Rotation and Gray-Scale Transform-Invariant Texture Classification Using Spiral Resampling, Subband Decomposition, and Hidden Markov Model

Wen-Rong Wu, *Member, IEEE,* and Shieh-Chung Wei

***Abstract*—This paper proposes a new texture classification algorithm that is invariant to rotation and gray-scale transformation. First, we convert two-dimensional (2-D) texture images to one-dimensional (1-D) signals by spiral resampling. Then, we use a quadrature mirror filter (QMF) bank to decompose sampled signals into subbands. In each band, we take high-order autocorrelation functions as features. Features in different bands, which form a vector sequence, are then modeled as a hidden Markov model (HMM). During classification, the unknown texture is matched against all the models and the best match is taken as the classification result. Simulations showed that the highest correct classification rate for 16 kinds of texture was 95.14%.**

I. INTRODUCTION

TEXTURE analysis plays an important role in computer vision and pattern recognition, and is widely applied to many areas such as industrial automation, biomedical image processing, and remote sensing. In this paper, we are concerned with the texture classification problem. Many methods presented in the literature approach the problem by assuming that samples of a texture all possess the same orientation and the same gray scale. When this assumption is not valid, most of these methods performs poorly.

In earlier approaches, attention was focused on analysis of the first-order or second-order statistics of textures [1]–[8]. Later, Gaussian Markov random fields or Gibbs random fields [9]–[15] were proposed to model textures. The advantage of this approach is that few parameters often describe a texture well. Laws [16] used local linear transformations, which can detect certain types of pattern commonly found in texture images, to compute features. His work introduced the concept of multichannel processing. Laws' work was extended by a number of researchers [17]–[20]. Recently, multiresolution and multichannel methods have been widely studied [21]–[26]. In particular, the wavelet transform and the Gabor filter play important roles. The wavelet transform is a multiresolution technique, which can be implemented as a pyramid or tree structure, and is similar to subband decomposition. The Gabor filter can extract the information in an ellipse-shaped frequency band and can be designed to detect some quasiperiodic texture patterns.

Kashyap and Khotanzad [14] first recognized the importance of rotation-invariant texture classification and developed a circular autoregressive (AR) model. However, their model could only be used for textures with no strong directionality. They then incorporated two AR models to overcome this problem. Strictly speaking, this approach is not truly invariant to rotation since the feature derived from AR coefficients is only invariant to rotation in multiples of 45 °. Cohen, Fan, and Patel [15] modeled textures as Gaussian Markov random fields and used the maximum likelihood (ML) method to estimate coefficients and rotation angles, later applying a generalized likelihood test for classification. This scheme is theoretically appealing. The problem is that the likelihood function is highly nonlinear and local minima may exist. In addition, the algorithm must be realized by using an iterative method that is computationally intensive. Chen and Kundu [26] used multichannel subband decomposition and a hidden Markov model (HMM) to solve the problem. They used a two-dimensional (2-D) quadrature mirror filter (QMF) bank to decompose a 2-D image into subband images and modeled the features of these subband images as an HMM. Texture samples with different orientations are treated as being in the same class. Although this approach did show promising results in experiments, it has some drawbacks. It is obvious that for an image, signal components in each subband will be different for different orientations. If we force texture samples with different orientations into the same class, variations in feature vectors will increase. When the number of texture classes is large, the classification result can be seriously affected.

Scenes imaged by different equipment or at different times may have different gray-scale values. This result is due to a complex process and is not easily modeled. In [26], Chen and Kundu used a simple model, essentially a linear function with an offset, to describe the phenomenon. They called this a gray-scale transform. Using this model, a transformed gray scale may be corrected by histogram equalization; however, this will increase the complexity of the classifier. Since it is desirable to have features invariant to gray-scale transformation, Chen and Kundu used the coefficients of skewness and kurtosis as features to achieve invariance. We will develop a more general method to cope with this problem.

In this paper, we propose a new scheme to attack the problem of rotation and gray-scale transform-invariant texture

Manuscript received October 21, 1994; revised January 13, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Maria Petrou.

W.-R. Wu is with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, ROC (e-mail: wrwu@cc.nctu.edu.tw).

S.-C. Wei is with Southern Information System Inc., Hsinchu, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7149(96)07187-4.

classification. The distinct feature of our approach is that we process the (1-D) signal obtained by spiral resampling. Given a digitized image, we resample the image along a spiral contour. The sampled 1-D signal is then considered as an observation of some random process. Because of the inherent characteristics, the statistical properties of the resampled signal is little affected by plane rotation. This is a significant advantage for HMM modeling. After resampling, the 1-D signal is passed through a QMF bank and features are extracted from each band. A normalization operation is performed at this stage to achieve invariance to gray-scale transformation. The sequence of subband features is then modeled as an HMM. The HMM is important here, since it explores dependencies among these subband features. During recognition, model parameters of an unknown texture are calculated and matched with those of known models and the one with the highest probability is selected as the output. Simulations show that our approach outperforms that in [26].

This paper is organized as follows. Section II describes what spiral sampling is and how to take spiral sampling evenly in texture images. Section III briefly describes subband decomposition, QMF banks, and the HMM. Section IV introduces the whole structure of our texture classification scheme. Then we performed simulations and report results in Section V. Finally, conclusions are drawn in Section VI.

II. Spiral Sampling

A. Spiral Sampling Theorem

A continuous spiral can be described as

$$\theta = C\rho \tag{1}$$

where ρ is the radius, θ is the corresponding angle, and C is a constant describing the radial distance. The radial distance is defined as the smallest distance between two points with the same angle (modulus of 2π). In real applications, we use discrete spirals. This can be done by sampling ρ. Generally, we want a uniform sampled spiral. To get that, we first have to consider the problem of coordinate transformation. Assume that the mapping of a point on a Cartesian plane $w = (\rho, \theta)$ onto a point on another Cartesian plane $z = (x, y)$ is performed according to the transformation $z = f(w)$. Let a sequence of points w_k, for $k = 1, 2, \cdots, N$ on plane w be mapped to plane z, and define the length between two consecutive points w_k, w_{k+1} in w plane and z_k, z_{k+1} in z plane as

$$|\triangle w_k| = |w_{k+1} - w_k|, \qquad |\triangle z_k| = |z_{k+1} - z_k|. \tag{2}$$

If $f(\cdot)$ is analytic and $|\triangle w_k|$ is small, then we have [28] the following:

$$|\triangle z_k| = |f'(w_k)| \cdot |\triangle w_k| \tag{3}$$

where

$$|f'(w_k)| = \sqrt{J\left(\frac{x, y}{\rho, \theta}\right)} \tag{4}$$

and $J(\frac{x,y}{\rho,\theta})$ is the Jacobian. Consider the transformation from $w = (\rho, \theta)$ to $z = (x, y)$ and the transformation function is

$$x = \rho\cos\theta, \qquad y = \rho\sin\theta. \tag{5}$$

Then, (1) is a straight line in w plane and a spiral in z plane. The Jacobian is

$$J\left(\frac{x, y}{\rho, \theta}\right) = \begin{vmatrix} \frac{\partial x}{\partial \rho} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial \rho} & \frac{\partial y}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos\theta & -\rho\sin\theta \\ \sin\theta & \rho\cos\theta \end{vmatrix} = \rho. \tag{6}$$

Then, $|f'(w)| = \sqrt{\rho}$. If we sample ρ uniformly, we have

$$\rho_k = k\Delta, \qquad \theta_k = C\rho_k \tag{7}$$

where $k = 0, 1, 2, \cdots$, and Δ is a constant describing the density of spiral samples. In this case, it is easy to see that $|\triangle w_k|$ is a constant.

$$\begin{aligned} |\triangle w_k| &= \sqrt{[(k+1)\Delta - k\Delta]^2 + C^2[(k+1)\Delta - k\Delta]^2} \\ &= \sqrt{1 + C^2}\Delta. \end{aligned} \tag{8}$$

This indicates that the sampling density is uniform along the straight line on the w-plane. However, from (3), we find that

$$|\triangle z_k| = \sqrt{k\Delta} \cdot |\triangle w_k|. \tag{9}$$

$|\triangle z_k|$ is not a constant. The sampling density is not uniform along the spiral on the z-plane. The bigger the k, the lower the density is; therefore, we have to consider another kind of transformation.

