Chapter 15

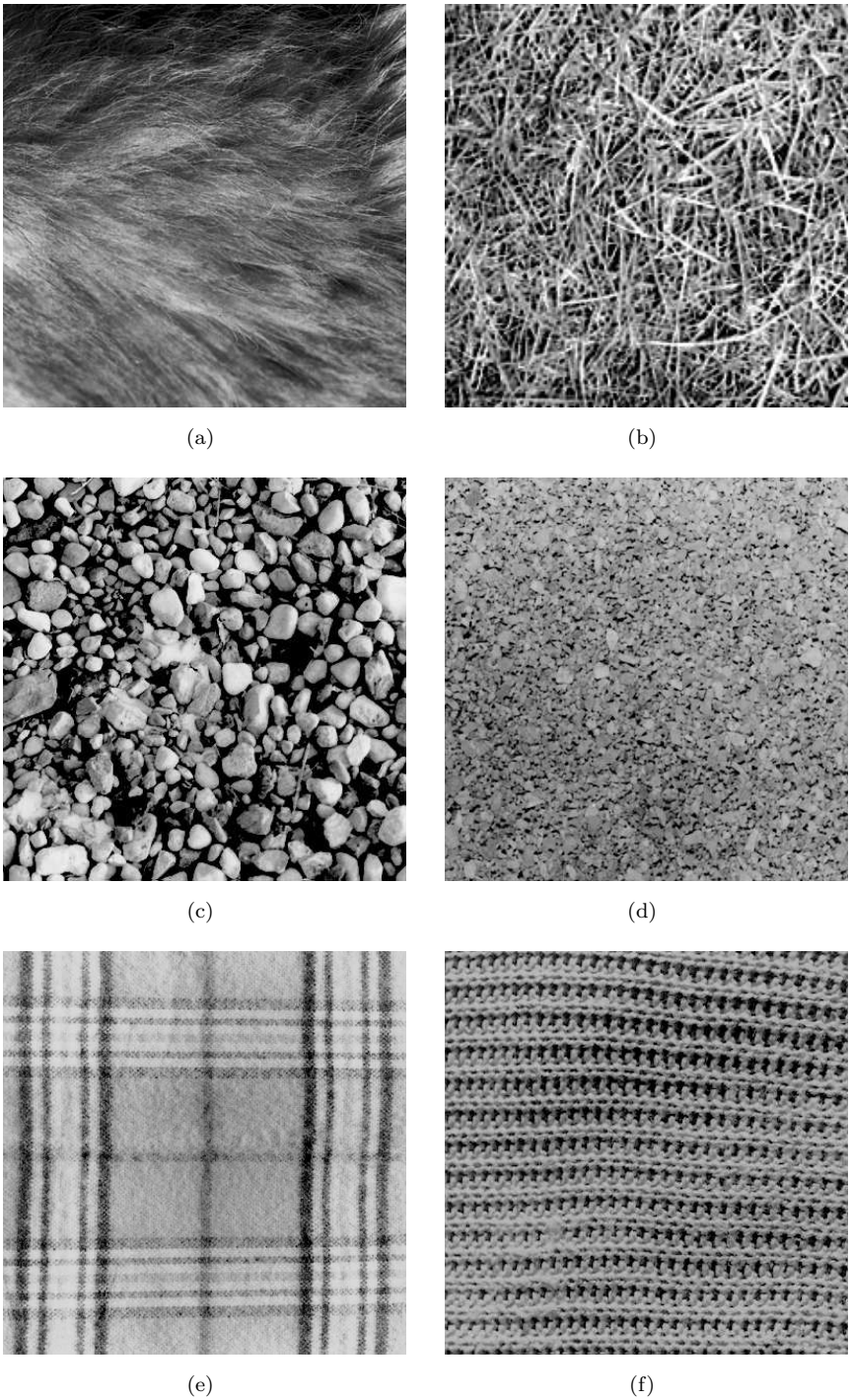
Texture

Texture refers to properties that represent the surface or structure of an object (in reflective or transmissive images, respectively); it is widely used, and perhaps intuitively obvious, but has no precise definition due to its wide variability. We might define texture as *something consisting of mutually related elements*; therefore we are considering a group of pixels (a **texture primitive** or **texture element**) and the texture described is highly dependent on the number considered (the texture scale) [Haralick, 1979]. Examples are shown in Figure 15.1; dog fur, grass, river pebbles, cork, checkered textile, and knitted fabric. Many other examples can be found in [Brodatz, 1966].

Texture consists of texture **primitives** or texture **elements**, sometimes called **texels**. Primitives in grass and dog fur are represented by several pixels and correspond to a stalk or a pile; cork is built from primitives that are comparable in size with pixels. It is difficult, however, to define primitives for the checkered textile or fabric, which can be defined by at least two hierarchical levels. The first level of primitives corresponds to textile checks or knitted stripes, and the second to the finer texture of the fabric or individual stitches. As we have seen in many other areas, this is a problem of **scale**; texture description is **scale dependent**.

The main aim of texture analysis is texture recognition and texture-based shape analysis. Textured properties of regions were referred to many times while considering image segmentation (Chapter 6), and derivation of shape from texture was discussed in Chapter 11. People usually describe texture as **fine**, **coarse**, **grained**, **smooth**, etc., implying that some more precise features must be defined to make machine recognition possible. Such features can be found in the **tone** and **structure** of a texture [Haralick, 1979]. Tone is based mostly on pixel intensity properties in the primitive, while structure is the spatial relationship between primitives.

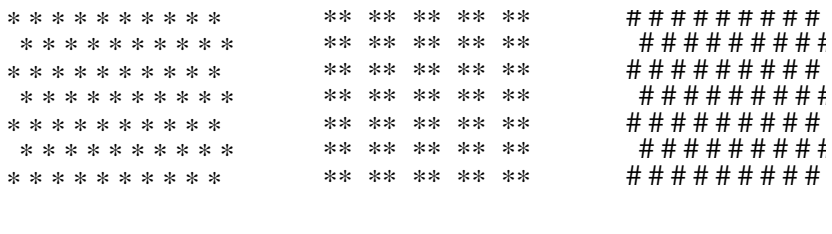
Each pixel can be characterized by its location and tonal properties. A texture primitive is a contiguous set of pixels with some tonal and/or regional property, and can be described by its average intensity, maximum or minimum intensity, size, shape, etc. The spatial relationship of primitives can be random, or they may be pairwise dependent, or some number of primitives can be mutually dependent. Image texture is then described by the number and types of primitives and by their spatial relationship.



(a) (b) (c) (d) (e) (f)

Figure 15.1: Textures: (a) dog fur; (b) grass; (c) river pebbles; (d) cork; (e) checkered textile; (f) knitted fabric.

Figures 15.1a,b and 15.2a,b show that the same number and the same type of primitives does not necessarily give the same texture. Similarly, Figures 15.2a and 15.2c show that the same spatial relationship of primitives does not guarantee texture uniqueness, and therefore is not sufficient for texture description. Texture tone and structure are not independent; textures always display both tone and structure even though one or the other usually dominates, and we usually speak about one or the other only. Tone can be understood as tonal properties of primitives, taking primitive spatial relationships into consideration. Structure refers to spatial relationships of primitives considering their tonal properties as well.



(a) (b) (c)

Figure 15.2: Artificial textures.

If the texture primitives in the image are small and if the tonal differences between neighboring primitives are large, a **fine** texture results (Figures 15.1a,b and 15.1d). If the texture primitives are larger and consist of several pixels, a **coarse** texture results (Figures 15.1c and 15.1e). Again, this is a reason for using both tonal and structural properties in texture description. Note that the fine/coarse texture characteristic depends on scale.

Further, textures can be classified according to their strength—texture strength then influences the choice of texture description method. **Weak** textures have small spatial interactions between primitives, and can be adequately described by frequencies of primitive types appearing in some neighborhood. Because of this, many statistical texture properties are evaluated in the description of weak textures. In **strong** textures, the spatial interactions between primitives are somewhat regular. To describe strong textures, the frequency of occurrence of primitive pairs in some spatial relationship may be sufficient. Strong texture recognition is usually accompanied by an exact definition of texture primitives and their spatial relationships.

It remains to define a constant texture. One existing definition [Sklansky, 1978] claims that '*an image region has a constant texture if a set of its local properties in that region is constant, slowly changing, or approximately periodic*'. The set of local properties can be understood as primitive types and their spatial relationships. An important part of the definition is that the properties must be repeated inside the constant texture area. How many times must the properties be repeated? Assume that a large area of constant texture is available, and consider smaller and smaller parts of that texture, digitizing it at constant resolution as long as the texture character remains unchanged. Alternatively, consider larger and larger parts of the texture, digitizing it at constant raster, until details become blurred and the primitives finally disappear. We see that image resolution (scale) must be a consistent part of the texture description; if the image resolution is appropriate, the texture character does not change for any position in our window.

Two main texture description approaches exist—**statistical** and **syntactic** [Haralick, 1979]. Statistical methods compute different properties and are suitable if texture primitive sizes are comparable with the pixel sizes. Syntactic and **hybrid** methods (combination of statistical and syntactic) are more suitable for textures where primitives can be assigned a label—the primitive type—meaning that primitives can be described using a larger variety of properties than just tonal properties; for example, shape description. Instead of tone, brightness will be used more often in the following sections because it corresponds better to gray-level images.

