Texture Classification Using Spectral Histograms

Xiuwen Liu, *Senior Member, IEEE,* and DeLiang Wang, *Senior Member, IEEE*

***Abstract*—Based on a local spatial/frequency representation,we employ a spectral histogram as a feature statistic for texture classification. The spectral histogram consists of marginal distributions of responses of a bank of filters and encodes implicitly the local structure of images through the filtering stage and the global appearance through the histogram stage. The distance between two spectral histograms is measured using χ^2-statistic. The spectral histogram with the associated distance measure exhibits several properties that are necessary for texture classification. A filter selection algorithm is proposed to maximize classification performance of a given dataset. Our classification experiments using natural texture images reveal that the spectral histogram representation provides a robust feature statistic for textures and generalizes well. Comparisons show that our method produces a marked improvement in classification performance. Finally we point out the relationships between existing texture features and the spectral histogram, suggesting that the latter may provide a unified texture feature.**

***Index Terms*—Filtering, spectral histogram, texture analysis, texture classification.**

I. Introduction

TEXTURE classification is a fundamental problem in computer vision with a wide variety of applications [40]. Two fundamental issues in texture classification are how to characterize textures using derived features and and how to define a robust distance/similarity measure between textures, which remain elusive despite considerable efforts in the literature [34]. Because images of the same underlying texture can vary significantly, textural features must be invariant to (large) image variations and at the same time sensitive to intrinsic spatial structures that define textures. Because there is no obvious feature common for all texture images, texture features are often proposed based on assumptions for mathematical convenience or task-specific heuristics (see [35], [40] for reviews). For example, geometric properties based on the texture elements are often used for textures with periodic structures [39]. Early features including cooccurrence matrices [17] and Markov random field models [7] have limited expressive power because the analysis of spatial interaction is limited to a relatively small neighborhood. As a result, the adequacy of these features for characterizing various textures is rarely checked.

Manuscript received November 29, 2001; revised January 23, 2003. This work was supported in part by NIMA grant (NMA201-01-2010) to X. Liu, an ONR Young Investigator Award (N00014-96-1-0676), an NSF grant (IIS-0081058), and an AFOSR grant (F49620-01-1-0027) to D. L. Wang. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Josiane B. Zerubia.

X. Liu is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306-4530 USA (e-mail: liux@cs.fsu.edu).

D. L. Wang is with the Department of Computer and Information Science, Center for Cognitive Science, The Ohio State University, Columbus, OH 43210 USA (e-mail: dwang@cis.ohio-state.edu).

Digital Object Identifier 10.1109/TIP.2003.812327

On the other hand, studies on the human visual system suggest that it transforms a retinal image into a local spatial/frequency representation [4], [10], which can be computationally simulated by convolving the input image with a bank of filters with tuned frequencies and orientations. The mathematical framework for the local spatial/frequency representation was laid out by Gabor [13] and extended by Daughman [8]. Recently, this theory has also been confirmed by deriving similar feature detectors from natural images [33]. These advances have inspired much research in texture classification based on filtering (see [34] for a review). In this framework, a texture image is transformed into feature vectors by filtering the input image using a bank of filters, followed by some nonlinearity and smoothing steps [34]. The nonlinearity is necessary for texture classification, since, otherwise, filter responses cannot discriminate textures with the same mean intensity (see, e.g., Malik and Perona [29]); the smoothing is necessary since the filter responses are not homogeneous even within a homogeneous texture region. While the nonlinearity and smoothing steps are critical for texture classification, current research focuses instead on the filtering stage, i.e., deriving optimal filters for texture classification based on certain optimization criteria. As a result, while both the theoretical and numerical aspects of filter design for texture classification are well studied [30], the recent comprehensive study by Randen and Husoy [34] showed that the texture classification performance is very limited for real textures. This study clearly leads to the need for studying statistic features beyond the filtering stage for texture classification.

Recently, Heeger and Bergen [18] proposed a texture synthesis algorithm that can match texture appearance. The algorithm transforms a random noise image into one with similar appearance to a given target image by matching independently the histograms of image pyramids constructed from the random and target images. The success of synthesizing natural textures based on histograms has motivated considerable research [43], [44]. Zhu *et al.* [44] proposed a theory for learning probability models by matching histograms based on a maximum entropy principle. Zhu *et al.* [43] studied efficient sampling algorithms for matching histograms. While these synthesis methods provide features to characterize a single texture, the effectiveness of these features for texture classification is not known as a good synthesis model does not imply a good classification model (see Section III). Also, while these synthesis methods are proposed to model homogeneous textures, natural textures are rarely homogeneous due to deformations and other variations; these variations require a robust distance measure between textures so that the distance between images of the same texture is small and that among images from different textures is large. Furthermore, as the features depend on the choice of filters, there is no systematic algorithm to choose filters for texture classification. In addition, texture classification is often done based on rela-

tively small image windows and the effect of the window size on histogram-based representations needs to be studied.

Motivated by the research on texture synthesis, we propose a local spectral histogram, consisting of marginal distributions of responses from a bank of filters for an image window, as a feature statistic for texture classification. We define a distance between two image windows as the χ^2-statistic of their spectral histograms, which exhibits nonlinearity that is consistent with the human texture perception [27]. Our work presented elsewhere [27] demonstrates that this proposed model provides a satisfactory account for a systematic set of human texture discrimination data. We propose a filter selection algorithm to optimize the classification performance of a given dataset. We report that the spectral histogram produces good classification results. A systematic comparison with other methods documented in [34] demonstrates that our approach yields far better results.

This paper is organized as follows. Section II introduces the local spectral histogram model and discusses its properties for texture classification. Section III presents a filter selection algorithm and shows the classification results on a natural texture dataset. Section IV compares our method with existing approaches. Section V discusses the relations of the spectral histogram model with previous ones and some related issues. Section VI concludes the paper.

II. Local Spectral Histogram

A. Definition

Given an image window $\mathbf{W}$ and a bank of filters $\{F^{(\alpha)}, \alpha = 1, 2, \ldots, K\}$, [1] we compute, for each filter $F^{(\alpha)}$, a sub-band image $\mathbf{W}^{(\alpha)}$ through linear convolution. [2] That is, $\mathbf{W}^{(\alpha)}(\vec{v}) = F^{(\alpha)} * \mathbf{W}(\vec{v}) = \sum_{\vec{u}} F^{(\alpha)}(\vec{u})\mathbf{W}(\vec{v} - \vec{u})$, at pixel location $\vec{v}$, where a circular boundary condition is used. For $\mathbf{W}^{(\alpha)}$, we define the marginal distribution, or histogram as

$$H_{\mathbf{W}}^{(\alpha)}(z) = \frac{1}{|\mathbf{W}|} \sum_{\vec{v}} \delta\left(z - \mathbf{W}^{(\alpha)}(\vec{v})\right) \tag{1}$$

where $\delta()$ is the Dirac delta function. We then define the spectral histogram with respect to the chosen filters as

$$H_{\mathbf{W}} = \left(H_{\mathbf{W}}^{(1)}, H_{\mathbf{W}}^{(2)}, \ldots, H_{\mathbf{W}}^{(K)}\right). \tag{2}$$