Let $r = \rho^2$ and consider the transformation from plane $v = (r, \theta)$ to plane $z = (x, y)$. Using (5) and the chain rule, we have the Jacobian

$$J\left(\frac{x, y}{r, \theta}\right) = \begin{vmatrix} \frac{1}{2r}\cos\theta & -r\sin\theta \\ \frac{1}{2r}\sin\theta & r\cos\theta \end{vmatrix} = 1 \tag{10}$$

and $|f'(v)| = 1$. Define $|\triangle v_k| = |v_{k+1} - v_k|$. If we sample r as that in (7), i.e., $r = k\Delta$, we have $|\triangle v_k| = |\triangle z_k|$. Note that the relationship of r to θ becomes $\theta = C\sqrt{r}$, which is no longer a straight line and, thus, $|\triangle v_k|$ is not a constant. Consider two consecutive points $v_{k+1} = ((k+1)\Delta, C\sqrt{(k+1)\Delta})$ and $v_k = (k\Delta, C\sqrt{k\Delta})$. $|\triangle v_k|$ is calculated as

$$\begin{aligned} |\triangle v_k| &= \sqrt{[(k+1)\Delta - k\Delta]^2 + [C\sqrt{(k+1)\Delta} - C\sqrt{k\Delta}]^2} \\ &= \sqrt{(\Delta + C^2)[(2k+1) - 2\sqrt{k(k+1)}]\Delta} \end{aligned} \tag{11}$$

which is dependent on k. However, when k is large, we can make an approximation that $\sqrt{k(k+1)} \cong k$. Thus

$$|\triangle v_k| \cong \sqrt{(\Delta + C^2)\Delta} \tag{12}$$

and $|\triangle v_k| \cong |\triangle v_{k+1}|$. This means that the sampling density on the z plane as well as on the v plane are approximately uniform. It can be seen that $\sqrt{k(k+1)}$ converges to k quickly.

The sampling technique described above gives a uniformly sampled spiral, but this doesn't mean that the sampling positions are evenly spaced. Define the radial distance, d_r, as the smallest distance between two points with same angle (modulus of 2π), and the lateral distance, d_l, as the distance between two consecutive spiral points. To have evenly spaced spiral sample, d_l, d_r must be constant and identical. Since we

have made d_l constant above and d_r is inherently constant, the only thing left is to choose Δ and C such that d_r and d_l are equal to a desired value. This is done as follows.

First, we choose a desired value d. Consider the two points (ρ, θ) and $(\rho', \theta + 2\pi)$. Let them satisfy $\theta = C\rho$ and $\theta + 2\pi = C\rho'$. Subtracting the first equation from the second and noting that $\rho' - \rho = d$, we have

$$C = \frac{2\pi}{d}. \tag{13}$$

Now we need to find the value of Δ. The $k+1$th sample of the spiral is $(\sqrt{k\Delta}, C\sqrt{k\Delta})$ and $k+2$th sample is $(\sqrt{(k+1)\Delta}, C\sqrt{(k+1)\Delta})$ on the w plane. Fig. 1 plots the two points mapped to the z plane. From the trigonometric property, we know that

$$\begin{aligned} \overline{BC}^2 &= \overline{AB}^2 + \overline{AC}^2 - 2\overline{AB}\cdot\overline{AC}\cos\phi \\ d^2 &= (k+1)\Delta + k\Delta - 2\sqrt{(k+1)\Delta}\sqrt{k\Delta} \\ &\quad \times \cos\{C[\sqrt{(k+1)\Delta} - \sqrt{k\Delta}]\}. \end{aligned} \tag{14}$$

Assume that Δ_k is the solution of Δ when k is substituted. We can solve Δ_k by some numerical method. For different k's, Δ_k's are not completely the same (though they are very close). We plot Δ_k for $k = 0, 1, 2, \cdots, N-1$ with $d = 5$ in Fig. 2 (N is the number of samples). Except for the first few values, Δ_k's are almost the same, especially for large k. So we take Δ_k with large k as our Δ. We have performed simulations to find the relationship between d and Δ. For ease of future use, we apply the least-squares curve-fitting technique to obtain the following mathematical formula (for $k = 100$):

$$\Delta = 0.317\,914\,367\,11 \cdot d^2. \tag{15}$$

(14) and (15) are plotted in Fig. 3. From this figure, we find that the fitting is so good that we cannot distinguish the approximation from the actual one. In summary, we obtain evenly spaced spiral data by the following procedure:

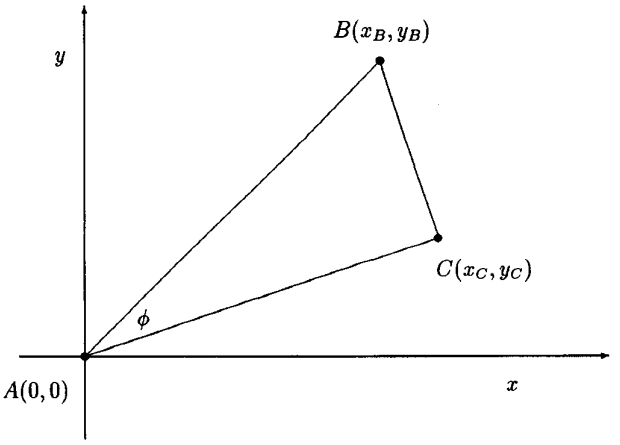
1) Determine the value of d.
2) Calculate the value of C by (13).
3) Calculate the value of Δ by (15).
4) For $k = 0, 1, 2, \cdots, N-1$

$$\rho_k = \sqrt{k\Delta}, \qquad \theta_k = C\rho_k. \tag{16}$$

5) Transform to the xy-plane.

$$x_k = \rho_k \cos\theta_k, \qquad y_k = \rho_k \sin\theta_k. \tag{17}$$

The data set $\{(x_k, y_k) | k = 0, 1, \cdots, N-1\}$ is the evenly-spaced spiral data. An example of this spiral is shown in Fig. 4.



$$(\sqrt{(k+1)\Delta}, C\sqrt{(k+1)\Delta}) \xrightarrow{f(w)} (x_B, y_B)$$

$$(\sqrt{k\Delta}, C\sqrt{k\Delta}) \xrightarrow{f(w)} (x_C, y_C)$$

$$\overline{AB} = \sqrt{(k+1)\Delta}$$

$$\overline{AC} = \sqrt{k\Delta}$$

$$\phi = C\left[\sqrt{(k+1)\Delta} - \sqrt{k\Delta}\right]$$

$$\overline{BC} = d$$

Fig. 1. Plot of $k+1$th and $k+2$th sample of $\theta = c\rho$.

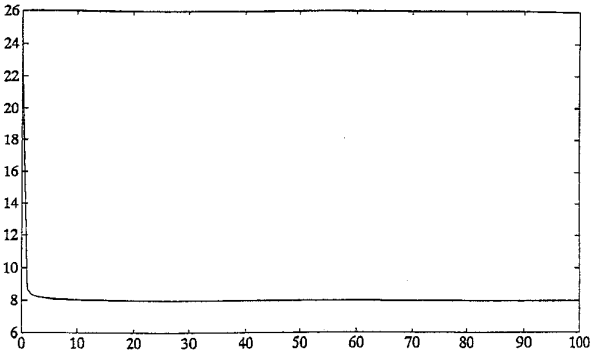


Fig. 2. Relationship of k and Δ_k for $d = 5$.

B. The Rotation-Invariant Property

Given a continuous image, the next problem we are concerned with is whether the spiral sampling will give similar signals no matter what degree to which the image is rotated. We use the word "similar" here, since it is almost impossible to obtain the exactly same signals. First, consider the nonuniform sampling in (7). Let the spiral center coincide with the center of rotation. It is simple to see that, as far as sampling is concerned, rotating an image is equivalent to rotating the spiral (the image remains unchanged). From (7), we see that $\theta_k = kC\Delta$. Thus, if we let m be an integer and rotate the spiral by an angle of $mC\Delta$, all the sampling points will move inwardly/outwardly along the radial direction. If the radial distance d_r is small, the sampled signal of the rotated spiral is simply a shift version of the original. We depict this property in the example shown in Fig. 5. In the figure, $s(1)$–$s(N)$ denote the sampling positions of the original spiral, and $r(1)$–$r(N)$ denote those of the rotated spiral. Let the polar coordinates of $s(k)$ and $r(k)$ be (ρ_k^s, θ_k^s) and

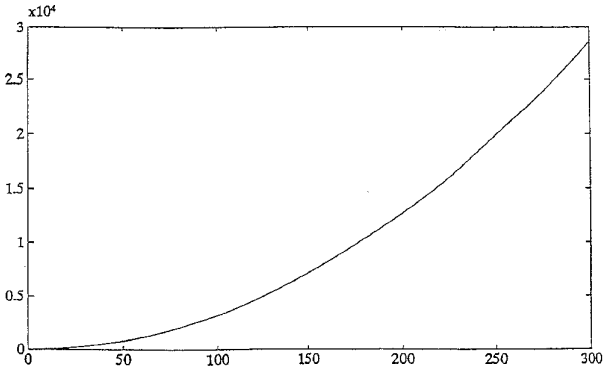


Fig. 3. Relationship of d and Δ.

