

Gradient texture unit coding for texture analysis

Chein-I Chang, FELLOW SPIE

Yuan Chen

University of Maryland

Department of Computer Science and

Electrical Engineering

Remote Sensing Signal and Image

Processing Laboratory

Baltimore County, Baltimore, Maryland

21250

Abstract. Texture is one of many important features to capture in image characteristics. A recent texture unit-based texture spectrum approach, referred to as texture unit coding (TUC) developed by Wang and He has shown promise in texture classification. We present a new texture feature extraction coding, called gradient texture unit coding (GTUC) that is based on Wang and He's texture unit to capture gradient changes in a texture unit. Since the GTUC also generates an 8-D ternary texture feature vector in the same way that the TUC does, a GTUC-generated feature vector can be further represented by a number in the same range generated by the TUC. As a result, the GTUC-generated numbers also form a texture spectrum similar to that formed by the TUC-generated numbers. By normalizing a texture spectrum as a probability distribution, this work further develops an information divergence (ID)-based discrimination criterion to measure the discrepancy between two texture spectra, a concept yet to be explored in texture analysis. To compare the GTUC to the TUC in texture classification, several criteria used in hyperspectral image analysis are also introduced for performance analysis. © 2004 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.1768183]

Subject terms: gradient texture unit coding; gradient texture unit number; information divergence; relative discriminatory probability; relative discriminatory power; texture unit; texture unit coding; texture unit numbers; texture spectrum.

Paper TPR-031 received Nov. 13, 2003; revised manuscript received Jan. 8, 2004; accepted for publication Mar. 15, 2004.

1 Introduction

Texture is one of the fundamental features used to describe image characteristics. One commonly used approach is the gray-level co-occurrence matrix that provides gray-level transition information between two gray levels.¹ Using a rather different approach, Wang and He² considered a 3×3 window as a texture unit (TU) along with its eight-neighbor connectivity³ to generate an 8-D texture feature vector for each pixel that represents the gray-level changes between the pixel at the center of the texture unit, referred to as a seed pixel, and its eight neighboring pixels specified by eight orientations 0, 45, 90, 135, 180, 225, 270, and 315 deg, shown in Fig. 1. As a result, each pixel can produce an 8-D texture feature vector that describes gray-level changes in a texture unit along with these eight orientations. Such an 8-D texture feature vector can be further converted to a texture unit number (TUN) in a ternary representation that specifies a particular texture pattern. Because each pixel in an image generates its own TUN via a TU, these TUNs can be used to form a texture spectrum in the same way that gray-level values of image pixels form an image histogram. The only difference is that the x axis of the texture spectrum is specified by TUNs instead of the gray levels in the image histogram. By virtue of the texture spectrum, Wang and He have investigated various applications.⁴⁻⁶

We explore the concept of Wang and He's texture unit and further develop a new texture feature extraction coding method, called gradient texture unit coding (GTUC) that encodes gradient changes in gray levels between the seed pixel in a texture unit and its two neighboring pixels in a TU, along with two different orientations. In other words,

the GTUC extends the Wang and He method, referred to as texture unit coding (TUC) in this work, in the sense that the GTUC captures the gray-level changes among three pixels rather than two pixels considered in the TUC. More specifically, unlike the TUC that encodes a change in gray level along a particular orientation specified by the seed pixel and one of its eight neighboring pixels in a texture unit, the GTUC dictates texture patterns that describe gradient changes in two orientations specified by the seed pixel and two of its eight neighboring pixels in the texture unit. Consequently, the GTUC can be interpreted as a gradient method of Wang and He's TUC, since it calculates the gray-level changes in two TUNs generated by Wang and He's TUC.

One of the unique features resulting from the TUC is that an 8-D ternary texture feature vector generated by the TUC can be represented by a numerical number, the texture unit number (TUN), ranging from 0 to $3^8 - 1 = 6560$. In analogy with gray levels that create an image histogram for an image, these TUC-generated TUNs can also be used to form a texture spectrum for a texture image. Similarly, the proposed GTUC also produces an 8-D ternary gradient texture feature vector that generates a numerical value in the same range $\{0, 1, \dots, 6560\}$, referred to as the gradient texture unit number (GTUN) in the same fashion that the TUN is generated by the TUC. As a result, a texture spectrum can also be produced from such GTUNs for a texture image. More interestingly, the relationship between the TUC and the GTUC seems to suggest that what the texture spectrum of TUNs is to the texture spectrum of the GTUNs is similar

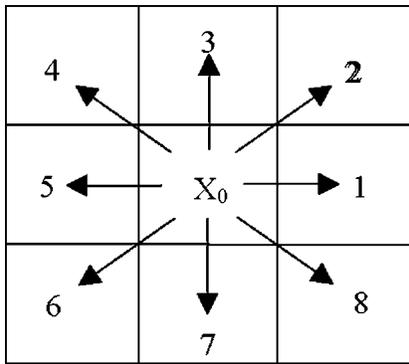


Fig. 1 Eight orientations.

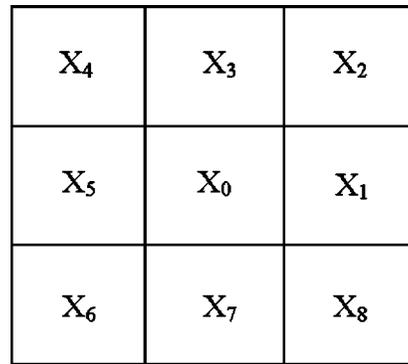


Fig. 2 A texture unit.

to what the gray-level image histogram is to spatial gray-level co-occurrence matrix (SGLCM).

In this work, the two texture feature extraction coding methods, TUC and GTUC, are investigated and compared for performance evaluation. Since it is generally difficult to compare two texture spectra by visual inspection, a texture spectrum is further normalized to unity to form a probability distribution so that the information divergence (ID),^{7,8} also known as Kullback-Leibler information distance, can be used to measure the discrepancy between two texture spectra. To compare the effectiveness of the TUC and the GTUC, several criteria previously developed in Refs. 8 and 9, along with the commonly used minimum distance in Ref. 5, are used to perform comparative analysis.

The remainder of this work is organized as follows. Section 2 describes the concept of texture spectrum (TS) and the texture unit coding (TUC) developed by Wang and He. Section 3 develops a new texture feature extraction coding, called gradient texture unit coding (GTUC). Section 4 introduces the information divergence and several criteria that can be used to measure the effectiveness of the TUC and the GTUC. Section 5 conducts experiments for performance comparison. As a concluding section, Sec. 6 is included to summarize the results and contributions accomplished in this work.

2 Texture Unit Coding

The spatial gray-level co-occurrence matrix (SGLCM) has been widely used in texture classification due to its ability in capturing transitions between all pairs of two gray levels (not necessarily distinct).¹ More specifically, it is a matrix \mathbf{W} whose elements are specified by frequencies of transitions from one gray to another in a certain way. Many SGLCM-based methods have been developed and reported in the literature. Recently, an alternative approach has been developed by Wang and He.^{2,4,5} It considers a 3×3 window as a texture unit and uses gray-level changes between the center pixel in the texture unit and one of its eight neighboring pixels as texture features, from which a texture spectrum could be generated for texture classification. The concept of using the texture spectrum has shown promise in various applications and can be described as follows.

2.1 Texture Unit

A texture unit (TU), described in Fig. 2, has the pixel X_0 as the center pixel, which is the pixel currently being pro-

cessed, referred to as the seed pixel, and eight surrounding pixels labeled by $X_1, X_2, X_3, X_4, X_5, X_6, X_7,$ and X_8 considered to be neighboring pixels of X_0 . According to Ref. 3, the four pixels $X_1, X_3, X_5,$ and X_7 are the first-order neighboring pixels of V_0 , which form the first-order four-neighbor connectivity, while $X_2, X_4, X_6,$ and X_8 are the second-order neighboring pixels of X_0 to constitute the second-order four-neighbor connectivity. If we assume that $V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7,$ and V_8 are the gray levels corresponding to $X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7,$ and X_8 respectively, Wang and He defined a texture feature number (TFN) E_i associated with a neighboring pixel X_i as follows,

$$E_i = \begin{cases} 0; & \text{if } V_i < V_0 - \Delta \\ 1; & \text{if } |V_i - V_0| \leq \Delta \quad \text{for } i = 1, 2, \dots, 8, \\ 2; & \text{if } V_i > V_0 + \Delta \end{cases} \quad (1)$$

where Δ is a gray-level tolerance to be determined.

