Bayesian Texture Classification Based on Contourlet Transform and BYY Harmony Learning of Poisson Mixtures

Yongsheng Dong and Jinwen Ma

***Abstract*—As a newly developed 2-D extension of the wavelet transform using multiscale and directional filter banks, the contourlet transform can effectively capture the intrinsic geometric structures and smooth contours of a texture image that are the dominant features for texture classification. In this paper, we propose a novel Bayesian texture classifier based on the adaptive model-selection learning of Poisson mixtures on the contourlet features of texture images. The adaptive model-selection learning of Poisson mixtures is carried out by the recently established adaptive gradient Bayesian Ying-Yang harmony learning algorithm for Poisson mixtures. It is demonstrated by the experiments that our proposed Bayesian classifier significantly improves the texture classification accuracy in comparison with several current state-of-the-art texture classification approaches.**

***Index Terms*—Bayesian Ying-Yang (BYY) harmony learning system, contourlet transform, model selection, Poisson mixtures, texture classification.**

I. Introduction

TEXTURE classification is an important but difficult task in image processing and pattern recognition. From different aspects, there have been established a variety of texture classification methods during the last three decades [1]–[17]. In fact, they can be broadly divided into two categories, namely, spatial-domain methods and transformed-domain methods, which are also referred to as filter-based methods or signal-processing methods [7].

Spatial-domain methods can be further divided into three subcategories, namely, structural methods [2], [3], statistical methods [1], and model-based methods [4], [5]. In fact, spatial structural methods are based on regular or semiregular placements of textural primitives [2], [3]. In the case of observable or visual textures, it is usually rather difficult to extract the primitives and their placements. Thus, these approaches are apt only for highly regular deterministic textures. Most of spatial statistical and model-based approaches for texture classification, such as gray-level co-occurrence matrices [1] and Markov random field models [4], [5], are restricted to the analysis of spatial interactions over relatively small neighborhoods. Therefore, their performances are only good for the class of so-called microtextures.

To alleviate these essential problems of spatial-domain methods, transformed-domain methods have been proposed from a different point of view [6]–[16]. The texture classification system using a transformed-domain method involves two steps, i.e., feature extraction, where a group of texture features are extracted from the domain into which the image is transformed, and classification, where a class label is assigned to an input texture image according to the extracted texture features. The commonly used transforms include the Radon [11], Gabor [13], ranklet [10], wavelet [6]–[9], [14], [15], and ridgelet [12] transforms. Among these transforms, the wavelet transform has gained popular applications in the fields of image processing and classification [6]–[9], [14], [15], [18]–[20]. The main reason is that the wavelet transform enables the decomposition of the image into different frequency subbands, being similar in the way the human visual system operates. However, 2-D wavelets lack directionality and are only good at catching point discontinuities but do not capture the geometrical smoothness of the contours.

As a newly developed 2-D extension of the wavelet transform using multiscale and directional filter banks (DFBs), the contourlet transform can effectively capture the intrinsic geometrical structures, which are a key to visual information processing, because the contourlet expansion can achieve the optimal approximation rate for piecewise smooth functions with C^2 contours in some sense [23]. Recently, the contourlet transform has been successfully used in content-based texture retrieval [26], palmprint classification, and handwritten numeral recognition [27]. In this paper, we utilize the contourlet transform to extract the features of a texture image for the purpose of texture classification.

On the other hand, it is very important to design an efficient texture classifier based on the contourlet features in the transformed domain. From the point of view of statistics, we can consider that each of the contourlet features extracted from a random image of one texture class is approximately subject to a Poisson mixture distribution. Then, the Bayesian classifier can be built for texture classification if those Poisson mixtures for all the texture classes can be clarified via some learning algorithm.

The expectation maximization (EM) algorithm is the common method to estimate the parameters of a finite mixture

Manuscript received July 13, 2010; revised January 25, 2011 and August 11, 2011; accepted September 04, 2011. Date of publication September 22, 2011; date of current version February 17, 2012. This work was supported by the Natural Science Foundation of China under Grant 60771061. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mary Comer.

The authors are with the Department of Information Science, School of Mathematical Sciences, and the Laboratory of Mathematics and Applied Mathematics, Peking University, Beijing, 100871, China (e-mail: jwma@math.pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2168231

with a given sample data set [31], and it has been utilized for texture classification and segmentation [6], [21], [22], [26]. However, the EM algorithm is constructed under a framework of maximum likelihood and thus unable to make a model selection for Poisson mixtures [32], [37], i.e., to select an appropriate number of Poissons for a sample data set. Therefore, the learning algorithm that we need should own the ability to adaptively determine the number of Poissons in the sample data during the parameter learning, i.e., to make a model selection adaptively, because we cannot know the appropriate number of Poissons in the sample data of one contourlet feature for each texture class in advance. Fortunately, the adaptive gradient Bayesian Ying-Yang (BYY) harmony learning algorithm based on the BYY harmony learning system and theory [40], [41] has been recently established in [37] to solve these problems. Although this BYY harmony learning algorithm was already applied to texture classification directly on the data sets of texture images and led to a better classification accuracy than the bit-plane probability (BP) signature approach [9], it needs a relatively large number of samples to train the classifier, which is often practically unsatisfied. However, as we apply it on the contourlet features of the texture images, the texture classification accuracy can be significantly improved with the help of the good representation of the contourlet transform. Moreover, the training-data-demanding problem can be alleviated in a certain sense.

In this paper, we follow the transform-based texture classification paradigm and propose a novel Bayesian texture classifier via the BYY harmony learning of Poisson mixtures on the contourlet features of texture images. The comparative classification experiments show that our proposed Bayesian classifier outperforms several current state-of-the-art texture classification approaches.

The contourlet transform is introduced in Section II. In Section III, we describe the Poisson mixture model and the adaptive gradient BYY harmony learning algorithm for Poisson mixtures. Section IV presents our Bayesian texture classifier based on the contourlet transform and the adaptive model-selection learning of Poisson mixtures. The experimental results and comparisons are contained in Section V. Finally, we briefly conclude in Section VI.

II. Contourlet Transform

The contourlet transform has been recently developed in [23] in order to get rid of the limitations of wavelets. They utilized a double filter bank structure in which, at first, the Laplacian pyramid (LP) [24] is used to capture the point discontinuities and, then, a DFB [25] is used to link point discontinuities into the linear structure. Thus, the overall result of such a transform is based on an image expansion with basis elements such as contour segments, and thus, it is named the contourlet transform. More recent developments and applications on the contourlet transform can be found in [26]–[29].

Due to its cascade structure accomplished by combining the LP with a DFB at each scale, multiscale and directional decomposition stages in the contourlet transform are independent of each other. Therefore, one can decompose each scale into any arbitrary power-of-two's number of directions, and different scales can be decomposed into different numbers of directions. For simplicity, we impose that, in the pyramid DFB, the number of DFB decomposition levels is three at each scale of the pyramid to capture the directional information efficiently, that is, the number of directions at each scale is 8.

In the contourlet domain, the hidden Markov tree (HMT) [26] is suggested to model the contourlet coefficients in the fields of denoising and texture retrieval. Although the HMT model is quite efficient for some practical applications, its computational cost is very large. Here, we model the contourlet coefficients in each subband using the simpler probabilistic models, i.e., Poisson mixtures, rather than the relatively complicated HMT model.