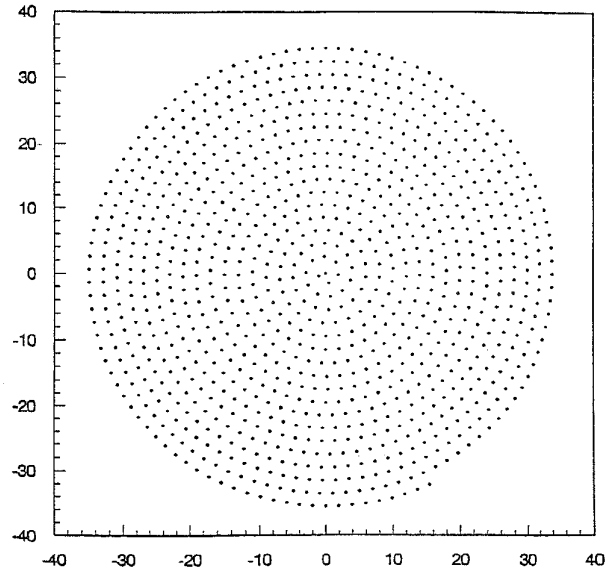


Fig. 4. Evenly spaced spiral sample with $d = 2$.

(ρ_k^r, θ_k^r), respectively. For this particular example, we see that θ_k^s and θ_{k-1}^r are identical for $k = 2, \cdots, N$. Although ρ_k^s and ρ_{k-1}^r are different, $|\rho_k^s - \rho_{k-1}^r| \leq d_r/2$. The equality holds when the rotation angle is 180°. Thus, if d_r is made small enough, $f[s(k)] \approx f[r(k-1)]$ for $k = 2, \cdots, N$ where $f[x]$ denotes the sampled signal at x position. In general, we can have $f[s(k)] \approx f[r(k-P+1)]$, $k = P, \cdots, N$ for a counterclockwise rotation and $f[s(k-P+1)] \approx f[r(k)]$ for a clockwise rotation where P is an integer and its value is determined by the sampling density and the rotation angle. For the uniform sampling scheme in (16), a similar formation does not exist. Therefore, the sampled signal of the rotated image will have some distortion compared to that of the unrotated one. However, as we will show later, the distortion is small and can be ignored in real applications.

To better understand the concept described above, we show a simple example of spiral sampling here. Let the intensity of an image be $f(x, y)$ and $f(x, y) = 3x - 2y$, where x and y are coordinates inside the image. The center of the spiral is set at the origin and the image size is 30 × 30. Fig. 7 shows two spiral-sampling results corresponding to the

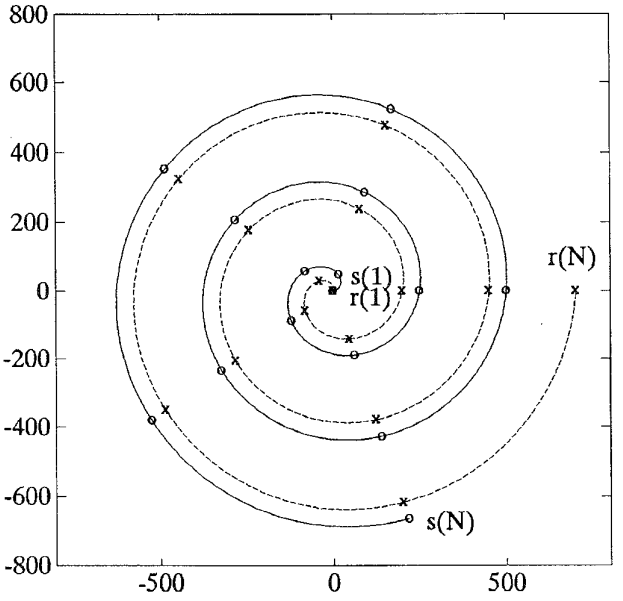


Fig. 5. Sampled spiral and its rotated version.

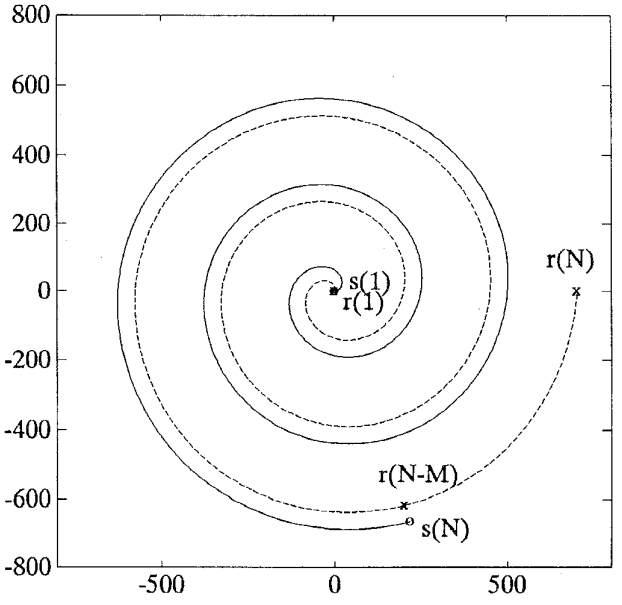


Fig. 6. Sampled spiral and its rotated version.

original (solid line) and the 90°-rotated (dashed line) images for the nonuniform sampling scheme. As we can see, apart from a shift, these two signals are almost identical. Fig. 8 shows the magnitude plots of their spectra. Fig. 9 shows the spirally sampled signals for the uniform sampling scheme and Fig. 10 is the magnitude plot of their spectra. From Fig. 9, we see that the amount of shift in the sampled signal of the rotated image is time varying (pretend that the horizontal axis represents time); the bigger the time index, the larger the shift is. Even with this distortion, the main characteristic of the signal is preserved. This can be seen from the waveforms and the peaks of their spectra.

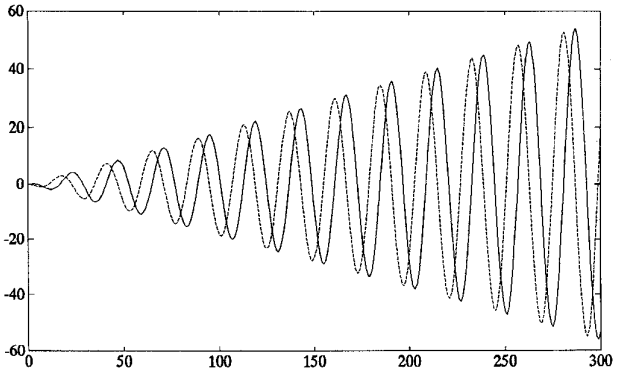


Fig. 7. Resulting signals for nonuniform spiral sampling.

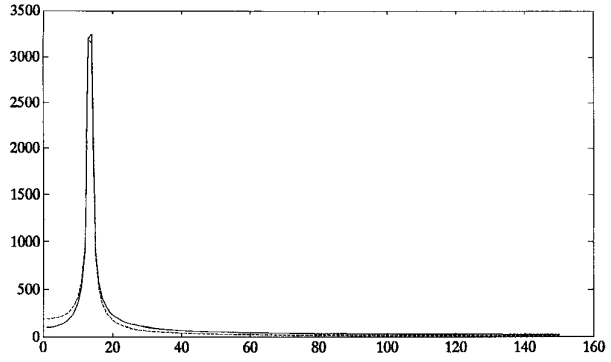


Fig. 8. Amplitude spectra of signals in Fig. 7.