2.2 Texture Spectrum

Since there are three values that an E_i in Eq. (1) can take, there are $3^8 = 6561$ combinations to cover all the possible values of $(E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8)$, each of which can be specified by a number $N(E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8)$, referred to as a texture unit number (TUN) of X_0 , using the following ternary representation.

$$N(E_1, E_2, \dots, E_8) = \sum_{i=1}^8 E_i \times 3^{i-1} = E_1 \times 3^0 + E_2 \times 3^1 + \dots + E_8 \times 3^7. \quad (2)$$

By virtue of a TU, each image pixel can generate an 8-D texture feature vector $(E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8)$ from which a TUN can be produced using Eq. (2). By treating such a TUN as a gray level, we can generate a texture spectrum for an image in a manner similar to a gray-level image histogram generated for an image. Since each of eight orientations in Fig. 1 can be used to start off as the first neighboring pixel X_1 , eight different texture spectra

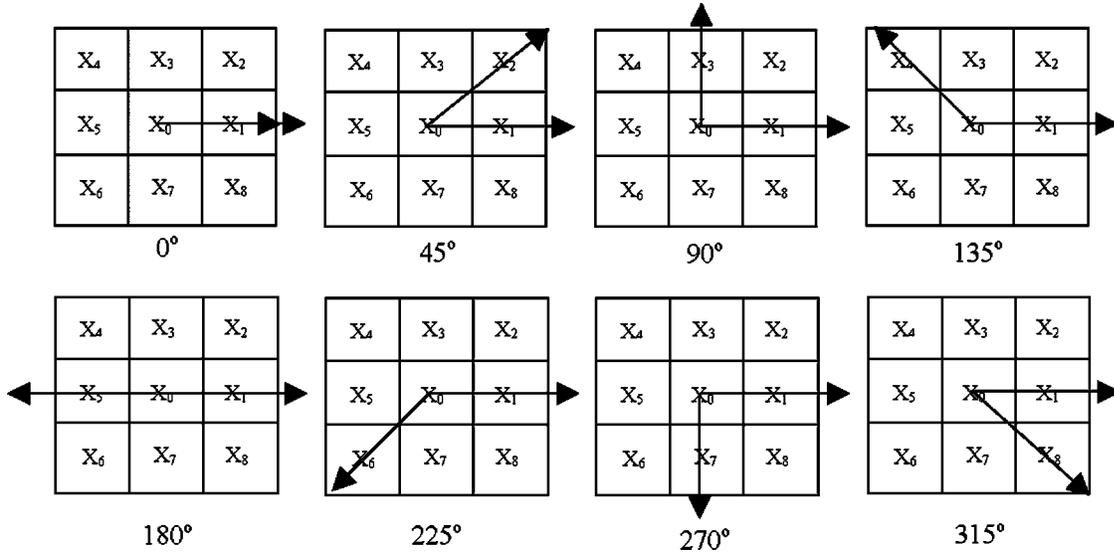


Fig. 3 Representation of eight different degrees, 0 deg ($X_1 - X_0 - X_1$), 45 deg ($X_1 - X_0 - X_2$), 90 deg ($X_1 - X_0 - X_3$), 135 deg ($X_1 - X_0 - X_4$), 180 deg ($X_1 - X_0 - X_5$), 225 deg ($X_1 - X_0 - X_6$), 270 deg ($X_1 - X_0 - X_7$), and 315 deg ($X_1 - X_0 - X_8$).

can be generated for each image. Using Eq. (2), we can further convert the TU-based texture spectrum to a probability distribution, p_{TUN} by

$$p_{\text{TUN}}(i) = \frac{N_{\text{TUN}}(i)}{N} \quad \text{for } i=0,1,\dots,6560, \quad (3)$$

where $N_{\text{TUN}}(i)$ is the number of pixels whose TUN is $N(E_1, E_2, \dots, E_8) = i$, defined by Eq. (2), and N is the total number of pixels in the image.

3 Gradient Texture Unit Coding

The texture spectrum produced by Eq. (2) is based on the correlation between the seed pixel X_0 and one of its eight neighboring pixels, $X_1, X_2, X_3, X_4, X_5, X_6, X_7$, and X_8 in Fig. 2. It does not take into account the spatial correlation of X_0 with two pixels in its eight-neighbor connectivity. In this section, we develop a new concept of gradient texture unit matrix, which captures the gray-level gradient changes between V_0 and V_i and between V_0 and V_j . More specifically, for the seed pixel X_0 in a TU, we define a gradient texture feature number (GTFN) with respect to any pair of its two neighboring pixels X_i and X_j as follows.

$$\nabla V_{ij} = \begin{cases} 0; & \text{if } |V_i - V_0| \leq \Delta \text{ and } |V_j - V_0| \leq \Delta \\ 1; & \text{if } |V_i - V_0| \leq \Delta \text{ and } |V_j - V_0| > \Delta \\ & \text{or } |V_i - V_0| > \Delta \text{ and } |V_j - V_0| \leq \Delta \\ 2; & \text{if } |V_i - V_0| > \Delta \text{ and } |V_j - V_0| > \Delta \end{cases} \quad \text{for } i, j=1,2,\dots,8. \quad (4)$$

In analogy with the 8-D texture feature vector defined by $(E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8)$ for the seed pixel X_0 , we can generate an 8×8 gradient texture feature matrix (GTFM) for X_0 by,

$$\nabla \mathbf{M}_{\text{TU}} = [\nabla V_{ij}]_{8 \times 8}, \quad (5)$$

whose (i, j) entry ∇V_{ij} is given by Eq. (4). It should be noted that the GTFM is symmetric and the diagonal elements of $\nabla \mathbf{M}_{\text{TU}}$ can only take values of either 0 or 2. So, we can convert the GTFM to a 36-D gradient texture feature vector where each of the 36 dimensions in the feature vector specifies a gradient change between a particular pair of two pixels X_i and X_j in Fig. 2. As a result, there are a total of $2^8 + 3^{28}$ gradient texture feature vectors. That is, for each fixed X_i in Fig. 2, there are eight pixels in the eight-neighbor connectivity of X_0 , including itself (i.e., X_i), can be chosen to be X_j where each of these pixels represents one particular orientation specified in Fig. 1. Figure 3 shows an example where X_i is chosen to be X_1 and the X_j runs from X_1, X_2 through X_8 , which represents eight different degrees, 0 deg ($X_1 - X_0 - X_1$), 45 deg ($X_1 - X_0 - X_2$), 90 deg ($X_1 - X_0 - X_3$), 135 deg ($X_1 - X_0 - X_4$), 180 deg ($X_1 - X_0 - X_5$), 225 deg ($X_1 - X_0 - X_6$), 270 deg ($X_1 - X_0 - X_7$), and 315 deg ($X_1 - X_0 - X_8$). Similarly, we can also choose X_i to be X_2 and the X_j to run from X_2, X_3, \dots, X_8 through X_1 , as shown in Fig. 4 to also represent eight degrees, 0 deg ($X_2 - X_0 - X_2$), 45 deg ($X_2 - X_0 - X_3$), 90 deg ($X_2 - X_0 - X_4$), 135 deg ($X_2 - X_0 - X_5$), 180 deg ($X_2 - X_0 - X_6$), 225 deg ($X_2 - X_0 - X_7$), 270 deg ($X_2 - X_0 - X_8$), and 315 deg ($X_2 - X_0 - X_1$). Comparing Fig. 4 to Fig. 3, we note that the only thing that matters is the degree formed by X_i and X_j regardless of what is chosen for X_i . In other words, the degree of the angle formed by X_i and X_j only depends on the difference of subscript, i.e., $|i - j|$. In this case, the eight orientations in Fig. 1 also correspond to the eight degrees specified by 0 deg ($X_i - X_0 - X_{i+1}$), 45 deg ($X_i - X_0 - X_{i+2}$), 90 deg ($X_i - X_0 - X_{i+3}$), 135 deg ($X_i - X_0 - X_{i+4}$), 180 deg ($X_i - X_0$

