Texture Classification Using Spectral Histograms

Xiuwen Liu and DeLiang Wang
Department of Computer and Information Science
The Ohio State University
2015 Neil Avenue, 395 Dreese Lab.
Columbus, OH 43210-1277
USA

Tel: (614) 292-7402 Fax: (614) 292-2911
Email: \{liux,dwang\}@cis.ohio-state.edu
http://www.cis.ohio-state.edu/~liux
http://www.cis.ohio-state.edu/~dwang
Texture Classification Using Spectral Histograms

Xiuwen Liu and DeLiang Wang*
Department of Computer and Information Science
Center for Cognitive Science
The Ohio State University, Columbus, OH 43210, USA
{liux, dwang}@cis.ohio-state.edu

Abstract — Based on a local spatial/frequency representation, we propose a spectral histogram as a feature statistic for characterizing texture appearance. The spectral histogram consists of marginal distributions of responses of a bank of filters and encodes implicitly the structure of images. The distance between two spectral histograms is measured using \( \chi^2 \)-statistic. The spectral histogram with the associated distance measure exhibits several properties that are necessary for texture discrimination and classification. The spectral histogram provides a generic feature for texture as well as non-texture images, where the uniform image is a special case with a unique pattern. The spectral histogram is a nonlinear operator, consistent with the nonlinearity in human perception. Our classification experiments reveal that it can generalize well even with a small number of training samples and the classification result does not depend on a particular form of distance measure. We have obtained very good results on natural images. Comparison shows that our method produces a marked improvement in classification performance, which suggests a significant advance in texture classification. We also provide a qualitative explanation on the unsatisfactory performance of other approaches. Finally we point out the relationships between existing texture features and the spectral histogram, showing that the latter provides a unified texture feature.

Index Terms — Spectral histogram, Texture classification, Texture analysis, Filtering, Spatial/frequency representation.

1 Introduction

Texture analysis is a fundamental problem in computer vision and computer graphics with a wide variety of applications from realistic texture synthesis, image understanding [32], to, more recently, querying by image content [36]. A critical issue in texture analysis and modeling is how to characterize textural structure and define a perceptually meaningful distance/similarity measure between textures, which remains elusive despite considerable...
efforts in the literature [27]. Features are in general proposed based on assumptions for mathematical convenience or task-specific heuristics (see [32, 28] for recent reviews). Because there is no obvious feature common for all texture images, often features are derived for a specific class of texture images. For textures with periodic structures, geometric properties based on the texture elements are often used [31]. Early features including cooccurrence matrices [14] and Markov random field models [6] have limited expressive power because the analysis of spatial interaction is limited to a relatively small neighborhood.

There are two major reasons for the difficulty in texture classification. The first is the lack of an explicit model for texture. Most texture classification models are based on a cascade of filters that transforms a texture image into some classification score. But the adequacy of the filters for characterizing various textures is rarely checked. The second reason is the lack of an effective measure for comparing two texture patches that is consistent with human texture discrimination.

Motivated by the research on human texture perception [4] and subsequent modeling work [15, 37, 38], we propose a local spectral histogram, consisting of marginal distributions of responses from a bank of filters for an image patch, as a generic feature statistic for image organization. In particular, the spectral histogram provides an explicit model for texture and extensive simulations suggest that texture images sharing the same spectral histogram are perceptually similar [38, 21]. Furthermore, we define a perceptual distance between two image patches as the $\chi^2$-statistic of spectral histograms, which exhibits nonlinearity and asymmetry that is consistent with the human texture perception [21]. Our work presented elsewhere [21] demonstrates that this proposed model provides a satisfactory account for a systematic set of human texture discrimination data. This paper focuses on the problem of texture classification. We report that the spectral histogram produces very good classification results. A systematic comparison with other methods documented in [27] demonstrates that our approach yields far better results.

2 Local Spectral Histogram

In the 1980s, significant advances in visual perception were made [3, 8], and revealed that the human visual system transforms a retinal image into a local spatial/frequency representation, 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 [11]. Recently, this theory has also been confirmed by deriving similar feature detectors from natural images based on certain optimization criteria [25]. These advances have inspired much research in texture classification and segmentation. Within this framework, however, statistic features still need to be chosen because filter responses may not be homogeneous within homogeneous textures and may not be sufficient. Intuitively, texture appearance can not be characterized by very local pixel values because texture is a regional property. If one wants to define a homogeneous feature within a texture region, it is necessary to integrate responses from filters with multiple orientations and scales. For example, Unser [33] used variances from different filters to characterize textures.