III. Poisson Mixtures and its BYY Harmony Learning

As well known, Poisson distribution is a typical discrete probabilistic model for count events or data, such as the number of telephone calls you receive in an hour, the number of dandelions per square meter on the college playing field, and the number of cars per mile broken down on the hard shoulder of the motor way. Mathematically, a univariate Poisson (probability) distribution is defined as $p(x|\theta) = (\theta^x/x!)e^{-\theta}$, $x = 0, 1, 2, \ldots$, where $\theta > 0$ is the parameter and $x!$ is the factorial of x.

In many practical applications, the observed data can be considered being generated from a number of components that are linearly mixed in certain proportions. That is, the observed data are subject to a finite mixture distribution. If all the components are Poisson distributions, the finite mixture is called a Poisson mixture [30], [37], which can be given as follows:

$$q(x|\Theta_k) = \sum_{j=1}^{k} \alpha_j q(x|\theta_j) \tag{1}$$

where $\Theta_k = \{\alpha_j, \theta_j\}_{j=1}^k$ and $q(x|\theta_j) = (\theta_j^x/x!)e^{-\theta_j}$ are Poisson distributions, values of $\alpha_j \geq 0$ are the mixing proportions of these Poisson components with the constraint that $\sum_{j=1}^k \alpha_j = 1$, and k is just the number of Poisson components.

To make the model selection adaptively during the parameter learning, the adaptive gradient BYY learning algorithm [37] was established for Poisson mixtures. For clarity, we still give a brief derivation of the adaptive gradient BYY learning algorithm for Poisson mixtures and leave the details in [37].

For the Poisson mixture modeling, a bidirectional architecture of the BYY learning system has been established such that its BYY harmony learning is equivalent to the parameter learning with adaptive model selection on Poisson mixtures [37]. Specifically, given a sample data set $D_x = \{x_t\}_{t=1}^N$ from the original Poisson mixture, the learning task on this bidirectional architecture is to maximize the following harmony function:

$$J(\Theta_k) = \frac{1}{N} \sum_{t=1}^{N} J_t(\Theta_k) \tag{2}$$

where

$$J_t(\Theta_k) = \sum_{j=1}^{k} \frac{U_j(x_t)}{\sum_{i=1}^k U_i(x_t)} \ln U_j(x_t) \tag{3}$$

where $U_i(x_t) = \alpha_j q(x_t|\theta_j)$, $j = 1, 2, \ldots, k$, and $q(x_t|\theta_j) = (\theta_j^{x_t}/x_t!)e^{-\theta_j}$.

For convenience of derivation, we utilize the following so-called softmax representations of the mixing proportions $\alpha_j : \alpha_j = e^{\beta_j}/\sum_{i=1}^{k} e^{\beta_i}$, $j = 1, 2, \ldots, k$, where $-\infty < \beta_1, \ldots, \beta_k < +\infty$. In such a way, α_j computed with any group β_j will certainly satisfy conditions $\alpha_j \geq 0$ and $\sum_{j=1}^{k} \alpha_j = 1$.

In this way, we can obtain the partial derivatives of $J_t(\Theta_k)$ at the sample point x_t with respect to β_j and θ_j as follows:

$$\frac{\partial J_t(\Theta_k)}{\partial \beta_j} = \frac{1}{q(x_t|\Theta_k)} \sum_{i=1}^{k} A(x_t, i)(\delta_{i,j} - \alpha_j) U_i(x_t) \quad (4)$$

$$\frac{\partial J_t(\Theta_k)}{\partial \theta_j} = \frac{1}{q(x_t|\Theta_k)} A(x_t, i)\alpha_j \frac{\partial q(x_t|\theta_j)}{\partial \theta_j} \quad (5)$$

where $A(x_t) = [1 - \sum_{l=1}^{k}(p(l|x_t) - \delta_{il}) \ln U_l(x_t)]$, $p(l|x_t) = (\alpha_l q(x_t|\theta_l)/q(x_t|\Theta_k))$, $\delta_{i,j}$ is the Kronecker function, and $(\partial q(x_t|\theta_j)/\partial \theta_j) = (\theta_j^{x_t - 1}/x_t!)e^{-\theta_j}(x_t - \theta_j)$.

According to (4) and (5), we can construct the following adaptive gradient BYY learning rule of β_j and θ_j:

$$\beta_j^{new} = \beta_j^{old} + \eta \frac{\partial J_t(\Theta_k)}{\partial \beta_j} \quad (6)$$

$$\theta_j^{new} = \theta_j^{old} + \eta \frac{\partial J_t(\Theta_k)}{\partial \theta_j} \quad (7)$$

where $\eta > 0$ is the learning rate that starts from a reasonable initial value and gradually decreases to zero in the Robbin–Monro stochastic approximation manner [36].

According to the BYY harmony learning principles [37], [40], [41], as the adaptive gradient BYY learning algorithm with k being larger than the number of actual Poissons has converged, i.e., $J(\Theta_k)$ arrives at a maximum, the model selection is adaptively made with the parameter learning or estimation by forcing the mixing proportions of those extra Poissons to attenuate to zero.

IV. Proposed Bayesian Texture Classifier

Upon previous preparations, we can now present the Bayesian texture classifier based on the BYY harmony learning of Poisson mixtures on the contourlet features of texture images in the following two subsections, respectively.

A. Feature Extraction Via the Contourlet Transform

As previously discussed, the contourlet transform is more efficient in capturing the contours of an image than the wavelet transform. In order to classify the textures more efficiently, we can decompose a texture image into some subbands by the contourlet transform and extract some features from each subband to represent it. Thus, these contourlet features can be used to represent the texture image more efficiently for texture classification. Moreover, each contourlet feature can be regarded as a random variable subject to a certain Poisson mixture distribution.

For a texture image, denoted by $a_o[\mathbf{n}] \in L_2(R^2)$, we can decompose it via the discrete contourlet transform into a set of coefficients $\{a_I[\mathbf{n}], c_{i,j}^{(l_i)}[\mathbf{n}]\}$, $i = 1, 2, \ldots, I$ and $j = 0, 1, \cdots, 2^{l_i} - 1$. Note that indexes i, j, and $\mathbf{n}$ specify the scale, the direction, and the location, respectively. The number of DFB decomposition levels varies with scale i, and thus, we denote it by l_i. I is the highest level of LP decomposition. Fig. 1 shows the contourlet transform on the "Tile.0006" texture image from the VisTex database [33]. For the visual clarity, only two-scale decompositions are shown. The image is decomposed into a low-pass subband and 16 bandpass directional subbands, where the highest level of the LP decomposition level is 2 and the number of DFB decomposition levels is 3 at each scale ($l_i = 3, i = 1, 2$), i.e., the number of directional subbands is 8 at each scale.