Research on pre-attentive (early) vision [Julesz, 1981] shows that human ability to recognize texture quickly is based mostly on **textons**, which are elongated blobs (rectangles, ellipses, line segments, line ends, crossings, corners) that can be detected by pre-attentive vision, while the positional relationship between neighboring textons must be done slowly by an attentive vision sub-system. As a result of these investigations, methods based on texton detection and texton density computation were developed.

15.1 Statistical texture description

Statistical texture description methods describe textures in a form suitable for statistical pattern recognition. As a result of the description, each texture is described by a feature vector of properties which represents a point in a multi-dimensional feature space. The aim is to find a deterministic or probabilistic decision rule assigning a texture to some specific class (see Chapter 9).

15.1.1 Methods based on spatial frequencies

Measuring spatial frequencies is the basis of a large group of texture recognition methods. Textural character is in direct relation to the spatial size of the texture primitives; coarse textures are built from larger primitives, fine textures from smaller primitives. Fine textures are characterized by higher spatial frequencies, coarse textures by lower spatial frequencies.

One of many related spatial frequency methods evaluates the **autocorrelation function of a texture**. In an autocorrelation model, a single pixel is considered a texture primitive, and primitive tone property is the gray-level. Texture spatial organization is described by the correlation coefficient that evaluates linear spatial relationships between primitives. If the texture primitives are relatively large, the autocorrelation function value decreases slowly with increasing distance, while it decreases rapidly if texture consists of small primitives. If primitives are placed periodically in a texture, the autocorrelation increases and decreases periodically with distance.

Texture can be described using the following algorithm.

Algorithm 15.1: Autocorrelation texture description

1. Evaluate autocorrelation coefficients for several different values of parameters p, q:

$$C_{ff}(p,q) = \frac{MN}{(M-p)(N-q)} \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} f(i,j) f(i+p, j+q)}{\sum_{i=1}^{M} \sum_{j=1}^{N} f^2(i,j)}, \tag{15.1}$$

where p, q is the position difference in the i, j direction, and M, N are the image dimensions.

2. Alternatively, the autocorrelation function can be determined in the frequency domain from the image power spectrum [Castleman, 1996]:

$$C_{ff} = \mathcal{F}^{-1}\{|F|^2\} . \tag{15.2}$$

If the textures described are circularly symmetric, the autocorrelation texture description can be computed as a function of the absolute position difference not considering direction—that is, a function of one variable.

Spatial frequencies can also be determined from an **optical image transform** (recall that the Fourier transform can be realized by a convex lens—see Section 3.2) [Shulman, 1970], a big advantage of which is that it may be computed in real time. The Fourier transform describes an image by its spatial frequencies; average values of energy in specific wedges and rings of the Fourier spectrum can be used as textural description features (see Figure 15.3). Features evaluated from rings reflect coarseness of the texture—high energy in large-radius rings is characteristic of fine textures (high frequencies), while high energy in small radii is characteristic of coarse textures (with lower spatial frequencies). Features evaluated from wedge slices of the Fourier transform image depend on directional properties of textures—if a texture has many edges or lines in a direction ϕ, high energy will be present in a wedge in direction $\phi + \pi/2$.

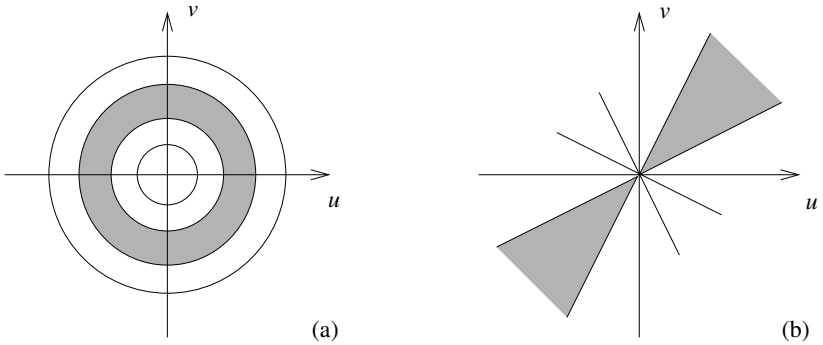


Figure 15.3: Partitioning of Fourier spectrum. (a) Ring filter. (b) Wedge filter reflecting the Fourier spectrum symmetry.

Similarly, a **discrete image transform** may be used for texture description. A textured image is usually divided into small square non-overlapping subimages. If the subimage size is $n \times n$, the gray-levels of its pixels may be interpreted as an n^2-dimensional vector, and an image can be represented as a set of vectors. These vectors are transformed applying a Fourier, Hadamard, or other discrete image transform (Section 3.2). The new co-ordinate system's basis vectors are related to the spatial frequencies of the original texture image and can be used for texture description [Rosenfeld, 1976]. When description of noisy texture becomes necessary, the problem becomes more difficult. From a set of 28 spatial frequency-domain features, a subset of features insensitive to additive noise was extracted (dominant peak energy, power spectrum shape, entropy) in [Liu and Jernigan, 1990].

Spatial frequency texture description methods are based on a well-known approach. Despite that, many problems remain—the resulting description is not invariant even to monotonic image gray-level transforms; further, it can be shown [Weszka et al., 1976] that the frequency-based approach is less efficient than others. A joint spatial/spatial frequency approach is recommended; the Wigner distribution was shown to be useful in a variety of synthetic and Brodatz textures [Reed et al., 1990].

15.1.2 Co-occurrence matrices

The co-occurrence matrix method of texture description is based on the repeated occurrence of some gray-level configuration in the texture; this configuration varies rapidly with distance in fine textures and slowly in coarse textures [Haralick et al., 1973]. Suppose the part of a textured image to be analyzed is an $M \times N$ rectangular window. An occurrence of some gray-level configuration may be described by a matrix of relative frequencies $P_{\phi,d}(a,b)$, describing how frequently two pixels with gray-levels a, b appear in the window separated by a distance d in direction ϕ. These matrices are symmetric if defined as given below. However, an asymmetric definition may be used, where matrix values are also dependent on the direction of co-occurrence. A co-occurrence matrix computation scheme was given in Algorithm 4.1.

Non-normalized frequencies of co-occurrence as functions of angle and distance can be represented formally as

$$\begin{aligned}
P_{0^\circ,d}(a,b) &= \big|\big\{[(k,l),(m,n)] \in D : \\
&\qquad k-m=0, |l-n|=d, f(k,l)=a, f(m,n)=b\big\}\big| \\
P_{45^\circ,d}(a,b) &= \big|\big\{[(k,l),(m,n)] \in D : \\
&\qquad (k-m=d, l-n=-d) \vee (k-m=-d, l-n=d), f(k,l)=a, f(m,n)=b\big\}\big| \\
P_{90^\circ,d}(a,b) &= \big|\big\{[(k,l),(m,n)] \in D : \\
&\qquad |k-m|=d, l-n=0, f(k,l)=a, f(m,n)=b\big\}\big| \\
P_{135^\circ,d}(a,b) &= \big|\big\{[(k,l),(m,n)] \in D : \\
&\qquad (k-m=d, l-n=d) \vee (k-m=-d, l-n=-d), f(k,l)=a, f(m,n)=b\big\}\big| ,
\end{aligned} \tag{15.3}$$

where $|\{\ldots\}|$ refers to set cardinality and $D = (M \times N) \times (M \times N)$.

An example illustrates co-occurrence matrix computations for the distance $d = 1$. A 4×4 image with four gray-levels is presented in Figure 15.4. The matrix $P_{0^\circ,1}$ is constructed

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Figure 15.4: Gray-level image.

as follows: The element $P_{0^\circ,1}(0,0)$ represents the number of times the two pixels with gray-levels 0 and 0 appear separated by distance 1 in direction 0°; $P_{0^\circ,1}(0,0) = 4$ in this case. The element $P_{0^\circ,1}(3,2)$ represents the number of times two pixels with gray-levels

3 and 2 appear separated by distance 1 in direction 0°; $P_{0°,1}(3,2) = 1$. Note that $P_{0°,1}(2,3) = 1$ due to matrix symmetry:

$$P_{0°,1} = \begin{vmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{vmatrix} \qquad P_{135°,1} = \begin{vmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{vmatrix}$$

The construction of matrices $P_{\phi,d}$ for other directions ϕ and distance values d is similar.

Texture classification can be based on criteria derived from the following co-occurrence matrices.

- **Energy**, or angular second moment (an image homogeneity measure—the more homogeneous the image, the larger the value)

$$\sum_{a,b} P^2_{\phi,d}(a,b) \,. \tag{15.4}$$

- **Entropy:**

$$\sum_{a,b} P_{\phi,d}(a,b) \, \log_2 P_{\phi,d}(a,b) \,. \tag{15.5}$$

- **Maximum probability:**

$$\max_{a,b} P_{\phi,d}(a,b) \,. \tag{15.6}$$

- **Contrast** (a measure of local image variations; typically $\kappa = 2, \lambda = 1$):

$$\sum_{a,b} |a-b|^\kappa \, P^\lambda_{\phi,d}(a,b) \,. \tag{15.7}$$

- **Inverse difference moment:**

$$\sum_{a,b;a\neq b} \frac{P^\lambda_{\phi,d}(a,b)}{|a-b|^\kappa} \,. \tag{15.8}$$

- **Correlation** (a measure of image linearity, linear directional structures in direction ϕ result in large correlation values in this direction):

$$\frac{\sum_{a,b}\left[(ab)P_{\phi,d}(a,b)\right] - \mu_x\mu_y}{\sigma_x\sigma_y} \,, \tag{15.9}$$

  where μ_x, μ_y are means and σ_x, σ_y are standard deviations

$$\mu_x = \sum_a a \sum_b P_{\phi,d}(a,b) \,, \qquad \sigma_x = \sum_a (a-\mu_x)^2 \sum_b P_{\phi,d}(a,b) \,,$$
$$\mu_y = \sum_b b \sum_a P_{\phi,d}(a,b) \,, \qquad \sigma_y = \sum_b (b-\mu_x)^2 \sum_a P_{\phi,d}(a,b) \,.$$

Following is a general algorithm for texture description based on co-occurrence matrices.