Here, the concatenation of different histograms assumes the independence among different filters; under the independence assumption, the distance between two image windows is simplified to the sum of the distance between the corresponding histograms of each filter as shown in (3). This is justified for natural images since edge-like filters are empirically shown to be the independent components of natural images [2]. The spectral histogram of an image or an image window is essentially a vector consisting of marginal distributions of filter responses and integrates responses of different filters to form a texture feature. The size of the input image window is called *integration scale*. Because the marginal distribution of each filter response is a probability distribution, we define a distance between two spectral histograms $H_{\mathbf{W}_1}$ and $H_{\mathbf{W}_2}$ as

$$\begin{aligned}\chi^2(H_{\mathbf{W}_1}, H_{\mathbf{W}_2}) &= \sum_{\alpha=1}^{K} \sum_{z} \frac{\left(H_{\mathbf{W}_1}^{(\alpha)}(z) - H_{\mathbf{W}_2}^{(\alpha)}(z)\right)^2}{H_{\mathbf{W}_1}^{(\alpha)}(z) + H_{\mathbf{W}_2}^{(\alpha)}(z)} \\ &= \sum_{\alpha=1}^{K} \chi^2\left(H_{\mathbf{W}_1}^{(\alpha)}, H_{\mathbf{W}_2}^{(\alpha)}\right)\end{aligned} \tag{3}$$

where χ^2-statistic is a first-order approximation of the Kullback-Leibler divergence and used widely to compare histograms.

B. Properties

The spectral histogram integrates responses of different filters and provides a normalized feature statistic to compare images of different sizes. Some of its properties are discussed below.

Property 1: A spectral histogram is translation invariant.

This property is easy to see from the definition of the spectral histogram. Because filter responses depend only on relative locations of pixels, the absolute position of an image window does not affect its spectral histogram. This is essential for any texture model to characterize texture appearance.

Property 2: A spectral histogram is a nonlinear operator.

In other words, the spectral histogram of images does not satisfy the linearity conditions due to the histogram operation given in (1). As discussed earlier, some form of nonlinearity is necessary for texture classification and the intrinsic nonlinearity of the spectral histogram model makes an additional nonlinearity step not needed while it is necessary for other filtering-based methods [34]. For nonconstant filters, the nonlinearity of the spectral histogram can also be caused by the dependency among pixel values, and this makes the spectral histogram sensitive to texture structures.

Property 3: With sufficient filters, a spectral histogram can uniquely represent any image up to a translation.

To show this, let $\mathbf{I}$ be an image defined on a finite lattice $\mathcal{L}$. If $\mathbf{I}(\vec{v}) = 0, \forall \vec{v} \in \mathcal{L}$, the proposition holds. Assume that $\sum_{\vec{v}} |\mathbf{I}(\vec{v})| > 0$, we choose two filters, the intensity filter $F^{(1)} = \delta()$ and $F^{(2)}(\vec{v}) = \mathbf{I}(\vec{v}_0 - \vec{v}), \forall \vec{v} \in \mathcal{L}$, where $\vec{v}_0 \in \mathcal{L}$. It is sufficient to show that $\forall \mathbf{J}$, an image defined on the finite lattice $\mathcal{L}$, $\mathbf{J}$ is equivalent to $\mathbf{I}$ up to a translation if $H_{\mathbf{J}}^{(1)}(z) = H_{\mathbf{I}}^{(1)}(z)$ and $H_{\mathbf{J}}^{(2)}(z) = H_{\mathbf{I}}^{(2)}(z), \forall z$. If $H_{\mathbf{J}}^{(1)} = H_{\mathbf{I}}^{(1)}$, $\mathbf{J}$ must be a permutation of $\mathbf{I}$ in terms of the group of pixels. For $F^{(2)}$, the maximum response of the filter is bounded

$$\begin{aligned}\mathbf{I}^{(2)}(\vec{v}) &= F^{(2)} * \mathbf{I}(\vec{v}) = \sum_{\vec{u}} F^{(2)}(\vec{u})\,\mathbf{I}(\vec{v} - \vec{u}) \\ &\le \sqrt{\sum_{\vec{u}} \left(F^{(2)}(\vec{u})\right)^2} \sqrt{\sum_{\vec{u}} \left(\mathbf{I}(\vec{v} - \vec{u})\right)^2} \\ &= \sum_{\vec{u}} \left(\mathbf{I}^{(2)}(\vec{u})\right)^2\end{aligned}$$

due to Cauchy-Schwartz' inequality. The maximum is achieved when $\vec{v} = \vec{v}_0$. Similarly, if $\mathbf{J}$ is a permutation of $\mathbf{I}$ and $H_{\mathbf{J}}^{(2)}(z) = H_{\mathbf{I}}^{(2)}(z), \forall z$, $\mathbf{J}$ must be equivalent to $\mathbf{I}$ up to a translation to achieve the same maximum response.

[1] We require that $\sum_{\vec{u}} |F^{(\alpha)}(\vec{u})| > 0$ for any α. In other words, $F^{(\alpha)}$ must have some nonzero coefficients.

[2] In this work, we restrict the definition of the spectral histogram to linear filters, even through nonlinear filters such as the power spectrum of filter pairs can also be included.

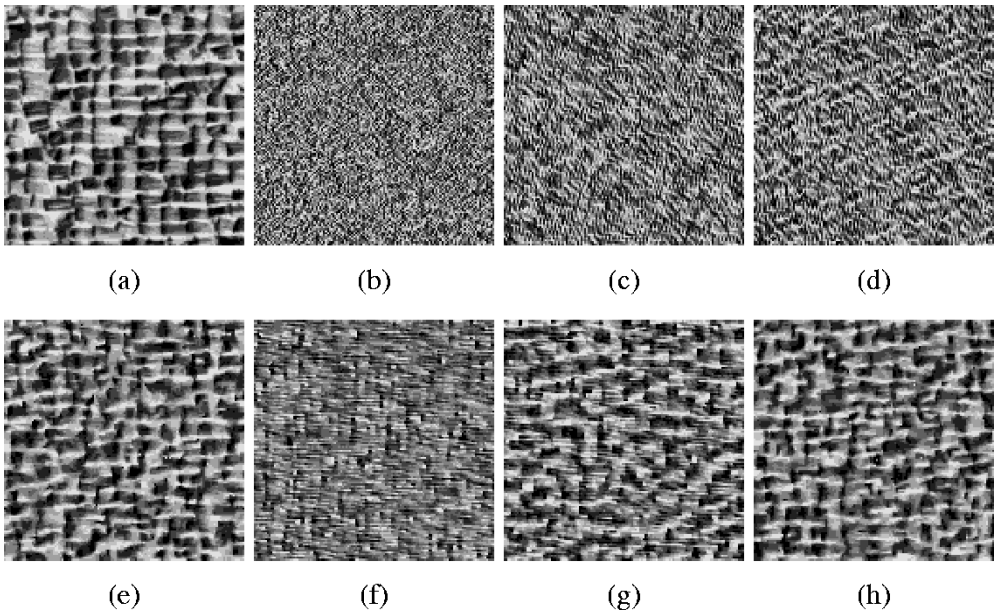

Fig. 1. A typical image that satisfies $H_{\mathbf{I}} = H_{\text{obs}}$ with different filters. The size of the images is 128×128. Here the filters correspond to the ones in Table I for texture classification. (a) A texture image. (b) Three filters. (c) Five filters. (d) Seven filters. (e) Forty filters. (f) Three filters without the intensity filter. (g) Six filters without the intensity filter. (h) Thirty-nine filters without the intensity filter.

