</td>
</tr>
<tr>
<td>100</td>
<td>8377.35</td>
<td>7363.20</td>
<td>2024.44</td>
<td>2138.03</td>
<td>7886.21</td>
<td>9332.66</td>
<td>2037.24</td>
<td>9236.11</td>
<td>2583.42</td>
</tr>
</tbody>
</table>

(a) Weighting coefficient set 1

<table>
<thead>
<tr>
<th>No. of Clusters</th>
<th>ES</th>
<th>Eq. (43)</th>
<th>Eq. (44)</th>
<th>Eq. (45)</th>
<th>Hor</th>
<th>Ver</th>
<th>Hybrid-I</th>
<th>Hybrid-II</th>
<th>Hybrid-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>151.82</td>
<td>116.99</td>
<td>65.97</td>
<td>64.70</td>
<td>123.59</td>
<td>132.70</td>
<td>63.77</td>
<td>122.59</td>
<td>74.37</td>
</tr>
<tr>
<td>50</td>
<td>1937.00</td>
<td>1178.65</td>
<td>515.86</td>
<td>467.14</td>
<td>1276.80</td>
<td>1328.60</td>
<td>442.97</td>
<td>1051.77</td>
<td>580.46</td>
</tr>
<tr>
<td>100</td>
<td>7486.68</td>
<td>3937.28</td>
<td>1603.83</td>
<td>1375.77</td>
<td>4161.65</td>
<td>4191.36</td>
<td>1251.70</td>
<td>2887.98</td>
<td>1702.80</td>
</tr>
</tbody>
</table>
The processing time (in seconds) for indexing and searching of the image database system is employed as the performance measure. The proposed fast indexing (clustering) approach is compared with the exhaustive search approach, denoted by ES. The proposed fast searching approach is compared with four existing searching approaches: (1) ES; (2) clustering all the images in the database system first, and exhaustive searching among a small number of clusters (denoted by CES); (3) fast searching for all the images in the database system without clustering (denoted by FSWC), which is the degenerated version of the proposed fast searching approach without clustering; and (4) the hybrid tree (denoted by HT). The HT is a very good tree-based multidimensional indexing and searching structure, which significantly outperforms both purely DP-based and SP-based index mechanisms at all dimensionalities for large databases. The source code of the hybrid tree is available on http://www.ics.uci.edu/~kaushik/research/htree.html.

In this paper, the dimensionality of the uncompressed image feature vector is 145, including 128 color features, 7 shape features, and 10 texture features, whereas the dimensionality of the compressed image feature vector is 18. The six proposed inequalities are tested individually and together. For the uncompressed images, the \((p, q)\) in Eq. (53) is set to be (5,29) and (29,5), denoted by Hor and Ver, respectively. For the DCT-compressed images, the \((p, q)\) in Eq. (53) is set to be (3,6) and (6,3), also denoted by Hor and Ver, respectively. The hybrid inequality includes three variations: (1) applying Eqs. (44) and (45) (denoted by Hybrid-I); (2) applying Eq. (43), Hor, and Ver (denoted by

Table 2 The processing times (in seconds) for clustering the 10,000 DCT-compressed images into 10, 50, and 100 clusters by the ES and the proposed approaches applying different sets of inequalities: (a) weighting coefficient set 1 and (b) weighting coefficient set 2.

<table>
<thead>
<tr>
<th>No. of Clusters</th>
<th>ES</th>
<th>Eq. (43)</th>
<th>Eq. (44)</th>
<th>Eq. (45)</th>
<th>Hor</th>
<th>Ver</th>
<th>Hybrid-I</th>
<th>Hybrid-II</th>
<th>Hybrid-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>61.40</td>
<td>32.30</td>
<td>22.52</td>
<td>27.19</td>
<td>43.33</td>
<td>58.99</td>
<td>21.75</td>
<td>37.29</td>
<td>26.09</td>
</tr>
<tr>
<td>50</td>
<td>253.09</td>
<td>87.06</td>
<td>51.47</td>
<td>71.34</td>
<td>122.21</td>
<td>189.33</td>
<td>43.72</td>
<td>90.30</td>
<td>60.70</td>
</tr>
<tr>
<td>100</td>
<td>1528.96</td>
<td>416.71</td>
<td>237.44</td>
<td>355.48</td>
<td>591.38</td>
<td>988.27</td>
<td>224.75</td>
<td>380.09</td>
<td>269.03</td>
</tr>
</tbody>
</table>

The average processing times (in seconds) for searching the 10 most similar images from the 10,000 uncompressed images by the proposed approaches applying different sets of inequalities: (a) weighting coefficient set 1 and (b) weighting coefficient set 2.

