Image Databases: Search and Retrieval of Digital Imagery
Edited by Vittorio Castelli, Lawrence D. Bergman

ISBNs: 0-471-32116-8 (Hardback); 0-471-22463-4 (Electronic)

16 Compressed or Progressive Image Search

SETHURAMAN PANCHANATHAN
Arizona State University, Tempe, Arizona

16.1 INTRODUCTION

The explosive growth of multimedia data, both on the Internet and in domain-specific applications, has produced a pressing need for techniques that support efficient storage, transmission, and retrieval of such information. Indexing (Chapters 7, 14, and 15) and compression (Chapter 8) are complementary methods that facilitate storage, search, retrieval and transmission of imagery, and other multimedia data types.

The voluminous nature of visual information requires the use of compression techniques, in particular, of lossy methods. Several compression schemes have been developed to reduce the inherent redundancy present in visual data. Recently, the International Standards Organization and CCITT have proposed a variety of standards for still image and video compression. These include JPEG, MPEG-1, MPEG-2, H.261, and H.263. In addition, compression standards for high-resolution images, content-based coding, and manipulation of visual information, such as JPEG 2000 and MPEG-4, have been recently defined. These standardization efforts highlight the growing importance of image compression.

Typically, indexing and compression are pursued independently (Fig. 16.1), and the features used for indexing are extracted directly from the pixels of the original image. Many of the current techniques for indexing and navigating repositories containing compressed images require the decompression of all the data in order to locate the information of interest. If the features were directly extracted in the compressed domain, the need to decompress the visual data and to apply pixel-domain-indexing techniques would be obviated. Compressed-domain indexing (CDI) techniques are therefore important for retrieving visual data stored in compressed format [1,2].

As the principal objectives of both compression and indexing are extraction and compact representation of information, it seems obvious to exploit the

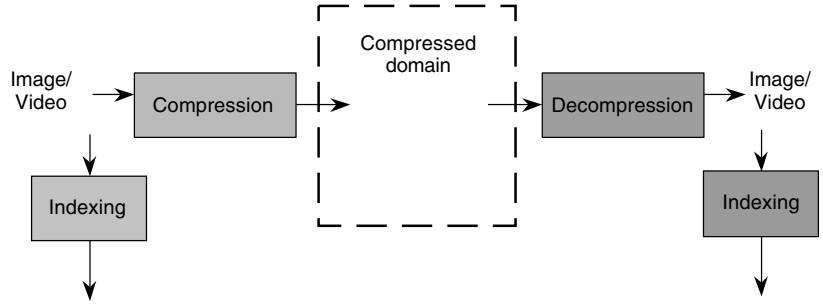


Figure 16.1. Pixel domain indexing and compression system.

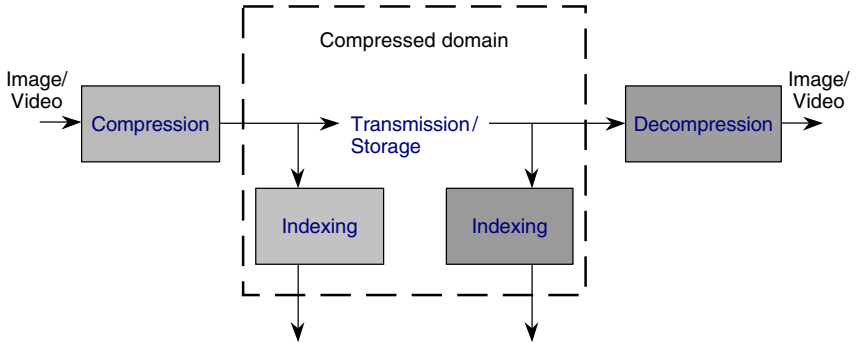


Figure 16.2. Compressed domain indexing system.

commonalties between the two approaches. CDI has the inherent advantage of efficiency and reduced complexity. A straightforward approach to CDI (Fig. 16.2) is to apply existing compression techniques and to use compression parameters and derived features as indices.

The focus of this chapter is indexing of image data. These techniques can often be applied to video indexing as well. A variety of video-indexing approaches have been proposed, in which the spatial (i.e., within-frame) content is indexed using image-indexing methods and the temporal content (e.g., motion and camera operations) is indexed using other techniques.

The following section analyzes and compares image-indexing techniques for compression based on the discrete Fourier transform (DFT), the Karhunen-Loeve transform (KLT), the discrete cosine transform (DCT), wavelet and subband coding transforms, vector quantization, fractal compression, and hybrid compression.

16.2 IMAGE INDEXING IN THE COMPRESSED DOMAIN

CDI techniques (summarized in Table 16.1) can be broadly classified into two categories: transform-domain and spatial-domain methods. Transform-domain

techniques are generally based on DFT, KLT, DCT, subband, or wavelet transforms. Spatial-domain techniques include vector quantization (VQ) and fractal-based methods. We now present the details of the various compressed-domain image-indexing techniques along with derived approaches.

Table 16.1. Compressed Domain Indexing Approaches

Technique	Characteristics/Advantages	Disadvantages	References
	Transform Domain		
Discrete Fourier transform	Uses complex exponentials as basis functions. The magnitudes of the coefficients are translation invariant. Spatial domain correlation can be computed by the product of the transforms.	Lower compression efficiency.	[3,4]
Discrete cosine transform	Uses real sinusoidal basis functions Has energy compaction efficiency close to optimal KLT.	Block DCT produces blocking artifacts.	[5,6]
Karhunen-Loeve transform	Employs the 2nd order statistical properties of an image for coding. Provides maximum energy compaction among linear transformations. It minimizes the mean-square error for any image among linear transformations.	Is data dependent: basis images for each subimage has to be obtained, and hence has high computational cost.	[7]
Discrete wavelet transform	Numerous basis functions exist. There is no blocking of data as in DCT. Yields, as by-product, a multiresolution pyramid. Better adaptation to nonstationary signals. High decorrelation and energy compaction.	Chip-sets for real-time implementation is not readily available.	[8,9]

(*Continued overleaf*)

Table 16.1. (*Continued*)

Technique	Characteristics/Advantages	Disadvantages	References
	Spatial Domain		
Vector quantization	Fast decoding. Reduced decoder hardware requirements makes it attractive for low power applications. Asymptotically optimum for stationary signals.	A codebook has to be available at both the encoder and decoder, or has to be transmitted along with the image. Encoding and codebook generation are highly complex.	[10]
Fractals	Exploits self-similarity to achieve compression. Potential for high compression.	Computationally intensive, hence hinders real-time implementation.	[11]

16.2.1 Discrete Fourier Transform

The Fourier transform is an important tool in signal and image processing [3,4,7]. Its basis functions are complex exponentials. Fast algorithms to compute its discrete version (DFT) can be easily implemented in hardware and software. From the viewpoint of image compression, the DFT yields a reasonable coding performance, because of its good energy-compaction properties.

Several properties of the DFT are useful in indexing and pattern matching. First, the magnitudes of the DFT coefficients are translation-invariant. Second, the cross-correlation of two images can be efficiently computed by taking the product of the DFTs and inverting the transform.

16.2.1.1 Using the Magnitude and Phase of the Coefficients as Index Key. A straightforward image-indexing approach is to use the magnitude of the Fourier coefficients as an index key. The Fourier coefficients are scanned in a zigzag fashion and normalized with respect to the size of the corresponding image. In order to enable fast retrieval, only a subset of the coefficients (approximately 100) from the zigzag scan is used as a feature vector (index). The index of the query image is compared with the corresponding indices of the target images stored in the database, and the retrieved results are ranked in decreasing order of similarity. The most commonly used similarity metrics are the mean absolute difference (MAD) and the mean square error (MSE).

We now evaluate the retrieval performance of this technique using a test database containing 500 images of wildlife, vegetation, natural scenery, birds, buildings, airplanes, and so forth. When the database is assembled, the target images undergo a Fourier transform, and the coefficients are stored along with the image data. When a query is submitted, the Fourier transform of the template

image is also computed, and the required coefficients extracted. Figure 16.3 shows the retrieval results corresponding to a query (top) image using the first 100 coefficients and the MAD error criterion. The 12 highest ranked images are arranged from left to right and top to bottom. It can be seen that the first and second ranked images are similar to the query image, as expected. However, it is immediately clear that a human would rank the 12 images returned by the system quite differently.

Note that in the example of Figure 16.3, the feature vector is composed of low-frequency Fourier coefficients. Consequently, images having similar average intensity values are considered similar by the system. However, if edge information is relevant, higher frequency coefficients and features derived from them should be used. Additionally, the direction and angle information is embedded in the phase component of the Fourier transform, rather than in its magnitude. The phase component and the high-frequency coefficients are therefore useful in retrieving images that have similar edge content. Figure 16.4 is the result of a query on a database in which indexing relies on directional features extracted from the DFT. Note that the retrieved images either contain a boat, such as the query image, or depict scenes with directional content similar to that of the query image. The union of results based on these low and high frequency component features has the potential for good overall retrieval performance.