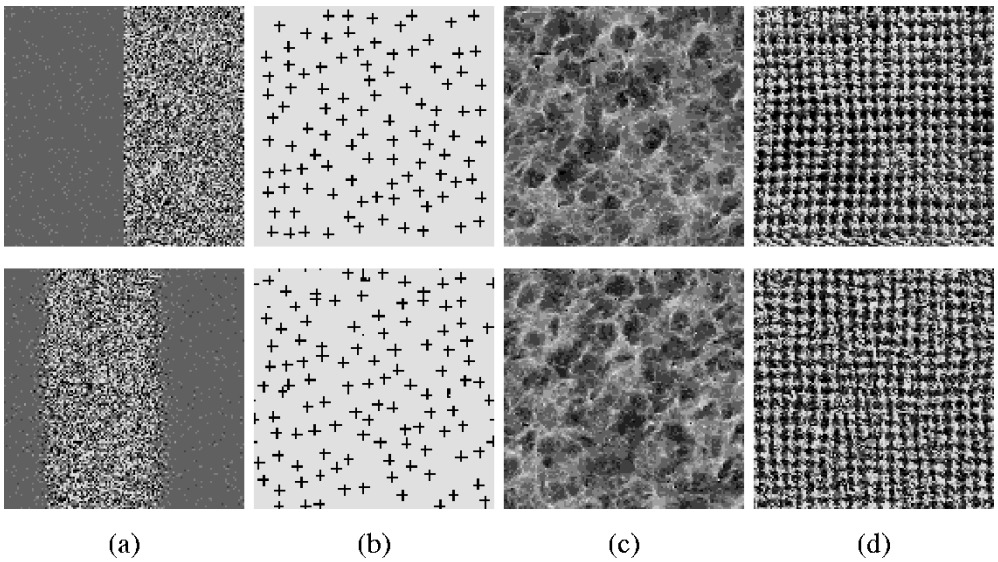

Fig. 2. Different types of images characterized by spectral histograms. Top row shows observed images and the bottom row the corresponding typical image that shares the same spectral histogram. (a) A gray-level image consisting of two piece-wise constant regions with additive Gaussian noise. (b) A synthetic texture consisting of cross elements. (c) A stochastic image. (d) An image with periodic structures.

While this property does not provide a practical way to choose filters, it shows that the marginal distributions are sufficient to characterize the joint distributions implicitly defined by an image and thus justifies the sufficiency of using only marginal distributions in the spectral histogram. In practice, this property is approximated by using a large number of filters. Intuitively, each filter provides some constraints on the images that can satisfy $H_{\mathbf{I}} = H_{\text{obs}}$. With constraints imposed by sufficiently many filters, the solution to the equation $H_{\mathbf{I}} = H_{\text{obs}}$ would converge to the observed image up to a translation. Fig. 1 gives an illustration. With more filters, the images that share the observed spectral histogram become more similar to the observed.

Property 4: All the images sharing a spectral histogram define an equivalent class.

Essentially, spectral histograms divide all the images into equivalent classes [43]. Extensive simulations suggest that the spectral histogram is sufficient in characterizing texture appearance [24], [43] when filters are chosen properly. In other words, all the images with the same spectral histogram are perceptually similar in that perceptually similar textures are synthesized by matching the spectral histogram. The top row of Fig. 2 shows four texture images and the bottom row shows corresponding typical images by satisfying the constraints $H_{\mathbf{I}} = H_{\text{obs}}$, where $\mathbf{I}$ is an image, $H_{\mathbf{I}}$ its spectral histogram, and H_{obs} the spectral histogram of the observed image. Due to the high dimensionality of $\mathbf{I}$, the constraints have to be satisfied through stochastic simulation because traditional deterministic searching methods are computationally not feasible. These

examples shown in Fig. 2 were generated using a Gibbs sampler [14], [43]. In Fig. 2(a), the spectral histogram captures the perceptual appearance of both regions. Given that the circular boundary is used, the synthesized image matches closely the observed image. Fig. 2(b) shows a synthetic texture image, where the spectral histogram captures the texture element and its density. Fig. 2(c) and (d) show that the spectral histograms of two stochastic textures capture their perceptual appearance well.

C. Implementation Issues

Because a spectral histogram is defined with respect to a bank of filters, the first implementation issue is what filters should be used so that various textures can be modeled effectively. Here we use four different types of filters suggested from the studies of visual perception and the empirical studies of independent components of natural images [2], [33], [44].

1) The intensity filter, which is the $\delta()$ function and captures the intensity value at a given pixel.
2) Difference or gradient filters. We use four of them: $D_x = [0.0 \quad -1.0 \quad 1.0]$, $D_{xx} = [-1.0 \quad 2.0 \quad -1.0]$, $D_y = \begin{bmatrix} 0.0 \\ -1.0 \\ 1.0 \end{bmatrix}$, $D_{yy} = \begin{bmatrix} -1.0 \\ 2.0 \\ -1.0 \end{bmatrix}$.
3) Laplacian of Gaussian filters:
$$LoG(x, y|T) = \left(x^2 + y^2 - T^2\right) e^{(-x^2+y^2)/T^2} \tag{4}$$
where $T = \sqrt{2}\sigma$ determines the scale of the filter and σ is the variance of the Gaussian function. These filters are referred to as $LoG(T)$.
4) The Gabor filters with both sine and cosine components:
$$Gabor(x, y|T, \theta) = e^{-1/2T^2(4(x\cos\theta + y\sin\theta)^2 + (-x\sin\theta + y\cos\theta)^2)} e^{-i2\pi/T(x\cos\theta + y\sin\theta)} \tag{5}$$
where T is a scale. The cosine and sine components of these filters are referred to as $Gcos(T, \theta)$ and $Gsin(T, \theta)$, respectively.

These filters provide efficient ways of extracting spatial structures at different orientations and frequencies and empirically have shown to be effective for different kinds of textures (see Fig. 2 for some examples). Given these families of filters, the optimal ones for classification of a given set of textures depend on the characteristics of the input textures; they are selected by a filter selection algorithm presented in Section III-A.