<table>
<thead>
<tr>
<th>No. of Clusters</th>
<th>Eq. (43)</th>
<th>Eq. (44)</th>
<th>Eq. (45)</th>
<th>Hor</th>
<th>Ver</th>
<th>Hybrid-I</th>
<th>Hybrid-II</th>
<th>Hybrid-III</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.3109</td>
<td>0.2032</td>
<td>0.2060</td>
<td>0.3312</td>
<td>0.4119</td>
<td>0.2037</td>
<td>0.4334</td>
<td>0.2796</td>
</tr>
<tr>
<td>50</td>
<td>0.1351</td>
<td>0.1219</td>
<td>0.1236</td>
<td>0.1434</td>
<td>0.1769</td>
<td>0.1230</td>
<td>0.1862</td>
<td>0.1659</td>
</tr>
<tr>
<td>100</td>
<td>0.1466</td>
<td>0.1373</td>
<td>0.1401</td>
<td>0.1549</td>
<td>0.1889</td>
<td>0.1395</td>
<td>0.1972</td>
<td>0.1845</td>
</tr>
</tbody>
</table>

The proposed fast indexing (clustering) approach is compared with the exhaustive search approach, denoted by ES. The proposed fast searching approach is compared with four existing searching approaches: (1) ES; (2) clustering all the images in the database system first, and exhaustive searching among a small number of clusters (denoted by CES); (3) fast searching for all the images in the database system without clustering (denoted by FSWC), which is the degenerated version of the proposed fast searching approach without clustering; and (4) the hybrid tree (denoted by HT). The HT is a very good tree-based multidimensional indexing and searching structure, which significantly outperforms both purely DP-based and SP-based index mechanisms at all dimensionalities for large databases. The source code of the hybrid tree is available on http://www.ics.uci.edu/~kaushik/research/htree.html.

In this paper, the dimensionality of the uncompressed image feature vector is 145, including 128 color features, 7 shape features, and 10 texture features, whereas the dimensionality of the compressed image feature vector is 18. The six proposed inequalities are tested individually and together. For the uncompressed images, the \((p, q)\) in Eq. (53) is set to be (5,29) and (29,5), denoted by Hor and Ver, respectively. For the DCT-compressed images, the \((p, q)\) in Eq. (53) is set to be (3,6) and (6,3), also denoted by Hor and Ver, respectively. The hybrid inequality includes three variations: (1) applying Eqs. (44) and (45) (denoted by Hybrid-I); (2) applying Eq. (43), Hor, and Ver (denoted by
Hybrid-II; and (3) applying Eq. (44), Hor, and Ver (denoted by Hybrid-III).

The processing times (in seconds) for indexing (clustering) the 10,000 uncompressed images and the 10,000 compressed images, respectively, with 10, 50, and 100 clusters by the ES and the proposed approaches applying different sets of inequalities (two different weighting coefficient sets) are listed in Tables 1 and 2. The average processing times (in seconds) for searching the 10 most similar images from the 10,000 uncompressed images and the 10,000 compressed images, respectively, for 250 query images randomly selected from the same image database by the proposed approaches applying different sets of inequalities (two different weighting coefficient sets) are listed in Tables 3 and 4. Here when the size of an image database is 2000, 5000, and 10,000, the image database is clustered into 10, 20, and 50 clusters (i.e., $K = 10, 20, 50$), respectively, and the corresponding searching processes are performed in the closest 2, 2, and 5 clusters (i.e., $M = 2, 2, 5$), respectively.