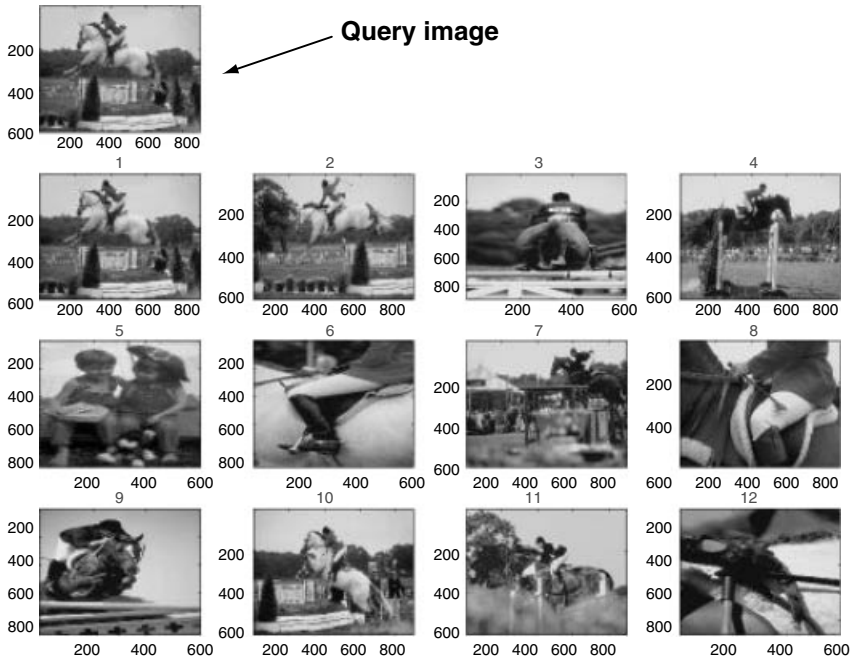


Figure 16.3. Query results when the indexing key is the amplitude of the Fourier coefficients. The query image is shown at the top. The top 12 matches are sorted by similarity score and displayed in left-to-right, top-to-bottom order.

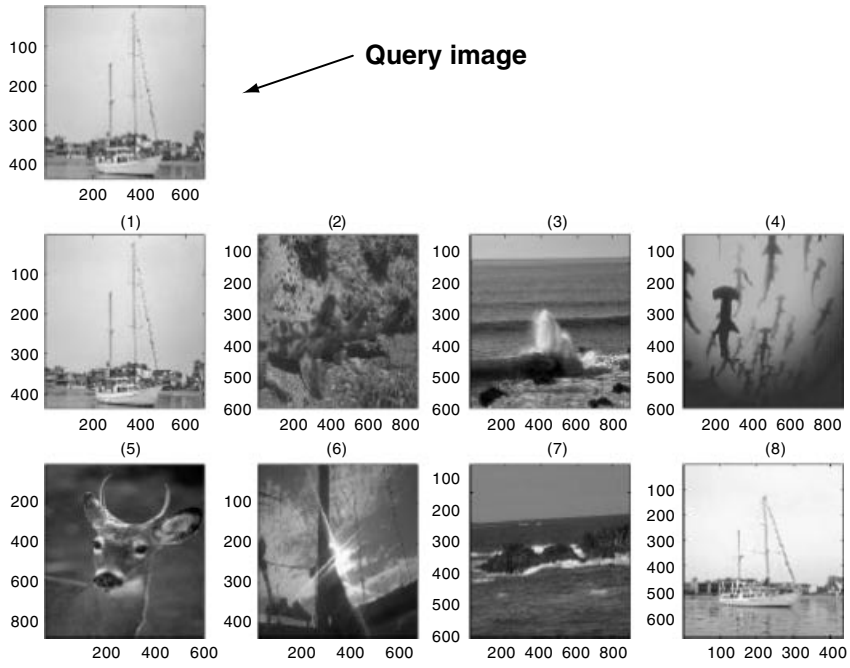


Figure 16.4. Query example based on the angular information of the Fourier coefficients.

16.2.1.2 Template Matching Using Cross-Correlation. Template matching can be performed by computing the two-dimensional cross-correlation between a query template and a database target, and analyzing the value of the peaks. A high value of a cross-correlation denotes a good match. Although cross-correlation is an expensive operation in the pixel domain, it can be efficiently performed in the Fourier domain, where it corresponds to a product of the transforms. This property has been used as the basis for image indexing schemes. For example, Ref. [12] discusses an algorithm in which the threshold used for matching based on intensity and texture is computed. This threshold is derived using the cross-correlation of Fourier coefficients in the transform domain.

16.2.1.3 Texture Features Derived from the Fourier Transform. Several texture descriptors based on FT have been proposed. Augusteijn, Clemens, and Shaw [13] evaluated their effectiveness in classifying satellite images. The statistical measures include the maximum coefficient magnitude, the average coefficient magnitude, the energy of the magnitude, and the variance of the magnitude of Fourier coefficients. In addition, the authors investigated the retrieval performance based on the radial and angular distribution of Fourier coefficients. They observed that the radial and angular measures provide good classification performance when a few dominant frequencies are present. The statistical measures provide a satisfactory performance in the absence of dominant

frequencies. Note that the radial distribution is sensitive to texture coarseness, whereas the angular distribution is sensitive to the directionality of textures. The performance of the angular distribution of Fourier coefficients for image indexing has been evaluated in Ref. [14]. Here, the images are first preprocessed with a low-pass filter, the FFT is calculated, and the FFT spectrum is then scanned by a revolving vector exploring a 180° range at fixed angular increments. The angular histogram is finally calculated by computing, for each angle, the sum of the image-component contributions. While calculating the sum, only the middle-frequency range is considered, as it represents visually important image characteristics. The angular histogram is used as an indexing key. This feature vector is invariant with respect to translations in the pixel domain, but not to rotations in the pixel domain, which correspond to circular shifts of the histogram.

There have been numerous other applications of the Fourier transform to indexing. The earlier-mentioned techniques demonstrate the potential for combining compression and indexing for images using FT.

16.2.2 Karhunen-Loeve Transform (KLT)

The Karhunen-Loeve transform (principal component analysis) uses the eigenvectors of the autocorrelation matrix of the image as basis functions. If appropriate assumptions on the statistical properties of the image are satisfied, the KLT provides the maximum energy compaction of all invertible transformations. Moreover, the KLT always yields the maximum energy compaction of all invertible *linear* transformations. Because the KLT basis functions are not fixed but image-dependent, an efficient indexing scheme consists of projecting the images onto the K-L space and comparing the KLT coefficients.

KLT is at the heart of a face-recognition algorithm described in Ref. [15]. The basis images, called *eigenfaces*, are created from a randomly sampled set of face images. To construct the index, each database image is projected onto each of the eigenfaces by computing their inner product. The result is a set of numerical coefficients, interpreted as the coordinates of the image in the eigenface-reference system. Hence, each image is represented as a point in a high-dimensional Euclidean space. Similarity between images is measured by the Euclidean distance between their representative points.

During query processing, the coordinates of the query image are computed, the closest indexed representative points are retrieved, and the corresponding images returned to the user. The user can also specify a distance threshold to control the allowed dissimilarity between the query image and the results.

It is customary to arrange the eigen-images in decreasing order of the magnitude of the corresponding eigenvalue. Intuitively, this means that the first eigenimage is the direction along which the images in the database vary the most, whereas the last eigenimage is the direction along which the images in the database vary the least. Hence, the first few KLT coefficients capture the most salient characteristics of the images. These coefficients are the *most expressive*

features (MEFs) of an image, and can be used as index keys. The database designer should be aware of several limitations in this approach. First, eigenfeatures may represent aspects that are unrelated to recognition, such as the direction of illumination. Second, using a larger number of eigenfeatures does not necessarily lead to better retrieval performances. To address this last issue, a discriminant Karhunen-Loeve (DKL) projection has been proposed in Ref. [16]. Here, the images are grouped into semantic classes, and KLT coefficients are selected to simultaneously maximize between-class scatter and minimize within-class scatter. DKL yields a set of *most discriminating features* (MDFs). Experiments suggest that DKL results in an improvement of 10 to 30 percent over KLT on a typical database.

16.2.2.1 Dimensionality Reduction Using KLT. KLT has also been applied to reduce the dimensionality of texture features for classification purposes, as described, for instance, in Ref. [17]. Note that KLT is generally not used in traditional image coding (i.e., compression) because it has much higher complexity than competitive approaches. However, it has been used to analyze and encode multispectral images Ref. [18], and it might be used for indexing in targeted application domains, such as remote sensing.

16.2.3 Discrete Cosine Transform

DCT, a derivative of DFT, employs real sinusoids as basis functions, and, when applied to natural images, has energy compaction efficiency close to the optimal KL Transform. Owing to this property and to the existence of efficient algorithms, most of the international image and video compression standards, such as JPEG, MPEG-1, and MPEG-2, rely on DCT for image compression.

Because block-DCT is one of the steps of the JPEG standard and as most photographic images are in fact stored in JPEG format, it seems natural to index the DCT parameters. Numerous such approaches have been proposed in the literature. In the rest of this section, we present in detail the most representative DCT-based indexing techniques and briefly describe several related schemes.

16.2.3.1 Histogram of DC Coefficients. This technique uses the histogram of the DC image parameters as the indexing key. The construction of the DC image is illustrated in Figure 16.5. Each image is partitioned into nonoverlapping blocks of 8×8 pixels, and each block is transformed using the two-dimensional DCT. Each resulting 8×8 block of coefficients consists of a DC value, which is the lowest frequency coefficient and represents the local average intensity, and of 63 AC values, capturing frequency contents at different orientations and wavelengths. The collection of the DC values of all the blocks is called the DC image. The DC image looks like a smaller version of the original, with each dimension reduced by a factor of 8. Consequently, the DC image also serves as a thumbnail version of the original and can be used for rapid browsing through a catalog.