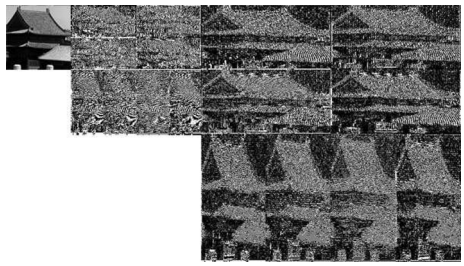


Fig. 1. Contourlet transform of the "$Tile.0006$" texture.

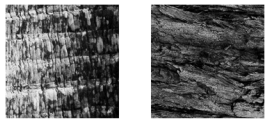
The purpose of the feature extraction is to obtain a set of texture measures that can be used to discriminate among different texture classes. As we all know, a widely used wavelet feature is the energy of each wavelet subband [6], [8]. Similarly, in the contourlet domain, we extract the norm-1 energy of each subband as a feature to represent the energy of the subband. However, from the statistical viewpoint, the degree of the dispersion of the coefficients is also a metric that can discriminate different texture classes. Hence, we extract the average absolute deviation from the absolute mean and the range in each subband as the features that measure the degree of dispersion of the coefficients. The former measures the degree of deviation from the energy, and the latter measures the degree of dispersion of the minus coefficients and the plus coefficients because the distribution of coefficients in one subband is not strictly symmetric generally. However, in order to construct the probabilistic model conveniently, for each of the aforementioned three features in one subband, we define a new equivalent feature variable with a smaller variance by multiplying a factor less than one to the primary feature. The details are as follows.

Here, we impose that the number of DFB decomposition levels at each scale is 3 ($l_i = 3, i = 1, 2, \ldots, I$), i.e., the number of directional subbands at each scale is 8. For clarity, we denote the jth directional subband at the ith scale by the $M_{i,j} \times N_{i,j}$ coefficient matrix $c_{i,j}[\mathbf{n}]$ ($c_{i,j}[\mathbf{n}]$ is the simplified form of $c_{i,j}^{(l_i)}[\mathbf{n}]$). Then, the features in $c_{i,j}[\mathbf{n}]$ are given by

$$f_{i,j}^1 = \frac{1}{i \cdot M_{i,j} N_{i,j}} \sum_{x=1}^{M_{i,j}} \sum_{y=1}^{N_{i,j}} |w_{i,j}(x, y)| \quad (8)$$

$$f_{i,j}^2 = \frac{1}{i \cdot M_{i,j} N_{i,j}} \sum_{x=1}^{M_{i,j}} \sum_{y=1}^{N_{i,j}} \left| |w_{i,j}(x, y)| - E_{i,j}^1 \right| \quad (9)$$

$$f_{i,j}^3 = \frac{1}{i^2} \left[\max_{x,y} w_{i,j}(x, y) - \min_{x,y} w_{i,j}(x, y) \right] \quad (10)$$



(a) (b)

Fig. 2. Two texture images. (a) Bark.0000. (b) Bark.0004.

where $E^1_{i,j} = i \cdot f^1_{i,j} = (1/M_{i,j}N_{i,j}) \sum_{x=1}^{M_{i,j}} \sum_{y=1}^{N_{i,j}} |w_{i,j}(x,y)|$ and $w_{i,j}(x,y) \in c_{i,j}[\mathbf{n}]$, where $x = 1, 2, \ldots, M_{i,j}$ and $y = 1, 2, \ldots, N_{i,j}$.

For the wavelet-based texture classification approaches, the coefficients in the low-frequency subband are generally ignored. However, those coefficients still contain some valuable information of the image. Thus, it is better to use them for texture classification. We denote the low-frequency subband by the $M_I \times N_I$ coefficient matrix $a_I[\mathbf{n}]$ and then obtain the features in $a_I[\mathbf{n}]$ by

$$g^1_I = \frac{1}{2^I \cdot M_I N_I} \sum_{x=1}^{M_I} \sum_{y=1}^{N_I} |w_I(x,y)| \tag{11}$$

$$g^2_I = \frac{1}{2^I \cdot M_I N_I} \sum_{x=1}^{M_I} \sum_{y=1}^{N_I} \left| |w_I(x,y)| - E^1_I \right| \tag{12}$$

$$g^3_I = \frac{1}{2^I} \left[\max_{x,y} w_I(x,y) - \min_{x,y} w_I(x,y) \right] \tag{13}$$

where $E^1_I = 2^I \cdot g^1_I = (1/M_I N_I) \sum_{x=1}^{M_I} \sum_{y=1}^{N_I} |w_I(x,y)|$ and $w_I(x,y) \in a_I[\mathbf{n}]$, where $x = 1, 2, \ldots, M_I$ and $y = 1, 2, \ldots, N_I$.

In this way, the feature vector of each texture patch from a texture image is composed of three low-frequency features and $24 * I$ directional features, i.e.,

$$\mathbf{F} = \left(g^1_I, g^2_I, g^3_I, \mathbf{f}_I, \cdots, \mathbf{f}_1\right) \tag{14}$$

where $\mathbf{f}_i = (f^1_{i,1}, \ldots, f^1_{i,8}, f^2_{i,1}, \ldots, f^2_{i,8}, f^3_{i,1}, \ldots, f^3_{i,8})$ contains all the 24 features at the ith scale.

For the sake of clarity, the feature vector can be rewritten as

$$\mathbf{F} = (f(m))_{m=1}^{M} \tag{15}$$

where $M = 24 * I + 3$. Obviously, each feature $f(m)$ is nonnegative.

For illustration, we select two typical 512×512 texture images "Bark.0000" and "Bark.0004" from the VisTex database [33], which are shown in Fig. 2. As shown, the two texture images are very homogeneous. Then, we divide each of the two texture images into 16 nonoverlapping 128×128 patches and number them from 1 to 16. To extract the aforementioned defined features, we perform a four-scale contourlet transform to these patches, i.e., $I = 4$, and we can obtain the feature vector with 99 components. Fig. 3(a) and (b) shows the sketches of feature $f^2_{3,1}$ on the 16 patches from the two texture images, respectively.

Generally speaking, the minimum distance (MD) classifier can be used to classify the texture images when all the features have been obtained. However, from Fig. 3(a) and (b), it is shown that the contourlet feature $f^2_{3,1}$ of "Bark.0000" ranges from 6.34 to 16.59, whereas the contourlet feature $f^2_{3,1}$ of "Bark.0004" ranges from 6.89 to 11.3. Thus, the MD classifier is unsuitable for this feature. Thus, we here abandon the MD classifier but try to seek a Bayesian classifier for our texture classification. In order to do so, we need to model the contourlet features of each texture class by a reasonable probabilistic model. Usually, Gaussian mixtures can be employed to do such a job [38], [39]. However, the contourlet features such as $f^2_{3,1}$ are discretely distributed with a limited number of values. It is difficult to get a good estimation of the parameters in the Gaussian mixture model with only a small number of sample data via the current learning algorithms [38]–[41]. Oppositely, Poisson mixtures may be more efficient in this situation because only one parameter needs to be estimated for each Poisson component [30], [37]. As a random variable for a Poisson mixture is limited to be nonnegative integers, we can quantize the feature values into integers using the so-called deadzone quantization (step size $= 1$) suggested in [9]. By the adaptive gradient BYY harmony learning algorithm [37], we can obtain the distributions of feature $f^2_{3,1}$ for the two texture images or classes being shown in Fig. 4(a) and (b), respectively. It is shown that the distributions of $f^2_{3,1}$ for the two texture images are quite different from each other, and thus, we can make use of them to construct a Bayesian classifier to distinguish these two texture classes that the two images represent.