The two different weighting coefficient sets are (1) identical weighting coefficients for all the dimensions of a feature vector (denoted by weighting coefficient set 1), in this case, the weighted square sum error distance is equivalent to the square sum error distance employed in VQ; and (2) $1/25, 1/640, 2/35$ for each dimension of the texture, color, and shape features, respectively, for uncompressed images and a set of weighting coefficients, $\{3/28, 2/28, 2/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28, 1/28\}$, for compressed images (denoted by weighting coefficient set 2). For the weighting coefficient set 2 of the feature vectors for uncompressed images, the weighting coefficients for the texture and shape features are

<table>
<thead>
<tr>
<th>Table 4</th>
<th>The average processing times (in seconds) for searching the 10 most similar images from the 10,000 DCT-compressed images by the proposed approaches applying different sets of inequalities: (a) weighting coefficient set 1 and (b) weighting coefficient set 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proposed (Different Sets of Inequalities)</td>
</tr>
<tr>
<td></td>
<td>Eq. (43)</td>
</tr>
<tr>
<td>No. of Clusters</td>
<td>Eq. (45)</td>
</tr>
<tr>
<td>10</td>
<td>0.0368</td>
</tr>
<tr>
<td>50</td>
<td>0.0280</td>
</tr>
<tr>
<td>100</td>
<td>0.0313</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eq. (43)</td>
</tr>
<tr>
<td>No. of Clusters</td>
<td>Eq. (45)</td>
</tr>
<tr>
<td>10</td>
<td>0.0346</td>
</tr>
<tr>
<td>50</td>
<td>0.0280</td>
</tr>
<tr>
<td>100</td>
<td>0.0313</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5</th>
<th>The processing times (in seconds) of each iteration for designing a codebook containing 256 ($K = 256$) codevectors for VQ (equal weighting coefficients) by the existing approach (ES) and the proposed approaches applying different sets of inequalities.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed (Different Sets of Inequalities)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ES</td>
</tr>
<tr>
<td>Time</td>
<td>4227.40</td>
</tr>
</tbody>
</table>
Fast indexing and searching strategies...

Table 6 The average processing times (in seconds) for searching the 10 most similar images from the 10,000 uncompressed images by the four existing approaches ES, CES, FSWC, HT, and the proposed approach with $K=50$ and $M=5$ (weighting coefficient sets 1 and 2 are denoted by Set 1 and Set 2, respectively).

<table>
<thead>
<tr>
<th></th>
<th>ES</th>
<th>CES</th>
<th>FSWC</th>
<th>HT</th>
<th>Proposed</th>
<th>Percentage of Improvement with Respect to HT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>0.5019</td>
<td>0.0635</td>
<td>0.2673</td>
<td>0.4192</td>
<td>0.0539</td>
<td>87.14%</td>
</tr>
<tr>
<td>Set 2</td>
<td>0.5074</td>
<td>0.0674</td>
<td>0.0474</td>
<td>0.4436</td>
<td>0.0496</td>
<td>88.82%</td>
</tr>
</tbody>
</table>

Table 7 The average processing times (in seconds) for searching the 10 most similar images from the 10,000 DCT-compressed images by the four existing approaches ES, CES, FSWC, HT, and the proposed approach with $K=50$ and $M=5$ (weighting coefficient sets 1 and 2 are denoted by Set 1 and Set 2, respectively).

<table>
<thead>
<tr>
<th></th>
<th>ES</th>
<th>CES</th>
<th>FSWC</th>
<th>HT</th>
<th>Proposed</th>
<th>Percentage of Improvement with Respect to HT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>0.0932</td>
<td>0.0106</td>
<td>0.0050</td>
<td>0.0060</td>
<td>0.0044</td>
<td>26.67%</td>
</tr>
<tr>
<td>Set 2</td>
<td>0.0998</td>
<td>0.0114</td>
<td>0.0062</td>
<td>0.0124</td>
<td>0.0048</td>
<td>61.29%</td>
</tr>
</tbody>
</table>

Upgraded and that for the color features are degraded. The color features consist of 128 dimensions among the 145 dimensions, which may dominate the whole feature vector if the importances of all the 145 dimensions are identical. We select that the percentages for texture and shape features are 40%, respectively, and 20% for color features. For the weighting coefficient set 2 of the feature vectors for DCT-compressed images, the first and tenth dimensions correspond to the dc information, which represent the average energy (the most important features) of the image. Therefore, the weighting coefficients of these two dimensions are highlighted and the others are degraded. Some query results for several uncompressed and compressed images using two different weighting coefficient sets are shown in Figs. 6–11. The query results for an uncompressed image retrieved from only 1, 2, and 50 (ES) $(M=1,2,5)$ closest clusters within the 50 ($K=50$) clusters of the 10,000 uncompressed images are, respectively, shown in Figs. 12–14.