The histogram of the DC image, which is used as a feature vector, is computed by quantizing the DC values into N bins and counting the number of coefficients

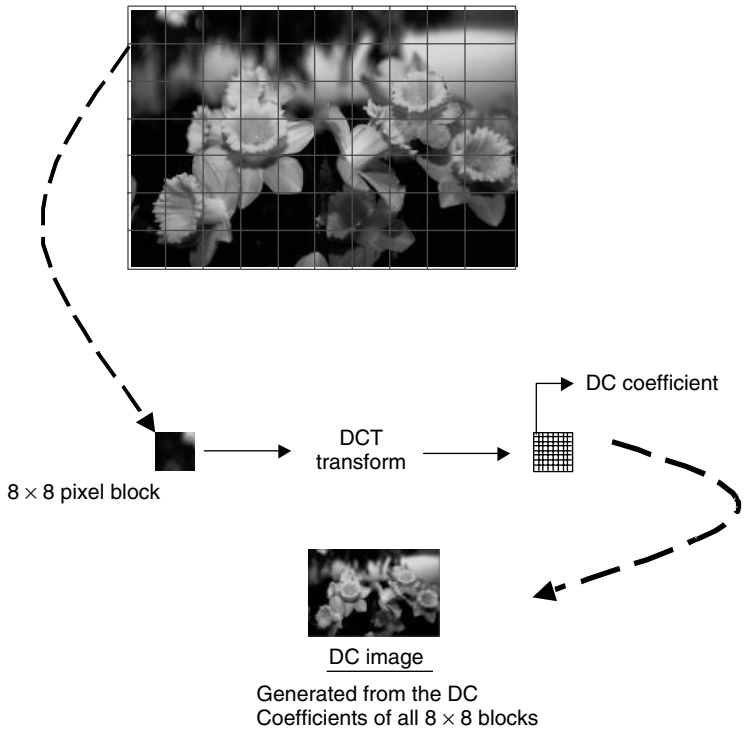
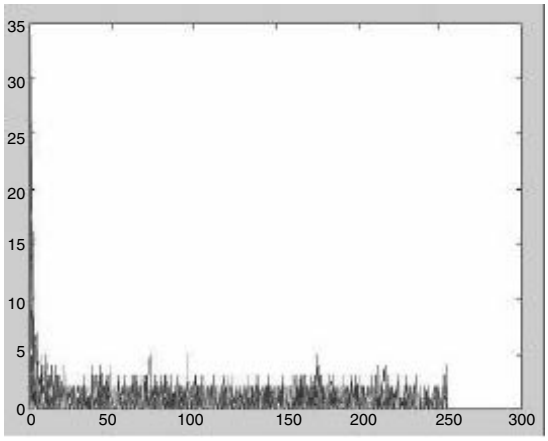


Figure 16.5. DC image derived from the original image through the DCT transform.

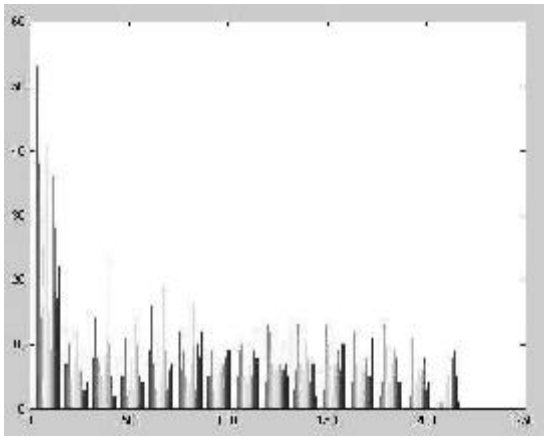
that fall within each bin. It is then stored and indexed in the database. In the example shown in Figure 16.6, 15 bins have been used to represent the color spectrum of the DC image. These values are normalized in order to make the feature vectors invariant to scaling.

When a query is issued, the quantized histogram of the DC image is extracted from the query image and compared with the corresponding feature vectors of all the target images in the database. Similarity is typically computed using the histogram intersection method (see Chapter 11) or the weighted distance between the color histograms [19]. The best matches are then displayed to the user as shown in Figure 16.7. As histograms are invariant to rotation and have been normalized, this method is invariant to both rotation and scaling.

16.2.3.2 Indexing with the Variance of the AC Coefficients. A feature vector of length 63 can be constructed by computing the variance of the individual AC coefficients across all the 8×8 DCT blocks of the image. Because natural images contain mostly low spatial frequencies, most high-frequency variances will be small and play a minor role in the retrieval. In practice, good retrieval performance can be achieved by relying just on the variances of the first eight AC coefficients. This eight-component feature vector represents the overall texture of



(a)



(b)

Figure 16.6. **(a)** Histogram of DC image; **(b)** Binwise quantized DC image.

the entire image. Figure 16.8 shows some retrieval results using this approach. It is worthwhile noting that the runtime complexity of this technique is smaller than that of traditional transform features used for texture classification and image discrimination, as reported in Ref. [20].

16.2.3.3 Indexing with the Mean and Variance of the Coefficients. A variety of other DCT-based indexing approaches have appeared in recent literature. A method based on the DCT transform of 4×4 blocks, which produces 16 coefficients per block, is described in Ref. [20]. The variance and the mean of the absolute values of each coefficient are calculated over the blocks spanning the entire image. This 32-component feature vector represents the texture of

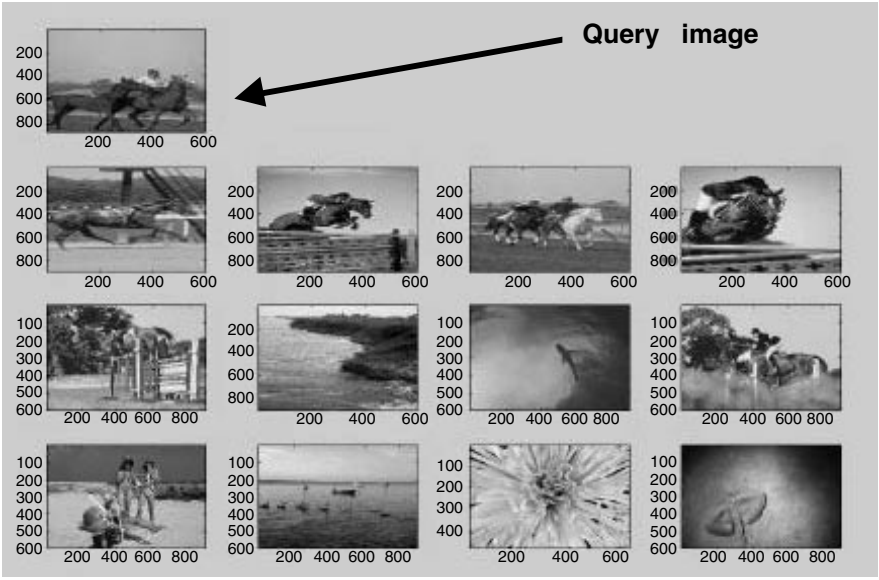


Figure 16.7. Image retrieval based on the histogram of DC image.

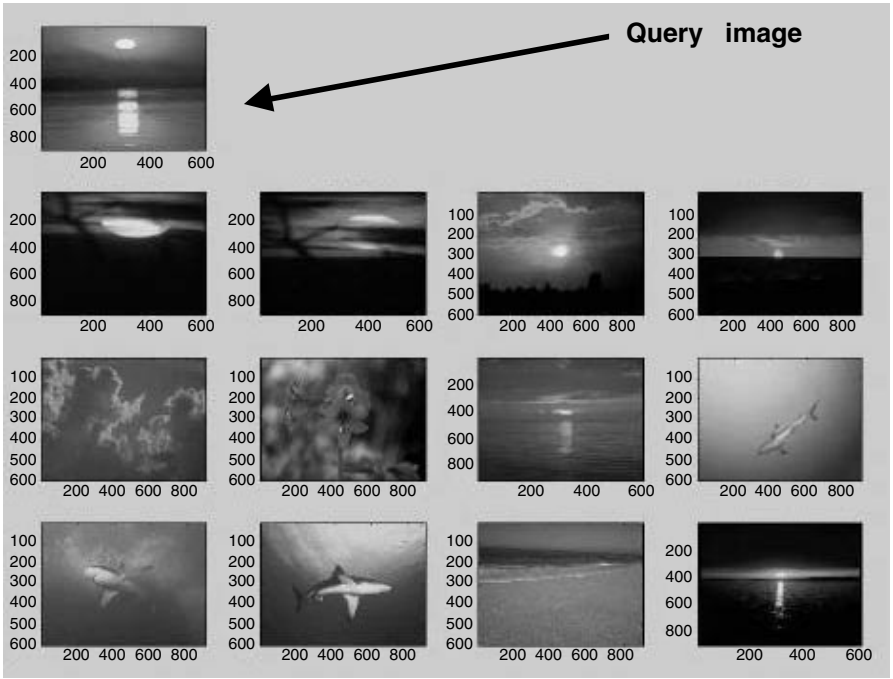


Figure 16.8. Retrieval based on the variance of the first eight AC coefficients of images.

the whole image. A Fisher discriminant analysis (FDA) is used to reduce the dimensionality of the feature vector, which is then used for indexing.

16.2.3.4 Combining DCT Coefficients with Spatial Information. A technique for image retrieval using JPEG has been detailed in Ref. [21]. This technique is based on the mutual relationship between the DCT coefficients of unconnected regions in both the query image and the target image. The image spatial plane is divided into 2 K windows; the size of the windows is a function of the size of the image and is selected to be the smallest multiple of 8 that is less than the initial window size. These windows are randomly paired into K pairs. The average of each DCT coefficient from all the 8×8 JPEG blocks in each window is computed. The DCT coefficients of the windows in the same pair are compared. If the DCT coefficient in one window is greater than the corresponding one in the other window, a "1" is assigned, otherwise a "0" is assigned. Each window pair yields 64 binary values, and therefore each image is represented by a binary feature vector of length 64^*K. The similarity of the query and target images can be determined by employing the Hamming Distance of their binary feature vectors.