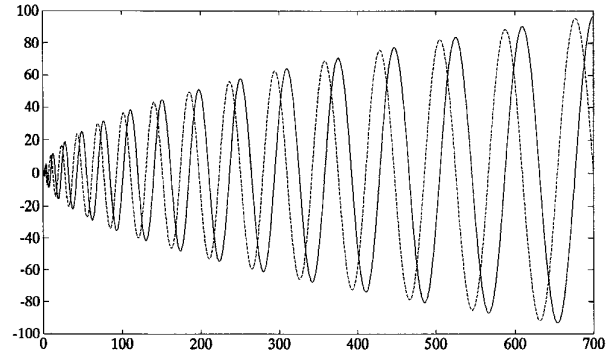


Fig. 9. Resulting signals for uniform spiral sampling.

Now we analyze signal distortion due to rotation in the uniform sampling scheme. In Fig. 6, we show two spirals: $s(1)$–$s(N)$ is the original one and $r(1)$–$r(N)$ is the rotated one. From the figure, it is apparent that the sampled signal at $s(N)$ does not correspond to that of $r(N)$; instead, it corresponds to $r(N-M)$ where M is some integer. Then, the sampled signal at $r(N-M)$ is shifted by the arc length from $r(N-M)$ to $r(N)$. Note that there is no shift at the starting point. Thus, the original spiral from $s(1)$ to $s(N)$ is compressed into the rotated spiral from $r(1)$ to $r(N-M)$. This makes the amount of shift time variant and distorts the sampled signal. The arc length from $r(N-M)$ to $r(N)$ is the maximum amount of shift. If this amount is small compared with the signal length, the compression effect is not significant. Let R be equal to the ratio of the number of sampling points on the arc $r(N-M)$–$r(N)$ and the total sampling points on the spiral. Consider a square image with side-length W. For simplicity, let the d value in (15) be 1 and the end point of the spiral reach the boundary of the image. The number of the sampling points on the arc $r(N-M)$ to $r(N)$ can be approximated by $\pi W(\theta/2\pi)$ where θ is the rotation angle and the number of sampling points on the whole spiral can be approximated by $W^2(\pi(W/2)^2/W^2) = W^2\pi/4$. Thus, R equals $4\theta/2W\pi$. In the worst case, where $\theta = \pi$, $R = 2/W$. For a 128 × 128 image, this ratio is less than 2%. The distortion is then small and can be ignored.

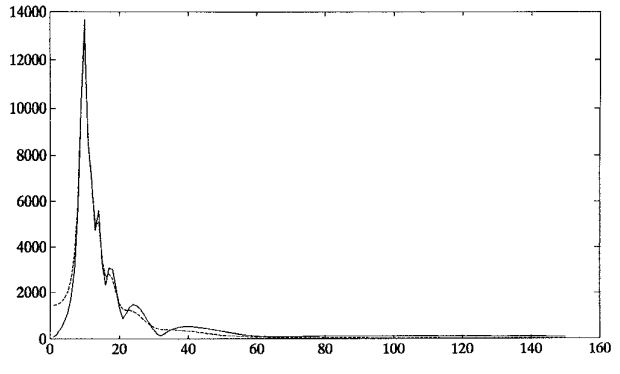


Fig. 10. Amplitude spectra of signals in Fig. 9.

The above results were derived for an image rotated to different angles. Given another texture image of the same class, if we apply the spiral sampling scheme, the resulting signal may be totally different from previous ones. However, we can assume that the signal has statistical properties similar to those of previous ones. Note that this is an extension of the common assumption that sampled signals along straight lines with the same slopes have similar statistical properties for texture images of the same class. Based on this assumption and our previous results, we conclude that the spirally sampled signals have similar statistical properties for texture images of the same class, and these properties are present regardless of the rotation center and the rotation angle. This is the foundation of our approach.

In most applications, we deal with digitized images, which means we have to resample the images. Interpolation techniques may be useful here. Since the spiral sampling is not typical, some form of aliasing may occur. However, if the resampling density is comparable to the original digitized image, this effect is small. Note that even the resampled signal is aliased, we can still carry out classification as long as we can extract useful information from the sampled signal. This is different from *reconstruction*, where aliasing may not be acceptable.

III. Subband Decomposition and HMM

A. Subband Decomposition

We treat the 1-D resampled signals from texture images in the same class as observations from a random process. Since

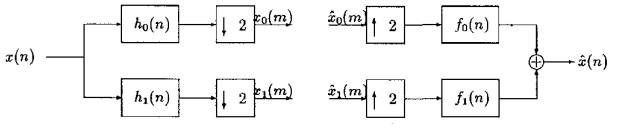


Fig. 11. Two-band analysis/synthesis QMF.

the 1-D signal carries 2-D information, the power spectrum may have a complicated shape and is not easily characterized using conventional methods. Even if we apply a high-order AR model, for example, we may still not be able to describe local properties of the power spectrum. However, if we consider the power spectrum just inside a small bandwidth, a low-order AR model may be sufficient. In other words, we can divide the power spectrum into consecutive segments and use a low-order AR model for each segment. This is similar to the spline-function approach in the curve-fitting problem. The subband decomposition is a perfect way to realize the idea mentioned above. Note that the decomposed signals are also random. In what follows, we briefly describe the basic concept of subband decomposition.

Subband decomposition can be realized by the QMF bank [30]. The QMF bank is composed of analysis filters (decimators) that are used to partition the signal into consecutive frequency bands, and synthesis filters (interpolators) that are used to combine the partitioned signals back into the original signal. The principle of QMF bank design can be illustrated by considering the two-band analysis/synthesis structure shown in Fig. 11. In this figure, the filters $h_0(n)$ and $h_1(n)$ represent the lowpass and highpass analysis filters. Similarly, the filters $f_0(n)$ and $f_1(n)$ represent the lowpass and highpass synthesis filters. $x(n)$, $x_0(m)$, $x_1(m)$, $\hat{x}_0(m)$, and $\hat{x}_1(m)$ denote the original signal, the output signal of $h_0(n)$, the output signal of $h_1(n)$, the input signal of $f_0(n)$, and the input signal of $f_1(n)$.

Let the Fourier transform of $x(n)$, $x_0(m)$, $x_1(m)$, $h_0(n)$, and $h_1(n)$ be $X(e^{j\omega})$, $X_0(e^{j\omega})$, $X_1(e^{j\omega})$, $H_0(e^{j\omega})$, and $H_1(e^{j\omega})$. We then have

$$X_0(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega/2})H_0(e^{j\omega/2}) + X(e^{(j\omega+2\pi)/2})H_0(e^{(j\omega+2\pi)/2})] \quad (18)$$

$$X_1(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega/2})H_1(e^{j\omega/2}) + X(e^{(j\omega+2\pi)/2})H_1(e^{(j\omega+2\pi)/2})]. \quad (19)$$

Similarly, let $\hat{X}_0(e^{j\omega})$, $\hat{X}_1(e^{j\omega})$, $F_0(e^{j\omega})$, $F_1(e^{j\omega})$ and $\hat{X}(e^{j\omega})$ be the Fourier transforms of $\hat{x}_0(m)$, $\hat{x}_1(m)$, $f_0(n)$, $f_1(n)$, and $\hat{x}(n)$, respectively. Then the synthesis relationship is

$$\hat{X}(e^{j\omega}) = \hat{X}_0(e^{j2\omega})F_0(e^{j\omega}) + \hat{X}_1(e^{j2\omega})F_1(e^{j\omega}). \quad (20)$$

Combining (18) and (19) and using the relationships in which $\hat{X}_0(e^{j\omega}) = X_0(e^{j\omega})$, $\hat{X}_1(e^{j\omega}) = X_1(e^{j\omega})$, we obtain the output frequency response in the form

$$\hat{X}(e^{j\omega}) = \frac{1}{2}[H_0(e^{j\omega})F_0(e^{j\omega}) + H_1(e^{j\omega})F_1(e^{j\omega})]X(e^{j\omega}) + \frac{1}{2}[H_0(e^{j(\omega+\pi)})F_0(e^{j\omega}) + H_1(e^{j(\omega+\pi)})F_1(e^{j\omega})] \times X(e^{j(\omega+\pi)}). \quad (21)$$

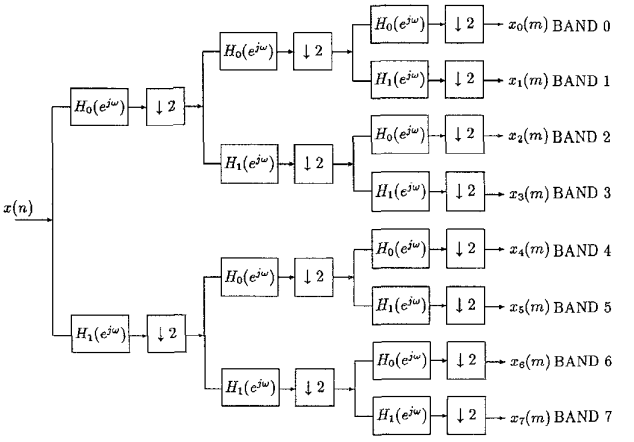


Fig. 12. Example of three-stage tree structure for an eight-band filter.

