WaveGuide: A Joint Wavelet-Based Image Representation and Description System

Kai-Chieh Liang, *Member, IEEE,* and C.-C. Jay Kuo, *Fellow, IEEE*

***Abstract*—Data representation and content description are two basic components required by the management of any image database. A wavelet-based system, called the WaveGuide, which integrates these two components in a unified framework, is proposed in this work. In the WaveGuide system, images are compressed with the state-of-the-art wavelet coding technique and indexed with color, texture, and object shape descriptors generated in the wavelet domain during the encoding process. All the content descriptors are extracted by machines automatically with a low computational complexity and stored with a low memory space. Extensive experiments are performed to demonstrate the performance of the new approach.**

***Index Terms*—Data compression, image description, image indexing, image representation, image retrieval, wavelet coding.**

I. Introduction

DUE TO THE tremendous growth of multimedia information, effective management of multimedia archiving and storage systems becomes more important and challenging. For example, a remote sensing satellite, which generates seven band images including three visible and four infrared spectrum regions, produces around 5000 images per week. Each single spectral image, which corresponds to a 170 km $\times$ 185 km of the earth region, requires 200 Mega bytes of storage. It is estimated that the amount of data originated from satellite systems will reach a terabyte per day [4]. To store, index, and retrieve such a huge amount of data is a very challenging task. Generally speaking, data representation and content description are two basic components required by the management of any multimedia database. As far as the image database is concerned, the former is concerned with image storage while the latter is related to image indexing and retrieval.

Current commercial systems support image indexing based on the use of keywords or text phrases associated with images. The keyword is a high-level tool of content description, and has been successfully applied to textual databases. However, there are limits [1], [13], [14], [30] in applying the same technique to image indexing. First, it is often difficult to describe the content of an image such as complicated texture patterns with human languages. Second, manual annotation of text phrases for a large database takes a lot of time and effort. Third, since users may have different interests in the same data, it is difficult to describe an image with a complete set of key words. Finally, even if all relevant image characteristics are annotated, difficulty may still arise due to the use of different indexing languages or vocabularies by different users. Another approach to image indexing and retrieval is to exploit low-level image description tools such as color, shape, and texture features that can be automatically extracted by machines [1], [3], [13], [14], [30], [33], [34], [42]. It requires much less effort in comparison with manual annotation. It is worthwhile to emphasize that machine-extractable features are not meant to replace keywords. High- and low-level description tools are compatible in the sense that they provide content information at different abstract levels, i.e., semantic and feature spaces, respectively. As stated in the document of the MPEG7 standard [44], a motion picture can be described by keywords such as its title, director, date of release, production company, etc., as well as machine-extractable features including color or texture components of dominant frames, motion information, and audio characteristics. It is desirable to integrate features in different aspects and at different abstract levels to make the description complete.

A joint set of image descriptors including texture, color, and shape features extracted from the wavelet domain is explored for image indexing in this work. We allow an adjustable weighting in combining these features together according to distinctive characteristics of the query image. For example, if the query image is special in its color attribute, the color can be used as the main attribute while texture and shape features as auxiliary attributes for similar image search in the database. A closely related issue is that images are stored in a certain compression format in the database. Compressed images are composed by decorrelated bit streams of random zero's and one's, from which image features are difficult to extract. A straightforward approach to indexing a compressed image is to decode the bit stream for the original image, and then extract content descriptors accordingly. This indexing procedure is, however, inefficient, since it takes extra time and computational complexity. Thus, besides bit rates, distortion and complexity, a fourth criterion of evaluating an image coding scheme was proposed in [32] based on its content accessibility. With the new criterion, a good coding technique should provide content access without fully decoding the bit

Manuscript received June 8, 1998; revised March 17, 1999. This work was supported by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsuhan Chen.

K.-C. Liang is with Sharp Microelectronics of the Americas, Huntington Beach, CA 92647-2053 USA (e-mail: liangv@sharpsec.com).

C.-C. J. Kuo is with the Integrated Media Systems Center and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: cckuo@sipi.usc.edu).

Publisher Item Identifier S 1057-7149(99)08751-5.

stream (or with the least amount of decoding) for the indexing purpose. Furthermore, if the size of content descriptors is not well controlled, the complexity of feature matching can grow and slow down the retrieval process. The amount of indexing information required may take up the memory space and can deteriorate storage efficiency of a database. Since image content description and compression are closely related in terms of content accessibility and indexing overhead, they should be examined together in the design of an image database.

In this research, we aim at an integrated system for both image content description and compression, and adopt the state-of-the-art wavelet-based image coding technique [23], [37], [38], [45]. The wavelet-based coding scheme provides superior coding efficiency and new functionalities such as resolution and quality scalability. Since a wavelet coded bit stream can be progressively decoded, the image content of a certain resolution can be conveniently accessed at a certain stage of decompression. Thus, the wavelet-based image representation provides a good joint framework for coding and content description. Furthermore, we carefully control the size of content descriptors both in terms of the number of indexing elements and the number of bytes to reduce the matching complexity as well as the memory space. In other words, the amount of "bits about bits" (content descriptors) [44] is exceedingly smaller than that of "data bits" [44] in our designed system. There has been growing interest about content-based image retrieval for last several years. Examples include the IBM QBIC system [1], [13], [14], [30], MIT Photobook system [33], [34], the Columbia VisualSEEk system [42], and the Virage system [3], among many others. These systems provide a set of features for content description and a user-machine interface for image search and browsing. However, they do not address the issue of indexing a compressed image database. As to previous work on wavelet domain features, most research focused on the use of wavelet coefficients to classify and differentiate textures [6], [25], [27], [39], while some considered the use of color histograms of coefficients [28], [29] and locations of significant coefficients to facilitate image retrieval [19]. Excellent performances have been reported in these papers. However, an integrated coding and content description system is seldom examined. A wavelet-based prototype system, called the WaveGuide, for image indexing, search, browsing, and compression is presented in this work.

This paper is organized as follows. An effective image representation scheme by using the successive approximation quantization (SAQ) and the bit plane structure of wavelet coefficients is reviewed in Section II. Image content descriptors based on quantized wavelet coefficients for texture, color, and shape feature extraction are examined, respectively, in Sections III–V. The WaveGuide prototype system and experimental results are provided in Section VI. Finally, concluding remarks are given in Section VII.

II. Wavelet-Based Image Representation

Several wavelet-based coding methods have been proposed recently such as the embedded zerotree wavelet (EZW) [38], the layer zero coding (LZC) [45], the set partitioning in hierarchical trees (SPIHT) [37], the rate-distortion optimized wavelet packet (WP) [23], the multithreshold wavelet coding (MTWC) [48], etc. All the above methods consist of these three stages:

1) application of the wavelet transform to a given image;
2) SAQ of wavelet coefficients to obtain a bit plane representation of the wavelet coefficients;
3) effective entropy coding of the resulting bit planes.

These methods are similar in the first two stages, but different in the last stage. In the entropy coding stage, one can classify the bit stream into two parts, i.e., the structured and the unstructured zero-one patterns, and encode them differently. As far as the content description is concerned, it is difficult to use the output bit stream from the third stage since it is highly dependent on the algorithm for zero grouping [37], [38] and the entropy coder [5], [49]. In comparison, the content representation of the output from the 2nd stage is very robust. Quantized wavelet coefficients provide a very good spatial-frequency representation of the original image, while the bit plane structure allows a fast computation of the histogram of wavelet coefficients. Thus, quantized wavelet coefficients are chosen to be the image representation method.