Another implementation issue is how to quantize and estimate the histogram of a filtered image. In theory, the histogram is an approximation of the underlying distribution of filter responses. With sufficient data and sufficient number of histogram bins, the histogram can represent the underlying distribution with arbitrary accuracy. For texture classification, the integration scale we use is often small and only a limited number of samples is available to compute the histogram. In order to have a good approximation of the underlying distribution, we have to choose the number of bins and where the bins should be. In our implementation, the number of bins is specified as a parameter for each filter. When the number of bins is given, we find the average of minimum and maximum filter responses from training images and divide the filter response range into the given number of bins evenly. Given the histogram bin number and bin ranges, a direct implementation of (1) can have a large variance, which can cause a large error for the χ^2-distance between marginal distributions. To overcome this problem, we use Parzen windows [11] to estimate the marginal distribution based on the filter responses, given by

$$H_{\mathbf{W}}^{(\alpha)}(x_1 : x_2) = \frac{\int_{x_1}^{x_2} \sum_{\vec{v}} g\left(x, \mathbf{W}^{(\alpha)}(\vec{v})\right) dx}{\int \sum_{\vec{v}} g\left(x, \mathbf{W}^{(\alpha)}(\vec{v})\right) dx}. \tag{6}$$

Here, g is a kernel function (Gaussian kernel is used in this paper), and x_1 and x_2 are the minimum and maximum values of the given bin. While (6) provides a better estimate of the marginal distribution, it is computationally expensive. We approximate (6) by the following two steps. First we sample the function at the middle value of each bin and then smooth the samples. The smoothing is implemented by applying a small Gaussian kernel for a specified number of times determined by the scale of the Gaussian kernel. This provides a more reliable estimation of the marginal distribution and thus more accurate χ^2-distance between two spectral histograms.

Alternatively, one can adopt a parametric model and then estimate the model parameters from the training samples. A particular two-parameter distribution model [16] provides a good approximation for histograms of filter responses for a variety of real images, which leads to analytical probability models of natural images [37].

III. Texture Classification

We apply spectral histograms and χ^2-statistic as a distance measure to texture classification. Given a database with M texture types, we represent each type m by the average spectral histogram H_{obs^m} of available training samples, defined as

$$H_{\text{obs}^m} = \frac{1}{M^m} \sum_{i=1}^{M^m} H_{W_i^m} \tag{7}$$

where W_i^m is a training sample of texture type m and M^m is the total number of training samples for type m. Because our primary goal is to demonstrate the effectiveness of the spectral histogram as a texture feature, we use a *minimum-distance* classifier for a new sample $\mathbf{W}_i$, given by

$$m_{\mathbf{W}_i}^* = \min_m \chi^2\left(H_{\mathbf{W}_i}, H_{\text{obs}^m}\right). \tag{8}$$

Other classification approaches can also be applied [11] and issues related to the choice of classifiers are not discussed in this paper. In the spectral histogram framework, to measure the gain using a particular set of filters S, we define classification gain as

$$G(S) = \frac{(1 - C_{err}(S))}{(\frac{1}{M})} = M\left(1 - C_{err}(S)\right) \tag{9}$$

where $C_{err}(S)$ is the classification error rate, M is the total number of classes in the database, and thus $1/M$ is the expected correct classification rate based on a random decision. $G(S)$ measures the effectiveness of filters in S more objectively than $C_{err}(S)$ because $C_{err}(S)$ is closely related to M. Here we use

$$B = \{F^{(1)}, F^{(2)}, \cdots, F^{(K)}\}, \quad S = \phi$$

repeat

for each filter $F^{(\alpha)}$ in B

calculate $G_V(S \cup F^{(\alpha)})$

$\alpha^* = \max_\alpha G_V(S \cup F^{(\alpha)})$

$E = G_V(S \cup F^{(\alpha^*)}) - G_V(S)$

if $E > \epsilon$ then

$S = S \cup F^{(\alpha^*)}, \quad B = B \setminus F^{(\alpha^*)}$

until $E \leq \epsilon$

Fig. 3. Filter selection algorithm. Here, B is the subset of the filters that has not been chosen, S is the subset that has been chosen, and ϵ a threshold. Initially B consists of all the available filters and S an empty set.

$G(S)$ to emphasize that the classification gain depends on the filters in S. Because the spectral histogram representation depends critically on the filters used, we present our filter selection algorithm first and then our classification results on a natural texture dataset.

A. Filter Selection

As demonstrated in Fig. 1, the particular set of images that is characterized by a spectral histogram critically depends on the filters used. In one extreme, no filter is used and all images are admitted in the set. In the other extreme, when infinitely many filters are used, only the original image and its translations are admitted. In addition, for a given set of textures, some of the filters are more effective than others. To address these issues, we propose a method that selects a subset of filters from a large filter bank by optimizing the classification performance of training samples [41]. To estimate the classification performance, we divide the available samples into a training set T (used to train the chosen classifier), and a validation set V (used to estimate the performance), known as cross-validation [11]. Specifically, for a given set of filters S, we calculate H_{obs^m} using (7) for each texture class using the samples in T and then we classify the samples in V with (8) and calculate the classification gain $G_V(S)$ using (9). To be computationally efficient, we use a greedy algorithm. In other words, we choose filters one by one so that the next one has the maximum $G_V(S)$ with the ones already chosen. The filter selection algorithm is shown in Fig. 3. The computational complexity of the algorithm depends on the complexity of calculating $G_V(S)$ and K (the number of filters in B). For the minimum distance classifier used in this paper, the complexity is $O(K^2 \times (|T| + M \times |V|))$, where $|T|$ is the time to compute H_{obs^m}, and $M \times |V|$ to compute $G_V(S)$ given H_{obs^m}.

To demonstrate the effectiveness of the filter selection algorithm, we use a texture database that consists of 40 Brodatz texture images, 10 of which are shown in Fig. 4. Initially there are 40 filters. Table I shows the classification gain along with the computation time for classification with respect to the number of filters. Here, the computation time is the time for classifying test images relative to that of using only the intensity filter. For texture classification, it is often desirable to use only derivative filters. The last three columns in Table I show the result without the intensity filter. As Table I shows, the filter selection algorithm essentially chooses the most effective filters. The row with a star shows the filters chosen automatically for optimal classification, whose performances are better than those of all the available filters. This illustrates clearly a key difference between texture classification and texture synthesis. As Fig. 2 shows, more filters clearly give better synthesis results. However, more filters may not improve the classification performance and in fact may give worse performance. In this particular case, while the difference in classification gain between the optimal choice and all the available filters is not significant, the difference in computation time is very significant, which can be critical for some applications.

B. Classification Experiments

We apply our classification method to the 40-texture dataset shown in Fig. 4. This dataset is challenging because there are significant variations within some textures and some of them are very similar to each other. At a given integration scale, we partition the images into nonoverlapping samples, which are then divided into disjoint training and testing set. Here seven filters including the intensity, $Gcos(2, 0^\circ)$, $Gcos(2, 30^\circ)$, $Gcos(3, 30^\circ)$, $Gcos(3, 60^\circ)$, $Gcos(3, 90^\circ)$, and $Gcos(5, 150^\circ)$, are selected automatically by the filter selection algorithm (see Table I) and are used to compute the spectral histogram.