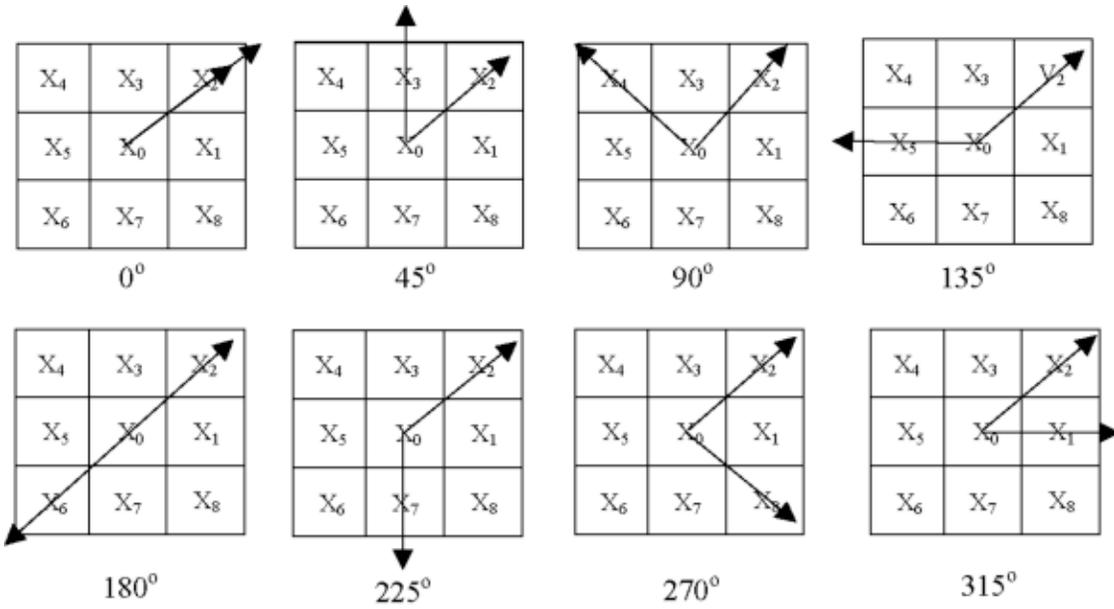


Fig. 4 Representation of eight different degrees, 0 deg ($X_2 - X_0 - X_2$), 45 deg ($X_2 - X_0 - X_3$), 90 deg ($X_2 - X_0 - X_4$), 135 deg ($X_2 - X_0 - X_5$), 180 deg ($X_2 - X_0 - X_6$), 225 deg ($X_2 - X_0 - X_7$), 270 deg ($X_2 - X_0 - X_8$), and 315 deg ($X_2 - X_0 - X_1$).

$-X_{i+5}$), 225 deg ($X_i - X_0 - X_{i+6}$), 270 deg ($X_i - X_0 - X_{i+7}$), and 315 deg ($X_i - X_0 - X_{i+8}$) for $i=2,3,\dots,8$. Therefore, we can always fix X_i at X_1 and let the X_j run from X_1, X_2 through X_8 and generate a gradient texture unit number (GTUN) for each image pixel, denoted by $N(\nabla V_{1,1}, \nabla V_{1,2}, \dots, \nabla V_{1,8})$ in the same way that the TUN is generated by Eq. (2) using the following ternary representation,

$$\begin{aligned}
 &N(\nabla V_{1,1}, \nabla V_{1,2}, \dots, \nabla V_{1,8}) \\
 &= \sum_{j=1}^8 \nabla V_{1,j} \times 3^{i-1} \\
 &= \nabla V_{1,1} \times 3^0 + \nabla V_{1,2} \times 3^1 + \dots + \nabla V_{1,8} \times 3^7. \quad (6)
 \end{aligned}$$

Interestingly, there are also $3^8 = 6561$ combinations in Eq. (6) to generate GTUNs. Like Wang and He's TUNs, these GTUNs also form a gradient texture spectrum that can be used for texture analysis. In particular, we can further convert the GTU-based texture spectrum to a probability distribution, p_{GTUN} by

$$p_{GTUN}(i) = \frac{N_{GTUN}(i)}{N} \quad \text{for } i=0,1,\dots,6560, \quad (7)$$

where $N_{GTUN}(i)$ is the number of pixels whose GTUN is $N(\nabla V_{1,1}, \nabla V_{1,2}, \dots, \nabla V_{1,8}) = i$, defined by Eq. (6), and N is the total number of pixels in the image.

4 Texture Discrimination Measures

As described in Secs. 2 and 3, Wang and He's texture spectrum and our proposed gradient texture spectrum can be considered as a probability distribution defined on a probability space (Ω, Σ, P) with the sample space

$\Omega = \{0,1,\dots,6560\}$, the power set of Ω as the even space Σ , and the probability measure P specified by either p_{TUN} via Eq. (3) or p_{GTUN} via Eq. (7). This interpretation allows us to take advantage of the well-established information theory to analyze the texture spectrum. In this section, we present three texture spectrum discrimination measures that were previously derived from hyperspectral imagery^{8,9} and can be used to measure similarity between two texture images. Let m denote a probability measure specified by any texture feature extraction method such as p_{TUN} , p_{GTUN} . Assume that $\alpha_m = (\alpha_{m,0}, \alpha_{m,1}, \dots, \alpha_{m,6560})^T$ and $\beta_m = (\beta_{m,0}, \beta_{m,1}, \dots, \beta_{m,6560})^T$ are texture spectra or gradient texture spectra of two texture images denoted by A and B .

4.1 Information Divergence

For a given image A , the self-information provided by α_m for TUN or GTUN = j is defined by

$$I_j(\alpha_m) = -\log \alpha_{m,j}. \quad (8)$$

Similarly, for another image B , the self-information provided by β_m for TUN or GTUN = j is defined by

$$I_j(\beta_m) = -\log \beta_{m,j}. \quad (9)$$

Using Eqs. (8) and (9), we can further define $D_j(\alpha_m \| \beta_m)$, the discrepancy in the self-information of j in β_m relative to the self-information of j in α_m by

$$D_j(\alpha_m \| \beta_m) = I_j(\beta_m) - I_j(\alpha_m) = \log(\alpha_{m,j} / \beta_{m,j}). \quad (10)$$

Averaging $D_j(\alpha_m \| \beta_m)$ in Eq. (10) over all j 's with respect to α_m results in

$$\begin{aligned}
 D(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m) &= \sum_{j=0}^{6560} D_j(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m) \alpha_{m,j} \\
 &= \sum_{j=0}^{6560} \alpha_{m,j} \log(\alpha_{m,j} / \beta_{m,j}), \quad (11)
 \end{aligned}$$

where $D(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m)$ is the average discrepancy in the self-information of $\boldsymbol{\beta}_m$ relative to the self-information of $\boldsymbol{\alpha}_m$. In the context of information theory, $D(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m)$ in Eq. (11) is called the relative entropy of $\boldsymbol{\beta}_m$ with respect to $\boldsymbol{\alpha}_m$, which is also known as the Kullback-Leibler information measure, directed divergence, or cross entropy.⁷ Similarly, we can also define the average discrepancy in the self-information of $\boldsymbol{\alpha}_m$ relative to the self-information of $\boldsymbol{\beta}_m$ by

$$\begin{aligned}
 D(\boldsymbol{\beta}_m \parallel \boldsymbol{\alpha}_m) &= \sum_{j=1}^L D_j(\boldsymbol{\beta}_m \parallel \boldsymbol{\alpha}_m) \beta_{m,j} \\
 &= \sum_{j=0}^{6560} \beta_{m,j} \log(\beta_{m,j} / \alpha_{m,j}). \quad (12)
 \end{aligned}$$

Summing Eqs. (11) and (12) yields information divergence (ID) defined by

$$\text{ID}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = D(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m) + D(\boldsymbol{\beta}_m \parallel \boldsymbol{\alpha}_m), \quad (13)$$

which can be used to measure the similarity of texture patterns between two texture images A and B . It should be noted that while $\text{ID}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$ is symmetric, $D(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m)$ is not. This is because $\text{ID}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = \text{ID}(\boldsymbol{\beta}_m, \boldsymbol{\alpha}_m)$, and $D(\boldsymbol{\alpha}_m \parallel \boldsymbol{\beta}_m) \neq D(\boldsymbol{\beta}_m \parallel \boldsymbol{\alpha}_m)$.