Recently, Heeger and Bergen [15] proposed a texture synthesis algorithm that can match
texture appearance. The algorithm tries to transform a random noise image into an image with similar appearance to a given target image by matching independently the histograms of image pyramids constructed from the random and target images. Zhu et al. [37] proposed a theory for learning probability models by matching histograms based on a maximum entropy principle. Based on the local spatial/frequency representation and inspired by the research on texture synthesis, we define a feature - spectral histogram - to capture the perceptual appearance of textures.

2.1 Definition

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

$$H^{(\alpha)}_W(z) = \frac{1}{|W|} \sum_{\vec{v}} \delta(z - W^{(\alpha)}(\vec{v})).$$

(1)

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

$$H_W = (H^{(1)}_W, H^{(2)}_W, \ldots, H^{(K)}_W).$$

(2)

The spectral histogram of an image or an image patch 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 or the input image patch is called an integration scale. Figure 4 shows the spectral histogram of several texture images. Because the marginal distribution of each filter response is a distribution, a similarity measure can be defined as $\chi^2$-statistic, which is a first-order approximation of the Kullback-Leibler divergence and used widely to compare two histograms $H_{W_1}$ and $H_{W_2}$

$$\chi^2(H_{W_1}, H_{W_2}) = \sum_{\alpha=1}^{K} \sum_z \frac{(H^{(\alpha)}_{W_1}(z) - H^{(\alpha)}_{W_2}(z))^2}{H^{(\alpha)}_{W_1}(z) + H^{(\alpha)}_{W_2}(z)}.$$

(3)

2.2 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.

**Proposition 1** The 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 $W$ does not affect its spectral histogram. This is essential for any texture model to characterize texture appearance.

---

\(^1\)We require that $\sum_{\vec{u}} |F^{(\alpha)}(\vec{u})| > 0$ for any $\alpha$. 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.
**Proposition 2** The spectral histogram is a nonlinear operator.

The nonlinearity is because the histogram operation given in (1) is nonlinear. Let $I$ be a nonzero uniform image, i.e., $I(u) = c$ for all $u$, where $c$ is a nonzero constant. Let $I = \beta I + (1 - \beta)I = I_1 + I_2$, where $0 < \beta < 1$. $I^{(\alpha)}(v) = F^{(\alpha)} * I(v) = \sum_u F^{(\alpha)}(u)c = c_1$ for all $v$. Because of the linear convolution, we have $I_1^{(\alpha)}(v) = \beta c_1$ and $I_2^{(\alpha)}(v) = (1 - \beta)c_1$. Thus we have $H_1^{(\alpha)}(z) = \delta(z - c_1)$, $H_{I_1}^{(\alpha)}(z) = \delta(z - \beta c_1)$, and $H_{I_2}^{(\alpha)}(z) = \delta(z - (1 - \beta)c_1)$. Because $c_1$ is not zero and $0 < \beta < 1$, we have $H_{I_1}^{(\alpha)} \neq H_{I_1}^{(\alpha)} + H_{I_2}^{(\alpha)}$.

In general, a uniform image can be decomposed into images with structures. Figure 1(b) and (c) show one such decomposition, where the summation of Figure 1(b) and (c) gives Figure 1(a), a uniform image. For non-constant filters, it is easy to see that the nonlinearity of the spectral histogram can also be caused by the dependency among pixel values, or texture structures. This example shows that any linear operator is inadequate for capturing texture structures.

The nonlinearity of human texture discrimination was demonstrated by Williams and Julesz [34]. The ultimate goal of texture classification is to classify textures in a way that is consistent with human perception. A critical question is whether the nonlinearity of an operator is consistent with human texture discrimination. We have shown that the spectral histogram can match existing psychophysical data well [21]. This also justifies why our classification results shown in Sections 3 and 4 are far better than many existing approaches.

![Figure 1](image)

(a) (b) (c)

Figure 1: A uniform image can be decomposed into images with structures. Here (a) is the summation of (b) and (c). Black pixels are 0 and white pixels 255.