16.2.3.5 Comparing Edges Using DCT. Many content-based indexing and retrieval methods rely on the discriminating power of edge information: similar images often have similar edge content. A technique to detect oriented line features using DCT coefficients has been presented in Ref. [22]. This technique is based on the observation that predominantly horizontal, vertical, and diagonal features produce large values of DCT coefficients in vertical, horizontal, and diagonal directions, respectively. It has been noted that a straight line of slope m in the spatial domain generates a straight line with a slope of approximately $1/m$ in the DCT domain. The technique can be extended to search for more complex features composed of straight-line segments. A DCT-based approach to detect edges in regions of interest within an image has been discussed in Ref. [23]. Here the *edge orientation, edge offset from center*, and *edge strength* are estimated from the DCT coefficients of an 8×8 block. The orientations are horizontal, vertical, diagonal, vertical-dominant, and horizontal-dominant. Experimental results presented in [23] demonstrate that the DCT-based edge detection provides a performance comparable to the Sobel edge-detection operator (see the chapter on Image Analysis and Computer Vision in [7]).

16.2.4 Subbands/Wavelets/Gabor Transform

Recently, subband coding and the discrete wavelet transforms (DWT) have become popular in image compression and indexing applications [8,24] (Chapter 8 describes in detail the DWT and its use for compression). These techniques recursively pass an image through a pair of filters (one low pass and the other high pass), and decimate the filter outputs (i.e., discard every other sample) in order to maintain the same data rate as the input signal. As the length

of the filter outputs is halved after each iteration, this decomposition is often called dyadic. Two-dimensional signals (images) are commonly encoded with separable filters. This means that one step of the decomposition (the combination of filtering and downsampling) is performed independently on each row of the image, and then the same step is performed on each column of the transformed-rows matrix. This process produces four subbands, labeled LL, LH, HL, and HH, respectively, to denote which filters (L = low pass, H = high pass) were applied in the row and column directions. If more levels are desired, separable filtering is applied to the subband obtained by low-pass filtering both rows and columns. An interesting feature of these transforms is that they are applied to the entire image, rather than to blocks as in JPEG. Consequently, there are no blocking artifacts in images that undergo lossy compression with wavelets or subband coding (see Chapter 8 for a discussion of artifacts introduced by lossy compression schemes and of the advantages of DWT over the previously mentioned schemes).

The difference between DWT, subband coding, and the wavelet-packets transform lies in the recursive application of the combination of filtering and downsampling. The DWT recursively decomposes only the LL subband; subband filtering recursively decomposes all four subbands; the wavelet-packets transform lies in the middle and adaptively applies the decomposition to subbands where it yields better compression. An example image along with the corresponding two level wavelet transform is shown in Figure 16.9.

The Gabor transform (described also in Chapter 12) is similar to the wavelet transform, but its basis functions are complex Gaussians, and hence have optimal time-frequency localization [9]. Unlike the wavelet transform, which uses a complete set of basis functions (i.e., the number of coefficients of the transform is the same as the number of pixels in the image, and the transform is invertible), the Gabor transform is in general overcomplete, that is, redundant. Several indexing approaches based on Subband, Wavelet, and Gabor transforms have appeared in recent literature.

Figure 16.9. Original lena image and the corresponding two-level wavelet decomposed image.

16.2.4.1 Direct Comparison of Wavelet Coefficients. An indexing technique based on direct comparison of DWT coefficients is presented in Ref. [25]. Here, all images are rescaled to 128×128 pixels, and transformed. The average color, the sign (positive or negative), and the positions of the M DWT coefficients having the largest magnitude (the authors have used $M = 40$ to 60) are calculated for each image and become the index key. Good indexing performance has been reported. However, the index is dependent on the location of DWT coefficients. Hence, translated or rotated versions of the query image may not be retrieved using this technique.

16.2.4.2 Indexing on the Lowest-Resolution Subbands. A technique similar to that of Ref. [25] has been detailed in Ref. [26]. All the images are rescaled to 128×128 pixels and transformed with a four-level DWT. Only the four lowest-resolution subbands (having size 8×8 pixels), denoted by S_L (low pass), S_H (horizontal band), S_V (vertical band), and S_D (diagonal band), as shown in Figure 16.10 are considered for indexing. Image matching is then performed using a three-step procedure. In the first stage, 20 percent of the images are retrieved using the variance of the S_L band. In the second stage, pruning is performed using the difference between the S_L coefficients of the query and of the target images. Finally, the differences between the S_L, S_H, S_V, and S_D coefficients of the query and target images are used to select the query results. For color images, this procedure is repeated on all three-color channels. The complexity of this technique is small due to its hierarchical nature. Although it provides a performance improvement over the techniques detailed in Ref. [26], this indexing scheme is still not robust to translation and rotation.

16.2.4.3 Histogram of the Wavelet Transform. This section describes a wavelet-based indexing technique that relies on the histogram of the wavelet transform, called the wavelet histogram technique (WHT). To motivate the approach, we note that comparing histograms in the pixel domain may result in rather erroneous retrievals. As an example, consider Fig. 16.11, which shows two images, one of a lady and the other of an owl. The spatial histograms of these images are very similar. A wavelet histogram, however, uses the wavelet coefficients of the high-pass bands, which contain discriminative textural properties such as directionality.

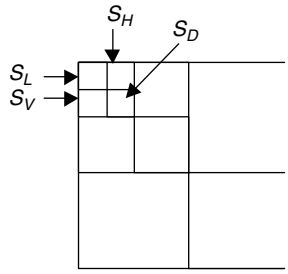


Figure 16.10. Four-stage wavelet decomposition showing S_L, S_H, S_V, and S_D.

Lady Owl

Figure 16.11. Images with similar histograms.

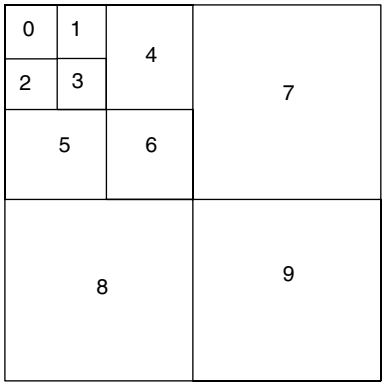


Figure 16.12. Three-stage wavelet decomposition showing the nine high-pass bands.

In the WHT technique, a three-level wavelet transform of the image is computed as shown in Figure 16.12 (which has 10 subbands), and a 512-bin histogram is generated from the coefficients in the nine high-pass subbands.

It is known that when a texture image is transformed into the wavelet domain, most of the energy characterizing the texture is compacted into the high frequency subbands. The WHT is therefore used to discriminate among different global textures for it captures the energy content in the higher frequency subbands. Queries are executed by comparing the histograms of the query image with those of the images in the database. The wavelet histogram technique yields better retrieval performance than techniques relying on pixel-domain histograms. The only drawback of this method is that it may not be well suited for natural images that contain large objects as these objects tend to have the same textural properties at the global level.

Figure 16.13 compares the histogram of the wavelet coefficients of the two images shown in Figure 16.11. It is immediately apparent that the histograms are rather different and that the corresponding images would not have been declared similar by a WHT-based matching technique.

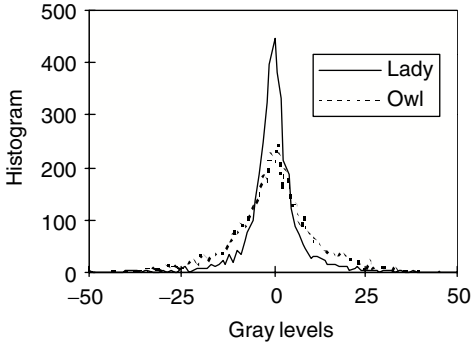


Figure 16.13. Wavelet histogram of the images in Fig. 16.11.

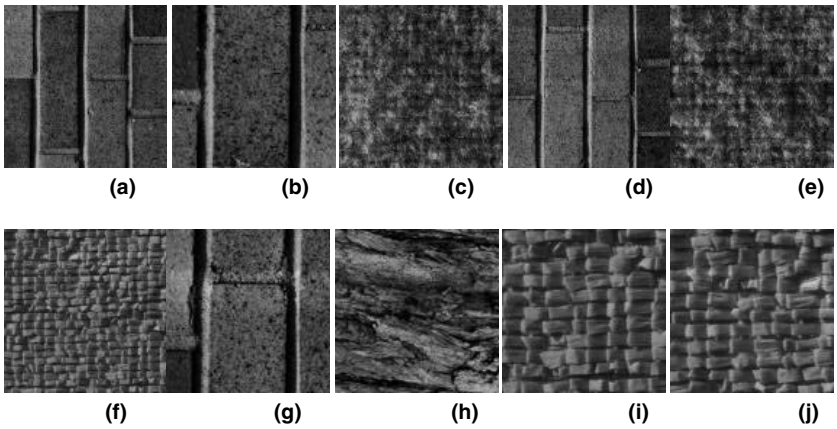


Figure 16.14. Images retrieved using WHT: **(a)** query image, **(b)**–**(j)** retrieved images in monotonically increasing order of distance.

An example of texture retrieval using WHT is shown in Figure 16.14. Here, a brick texture (Fig. 16.14a) is used as the query image. The database contains seven other similar brick textures (three of them have the same scale and four of them have a lower scale). The nine retrieved images (Fig. 16.14b to j) contain one brick texture with the same scale (Fig. 16.14d) and two brick textures with a lower scale (Fig. 16.14b and Fig. 16.14g).