5 Concluding Remarks

Based on the simulation results obtained in this study, several observations can be found. First, based on the simulation results shown in Tables 1 and 2, the processing times for clustering the stored images in the database system of the proposed approaches are better than that of the existing approach (ES) for comparison.

Second, based on the simulation results shown in Tables 1 to 4, within both the indexing (clustering) and searching phases, Eq. (44) and Hybrid-I [Eqs. (44) and (45)] are the two most effective sets of inequalities for the proposed approaches applying different sets of inequalities. Because in most cases, applying Hybrid-I is better than applying Eq. (44), it is recommended that the proposed approach apply Hybrid-I.

Third, based on the simulation results shown in Tables 1 to 4, the performances of the proposed approaches depend on the characteristics of feature-based image data. For the uncompressed image data, the proposed inequality using variance [Eq. (44)] is a good inequality to reject many unlikely feature vectors (images) so that a great deal of computation time can be saved, whereas the other inequalities are not so good for such a type of feature-based image data. For the compressed image data, the proposed inequality using variance [Eq. (44)] is a good inequality to reject unlikely feature vectors (images), whereas the proposed Hybrid-I is a better set of inequalities to reject many unlikely feature vectors (images).

Fourth, based on the simulation results shown in Tables 6 and 7 and Figs. 4 and 5, the average processing times for searching the 10 most similar images from the 10,000 images of the proposed approaches are better than that of the existing approaches for comparison, namely, ES, CES, FSWC, and HT. Although the HT is suitable for low- and medium-dimensional vectors (e.g., $k = 18$) and outperforms other tree-based approaches, the performance of the HT is degraded, as compared to the proposed approach, when the dimensionality of the feature vector is a large number (e.g., $k = 145$). The performance characteristics of the other existing approaches, CES and FSWC, are similar to that of the HT. That is, the proposed approach is more insensitive...
Fig. 6  Query results for an uncompressed image using weighting coefficient set 1: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.

Fig. 7  Query results for an uncompressed image using weighting coefficient set 2: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.

Fig. 8  Query results for an uncompressed image using weighting coefficient set 1: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.
to feature dimension variation than the existing approaches for comparison. The reason is that within the proposed approaches, a high-dimensional feature vector is projected into the low-dimensional "working" vector space, including the weighted sum, the weighted variance, the weighted vertical projection, and the weighted horizontal projection, etc., so that the dimensionality of the "working" vector space is greatly reduced. Because the processing time or the computational complexity of the distance calculations between two feature vectors of the "working" vector space will be greatly reduced, the proposed approaches will be more efficient in high-dimensional feature vector space than the existing approaches for comparison.

Fifth, based on the simulation results shown in Tables 1 to 5, the weighting coefficients of feature vectors will influence the performances of the proposed approaches. Different sets of inequalities are usually suitable for the same feature vectors using different weighting coefficients. For example, if all the weighting coefficients are identical, the weighted square sum error distance employed in this paper will be equivalently reduced to the square sum error distance employed in VQ. Based on the simulation results shown in Table 5, the processing time for designing a codebook for vector quantization by using the proposed approaches using equal weighting coefficients is only about 6% of the processing time required by the ES approach. In VQ, a feature vector (codevector) is formed by the pixels within an image block, which are highly concentrated about the feature (pixel) mean. That is, two image blocks having similar pixel means may be very similar. Therefore, the inequalities developed for VQ are very effective for clustering codevectors. However, because the dynamic ranges of different feature components are greatly different in this paper, if the weighting coefficients are not properly adjusted, the proposed inequalities in this paper will be less effective than that for VQ. For example, the sum of the 128 color features is equal to the size of an image. If all the weighting coefficients are identical, the weighted sums will be the same for all equal-sized images. Additionally, if the dimensionality of the color features is relatively high and the weighted sums of all feature vectors are very similar, the proposed inequalities using the weighted sums [Eqs. (43), (50), (51), and (53)] will not be effective. Therefore, all the stored images in an image database system are sorted according to their variances, instead of their weighted sums.