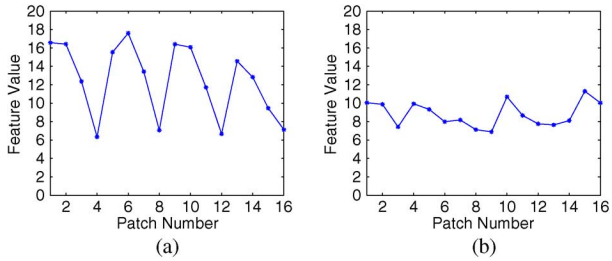


Fig. 3. Sketches of feature $f^2_{3,1}$ with respect to the patch number for (a) "Bark.0000" and (b) "Bark.0004."

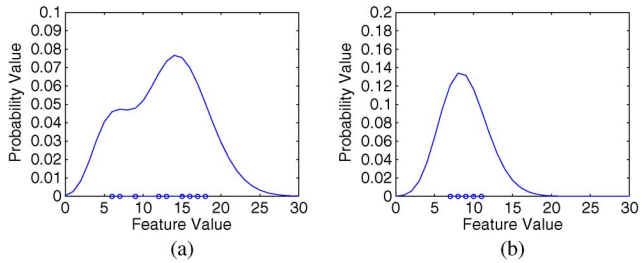


Fig. 4. Trained Poisson mixture probability distributions of feature $f^2_{3,1}$ for (a) "Bark.0000" and (b) "Bark.0004."

For simplicity, we still denote the quantized feature vector by $\mathbf{F}$, i.e., the feature vector in the following will further represent the feature vector quantized by the deadzone quantization without explanation. Meanwhile, we assume that each component of $\mathbf{F}$, i.e., $f(m)$, follows a Poisson mixture, i.e.,

$$P(f(m)) = \sum_{i=1}^{k_m} \alpha_{m,i} q(f(m)|\theta_{m,i}) \tag{16}$$

where all components $q(f(m)|\theta_{m,j})$ are Poisson distributions, $\alpha_{m,j}$ is the proportion of Poisson j in the mixture population, and k_m is the number of actual Poisson components corresponding to variable $f(m)$.

Moreover, we assume that all the components in $\mathbf{F}$ are independent. A brief explanation about the rationality of this independence assumption is given as follows. First, in order to check the independence of the contourlet feature variables, we randomly chose three feature variables in each directional subband and investigated their probability distributions over all the images of a data set. It was found that they have less dependence relations according to the statistics. Thus, these feature variables can be regarded as (at least approximately) being independent. Similarly, three features in the low-pass subband can be also regarded as being independent due to the fact that there are also less dependence relations among them. Second, many established wavelet-based methods are based on the assumption that all detail subbands are statistically independent, such as [15] and [18]. Thus, it is reasonable that contourlet subbands are assumed to be independent due to the fact that the contourlet transform is a 2-D extension of the wavelet transform. Based on these two aspects, we can make the assumption of the independence of all the contourlet feature variables in $\mathbf{F}$. Next, we will design a Bayesian classifier based on Poisson mixture models of these features.

B. Bayesian Texture Classifier Based on Poisson Mixtures

Based on these contourlet features data, we can construct a Bayesian texture classifier through the posteriori probability as follows:

$$P(T_c|\mathbf{F}) = \frac{P(\mathbf{F}|T_c)P(T_c)}{\sum_{i=1}^{C} P(\mathbf{F}|T_i)P(T_i)} \tag{17}$$

where T_c represents texture class c ($c = 1, 2, \ldots, C$), $P(T_c)$ is the prior probability of T_c, and $P(\mathbf{F}|T_c)$ denotes the conditional probability of the feature vector $\mathbf{F}$ with respect to the texture class T_c. After all the components in the posteriori probability formula are trained, the Bayesian classifier just assigns the texture class index c^* gaining the highest $P(T_c|\mathbf{F})$ to $\mathbf{F}$ or the corresponding texture image.

In the general case of texture classification on a given set of texture images, we can maintain that $P(T_c)$ is equally distributed, i.e., $P(T_c) = (1/C)$. Then, we need only to learn the parameters of those conditional probability distributions $P(\mathbf{F}|T_c)$. In fact, under the independence assumption of these contourlet features, the conditional probability distribution $P(\mathbf{F}|T_c)$ takes the following product form:

$$P(\mathbf{F}|T_c) = \prod_{m=1}^{M} P(f(m)|T_c) \tag{18}$$

where $P(f(m)|T_c) = \sum_{i=1}^{k_m} \alpha_{c,m,i} q(f(m)|\theta_{c,m,i})$ is just the Poisson mixture distribution with parameters $\{\alpha_{c,m,i}, \theta_{c,m,i}, i = 1, 2, \ldots, k_m\}$ to represent the probability of the mth contourlet feature $f(m)$ given the texture class T_c. For this Poisson mixture model, the model scale k_m is not known in advance.

TABLE I
LEARNING SCHEME OF THE BAYESIAN TEXTURE CLASSIFIER BASED ON THE BYY HARMONY LEARNING OF POISSON MIXTURES ON THE CONTOURLET FEATURES

[Input:] Training texture patches selected from each texture image or class (the number of training patches for each texture image is equal).
[Output:] The parameter sets of all the Poisson mixtures for each texture class.

(1) Decompose a patch of a given texture image or class c with contourlets into an output of I scales with 8 directional subbands at each scale.
(2) Calculate the value of the feature vector (15) with Eqs (8)-(13) for the patch and obtain the value of the quantized feature vector $\mathbf{F}$ by the deadzone quantization.
(3) Repeat the above two steps for all the training patches of the c-th texture class and obtain the training samples of the feature vector $\mathbf{F}$ for the c-th class.
(4) Implement the adaptive gradient BYY algorithm on the m-th components of all the training samples of the feature vector $\mathbf{F}$ for the c-th texture class, and obtain the parameters of the Poisson mixture distribution of the variable $f(m)$: $\Theta_{c,m} = \{\alpha_{c,m,i}, \theta_{c,m,i}, i = 1, 2, \cdots, k_m\}$.
(5) Repeat Step (4) for all the m components of the training samples of the feature vector $\mathbf{F}$ for the c-th texture class, and obtain the c-th class parameters: $\Theta_c = \{\Theta_{c,m}, m = 1, 2, \cdots, M\}$.
(6) Repeat the above steps for all texture classes, i.e., $c = 1, 2, \cdots, C$.

In such a way, the learning task of the Bayesian classifier becomes the parameter estimation and the model selection on Poisson mixtures for $f(m)$. In this case, we can utilize the adaptive gradient BYY learning algorithm [37] to learn the parameters of these Poisson mixtures with adaptive model selection and finally build the Bayesian classifier on the contourlet features of texture images. The details of our Bayesian classifier learning scheme are summarized in Table I.