4.2 Relative Discriminatory Probability

Let $\{\mathbf{s}_k\}_{k=1}^K$ be K texture images in the set Δ , which can be considered as a database, and \mathbf{t} be any specific target texture image to be identified using Δ . We define the discriminatory probabilities of all \mathbf{s}_k 's in Δ relative to \mathbf{t} as follows.

$$p_{\mathbf{t},\Delta}^m(k) = m(\mathbf{t}, \mathbf{s}_k) / \sum_{i=1}^K m(\mathbf{t}, \mathbf{s}_i) \quad \text{for } k=1, 2, \dots, K, \quad (14)$$

where $\sum_{i=1}^K m(\mathbf{t}, \mathbf{s}_i)$ is a normalization constant determined by \mathbf{t} and Δ . The resulting probability vector $\mathbf{p}_{\mathbf{t},\Delta}^m = [p_{\mathbf{t},\Delta}^m(1), p_{\mathbf{t},\Delta}^m(2), \dots, p_{\mathbf{t},\Delta}^m(K)]^T$ is called the relative discriminatory probability (RDPB) of Δ with respect to \mathbf{t} or the spectral discriminatory probability vector of Δ relative to \mathbf{t} . Then, using Eq. (14) we can identify \mathbf{t} via Δ by selecting the one with the smallest relative spectral discriminability probability. If there is a tie, either one can be used to identify \mathbf{t} .

4.3 Relative Discriminatory Entropy

Since $\mathbf{p}_{\mathbf{t},\Delta}^m = [p_{\mathbf{t},\Delta}^m(1), p_{\mathbf{t},\Delta}^m(2), \dots, p_{\mathbf{t},\Delta}^m(K)]^T$ given by Eq. (14) is the relative discriminability probability vector of \mathbf{t} using a selective set of texture images $\Delta = \{\mathbf{s}_k\}_{k=1}^K$, we can further define the relative discriminatory entropy (RSDE) of the spectral signature \mathbf{t} with respect to the set Δ , denoted by $H_{\text{RSDE}}^m(\mathbf{t}, \Delta)$ by

$$H_{\text{RSDE}}^m(\mathbf{t}, \Delta) = - \sum_{k=1}^K p_{\mathbf{t},\Delta}^m(k) \log_2 p_{\mathbf{t},\Delta}^m(k). \quad (15)$$

Equation (15) provides an uncertainty measure of identifying \mathbf{t} resulting from using $\Delta = \{\mathbf{s}_k\}_{k=1}^K$. A higher $H_{\text{RSDE}}^m(\mathbf{t}, \Delta)$ may have less chance to identify \mathbf{t} .

4.4 Relative Discriminatory Power

Let \mathbf{d} be a texture image. Assume that \mathbf{s} and \mathbf{s}' are a pair of two texture images to be compared using the \mathbf{d} as a reference texture image. The RDPW of $m(\cdot)$, denoted by $\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$ is defined by

$$\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d}) = \max \left\{ \frac{m(\mathbf{s}, \mathbf{d})}{m(\mathbf{s}', \mathbf{d})}, \frac{m(\mathbf{s}', \mathbf{d})}{m(\mathbf{s}, \mathbf{d})} \right\}. \quad (16)$$

More precisely, $\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$ selects as the discriminatory power of $m(\cdot)$ the maximum of two ratios, the ratio of $m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$ to $m(\mathbf{s}', \mathbf{s}; \mathbf{d})$, and the ratio of $m(\mathbf{s}', \mathbf{s}; \mathbf{d})$ to $m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$. The $\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$ defined by Eq. (16) provides a quantitative index of discrimination capability of a specific texture feature coding method $m(\cdot)$ between two texture images \mathbf{s} and \mathbf{s}' relative to \mathbf{d} . Obviously, the higher the $\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$ is, the better discriminatory power $m(\cdot)$ is. In addition, $\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d})$ is symmetric and bounded below by one, i.e., $\text{RDPW}^m(\mathbf{s}, \mathbf{s}'; \mathbf{d}) \geq 1$ with equality if and only if $\mathbf{s} = \mathbf{s}'$.

5 Experiments

In this section, the same four texture images of size 640×640 pixels labeled in Fig. 5(a) that were used in Refs. 2, 4, 5, and 6 were also used for experiments to conduct a comparative study between Wang and He's TUC and our proposed GTUC for performance evaluation using the minimum distance (MD) and the information divergence (ID) as performance measures. The tolerance Δ was set to 3, which was an empirical choice. It should also be noted that both MD and ID were performed on the probability distributions produced by Eqs. (3) and (7). The images are selected from Brodatz's natural texture images,¹⁰ image A (beach sand), image B (water), image C (pressed cork), and image D (fur hide of an unborn calf). To make our analysis simple and more effective, the upper left corner of size 64×64 pixels cropped from each of the four images in Fig. 5 was used for our experiments. They are enlarged and shown right beneath each of the four images labeled by Fig. 5(b). Figures 6 and 7 show examples of eight texture spectra of the TUNs and GTUNs generated for the four texture images in Fig. 5, respectively, with eight different starting positions from X_1, X_2, \dots, X_8 . As we can see from Figs. 6 and 7, the TUC-generated and the GTUC-generated texture spectra were quite different, thus provide different levels of texture information for analysis. In particular, compared to TUC-generated texture spectra for four images, which all looked very similar from visual inspection, the GTUC-generated texture spectrum of image A was quite different from those of images B, C, and D. So, it could be singled out immediately without difficulty. If we further examined more closely the GTUC-generated texture spectrum of im-