The SAQ of wavelet coefficients and its corresponding bit plane structure is briefly reviewed below. To illustrate the SAQ procedure, let us consider a set of N wavelet coefficients with magnitudes W_0, $W_1, \cdots$, and W_{N-1}. Note that since signs of wavelet coefficients are usually coded separately, one can focus on the magnitude quantization only. In SAQ, a sequence of thresholds T_0, $T_1, \cdots$, and T_L are adopted for quantization, and they are related via

$$T_l = T_{l-1}/2, \qquad l = 1, 2, \cdots, L.$$

With the initial threshold to be one half of the maximum magnitude, i.e.,

$$T_0 = \tfrac{1}{2} \max_i |W_i|.$$

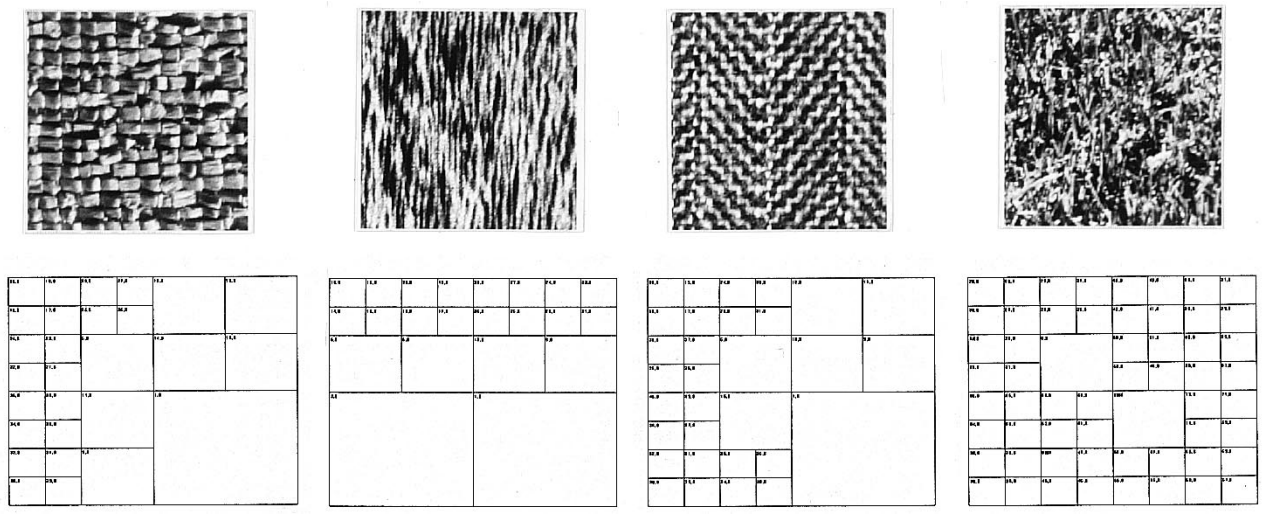
For a given threshold value T_l, we scan all wavelet coefficients with two passes, i.e., the dominant pass and the subordinate pass. In the dominant pass, we identify significant coefficients depending on whether they are larger or smaller than the current threshold. In the subordinate pass, we perform the magnitude refinement of all coefficients that are identified as significant earlier. During the coding process, a binary map B called the significance map [38] is maintained to store the coordinates of significant coefficients so that the coder knows the locations of significant as well as insignificant coefficients.

If coefficient W_n is identified as significant at quantization level l, the quantized magnitude of W_n can be written as a binary representation of the following form:

$$Q(W_n) = (1, B_{n,l+1}, B_{n,l+2}, \cdots)$$

where $B_{n,l+1}$, $B_{n,l+2}, \cdots$ are refinement bits at quantization level $l+1$, $l+2$, and so on. The reconstructed (or dequantized) value of W_n is equal to

$$\hat{W}_n = \frac{3}{2}T_l + f(B_{n,l+1})\frac{T_l}{2} + f(B_{n,l+2})\frac{T_l}{4} + \cdots$$



(a) (b) (c) (d)

Fig. 1. Textured images and their corresponding wavelet packet decompositions: (a) raffia, (b) water, (c) weave, and (d) grass.

where $f(B)$ takes values of $+1$ and -1 when B is equal to one and zero, respectively. For more details, we refer to [38].

To summarize, with SAQ and the bit plane representation, coefficients with a larger magnitudes are identified as significant at an earlier quantization step with more refinement bits following. This approach allows a progressive representation of a given image.

III. Texture Descriptor

Textures provide important surface characteristics of image objects and are widely chosen as features for image classification and retrieval. Early work considered the statistics (e.g., second-order statistics) or the distribution model (e.g., the Gibbs distribution model) of textures [7], [11], [12], [16]. The difficulty with traditional methods lies in the lack of an adequate multiresolution tool. Laws [24] used a local linear transformation and the energy computation to extract texture features and obtained very good results. Laws' method can be interpreted as a multichannel (or subband) decomposition approach. Furthermore, it is known that many natural textures can be modeled as quasiperiodic patterns. Research of human vision system (HVS) shows that the time-frequency representation [8], [36] can preserve both global and local information well and is suitable for the modeling of quasiperiodic signals. The wavelet-based approach [6], [25] integrates the multiresolution and the space-frequency properties naturally, and has demonstrated a remarkable performance for texture classification and analysis.

Chang and Kuo [6] used the wavelet packet transform to decompose significant subbands adaptively for texture description. As shown in Fig. 1, each textured pattern has its own decomposition structure, and significant subbands are decomposed into finer subbands successively. Here, we do not attempt to use the complicated decomposition structure as the feature directly. Instead, we use the number of significant coefficients in a subband as feature, which serves as a rough indicator of the significance of a particular subband. Generally speaking, the larger the number is, the higher energy the subband possesses. This number also correlates well with the coding bit rates, since it takes approximately 1 bit to refine a significant coefficient for an additional 1 bit precision in SAQ.

For a given texture and a predetermined threshold value T_l, we propose to measure the importance of a subband by counting the number of significant coefficients, i.e.,

$$N_i(T_l) = |\{W(j,k) \in S_i, |W(j,k)| \geq T_l\}| \tag{1}$$

where S_i denotes the ith subband and $W(j,k)$ is the wavelet coefficient at coordinate (j,k). Furthermore, we can measure the relative importance of a subband by considering the normalized value of (1), i.e.,

$$b_i = \frac{N_i}{\sum_i N_i} \tag{2}$$

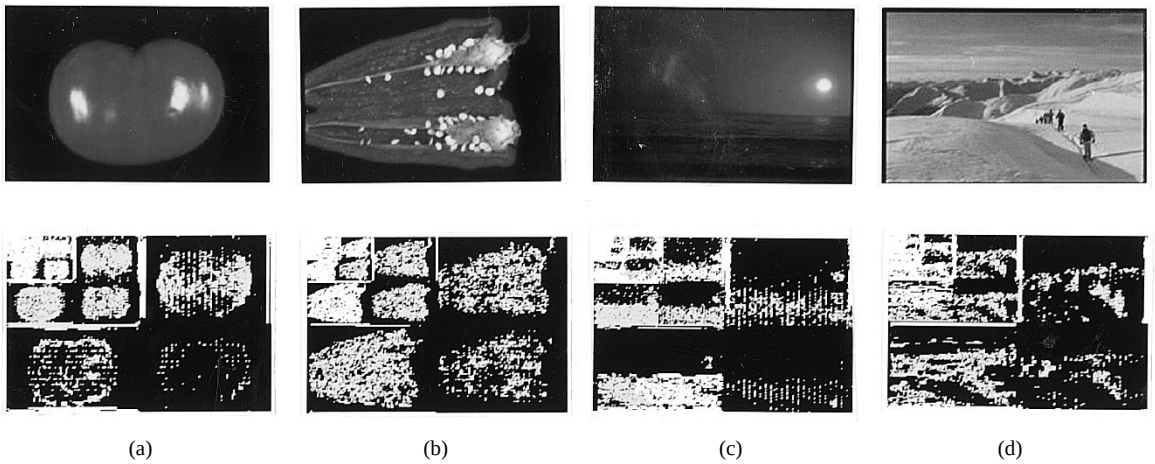
where N_i is the number of significant coefficients in the ith subband.