Fig. 5(a) shows the classification gain with respect to the integration scale on the 40-texture dataset using the seven filters. To avoid the bias due to the particular images in the training and test set, we randomly divide the total samples into disjoint training and test set and we repeat the classification experiment 100 times and collect the average, best, and worst performance. Here 1/2 of the available samples are used for for training and the remaining ones for testing. This result shows several important aspects of texture classification. 1) It shows clearly that a reasonable integration scale is needed in order to discriminate textures as the texture structures have certain spatial extent. Here the average classification gain at integration scale 8×8 is 21.34 (corresponding to a classification error of 46.64%), and it improves to 35.86 (corresponding to a classification error of 10.35%) at integration scale 24×24. 2) Given a reasonable intergration scale, the spectral histogram provides a robust feature statistic for classification. The average classification gain at integration scales of 32×32 or larger is better than 37 (corresponding to an error less than 7.5%). If we allow the correct one within the three closest classes, the classification error is less than 1% for all the 100 trials. Given the significant variations within textures and similarities between textures, the performance is sigficantly better than exisiting filter-based methods (see Section IV for comparisons). 3) It shows also that the spectral histogram is not sensitive to particular images in the training and test sets as the best and worst are close to the average of 100 trials. At integration scale 24×24, the average gain is 35.86, the worst 35.50, and the best 36.36.

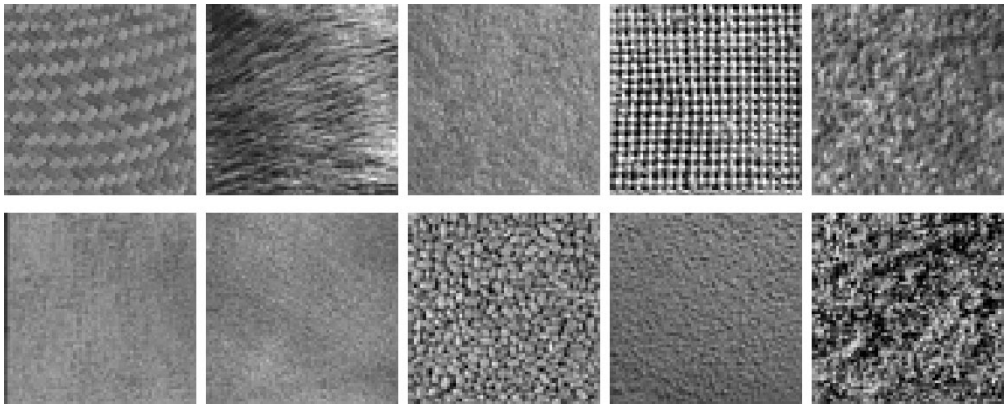

Fig. 4. Ten of the 40 textures used in the classification experiments. The input image size is 256×256. These images are available at http://www-dbv.cs.uni-bonn.de/image/texture.tar.gz.

TABLE I
CLASSIFICATION GAIN AND COMPUTATION TIME WITH RESPECT TO FILTERS CHOSEN BY THE FILTER SELECTION ALGORITHM

# of filters	Classification gain	Computation time	# of filters (no intensity)	Classification gain	Computation time
1	23.63	1.00	1	22.56	1.26
2	31.41	3.69	2	30.86	3.92
3	34.19	8.59	3	32.22	8.72
5	35.72	16.16	5	35.44	18.45
7*	36.84	32.30	6*	36.68	39.31
40	36.69	690.86	39	35.13	690.22

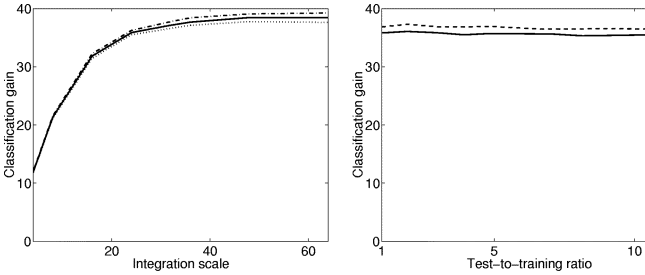

Fig. 5. Classification result for the 40-texture dataset. (a) The average (solid), best (dash-dotted) and worst (dotted) classification gain of 100 trials with randomly divided training and test set with respect to the integration scale. (b) The classification gain with respect to test-to-training sample ratio. Solid line—integration scale 32×32; dashed line—integration scale 24×24.

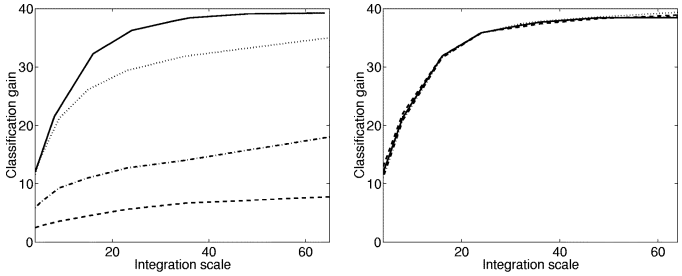

Fig. 6. Comparison of different features and distance measures of the 40 texture dataset. (a) Classification gain for different features. Dashed line—intensity mean; dash-dotted line—intensity mean and variance; dotted line—intensity histogram; solid line—spectral histogram of the seven filters. (b) Classification gain for commonly used distance measures for histograms. Solid line—χ^2-statistic; dotted line—L_1-norm; dashed line—L_2-norm; dash-dotted—Kullback-Leibler divergence.

To demonstrate the generalization capability of the spectral histogram, Fig. 5(b) shows the classification gain at two integration scales with respect to the test-to-training sample ratio. In both cases, the classification gain does not change much for ratios between 1:1 and 12:1. This confirms the generalization capability of spectral histograms in characterizing texture images.