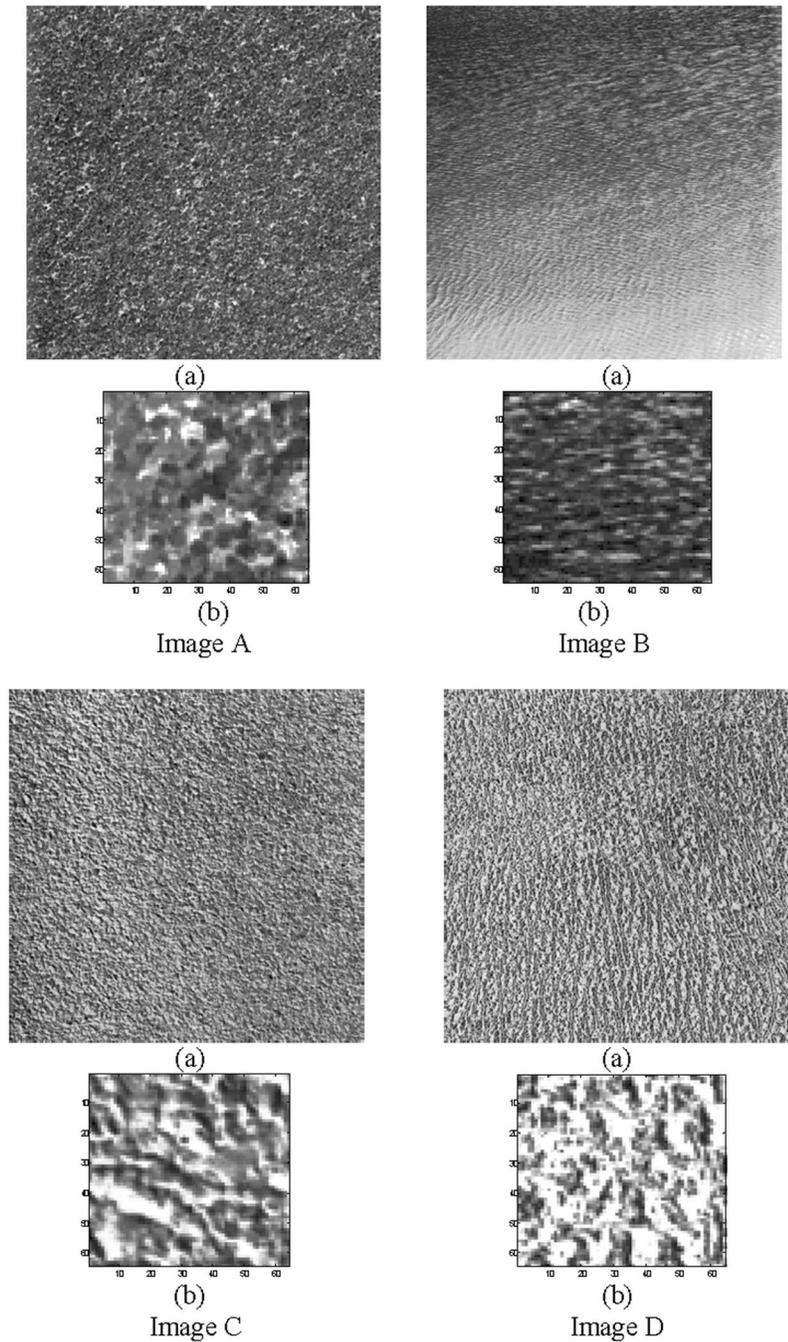


Fig. 5 Four texture images: (a) original image; and (b) a subimage of 64×64 cropped from the upper left corner of each original image.

age B, it was also distinct from images C and D. As for images C and D, their corresponding GTUC-generated texture spectra were very similar and difficult to discern one from another. Based on this observation, we could conclude that the four texture images could be grouped into three different classes, image A, image B, and a class comprised of images C and D. However, such categorization cannot be made easily by visually examining the TUC-generated texture spectra in Fig. 6.

To avoid subjective visual inspection, Tables 1 and 2 tabulate the results of applying the TUC and GTUC to all the four texture images in Fig. 5(b) using the MD and ID as

distance measures, respectively. To simplify analysis, the notations $TUC/MD(X, Y)$, $TUC/ID(X, Y)$, $GTUC/MD(X, Y)$, and $GTUC/ID(X, Y)$ are introduced to represent the distance between images X and Y measured by the TUC using MD and ID, the GTUC using MD and ID, respectively. Using these four definitions, Table 1 shows that $TUC/MD(A, D) = 0.0023$, $TUC/MD(B, A) = 0.0026$, and $TUC/MD(C, D) = TUC/MD(D, C) = 0.0028$, all of which were the smallest values yielded by TUC/MD for each of the four texture images in Fig. 5(b). It implied that image A was more similar to image D than to images B

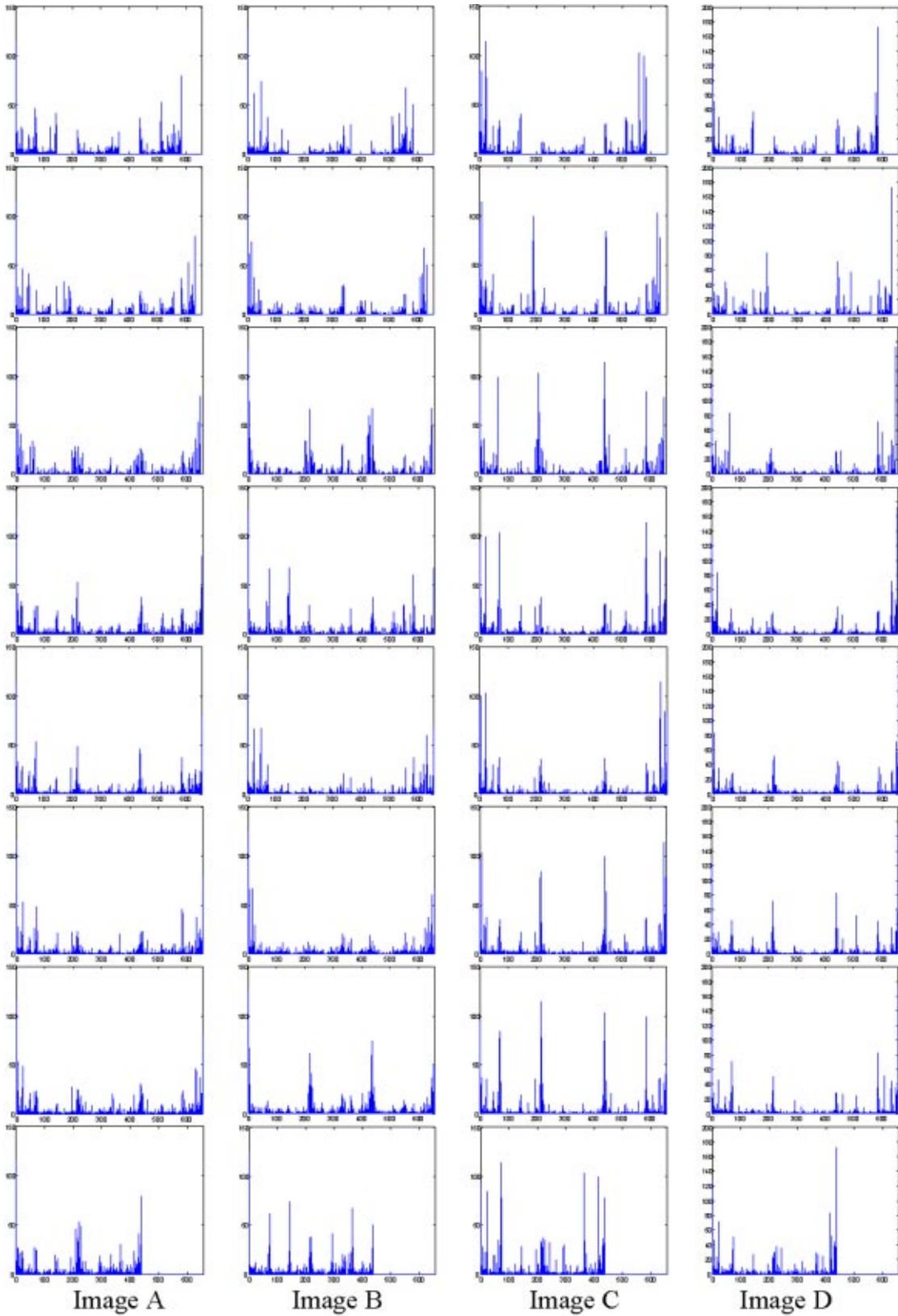


Fig. 6 Eight texture spectra of the TUNs generated for the four texture images in Fig. 5, respectively, with eight different starting positions from $X_1, X_2, \dots, X_7, X_8$.