Generating the wavelet histogram is a computationally demanding process. All high-pass bands must be upsampled and filtered using wavelet synthesis filters (the filters for the direct DWT are called "analysis filters," whereas those for the inverse DWT are called "synthesis filters"; in general, analysis and synthesis filters are different) to produce an image of the same size as the original, before the histogram can be computed. The process is repeated for each of the

subbands. Also, comparing large histograms (with 512 bins) is an expensive operation to perform at query-execution time. Comparing features derived from the histograms, such as moments, can achieve a substantial reduction in the complexity of WHT.

16.2.4.4 Combining Wavelet and Pixel-Domain Features. A joint moment and wavelet indexing technique, using the Legendre moments of the original image histogram along with the shape and variance parameters of the wavelet subimages, has been presented in Ref. [27]. Substantial reduction in complexity can be achieved by comparing moments instead of directly comparing histograms. This indexing technique combining moments and wavelets provides a 4 percent improvement over the technique that uses only the Legendre moments.

16.2.4.5 Comparing the Histograms of Individual Subbands. Similar images have similar directional information. Directionality is preserved under translation and small rotations: for example, pictures of the same scene obtained from similar viewing angles have similar directional content. A technique that captures directionality by relying on the wavelet transform has been presented in Ref. [28]. The different subbands of the wavelet transform capture different directional content. In a one-level transform, LL, HL, LH, and HH denote the four subbands, where the first letter refers to the filter used to transform the individual rows, and the second letter refers to the filter used to transform the columns (L = low pass, H = high pass). The HL subband captures vertical lines, the LH subband captures horizontal lines, and the HH subband captures diagonal lines. When the transform has multiple levels, each subband captures directional information at a different scale. Although different images might have similar overall wavelet histograms, they might have different subband statistics.

The complexity of directly comparing the histograms of all the subbands is high and can be substantially reduced by comparing parameters extracted from the histograms. The histograms of high-pass wavelet subbands can be modeled using generalized Gaussian density (GGD) functions [29], which are expressed in terms of two parameters — σ (standard deviation) and γ (shape parameter). If the target and query images are different, the two parameters are likely to be different. The "distance" between two images f and g can be expressed in terms of standard deviations and shape parameters as

$$d(f, g) = \sum_{k=1}^{Q} B_k(\gamma_{f_k} - \gamma_{g_k})^2 + \sum_{k=1}^{Q} A_k(\sigma_{f_k} - \sigma_{g_k})^2, \tag{16.1}$$

where Q is the number of wavelet bands used for comparison and A_k and B_k are appropriate weights that are empirically determined to optimize the retrieval performance.

16.2.4.6 Rotationally Invariant Wavelet Transforms. A complex wavelet transform, in which the magnitude of the DWT coefficients is invariant under rotation, has been presented in Ref. [30]. The mother wavelet, which is the basic wavelet not subjected to any scaling or translation, is defined in polar coordinates. An experiment on a set of English character images shows that this technique performs better than complex Zernike moments [31,32], the magnitude of which is rotation invariant.

Nonlinear wavelet transforms that are invariant under scale, rotation, and shift (SRS) transformations have been described in Ref. [33]. The paper also contains an algorithm that adaptively changes the mother function according to the input signal to provide SRS invariance. This technique of changing the basis function is called dilation. The concept of *super wavelets* has also been introduced in the same paper, using a linear combination of adaptive wavelets that is subjected to dilation, thereby handling the scale changes in the signal.

16.2.4.7 Separating Edges and Texture Information. An image coding technique that separates edge information from texture information has been detailed in Ref. [9]. Edges at different levels of decomposition, also called multiscale edges, along the main directions (horizontal, vertical, or diagonal) are detected from the local maxima of the wavelet transform subbands. The image is reconstructed from the wavelet subbands using synthesis filters and an error image is computed by subtracting the reconstructed image from the original image. The error image thus obtained provides a significant amount of texture information. The textures are coded with a standard orthogonal wavelet transform. The multiscale edges have the potential to provide a good indexing performance.

16.2.4.8 Texture Indexing Using Irregular Tree Decomposition. A texture analysis scheme using an irregular tree decomposition, in which the middle resolution subband coefficients are used for texture matching, has been presented in Ref. [34]. In this scheme, a J dimensional feature vector is generated consisting of the energy of J subbands that contain most of the energy. Indexing is done by matching the feature vector of the query image with those of the target images in a database. For texture classification, superior performance can be obtained by training the algorithm. Such a method of classification is called *supervised classification*. Here, for each representative class of textures that is known a priori, the training process finds the most important subbands in terms of energy content. A query image can then be categorized into one of the texture classes by employing a suitable distance measure between the feature vectors of the query image and the representative classes.

16.2.4.9 Describing Texture Using the Subband Energy. In Ref. [20] the global texture properties of images are described using the energy of DWT subbands. Images are transformed with a five-level DWT, producing 16 subbands, and the variance and mean absolute value within each subband are computed. The result is a 32-dimensional texture vector. The method

was compared to three alternatives, namely, straight subband coding with 16 subbands, Mandala DCT [35,36] (standard block DCT in which the coefficients are rearranged into blocks having the same frequency content), and straight blocking in the pixel domain.

Comparison of the four methods was performed on images from the Brodatz set, which are examples of uniform textures. The goal of the experiment was to determine which feature set is better for texture classification. Dimensionality reduction was performed using standard Fisher Discriminant Analysis, and dissimilarity between images was measured in terms of the Mahalanobis distance (see Chapter 14.2.3) between the feature vectors. The experiments in Ref. [20] showed that wavelet- and subband-coding-derived features perform significantly better than Mandala DCT and spatial domain blocking. More recent results [37,38], however, show that, at least for scientific data sets, features derived from the Gabor wavelets and texture features computed from the gray scale levels of the images significantly outperform this method.

16.2.4.10 Combining Wavelet and Hidden Markov Models. A two-stage, rotation- and gray scale transformation-invariant texture recognition technique that combines wavelets and hidden Markov models (HMM) has been discussed in Ref. [39]. In the first stage, gray scale transformation-invariant features are extracted from each subband. In the second stage, the sequence of subbands is modeled as an HMM. Texture is represented by a set of texture classes. Each texture class is associated with a specific HMM. The HMM is used to model the statistical dependence among these subbands and is able to capture changes caused by rotation. During recognition, the unknown texture is matched against all the models in a fixed collection and the model with the best match identifies the texture class.

16.2.4.11 Robustness to Illumination Changes Using Wavelet Histograms. Most of the indexing algorithms presented in the literature assume that the illumination levels of the images are similar. The indexing performance may substantially degrade if this is not the case.

A technique that is robust to changes in illumination and relies on the histogram of the wavelet transform is discussed in Ref. [40]. Changes in illumination level are estimated using scale-invariant moments of the wavelet histogram. The parameters s and g of each subband (Section 16.2.4.5) of the target image are modified to account for illumination changes. Matching can then be carried out using Eq. (16.1).

16.2.4.12 Describing Textures Using Gabor Wavelets. An indexing technique using Gabor wavelets (Chapter 12) is detailed in Ref. [41]. Each image is decomposed into four scales and six orientations. A feature vector, of dimension 48, is then formed using the mean and standard deviation within each subband. The similarity between the query image and a target image is determined by the similarity of their feature vectors. Gabor wavelets capture directionality better

than separable wavelets, which describe only vertical, horizontal, and diagonal (45 degrees) directions. However, the Gabor transform is computationally much more expensive than the dyadic, separable decomposition.

16.2.4.13 Computing Cross-Correlation in the Subband Domain. A correlation-based pattern matching approach operating in the subband domain is discussed in Ref. [42]. Cross-correlation between two images can be expressed in terms of cross-correlations of the corresponding subbands. Consider two one-dimensional signals $x(n)$ and $w(n)$, having z-transforms $W(z)$ and $X(z)$. It is well known, from elementary signal processing, that the cross-correlation of the signals can be expressed in the z-transform domain as $\widetilde{W}(z)X(z)$, where the tilde operator denotes the complex conjugate. Let the low-pass and high-pass analysis filters be $H_0(z)$ and $H_1(z)$, respectively. Denoting the decompositions of $x(n)$ and $w(n)$ into two subbands by $\{y_0(n), y_1(n)\}$ and $\{z_0(n), z_1(n)\}$, we can obtain the cross-correlation of $x(n)$ and $w(n)$ as:

$$\widetilde{W}(z)X(z) = \sum_{i,j=0}^{1} F_{ij}(z)\widetilde{Z}_i(z^2)Y_j(z^2) \tag{16.2}$$

where $F_{ij}(z) = H_i(z)\tilde{H}_j(z)$, $i, j = 0, 1$.

Eq. (16.2) shows that the cross-correlation of two signals equals the weighted sum of the cross-correlations of their corresponding subbands. Although Eq. (16.2) can still be computationally expensive, it was conjectured in Ref. [42] that a reasonable estimate of the correlation peaks can be obtained by just considering the subbands with highest energy. A characteristic of this technique is that the location of cross-correlation maxima (that correspond to matches) can be immediately derived from Eq. (16.2). This is an advantage over Fourier–Transform-based methods in which cross-correlation is very efficiently computed by coefficient-wise multiplication of the transforms but the inverse transform of this product must be computed to retrieve the locations and the values of the maxima.

16.2.4.14 Combining Image Coding and Indexing. A joint wavelet-based image representation and description system has been presented in Ref. [43]. In this system, images are compressed with a wavelet-coding technique and indexed using color, texture, and shape descriptions generated in the wavelet domain during the encoding process. All the features (texture, color, and shape) are based on the most significant wavelet coefficients and their energy distribution across subbands and decomposition levels. As the images are compressed and indexed at the same time, the image database management problem can be greatly simplified.