Sixth, based on the simulation results shown in Tables 1 to 4, if the number K of clusters is increased, some sets of inequalities become more effective, whereas some sets of inequalities become less effective.

Seventh, based on the simulation results shown in Figs. 6 to 11, the retrieved images (ranks 1 to 10) may be different for different weighting coefficient sets. That is so because the values of the weighted square sum error distance are different for different weighting coefficient sets.

Eighth, Jain and Vailaya addressed an image retrieval method for large image databases. They used CLUSTER, a squared-error clustering algorithm, to reduce the search space. The query image is compared only with the cluster centers to determine its cluster membership. Once the cluster membership of the query image is established, it must be matched only with images in that cluster (the closest cluster). Based on the simulation results shown in Figs. 12 to 14, the query results retrieved from only one closest cluster may not be the same as that of the ES. That is, the retrieved results may be incorrect if only one closest cluster is considered. In this study, for example, the query image will be compared with the five closest clusters within the 50 clusters, and the query results are the same as that of the exhaustive search approach.
Fig. 11 Query results for a DCT-compressed image using weighting coefficient set 2: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.

Fig. 12 Query results for an uncompressed image retrieved from only one closest cluster within the 50 clusters of the 10,000 uncompressed images: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.

Fig. 13 Query results for an uncompressed image retrieved from two closest clusters within the 50 clusters of the 10,000 uncompressed images: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.
Fast indexing and searching strategies...

Finally, referring to Eq. (8), different query (searching) approaches may return slightly different results with different processing times. A “good” query (searching) approach will return the same result as that returned by the exhaustive search approach. Based on the “goodness” (precision) criterion, the query (searching) approach proposed by Jain and Vailaya⁶ is one of the fast searching approaches, whereas their query approach is not a “good” query approach (Figs. 12 to 14). Based on the performance measure (the processing time) and the “goodness” (precision) criterion, the proposed query (searching) approaches are “fast” and “good” query (searching) approaches. The four existing approaches for comparison are all “good” query approaches based on the “goodness” (precision) criterion, but the proposed approaches outperform (are faster than) the four existing approaches for comparison, based on the performance measure (the processing time).

In this paper, fast indexing and searching strategies for image database systems were proposed. A set of proposed inequalities based on the weighted square sum error distance is used to speed up both the image indexing (clustering) and searching processes for feature-based image database systems. Based on the simulation results obtained in this paper, in terms of performance measure, namely, the processing time for indexing and searching in feature-based image database systems, the proposed approaches are superior to several existing approaches for comparison. In particular, based on the test image databases in this study, the percentage of improvement of the proposed approach with respect to the HT approach⁷ can achieve about 87.14 and 88.82 and 26.67 and 61.29% for 145-D uncompressed feature vectors and 18-D compressed feature vectors using two weighting coefficients sets, respectively.

Due to the fast that the performance of the proposed approach depends on the employed image features and the selected weighting coefficients, different weighting coefficients will lead to different clustering and searching results. Because the selection of the weighting coefficients is usually application-dependent, only some sets of coefficients were tested here. If the image features extracted are good enough and the weighting coefficients are adjusted properly, the proposed approaches will find the expected results efficiently. In addition, the proposed approach is not only effective for indexing and searching in feature-based image database systems, but is also suitable for any other multi-dimensional data indexing and searching based on the weighted square sum error distance or the square sum error distance, such as closest code word search and codebook design in VQ.

Fig. 14 Query results for an uncompressed image retrieved by the exhaustive search from the 10,000 uncompressed images: (a) the query image and the first retrieved image and (b) to (j) the retrieved images from ranks 2 to 10.

Acknowledgments

This work was supported in part by National Science Council, Republic of China under Grants NSC 89-2213-E-194-026 and NSC 90-2213-E-194-039.

References