**Proposition 3** With sufficient filters, the spectral histogram can uniquely represent any image up to a translation.

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

---

A normalization is needed when two spectral histograms are summed together, as shown in (1).
optimal response of the filter is bounded

\[
\begin{align*}
I^{(2)}(\vec{v}) &= F^{(2)} \cdot I(\vec{v}) \\
&= \sum_{\vec{u}} F^{(2)}(\vec{u}) I(\vec{v} - \vec{u}) \\
&\leq \sqrt{\sum_{\vec{u}} (F^{(2)}(\vec{u}))^2} \sqrt{\sum_{\vec{u}} (I(\vec{v} - \vec{u}))^2} \\
&= \sqrt{\sum_{\vec{u}} (I^{(2)}(\vec{u}))^2}
\end{align*}
\]

due to Cauchy-Schwartz’ inequality. The optimal is achieved when \(\vec{v} = \vec{v}_0\). Similarly, if \(J\) is a permutation of \(I\) and \(H^{(2)}_J = H^{(2)}_I\), \(J\) must be equivalent to \(I\) up to a translation to achieve the same optimal response. □

In practice, however, one cannot use the image itself as a filter, which requires infinite number of filters. For a given bank of filters, each filter provides some constraints on the images that can satisfy \(H_I = H_{\text{obs}}\). With more constraints imposed by more filters, the solution to the equation \(H_I = H_{\text{obs}}\) would converge to the observed image up to a translation. With this proposition, the spectral histogram provides a general framework to represent and code images, and textures in particular. Traditional multiscale analysis can be applied simply by choosing filters from a family with different scales. This paper does not address issues related to filter selection as a fixed set of filters is used for all experiments. A filter selection procedure similar to [37] can be implemented when a set of textures to be classified is given.

A coding scheme based on the spectral histogram may also provide a solution to a fundamental problem in image coding. Humans encode both uniform and white noise images very efficiently, while images with structures are the most difficult to encode. By using the spectral histogram, both uniform and white noise images can be encoded efficiently with no perceptual difference in the decoded images. Here the decoded images are typical samples that satisfy \(H_I = H_{\text{obs}}\). Figure 2(a) shows a white noise image and Figure 2(b) and (c) are two decoded images that satisfy \(H_I = H_{\text{obs}}\), where only the intensity filter is used. Even though Figure 2(b) and (c) are very different from Figure 2(a) according to PSNR (see equation (9)), there is no perceptual difference among them. This may lead to a new coding method which is more perceptually meaningful.

![Figure 2: A white noise image and two decoded images that satisfy $H_w = H_{\text{obs}}$. Here only the intensity filter is used. a) A white noise image. (b) and (c) Decoded images.](image)

**Proposition 4** All the images sharing the same spectral histogram define an equivalent class.

Essentially, spectral histograms divide all the images into equivalent classes [38]. Extensive simulations suggest that the spectral histogram is sufficient in characterizing texture
appearance [38, 21] 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 Figure 3 shows four texture images, whose spectral histograms are shown in Figure 4, and the bottom row shows corresponding typical images by satisfying the constraints $H_I = H_{obs}$, where $I$ is an image, $H_I$ its spectral histogram, and $H_{obs}$ the spectral histogram of the observed image. Due to the high dimensionality of $I$, the constraints have to be satisfied through stochastic simulation because traditional searching methods are computationally not feasible. These examples shown in Figure 3 were generated using a Gibbs sampler [12, 38]. In Figure 3(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. Figure 3(b) shows a synthetic texture image, where the spectral histogram captures the texture element and its density. Figure 3(c) and (d) show that the spectral histograms of two stochastic textures capture their perceptual appearance well.

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 [2, 22, 27]. 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 [18]. 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 Figure 3(b). While both images in Figure 3(b) consist of crosses with similar distribution, two images cannot be aligned under some 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 [27].

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 non-texture images becomes unnecessary. While the spectral histogram here is used primarily for textures with roughly repeated patterns, our informal study suggests that the spectral histogram can also be applied to classify faces and hand-written numbers, consistent with a recent study on object recognition using multidimensional histograms [29].

2.3 Choice of Filters

A spectral histogram is defined with respect to a bank of filters and an obvious question is what filters should be used. We use four different types of filters suggested from the studies of visual perception.

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:
Figure 3: 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. The corresponding spectral histograms are shown in Figure 4. (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.

\[ D_x = C \cdot \begin{bmatrix} 0.0 & -1.0 & 1.0 \end{bmatrix} \]

\[ D_y = C \cdot \begin{bmatrix} 0.0 \\ -1.0 \\ 1.0 \end{bmatrix} \]

\[ D_{xx} = C \cdot \begin{bmatrix} -1.0 & 2.0 & -1.0 \end{bmatrix} \]

\[ D_{yy} = C \cdot \begin{bmatrix} -1.0 \\ 2.0 \\ -1.0 \end{bmatrix} \]

Here \( C \) is a normalization constant.