The texture feature as given in (1) and (2) can be viewed as a simplified histogram of wavelet coefficients in each subband with two quantization bins (i.e., significant and insignificant level). One important issue is the appropriate choice of the parameter T_l in (1) so that the image would have neither too many nor too few significant coefficients. Let us consider an extreme example. That is, (1) is computed with a very small T_l so that all coefficients become significant. For such a case, the discriminating power of significant coefficient distribution among subbands is lost. In this work, we compute (1) with respect to the threshold T_{10} of the tenth layer where about 50% of coefficients in images are significant.

We define the texture similarity as the L_p-distance between feature vectors

$$d_{L_p}(a,b) = \left(\sum_i w_i \cdot |a_i - b_i|^p \right)^{1/p} \tag{3}$$

where a and b are the feature vectors of two images and w_i is the weighting factor for the ith subband. The choice $p = 1$



(a) (b) (c) (d)

Fig. 2. Several images and their corresponding significant maps: (a) tomato, (b) pepper, (c) sunset, and (d) ski.

is used in our implementation. One reasonable choice of w_i is the inverse of the total number of wavelet coefficients in subband i. Equation (3) is actually applicable regardless of the decomposition structure. For the pyramid transform, where only the coarsest subbands are recursively decomposed, we can measure the similarity directly between images. For the adaptive WP transform, the distance measure can be more complicated, since two images may not have the same WP decomposition structure. Here, we adopt a simple rule to handle this situation. That is, when a further decomposed subband is matched to an undecomposed subband, we simply assume that the undecomposed one has four pseudo-child subbands and each child node has one quarter of its parent's significance, as illustrated in Fig. 3.

The numbers of significant coefficients in a four-scale pyramid transform of four test textures are listed in Table I, where the last digit in each subband denotes the level of decomposition.[1] Letters H and L denote the high and low frequency subbands and subscripts x and y denote the x- and y-directions, respectively. The four textures are shown in Fig. 1 for visual comparison while the corresponding wavelet packet decomposition for each texture is also provided. We can see from Table I and Fig. 1 that when a subband in the pyramid structure is significant, i.e., having certain amount of significant coefficients, it is refinely decomposed in the wavelet packet structure. In other words, our feature can effectively represent the importance of subbands as the structures of wavelet packet decomposition. In this work, we use the pyramid transform for our experiments.

It is worthwhile to point out that the texture feature discussed above is sensitive to image orientation. For example, by rotating an image which has a large number of significant coefficients in the H_xL_y subbands with 90 degrees, most significant coefficients are now located in the L_xH_y subbands with the same content. Thus, this feature can only be used to retrieve similar images with the same orientation. Usually, there are several textures in a natural image with or without a dominant texture component, which complicates its texture description. One may use an image segmentation method to partition the image into several regions and then obtain texture feature for each region. For the image indexing and retrieval application, a very accurate segmentation result may not be essential. What is needed is to detect prominent regions with distinctive features [10], [26]. To serve this objective, the map of significant wavelet coefficients provides an efficient tool for image segmentation. Four images and their significant coefficient maps are given in Fig. 2. Two of them are object images (tomato and pepper) while the other two are scene images (sunset and ski). Objects in the tomato and pepper image are clearly shown, and different textured regions in the sunset (the sky and the ocean) and ski images (the sky

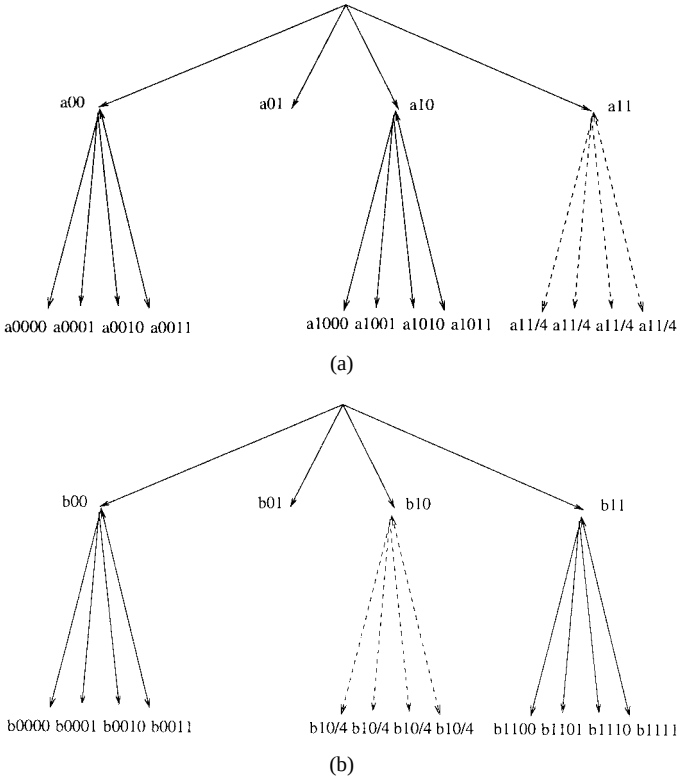


Fig. 3. Similarity measure of the subband significance feature between two wavelet packet structures.

[1] Zero denotes the highest frequency subbands and 3 the lowest frequency subbands.

TABLE I
NUMBERS OF SIGNIFICANT WAVELET COEFFICIENTS IN A FOUR-SCALE PYRAMID WAVELET DECOMPOSITION FOR FOUR TEXTURES

Subband	Raffia	Water	Weave	Grass
L_xL_y3	0.010	0.011	0.007	0.008
H_xL_y3	0.010	0.010	0.007	0.007
L_xH_y3	0.010	0.009	0.007	0.008
H_xH_y3	0.011	0.009	0.007	0.007
H_xL_y2	0.040	0.041	0.028	0.028
L_xH_y2	0.041	0.029	0.028	0.028
H_xH_y2	0.036	0.029	0.029	0.027
H_xL_y1	0.121	0.159	0.106	0.107
L_xH_y1	0.146	0.053	0.106	0.095
H_xH_y1	0.077	0.045	0.080	0.087
H_xL_y0	0.171	0.449	0.279	0.296
L_xH_y0	0.296	0.086	0.221	0.190
H_xH_y0	0.030	0.072	0.097	0.108

and the snow) are detected, respectively, in the corresponding significant coefficient maps. The proposed texture extraction method can be performed in each textured region.

IV. COLOR DESCRIPTOR

Color has long been recognized as a useful feature for content-based retrieval [1], [3], [13], [14], [30], [33], [34], [40]–[42], [50]. One distinctive property of the color feature is its invariance under translation and rotation about the viewing axis and its slow variation under the change of the viewing angle, scale, and occlusion [41]. The color of an image is usually represented by the statistics (histogram) of the tristimulus values of pixels, such as (R,G,B) or (Y,U,V), based on the entire image or selected areas in the space domain. In this work, we investigate the wavelet color content description in the YUV coordinate system.