Algorithm 15.2: Co-occurrence method of texture description

1. Construct co-occurrence matrices for given directions and given distances.
2. Compute texture feature vectors for four directions ϕ, different values of d, and the six characteristics. This results in many correlated features.

The co-occurrence method describes second-order image statistics and works well for a large variety of textures (see [Gotlieb and Kreyszig, 1990] for a survey of texture descriptors based on co-occurrence matrices). Good properties of the co-occurrence method are the description of spatial relations between tonal pixels, and invariance to monotonic gray-level transformations. On the other hand, it does not consider primitive shapes, and therefore cannot be recommended if the texture consists of large primitives. Memory requirements represent another big disadvantage, although this is definitely not as limiting as it was a few years ago. The number of gray-levels may be set to 32 or 64, which decreases the co-occurrence matrix sizes, but loss of gray-level accuracy is a resulting negative effect (although this loss is usually insignificant in practice).

Although co-occurrence matrices give very good results in discrimination between textures, the method is computationally expensive. A fast algorithm for co-occurrence matrix computation is given in [Argenti et al., 1990], and a modification of the method that is efficiently applicable to texture description of detected regions is proposed in [Carlson and Ebel, 1988], in which a co-occurrence array size varies with the region size.

15.1.3 Edge frequency

Methods discussed so far describe texture by its spatial frequencies, but comparison of edge frequencies in texture can be used as well. Edges can be detected either as micro-edges using small edge operator masks, or as macro-edges using large masks [Davis and Mitiche, 1980]. The simplest operator that can serve this purpose is Robert's gradient, but virtually any other edge detector can be used (see Section 5.3.2). Using a gradient as a function of distance between pixels is another option [Sutton and Hall, 1972]. The distance-dependent texture description function $g(d)$ can be computed for any subimage f defined in a neighborhood N for variable distance d:

$$\begin{aligned} g(d) = \quad & \big|f(i,j) - f(i+d,j)\big| + \big|f(i,j) - f(i-d,j)\big| \\ & + \big|f(i,j) - f(i,j+d)\big| + \big|f(i,j) - f(i,j-d)\big| \,. \end{aligned} \tag{15.10}$$

The function $g(d)$ is similar to the negative autocorrelation function; its minimum corresponds to the maximum of the autocorrelation function, and its maximum corresponds to the autocorrelation minimum.

Algorithm 15.3: Edge-frequency texture description

1. Compute a gradient $g(d)$ for all pixels of the texture.
2. Evaluate texture features as average values of gradient in specified distances d.

Dimensionality of the texture description feature space is given by the number of distance values d used to compute the edge gradient.

Several other texture properties may be derived from first-order and second-order statistics of edge distributions [Tomita and Tsuji, 1990].

- **Coarseness:** Edge density is a measure of coarseness. The finer the texture, the higher is the number of edges present in the texture edge image.
- **Contrast:** High-contrast textures are characterized by large edge magnitudes.
- **Randomness:** Randomness may be measured as entropy of the edge magnitude histogram.
- **Directivity:** An approximate measure of directivity may be determined as entropy of the edge direction histogram. Directional textures have an even number of significant histogram peaks, directionless textures have a uniform edge direction histogram.
- **Linearity:** Texture linearity is indicated by co-occurrences of edge pairs with the same edge direction at constant distances, and edges are positioned in the edge direction (see Figure 15.5, edges a and b).
- **Periodicity:** Texture periodicity can be measured by co-occurrences of edge pairs of the same direction at constant distances in directions perpendicular to the edge direction (see Figure 15.5, edges a and c).
- **Size:** Texture size measure may be based on co-occurrences of edge pairs with opposite edge directions at constant distance in a direction perpendicular to the edge directions (see Figure 15.5, edges a and d).

Note that the first three measures are derived from first-order statistics, the last three measures from second-order statistics.

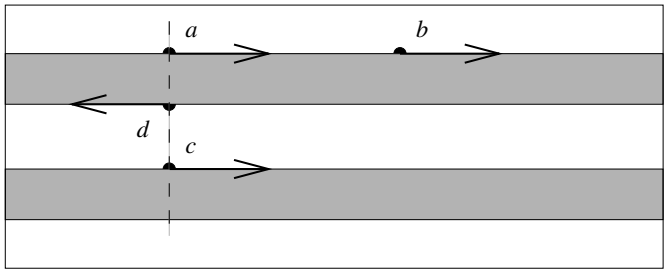


Figure 15.5: Texture linearity, periodicity, and size measures may be based on image edges. *Adapted from [Tomita and Tsuji, 1990].*

Many existing texture recognition methods are based on texture detection. The concepts of pre-attentive vision and textons have been mentioned, which are also based mostly on edge-related information. A zero-crossing operator was applied to edge-based texture description in [Perry and Lowe, 1989]; the method determines image regions of a constant texture, assuming no a priori knowledge about the image, texture types, or scale. Feature analysis is performed across multiple window sizes.

A slightly different approach to texture recognition may require detection of borders between homogeneous textured regions. A hierarchical algorithm for textured image segmentation is described in [Fan, 1989], and a two-stage contextual classification and

segmentation of textures based on a coarse-to-fine principle of edge detection is given in [Fung et al., 1990]. Texture description and recognition in the presence of noise represents a difficult problem. A noise-tolerant texture classification approach based on a Canny-type edge detector is discussed in [Kjell and Wang, 1991] where texture is described using periodicity measures derived from noise-insensitive edge detection.

15.1.4 Primitive length (run length)

A large number of neighboring pixels of the same gray-level represents a coarse texture, a small number of these pixels represents a fine texture, and the lengths of texture primitives in different directions can serve as a texture description [Galloway, 1975]. A primitive is a maximum contiguous set of constant-gray-level pixels located in a line; these can then be described by gray-level, length, and direction. The texture description features can be based on computation of continuous probabilities of the length and the gray-level of primitives in the texture.

Let $B(a,r)$ be the number of primitives of all directions having length r and gray-level a, M, N the image dimensions, and L the number of image gray-levels. Let N_r be the maximum primitive length in the image. The texture description features can be determined as follows. Let K be the total number of runs

$$K = \sum_{a=1}^{L} \sum_{r=1}^{N_r} B(a,r) \,. \tag{15.11}$$

Then:

- **Short primitives emphasis:**

$$\frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{N_r} \frac{B(a,r)}{r^2} \,. \tag{15.12}$$

- **Long primitives emphasis:**

$$\frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{N_r} B(a,r) r^2 \,. \tag{15.13}$$

- **Gray-level uniformity:**

$$\frac{1}{K} \sum_{a=1}^{L} \left[\sum_{r=1}^{N_r} B(a,r) \right]^2 \,. \tag{15.14}$$

- **Primitive length uniformity:**

$$\frac{1}{K} \sum_{r=1}^{N_r} \left[\sum_{a=1}^{L} B(a,r) \right]^2 \,. \tag{15.15}$$

- **Primitive percentage:**

$$\frac{K}{\sum_{a=1}^{L} \sum_{r=1}^{N_r} r\, B(a,r)} = \frac{K}{MN} \,. \tag{15.16}$$

A general algorithm might then be the following.

Algorithm 15.4: Primitive-length texture description

1. Find primitives of all gray-levels, all lengths, and all directions in the texture image.
2. Compute texture description features as given in (15.12)–(15.16). These features then provide a description vector.

15.1.5 Laws' texture energy measures

Laws' texture energy measures determine texture properties by assessing average gray-level, edges, spots, ripples, and waves in texture [Laws, 1979; Wu et al., 1992]. The measures are derived from three simple vectors: $L_3 = (1, 2, 1)$ which represents averaging; $E_3 = (-1, 0, 1)$ calculating first difference (edges); and $S_3 = (-1, 2, -1)$, corresponding to the second difference (spots). After convolution of these vectors with themselves and each other, five vectors result:

$$\begin{aligned} L_5 &= (1, 4, 6, 4, 1)\,, \\ E_5 &= (-1, -2, 0, 2, 1)\,, \\ S_5 &= (-1, 0, 2, 0, -1)\,, \\ R_5 &= (1, -4, 6, -4, 1)\,, \\ W_5 &= (-1, 2, 0, -2, -1)\,. \end{aligned} \tag{15.17}$$

Mutual multiplying of these vectors, considering the first term as a column vector and the second term as a row vector, results in 5×5 Laws' masks. For example, the following mask can be derived

$$L_5^T \times S_5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}. \tag{15.18}$$

By convoluting the Laws' masks with a texture image and calculating energy statistics, a feature vector is derived that can be used for texture description.