16.2.4.15 Remarks. In summary, wavelets are being increasingly considered as the leading candidates for compression in standards such as JPEG 2000 and

MPEG 4 due to their superior coding. Wavelet-based techniques serve the joint purposes of compression and indexing of visual content.

16.2.5 Vector Quantization (VQ)

Vector quantization (VQ) is an efficient technique for low bit rate image and video compression [44]. VQ pushes most of the complexity into the compressor, to allow for very efficient and simple decompression. It is possible to write very fast and efficient software decoders for VQ, which makes this scheme appealing for general-purpose computers. It is also possible to build simple and small hardware decoders, which makes VQ attractive for low-power applications such as portable video-on-demand over wireless networks.

VQ uses for compression a codebook containing $K = 2^{NR}$ code words. The codebook could be defined a priori, in which case it could be known to both the encoder and the decoder and would not need to be stored with the compressed data or constructed adaptively for each image, in which case it must be stored with the image. Each code word is an L-dimensional vector and has a unique label. The number L, N, and R are parameters of the algorithm, discussed later.

During compression, the image is first decomposed into nonoverlapping regular tiles containing L pixels each, which are represented as L-dimensional vectors. For example, the image can be tiled into 2×2 nonoverlapping blocks, and the pixels of each block arranged in a 4-dimensional vector. In a gray scale image in which the luminance is described by 8 bits, there are 2^{8L} possible such vectors. It is customary to write this number as 2^N, where N $(= 8L$ in this example) is the number of bits required to describe each possible vector. Each input vector is mapped onto the label of the closest code word as shown in Figure 16.15, using, for instance, the nearest-neighbor rule. Because there are 2^{NR} distinct code words, we need NR bits to represent each label, whereas to represent the original tile we need N bits. Hence, the number $1/R$ is the compression ratio

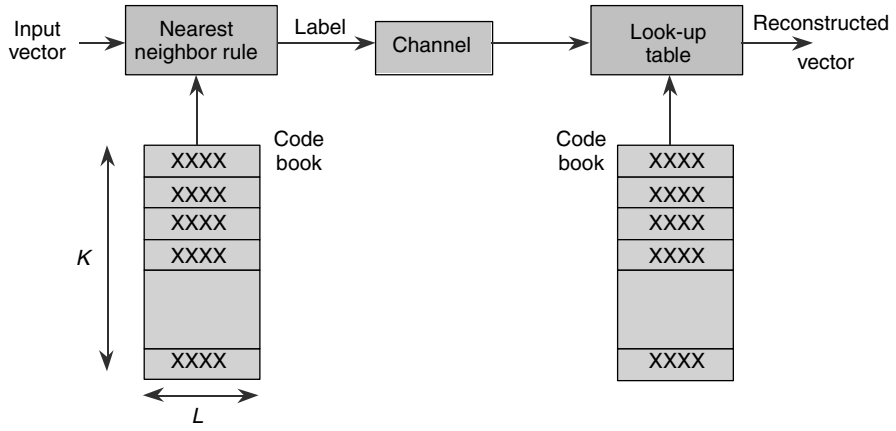


Figure 16.15. Compression and decompression using vector quantization (VQ).

(expressed as the ratio of the original image size to the compressed size), and R is called the *rate* of the quantizer.

Thus, after encoding, the image has been replaced by a sequence of code word labels, one per tile. Image reconstruction is implemented as a simple table lookup procedure: the code word label is used as an index into a table containing the code words; the corresponding code word is retrieved and written into the appropriate image tile.

For a fixed value of L, a larger codebook yields a lower compression ratio, makes encoding more computationally expensive, but provides on average smaller differences between the original and the reconstructed image. Note that the length L of the vector should also be carefully selected. Larger values of L correspond to larger block sizes: images are then divided in a smaller number of blocks, and hence more quickly decompressed; at the same time, larger codebooks are needed to yield the desired reconstruction performance, which complicates the codebook generation and the image encoding and, if the codebook is stored with the image, reduces the compression ratio.

The coding performance of VQ for different vector dimensions and code word sizes corresponding to two different compression ratios is shown in Figure 16.16.

VQ can also be used to construct a lower-resolution version of the original image. The set of tile labels can be represented as an image, called the "label map," and each label can be assigned an appropriate gray level (for example, the average of the entries of the corresponding code word) so that the label map resembles a scaled version of the original image. The effect of appropriately selecting labels for similar code words can be seen in the label map of Figure 16.17b, which looks like a scaled version of the original.

A sampling of VQ-based indexing techniques is now presented.

16.2.5.1 Indexing with the Histogram of the Label Map. In this technique the images are compressed using VQ, and the labels of the tiles are stored in the database [10]. The histogram of the labels serves as an index to the image. The histogram of the labels of an image is a K-dimensional vector (where $K = 2^{NR}$

(a) **(b)** **(c)**

Figure 16.16. **(a)** Original image, **(b)** Reconstructed image at a compression ratio 128:1, Block size: 8 by 8 pixels (64-D vector), codebook size: 16 code words, **(c)** Reconstructed image at a compression ratio 14:1 Block size: 4 by 4 pixels (16-D vector), Codebook size: 512 code words.

(a) Original image **(b)** Label map

Figure 16.17. Images and their label maps using ordered codebook VQ.

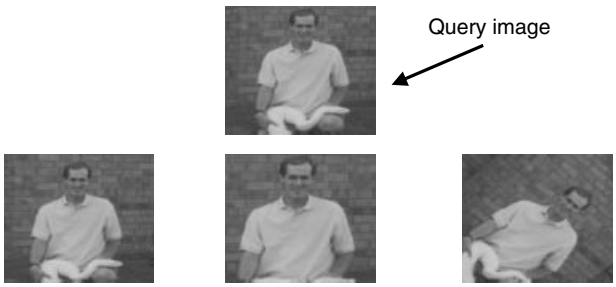


Figure 16.18. Retrieval using the histogram of labels approach in VQ compressed domain. The image at the top is the query image, and the results are arranged from left to right in decreasing matching score.

is the number of code words in the codebook), the ith entry of which contains number of tiles having label i. The similarity between two images is measured using the histogram intersection (INTR) or the Euclidean (EUCL) distance metric. The retrieval results for a query image from a database of 3000 images are shown in Figure 16.18.

For an image of size $X \times Y$ pixels, the computation of the gray-level histogram in the pixel domain has a complexity of $O(XY)$, whereas the computation of label histogram has a lower complexity of $O(XY/L)$ where L is the dimension of the code words.

16.2.5.2 The Universal Quantizer Usage Map as an Indexing Key. This technique is based on adaptive VQ using a large universal codebook [45]. Here, a map of the used code words is generated for each image and is stored along with the image in the database. To illustrate this, let us say the image to be coded is tiled and represented as a collection of L-dimensional vectors, which are then translated into labels by the quantizer. If the universal codebook is very large, the image will use only a subset of the available labels (in fact, if the image has size $X \times Y$ pixels, it can use at most $X \times Y/L$ different code words). A usage map, indicating which code words have been used, can then be associated with the image. Although the usage map could be just a list of the used code

words, more commonly it consists of a bit-array of length K, containing a "1" in the positions corresponding to used code words and a "0" everywhere else. The usage map defines a reduced codebook containing only the code words used. The labels assigned by the quantizer to the image tiles are used to index the reduced codebook rather than the large, universal codebook. Hence, if the image uses only k code words, each compressed tile can be represented by $\log(k)$ bits, instead of the $\log(K)$ bits required to use the entire large codebook. A reduced codebook constructed with the bit-array method is shown in Figure 16.19.

Usage maps reduce the quantizer bit rate. With this method, the compressed representation of an image consists of the labels and of the usage map (since the codebook is universal, it is assumed that both encoder and decoder have a copy of it). This technique requires only K bits to represent the used code words, where again K is the universal codebook size.

The universal quantizer usage map can be used as an index. The key observation is that similar images have in general similar reduced codebooks. The usage map corresponding to an image constitutes a feature vector, which is used as an index to store and retrieve the image. If the usage maps are stored as bit vectors, the dissimilarity between two images can be computed by taking the bitwise exclusive OR of the corresponding usage maps, which produces a bit vector wherein code words that belong to only one of the reduced codebooks are marked with a "1." The number of bits that are "1" in the bit vector is then used as a measure of dissimilarity. Mathematically, the similarity between two images f_m and f_n is measured using the following equation:

$$S(f_m, f_n) = \sum_{i=0}^{N-1} [u(f_m, i) \oplus u(f_n, i)] \tag{16.3}$$

where $\oplus$ is the XOR operation.

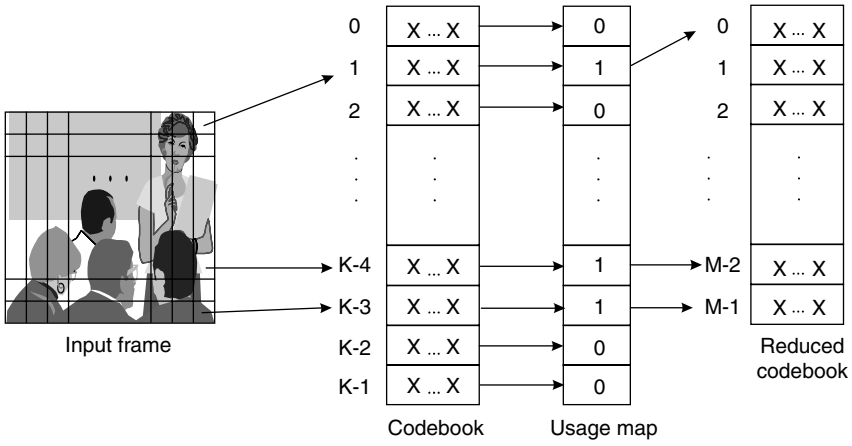


Figure 16.19. Usage map generation.