To provide numerical justifications of the proposed representation, we have compared the spectral histogram with other commonly used features and distance measures. Fig. 6(a) shows the classification gain for features commonly used for intensity images, including the mean value, combination of mean and variance values, and intensity histogram. As we can see from Fig. 6(a), the mean and Gaussian models are not sufficient for characterizing those images and generate worst results. The comparison shows that the smoothing stage commonly used in texture classification methods [34] is not optimal; the distribu-

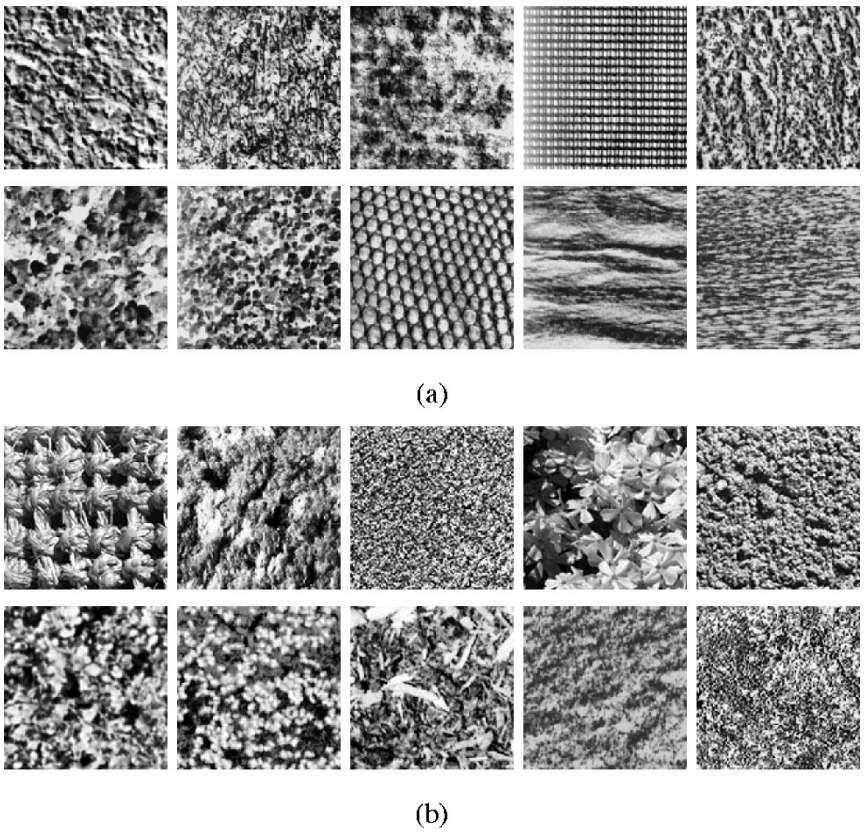

Fig. 7. Ten-texture image groups used in [34]. The image size is 128 × 128. (a) The images in Fig. 11(h) of [34]. (b) The images in Fig. 11(i) of [34].

tion of local features is far more effective in discriminating textures. This is also consistent with the comparisons shown in the next section. Fig. 6(b) compares several commonly used distance measures for histograms, including L_1 norm, L_2 norm, Kullback-Leibler divergence, and χ^2-statistic. For texture classification using spectral histograms, Fig. 6(b) shows that different measures give very similar results, suggesting that spectral histograms is insensitive to a particular form of distance measure.

IV. Comparison With Existing Approaches

Several comparative studies about texture features have been conducted. Ohanian and Dubes [31] studied the performance of various texture features, including fractal features, cooccurrence features, Markov random field features, and Gabor features. However, the evaluation was done only on four classes of images and the conclusion may not be generalized. Ojala *et al.* [32] did a similar study on joint occurrences of feature pairs using nine texture images and the ones in [31]. Recently, Randen and Husoy [34] did an extensive comparative study for texture classification on cooccurrence methods, Law's texture measures, different filtering-based methods, and a neural network approach [19]. They used a supervised classifier of Kohonen [23] for most of their experiments. The filter responses at each pixel form a vector and the texture classification is to classify feature vectors. Because filters have a spatial extent, the receptive field of a vector overlaps heavily with the neighboring ones. We have applied our method to the same images with the same experimental settings. We use an integration scale 32×32 and the filters are chosen automatically from the 40 filters using the filter selection algorithm. We use a separate set of images for training and a separate set of images for testing as in [34].[3] The results for the two most challenging groups of texture images, shown in Fig. 7(a) and (b), are summarized in Table II, where the average performance and the best in tables 3, 6, 8, and 9 in [34] are shown. For these two groups, due to the inhomogeneity and large variations, texture types in local windows given by the integration scale are perceptually close and they require a very accurate texture model for classification. In addition, separate images are used for training and this creates additional difficulties for methods that cannot generalize well to new data. The classification gains of all the methods studied in [34] are shown in Fig. 8(a) and (b). Our method is significantly better than the best performance in [34]. Furthermore, the most errors of our method are from the texture pairs that are perceptually very similar. If we consider the two closest textures as correct, our method gives a classification gain of 9.50 and 9.58 respectively, corresponding to 95.0% and 95.8% correct classification rate. This comparison clearly suggests that classification based on filtering output is inadequate for characterizing texture appearance and an integration after filtering must be done. Our comparison strongly indicates that some representation like the

[3]In [34], windows that include different texture types are used to test texture boundary accuracy of classification methods. Due to the uncertainty principle [5], there is an intrinsic tradeoff between classification performance and boundary accuracy. Since a considerable integration scale is needed to characterize a texture (see Figs. 5 and 6), test windows are confined within a single texture here. With spectral histogram representations, the boundary localization is achieved by building a more accurate probability model after initial classification [28].

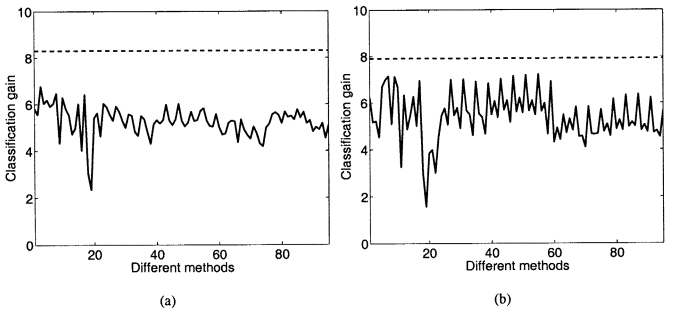

Fig. 8. The classification gain for all the methods in [34] for Fig. 7(a) and (b) respectively. In each plot, each data point represents one result in tables 3, 6, 8, and 9 of [34] and the dashed line is the result of the proposed classification method.

TABLE II
CLASSIFICATION GAINS OF METHODS SHOWN IN [34] AND THE PROPOSED METHOD

Texture	Methods in [34]		Proposed method	
group	Average	Best	First one correct	First two correct
Fig. 7(a)	5.26	6.77	8.31	9.50
Fig. 7(b)	5.40	7.22	7.91	9.58

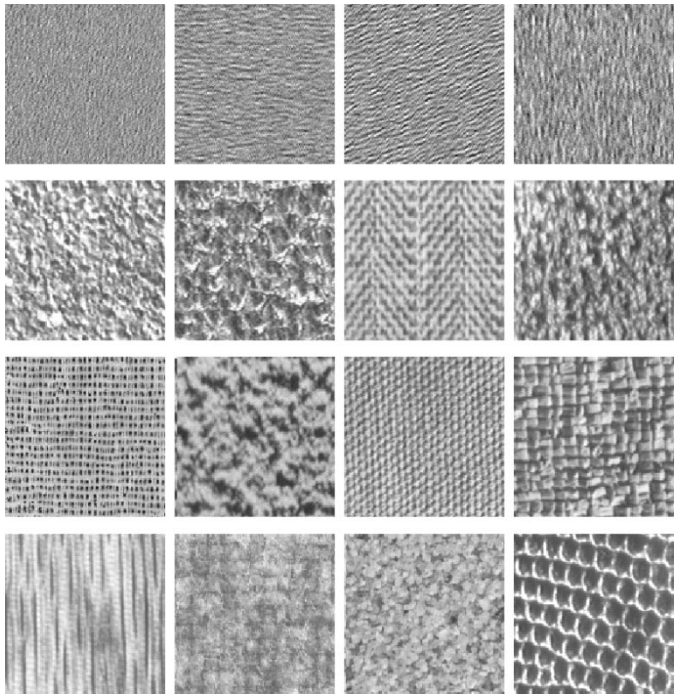

Fig. 9. Sixteen texture images used in [1]. Images in the first row are generated from Gaussian random fields, and remaining rows are from the Brodatz album. The image size is 128×128.

spectral histogram may be necessary in order to capture complex texture appearance.