15.1.6 Fractal texture description

Fractal-based texture analysis was introduced in [Pentland, 1984], where a correlation between texture coarseness and fractal dimension of a texture was demonstrated. A fractal is defined [Mandelbrot, 1982] as a set for which the Hausdorff-Besicovich dimension [Hausdorff, 1919; Besicovitch and Ursell, 1937] is strictly greater than the topological dimension; therefore, fractional dimension is the defining property. Fractal models typically relate a metric property such as line length or surface area to the elementary length or area used as a basis for determining the metric property; measuring coast length is a frequently used example [Mandelbrot, 1982; Pentland, 1984; Lundahl et al., 1986].

Suppose the coast length is determined by applying a 1-km-long ruler end to end to the coastline; the same procedure can be repeated with a 0.5-km ruler and other shorter or longer rulers. It is easy to understand that shortening of the ruler will be associated with an increase in the measured length. Importantly, the relation between the ruler length and the measured coast length can be considered a measure of the coastline's geometric properties, e.g., its roughness. The functional relationship between the ruler size r and the measured length L can be expressed as

$$L = c\, r^{1-D}\,, \tag{15.19}$$

where c is a scaling constant and D is the **fractal dimension** [Mandelbrot, 1982]. Fractal dimension has been shown to correlate well with the function's intuitive roughness.

While equation (15.19) can be applied directly to lines and surfaces, it is often more appropriate to consider the function as a stochastic process. One of the most important stochastic fractal models is the fractional Brownian motion model described in [Mandelbrot, 1982], which considers naturally rough surfaces as the end results of random walks. Importantly, intensity surfaces of textures can also be considered as resulting from random walks, and the fractional Brownian motion model can be applied to texture description.

Fractal description of textures is typically based on determination of fractal dimension and **lacunarity** to measure texture roughness and granularity from the image intensity function. The topological dimension of an image is equal to three—two spatial dimensions and the third dimension representing the image intensity. Considering the topological dimension T_d, the fractal dimension D can be estimated from the Hurst coefficient H [Hurst, 1951; Mandelbrot, 1982] as

$$H = T_d - D\,. \tag{15.20}$$

For images ($T_d = 3$), the Hurst parameter H or the fractal dimension D can be estimated from the relationship

$$E\left((\Delta f)^2\right) = c\left[(\Delta r)^H\right]^2 = c(\Delta r)^{6-2D}\,, \tag{15.21}$$

where $E()$ is an expectation operator, $\Delta f = f(i,j) - f(k,l)$ is the intensity variation, c is a scaling constant, and $\Delta r = \|(i,j) - (k,l)\|$ is the spatial distance. A simpler way to estimate fractal dimension is to use the following equation:

$$E\big(|\Delta f|\big) = \kappa(\Delta r)^{3-D}\,, \tag{15.22}$$

where $\kappa = E\big(|\Delta f|\big)_{\Delta r=1}$. By applying the log function and considering that $H = 3 - D$,

$$\log E\big(|\Delta f|\big) = \log\kappa + H\log(\Delta r)\,. \tag{15.23}$$

The parameter H can be obtained by using least-squares linear regression to estimate the slope of the curve of gray-level differences $gd(k)$ versus k in log–log scales [Wu et al., 1992]. Considering an $M \times M$ image f,

$$gd(k) = \frac{1}{\mu}\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-k-1}\big|f(i,j) - f(i,j+k)\big| + \sum_{i=0}^{M-k-1}\sum_{j=0}^{M-1}\big|f(i,j) - f(i+k,j)\big|\right)\,, \tag{15.24}$$

where $\mu = 2\,M(M-k-1)$. The scale k varies from 1 to the maximum selected value s. Fractal dimension D is then derived from the value of the Hurst coefficient. The approximation error of the regression line fit should be determined to prove that the analyzed texture is a fractal, and can thus be efficiently described using fractal measures. A small value of the fractal dimension D (large value of the parameter H) represents a fine texture, while large D (small H) corresponds to a coarse texture.

Single fractal dimension is not sufficient for description of natural textures. Lacunarity measures describe characteristics of textures of different visual appearance that have the same fractal dimension [Voss, 1986; Keller et al., 1989; Wu et al., 1992]. Given a fractal set A, let $P(m)$ represent the probability that there are m points within a box of size L centered about an arbitrary point of A. Let N be the number of possible points within the box, then $\sum_{m=1}^{N} P(m) = 1$ and the lacunarity λ is defined as

$$\lambda = \frac{M_2 - M^2}{M^2}\,, \tag{15.25}$$

where

$$M = \sum_{m=1}^{N} m\,P(m)\,, \qquad M_2 = \sum_{m=1}^{N} m^2 P(m)\,. \tag{15.26}$$

Lacunarity represents a second-order statistic and is small for fine textures and large for coarse ones.

A multi-resolution approach to fractal feature extraction was introduced in [Wu et al., 1992]. The multi-resolution feature vector MF that describes both texture roughness and lacunarity is defined as

$$MF = \left(H^{(m)}, H^{(m-1)}, \ldots, H^{(m-n+1)}\right), \tag{15.27}$$

where the parameters $H^{(k)}$ are estimated from pyramidal images $f^{(k)}$, where $f^{(m)}$ represents the full-resolution image of size $M = 2^m$, $f^{(m-1)}$ is the half-resolution image of size $M = 2^{m-1}$, etc., and n is the number of resolution levels considered. The multi-resolution feature vector MF can serve as a texture descriptor. Textures with identical fractal dimensions and different lacunarities can be distinguished, as was shown by classification of ultrasonic liver images in three classes—normal, hepatoma, and cirrhosis [Wu et al., 1992]. Practical considerations regarding calculation of fractal-based texture description features can be found in [Sarkar and Chaudhuri, 1994; Huang et al., 1994; Jin et al., 1995].

15.1.7 Multiscale texture description—wavelet domain approaches

Texture description is highly scale dependent. To decrease scale sensitivity, a texture may be described in multiple resolutions and an appropriate scale may be chosen to achieve the maximum discrimination [Unser and Eden, 1989]. **Gabor transforms** and **wavelets** (Section 3.2.7) are well suited to multi-scale texture characterization [Coggins and Jain, 1985; Mallat, 1989; Bovik et al., 1990; Unser, 1995]. Both are multi-scale spatial—spatial frequency filtering approaches, which were in the past dominated by Gabor filters. Recently, wavelets have been successfully applied to texture classification using pyramid- or tree-structured discrete wavelet transforms [Mallat, 1989; Chang and

Kuo, 1993] (Section 3.2.7), typically outperforming conventional texture characterization approaches.

In [Unser, 1995], overcomplete **discrete wavelet frames** were shown to outperform standard critically sampled wavelet texture feature extraction—the following paragraphs are based on Unser's work. Considering a discrete version of the wavelet transform in l_2 (the space of square summable sequences [Rioul, 1993]) textures are described using orthogonal wavelet frames. First, consider the principle of this approach using a single-dimensional signal x. Take a prototype filter h satisfying condition

$$H(z)\,H(z^{-1}) + H(-z)\,H(-z^{-1}) = 1\,, \tag{15.28}$$

where $H(z)$ is the z-transform of h [Oppenheim et al., 1999], and let the filter also be subjected to the lowpass constraint $H(z)|_{z=1} = 1$. A complementary high-pass filter g is then obtained by shift and modulation

$$G(z) = z\,H(-z^{-1})\,. \tag{15.29}$$

Using these two prototypes, a sequence of filters of increasing width can be iteratively generated as follows:

$$H_{i+1}(z) = H(z^{2^i})\,H_i(z)\,, \tag{15.30}$$

$$G_{i+1}(z) = G(z^{2^i})\,H_i(z)\,, \tag{15.31}$$

for $i = 0, \ldots, I-1$, initialized with $H_0(z) = 1$. The filters represent perfect reconstruction filter banks, which are used for definition of the individual wavelets used below. In the signal domain, a two-scale relationship is obtained:

$$\begin{aligned} h_{i+1}(k) &= [h]_{\uparrow 2^i} * h_i(k)\,, \\ g_{i+1}(k) &= [g]_{\uparrow 2^i} * h_i(k)\,, \end{aligned} \tag{15.32}$$

where $[.]_{\uparrow m}$ represents upsampling by a factor of m. In general, each iteration dilates the filters h_i and g_i by a factor of two; this sequence of filters is used to decompose the signal in sub-bands of approximately one octave each. Importantly, such filter sequences provide a full coverage of the frequency domain.

An orthogonal wavelet decomposition obtained by such a sequence of filters yields discrete normalized basis functions

$$\varphi_{i,l}(k) = 2^{i/2}\,h_i(k - 2^i l)\,, \tag{15.33}$$

$$\psi_{i,l}(k) = 2^{i/2}\,g_i(k - 2^i l)\,, \tag{15.34}$$

where i and l are scale and translation indices, respectively; the product $2^{i/2}$ is for inner product normalization. Consider a sequence of nested subspaces $l_2 \supset V_0 \supset V_1 \supset \ldots \supset V_I$. Here, $V_i = \text{span}\{\varphi_{i,l}\}_{l \in Z}$ is the approximation space at resolution i. Let subspaces W_i $(i = 1, \ldots, I)$ represent residue space at resolution i, defined as an orthogonal complement of V_i with respect to V_{i-1}, i.e., $V_{i-1} = V_i + W_i$ and $V_i \perp W_i$. The minimum l_2-norm approximation of x at scale i which corresponds to the orthogonal projection into V_i is given by

$$x_{(i)}(k) = \sum_{l \in Z} s_{(i)}(l)\,\varphi_{i,l}\,, \tag{15.35}$$

$$s_{(i)}(l) = \big\langle x(k), \varphi_{i,l}(k) \big\rangle_{l_2}\,, \tag{15.36}$$

where $\langle .,. \rangle$ represents the standard l_2 inner product and $\varphi_{i,0}(k) = 2^{i/2} h_i(k)$ is the discrete scaling function at resolution i. The residue (projection of x into W_i), is given by the complementary wavelet expansion

$$x_{(i-1)}(k) - x_{(i)}(k) = \sum_{l \in Z} d_{(i)}(l)\, \psi_{i,l}\,, \tag{15.37}$$

$$d_{(i)}(l) = \left\langle x(k), \psi_{i,l}(k) \right\rangle_{l_2}\,, \tag{15.38}$$

where $\psi_{i,0}(k) = 2^{i/2} g_i(k)$ is the discrete wavelet at scale i.