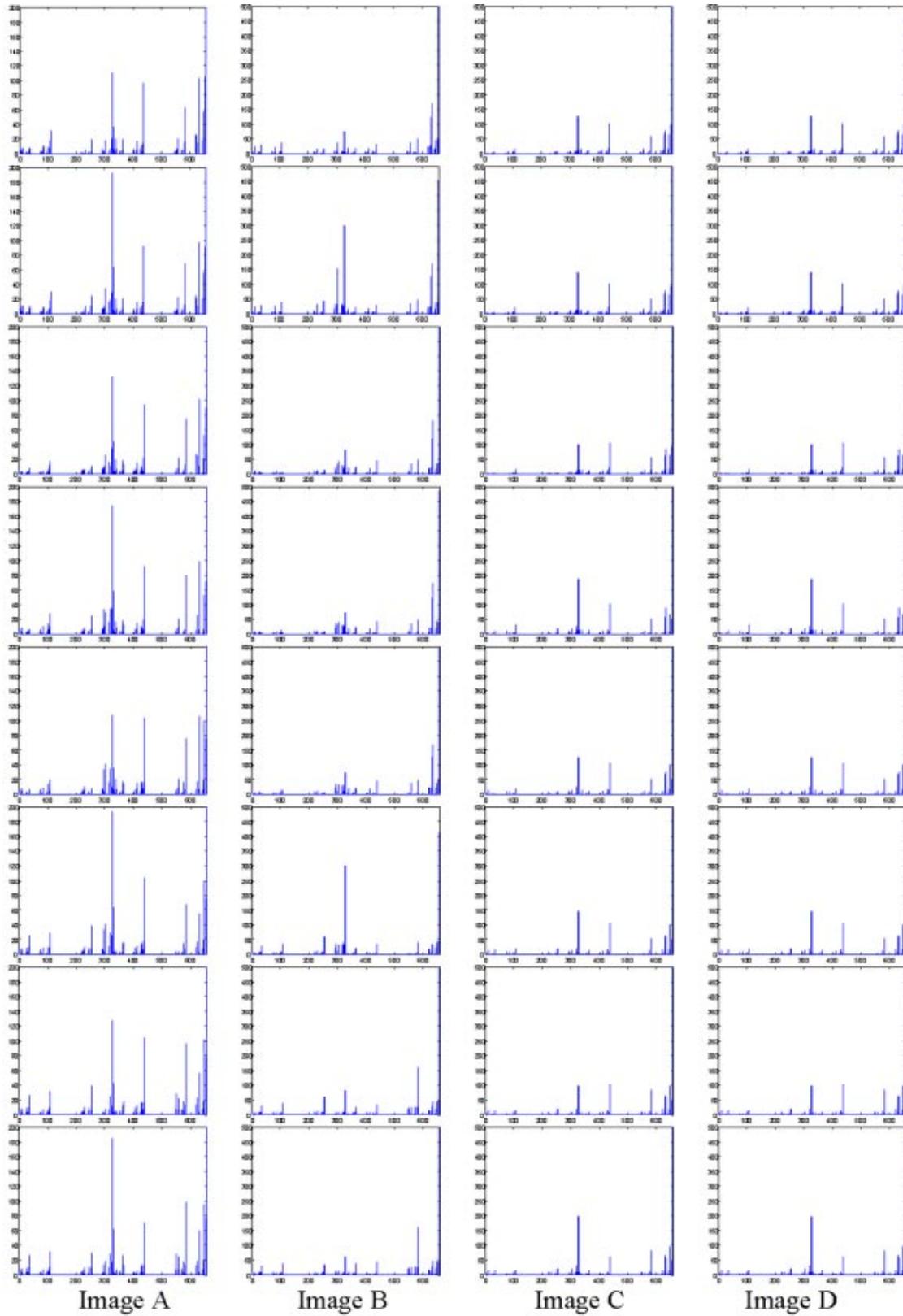


Fig. 7 Eight texture spectra of the GTUNs generated for the four texture images in Fig. 5, respectively, with eight different starting positions from $X_1, X_2, \dots, X_7, X_8$.

Table 1 Results of TUC and GTUC for the four texture images in Fig. 5(b) measured by MD.

GTUC	TUC			
	A	B	C	D
A	N/A	0.0026	0.0029	0.0023
B	0.0130	N/A	0.0059	0.0058
C	0.0153	0.0524	N/A	0.0028
D	0.0235	0.0663	0.0032	N/A

Table 2 Results of TUC and GTUC for the four texture images in Fig. 5(b) measured by ID.

GTUC	TUC			
	A	B	C	D
A	N/A	1.1002	0.6868	0.7341
B	0.7316	N/A	1.7370	1.5775
C	0.3613	1.3013	N/A	0.6615
D	0.4989	1.6178	0.2018	N/A

and C, image B was more similar to image A than to images C and D, and images C and D were similar to each other compared to images A and B.

Now, if the GTUC was used and their results are tabulated in Table 1, where $GTUC/MD(A,B)=GTUC/MD(A,B)=0.0140$, $GTUC/MD(C,D)=GTUC/MD(D,C)=0.00025$ were the smallest values produced by the GTUC/MD for each of the four texture images. The GTUC results suggested that images A and B were in one group, while images C and D were in another group.

Next, if the ID was implemented in place of the MD in Table 1, the results of the TUC/ID and GTUC/ID are tabulated in Table 2, where for each of the four texture images, their smallest values in Table 2 produced by TUC/ID and GTUC/ID, were $TUC/ID(A,C)=0.6868$, $TUC/ID(B,A)=1.1000$, $TUC/ID(C,D)=TUC/ID(D,C)=0.6615$, and $GTUC/ID(A,C)=0.3529$, $GTUC/ID(B,A)=0.6833$, $GTUC/ID(C,D)=GTUC/ID(D,C)=0.0172$. Interestingly, in this case, the TUC/MD and GTUC/MD produced consistent results. Both suggested that image A was more similar to image C than to images B and D, image B was more similar to image A than to images C and D, and images C and D were similar to each other compared to images A and B. Furthermore, if we take a close look at the ID values produced by GTUC in Table 2, we could conclude immediately that images C and D were in the same texture image class with similar texture patterns, because the value of $GTUC/ID(C,D)=GTUC/ID(D,C)$, 0.0172 was very small, while images A and B could be considered as two separate texture image classes, since $GTUC/ID(A,C)=0.3529$, $GTUC/ID(B,A)=0.6833$ were almost 20 and 40 times of the value 0.0172. The results in Tables 1 and 2 demonstrate that the ID was more reliable than the MD in terms of texture discrimination, and the GTUC performed more effectively than did the TUC.

To further substantiate our conclusions drawn from Tables 1 and 2, the RDPB, entropy, and RDPW were used

to evaluate the performance of TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID for comparative analysis. Tables 3–6 tabulate their results with images A, B, C, and D used as a reference image, respectively, where the smallest RDPB values and the largest RDPW values are highlighted and shaded for each of four reference images. As noted, a smaller value of RDPB yielded by one image indicates less discrimination between the image and the reference image. On the contrary, a larger value of RDPW between two images represents a more discriminatory power to discriminate the two images with respect to the reference image. As shown in Tables 3–6, when the ID was used, the TUC and GTUC produced consistent results with the GTUC having better discriminatory powers (i.e., higher RDPW values) and less entropies. On the other hand, when MD was used, the GTUC generally performed better than the TUC with significant reduction of entropies and substantially better discriminatory powers (see RDPW values in Tables 3–6). It should be noted that with image D as the reference image, the TUC/MD produced the smallest RDPB for image A. This contradicted with the results produced by the GTUC/MD, TUC/ID, and GTUC/ID, all of which yielded the smallest RDPB values for image D. If the results in Tables 3–6 are further taken into account, the TUC/MD performed the worst among the TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID.

It is also interesting to revisit image A. According to Tables 1 and 2, both TUC and GTUC had difficulty with discriminating image A from images B, C, and D. This is because image A has quite different texture patterns from images B, C, and D, as shown in Fig. 7. So, in this case, image A can be discriminated easily and should be removed from comparison with other images. If it was used as a reference image, the RDPB and RSPW values among images B, C, and D were not consistent, as reflected in Table 3. The results in Table 3 show that $TUC/MD(A,D)=0.2949$, $GTUC/MD(A,B)=0.2510$, $TUC/ID(A,C)$

Table 3 RDPB, RDPW, and entropy values of TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID using image A as the reference image.

Ref. image A	RDPB			Entropy	RDPW		
	B	C	D		B, C	B, D	C, D
TUC/MD	0.3333	0.3718	0.2949	1.5785	1.1155	1.1302	1.2608
GTUC/MD	0.2510	0.2954	0.4537	0.3009	1.1769	1.8076	1.5359
TUC/ID	0.4364	0.2724	0.2912	1.5514	1.6021	1.4986	1.0690
GTUC/ID	0.4596	0.2270	0.3134	1.5257	2.0247	1.4665	1.3806

Table 4 RDPB, RDPW, and entropy values of TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID using image B as the reference image.