The distribution of wavelet coefficients of an image is first studied for an appropriate choice of the color quantization scheme. Let us plot the magnitude distribution of pixels in the space domain and that of coefficients in the wavelet domain in Fig. 4(a) and (b), respectively, for the Lena image. Generally speaking, the magnitude distribution in the space domain can be quite different depending on the image characteristics, while the magnitude distribution of wavelet coefficients is close to the Laplacian function, which suggests a nonuniform quantization scheme by allocating more bins in the higher probability density area to achieve a better approximation of the distribution. The exponentially decaying shape of the wavelet coefficient distribution has been effectively exploited by modern wavelet coders [23], [37], [38], [45] via SAQ and the bit plane coding. Based on SAQ, it is straightforward to obtain the desired nonuniform histogram as shown in Fig. 5:

$$P(l)=\begin{cases} P_R\{T_l \le |W(j,k)| < 2T_l\}, & \text{if } 0 \le l \le L \\ P_R\{|W(j,k)| < T_L\}, & \text{if } l = L+1 \end{cases}$$

where T_l denotes the quantization threshold for layer $0 \le l \le L$. In other words, $P(l)$ corresponds to the probability of coefficients that are just identified significant when the quantization threshold is set to T_l. In our implementation, three 12-bin histograms are computed, respectively, from the luminance (Y) and the chrominance $(U$ and $V)$ components of a colorful image with respect to thresholds $T_0,\cdots,T_{11}$.

One popular metric employed to compare the distance of two color histograms is the L_p-norm as used in [1], [13], [14], [30], and [41]. There are other quantitative ways to characterize the shape of a histogram such as the mean, standard deviation, entropy, energy, etc. [31]. Here, we adopt the mean, the variance, and the skewness of a histogram as similarity metrics. For the Y component, we have

$$\mu_Y = \sum_{l=0}^{L+1} P_Y(l)\cdot c_Y(l),$$

$$\sigma_Y = \left(\sum_{l=0}^{L+1} P_Y(l)\cdot (c_Y(l)-\mu_Y)^2\right)^{1/2},$$

$$s_Y = \left(\sum_{l=0}^{L+1} P_Y(l)\cdot (c_Y(l)-\mu_Y)^3\right)^{1/3}$$

where μ,σ, and s denote the mean, variance, and skewness measures, respectively, P_Y the probability value, and $c_Y(l)$ the centroid for bin l. The centroid is often set to the center of a bin except for the last bin, which is set to zero, because insignificant coefficients are not coded and treated as zeros. Similar expressions can be written for the U and V components.

There are two reasons to justify the use of color moments as the similarity metric. First, according to the moment representation theorem, the infinite set of moments uniquely determine a probability distribution, and vice versa [20]. Since higher order moments decay faster, we can reduce the size of feature vectors as well as the complexity of feature matching by using the first three color moments. In the current context, instead of storing 36 probability values (for three 12-element histograms), we only have to store nine moments (i.e., three moments for each histogram) for the color description. Second, it was observed in [40] that color moments are more robust than the L_1-distance measure of the histogram difference. As shown in Fig. 6, three histograms are ordered in such a way that neighboring bins corresponding to similar colors. Histograms in (a) and (b) are more similar perceptually. However, with the L_1-distance computation, there is no match between (a) and (b) while there is one match between (a) and (c). In this example, the L_1-distance measure contradicts the human visual system (HVS) in the perception of color similarity. If the moments are used to model these histograms, (a) will be more similar to (b) than (c), since (a) and (b) are only slightly shifted from each other.

We defined color similarity as the L_1-distance between color moments

$$\begin{aligned} d_{L_1}(f,g) = {} & w_{\mu,Y}\cdot|\mu_{Y,f}-\mu_{Y,g}| + w_{\mu,U}\cdot|\mu_{U,f}-\mu_{U,g}| \\ & + w_{\mu,V}\cdot|\mu_{V,f}-\mu_{V,g}| + w_{\sigma,Y}\cdot|\sigma_{Y,f}-\sigma_{Y,g}| \\ & + w_{\sigma,U}\cdot|\sigma_{U,f}-\sigma_{U,g}| + w_{\sigma,V}\cdot|\sigma_{V,f}-\sigma_{V,g}| \\ & + w_{s,Y}\cdot|s_{Y,f}-s_{Y,g}| + w_{s,U}\cdot|s_{U,f}-s_{U,g}| \\ & + w_{s,V}\cdot|s_{V,f}-s_{V,g}| \end{aligned} \quad (4)$$

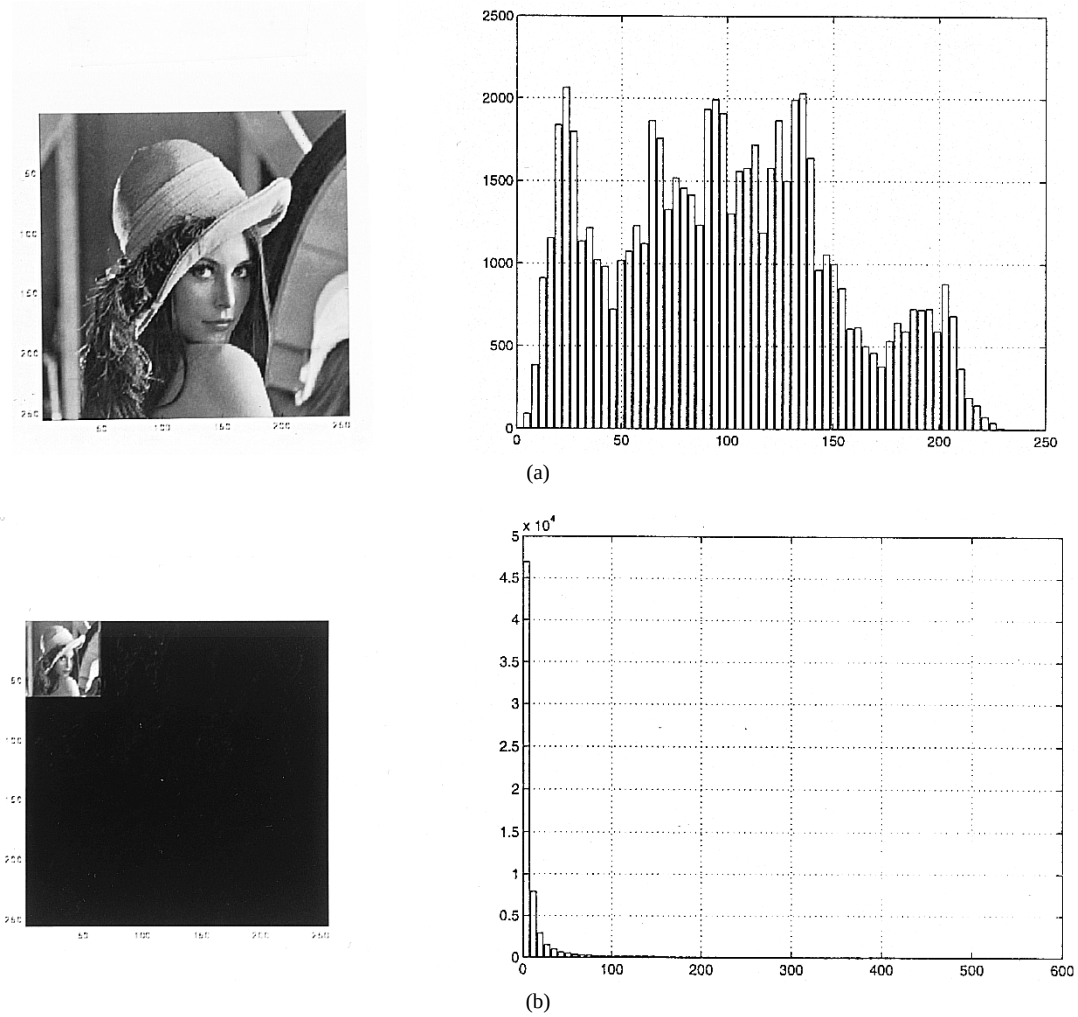


Fig. 4. Lena image and its magnitude distribution in the (a) space and (b) wavelet domains.