The first term in (21) represents the original signal, and the second term represents the undesired aliasing components. It has been shown that through proper design, aliasing due to decimation in the analysis structure can be exactly canceled by imaging due to interpolation in the synthesis structure [30]. The two-band design described above can be conveniently cascaded into a tree structure. Other kinds of implementation are possible. However, due to its simple structure, we consider only the tree structure here. An example of this structure is shown in Fig. 12.

B. Hidden Markov Model (HMM)

After subband decomposition, we obtain a set of subband signals. A feature vector is then extracted for each band. Since the decomposed signals are random, the vectors are also random. Here, we have two choices to continue the processing. The first is to combine all feature vectors into one big vector such that the conventional Bayesian classifier can be applied. However, note that there may be correlation between components of the big vector. Thus, we have to estimate its correlation matrix. Since the vector is usually large, this approach is difficult. We can ignore the correlation and estimate only the variance of each component, but this will degrade the performance of the classifier. The second choice is to put these vectors in order (from the lowest frequency band to the highest frequency band) to form a sequence of vectors. This sequence of vectors is seen as an observation of some vector process, which allows a model to be built for the process. The HMM has proven to be powerful in modeling a vector process. Thus, we adopted the HMM to implement our classification scheme. The theoretical background of the HMM and related problems are stated as follows.

A Markov process is a stochastic process whose past has no influence on the future if its present is specified. Let x_t denote a stochastic process and $f(\cdot)$ the density function. Then

$$f(x_t|x_{t-1}, x_{t-2}, \ldots) = f(x_t|x_{t-1}). \quad (22)$$

In an HMM [29], the observation, which is also called the symbol, does not have the Markov property. The symbol distribution is dependent on the so-called "state." It is the state

that possesses the Markov behavior. Let the state at time t be denoted as i_t. Then

$$f(i_t|i_{t-1}, i_{t-1}, \cdots, i_1) = f(i_t|i_{t-1}). \tag{23}$$

Usually, the number of states is finite, say L. Thus, L symbol distributions are associated with the L states. We now define the elements of the HMM as follows.

T = length of the observation sequence
L = number of states in the model
$Q = \{q_1, q_2, \cdots, q_N\}$ = HMM states
$O = \{o_t\}$ = observation sequence, $t = 1, 2, \cdots, T$
$\Pi = \{\pi_i\}$ = initial state probabilities
$\pi_i = P(q_i \text{ at } t = 1)$.
$A = \{a_{ij}\}$ = state transition probability
$a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)$.
$B = \{b_j(o_t)\}$ = observation symbol density
$b_j(o_t) = f(o_t|q_j)$.

For convenience, we use the compact notion $\lambda = (A, B, \Pi)$ to represent an HMM. Using the HMM with specified λ, an observation sequence $O = \{o_1, o_2, \cdots, o_T\}$ can be generated as follows:

1) Choose an initial state, i_1, according to the initial state distribution.
2) Set $t = 1$.
3) Choose o_t according to $b_{i_t}(o_t)$, the symbol probability density in state i_t.
4) Choose i_{t+1} according to $a_{i_t i_{t+1}}$, the state transition probability.
5) Set $t = t + 1$; return to step 3 if $t \leq T$; otherwise, terminate the procedure.

Similarly, sequences of observations can be modeled as HMM's. For the pattern recognition problem, λ can been taken as the feature vector. To realize pattern recognition with HMM's, we have to consider the following three problems.

P1—Given the model $\lambda = (A, B, \Pi)$ and observation sequence $O = o_1, o_2, \cdots, o_T$, what is the probability of the observation sequence $P(O|\lambda)$?

P2—Given the observation sequence $O = o_1, o_2, \cdots, o_T$, how can we select an optimal state sequence $I = i_1, i_2, \cdots, i_T$?

P3—Given the observation sequence $O = o_1, o_2, \cdots, o_T$, how do we find the model parameters $\lambda = (A, B, \Pi)$ to maximize $P(O|\lambda)$?

P1 and P2 are mainly concerned with the *matching* phase of the recognition process, and P3 is concerned with the *training* phase. The solutions of these three problems have been found. For details, see [29].

IV. Texture Classification

In order to solve the rotation and gray-scale transform-invariant problem for texture classification, we combine spiral sampling, subband decomposition, and the HMM techniques. The block diagram of the texture-classification algorithm is shown in Fig. 13. First, we obtain 1-D signals by spiral resampling, remove the signals' means, split them into consecutive frequency bands, and extract the normalized features from

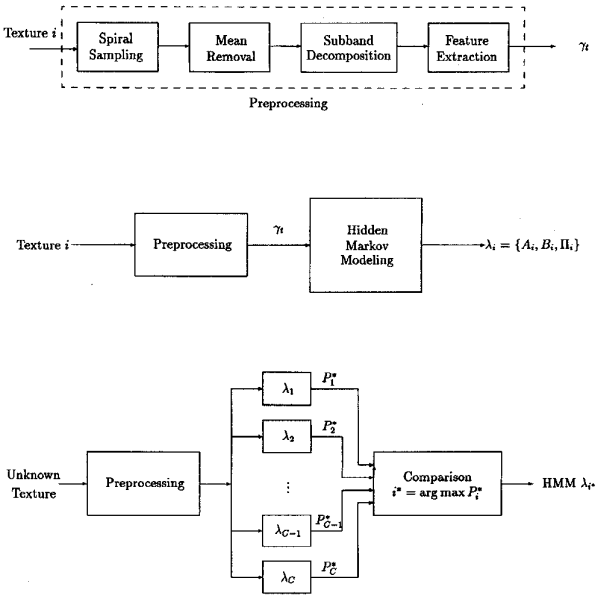


Fig. 13. Block diagram of texture recognition; training and testing.

each band with second-order or higher order autocorrelation functions. This is what we call the preprocessing stage. The purpose of spiral sampling is to solve the rotation-invariant problem, while mean removal and feature normalization solve the gray-scale transform-invariant problem. Features in each band then form a sequence of vectors that are modeled as an HMM. It is the HMM that describes the random nature of the texture signal. Note that even structured textures may be described using our method. This is the power of subband decomposition. For a structured texture, widely spaced samples can be correlated. Thus, it is difficult to use correlation functions to characterize the texture. However, in the subband domain, due to the filtering and decimation, signals become less correlated and correlation functions may be useful. After the features of the training samples have all been derived, we train the HMM parameter. Once the models are all built, an unknown object's features can be matched against each model and the best match is considered as the classification result. In the following, we describe each stage in detail.