To further illustrate our method, we have done a comparison with a method proposed by Azencott *et al.* [1]. In [1], a texture feature vector is proposed based on the spectral density of windowed Fourier filters, e.g., Gabor filters, and a distance between two textures is defined as a symmetrized Kullback distance between computed vectors. A minimum distance classifier is also used for texture classification. For an unbiased comparison, we use the same settings used in [1]. Each input texture image with the size of 128×128 is divided in 49 image patches with size 32×32 and thus adjacent patches are overlapped. We use the same seven filters as in the previous section to compute spectral histogram. The 16 texture images used in [1] are shown in Fig. 9; therefore there are 784 image patches in total.

Two classification experiments were reported in [1]. In the first experiment, the 49 patches of each image were divided into a training set of 21 patches and a test set of 28 patches. The result in [1] gives six misclassified patches, i.e., 1.34% classification error. For the same setting, our method gives only 1 misclassified patch, resulting in 0.22% classification error. In the second experiment, the training set is reduced to one patch per texture image. The result in [1] using the Kullback distance gives twenty-three misclassified patches. Our result gives only four misclassified patches. This comparison demonstrates the superior discrimination ability of the spectral histogram.

V. DISCUSSION

A. Relations to Existing Approaches

This paper focuses on texture classification using spectral histograms with a fixed set of filters. As we mentioned earlier, one can choose different filters to define different features. In this section, we point out the relations between the spectral histogram and other existing methods.

Before we discuss specific features for textures, we point out that uniform regions are simply a special case under the spectral histogram, thus the spectral histogram provides a unified feature for texture as well as nontexture images. However, textures are often studied separately from intensity images and texture features from other approaches may not be applicable to uniform images [6].

Texture analysis has been studied extensively and many methods have been proposed. Tuceryan and Jain [40] classified existing approaches into four categories, namely statistical methods, geometrical methods, model based methods, and signal processing methods. We discuss the relationships between each category and our proposed method.

Statistical methods, including cooccurrence matrices [17], autocorrelation features [40], and our method here, are based on the observation that a texture is defined by the spatial distribution of gray values. A cooccurrence matrix consists of the number of occurrences of a gray level pair with a specified distance apart. This can be viewed as a special case of k-gon statistics proposed by Julesz [20], [22]. Because the cooccurrence matrix cannot be used directly as texture features, a number of texture features were subsequently computed from the cooccurrence matrix [17]. It is easy to see that the cooccurrence matrix can also be defined as responses of a specifically designed gradient filter and thus a spectral histogram using gradient filters provides cooccurrence matrix features.

The class of geometrical methods is based on the assumption that a texture consists of repeated texture elements, such as the one shown Fig. 2(b). After the texture elements are identified,

geometrical properties of the element distribution can be used to characterize textures [40]. As shown in Fig. 2(b), the spectral histogram can characterize the texture element as well as its distribution without knowing the texture element. This provides a more elegant way to characterize textures with repeated texture elements.

Model based methods include Markov random fields [6], [7], [9], [15], [38], [44]. This class of methods can not only describe the texture through model parameters, which are learned from observed textures, but also synthesize it through sampling. In [44], for example, Zhu *et al.* proposed a FRAME model and showed that the model provides a unified framework for Markov random field models. In a limiting case, Wu *et al.* [42] proved the equivalence of a model specified by features such as spectral histogram [43] and a Gibbs model, a special case of which is shown in [12]. This relation shows that the spectral histogram provides an equivalent way of specifying a Markov random field, which avoids the parameter learning necessary for a Markov random field model.

Signal processing methods try to characterize textures by filter responses directly. Many of these models have been studied and compared in [34], including Laws filters, ring and wedge filters, Gabor filters, wavelet transforms, packets, frames, discrete cosine transforms, quadrature mirror filters, and a number of optimized filters for texture classification (see the references wherein). Even though the filters in many of those approaches were carefully designed and chosen, our comparison shows that this class of methods is inadequate to characterize and discriminate texture structures. This demonstrates that an integration of different filter responses such as the spectral histogram proposed here, is probably necessary while the specific form of filters is not critical [24].

B. Integration of Filter Responses

It is easy to see that a filter's response is inhomogeneous even to a homogeneous texture image. An inevitable issue common to all filter-based approaches is to form a feature which characterizes a texture region. To reduce the inhomogeneity of filter responses, spatial smoothing is commonly used [3], [29], [34]. The proposed spectral histogram model resolves this issue using histograms of filter responses within a spatial window. For a spatial window substantially larger than the size of basic elements in a texture, the spectral histogram is intrinsically insensitive to precise locations of texture elements. This is consistent with a study on human texture discrimination [21]. Because of this property, two images do not need to be aligned in order to be compared using spectral histograms. More importantly, because of the stochastic nature of textures, images of the same texture type may not be aligned, an example of which is shown in Fig. 2(b). While both images in Fig. 2(b) consist of crosses with similar distribution, two images cannot be aligned under simple transforms. The misalignment of textures can be a serious problem for approaches that use filter responses directly as features for texture classification, such as those studied in [34].

Note that the spectral histogram is defined on any type of images. Piece-wise constant images with additive Gaussian noise are a special case whose spectral histogram has a unique pattern. Under the spectral histogram representation, the distinction between texture and nontexture images becomes unnecessary. While the spectral histogram here is used primarily for textures with roughly repeated patterns, our study elsewhere suggests that the spectral histogram can also be applied to classify faces and 3-D objects [25], [26], consistent with a recent study on object recognition using multidimensional histograms [36].

VI. Conclusion

We have demonstrated that the spectral histogram provides a sufficient feature statistic for texture classification. The χ^2-statistic between spectral histograms provides a robust distance measure for comparing textures. We have proposed a filter selection algorithm for texture classification. With a wide range of integration scales and test-to-training ratios, we have obtained satisfactory classification results on natural texture datasets. Our comparison shows that the spectral histogram improves the classification performance significantly. By pointing out the relations between existing texture features and the spectral histogram, we suggest that the latter may provide a unified image feature statistic.