In summary, for the purpose of supervised texture classification, we first perform the contourlet transform on the patches of each texture image and compute the feature vectors of these patches. Then, we implement the adaptive gradient BYY learning algorithm to learn the parameters of the Poisson mixtures with adaptive model selection for each contourlet feature with the feature values of those training patches and finally build the Bayesian classifier. In the test stage, for an input patch, we only need to calculate the conditional probabilities $P(\mathbf{F}|T_c)$ of its feature vector $\mathbf{F}$ with respect to all the C classes, $c = 1, 2, \ldots, C$. As a result, the class label gaining the highest $P(\mathbf{F}|T_c)$ is assigned to it.

V. EXPERIMENTAL RESULTS

Here, various experiments are carried out to demonstrate our proposed Bayesian texture classifier based on the BYY harmony learning of Poisson mixtures for the contourlet features of texture images, being compared with several current state-of-the-art approaches of texture classification under different image environments.

A. Performance Evaluation

We first evaluate our proposed Bayesian texture classifier on a typical set of 24 gray 640×640 images from the Brodatz image base (referred to as set 1 and shown in Fig. 5), which were also

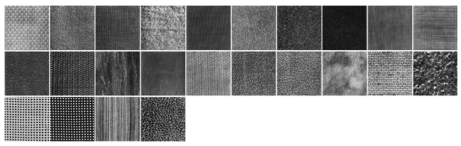


Fig. 5. Twenty-four Brodatz texture images of set 1.

used in [13]. In order to do so, we begin to describe how to set up those parameters in our learning paradigm.

In the feature extraction phase, we select the pyramid and directional filters by the "9–7" filters in the contourlet transform, which are the biorthogonal wavelet filters. In fact, with the other filter selection such as the "5–3" filters, the results are similar. On the other hand, we consider the contourlet transform with the number of contourlet decomposition scales varying from 1 to 5 for evaluation.

In the learning phase, we have investigated the distribution of $f(m)$ for $m = 1, 2, \ldots, M$. Numerous experiments reveal that its distribution is simple and can be approximated by the Poisson mixtures with the number of components less than 6. Thus, the number 6 is the value of k that should be selected and previously set for the adaptive gradient BYY harmony learning on Poisson mixtures. Now, we explain why such a value of k should be selected. In fact, the adaptive BYY harmony algorithm can make model selection adaptively during the parameter learning by forcing the mixing proportions of those $k-k_m$ extra Poissons to attenuate to zero. However, if k is too large (e.g., $k \geq 10$), the computational cost will be also large since much computation is needed for eliminating those extra Poissons. Thus, we cannot select a large number for k. In contrast, if k is too small (e.g., $k \leq 3$), Poisson mixtures may not be able to approximate the probability distributions of the contourlet features properly, which leads to a poor performance of texture classification. In this way and by experience, 6 is a reasonable choice of k in our situation.

According to the BYY best harmony learning principle [37], [40], [41], the adaptive gradient BYY algorithm for Poisson mixtures has converged or been stopped when the harmony function $J(\Theta_k)$ arrives or is stable at a maximum value. Thus, in our experiments, we stop the BYY harmony learning process if the absolute difference of the harmony functions at two sequential iterations of the algorithm, i.e., $|J(\Theta_k^{new}) - J(\Theta_k^{old})|$, becomes very small, being less than 10^{-4}. In addition, in order to keep a low computational cost of the algorithm in the general case, we also set an upper bound 10^3 for the number of iterations during the learning process. That is, the adaptive gradient BYY harmony learning algorithm is also stopped as long as the number of iterations is greater than 10^3. As a matter of fact, our experimental results reveal that the classification performance is almost the same if a larger upper bound is set for the number of iterations of the algorithm or a smaller threshold value is set for $|J(\Theta_k^{new}) - J(\Theta_k^{old})|$. The other parameters can be randomly initialized in certain intervals.

In our experiments, each image in set 1 is divided into 20 five nonoverlapping patches with the size of 128 × 128 pixels, and thus, there are 600 samples available. We select N training samples from each of 24 classes and let the other samples for test

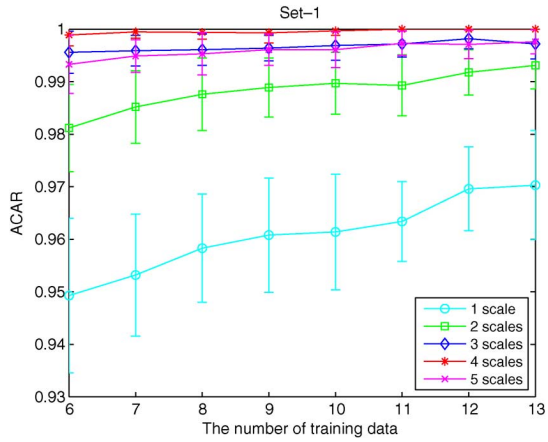


Fig. 6. Sketches of the ACARs of the proposed Bayesian texture classifier with respect to the number of training samples with the number of contourlet decomposition scales varying from 1 to 5.

for $N = 6, 7, \ldots, 13$. The partitions are furthermore randomly obtained, and the average classification accuracy rate (ACAR) is computed over the experimental results on 20 random splits of the training and test sets at each value of N. Fig. 6 plots the ACAR of our classifier versus the number of training samples N with error bars when the number of contourlet decomposition scales varies from 1 to 5. Note that each error bar in Fig. 6 is a distance of one standard deviation above and below the ACAR.

According to the experimental results, when the scale-1 contourlet decomposition is applied, we cannot obtain the enough directional contour information of the texture images. Thus, the ACAR of our Bayesian classifier is very poor. For the scale-2 or scale-3 contourlet decompositions, the ACAR of our Bayesian classifier significantly increases but does not reach the best value. For the scale-5 decompositions, we actually have 123 contourlet features to represent each texture. However, the sizes of the directional subbands at the fifth scale are very small, i.e., 2×4 and 4×2. Due to the small size of the subbands, estimated parameters are not reliable, resulting in poor classification accuracy rates with the scale-5 decompositions. As clearly shown in Fig. 6, the best performance of our Bayesian classifier is achieved when the scale-4 decompositions are performed.

It can be also observed from Fig. 6 that the difference between the ACARs for the scale-1 and scale-4 contourlet decompositions is about five percentage points, which is remarkable. However, the difference between the ACARs for the scale-2 and scale-4 contourlet decompositions is not so remarkable because it is less than one percentage point. It is the same for scale-3 or scale-5 contourlet decompositions. The highest ACAR of our method is 100% achieved at $N = 11, 12,$ and 13 when the scale-4 contourlet decompositions are performed, which shows that our Bayesian classifier is the optimal one on this texture set. On the other hand, the ACAR of our Bayesian classifier almost monotonically increases with the number of training samples. Although there is about two percentage points difference between the ACARs of $N = 6$ and $N = 13$ with the scale-1 contourlet decomposition, there is only about one percentage point difference between the ACARs of $N = 6$ and $N = 13$ with the

other scale contourlet decompositions. These indicate that the classification performance of our Bayesian classifier is not very sensitive to the number of training samples.