By combining the residues over all scales to a given depth I, a full discrete wavelet expansion of the signal is obtained

$$x(k) = \sum_{l \in Z} s_{(I)}(l)\, \varphi_{I,l} + \sum_{i=1}^{I} \sum_{l \in Z} d_{(i)}(l)\, \psi_{i,l}\,, \tag{15.39}$$

where d_i are wavelet coefficients and s_I are expansion coefficients of a coarser approximation $x_{(I)}$, see equation (15.35).

Importantly for texture analysis, the equations (15.36) and (15.38) can be obtained by simple filtering and down-sampling, yielding

$$\begin{aligned} s_{(I)}(l) &= 2^{I/2} [h_I^T * x]_{\downarrow 2^I}(l)\,, \\ d_{(i)}(l) &= 2^{i/2} [g_i^T * x]_{\downarrow 2^i}(l)\,, \end{aligned} \tag{15.40}$$

where $i = 1, \ldots, I$; $h^T(k) = h(-k)$; and $[.]_{\downarrow m}$ is down-sampling by a factor of m. An efficient algorithm based on direct filtering with a **filter bank** exists and is typically used for this purpose [Mallat, 1989].

When using the discrete wavelet transform coefficients for texture analysis, the most often used features are **wavelet energy signatures** [Chang and Kuo, 1993; Mojsilovic et al., 2000; Arivazhagan and Ganesan, 2003] and their second-order statistics [Van de Wouwer et al., 1999]. While the wavelet-based features can be used directly for texture description, this approach suffers from the lack of translation invariance. As described earlier, when attempting to state a definition of texture, this is an important property. One possibility to overcome this limitation is to compute the discrete wavelet transform for all possible shifts of the input signal, giving the following decomposition formulae:

$$\begin{aligned} s_I^{\mathrm{DWF}}(k) &= \left\langle h_I(k-l), x(k) \right\rangle_{l_2} = h_I^T * x(k)\,, \\ d_i^{\mathrm{DWF}}(k) &= \left\langle g_i(k-l), x(k) \right\rangle_{l_2} = g_i^T * x(k)\,, \end{aligned} \tag{15.41}$$

where $i = 1, \ldots, I$ and DWF denotes a discrete wavelet frame representation to distinguish this equation from the earlier equation (15.40). Hereon, we will deal with this DWF representation without specifically using the superscript. Equation (15.41) represents a non-sampled version of equation (15.40). The wavelet frame coefficients can be used for translation-invariant texture description. Importantly, a simple reconstruction formula exists and both decomposition and reconstruction can be obtained using filter banks [Unser, 1995].

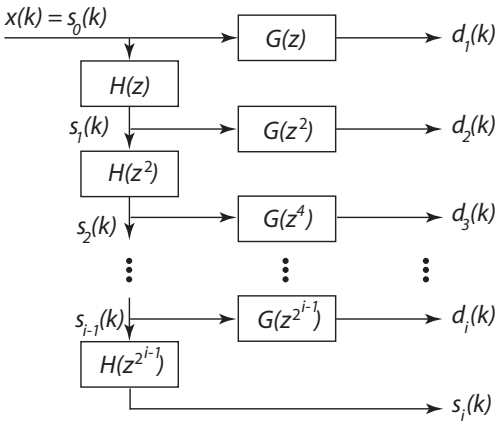


Figure 15.6: Fast iterative approach to discrete wavelet decomposition.

Practical implementation is based on the two-scale relationship given in equation (15.32) yielding a fast iterative decomposition algorithm

$$\begin{aligned} s_{i+1}(k) &= [h]_{\uparrow 2^i} * s_i(k)\,, \\ d_{i+1}(k) &= [g]_{\uparrow 2^i} * s_i(k)\,, \end{aligned} \tag{15.42}$$

where $i = 0, \ldots, I$, with the initial condition $s_0 = x$ (Figure 15.6). A convolution with the basic filters h and g is repeated in each step—yielding an approach the complexity of which is identical in each step and proportional to the number of samples.

Extending the single-dimensional case described above to higher-dimensional signals (to allow its use for texture image analysis) calls for the use of tensor product formulation [Mallat, 1989]. In a two-dimensional image, four distinct basis functions (four distinct filters) are defined corresponding to the different cross-products of the one-dimensional functions φ and ψ. The decomposition can therefore be obtained by successive one-dimensional processing along the rows and columns of an image. The output of the filter bank given in equation (15.41) can be rearranged in an N-component vector where N is the number of sub-bands

$$\mathbf{y}(k,l) = \big(y_i(k,l)\big)_{i=1,\ldots,N} = \big[s_I(k,l);\ d_I(k,l);\ \ldots;\ d_1(k,l)\big]^T\,. \tag{15.43}$$

For a spatial coordinate pair (k,l), the resulting $\mathbf{y}(k,l)$ is a linear transformation of the input vector $\mathbf{x}(k,l)$, which is a block representation of the input image centered at (k,l). Applying the 2D separable wavelet transform with a depth I yields $N = 1 + 3\,I$ features.

The texture can consequently be described by the set of N first-order probability density functions $p(y_i)$, for $i = 1, \ldots, N$. Also, a more compact representation can be obtained by using a set of channel variance features

$$v_i = \mathrm{var}\{y_i\} \tag{15.44}$$

(see [Unser, 1986] for justification of this approach). Needless to say, texture description capabilities of this methods depend on the appropriate choice of the filter bank.

The channel variances v_i can be estimated from average sum of squares over a region of interest R of the analyzed texture

$$v_i = \frac{1}{N_R} \sum_{(k,l)\in R} y_i^2(k,l)\,, \tag{15.45}$$

where N_R is the number of pixels in region R. As mentioned above, the lowpass condition requires $H(z)|_{z=1} = 1$, which in turn yields $E\{y_1\} = E\{x\}$, and $E\{y_i\} = 0$ for $i = 1, \ldots, N$. It is therefore recommended to subtract $E\{x\}^2$ from the lowpass channel feature to obtain a better estimate of the variance.

If the discrete wavelet transform is employed, a smaller number of coefficients results due to subsampling. Nevertheless, the variance estimates can be obtained in the same manner. However, an adverse effect on texture classification performance can be observed due to a resulting increased variability of features.

An assessment of the performance of the wavelet domain multiscale approach described above is given in [Unser, 1995]. In the experiments performed on 256×256 images of Brodatz textures [Brodatz, 1966], the wavelet and filter bank decompositions were performed by global processing of the analyzed images. The performance was assessed in 64 (8×8) non-overlapping subregions sized 32×32 pixels in which the texture was described using independent feature vectors $\mathbf{v} = (v_1, \ldots, v_N)$ evaluated according to equation (15.45), the assessment used a leave-one-out training/testing approach. The performance assessment demonstrated that the *discrete* wavelet frame approach always outperformed the *discrete* wavelet transform approach. It has showed that true multi-resolution feature extraction with two or three levels ($I = 2, 3$) is preferable to local single-scale analysis. The results also showed that even with $n = 0$, the DWF features performed very well. Importantly, the multiscale approach with 3 decompositions ($n = 3$) at 3 levels of scale ($I = 3$) and using 10 features outperformed (correctness of 99.2%) the single scale DWF approach ($n = 0$, $I = 1$) which used 4 features (correctness of 96.5%). This is notable since other comparison studies previously demonstrated that this DWF approach ($n = 0$, $I = 1$, equivalent to local linear transform using 2×2 Hadamard transform [Unser, 1986]) typically outperforms most other statistical texture description methods including co-occurrence matrices, correlation, etc. and can thus be used as a reference method for single-scale analysis. The studies in [Unser, 1995] also compare performance of various orthogonal and non-orthogonal wavelet transforms.

Comparison of texture classification behavior of Gabor transforms and wavelets is given in [Vautrot et al., 1996]. If texture segmentation is a goal, a coarse-to-fine multi-resolution strategy may be used approximate position of borders between texture regions being detected first in a low-resolution image, and accuracy being improved in higher resolutions using the low-level segmentation as a priori information. Wavelet-domain hidden Markov models (Section 10.9), and especially the hidden Markov trees are designed directly considering the intrinsic properties of wavelet transforms and combine the multiscale approach offered by wavelets with modeling of mutual statistical dependencies and non-Gaussian statistics frequently encountered in real-world texture analysis [Crouse et al., 1998]. The hidden Markov tree is a finite-state machine in the wavelet domain (usually using 2 states), effectively characterizing the joint statistics of the discrete wavelet transform by capturing inter-state dependencies of wavelet coefficients via Markov chains across scales [Fan and Xia, 2003].