Ref. image B	RDPB			Entropy	RDPW		
	A	C	D		A, C	A, D	C, D
TUC/MD	0.1818	0.4126	0.4056	1.5022	2.2695	2.2310	1.0173
GTUC/MD	0.0987	0.3979	0.5034	0.5639	4.0314	5.1003	1.2651
TUC/ID	0.2492	0.3935	0.3573	1.5595	1.5791	1.4338	1.1013
GTUC/ID	0.2004	0.3565	0.4431	1.5155	1.7789	2.2111	1.2429

Table 5 RDPB, RDPW, and entropy values of TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID using image C as the reference image.

Ref. image C	RDPB			Entropy	RDPW		
	A	B	D		A, B	A, D	B, D
TUC/MD	0.2500	0.5086	0.2414	1.4911	2.0344	1.0356	2.1069
GTUC/MD	0.2158	0.7391	0.0451	0.3417	3.4249	4.7849	16.3880
TUC/ID	0.2226	0.5630	0.2144	1.4254	2.5292	1.0382	2.6259
GTUC/ID	0.1938	0.6980	0.1082	1.1680	3.6017	1.7911	6.4510

Table 6 RDPB, RDPW, and entropy values of TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID using image D as the reference image.

Ref. image D	RDPB			Entropy	RDPW		
	A	B	C		A, B	A, C	B, C
TUC/MD	0.2110	0.5321	0.2569	1.4617	2.5218	1.2175	2.0712
GTUC/MD	0.2527	0.7129	0.0344	0.4132	2.8211	7.3459	20.7238
TUC/ID	0.2470	0.5305	0.2225	1.4659	2.1478	1.1101	2.3843
GTUC/ID	0.2152	0.6978	0.0870	1.1457	3.2426	2.4736	8.0207

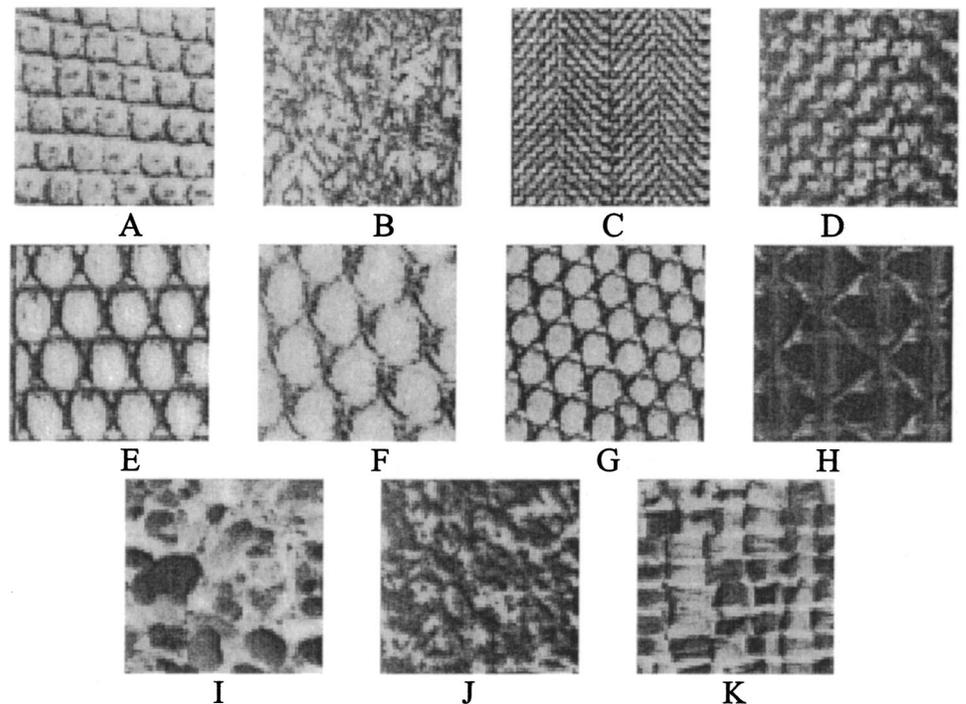


Fig. 8 Additional 11 texture images used for experiments.

Table 7 Results of TUC and GTUC for the 11 texture images measured by MD.

GTUC	TUC										
	A	B	C	D	E	F	G	H	I	J	K
A	N/A	0.0034	0.0102	0.0034	0.0029	0.0026	0.0033	0.0050	0.0025	0.0023	0.0035
B	0.0069	N/A	0.0053	0.0019	0.0026	0.0018	0.0026	0.0046	0.0030	0.0022	0.0045
C	0.0288	0.0139	N/A	0.0083	0.0120	0.0094	0.0113	0.0129	0.0105	0.0094	0.0135
D	0.0174	0.0045	0.0240	N/A	0.0025	0.0026	0.0024	0.0037	0.0018	0.0016	0.0021
E	0.0134	0.0104	0.0457	0.0067	N/A	0.0013	0.0014	0.0031	0.0024	0.0020	0.0026
F	0.0147	0.0173	0.0599	0.0143	0.0020	N/A	0.0019	0.0023	0.0022	0.0021	0.0032
G	0.0101	0.0034	0.0272	0.0020	0.0031	0.0079	N/A	0.0039	0.0024	0.0022	0.0025
H	0.0558	0.0524	0.1134	0.0357	0.0176	0.0141	0.0313	N/A	0.0027	0.0033	0.0039
I	0.0141	0.0098	0.0434	0.0054	0.0013	0.0031	0.0024	0.0185	N/A	0.0012	0.0016
J	0.0130	0.0048	0.0299	0.0018	0.0031	0.0078	0.0007	0.0284	0.0019	N/A	0.0024
K	0.0131	0.0066	0.0358	0.0030	0.0027	0.0065	0.0015	0.0250	0.0013	0.0014	N/A

Table 8 Results of TUC and GTUC for the 11 texture images measured by ID.

GTUC	TUC										
	A	B	C	D	E	F	G	H	I	J	K
A	N/A	0.6190	1.0211	0.5868	0.6124	0.5415	0.5972	1.0271	0.5740	0.5407	0.6728
B	0.1518	N/A	0.6988	0.4143	0.5038	0.4086	0.4847	0.9532	0.6184	0.4942	0.7481
C	0.2262	0.1420	N/A	0.9479	1.1632	0.9018	1.1686	1.0210	1.2155	1.0977	1.2725
D	0.2437	0.0951	0.2544	N/A	0.5085	0.5111	0.4868	0.6748	0.3780	0.3436	0.3485
E	0.2475	0.2529	0.3502	0.2260	N/A	0.3652	0.3922	0.7532	0.5780	0.4870	0.5982
F	0.2518	0.3091	0.4101	0.2903	0.1865	N/A	0.4255	0.6689	0.5447	0.5052	0.6618
G	0.1798	0.1329	0.3064	0.1461	0.1759	0.2251	N/A	0.7993	0.5065	0.4729	0.5275
H	0.8834	0.8116	0.8830	0.5934	0.4897	0.4673	0.6440	N/A	0.6417	0.6713	0.8145
I	0.2235	0.2087	0.3684	0.1801	0.2479	0.2048	0.1599	0.5417	N/A	0.3232	0.3596
J	0.2144	0.1230	0.2833	0.1154	0.2358	0.2531	0.1238	0.5225	0.1193	N/A	0.4471
K	0.2194	0.1617	0.3823	0.1319	0.2578	0.2988	0.1378	0.6847	0.1499	0.1504	N/A

Table 9 RDPB of TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID using image K as the reference image.