Acknowledgment

The authors would like to thank the anonymous reviewers whose comments have improved the presentation of this paper significantly and acknowledge the reviewer who suggested the cross-validation technique used in the present filter selection algorithm.

References

[1] R. Azencott, J. P. Wang, and L. Younes, "Texture classification using windowed Fourier filters," *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 19, no. 2, pp. 148–153, 1997.

[2] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, pp. 3327–3338, 1997.

[3] J. R. Bergen and E. H. Adelson, "Early vision and texture perception," *Nature*, vol. 333, pp. 363–364, 1988.

[4] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *J. Physiol. (Lond.)*, vol. 197, pp. 551–566, 1968.

[5] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[6] E. Cesmeli and D. L. Wang, "Texture segmentation using Gaussian-Markov random fields and neural oscillator networks," *IEEE Trans. Neural Networks*, vol. 12, pp. 394–404, 2001.

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

[8] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, no. 7, pp. 23–26, 1985.

[9] J. L. Davidson, N. Cressie, and X. Hua, "Texture synthesis and pattern recognition for partially ordered Markov models," *Pattern Recognit.*, vol. 32, no. 9, pp. 1475–1505, 1999.

[10] R. L. De Valois and K. K. De Valois, *Spatial Vision*. New York: Oxford Univ. Press, 1988.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[12] I. M. Elfadel and R. W. Picard, "Gibbs random fields, cooccurrences, and texture models," *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 16, no. 1, pp. 24–37, 1994.

[13] D. Gabor, "Theory of communication," *J. IEE (Lond.)*, vol. 93, pp. 429–457, 1946.

[14] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[15] G. L. Gimel'farb, *Image Textures and Gibbs Random Fields*. Dordrecht, The Netherlands: Kluwer, 1999.

[16] U. Grenander and A. Srivastava, "Probability models for clutter in natural images," *IEEE Trans. Pattern Anal. Machine Intell.*, submitted for publication.
[17] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786–804, 1979.
[18] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. SIGGRAPHS*, 1995, pp. 229–238.
[19] A. K. Jain and K. Karu, "Learning texture discrimination masks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 2, pp. 195–205, 1996.
[20] B. Julesz, "Visual pattern discrimination," *IRE Trans. Inform. Theory*, vol. 8, pp. 84–92, 1962.
[21] ——, "Textons, the elements of texture perception and their interactions," *Nature*, vol. 290, pp. 91–97, 1981.
[22] ——, *Dialogues on Perception*. Cambridge, MA: MIT Press, 1995.
[23] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, 1990.
[24] X. Liu, "Computational Investigation of Feature Extraction and Image Organization," Ph.D., The Ohio State Univ., Columbus, 1999.
[25] X. Liu and L. Cheng, "Independent spectral representation of images for recognition," J. Opt. Soc. Amer. A, 2003.
[26] X. Liu and A. Srivastava, "3D object recognition using perceptual components," in *Proc. Int. Joint Conf. Neural Networks*, vol. 1, 2001, pp. 553–558.
[27] X. Liu and D. L. Wang, "A spectral histogram model for textonn modeling and texture discrimination," *Vis. Res.*, vol. 42, no. 23, pp. 2617–2634, 2002.
[28] X. Liu, D. L. Wang, and A. Srivastava, "Image segmentation using local spectral histograms," in *Proc. Int. Conf. Image Processing*, 2001.
[29] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Opt. Soc. Amer. A*, vol. 7, no. 5, pp. 923–932, 1990.
[30] A. Mojsilovic, M. V. Popovic, and D. M. Rackov, "On the selection of an optimal wavelet basis for texture classification," *IEEE Trans. Image Processing*, vol. 9, pp. 2043–2050, Dec. 2000.
[31] P. P. Ohanian and R. C. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognit.*, vol. 25, no. 8, pp. 819–833, 1992.
[32] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.
[33] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.
[34] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 21, no. 4, pp. 291–310, 1999.
[35] T. R. Reed and J. M. H. Du Buf, "A review of recent texture segmentation and feature extraction techniques," *Comput. Vis., Graph., Image Process.: Image Understand.*, vol. 57, no. 3, pp. 359–372, 1993.
[36] B. Schiele and J. L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms," *Int. J. Comput. Vis.*, vol. 36, no. 1, pp. 31–50, 2000.
[37] A. Srivastava, X. Liu, and U. Grenander, "Universal analysis forms for modeling image probabilities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, 2002.
[38] P. H. Suen and G. Healey, "Modeling and classifying color textures using random fields in a random environment," *Pattern Recognit.*, vol. 32, no. 6, pp. 1009–1017, 1999.
[39] M. Tuceryan and A. K. Jain, "Texture segmentation using Voronoi polygons," *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 12, pp. 211–216, 1990.
[40] ——, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds, Singapore: World Scientific, 1993, pp. 235–276.
[41] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York: Springer-Verlag, 2000.
[42] Y. N. Wu, S. C. Zhu, and X. Liu, "Equivalence of Julesz ensembles and FRAME models," *Int. J. Comput. Vis.*, vol. 38, no. 3, pp. 245–261, 2000.
[43] S. C. Zhu, X. Liu, and Y. Wu, "Exploring texture ensembles by efficient Markov chain Monte Carlo," *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 22, pp. 554–569, 2000.
[44] S. C. Zhu, Y. N. Wu, and D. B. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Comput.*, vol. 9, no. 8, pp. 1627–1660, 1997.

Xiuwen Liu (SM'97) received the B.Eng. degree in computer science in 1989 from Tsinghua University, Beijing, China, and the M.S. degrees in geodetic science and surveying in 1995 and computer and information science in 1996, and the Ph.D. degree in computer and information science in 1999 all from The Ohio State University, Columbus.

From August 1989 to February 1993, he was with Department of Computer Science and Technology, Tsinghua University. Since 2000, he has been with the Department of Computer Science of the Florida State University, Tallahassee. His current research interests include low dimensional representations of images, statistical pattern recognition, manifold-based optimization and inference algorithms, and computational modeling of visual perception and inference.

DeLiang Wang (SM'96) received the B.S. degree in 1983 and the M.S. degree in 1986 from Peking (Beijing) University, Beijing, China, and the Ph.D. degree in 1991 from the University of Southern California, Los Angeles, all in computer science.

From July 1986 to December 1987, he was with the Institute of Computing Technology, Academia Sinica, Beijing. Since 1991, he has been with the Department of Computer and Information Science and the Center for Cognitive Science at The Ohio State University, Columbus, where he is currently a Professor. From October 1998 to September 1999, he was a Visiting Scholar in the Vision Sciences Laboratory at Harvard University, Cambridge, MA. His present research interests include machine perception, neurodynamics, and computational neuroscience.

Dr. Wang is a member of IEEE Neural Networks and Signal Processing Societies and the International Neural Network Society. He is a recipient of the 1996 U.S. Office of Naval Research Young Investigator Award.