15.1.8 Other statistical methods of texture description

A brief overview of some other texture description techniques will illustrate the variety of published methods; we present here only the basic principles of some additional approaches [Haralick, 1979; Ahuja and Rosenfeld, 1981; Davis et al., 1983; Derin and Elliot, 1987; Tomita and Tsuji, 1990].

The **mathematical morphology** approach looks for spatial repetitiveness of shapes in a binary image using structure primitives (see Chapter 13). If the structuring elements consist of a single pixel only, the resulting description is an autocorrelation function of the binary image. Using larger and more complex structuring elements, general correlation can be evaluated. The structuring element usually represents some simple shape, such as a square, a line, etc. When a binary textured image is eroded by this structuring element, texture properties are present in the eroded image [Serra and Verchery, 1973]. One possibility for feature vector construction is to apply different structuring elements to the textured image and to count the number of pixels with unit value in the eroded images, each number forming one element of the feature vector. The mathematical morphology approach stresses the shape properties of texture primitives, but its applicability is limited due to the assumption of a binary textured image. Methods of gray-level mathematical morphology may help to solve this problem. The mathematical morphology approach to texture description is often successful in granulated materials, which can be segmented by thresholding. Using a sequence of openings and counting the number of pixels after each step, a texture measure was derived in [Dougherty et al., 1989].

The **texture transform** represents another approach. Each texture type present in an image is transformed into a unique gray-level; the general idea is to construct an image g where the pixels $g(i,j)$ describe a texture in some neighborhood of the pixel $f(i,j)$ in the original textured image f. If micro-textures are analyzed, a small neighborhood of $f(i,j)$ must be used, and an appropriately larger neighborhood should be used for description of macro-textures. In addition, a priori knowledge can be used to guide the transformation and subsequent texture recognition and segmentation. Local texture orientation can also be used to transform a texture image into a feature image, after which supervised classification is applied to recognize textures.

Linear estimates of gray-levels in texture pixels can also be used for texture description. Pixel gray-levels are estimated from gray-levels in their neighborhood—this method is based on the **autoregression texture model**, where linear estimation parameters are used [Deguchi and Morishita, 1978]. The model parameters vary substantially in fine textures, but remain mostly unchanged if coarse texture is described. The autoregression model has been compared with an approach based on second-order spatial statistics [Gagalowicz et al., 1988]; it was found that even if the results are almost the same, spatial statistics performed much more quickly and reliably.

The **peak and valley** method [Mitchell et al., 1977; Ehrick and Foith, 1978] is based on detection of local extrema of the brightness function in vertical and horizontal scans of a texture image. Fine textures have a large number of small-sized local extrema, coarse textures are represented by a smaller number of larger-sized local extrema—higher peaks and deeper valleys.

The sequence of pixel gray-levels can be considered a **Markov chain** in which the transition probabilities of an m^{th}-order chain represent $(m+1)^{th}$-order statistics of textures [Pratt and Faugeras, 1978]. This approach may also be used for texture synthesis [Gagalowicz, 1979].

Many of the texture description features presented so far are interrelated; the Fourier power spectrum, the autoregression model, and autocorrelation functions represent the same subset of second-order statistics. The mathematical relationships between texture description methods are summarized in [Tomita and Tsuji, 1990], an experimental comparison of performance between several methods can be found in [Du Buf et al., 1990;

Iversen and Lonnestad, 1994; Zhu and Goutte, 1995; Wang et al., 1996], and criteria for comparison are discussed in [Soh et al., 1991].

It has been shown that higher than second-order statistics contain little information that can be used for texture discrimination [Julesz and Caelli, 1979]. Nevertheless, identical second-order statistics do not guarantee identical textures; examples can be found in [Julesz and Bergen, 1987] together with a study of human texture perception. Texture-related research of the human visual system seems to bring useful results, and a texture analysis method based on studies of it was designed to emulate the process of texture feature extraction in each individual channel in the multi-channel spatial filtering model of perception [Rao, 1993]. Results of the texture recognition process were compared with co-occurrence matrix recognition, and the model-based approach gave superior results in many respects.

15.2 Syntactic texture description methods

Syntactic and hybrid texture description methods are not as widely used as statistical approaches [Tomita et al., 1982]. **Syntactic** texture description is based on an analogy between the texture primitive spatial relations and the structure of a formal language. Descriptions of textures from one class form a language that can be represented by its grammar, which is inferred from a training set of words of the language (from descriptions of textures in a training set)—during a learning phase, one grammar is constructed for each texture class present in the training set. The recognition process is then a syntactic analysis of the texture description word. The grammar that can be used to complete the syntactic analysis of the description word determines the texture class (see Section 9.4).

Purely syntactic texture description models are based on the idea that textures consist of primitives located in almost regular relationships. Primitive descriptions and rules of primitive placement must be determined to describe a texture [Tsuji and Tomita, 1973; Lu and Fu, 1978]. Primitive spatial relation description methods were discussed at the beginning of this chapter. One of the most efficient ways to describe the structure of primitive relationships is using a grammar which represents a rule for building a texture from primitives, by applying transformation rules to a limited set of symbols. Symbols represent the texture primitive types and transformation rules represent the spatial relations between primitives. In Chapter 9 it was noted that any grammar is a very strict formula. On the other hand, textures of the real world are usually irregular, and structural errors, distortions, or even structural variations are frequent. This means that no strict rule can be used to describe a texture in reality. To make syntactic description of real textures possible, variable rules must be incorporated into the description grammars, and non-deterministic or stochastic grammars must be used (see Section 9.4 and [Fu, 1974]). Further, there is usually no single description grammar for a texture class, which might be described by an infinite number of different grammars using different symbols and different transformation rules, and different grammar types as well. We will discuss chain grammars and graph grammars in the next sections, and other grammars suitable for texture description (tree, matrix) can be found in [Ballard and Brown, 1982; Fu, 1982; Vafaie and Bourbakis, 1988]. Another approach to texture description using generative principles is to use **fractals** [Mandelbrot, 1982; Barnsley, 1988].

15.2.1 Shape chain grammars

Shape chain grammars, whose definition matches that given in Section 9.4, are the simplest grammars that can be used for texture description. They generate textures beginning with a start symbol followed by application of transform rules, called **shape rules**. The generating process is over if no further transform rule can be applied. Texture synthesis consists of several steps. First, the transform rule is found. Second, the rule must be geometrically adjusted to match the generated texture exactly (rules are more general; they may not include size, orientation, etc.).

Algorithm 15.5: Shape chain grammar texture synthesis

1. Start a texture synthesis process by applying some transform rule to the start symbol.
2. Find a part of a previously generated texture that matches the left side of some transform rule. This match must be an unambiguous correspondence between terminal and non-terminal symbols of the left-hand side of the chosen transform rule with terminal and non-terminal symbols of the part of the texture to which the rule is applied. If no such part of the texture can be found, stop.
3. Find an appropriate geometric transform that can be applied to the left-hand side of the chosen rule to match it to the considered texture part exactly.
4. Apply this geometric transform to the right-hand side of the transform rule.
5. Substitute the specified part of the texture (the part that matches a geometrically transformed left-hand side of the chosen rule) with the geometrically transformed right-hand side of the chosen transform rule.
6. Continue with step 2.

We can demonstrate this algorithm on an example of hexagonal texture synthesis. Let V_n be a set of non-terminal symbols, V_t a set of terminal symbols, R a set of rules, S the start symbol (as in Section 9.4). The grammar [Ballard and Brown, 1982] is illustrated in Figure 15.7, which can then be used to generate hexagonal texture following

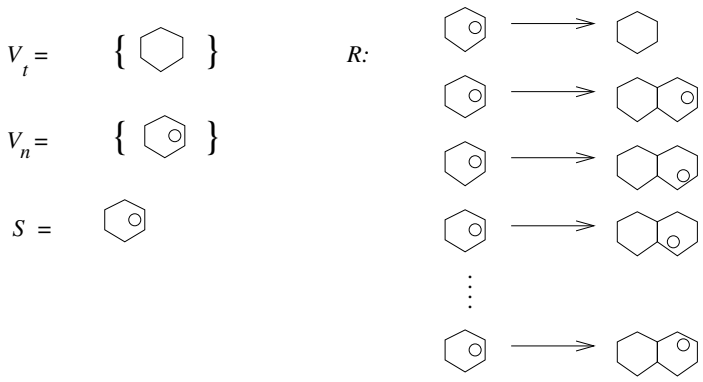


Figure 15.7: Grammar generating hexagonal textures.

Algorithm 15.5—note that the non-terminal symbol may appear in different rotations. Rotation of primitives here is represented by a small circle attached to one side of the primitive hexagon in Figure 15.7. Recognition of hexagonal textures is the proof that the texture can be generated by this grammar; the texture recognition uses syntactic analysis as described in Section 9.4. Note that the texture shown in Figure 15.8a will be accepted by the grammar (Figure 15.7), and recognized as a hexagonal texture. Figure 15.8b will be rejected—it is not a hexagonal texture according to the definition of Figure 15.7.

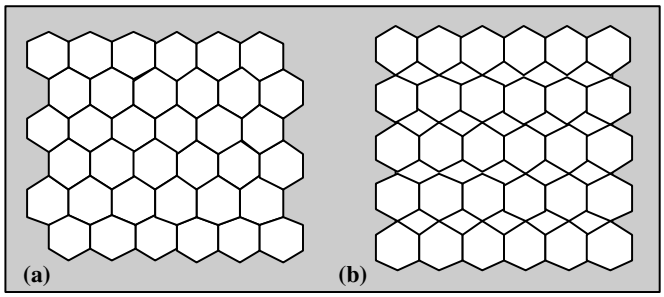


Figure 15.8: Hexagonal textures. (a) Accepted; (b) Rejected.