3. Laplacian of Gaussian filters:

\[ \text{LoG}(x, y | T) = C \cdot (x^2 + y^2 - T^2)e^{-\frac{x^2+y^2}{2T^2}}, \]  \hfill (4)

where \( C \) is a constant for normalization and \( T = \sqrt{2}\sigma \) determines the scale of the filter and \( \sigma \) is the variance of the Gaussian function. These filters are referred to as \( \text{LoG}(T) \).

4. The Gabor filters with both sine and cosine components:

\[ Gabor(x, y | T, \theta) = \frac{C}{2\pi T^2} \left( e^{-\frac{1}{2\pi^2} \frac{1}{T^2} (x \cos \theta + y \sin \theta)^2 + (-x \cos \theta - y \sin \theta)^2} \right) \]

\[ e^{-i \frac{2\pi}{T} (x \cos \theta - y \sin \theta)}. \]  \hfill (5)

Here \( C \) is a normalization constant and \( T \) is a scale. The cosine and sine components of these filters are referred to as \( G\cos(T, \theta) \) and \( G\sin(T, \theta) \) respectively.
Throughout our experiments, we do not change filters. We use eight filters to compute spectral histograms for texture classification:

- The intensity filter.
- Two gradient filters $D_{xx}$ and $D_{yy}$.
- Two LoG filters with $T = 2$ and 5.
- Three cosine Gabor filters with $T = 6$ and three orientations $\theta = 30^\circ$, $90^\circ$, and $150^\circ$.

This set of filters captures the isotropic as well as oriented structures and gives good classification results empirically.

3 Texture Classification

As described above, the spectral histogram of a local window in an image provides a feature that captures the perceptual appearance of the image, which can be a uniform gray value image, a deterministic texture image with individual elements, or a stochastic texture image. The remainder of this paper focuses on classification with given training samples.

3.1 Classification Experiments

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

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

where $W_i^m$ is a training sample of texture type $m$ and $M^m$ is the total number of training samples for type $m$. As demonstrated later, the number of training samples is not critical for spectral histograms due to its descriptive capability for texture and grey-level images.

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 $W_i$, given by

$$m_{W_i}^* = \min_m \chi^2(H_{W_i}, H_{\text{obs}}^m).$$

(6)

Other classification approaches can also be applied [9] and issues related to the choice of classifier are not discussed in this paper.

We use a texture database that consists of 40 Brodatz texture images, shown in Figure 5. At a given integration scale, we partition the images into non-overlapping samples, which are divided into disjoint training and testing sets.

Our classification result at integration scale $35 \times 35$ is given in Figure 6, which shows the misclassified samples with 1:1 test-to-training ratio. The classification error is 4.17%, which is mostly due to the inhomogeneity in the images. For texture type $m$, the relative inhomogeneity is measured by

$$\rho^m = \max_i \chi^2(H_{W_i}^m, H_{\text{obs}}^m)/\min_{j \neq m} \chi^2(H_{\text{obs}}^m, H_{\text{obs}}^j),$$

(7)

where $W_i^m$ is a training sample for class $m$. As shown by the dotted line in Figure 6, $\rho^m$ measures the inhomogeneity in images, and it predicts the classification error well. When $\rho^m < 1$, the classification result tends to be very good. Large error occurs when $\rho^m > 1$.

To measure the gain using a feature statistic, we define a classification gain as

$$G = (1 - C_{\text{err}})/(1/M) = M(1 - C_{\text{err}}),$$

(8)

where $C_{\text{err}}$ is the classification error rate, and $M$ is the total number of classes in the database. Here $1/M$ is the expected correct classification rate based on a random decision. $G$ measures the effectiveness of a feature statistic more objectively than $C_{\text{err}}$ because $C_{\text{err}}$ is closely related to $M$.

To demonstrate the generalization capability of the spectral histogram, Figure 7(a) shows the classification gain at two integration scales with respect to test-to-training sample ratio. In both cases, the classification gain does not change much for ratios between 1:1 and 10:1. This confirms the power of spectral histograms in characterizing texture images.