A. Preprocessing/Spiral Resampling

In simulations, we need many rotated images for training and testing. Theoretically, we have to rotate actual texture pictures and scan them. Since this is time consuming, we instead rotated digitized images and spirally resampled them. As we mentioned in Section II, this is equivalent to rotating spiral contours and letting texture images remain unchanged. The rotated spiral can easily be accomplished as follows:

$$\rho_k = \sqrt{k\Delta}, \qquad \theta_k = C\rho_k + \theta_c \tag{24}$$

where θ_c denotes the rotation angle. Let the image size be $W \times W$. The following constraint must be satisfied for sampling:

$$\sqrt{(N-1)\Delta} \leq \frac{W}{2} \tag{25}$$

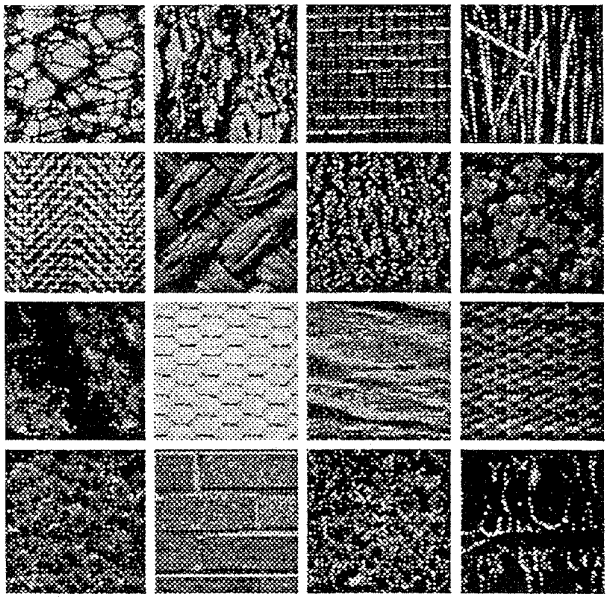


Fig. 14. Textures used in experiments (from top to bottom, left to right: D111, D12, D14, D15, D17, D18, D24, D28, D2, D34, D37, D55, D92, D94, D9, D10).

where N is the number of samples. We let the equality hold and obtain the maximum Δ as follows:

$$\Delta = \frac{W^2}{4(N-1)}. \tag{26}$$

B. Preprocessing/Mean Removal

The gray-scale transform mentioned in Section I can be described as follows:

$$I_t(x,y) = \alpha I(x,y) + \beta \tag{27}$$

where $I(x,y)$ is the gray level at (x,y), α is a positive scale factor, and β is a shift factor. Here, we assume that $0 \le I_t(x,y) \le 255$. In classification, images transformed with different α's and β's are considered identical. Thus, we need an algorithm that is invariant to gray-scale transformation. This is accomplished in two steps. First, we remove the mean of the sampled signal to null the effect of β. The effect of scale constant α can be removed in feature formulation as described below.

C. Preprocessing/Subband Decomposition

The mean removed signal is then passed to the QMF bank. Let the original sampled signal be $g(k)$, the gray-scale transformed signal be $g_t(k)$, the mean removal signal of $g(k)$ be $\tilde{g}(k)$, and the mean removal signal of $g_t(k)$ be $\tilde{g}_t(k)$. We now show that the subband signals of $\tilde{g}(k)$ and $\tilde{g}_t(k)$ also differ in a scalar factor α. Consider a two-band case. Let the subband signals of $\tilde{g}(k)$ and $\tilde{g}_t(k)$ be $x_0(k), x_1(k)$ and $x_{t0}(k), x_{t1}(k)$, respectively. From (18) and (19), we can obtain

$$X_i(e^{j\omega}) = \frac{1}{2}[\tilde{G}(e^{j\omega/2})H_i(e^{j\omega/2}) + \tilde{G}(e^{(j\omega+2\pi)/2})H_i(e^{(j\omega+2\pi)/2})] \tag{28}$$

$$X_{ti}(e^{j\omega}) = \frac{1}{2}[\tilde{G}_t(e^{j\omega/2})H_i(e^{j\omega/2}) + \tilde{G}_t(e^{(j\omega+2\pi)/2})H_i(e^{(j\omega+2\pi)/2})] \tag{29}$$

where $\tilde{G}(e^{j\omega}), \tilde{G}_t(e^{j\omega}), X_i(e^{j\omega})$, and $X_{ti}(e^{j\omega})$ are the Fourier transforms of $\tilde{g}(k), \tilde{g}_t(k), x_i(k)$ and $x_{ti}(k)$ and i=0 or 1. Since the mean is removed, we have $\tilde{G}_t(e^{j\omega}) = \alpha \cdot \tilde{G}(e^{j\omega})$. Combining this with (28) and (29), we have the following result:

$$X_{ti}(e^{j\omega}) = \alpha \cdot X_i(e^{j\omega}) \tag{30}$$

or $x_{ti}(k) = \alpha \cdot x_i(k)$.

D. Preprocessing/Feature Extraction

As mentioned previously, the subband decomposed signals are random. Here, we model each subband decomposed signal as an observation of a non-Gaussian random process, which is the output of a linear system with a non-Gaussian white-noise input. Conventional approaches to characterizing random processes use second-order statistics. However, as we know, second-order statistics do not provide phase information about the linear system. Phase information is important in some applications. Thus, we propose to use high-order statistics, specifically the high-order autocorrelation functions, as features. Let the number of subbands be B and the number of samples in each band be D ($N = BD$). For the ith band, the autocorrelation function is defined as

$$R_i(\tau_1, \tau_2, \cdots, \tau_M) = E\{x_i(k)x_i(k-\tau_1)\cdots x_i(k-\tau_M)\} \tag{31}$$

$$\cong \frac{1}{D}\sum_{k=1}^{D-\tau_M}\{x_i(k)x_i(k+\tau_1)\cdots x_i(k+\tau_M)\} \tag{32}$$

for $i = 0, 1, 2, \cdots, B-1$, where $\tau_1 \le \tau_2 \le \cdots \le \tau_M$ are positive integer values. From experiments, we found that the unsigned autocorrelation functions are more useful for classification. Thus, we redefine

$$R_i(\tau_1, \tau_2, \cdots, \tau_M) = \frac{1}{D}\sum_{k=1}^{D-\tau_M}|x_i(k)x_i(k+\tau_1)\cdots x_i(k+\tau_M)|. \tag{33}$$

The unsigned autocorrelation functions will produce large values when their order is high. This is equivalent to putting more weights on those features. To avoid this problem, we use the following treatment:

$$\Gamma_i(\tau_1, \tau_2, \cdots, \tau_M) = \sqrt[M]{R_i(\tau_1, \tau_2, \cdots, \tau_M)}. \tag{34}$$

It is simple to show that

$$\Gamma_{ti}(\tau_1, \tau_2, \cdots, \tau_M) = \alpha \cdot \Gamma_i(\tau_1, \tau_2, \cdots, \tau_M) \tag{35}$$

where Γ_t denotes the correlation function of the transformed signal. Finally, to obtain scale invariance, we normalize Γ_i.

$$\gamma_i(\tau_1, \tau_2, \cdots, \tau_M) = \frac{\Gamma_i(\tau_1, \tau_2, \cdots, \tau_M)}{\sum_{j=1}^{B}\Gamma_j(\tau_1, \tau_2, \cdots, \tau_M)} \tag{36}$$

for $i = 0, 1, 2, \cdots, B-1$, and it is easy to see that

$$\gamma_{ti}(\tau_1, \tau_2, \cdots, \tau_M) = \gamma_i(\tau_1, \tau_2, \cdots, \tau_M). \quad (37)$$

γ_i's for different sets of $\{\tau_1, \cdots, \tau_M\}$'s are then taken as features, and they are invariant to gray-scale transformation.

E. Training the HMM

The HMM training algorithm associated with the state-optimized maximum-likelihood criterion is iterative, and is called the segmental K-means algorithm [31]; it is described below.

1) *Clustering*: The training feature vectors are clustered. If a feature vector is in the ith cluster, it is assigned to state i. Thus, the number of clusters is the number of states.
2) *Calculation of the Initial Probability*:

$$\hat{\pi}_i = \frac{\text{Number of occurrences of } \{o_1 \in \text{state } i\}}{\text{Number of training samples}} \quad (38)$$

where $1 \le i \le L$ and L is the number of states.
3) *Calculation of the Transition Probability*:

$$a_{ij} = \frac{\text{Number of transitions from state } i \text{ to state } j}{\text{Number of transitions from state } i} \quad (39)$$

where $1 \le i, j \le L$.
4) *Calculation of the Symbol Probability Density Function for Each State*: We assume that the symbol density of each state is Gaussian.

$$b_j(o_t) = \frac{1}{(2\pi)^K |V_j|^{1/2}} \exp\left[-\frac{1}{2}(o_t - \mu_t)V_j^{-1}(o_t - \mu_t)^t\right] \quad (40)$$

where $1 \le j \le L$, o_t is the feature vector and K is its dimension. The mean and covariance matrix of each state can be calculated by

$$\hat{\mu}_i = \frac{1}{N_i}\sum_{o_t \in i} o_t, \qquad 1 \le i \le L \quad (41)$$

$$\hat{V}_i = \frac{1}{N_i}\sum_{o_t \in i}(o_t - \hat{\mu}_i)(o_t - \hat{\mu}_i)^T, \quad 1 \le i \le L \quad (42)$$

where N_i is the number of features assigned to state i. After the three parameter sets are obtained, the model $\hat{\lambda}$ is built; that is, $\hat{\Pi} = \{\hat{\pi}_i\}, \hat{A} = \{\hat{a}_{ij}\}, \hat{B} = \{\hat{b}_j(o_t)\}$ and $\hat{\lambda} = \{\hat{\Pi}, \hat{A}, \hat{B}\}$.
5) *Search for the Optimal State Sequence for Each Training Sample*: For a sequence of feature vectors, find the optimal state sequence $Q^* = \{q_1^*, q_2^*, \ldots, q_T^*\}$ for the given model parameters $\hat{\lambda}$ using the Viterbi algorithm, and assign feature vectors to corresponding states, $o_t \in$ *state* i if $q_t^* = i$, for $1 \le t \le T$ and $1 \le i \le L$.