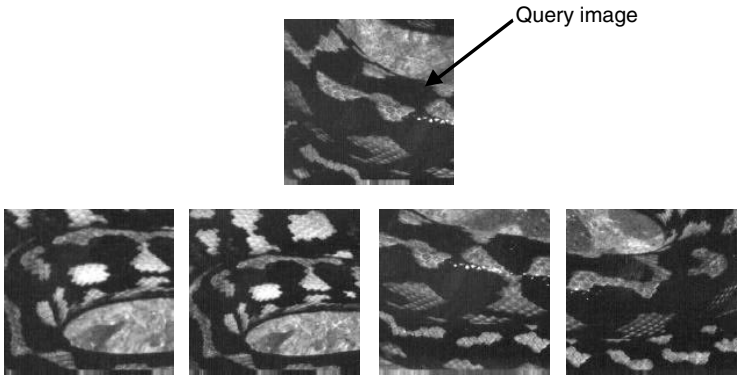


Figure 16.20. Retrieval results based on usage map in VQ compressed domain.

As the usage map is constructed while encoding the image, the described technique does not have a preprocessing cost. The cost of storing the index keys is $O(K)$ bits per image. Searching also requires $O(K)$ operations per target image. A downside of the technique is the coding complexity, which grows with the size of the codebook: for large values of the tile size L, compressing an image of size $X \times Y$ pixels can require $O(XYK)$ operations (which can be reduced to $O(XY \log(K))$ with tree-structured vector quantizers).

The retrieval results corresponding to this technique are shown in Figure 16.20.

16.2.5.3 Content-Based Retrieval of Remotely Sensed Images. A VQ indexing technique for content-based retrieval of remotely sensed images is discussed in Ref. [46]. Assignment of code words for image blocks involves minimization of a weighted error that incorporates a distortion measure. Various distortion measures have been proposed in an effort to enhance the performance of the VQ code words as content descriptors. Two types of query, query-by-class and query-by-value, have been tested with this technique. Query-by-class involves classification of each pixel in an image into predefined classes, whereas query-by-value involves matching on the basis of distribution of intensity values in various bands of a multispectral remotely sensed image. Query-by-value makes optimum use of VQ code words as these code words are directly indicative of the multispectral intensity distributions in the image. Query-by-class, on the other hand, performs some transformations so that VQ code words can be used to answer the queries. As a result of this, it has been found that query-by-value outperforms query-by-class for this VQ indexing technique.

16.2.6 Fractals/Affine Transform

Fractals are irregular patterns made up of parts that are in some way similar to the whole pattern, for example, twigs and tree branches. This property is called self-similarity or self-symmetry. Fractal image compression [11] uses the concepts

of fractal geometry to encode natural images through a small set of parameters. Natural images often display fractal characteristics and portions of images can be well approximated by an appropriate fractal image.

In block-fractal coding, an image is partitioned into a collection of nonoverlapping regions known as *range blocks*. The coding process uses a collection of *domain blocks*, which form a dictionary, and a set of parametric fractal transformations known as *fractal codes*. Each range block is transformed with one of the transformations and the result is compared with all domain blocks. The transformation and mapping steps are repeated for all fractal codes and the best match is found. The range block is associated with the best matching domain block and with the fractal code that produced the match. The block is then encoded by replacing it with the parameters of the best fractal code, and thus the compressed image is a collection of fractal parameters. The size of the compressed image is the product of the number of bits required to describe the fractal parameters of a fractal code times the number of range blocks contained in the image. For a fixed family of fractal codes, larger range blocks yield higher compression ratio, but potentially worse fidelity to the original.

16.2.6.1 Matching Fractal Codes. A texture-based image retrieval technique by which image similarity is detected by matching fractal codes has been presented in Ref. [47]. Here, each database image is decomposed into blocks, each of which is encoded using fractal codes. The collection of the fractal codes of a query image is used as a key and matched with the fractal codes of the images in a database.

16.2.6.2 Indexing Based on Matching Domain Blocks. A comparison of the performance of wavelet and fractals in image retrieval is presented in Ref. [48]. In the wavelet domain, the mean absolute value and variance of different subbands are used as image features. In the fractal domain, fractal coding of the query and target images (denoted by M_1 and M_2 respectively) is performed jointly, with the best matching domain blocks for range blocks of image M_1 searched in both M_1 and M_2. The similarity between M_1 and M_2 is measured as the ratio of the number of matching domain blocks found in M_1 and M_2. Simulation results indicate that wavelets are more effective for images in which texture is predominant, whereas fractals performs relatively well over a variety of images.

16.2.7 Hybrid Schemes

Hybrid schemes are a combination of two or more basic coding techniques [8,49]. Their goal is to exploit the advantages of the associated compression techniques in order to provide a superior coding performance. The indices can therefore be a combination of features derived from the individual coding techniques. This section contains a brief description of several hybrid schemes.

16.2.7.1 Combining Wavelets and VQ. A wavelet-based indexing technique using vector quantization has been described in Ref. [50]. Here, the images are

first decomposed using the wavelet transform, and the transform subbands are then encoded using vector quantization. The collection of codebook labels of each image forms the feature vector used to index the database.

16.2.7.2 Combining Wavelets and Tree-Structured Vector Quantizers. A representation methodology for large databases of pulsed radar returns from naval vessels, which economizes memory and minimizes search time, is presented in Ref. [51]. This technique uses a hierarchical representation by developing clustering schemes based on wavelet representations of these signals. Additionally, a sophisticated version of vector quantization called *tree structured* VQ (TSVQ) is used to further cluster and compress the wavelet representations of the radar signals in a way that permits a hierarchical search across resolutions.

16.2.7.3 Combining Spatial Segmentation, DWT, and VQ. Another technique combining vector quantization, spatial segmentation, and discrete wavelet transform, has been proposed [52] to develop a coding scheme for content-based retrieval. When an image is inserted in the database, it is first decomposed into 8×8 nonoverlapping blocks. Next, a segmentation algorithm is applied to the image to define *image regions* and *objects*. Image regions and objects denote entities of interest in a given image database. Each segmented region in the image is then covered with the smallest possible rectangular collection of the previously defined 8×8 blocks. This collection is called a superblock. The next step consists of applying a wavelet transform to each superblock. The number of levels of decomposition depends on the size of the superblock. The wavelet transform is applied until the lowest frequency subband is of size 8×8. Finally, a coding scheme that incorporates Vector Quantization is applied to map the wavelet transform subbands of the superblocks to finite codebooks.

User-defined queries, in the form of sample images, are provided as input to the wavelet transform. The resulting subbands are mapped to find the appropriate VQ code words for retrieval. The file headers of the image files contain information in terms of a coarse-to-fine discriminant structure. A technique of *progressive refinement retrieval* is adopted to successively reduce the number of files searched at each level of coarseness as more bits are read from the headers.

16.2.7.4 Combining DCT and VQ. A VQ-based face-recognition technique in the DCT domain has been detailed in Ref. [53]. When the database is first created, a set of training images is selected, their block-DCTs are computed, and a k-means quantizer [7] is trained on the vector of DCT coefficients in order to produce a database codebook. When an image is inserted in the database, its block-DCT is quantized using the trained quantizer. Similarity between the query and database images is determined by maximizing a cost function. The retrieval performance has been evaluated based on feature selection, codebook size, and feature dimensionality. For feature selection, transform-based techniques are compared with pixel-intensity techniques, both of which operate on blocks.

It is observed that block-based transform techniques are better in terms of performance efficiency than block-based pixel-intensity techniques. In terms of feature dimension, transform-based techniques show the same performance results as pixel-intensity-based techniques for lower dimensions of feature vectors. Increasing the codebook size also appears to yield better performance.

16.3 CONCLUSION

The success of compressed domain indexing greatly depends on how well the compression or coding technique represents the discriminative features in images. It is generally difficult to compare the performance of the various techniques. KLT, though statistically optimal (if the process is Gaussian), is computationally intensive and requires the basis images to be stored in the database, which yields poor overall compression rates unless the database size is large. Block-DCT in JPEG provides a good coding and indexing performance; nevertheless, the block structure was not originally intended for indexing and can have negative effects. Wavelet-based techniques are promising for indexing applications because of their inherent multiresolution capability, of the simplicity of edge and shape detection, and of the availability of information. However, they have been known to perform well only on images that contain significant textural information. Vector quantization produces the indexing keys while compressing the image but suffers from the additional overhead and approximations involved in codebook generation. Although fractals have a potential to provide good coding performance, they are image-dependent. Finally, the complexity of both VQ and fractal coding is highly asymmetric, namely, coding is computationally intensive, while decoding is fast.

16.4 SUMMARY

In this chapter, we have discussed a variety of compressed-domain indexing approaches and presented examples of techniques derived from each approach. Existing compressed-domain indexing techniques are primarily oriented toward extracting low-level features from the compression parameters. There are extensive similarities between the processes of compression and indexing. For example, the goal of lossy compression is to minimize a cost function that combines distortion, bit rate, and complexity, whereas the purpose of indexing is to compactly represent the visual content in an image or video sequence to facilitate fast retrieval. As both compression and indexing rely on efficient extraction of visual content, they can be addressed jointly. Future research will not only involve investigations of how to use existing compression parameters for indexing, but also explore the possibility of employing the spatio-temporal indexing parameters for compression. Furthermore, novel joint compression indexing schemes will need to be devised that yield very low data rate with high retrieval performance.