It should be noted that the error bars are also shown in Fig. 6. For the five decompositions, the value of standard deviation at each value of N is different, and for the scale-4 contourlet decompositions, the value of standard deviation decreases from 0.21% to 0 when N increases from 6 to 13, which implies that the variation of the classification rates for a different number of training samples is small and thus affirms the robustness of our Bayesian classifier. Moreover, the value of standard deviation with the scale-4 contourlet decompositions at each value of N is less than those with the other scale contourlet decompositions.

Finally, we compare our Bayesian classifier with the dominant local-binary-pattern (LBP) method [13] on this texture image set. The ACAR of our Bayesian classifier with 13 training samples and the scale-4 contourlet decompositions is 100%. That is, our Bayesian classifier can produce the best result of texture classification on this texture set. Certainly, the ACAR of our Bayesian classifier is higher than that of the dominant LBP method, i.e., 99.54%, reported in [13].

B. Comparison With the Other Existing Methods

1) Methods in the Comparison Study:

1) SVD + KLD [7]: The singular value decomposition (SVD) is applied on wavelet transformation coefficients of image textures, and then, the probability density function of the singular values of wavelet transformation coefficients is modeled as an exponential function. The model parameter of the exponential function is estimated using maximum likelihood estimation technique. The Kullback–Leibler distance (KLD) between estimated model parameters of image textures is used as a similarity metric to perform the classification using MD classifier.
2) HMT + MD [26]: Contourlet coefficients are modeled using an HMT model with Gaussian mixtures that can capture all interscale, interdirection, and interlocation dependences. The parameters of the HMT model are extracted as image features. The KLDs between the input image model and the mean of each database image model are measured. The MD classifier is used for texture classification. (The software for the implementation of the contourlet HMT model can be downloaded from MATLAB Central [35].)
3) HMT + KNN [26]: Same as HMT + MD but the K-nearest-neighbor (KNN) classifier with $K = 1$ is adopted. The KLDs between the input image model and each training image model are measured. The test sample is assigned to a class to which the closest training sample belong.
4) BP + L1 [9]: Each quantized wavelet coefficient is expanded into n binary bit planes, and the marginal distribution is then modeled by the product of these n independent Bernoulli distributions. The concatenation of the model parameters for all high-frequency subbands forms the so-called BP signature. A weighted L1-norm is used for similarity measurement. Then, the input texture image is assigned to a class according to the MD classifier.

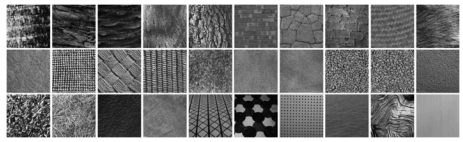


Fig. 7. Thirty VisTex texture images of set 2.

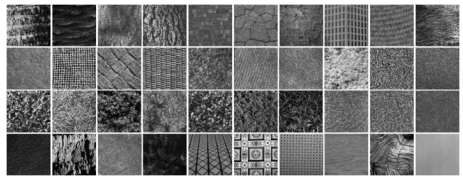


Fig. 8. Forty VisTex texture images of set 3.

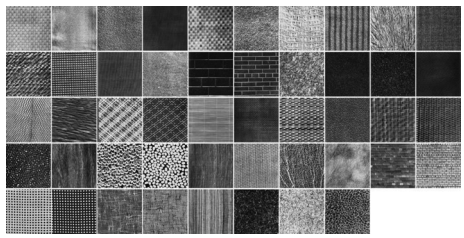


Fig. 9. Forty-eight Brodatz texture images of set 4.

5) BC-PMC: It is our proposed Bayesian Classifier based on the Poisson Mixture learning on the Contourlet features.

2) Data Sets: To evaluate the classification performance of the five methods, we utilize four large texture data sets, which consist of two Vistex data sets [33] and two Brodatz data sets [34]. The first Vistex data set consists of 30 of 512 × 512 texture images (denoted by set 2 and shown in Fig. 7), and the number of 512 × 512 texture images in the other Vistex data set (denoted by set 3 and shown in Fig. 8) is 40. The first Brodatz data set consists of 48 of 640 × 640 texture images (denoted by set 4 and shown in Fig. 9), and the other is the entire Brodatz texture data set consisting of 111 of 640 × 640 texture images (denoted by set 5).

In the experiments on sets 2 and 3, each texture image is divided into 16 of 128 × 128 nonoverlapping subimages, and thus, the two resulting databases are composed of 480 and 640 texture patches, respectively. In the experiments on sets 4 and 5, each texture image is divided into 16 of 160 × 160 nonoverlapping subimages, and thus, we can obtain 768 and 1776 texture patches, respectively. For each of the four data sets of texture patches, eight texture samples are selected from each texture class to design the classifiers; the others are adopted as test samples. The partitions are furthermore randomly obtained, and the ACAR is computed over the experimental results on 20 random splits of the training and test sets.

3) Results: The results are reported in Table II after implementing the SVD + KLD, HMT + MD, HMT + KNN, BP + L1, and BC-PMC approaches on the four data sets. As shown in Table II, for the VisTex data set 2, BC-PMC outperforms the other four methods by 6.12%–46.00%. Similarly, for the VisTex data set 3, BC-PMC performs better than them by 7.87%–57.15%. For the Brodatz data sets 4 and 5, BC-PMC can

TABLE II
ACAR (IN PERCENT) OF THE FIVE METHODS ON THE FOUR LARGE TEXTURE DATA SETS

	Set-2	Set-3	Set-4	Set-5
SVD + KLD	45.60 ± 1.86	34.64 ± 2.05	64.62 ± 1.65	32.66 ± 1.10
HMT + MD	58.92 ± 2.91	56.08 ± 3.37	78.14 ± 2.93	n.a.
HMT + KNN	77.21 ± 2.02	75.73 ± 2.05	91.05 ± 1.36	n.a.
BP + L1	85.48 ± 1.98	83.91 ± 1.74	94.56 ± 0.84	78.03 ± 1.48
BC-PMC	91.60 ± 1.42	91.78 ± 1.38	97.70 ± 0.99	86.53 ± 1.12

TABLE III
ACAR (IN PERCENT) FOR EACH OF THE 48 TEXTURE CLASSES IN SET 4 WITH THE FIVE METHODS. COLUMN 1: OUR PROPOSED BC-PMC; COLUMN 2: BP + L1; COLUMN 3: HMT + KNN; COLUMN 4: HMT + MD; COLUMN 5: SVD + KLD