15.2.2 Graph grammars

Texture analysis is more common than texture synthesis in machine vision tasks (even if texture synthesis is probably more common in general, i.e., in computer graphics and computer games). The natural approach to texture recognition is to construct a planar graph of primitive layout and to use it in the recognition process. Primitive classes and primitive spatial relations must be known to construct such a graph; spatial relationships between texture primitives will then be reflected in the graph structure. Texture primitive classes will be coded in graph nodes, each primitive having a corresponding node in the graph, and two nodes will be connected by an arc if there is no other primitive in some specified neighborhood of these two primitives. The size of this neighborhood is the main influence on the complexity of the resulting planar graph—the larger the size of the neighborhood, the smaller the number of graph arcs. Note that choosing the neighborhood too large may result in no arcs for some nodes (the same may be true for the neighborhood being too small). Characteristic properties of some graphs used practically (relative neighborhood graphs, Gabriel graphs, Voronoi diagrams) are described in [Urquhart, 1982; Ahuja, 1982; Tuceryan and Jain, 1990]. These graphs are undirected since the spatial neighborhood relation is symmetric, with evaluated arcs and nodes. Each node is labeled with a primitive class to which it corresponds, and arcs are evaluated by their length and direction.

The texture classification problem is then transformed into a graph recognition problem for which the following approaches may be used.

1. Simplify the texture description by decomposition of the planar graph into a set of chains (sequences of adjacent graph nodes), and apply the algorithms discussed in the previous section. The chain descriptions of textures can represent border primitives of closed regions, different graph paths, primitive neighborhood, etc. A training set is constructed from the decomposition of several texture description planar graphs for

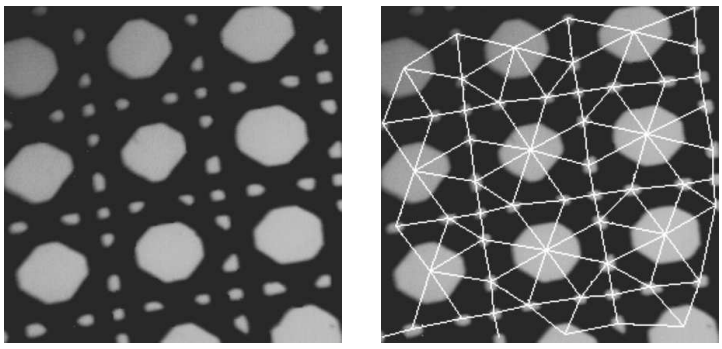
each texture class. Appropriate grammars are inferred which represent textures in the training sets. The presence of information noise is highly probable, so stochastic grammars should be used. Texture classification consists of the following steps.

- A classified texture is represented by a planar graph.
- The graph is decomposed into chains.
- The description chains are presented for syntactic analysis.
- A texture is classified into the class whose grammar accepts all the chains of the decomposed planar graph. If more than one grammar accepts the chains, the texture can be classified into the class whose grammar accepted the chains with the highest probability.

The main advantage of this approach is its simplicity. The impossibility of reconstructing the original planar graph from the chain decomposition is a disadvantage; it means that some portion of the syntactic information is lost during decomposition.

2. Another class of planar graph description may be represented by a stochastic graph grammar or by an extended graph grammar for description of distorted textures. This approach is very difficult from both the implementational and algorithmic points of view; the main problem is in grammar inference.
3. The planar graphs can be compared directly using graph matching approaches. It is necessary to define a 'distance' between two graphs as a measure of their similarity; if such a distance is defined, standard methods used in statistical classifier learning can be used—exemplar computation, cluster analysis, etc.

The syntactic approach is valued for its ability to describe a texture character at several hierarchical levels. It permits a qualitative analysis of textures, for decomposition into descriptive substructures (primitive grouping), to incorporate texture descriptions into the whole description of image, scene, etc. From this point of view, it significantly exceeds the complexity of simple object classification. Not considering the implementation difficulties, the second approach from the list above is recommended; if a descriptive graph grammar is chosen appropriately, it can generate a class of graphs independently of their size. It can be used if a pattern is sought in an image at any hierarchical level. An example of a planar graph describing a texture is shown in Figure 15.9.



(a) (b)

Figure 15.9: Planar graph describing a texture. (a) Texture primitives. (b) Planar graph overlaid.

15.2.3 Primitive grouping in hierarchical textures

Several levels of primitives can be detected in hierarchical textures—lower-level primitives form some specific pattern which can be considered a primitive at a higher description level (Figure 15.10). The process of detecting these primitive patterns (units) in a texture is called **primitive grouping**. Note that these units may form new patterns at an even higher description level. Therefore, the grouping process must be repeated until no new units can be formed.

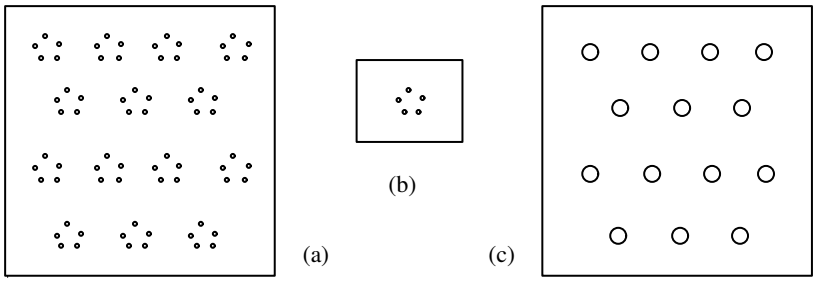


Figure 15.10: Hierarchical texture. (a) Texture. (b) A pattern formed from low-level primitives, this pattern can be considered a primitive in the higher level. (c) Higher-level texture.

Grouping makes a syntactic approach to texture segmentation possible. It plays the same role as local computation of texture features in statistical texture recognition. It has been claimed several times that different primitives and/or different spatial relationships represent different textures. Consider an example (Figure 15.11a) in which the primitives are the same (small circles) and textures differ in the spatial relations between primitives. If a higher hierarchical level is considered, different primitives can be detected in both textures—the textures do not consist of the same primitive types any more, see Figure 15.11b.

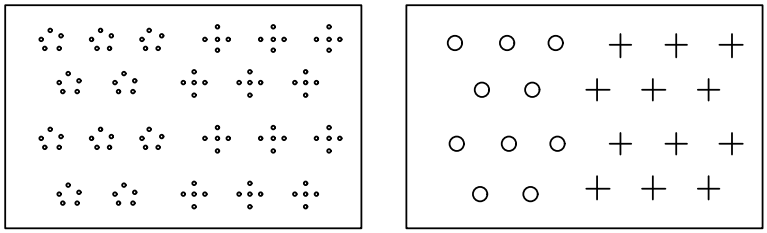


Figure 15.11: Primitive grouping. (a) Two textures, same primitives in the lowest description level. (b) The same two textures, different primitives in the higher description level.

A primitive grouping algorithm is described in [Tomita and Tsuji, 1990].

Algorithm 15.6: Texture primitive grouping

1. Determine texture primitive properties and classify primitives into classes.

2. Find the nearest and the second nearest neighbor for each texture primitive. Using the primitive class and distances to the nearest two neighboring primitives d_1, d_2, classify low-level primitives into **new** classes, see Figure 15.12.
3. Primitives with the same **new** classification which are connected (close to each other), are linked together and form higher-level primitives, see Figure 15.12.
4. If any two resulting homogeneous regions of linked primitives overlap, let the overlapped part form a separate region, see Figure 15.13.

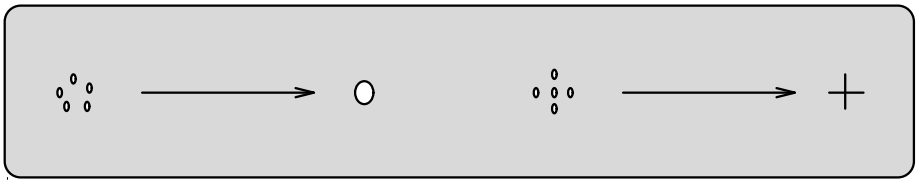


Figure 15.12: Primitive grouping—low-level primitive patterns are grouped into single primitives at a higher level.

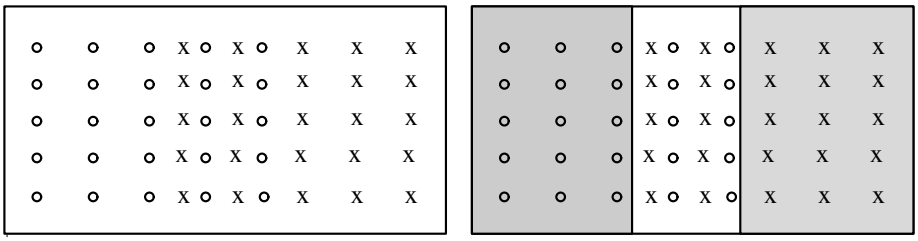


Figure 15.13: Overlap of homogeneous regions results in their splitting.

Regions formed from primitives of the lower level may be considered primitives in the higher level and the grouping process may be repeated for these new primitives. Nevertheless, sophisticated control of the grouping process is necessary to achieve meaningful results—it must be controlled from a high-level vision texture understanding sub-system. A recursive primitive grouping, which uses histograms of primitive properties and primitive spatial relations is presented in [Tomita and Tsuji, 1990] together with examples of syntactic-based texture segmentation results.