ACKNOWLEDGMENTS

The author would like to acknowledge Dr. Mrinal Mandal, Dr. Fayez Idris, Mr. Mohammed Zubair V, Mr. Jayank Bhalod, and Mr. Madhusudhana Gargesha for their valuable help in writing this chapter.

REFERENCES

1. M. O'Docherty and C. Daskalakis, Multimedia information systems: the management and semantic retrieval of all electronic datatypes, *Comput. J.* **34**(3), 225–238 (1991).
2. V.N. Gudivada and V.V. Raghavan, Content based image retrieval systems, *IEEE Comput.* **28**(9), 18–22 (1995).
3. J.D. Villasenor, Full-frame compression of tomographic images using the discrete Fourier transform, *Data Compression Conference*, 195–203 (1993).
4. G.M. Binge and E. Micheli-Tzanakou, Fourier compression-reconstruction technique (MRI application), *Proc. Annu. Int. Conf. IEEE Eng.* **2**, 524–525 (1989).
5. N. Ahmed, T. Natarajan, and K.R. Rao, Discrete Cosine Transform, *IEEE Trans. Comput.* **23**, 90–93 (1974).
6. K.R. Rao and P. Yip, *Discrete Cosine Transforms—Algorithms, Advantages, Applications*, Academic Press, New York, 1990.
7. A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.
8. M. Antonini et al., Image coding using wavelet transform, *IEEE Trans. Image Process.* **1**(2), 205–220 (1992).
9. J. Froment and S. Mallat, in Second generation compact image coding with wavelets, C.K. Chui, ed., *Wavelets: A Tutorial in Theory and Applications*, Academic Press, New York, 1992.
10. F. Idris and S. Panchanathan, Image indexing using vector quantization, *Proc. SPIE: Storage and Retrieval for Image and Video Databases III* **2420**, 373–380 (1995).
11. M.F. Barnsley and L.P. Hurd, *Fractal Image Compression*, AK Peters Ltd., Wellesley, Mass., 1993.
12. H.S. Stone, C.S. Li, Image matching by means of intensity and texture matching in the fourier domain, *Proc. SPIE* **2670**, 337–349 (1996).
13. M. Augusteijn, L.E. Clemens, and K.A. Shaw, Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier, *IEEE Trans. Geosci. Remote Sensing* **33**(3), 616–626 (1995).
14. A. Celentano and V.D. Lecce, A FFT based technique for image signature generation, *Proc. SPIE: Storage and Retrieval for Image and Video Databases V* **3022**, 457–466 (1997).
15. A. Pentland, R.W. Picard, and S. Sclaroff, Photobook: tools for content-based manipulation of image databases, *Proc. SPIE: Storage and Retrieval for Image and Video Databases II* **2185**, 34–47 (1994).
16. D.L. Swets and J. Weng, Using discriminant eigenfeatures for image retrieval, *IEEE Trans. Pattern Anal. Machine Intell.* **18**(8), 831–836 (1996).

17. X. Tang and W.K. Stewart, Texture classification using principal-component analysis techniques, *Proc. SPIE* **2315**, 22–35 (1994).
18. J.A. Saghri, A.G. Tescher, and J.T. Reagan, Practical transform coding of multispectral imagery, *IEEE Signal Process. Mag.* **12**(1), 33–43 (1995).
19. J. Hafner et al., Efficient color histogram indexing for quadratic form distance function, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(7), 729–736 (1995).
20. J.R. Smith and S.-F. Chang, Transform features for texture classification and discrimination in large image databases, *Proc. IEEE Int. Conf. Image Process.* **3**, 407–411 (1994).
21. M. Shneier and M.A. Mottaleb, Exploiting the JPEG Compression Scheme for Image Retrieval, *IEEE Trans. Pattern Anal. Machine. Intell.* **18**(8), 849–853 (1996).
22. A.A. Abdel-Malek and J.E. Hershey, Feature cueing in the discrete cosine domain, *J. Electron. Imaging* **3**, 71–80 (1994).
23. B. Shen and I.K. Sethi, Direct feature extraction from compressed images, *Proc. SPIE* **2670**, 404–414 (1996).
24. S.G. Mallat, A Theory for multiresolution signal representation: the wavelet decomposition, *IEEE Trans. Pattern Anal. Machine. Intell.* **11**(7), 674–693 (1989).
25. C.E. Jacobs, A. Finkelstein, and D.H. Salesin, Fast multiresolution image querying, *Proc. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, Los Angeles, Calif. August, 1995.
26. J.Z. Wang et al., Wavelet-based image indexing techniques with partial sketch retrieval capability, *Proceedings of the Forum on Research and Technology Advances in Digital Libraries*, Washington, DC, 13–24 (1997).
27. M.K. Mandal, S. Panchanathan, and T. Aboulnasr, Image indexing using translation and scale-invariant moments and wavelets, *Proc. SPIE: Storage and Retrieval for Image and Video Databases V* **3022**, 380–389 (1997).
28. M.K. Mandal, T. Aboulnasr and S. Panchanathan, Image indexing using moments and wavelets, *IEEE Trans. Consumer Electron.* **42**(3), 557–565 (1996).
29. K.A. Birney and T.R. Fischer, On the modeling of DCT and subband image data for compression, *IEEE Trans. Image Process.* **4**(2), 186–193 (1995).
30. F. Qi, D. Shen and L. Quan, Wavelet transform based rotation invariant feature extraction in object recognition, *Proc. Int. Symp. Inform. Theory Appl.* 221–224, (1994).
31. R.J. Prokop and A.P. Reeves, A survey of moment-based techniques for unoccluded object representation and recognition, *CVGIP: Graphical Models and Image Process.* **54**, 438–460 (1992).
32. M.R. Teague, Image Analysis via the General Theory of Moments, *J. Opt. Soc. Am.* **70**, 920–930 (1990).
33. P. Rashkovskiy and L. Sadovnik, Scale, rotation and shift invariant wavelet transform, *Proc. SPIE: Optical Pattern Recognition V* **2237**, 390–401 (1994).
34. T. Chang and C.C.J. Kuo, Texture analysis and classification with tree-structured wavelet transform, *IEEE Trans. Image Process.* **2**(4), 429–441 (1993).
35. J.S. Lee, R.C. Kim, and S.U. Lee, Transformed entropy-constrained vector quantizers employing mandala block for image coding, *Signal-Process. Image-Commun. J.* **7**, 75–92 (1995).

36. J.S. Lee, R.C. Kim, and S.U. Lee, Entropy-constrained vector quantization of images in the transform domain, *Proc. SPIE* **2308**, 434–445 (1994).
37. C.S. Li et al., Comparing texture feature sets for retrieving core images in petroleum applications, *Proc. SPIE: Storage and Retrieval for Image and Video Databases* **3665**, 2–11 (1999).
38. C.S. Li and V. Castelli, Deriving texture feature sets for content-based retrieval of satellite image databases, Calif. Santa Barbara, October 26–29, 1997.
39. J.L. Chen and A. Kundu, Rotation and gray scale invariant texture identification using wavelet decomposition and hidden markov model, *IEEE Trans. Pattern Anal. Machine Intell.* **16**(2), 208–214 (1994).
40. M.K. Mandal, S. Panchanathan, and T. Aboulnasr, Image indexing using translation and scale-invariant moments and wavelets, *Proc. SPIE: Storage and Retrieval for Image and Video Databases V* **3022**, 380–389 (1997).
41. B.S. Manjunath and W.Y. Ma, Texture features for browsing and retrieval of image data, *IEEE Trans. Pattern Anal. Machine Intell.* **18**(8), 837–841 (1996).
42. H. Wang and S.-F. Chang, Adaptive image matching in the subband domain, *Proc. of SPIE: VCIP* **2727**, 885–896 (1996).
43. K.-C. Liang and C.-C. Jay Kuo, WaveGuide: A joint wavelet-based image representation and description system, *IEEE Trans. Image Process.* **8**(11), 1619–1629 (1999).
44. F. Idris and S. Panchanathan, Video compression using frame adaptive vector quantization, *J. Vis. Commun. Image Represent.* **9**(2), 107–118 (1998).
45. F. Idris and S. Panchanathan, Storage and retrieval of compressed images, *IEEE Trans. Consumer Electron.* **41**, 937–941 (1995).
46. A. Vellaikal, C.C.J. Kuo, and S. Dao, Content-based retrieval of remote-sensed images using vector quantization, *Proc. SPIE* **2488**, 178–189 (1995).
47. A. Zhang, B. Cheng, and R.S. Acharya, Approach to query-by-texture in image database systems, *Proc. SPIE: Digital Image Storage and Archiving Systems* **2606**, 338–349 (1995).
48. A. Zhang et al., Comparison of wavelet transforms and fractal coding in texture-based image retrieval, *Proc. SPIE: Visual Data Exploration and Anal. III* **2656**, 116–125 (1996).
49. J. Li, Hybrid wavelet-fractal image compression based on a rate-distortion criterion, *Proc. SPIE: Visual Commun. Image Process.* **3024**, 1014–1025 (1997).
50. F. Idris and S. Panchanathan, Image indexing using wavelet vector quantization, *Proc. SPIE: Digital Image Storage Archiving Systems* **2606**, 269–275 (1995).
51. J.S. Barbas and S.I. Wolk, Efficient organization of large ship radar databases using wavelets and structured vector quantization, *Proc. Asilomar Conf. Signals, Syst. Comput.* **1**, 491–498 (1993).
52. M.D. Swanson, S. Hosur, and A.H. Tewfik, Image coding for content-based retrieval, *Proc. SPIE: VCIP* **2727**, 4–15 (1996).
53. C. Podilchuk and X. Zhang, Face recognition using DCT-based feature vectors, *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.* **4**, 2144–2147 (1996).