	1	2	3	4	5		1	2	3	4	5
D1	99.38	100	93.13	58.13	48.75	D49	100	100	100	98.75	100
D3	98.75	90.63	73.75	37.50	39.38	D52	95.00	41.25	96.25	56.88	69.38
D4	100	98.75	82.50	72.50	53.75	D56	100	100	100	100	83.75
D6	100	100	100	95.63	96.88	D57	100	100	90.00	76.25	76.25
D8	100	100	98.13	62.50	56.25	D64	100	100	98.13	98.75	65.63
D9	100	96.88	84.38	83.13	30.63	D65	100	100	98.75	95.63	86.88
D10	99.38	86.25	98.75	95.00	68.13	D66	100	100	100	100	91.88
D11	100	100	100	93.13	39.38	D68	100	100	100	99.38	58.75
D15	93.75	86.88	94.38	75.63	83.75	D74	99.38	98.75	99.38	100	58.13
D17	100	100	93.13	80.63	78.75	D75	97.50	100	83.13	75.00	96.88
D18	100	100	95.00	76.25	75.00	D79	100	100	96.25	93.75	61.25
D20	100	100	98.75	89.38	75.63	D82	100	100	99.38	63.13	84.38
D21	100	100	91.88	20.00	100	D87	100	100	97.50	71.25	42.50
D22	95.00	84.38	86.88	75.63	81.88	D93	100	90.00	86.88	21.25	15.00
D25	95.63	93.13	100	99.38	45.00	D94	94.38	73.75	84.38	63.75	45.63
D26	100	99.38	100	99.38	49.38	D95	100	100	98.13	99.38	85.63
D28	100	100	96.88	80.63	45.00	D101	100	88.13	43.13	40.63	55.00
D32	100	100	93.13	89.38	55.63	D102	100	88.13	50.00	58.75	63.75
D33	100	99.38	96.88	88.75	48.13	D103	69.38	86.25	70.63	68.13	52.50
D34	100	100	100	94.38	93.75	D104	60.00	81.88	56.88	46.25	61.25
D35	100	100	75.00	36.88	43.13	D106	100	100	99.38	99.38	90.00
D37	100	89.38	98.75	74.38	53.75	D109	93.75	75.63	95.63	91.88	44.38
D46	98.13	100	98.75	88.75	61.88	D110	100	90.00	84.38	93.13	56.88
D47	100	100	94.38	74.38	75.00	D111	100	100	98.13	98.13	57.50
						Mean	97.70	94.56	91.05	78.14	64.62

obtain the recognition rates 97.70% and 86.53%, respectively, and outperforms them by at least 3.14%.

To intensively compare with the four methods (SVD + KLD, HMT + MD, HMT + KNN, and BP + L1), the classification accuracy rates of all 48 texture classes are obtained and shown in Table III. From Table III, it can be observed that our method does not perform worse than the other four methods for 38 texture classes. Our Bayesian classifier arrives a 100% classification accuracy rate on 34 texture classes, which is larger than the number of the texture classes recognized with no error by BP + L1, i.e., 28, and clearly exceeds those by the other methods. The worst classification accuracy rates of the five methods, i.e., BC-PMC, BP + L1, HMT + KNN, HMT + MD, and SVD + KLD, are 60.00%, 41.25%, 43.13%, 20.00%, and 15.00%, respectively. Fig. 10(a) shows two textures that are easily confused in the experiments by our method, which is the main reason that we can obtain only 97.70% recognition rate rather than a higher ACAR for the whole data set. From Table III, it is shown that BP + L1 outperforms our method by about 20.00% for the two texture classes, which implies that the BP signatures in the undecimated wavelet domain is superior to our extracted features in the contourlet domain in distinguishing the two textures that have the confused directionality and gray level. In contrast, Fig. 10(b) shows five textures for which our method outperforms the other methods by more than 6.00%. From this figure, it can be seen that the textures better classified by our method exhibit more diverse directional components (such as circular or irregular shapes), on which the ACAR of BP + L1 is less than that of our method by about 10.00%–21.00%. This shows the superiority of the contourlet in capturing directional information. As far as the ACAR for the whole data set, i.e., the mean of the ACARs for all classes, is concerned, our proposed BC-PMC performs better than the four methods by 3.14%–33.08%.

In summary, we compare our proposed Bayesian texture classifier with four current state-of-the-art methods of texture classification and find out that our method significantly improves the classification accuracy rate on different sets of texture images.

C. Computational Cost

All the experiments in this paper have been implemented on a workstation with Intel Core i5 central processing unit (3.2 GHz) and 3-G random access memory in the MATLAB environment.

Table IV reports the time for texture classification (TTC) running each of the SVD + KLD, HMT + MD, HMT + KNN, BP + L1, and BC-PMC approaches on data sets 4 and 5. From Table IV, it is observed that the SVD + KLD method is the most efficient; the reason is that, since the probability density function

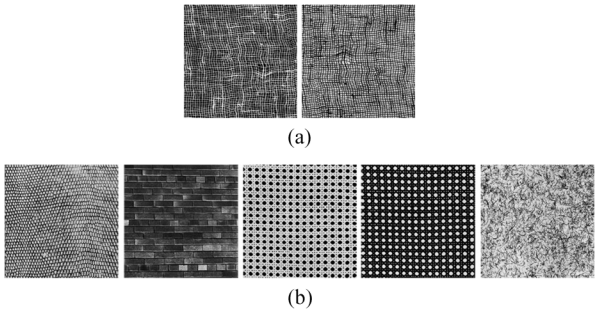


Fig. 10. (a) Two textures of set 4 confused by our proposed BC-PMC. From left to right: D103 and D104. (b) From left to right: D22, D94, D101, D102, and D110.

TABLE IV
TTC (IN SECONDS, $\times 10^3$) OF THE FIVE METHODS. COLUMN 1: OUR PROPOSED BC-PMC; COLUMN 2: BP + L1; COLUMN 3: HMT + KNN; COLUMN 4: HMT + MD; COLUMN 5: SVD + KLD

	1	2	3	4	5
Set-4	0.415	0.479	9.065	9.066	0.048
Set-5	0.830	1.058	n.a.	n.a.	0.127

of the singular values is modeled by an exponential density function, the parameter estimation requires a less number of computations. However, since only one parameter is used to represent each wavelet subband, this method does not have a good generalization ability when it is applied to the texture classification, which can be seen from the texture classification results given in the previous subsection. That is, its ACARs are really unsatisfactory. In contrast, HMT + MD is the most time-consuming method among them. Our BC-PMC is about 22 times faster than both HMT + MD and HMT + KNN on set 4, which implies that, as a different modeling method in the same contourlet transform domain, our proposed Poisson mixture learning is significantly more efficient than the training of the HMT model. In addition, our BC-PMC is also slightly more efficient than the current state-of-the-art BP + L1 method. In our BC-PMC approach, the most costly part is the parameter estimations of the Poisson mixtures by the BYY harmony learning. In the HMT + MD and HMT + KNN approaches, the EM algorithm was utilized to learn the parameters of the HMT model, which requires more computational cost than our method.

If we take into account the TTC and the ACAR, the experimental results clearly show that our proposed BC-PMC method outperforms other methods.

VI. CONCLUSION

We have established a novel kind of Bayesian texture classifier through the BYY harmony learning of Poisson mixtures on the contourlet features of texture images. The contourlet decomposition sets up an efficient representation of the image for texture classification, whereas the BYY harmony learning of Poisson mixtures can lead to a good parameter estimation with adaptive model selection. These two advantages enable the proposed Bayesian texture classifier to be more effective and efficient. The various experiments show that our proposed Bayesian classifier significantly improves the texture classification accuracy in comparison with the current state-of-the-art approaches.

It is important to note that our proposed Bayesian texture classifier is a naive one, which is constructed under the independence assumption of the extracted features. Although the intercoefficient dependences in the contourlet domain have been explored and modeled by the HMT method, the experimental results show that such a method is not more efficient and effective than our naive classifier. Moreover, our proposed Bayesian texture classifier can be further improved if a better and applicable BYY harmony learning algorithm for multivariate Poisson mixtures is established in the future.