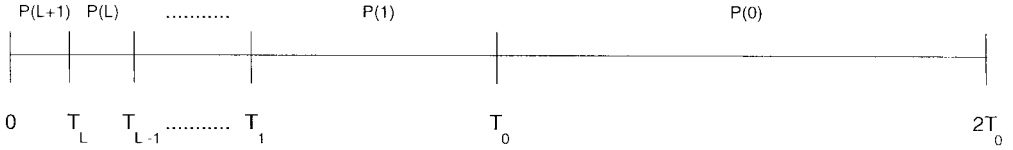


Fig. 5. Color histogram calculation based on a nonuniform quantization scheme.

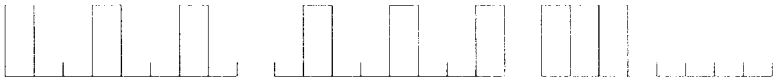


Fig. 6. Example illustrating the inefficiency of the L_1-distance of color histograms. [40].

where w's are the weighting parameters adjustable by users. In our implementation, we use the following empirical values:

$$\begin{bmatrix} w_{\mu,Y} & w_{\sigma,Y} & w_{s,Y} \\ w_{\mu,U} & w_{\sigma,U} & w_{s,U} \\ w_{\mu,V} & w_{\sigma,V} & w_{s,V} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The weights associated with the Y-component are decreased, since μ, σ, and s of U and V are smaller in comparison with those of Y.

V. Shape Descriptor

Previous work on spatial features focused on the shape of an image object. There is, however, no mathematically rigorous definition of shape similarity that accounts for semantic recognition or perceptual judgment of human beings [43]. Computational shape methods [43] include the moment-based matching algorithm, the parametric curve distance measure, the turning angle matching method, etc. Moment-based algorithms treat an image as a 2-D probability function and applies the moment theory to this function [17], [46]. The parametric

curve distance approach represents the object boundaries by a set of spline functions and measure the distances between boundary curves. In the turning angle method [2], the turning angle along the perimeter of an object is recorded by a function θ. Then, elements of θ are matched to those of other sets for similarity comparison. Pairing of elements could be multiple-to-one or one-to-one, but it must proceed monotonically through both sets. Basically, the above methods are performed in the space domain and does not fit our wavelet-domain framework. Besides, a binary shape image has to be generated from the image first by an automatic or a manual algorithm before any shape analysis.

A wavelet-based approach is adopted for shape analysis here. Again, we start with the significant map of SAQ, which is automatically generated during the coding process, as a binary description of an object, and compute the spatial moments accordingly [17], [31]. The significant map, as shown in Fig. 2, is a binary record of the address of significant coefficients. It contains spatial information of an image object in various scales (or resolutions) and frequency channels, therefore serving as an adequate tool for multiresolution shape analysis. Let us take the tomato image in Fig. 2(a) as an example, edges along the X-, Y-, and XY-directions are extracted in the H_xL_y, L_xH_y, and H_xH_y subbands, respectively, with different scales. Note that since topological attributes, such as the object area or the perimeter, are not necessarily connected in the map, it is difficult to apply the turning angle or the parametric curve distance methods to these data. In comparison, the moment-based approach is still applicable. The spatial moment of order (p,q) can be defined as follows [31]:

$$m_{pq} = \frac{1}{K^p J^q} \sum_k \sum_j (x_k - \overline{x})^p (y_j - \overline{y})^q B(j,k) \qquad (5)$$

where $B(j,k)$ is the binary value of the significant map (one for significant and zero for insignificant), J and K are the number of rows and columns of the corresponding subband,

$$x_k = k - \tfrac{1}{2}, \quad y_j = J + \tfrac{1}{2} - j$$

and $\overline{x}$ and $\overline{y}$ are mean values of x_k and y_j. Since central moments are computed with respect to centroids $\overline{x}$ and $\overline{y}$ in each direction, they are invariant under the translation of the object. We can further normalize the moments by the total object area to make it invariant to the change of scale via

$$\eta_{pq} = \frac{m_{pq}}{m_{00}^{\alpha}}, \quad \text{where} \quad \alpha = \frac{p+q}{2} + 1. \qquad (6)$$

As a result, η_{pq} is invariant both to translation and scale change. In our implementation, the spatial distribution in a subband is described by nine elements: two means ($\overline{x_k}$ and $\overline{y_j}$), three variances (η_{20}, η_{11}, and η_{02}), and four skewnesses (η_{30}, η_{21}, η_{12}, and η_{03}). In this work, we compute the moments of H_xL_y, L_xH_y, and H_xH_y subbands in the first three scales only. Thus, the shape feature vector consists of 81 float numbers. Note that it is not necessary to compute the moments in all the scales, because the resolution of shapes decrease in the coarse scales. We may select the number of scales according to the desired feature vector size and the image size. Again, the L_1-distance is used to measure the similarity of feature vectors.

It is worthwhile to compare our work with that of Jacob *et al.* [19]. In [19], coefficients are quantized into -1 (if negative and significant), $+1$ (if positive and significant), and 0 (if insignificant). Thus, a gray-level image can be represented by a map consisting of three values: -1, 0, and $+1$. Their similarity metric is the L_p-distance measure, which is suitable for comparing objects with the same shape and the same location in two images but sensitive to translation, rotation and scale change of objects. Thus, the application of their method to content-based image retrieval is too restrictive. In contrast, we measure the normalized central spatial moments of the significant map in each subband for similarity test, which are more widely applicable. Besides, our method uses a smaller percentage of significant coefficients (25%) than theirs (50%) in determining the image shape map.

VI. Waveguide Prototype

A. System and Interface

We have built a wavelet-based query-by-example prototype for image indexing, searching, browsing and compression called the WaveGuide system. As illustrated in Fig. 7, WaveGuide has two basic building modules: the coding and indexing module and the decoding and retrieval module. The first module contains building blocks for the wavelet transform, the successive approximation quantizer, and the entropy coder, and the texture, color, and shape feature extraction engines. They are used to generate the compressed bit streams and indexing files, respectively. Every input image is indexed and compressed simultaneously by using this module. For image retrieval, the user first selects a query image through the WWW interface. Then, the system compares its texture, color, and shape descriptors with those of the images in the database and find out good matches, which are decoded and displayed for the user. Due to the multiresolution property of wavelet coding, the decoding and transmission of wavelet-coded images can be progressive. Therefore, the browsing and confirmation of image candidates are effective.

WaveGuide is a query-by-example system. We provide example images to guide users through their search of desired targets. The concept of query-by-example is based on the observation that many users have only vaguely defined information needs so that they may be able to recognize what they are looking for rather than describing or sketching it [15]. For such an application, pictorial examples and an interactive and cooperative human-machine interface can be of great help. Currently, WaveGuide does not support direct query on features, i.e., we do not provide feature pallettes (pickers), sketch boards, or painting tools. This can be easily added in the next version of the system.