3.2 Comparison of Image Features

We have compared the spectral histogram with other commonly used features. Figure 7(b) 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
Figure 5: Forty textures used in the classification experiments. The input image size is $256 \times 256$. These images are available at http://www-db.cs.uni-bonn.de/image/texture.tar.gz.
Figure 6: Classification result for the 40-image database. For each image, solid line shows the number of misclassified samples out of a total of 24 testing samples, the dotted line shows $\rho^m$, and the dashed line indicates threshold 1.

see from Figure 7(b), the mean and Gaussian models are not sufficient for characterizing those images and they generate worst results. This suggests that smoothing algorithms (e.g., [26]) for reducing noise are not appropriate for texture classification and segmentation and should be avoided.

Figure 7(c) compares different features for texture images within the framework of the spectral histogram and shows that one type of filters, even widely Gabor filters, does not consistently work well for all the images. This is consistent with a recent study on different filters [27], where no conclusion about filter optimality could be drawn. But the spectral histogram of combined filters gives the best performance through all integration scales from a window as small as $5 \times 5$ to a window as large as $65 \times 65$. By comparing Figures 7(b) and 7(c), we conclude that the distribution of local features is more important than particular forms of local features.

Figure 7(d) compares different distance measures that are widely used to compare distributions and histograms, including $L_1$ norm, $L_2$ norm, Kullback-Leibler divergence, and $\chi^2$-statistic. For image classification using spectral histograms, Figure 7(d) shows clearly that different measures give very similar results, demonstrating that the spectral histogram is robust and does not rely on a particular form of distance measure.

4 Comparison with Existing Approaches

Several comparative studies about texture features have been conducted. Ohanian and Dubes [23] 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. [24] did a similar study on joint occurrences of feature pairs using nine texture images and the ones in [23].

Recently, Randen and Husoy [27] did an extensive comparative study for texture classification on cooccurrence methods, Law’s texture measures, different filtering-based methods,
Table 1: Classification errors of methods shown in [27] and the proposed method

<table>
<thead>
<tr>
<th>Texture group</th>
<th>Methods in [27]</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Best</td>
</tr>
<tr>
<td>Fig. 8</td>
<td>47.9 %</td>
<td>32.3 %</td>
</tr>
<tr>
<td>Fig. 9</td>
<td>89.0 %</td>
<td>84.9 %</td>
</tr>
<tr>
<td>Fig. 11</td>
<td>14.1 %</td>
<td>0.7 %</td>
</tr>
<tr>
<td>Fig. 12</td>
<td>11.3 %</td>
<td>2.5 %</td>
</tr>
</tbody>
</table>

and a neural network approach [16]. They used a supervised classifier of Kohonen [20] 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 neighboring ones. We have applied our method to the same images. We use an integration scale $35 \times 35$ and the same filters as for the 40-image database shown in the previous section. We use 1/3 samples for training and the remaining 2/3 samples for testing. In our example, the samples in the training set have no overlap with those samples in the testing set in their receptive fields. The results for four groups of texture images are summarized in Table I, where the average performance and the best in Tables 3, 6, 8, and 9 in [27] are shown.

The first group consists of 10 texture images, which is shown in Figure 8(a). In this group, each image is visually different from other images. The classification gain of all the methods studied in [27] is shown in Figure 8(b). Our method is significantly better than the best performance in [27]. The second group, shown in Figure 9(a), is very challenging for filtering methods due to the inhomogeneity within each texture region and similarity among different textures. For all the methods in [27], the performance is close to a random decision, as shown in Figure 9(b). Our method, however, gives only 17.5% classification error, which dramatically improves the classification performance. This suggests that filter-responses cannot be used directly as a texture feature due to their inhomogeneity. Also to compare filter responses directly, the inputs need to be aligned spatially, which are not true for texture images in general.

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

The striking difference in performance between all the methods studied in [27] and our method may be qualitatively explained using the example in Figure 10, which shows three perceptually similar texture images and a white noise image. Table 2 shows the peak signal to noise ratio (PSNR) similarity measure between images in Figure 10. PSNR is often used as a quality standard in image coding and compression and is defined as

$$PSNR(W^1, W^2) = 20 \log_{10} \left( \frac{255}{\sqrt{1/|W^1| \sum_i \sum_j (W^1_{i,j} - W^2_{i,j})^2}} \right).$$  

Note that a larger PSNR value indicates a better representation. Table 2 shows that Figure
10(d) is a better representation for the other three images. Since a distance measure of filter responses is closely related to PSNR, one can expect a close-to-random performance when perceptual structures are a dominating factor in textures to be classified. This is the case especially for images shown in Figure 9. The images are not homogeneous and the best characterization is using perceptual structures. This is further justified by the example shown in Figures 11 and 12. As shown in Figures 11(b) and 12(b), many methods studied in [27] have difficulties in classifying these pairs of textures. However, our method gives a perfect classification result in both cases.