ACKNOWLEDGMENT

The authors would like to thank Dr. S. K. Choy and Prof. C. S. Tong of Hong Kong Baptist University for providing the source code of the bit-plane-probability-based method (BP + L1).

REFERENCES

[1] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Texture features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
[2] T. H. Hong, C. R. Dyer, and A. Rosenfeld, "Texture primitive extraction using an edge-based approach," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-10, no. 10, pp. 659–675, Oct. 1980.
[3] P. Garcia-Sevilla and M. Petrou, "Classification of binary textures using the 1-D Boolean model," *IEEE Trans. Image Process.*, vol. 8, no. 10, pp. 1457–1462, Oct. 1999.
[4] F. S. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 803–808, Aug. 1991.
[5] A. Speis and G. Healey, "Feature extraction for texture discrimination via random field models with random spatial interaction," *IEEE Trans. Image Process.*, vol. 5, no. 4, pp. 635–645, Apr. 1996.
[6] S. C. Kim and T. J. Kang, "Texture classification and segmentation using wavelet packet frame and Gaussian mixture model," *Pattern Recognit.*, vol. 40, no. 4, pp. 1207–1221, Apr. 2007.
[7] S. Selvan and S. Ramakrishnan, "SVD-based modeling for image texture classification using wavelet transformation," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2688–2696, Nov. 2007.
[8] K. Huang and S. Aviyente, "Wavelet feature selection for image classification," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1709–1720, Sep. 2008.
[9] S. K. Choy and C. S. Tong, "Statistical properties of bit-plane probability model and its application in supervised texture classification," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1399–1405, Aug. 2008.
[10] M. Masotti and R. Campanini, "Texture classification using invariant ranklet features," *Pattern Regcognit. Lett.*, vol. 29, no. 14, pp. 1980–1986, Oct. 2008.
[11] G. Liu, Z. Lin, and Y. Yu, "Radon representation-based feature descriptor for texture classification," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 921–928, May 2009.
[12] S. Arivazhagan, L. Ganesan, and T. G. Subash Kumar, "Texture classification using ridgelet transform," *Pattern Regcognit. Lett.*, vol. 27, no. 16, pp. 1875–1883, Dec. 2006.
[13] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1107–1118, May 2009.
[14] S. K. Choy and C. S. Tong, "Supervised texture classification using characteristic generalized Gaussian density," *J. Math. Imag. Vis.*, vol. 29, no. 1, pp. 35–47, Sep. 2007.
[15] L. Li, C. S. Tong, and S. K. Choy, "Texture classification using refined histogram," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1371–1378, May 2010.
[16] H. Le Borgne, A. Guerin Dugue, and N. E. O'Connor, "Learning midlevel image features for natural scene and texture classification," *IEEE Trans. Circuits Syst. Vid. Technol.*, vol. 17, no. 3, pp. 286–297, Mar. 2007.
[17] H. Lategahn, S. Gross, T. Stehle, and T. Aach, "Texture classification by modeling joint distributions of local patterns with Gaussian mixtures," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1548–1557, Jun. 2010.

[18] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[19] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
[20] D. K. Hammond and E. P. Simoncelli, "Image modeling and denoising with orientation-adaptived Gaussian scale mixtures," *IEEE Trans. Image Process.*, vol. 17, no. 11, pp. 2089–2101, Nov. 2008.
[21] M. L. Comer and E. J. Delp, "Segmentation of textured images using a multiresolution Gaussian autoregressive model," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 408–420, Mar. 1999.
[22] M. L. Comer and E. J. Delp, "The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1731–1744, Oct. 2000.
[23] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.
[24] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983.
[25] R. H. Bamberger and M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 882–893, Apr. 1992.
[26] D. D.-Y. Po and M. N. Do, "Directional multiscale modeling of images using the contourlet transform," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1610–1620, Jun. 2006.
[27] G. Y. Che and B. Kegl, "Invariant pattern recognition using contourlets and AdaBoost," *Pattern Recognit.*, vol. 43, no. 3, pp. 579–583, Mar. 2010.
[28] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, Oct. 2006.
[29] R. Eslami and H. Radha, "Translation-invariant contourlet transform and its application to image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3352–3374, Nov. 2006.
[30] D. Karlis and E. Xekalaki, "Mixed Poisson distributions," *Int. Statist. Rev.*, vol. 73, no. 1, pp. 35–58, Apr. 2005.
[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.
[32] D. Karlis, "An EM algorithm for multivariate Poisson distribution and related models," *J. Appl. Stat.*, vol. 30, no. 1, pp. 63–77, 2003.
[33] [Online]. Available: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html
[34] [Online]. Available: http://www.ux.uis.no/ tranden/brodatz.html
[35] [Online]. Available: http://www.mathworks.cn/matlabcentral/index.html
[36] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.
[37] J. Ma, J. Liu, and Z. Ren, "Parameter estimation of Poisson mixture with automated model selection through BYY harmony learning," *Pattern Recognit.*, vol. 42, no. 11, pp. 2659–2670, Nov. 2009.
[38] J. Ma and J. Liu, "The BYY annealing learning algorithm for Gaussian mixture with automated model selection," *Pattern Recognit.*, vol. 40, no. 7, pp. 2029–2037, Jul. 2007.
[39] J. Ma and X. He, "A fast fixed-point BYY harmony learning algorithm on Gaussian mixture with automated model selection," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 701–711, Apr. 2008.
[40] L. Xu, "Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models," *Int. J. Neural Syst.*, vol. 11, no. 1, pp. 43–69, Feb. 2001.
[41] L. Xu, "BYY harmony learning, structural RPCL, and topological self-organizing on mixture modes," *Neural Netw.*, vol. 15, no. 8/9, pp. 1125–1151, Oct./Nov. 2002.

Yongsheng Dong received the M.S. degree in applied mathematics from Wuhan University, Wuhan, China, in 2005. He is currently working toward the Ph.D. degree in the Department of Information Science, School of Mathematical Sciences, Peking University, Beijing, China.

From 2005 to 2008, he was with Henan Polytechnic University, Jiaozuo, China. His current research interests include image processing, pattern recognition, wavelets, and statistical machine learning.

Jinwen Ma received the M.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 1988 and the Ph.D. degree in probability theory and statistics from Nankai University, Nankai, China, in 1992.

From July 1992 to November 1999, he was a Lecturer and an Associate Professor with the Department of Mathematics, Shantou University, Shantou, China. In 1995 and 2003, he was a Research Associate or a Fellow with the Department of Computer Science and Engineering, the Chinese University of Hong Kong, Shatin, Hong Kong. In December 1999, he was a full Professor with the Institute of Mathematic, Shantou University. Since September 2001, he has been with the Department of Information Science, School of Mathematical Sciences, Peking University, Beijing, China, where he is currently a full Professor and a Ph.D. tutor. From September 2005 to August 2006, he was a Research Scientist with Amari Research Unit, RIKEN Brain Science Institute, Wako, Japan. He has published over 100 academic papers on neural networks, pattern recognition, bioinformatics, and information theory.