To effectively access information in an image database, there are several design criteria for user interface [15]:

1) integration of various query mechanisms,
2) a visual or graphical user interface (GUI),
3) incorporation of user's relevant feedback,
4) support of user-guided navigation.

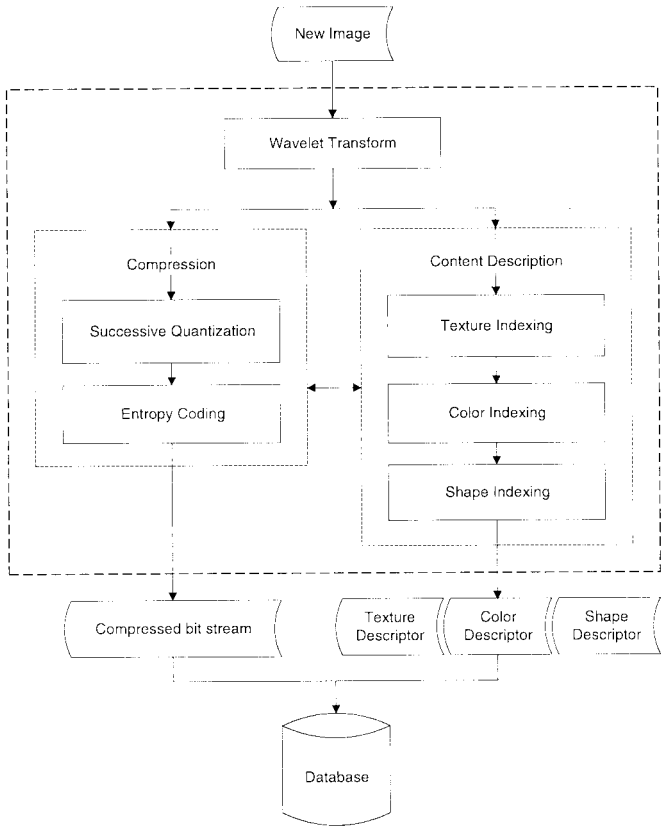


Fig. 7. Block diagram of the WaveGuide system. (a) Indexing and coding module. (b) Retrieval and decoding module.

To meet parts of these criteria, we implemented the WaveGuide interface on the World Wide Web (WWW) by using C, HTML, UNIX, and Common-Gateway-Interface (CGI). In the title Web page, we show a subset of images randomly selected from the database to give users an idea of the image types that can be accessed. The three query mechanisms—texture, color, and shape—are listed in the page. Users can select an image as the query image and select one or several features as querying features. The retrieved candidates are then displayed. Users can browse them or select one of them and several features for the next query process. Through such a process, users can access to the small portion of image data of their interest. Our primary goal with WaveGuide is to develop an efficient information filter to reduce the set of data that needs to be browsed at a later stage.

B. Wavelet Image Database

WaveGuide caters to still color images. At this moment, the experimental image database consists of 2127 natural images, including landscapes, animals, buildings, people, flowers, plants, etc. of size 192 $\times$ 128 from the Corel Professional Photos CD-ROM. Just as QBIC, we do not attempt to build up complicated data model with WaveGuide. The system concentrates on the signal processing techniques such as indexing, low-level feature extraction, similarity matching, and compression rather than techniques like data models, semantic world representation or annotation, which are usually applied in the systems developed by database researchers [15]. A complete data model for a natural image database can be a pretty difficult task. Due to the lack of data models, we also bypassed the use of query languages such as SQL or PSQL. Specifically, there are only two main data types in our system: scene and object, which is part of a scene.

The database is compressed in the YUV color space by the modern wavelet coder mentioned in Section II. In the wavelet transform, four and three levels of pyramid decompositions are performed for the luminance (Y) and chrominance $(U$ and $V)$ components, respectively. Consequently, there are thirteen subbands for the luminance component and ten subbands for each chrominance component. Texture, color, and shape features were extracted according to algorithms described in previous sections. Among these features, texture and shape descriptors are computed only from the Y-component of an image, and the color descriptors are computed from the Y-, U-, and V-components. The sizes of the proposed descriptors are given in Table II. The indexing cost is measured in terms of the number of elements in the feature vector and the number of bytes required to represent each feature vector (where 4 bytes is used to represent one floating number). Both the computational complexity and the memory requirement are controlled. As shown in Table II, the total size of the indexing file with our method is equal to 103 elements, which is less than that required by a 256-element color histogram [1], [13], [14], [30], [50].

C. Joint Content Description

The WaveGuide system allows users a number of content descriptors for image retrieval. When several descriptors are used simultaneously, it is necessary to integrate similarity scores resulting from the matching in different feature spaces. In this work, we adopt two methods to handle this issue

TABLE II
COMPARISON OF THE PROPOSED THREE DIFFERENT TYPES OF WAVELET DESCRIPTORS

Descriptor	Texture	Color	Shape
Type	no. of significant coef.	color moments	spatial moments
Size	13-element	9-element	81-element
Bytes	52	36	324

[14], [26]. The first one [14] is to normalize all scores in different spaces to the same range from zero to one, where zero represents perfect similarity (i.e., zero distance) and one, no similarity (i.e., the largest distance), and then add all the normalized scores with a weighting. Another approach [26] is to rank the images from one to N according to each individual score, where N is the total number of image items in the database. The final ranking of an image is the weighted sum of each individual ranking result. Both approaches were implemented in our work. It was observed that their performances are comparable. We choose the first approach as the default method in our system due to its lower computational complexity.

D. Retrieval Performance Evaluation

The retrieval performance of the WaveGuide prototype system has been evaluated. The retrieval efficiency is measured in terms of recall and precision [15], [29]. For each query image q in a database of size K ($K = 2127$ in our system), there are N_q such similar images. Let n_c, n_m, n_f, be the numbers of correct, missed, and false candidates, respectively, in the first M retrieved images with the smallest matching errors. The precision p_q and recall r_q for the query image q are defined as

$$p_q = \frac{n_c}{n_c + n_f} = \frac{n_c}{M} \tag{7}$$

and

$$r_q = \frac{n_c}{n_c + n_m} = \frac{n_c}{N_q}. \tag{8}$$

Ideally, we want a unity value of both parameters for a perfect recall and precision. In practice, since a perfect retrieval is difficult to obtain, a good balance between the two parameters is desired [15].

Preliminary experiments are carried out by using several image sets to evaluate the performance.[2] Note that the performance evaluation of image retrieval techniques is in general difficult since there is no commonly agreed image database for comparative study and the performance would highly depend on the selection of query image. Let us first demonstrate the retrieval of scene images. The test query set is the sunset image as shown in Fig. 8. Since these pictures are sunset scenes without clear objects, they can retrieved by texture and color. When the first Sunset image is used as the query image, eight items (i.e., the size of the query set) are retrieved from the database. The ideal retrieval will be that all images in the query set are ranked in the top eight positions.

[2]The color images can be accessed via the web site http://viola.usc.edu/extranet/IEEEIP 99nov/.

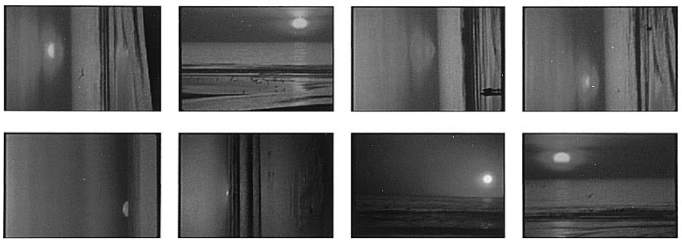


Fig. 8. Sunset query image set.