Ref. image K	RDPB										
	A	B	C	D	E	F	G	H	I	J	Entropy
TUC/MD	0.0879	0.1131	0.3392	0.0528	0.0653	0.0804	0.0628	0.0980	0.0402	0.0603	2.9764
GTUC/MD	0.1352	0.0681	0.3695	0.0310	0.0279	0.0671	0.0155	0.2580	0.0134	0.0144	2.5149
TUC/ID	0.1043	0.1160	0.1973	0.0540	0.0927	0.1026	0.0818	0.1263	0.0557	0.0693	3.2165
GTUC/ID	0.0852	0.0628	0.1485	0.0512	0.1001	0.1161	0.0535	0.2659	0.0582	0.0584	3.0869

Table 10 Comparison of computational complexity between TUC and GTUC.

CPU 600 MHz	Computing time (seconds)										
	A	B	C	D	E	F	G	H	I	J	
TUC	1.6230	1.6630	1.6030	1.5520	1.5620	1.9220	1.6830	1.6520	1.5920	1.7730	
GTUC	26.729	20.049	17.294	17.235	17.635	17.725	18.016	17.545	19.377	17.074	

=0.2724, GTUC/ID(A,C)=0.2270 yielded the smallest RDPB values. This fact implied that image A could be mistaken for any of the three images, images B, C, and D. The reason for this is because image A is so distinct from the other three images that using it as a reference image does not do any good in discrimination among images B, C, and D. However, if a complete spectrum of the RDPB values was taken into consideration, the three RDPB values of images B, C, and D for TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID were (0.3333, 0.3718, 0.2949), (0.2510, 0.2954, 0.4537), (0.4364, 0.2724, 0.2912), and (0.4596, 0.2270, 0.3134), respectively, where the TUC/MD generated three very close RDPB values compared to the GTUC/ID that produced the most distinct RDPB values. From this aspect, the GTUC/ID performed the best, while the TUC/MD was the worst.

To have a fair comparison between Wang and He's TUC and our GTUC, the four texture images used for our previous experiments were the same ones used in Wang and He's work in Refs. 2, 4, 5, and 6. However, similar experiments can be also carried out for any texture image. In doing so, 11 texture images labeled A through K in Fig. 8 were randomly selected from Brodatz's natural texture images,¹⁰ where the image K was chosen as a reference image to calculate the RDPB for comparison. Tables 7 and 8 tabulate the results of TUC and GTUC for these 11 texture images measured by the MD and the ID, respectively. Since the image K was used as a reference image, we looked at the last rows and last columns of Tables 7 and 8 to compare the performance of the TUC and the GTUC using the MD and the ID. Interestingly, the experimental results provided evidence that the two distance measures MD and ID performed differently, even though they were implemented in conjunction with the same texture feature coding method. According to Tables 7 and 8, the TUC and GTUC identified the image I (highlighted values) as the one closest to image K using the MD, compared to the image D (highlighted values) identified to be closest to the image K by the TUC and the GTUC using the ID. To further evaluate their relative discriminatory probabilities, Table 9 tabulates the RDPB of the TUC/MD, GTUC/MD, TUC/ID, and GTUC/ID using image K as a reference image, where their respective entropies were also calculated. As we can see from this table, the ID always yielded the smaller RDPB and entropy than the MD did when they were implemented with the same texture coding method. Since the tables for the RDPW are too large, their results are not included in this work. Nevertheless, those who are interested in the RDPW can compute their values directly from Tables 7 and 8.

To conclude this section, Table 10 provides a comparative analysis on the computational complexity of the TUC and GTUC calculated for the 11 texture images in Fig. 8, where a Pentium III PC with 600 MHz was used for performance evaluation. As documented in Table 10, the GTUC required about 17 to 26 sec to generate each image compared to about 1.5 to 2.0 sec required by the TUC, which is about 15 times in seconds in computational complexity. This additional computing time is attributed by the computation of the gradient texture feature matrix for the GTUC.

6 Conclusions

The concept of the texture spectrum recently proposed by Wang and He has found applications in pattern classification. It can be considered as a transform coding that makes use of a 3×3 texture unit (TU) to convert texture patterns of an image to various texture numbers that can be used to form a texture spectrum in the same way that an image gray-level histogram is created in histogram equalization for image enhancement. This work presents a new approach, called gradient texture unit coding (GTUC) that is derived from Wang and He's texture unit approach. Unlike Wang and He's texture unit coding (TUC) that encodes the gray-level changes between the seed (center) pixel and one of its adjacent pixels in a texture unit, the GTUC generates gradient texture numbers (GTUNs) that dictate gray-level changes between the seed pixel and its two adjacent pixels in a texture unit. As a result, the GTUC can be interpreted as a gradient of Wang and He's texture unit numbers (TUNs) because it captures the gray-level changes between two TUNs. Interestingly, both the Wang and He's TUC and the GTUC results in the same range of their corresponding texture unit numbers from 0 to 6560. Therefore, their spectra can be normalized to probability vectors and further measured by information divergence in conjunction with other criteria recently developed in hyperspectral image analysis.^{8,9} The main contributions of this work focus on the introduction of a new texture coding method, GTUC, and exploration of the TUC and GTUC in texture analysis from an information theory's point of view. The results presented are only based on texture analysis. They can also be implemented in conjunction with classification techniques to perform texture classification and pattern recognition in various areas such as medical imaging, remote sensing, etc. One such example using the concept of Wang and He's texture spectrum and TUC is mass detection in mammograms, where texture features provide crucial elements in classification.^{11,12} A new application of our GTUC to this area is currently under investigation.

References

1. R. M. Harlaick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst. Man Cybern.* **3**(6), 610–621 (1973).
2. L. Wang and D. C. He, "A new statistical approach to texture analysis," *Photogramm. Eng. Remote Sens.* **56**, 61–65 (1990).
3. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA (2002).
4. D. C. He and L. Wang, "Texture unit, texture spectrum and texture analysis," *IEEE Trans. Geosci. Remote Sens.* **28**(4), 509–512 (1990).
5. L. Wang and D. C. He, "Texture classification using texture spectrum," *Pattern Recogn.* **23**(8), 905–910 (1990).
6. D. C. He and L. Wang, "Texture features based on texture spectrum," *Pattern Recogn.* **24**(5), 391–399 (1991).
7. T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York (1991).
8. C.-I Chang, "An information theoretic-based approach to spectral variability, similarity and discriminability for hyperspectral image analysis," *IEEE Trans. Inf. Theory* **46**(5), 1927–1932 (2000).
9. C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Chap. 2, Kluwer Academic/Plenum Publishers, New York (2003).
10. P. Brodatz, *Texture—A Photographic Album for Artists and Designers*, Reinhold, New York (1968).
11. S. M. Guo, P. S. Liao, Y. C. Liao, S. C. Yang, P. C. Chung, and C.-I Chang, "Mass detection in mammography using texture analysis," *Chinese J. Radiology* **28**(3), 149–157 (2003).
12. Y. Liao, "Mass detection in mammography using texture analysis," MS Thesis, Dept. of Information Engineering, National Cheng Kung Univ., Tainan, Taiwan (2002).



Chein-I Chang received his BS degree from Soochow University, Taipei, Taiwan, MS degree from the Institute of Mathematics at National Tsing Hua University, Hsinchu, Taiwan, and MA degree from the State University of New York at Stony Brook, all in mathematics. He also received his MS and MSEE degrees from the University of Illinois at Urbana-Champaign and PhD degree in electrical engineering from the University of Maryland, College Park. He has

been with the University of Maryland, Baltimore County (UMBC) since 1987, as a visiting assistant professor from January 1987 to August 1987, assistant professor from 1987 to 1993, associate professor from 1993 to 2001, and professor in the Department of Computer Science and Electrical Engineering since 2001. He has three patents and several pending patents on image processing techniques for hyperspectral imaging and detection of microcalcifications. His research interests include automatic target recognition,

multispectral/hyperspectral image processing, medical imaging, and visual information systems. He authored a book, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, published by Kluwer Academic/Plenum Publishers. He is a senior member of IEEE and a member of Phi Kappa Phi and Eta Kappa Nu.



Yuan Chen received the BS degree in electrical engineering from Southeast University in 2001, and MS degree in electrical engineering from the University of Oklahoma in 2003. She is currently pursuing her PhD degree in the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County. Her research interests include image processing, medical imaging, and pattern recognition.