TABLE I
CLASSIFICATION RESULTS FOR EIGHT-BAND DECOMPOSITION: 16 IMAGES WERE USED FOR TRAINING AND 16 IMAGES FOR TESTING OF EACH TEXTURE; EACH ENTRY SHOWS CORRECT CLASSIFICATION RATE AND ASSOCIATED MISCLASSIFIED SAMPLES

State No.	Feature Set A	Feature set B	Feature set C	Feature set D	Feature set E
3	55.86(113)	73.44(68)	86.33(35)	20.31(204)	10.16(230)
4	67.58(83)	82.42(45)	86.72(34)	19.14(207)	12.50(224)
5	72.27(71)	83.98(41)	88.67(29)	19.53(206)	10.55(229)
6	75.78(62)	85.16(38)	91.80(21)	20.31(204)	11.72(226)
7	74.61(65)	87.89(31)	91.02(23)	20.31(204)	9.38(232)
8	75.78(62)	89.84(26)	89.45(27)	–	–
9	76.56(60)	87.50(32)	89.06(28)	–	–
10	78.91(54)	–	–	–	–

6) *Optimization*: If any state assignment of a feature in step 5 is different from the previous one, update the state assignment. Steps 5 and 6 are conducted for all feature vectors.
7) *Iteration*: Go back to step 2 and use the new state assignment to estimate model parameters. Repeat this procedure until the state assignment does not change.

The clustering scheme used in step 1 belongs to the well-known "split-and-merge" algorithm, the "split" phase of which operates as follows:

1) calculate the center (mean) of the all training features, then split it in two;
2) use the two new centers and a Euclidean minimum-distance classifier to divide training features into two clusters;
3) recalculate the centers of the two new clusters, then split each in two and divide the features into four clusters;
4) repeat the procedure until $2^l (2^l > L)$ clusters are divided, which concludes "split."

"Merge" operates as follows:

1) count the number of sample in each cluster;
2) if the number of sample in the smallest cluster is less than a preset number $F_{\min}$, calculate its Euclidean distance from the others and merge it with the closest one;
3) repeat 1 and 2 until the number of samples in every cluster is greater than $F_{\min}$;
4) find the closest pair of clusters and merge them;
5) repeat 4 until the number of clusters is L (the number of state in the HMM).

F. HMM Testing

Assume there are J kinds of models denoted by $\lambda_1, \lambda_2, \cdots, \lambda_J$. Given an unknown feature-vector sequence $O = \{o_1, o_2, \cdots, o_T\}$, we first calculate the maximum state-optimized likelihood function P_i^* against all the models using the Viterbi algorithm; that is, calculate $P(O, I^*|\lambda_k)$ for $k = 1, 2, \cdots, J$, where I^* denotes the optimal state sequence, then classify by finding the maximum value of $P(O, I^*|\lambda_k)$

$$i^* = \arg \max_{k=1,2,\ldots,J} P(O, I^*|\lambda_k). \quad (43)$$

The unknown feature-vector sequence O is then classified as class i^*.

V. Simulations

We used a simple eight-tap lowpass filter $h(n)$ to form the QMF bank; the coefficients are shown below [30].

$$h(0) = h(7) = 0.009\,387\,159$$
$$h(1) = h(6) = -0.070\,651\,83$$
$$h(2) = h(5) = 0.069\,428\,270$$
$$h(3) = h(4) = 0.489\,980\,800.$$

We selected normalized autocorrelation functions up to the fourth order as features. To test the effectiveness of high-order statistics, we formed the five feature sets listed below.

A: $\gamma(0)$.
B: $\gamma(0), \gamma(1)$.
C: $\gamma(0), \gamma(1), \gamma(0,1), \gamma(1,1)$.
D: $\gamma(0), \gamma(1), \gamma(0,0,1), \gamma(0,1,1), \gamma(1,1,1)$.
E: $\gamma(0), \gamma(1), \gamma(0,1), \gamma(1,1), \gamma(0,0,1), \gamma(0,1,1), \gamma(1,1,1)$.

Feature set A is just the signal power that corresponds to a first-order statistic. Feature set B includes a first-order and a second-order statistic, and is commonly used in conventional stochastic signal analysis. Feature set C includes a first-order, a second-order, and a third-order statistic. Feature set D replaces the third-order statistic in feature set C with a fourth-order statistic. Finally, feature set E includes all the features in feature sets A, B, C, and D. The simplest feature set has only one feature, while the most complicated one has seven.

For our experiments, we selected 16 textures from Brodatz's texture album [32] shown in Fig. 14. Texture images were scanned to have size 512 × 512. The image size used in classification was 128 × 128. For training, images were extracted from the 512 × 512 originals with centers evenly covering the entire image area. In the first experiment, we took 16 training images and 16 test images for each class of texture. Centers of test images were randomly chosen from the 512 × 512 images and rotated to random angles. The number of sampling data is 12 000. For a 128 × 128 image, there are 16 384 pixels; thus, the density of sampling was smaller than (but close to) the original. We used an eight-band decomposition, giving us 1500 data points in each subband. The minimum number of samples in a cluster was set at $F_{\min} = 10$.

Table I shows experiment results. The "–" signs in the table mean that the number of samples in a cluster was too small and the clustering algorithm failed. From the table, we find that feature set C performed best, yielding the highest correct classification rate, 91.80%. The performance of feature sets D and E was extremely poor, probably because they contained fourth-order statistics and needed more training data than feature sets A, B, and C. This problem can be remedied by increasing the number of bands or by increasing the number of training samples. More bands will allow more symbol observations and state transitions, which may, in turn, facilitate estimates of symbol distributions and state-transition matrices. It will also decrease each band's bandwidth and make correlation functions easier to be estimated. Table II lists the results of 16-band decomposition. Here, the performance of feature sets D and E showed great improvement, becoming comparable to feature sets B and C. However, note that the performance of feature sets A, B, and C remained approximately the same (except for cases with small state numbers). This indicates that feature sets A, B, and C achieved performance saturation. For these feature sets, the training sample size was sufficiently large. Worth noting is that increasing the number of bands decreases the number of samples in each band; this may increase estimate variances, and classification results may be adversely affected. This explains the performance degradation of feature sets A, B, and C for small-state numbers. The correct classification rates for feature sets D and E were still not high enough, meaning that more training samples were necessary.