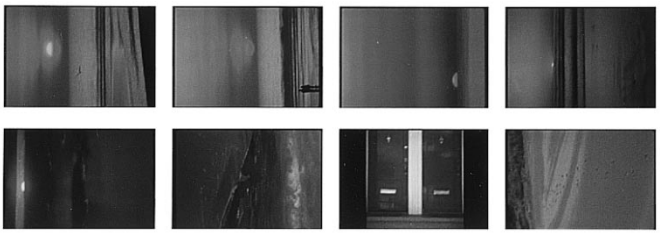


Fig. 9. Retrieved results based on the texture descriptor of the first sunset image, where items are ranked from the left to the right and then from the top to the bottom.

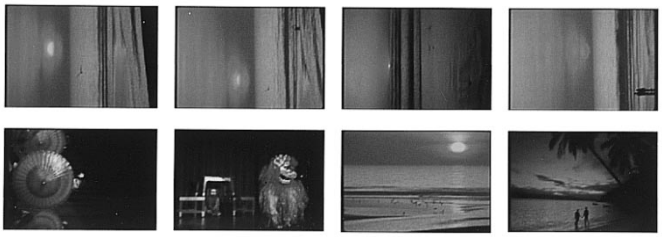


Fig. 10. Retrieved results based on the color descriptor of the first sunset image, where items are ranked from the left to the right and then from the top to the bottom.

Results by using the texture feature are shown in Fig. 9. It is clear that retrieved items have common textures such as sky, beach (sand), and water. Since the frequency feature is sensitive to the orientation, all the retrieved items have the same orientation. The three vertical Sunset queries—the second, seventh, and eighth images in the query image set—do not rank at the top eight positions. Besides, since the search is based on texture only, color unlikeness is possible (e.g., the color of items 6–8). In Fig. 9, four sunsets, the first to the fourth items are retrieved from the query set; the precision and the recall are both equal to 0.5. In addition to the four correct candidates, the fifth item is actually a similar image but from another category. Results of the query based on the color feature are shown in Fig. 10 for comparison. Since the color is a global feature, images of different orientations can be retrieved (e.g., the sixth to eighth items). Because images are matched by the color, results are not necessarily similar in textures. In Fig. 10, five sunset images, the first through the fourth and the seventh, are retrieved from the query set and one similar image, the eighth item, from other category. In this case, the precision and the recall rate are both $\frac{5}{8}$. Results by using a joint feature set of color and texture are shown in Fig. 11. The total distance is computed by the summation of 50% color distance and 50% texture distance. From Fig. 11, one can see that all items are similar in both color components

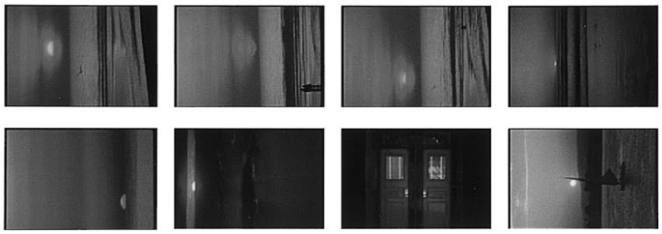


Fig. 11. Retrieved results based on joint texture and color descriptors of the first sunset image, where items are ranked from the left to the right and then from the top to the bottom.

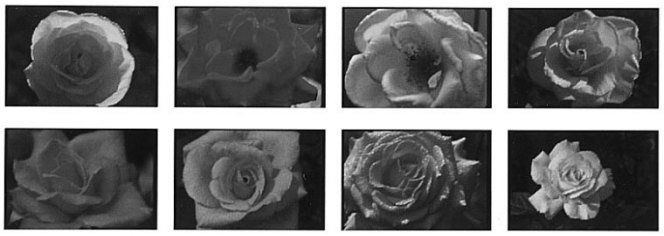


Fig. 12. Rose query image set.

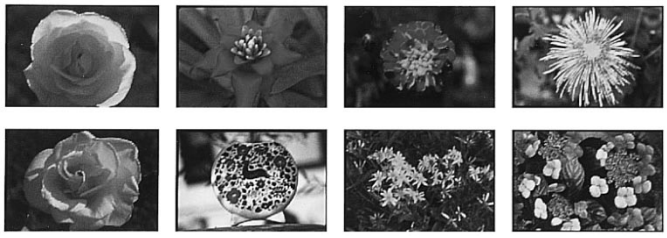


Fig. 13. Retrieved results based on the shape descriptor of the first rose image, where items are ranked from the left to the right and then from the top to the bottom.

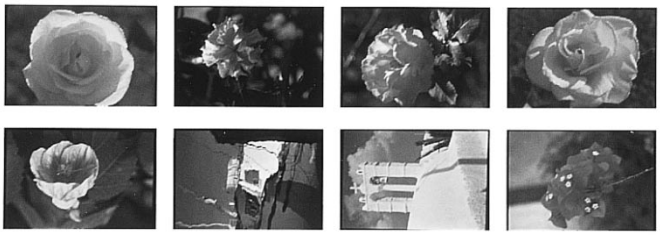


Fig. 14. Retrieved results based on the joint color, texture and shape descriptors of the first rose image, where items are ranked from the left to the right and then from the top to the bottom.

and textures, and the retrieval efficiency is superior to those of a single feature. In Fig. 11, there are five items from the sunset query set ranked in the top five positions, i.e., the precision and recall rate equal $\frac{5}{8}$.

The second query set is the rose image as shown in Fig. 12. Since these pictures have clear dominant objects, we use the shape feature as the main attribute, and the texture and color features as the auxiliary attributes. The first rose is used as the query image and eight items, which is the size of the query set, are retrieved from the database. Results by using the shape feature only is shown in Fig. 13. Items with clear round objects in the center, most of which are flowers, are retrieved. Results by using shape, texture, and color jointly are shown in Fig. 14. To emphasize the significance of the object shape, we compute the total distance based on 80% shape, 10% texture, and 10% color. We get a better result, i.e., images of a single flower with similar texture and color appear in the top positions.

VII. Conclusion and Extension

In this work, we proposed an integrated wavelet indexing and coding system and demonstrated the use of a joint feature set for content-based image retrieval. All the features—texture, color, and shape—are based on significant wavelet coefficients and their energy distribution among subbands and across quantization layers. In addition, sizes of content descriptors are carefully monitored to reduce the computational complexity and the memory space. Since images are compressed and indexed at the same time, the image database management problem can be greatly simplified.

The developed WaveGuide prototype system is far from completion. There are several parts which can be further improved. First, we would like to support the user feedback to achieve a truly interactive query process. Second, the database has to be enlarged to include more different types of images. Third, we would like to consider ways to reduce the amount of shape features and exploit the features in a more natural way. Fourth, it is interesting to see whether the wavelet-based descriptors can help in organizing the image database to facilitate the image retrieval process. Finally, it is important to find metrics to measure the performance of an image query engine in addition to precision and recall so that we can compare different query algorithms or systems in a more objective way.

References

[1] J. Ashley *et al.*, "Automatic and semi-automatic methods for image annotation and retrieval in QBIC," *SPIE Proc.: Storage and Retrieval for Image and Video Database III*, vol. 2420, pp. 24–35, Feb. 1995.

[2] E. M. Arkin *et al.*, "An efficient computable metric for comparing polygonal shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 209, Mar. 1991.

[3] J. R. Bach *et al.*, "The Virage image search engine: an open framework for image management," *SPIE Digital Image Storage Arch. Syst.*, vol. 2670, pp. 76–87, Feb. 1996.