To further illustrate our method, we have also done a comparison with a method proposed by Azencott, Wang, and Younes [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 eight filters as in the previous section to compute spectral histogram. The sixteen texture images used in [1] are shown in Figure 13; 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 images. The result in [1] using the Kullback distance 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 capability of the spectral histogram.

5 A Unified Texture Feature

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 relationships 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 non-texture images. However, textures are often studied separately from intensity images and texture features from other approaches may not be applicable to uniform images [5].

Texture analysis has been studied extensively and many methods have been proposed. Tuceryan and Jain [32] classified existing approaches into four categories. We discuss the relationships between each category and our proposed method.

5.1 Statistical Methods

Statistical methods, including cooccurrence matrices [14], autocorrelation features [32], 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 [17, 19]. It is easy to see that the cooccurrence matrix can also be defined as responses of a specifically designed gradient filter. Because the cooccurrence matrix cannot be used directly as texture features, a number of texture features were subsequently computed from the cooccurrence matrix [14]. We can see that a spectral histogram can also be defined using a set of gradient filters. The cooccurrence matrix is also related to the Markov random field models [10]. More recently, Gimelfarb studied a texture model based on pairwise interactions [13].

5.2 Geometrical Methods

This class of methods is based on the assumption that a texture consists of repeated texture elements, such as the one shown Figure 3(b). After the texture elements are identified, geometrical properties of the element distribution can be used to characterize textures [32]. As shown in Figure 3(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.

5.3 Model Based Methods

Model based methods include Markov random fields [6, 37, 5, 30, 7, 13]. 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 [37], for example, Zhu, Wu, and Mumford proposed a FRAME model and showed that the model provides a unified framework for Markov random field models. In a limiting case, Wu, Zhu, and Liu [35] proved the equivalence of a model specified by features such as spectral histogram [38] and a Gibbs model, a special case of which is shown in [10]. This relationship 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.
5.4 Signal Processing Methods

This class of methods tries to characterize textures by filter responses directly. Many of these models have been studied and compared in [27], 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 texture structure and appearance. 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 [21].

6 Conclusions

We have demonstrated that the spectral histogram provides a sufficient feature statistic for characterizing texture appearance. The $\chi^2$-statistic between spectral histograms provides a perceptually meaningful measure for textures. We have obtained very good results for natural images. Our comparison shows that the spectral histogram improves the classification performance significantly.

Acknowledgments

Authors would like to thank S. C. Zhu for his insightful discussions and valuable comments. This research is partially supported by an ONR Young Investigator Award (N00014-96-1-0676) to DLW and a Presidential Fellowship to XL from the Ohio State University.

References


Figure 7: (a) The classification gain with respect to test-to-training sample ratio for the 40-image database. Solid line - integration scale $35 \times 35$; dashed line - integration scale $23 \times 23$. (b) 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 8 filters. (c) Classification gain of spectral histograms with different filters. Dashed line - two gradient filters; dash-dotted line - six Gabor filters; dotted line - four LoG filters; Solid line - eight filters consisting of one intensity, two gradient, two LoG, and three Gabor ones. (d) Classification gain for different widely-used distance measures for histograms. Solid line - $\chi^2$-statistic; dotted line - $L_1$-norm; dashed line - $L_2$-norm; dash-dotted - Kullback-Leibler divergence.
Figure 8: A 10-texture image group used in [27]. (a) The texture images in Figure 11(h) of [27]. Each image is $256 \times 256$. (b) The classification gain for all the methods in [27]. Here each data point represents one result in Tables 3, 6, 8, and 9 of [27]. Here dotted line indicates the random decision and dashed line is the result of the proposed classification method.
Figure 9: Another 10-texture image group used in [27]. (a) The texture images in Figure 11(i) of [27]. Each image is $256 \times 256$. (b) See the caption of Figure 8 for explanation.

Figure 10: Three perceptually similar synthetic texture images and a white noise image.
Figure 11: A texture pair used in [27]. (a) The texture images in Figure 12(a) of [27]. Each image is $256 \times 256$. (b) See the caption of Figure 8 for explanation.
Figure 12: A texture pair used in [27]. (a) The texture images in Figure 12(c) of [27]. Each image is 256 × 256. (b) See the caption of Figure 8 for explanation.
Figure 13: 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 $128 \times 128$. 