TABLE II
Classification Results for 16-band Decomposition: 16 Images Are Used for Training and 16 Images for Testing of Each Texture; Each Entry Shows Correct Classification Rate and Associated Misclassified Samples

State No.	Feature Set A	Feature Set B	Feature Set C	Feature Set D	Feature Set E
3	42.58(147)	56.25(112)	63.67(93)	70.70(75)	75.39(63)
4	55.86(113)	71.09(74)	83.20(43)	80.08(51)	78.52(55)
5	72.66(70)	78.52(55)	81.64(47)	84.77(39)	83.59(42)
6	80.08(51)	82.81(44)	85.16(38)	86.33(35)	82.81(44)
7	74.22(66)	84.38(40)	86.33(35)	85.55(37)	83.20(43)
8	78.91(54)	86.33(35)	88.67(29)	87.50(32)	84.38(40)
9	82.42(45)	85.94(36)	88.29(30)	84.77(39)	85.16(38)
10	80.58(50)	87.11(33)	90.63(24)	85.94(36)	81.25(48)

We then increased the training (test) samples up to 36 for each texture, leaving the band number at 8. Table III shows the classification results. The highest correct classification rate was then 95.14%. Comparing Tables I and III, we see that feature sets A, B, and C have the same performance for 16 and 36 training samples; thus, increasing the number of training samples did not help. This agrees with the argument we made above; performance of feature sets A, B, and C had reached saturation. However, feature sets D and E improved significantly. Another interesting discovery is that for any state number, the performance ranking, from the best to the worst, is E, D, C, B, A. The best classification rate for features with conventional second-order statistics (feature set B) was 88.39%. From this, we realize that the high-order statistics do have the capability to distinguish between similar textures. Sixteen-band decomposition was also tried. Results were similar to those from the eight-band case. Thus, we can say that 36 training samples with eight-band decomposition is adequate for feature sets D and E. Feature set E with nine states had the best performance. Note that the performance does not vary much from five states to ten states. We also performed some experiments to verify invariance to gray-scale transformation. Two cases were tried: $\alpha = 0.5$, $\beta = 20$ and $\alpha = 1.2$, $\beta = 0$. We found that the classification results were hardly affected. The last experiment we did was to compare the performance of our approach with the conventional minimum-distance classifier. We let the experimental settings be identical with the previous one and tested only the performance of feature set E. The classification accuracy was about 76%, far less than the results from the HMM. We thus confirmed that the HMM does exploit more information from feature sequences.

TABLE III
CLASSIFICATION RESULTS FOR EIGHT-BAND DECOMPOSITION: 36 IMAGES ARE USED FOR TRAINING AND 36 IMAGES FOR TESTING OF EACH TEXTURE; EACH ENTRY SHOWS CORRECT CLASSIFICATION RATE AND ASSOCIATED MISCLASSIFIED SAMPLES

State No.	Feature Set A	Feature Set B	Feature Set C	Feature Set D	Feature Set E
3	56.08(253)	72.40(159)	82.81(99)	86.28(79)	89.41(61)
4	63.89(208)	81.60(106)	89.76(59)	92.19(45)	92.52(43)
5	70.14(172)	80.38(113)	87.67(71)	91.32(50)	94.44(32)
6	70.49(170)	83.33(96)	88.54(66)	91.49(49)	93.75(36)
7	75.17(143)	84.72(88)	90.80(53)	92.53(43)	94.79(30)
8	76.91(133)	88.39(67)	91.16(51)	91.84(47)	94.79(30)
9	78.82(122)	86.98(75)	90.63(54)	92.36(44)	95.14(28)
10	76.91(133)	88.02(69)	90.28(56)	92.01(46)	94.97(29)

VI. CONCLUSION

We have presented an algorithm for texture classification that is rotation and gray-scale transform-invariant. While the rotation invariance is achieved through spiral sampling and an HMM, gray-scale transform invariance is achieved by mean removal and a normalization technique. The subband decomposition in this scheme plays an important role by providing an effective means to explore dependencies among subband features through the use of the HMM. The features we used are high-order autocorrelation functions. It was shown that the features do capture the texture characteristics. The highest classification accuracy for sixteen kinds of texture was 95.14%.

Our research was preliminary, and classification results can be further improved. We summarize some possible approaches in the following:

1) The observation density function used in the HMM is the simplest one, the Gaussian density. This assumption may not be true. A more realistic one may be the Gaussian-mixture model.
2) The 1-D sampled signal can be nonstationary. Thus, a nonstationary HMM may model the signal more faithfully.
3) In subband decomposition, each band was set to have equal bandwidth. However, as we know, most signal power is usually concentrated in low-frequency bands. Therefore, subbands with unequal bandwidths may produce better results.
4) In our experiments, coordinates of spiral resampling points were quantized to the closest lattice points. We may use some interpolation technique that can determine sampled values more precisely. This may improve the result when the sampling density is high.

REFERENCES

[1] R. W. Conners and C. A. Harlow, "A theoretical comparison of texture algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 204–222, May 1980.

[2] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 269–285, Apr. 1976.

[3] R. M. Haralick, "Statistical and structural approaches to texture," in *Proc. IEEE*, vol. 67, pp. 786–804, May 1979.

[4] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[5] B. Julesz, "Visual pattern discrimination," *IRE Trans. Inform. Theory*, vol. 8, pp. 84–97, Feb. 1962.

[6] P. C. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 64–69, Jan. 1983.

[7] 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.

[8] L. S. Davis, S. A. Johns, and J. K. Aggarwal, "Texture analysis using generalized co-occurrence matrices," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 251–259, July 1979.

[9] P. De Souza, "Texture recognition via autoregression," *Pattern Recog.*, vol. 15, pp. 471–475, 1982.

[10] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 1, pp. 25–39, Jan. 1983.

[11] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 4, pp. 959–963, Aug. 1986.

[12] 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.

[13] A. L. Vickers and J. W. Modestino, "A maximum likelihood approach to texture classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, no. 1, pp. 61–68, Jan. 1982.

[14] R. L. Kashyap and A. Khotanzed, "A model-based method for rotation invariant texture classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 472–481, July 1986.

[15] F. S. Cohen, Z. Fan, and M. A. Patel, "Classification of rotation and scaled textured images using Gaussian Markov random field models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 2, pp. 192–202, Feb. 1991.

[16] K. L. Laws, "Rapid texture identification," in *Proc. SPIE*, vol. 238, pp. 376–380, 1980.

[17] M. Unser, "Local linear transforms for texture measurements," *Signal Processing*, vol. 11, pp. 61–79, 1986.

[18] M. Unser, "Sum and difference histograms for texture classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 118–125, Jan. 1986.

[19] P. Cohen, C. T. Ledinh, and V. Lacasse, "Classification of natural textures by means of two-dimensional orthogonal masks," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 37, pp. 125–128, Jan. 1989.

[20] J. Y. Hsiao and A. A. Sawchuk, "Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 1279–1292, Dec. 1989.

[21] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 717–728, July 1989.

[22] A. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55–73, 1990.

[23] D. Dunn, W. E. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 130–149, Feb. 1994.

[24] T. Chang and C. C. Jay Kuo, "Texture analysis and classification with tree-structured Wavelet transform," *IEEE Trans. Image Processing*, vol. 2, no. 10, pp. 429–441, Oct. 1993.

[25] A. Kundu and J.-L. Chen, "Texture classification using QMF bank-based subband decomposition," *Comput. Vision Graphics Image Processing*, vol. 54, no. 5, pp. 369–384, Sept. 1992.

[26] J.-L. Chen and A. Kundu, "Rotational and gray-scale transform invariant texture identification using wavelet decomposition and hidden Markov model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 208–214, Feb. 1994.

[27] P. P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 4, pp. 4–20, July 1987.

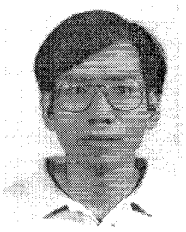
[28] C. R. Wylie and L. C. Barrett, *Advanced Engineering Mathematics*. New York: McGraw-Hill, pp. 984–989, 1982.

[29] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE Acoust., Speech, Signal Processing Mag.*, pp. 4–16, June 1986.

[30] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

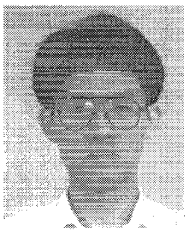
[31] B.-H. Juang and L. R. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov model," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 9, pp. 1639–1641, Sept. 1990.

[32] P. Brodatz, *Textures—A Photographic Album for Artists and Designers*. New York: Dover, 1965.



Wen-Rong Wu (S'87–M'89) was born in Taiwan, ROC, in 1958. He received the B.S. degree in mechanical engineering from Tatung Institute of Technology, Taiwan, in 1980, the M.S. degrees in mechanical and electrical engineering, and the Ph.D. degree in electrical engineering from the State University of New York at Buffalo in 1985, 1986, and 1989, respectively.

He has been a faculty member in the Department of Communication Engineering, National Chiao Tung University, Taiwan. His research interests include statistical signal processing and digital communications.



Shieh-Chung Wei was born in Taiwan, ROC, in 1968. He received the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1993.

Currently, he is a software engineer in Southern Information System Inc. His research interests include signal and image processing and telecommunications.