[4] T. Bell, "Remote sensing," *IEEE Spectrum*, vol. 32, pp. 24–34, Mar. 1995.

[5] ISO/IEC-JTC1/SC2, "Progressive bi-level image compression," ISO Std. CD 11544, ISO, Sept. 1991.

[6] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Processing*, vol. 2, pp. 432–435, Oct. 1993.

[7] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 195–219, Mar. 1987.

[8] H.-I. Choi and W. J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 862–871, June 1989.

[9] N. S. Chang and K. S. Fu, "Query-by-Pictorial-Example," *IEEE Trans. Softw. Eng.*, vol. SE-6, pp. 519–524, June 1980.

[10] S.-F. Chang and J. R. Smith, "Extracting multi-dimensional signal features for content-based visual query," in *Proc. SPIE Symp. Visual Communication and Signal Processing*, May 1995.

[11] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, Jan. 1987.

[12] O. D. Faugeras and W. K. Pratt, "Decorrelation methods of texture feature extraction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 323–332, July 1980.

[13] M. Flickner *et al.*, "Query by image and video content: The QBIC system," *IEEE Trans. Comput.*, vol. 28, pp. 23–31, Sept. 1995.
[14] C. Faloutsos *et al.*, "Efficient and effective querying by image content," *J. Intell. Inform. Syst.*, vol. 3, pp. 231–262, July 1994.
[15] W. I. Grosky, R. Jain, and R. Mehrotra, *The Handbook of Multimedia Information Management.* Englewood Cliffs, NJ: Prentice-Hall, 1997.
[16] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610–621, Nov. 1973.
[17] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 179–187, Feb. 1962.
[18] K. Hirata and T. Kato, "Query by visual example," in *Proc. Third Int. Conf. Extending Database Technology: Advances in Database Technology EDBT'92*, Mar. 1992, pp. 56–71.
[19] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," in *Proc. SIGGRAPH Computer Graphics*, Los Angeles, CA, 1995, pp. 278–280.
[20] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[21] K.-C. Liang and C.-C. J. Kuo, "Progressive image indexing and retrieval based on embedded wavelet coding," *Proc. IEEE Int. Conf. Image Processing*, 1997.
[22] S. P. Lloyd, "Least square quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 127–135, Mar. 1982.
[23] J. Li, P.-Y. Cheng, and C.-C. J. Kuo, "Image compression using fast rate-distortion optimized wavelet packet transform," submitted for publication.
[24] K. I. Laws, "Texture image segmentation," Ph.D. dissertation, Image Processing Inst., Univ. Southern Calif., Los Angeles, 1980.
[25] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, Nov. 1993, pp. 1186–1191.
[26] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 722–733, July 1996; also MIT Media Lab. Tech. Rep. 320.
[27] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 837–841, Aug. 1996.
[28] M. K. Mandal, T. Aboulnasr, and S. Panchanathan, "Image indexing using moments and wavelets," *IEEE Trans. Consum. Electron.*, vol. 42, pp. 557–565, Aug. 1996.
[29] 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*, vol. 3022, pp. 380–389, Feb. 1997.
[30] W. Niblack *et al.*, "The QBIC project: Querying images by content using color, texture and shape," *Proc. SPIE Storage and Retrieval for Image and Video Databases I*, vol. 1908, pp. 173–187, Feb. 1993.
[31] W. K. Pratt, *Digital Image Processing*, 2nd ed. New York: Wiley, pp. 559–563.
[32] R. W. Picard, "Content access for image/video coding: The fourth criterion," MIT Media Lab. Tech. Rep. 295.
[33] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image database," *Proc. SPIE: Storage and Retrieval for Image and Video Databases II*, vol. 2185, pp. 34–47, Feb. 1994.
[34] ———, "Photobook: Content-based manipulation of image databases," *Int. J. Comput. Vis.*, Fall 1995; also MIT Media Lab. Tech. Rep. 255.
[35] D. Papadias and T. Sellis, "A pictorial query-by-example language," *J. Vis. Lang. Comput.*, vol. 6, pp. 53–72.
[36] T. R. Reed and H. Wechsler, "Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1–12, Jan. 1990.
[37] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.
[38] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3452–3454, Dec. 1993.
[39] J. R. Smith and S.-F. Chang, "Quad-tree segmentation for texture-based image query," *Proc. Second ACM Int. Multimedia Conf.*, Oct. 1994.
[40] M. Stricker and M. Orengo, "Similarity of color images," *Proc. SPIE*, vol. 2420, pp. 381–392, Feb. 1995.
[41] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol 7, pp. 11–32, 1991.
[42] J. R. Smith and S.-F. Chang, "VisualSEEk: A fully automated content-based image query system," in *Proc. Fourth ACM Int. Multimedia Conf.*, Nov. 1996, pp. 87–98.
[43] B. Scassellati, S. Alexopoulos, and M. Flickner, "Retrieving images by 2D shapes: A comparison of computation methods with human perceptual judgments," *Proc. SPIE*, vol. 2185, pp. 2–14, 1994.
[44] *MPEG-7: Context and Objectives (Ver. 4)*, Stockholm, Sweden, July 1997.
[45] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.
[46] G. Taubin and D. B. Cooper, "Recognition and position of rigid objects using algebraic moment invariants," *Proc. SPIE Geometric Methods in Computer Vision*, vol. 1570, pp. 175–186, 1991.
[47] M. Vetterli and J. Kovacevĭć, "Wavelets and subband coding," in *Discrete-Time Bases and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[48] H. Wang and C.-C. J. Kuo, "A multi-threshold wavelet coder (MTWC) for high fidelity image compression," in *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, CA, 1997, Oct. 26–29.
[49] I. H. Witten, R. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.
[50] H. J. Zhang, Y. Gong, C. Y. Low, and S. W. Smoliar, "Image retrieval based on color features: An evaluation study," *Proc. SPIE: Digital Image Archiv. Storage Syst.*, vol. 2606, pp. 212–220, Oct. 1995.

Kai-Chieh Liang (M'98) was born in Taipei, Taiwan, R.O.C. He received the B.S. degree in electrical engineering from National Cheng-Kung University, Tainan, Taiwan, the M.S. degree in solid state electronics from National Chiao-Tung University, Hsinchu, Taiwan, and the Ph.D. degree in signal and image processing from University of Southern California, Los Angeles, in 1988, 1990, and 1998, respectively.

Since 1998, he has been a Senior Engineer with Sharp Microelectronics of the Americas, Huntington Beach, CA. His research interests include content-based image retrieval, wavelet indexing, wavelet compression, vector quantization, and filterbanks.

C.-C. Jay Kuo (S'83–M'86–SM'92–F'99) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1980, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively, all in electrical engineering.

He was Computational and Applied Mathematics Research Assistant Professor in the Department of Mathematics, University of California, Los Angeles, from October 1987 to December 1988. Since January 1989, he has been with the Department of Electrical Engineering-Systems and the Signal and Image Processing Institute, University of Southern California, Los Angeles, where he currently has a joint appointment as Professor of electrical engineering and mathematics. His research interests are in the areas of digital signal and image processing, audio and video coding, wavelet theory and applications, multimedia technologies, and large-scale scientific computing. He has authored more than 350 technical publications in international conferences and journals.

Dr. Kuo is the Editor-in-Chief for the *Journal of Visual Communication and Image Representation.* He served as Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING (1995–1998) and IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1995–1997). He received the National Science Foundation Young Investigator Award and Presidential Faculty Fellow Award in 1992 and 1993, respectively. He is a member of SIAM, ACM, and SPIE.