15.3 Hybrid texture description methods

Purely syntactic methods of texture description experience many difficulties with syntactic analyzer learning and with graph (or other complex) grammar inference. This is the main reason why purely syntactic methods are not widely used. On the other hand, a precise definition of primitives brings many advantages and it is not wise to avoid it completely. Hybrid methods of texture description combine the statistical and syntactic approaches; the technique is partly syntactic because the primitives are exactly defined, and partly

statistical because spatial relations between primitives are based on probabilities [Conners and Harlow, 1980a].

The hybrid approach to texture description distinguishes between weak and strong textures. The syntactic part of weak texture description divides an image into regions based on a tonal image property (e.g., constant gray-level regions) which are considered texture primitives. Primitives can be described by their shape, size, etc. The next step constructs histograms of sizes and shapes of all the texture primitives contained in the image. If the image can be segmented into two or more sets of homogeneous texture regions, the histogram is bi-modal and each primitive is typical of one texture pattern. This can be used for texture segmentation.

If the starting histogram does not have significant peaks, a complete segmentation cannot be achieved. The histogram-based segmentation can be repeated in each hitherto segmented homogeneous texture region. If any texture region consists of more than one primitive type, the method cannot be used and spatial relations between primitives must be computed. Some methods are discussed in [Haralick, 1979].

Description of strong textures is based on the spatial relations of texture primitives and two-directional interactions between primitives seem to carry most of the information. The simplest texture primitive is a pixel and its gray-level property, while the maximum contiguous set of pixels of constant gray-level is a more complicated texture primitive [Wang and Rosenfeld, 1981]. Such a primitive can be described by its size, elongatedness, orientation, average gray-level, etc. The texture description includes spatial relations between primitives based on distance and adjacency relations. Using more complex texture primitives brings more textural information. On the other hand, all the properties of single pixel primitives are immediately available without the necessity of being involved in extensive primitive property computations.

The hybrid multi-level texture description and classification method [Sonka, 1986] is based on primitive definition and spatial description of inter-primitive relations. The method considers both tone and structural properties and consists of several consequent steps. Texture primitives must be extracted first, and then described and classified. As a result of this processing stage, a classifier knows how to classify texture primitives. Known textures are presented to the texture recognition system in the second stage of learning. Texture primitives are extracted from the image and the first-level classifier recognizes their classes. Based on recognized texture primitives, spatial relations between primitive classes are evaluated for each texture from the training set. Spatial relations between texture primitives are described by a feature vector used to adjust a second-level classifier. If the second-level classifier is set, the two-level learning process is over, and unknown textures can be presented to the texture recognition system. The primitives are classified by the first-level classifier, spatial primitive properties are computed and the second-level classifier assigns the texture to one of the texture classes. Some hybrid methods use Fourier descriptors for shape coding and a texture is modeled by a reduced set of joint probability distributions obtained by vector quantization.

15.4 Texture recognition method applications

The estimated yield of crops or localization of diseased forests from remotely sensed data, automatic diagnosis of lung diseases from X-ray images, recognition of cloud types from meteorological satellite data, etc., are examples of texture recognition applications.

Textures are very common in our world, and application possibilities are almost unlimited. The effectiveness of various texture recognition methods is discussed in [Conners and Harlow, 1980b].

Texture recognition of roads, road crossings, buildings, agricultural regions, and natural objects, or classification of trees into five classes, belong to classical applications of spatial frequency-based texture description methods. An interesting proof of the role of textural information in outdoor object recognition was done by comparison of classification correctness if textural information was and was not used; spectral information-based classification achieved 74% correctly classified objects. Adding the textural information, accuracy increased to 99% [Haralick, 1979]. Real industrial applications of texture description and recognition are becoming more and more common. Examples can be found in almost all branches of industrial and biomedical activities—quality inspection in the motor or textile industries [Wood, 1990], workpiece surface monitoring, road surface skidding estimation, micro-electronics, remote sensing, mammography [Miller and Astley, 1992], MR brain imaging [Toulson and Boyce, 1992], pulmonary parenchyma characterization [Xu et al., 2006], three-dimensional texture images [Ip and Lam, 1995], content-based data retrieval from image databases, etc.

15.5 Summary

- **Texture**
  - Texture is widely used and intuitively obvious but has no precise definition due to its wide variability. One existing definition claims that '*an image region has a constant texture if a set of its local properties in that region is constant, slowly changing, or approximately periodic*'.
  - Texture consists of texture **primitives** (texture **elements**) called **texels**.
  - A texture primitive is a contiguous set of pixels with some tonal and/or regional property.
  - **Texture description** is based on **tone** and **structure**. Tone describes pixel intensity properties in the primitive, while structure reflects spatial relationships between primitives.
  - Texture description is **scale dependent**.
  - **Statistical** methods of texture description compute different texture properties and are suitable if texture primitive sizes are comparable with the pixel sizes.
  - **Syntactic** and **hybrid** methods (combination of statistical and syntactic) are more suitable for textures where primitives can be easily determined and their properties described.
- **Statistical texture description**
  - Statistical texture description methods describe textures in a form suitable for statistical pattern recognition. As a result of the description, each texture is described by a feature vector of properties which represents a point in a multi-dimensional feature space.
  - Coarse textures are built from larger primitives, fine textures from smaller primitives. Textural character is in direct relation to the spatial size of the texture primitives.

  - Fine textures are characterized by higher spatial frequencies, coarse textures by lower spatial frequencies.
  - Measuring spatial frequencies is the basis of a large group of texture recognition methods:
    * Autocorrelation function of a texture
    * Optical image transform
    * Discrete image transform
  - Texture description may be based on the repeated occurrence of some gray-level configuration in the texture; this configuration varies rapidly with distance in fine textures, slowly in coarse textures. **Co-occurrence matrices** represent such an approach.
  - **Edge frequency** approaches describe frequencies of edge appearances in texture.
  - In the **primitive length (run length)** approach, texture description features can be computed as continuous probabilities of the length and the gray-level of primitives in the texture.
  - **Laws' texture measures** determine texture properties by assessing average gray-level, edges, spots, ripples, and waves in texture.
  - **Fractal** approach to texture description is based on correlation between texture coarseness and fractal dimension and texture granularity and lacunarity.
  - **Wavelet** texture description
    * Wavelet texture description approaches are typically more efficient than other statistical texture analysis methods.
    * Wavelet energy signatures or their second-order statistics are frequently used.
    * Standard wavelet approaches are not translation invariant.
    * Discrete wavelet frames introduce the translational invariance and can be efficiently implemented using filter banks.
    * Wavelet-based hidden Markov models and hidden Markov trees incorporate independence between wavelet sub-bands for additional performance enhancement.
  - Other statistical approaches exist:
    * Mathematical morphology
    * Texture transform
  - Variety of texture properties may be derived from **first-order** and **second-order statistics** of elementary measures such as co-occurrences, edge distributions, primitive lengths, etc.
  - Higher than second-order statistics contain little information that can be used for texture discrimination.

- **Syntactic and hybrid texture description**
  - Syntactic texture description is based on an analogy between texture primitive spatial relations and structure of a formal language.
  - Hybrid methods of texture description combine the statistical and syntactic approaches; the technique is partly syntactic because the primitives are exactly defined, and partly statistical because spatial relations between primitives are based on probabilities.

- Purely syntactic texture description models utilize the idea that textures consist of primitives located in almost regular relationships. Primitive descriptions and rules of primitive placement must be determined to describe a texture.
- Textures of the real world are usually irregular, with frequent structural errors, distortions, and/or structural variations causing strict grammars to be inapplicable. To make syntactic description of real textures possible, variable rules must be incorporated into the description grammars, and non-deterministic or stochastic grammars must be used.
- Syntactic texture description methods include:
  - **Shape chain grammars**, which are the simplest grammars that can be used for texture description. They generate textures beginning with a start symbol followed by application of **shape transform rules**.
  - **Graph grammars**, an approach that constructs a **planar graph of primitive layout**. Primitive classes and primitive spatial relations must be known to construct such a graph; spatial relationships between texture primitives are reflected in the graph structure. The texture classification problem is then transformed into a **graph recognition** problem
- The syntactic approach is valued for its ability to describe a texture character at several **hierarchical levels**.
- **Primitive grouping** can be performed if lower-level primitives that form some specific pattern can be considered a primitive at a higher description level.
- Syntactic and hybrid texture description methods are not as widely used as statistical approaches.

15.6 References

Ahuja N. Dot pattern processing using Voronoi neighborhood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:336–343, 1982.

Ahuja N. and Rosenfeld A. Mosaic models for textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(1):1–11, 1981.

Argenti F., Alparone L., and Benelli G. Fast algorithms for texture analysis using co-occurrence matrices. *IEE Proceedings, Part F: Radar and Signal Processing*, 137(6):443–448, 1990.

Arivazhagan S. and Ganesan L. Texture classification using wavelet transform. *Pattern Recogn. Lett.*, 24:1513–1521, 2003.

Ballard D. H. and Brown C. M. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

Barnsley M. F. *Fractals Everywhere*. Academic Press, Boston, 1988.

Besicovitch A. S. and Ursell H. D. Sets of fractional dimensions (V): On dimensional numbers of some continuous curves. *Journal of the London Mathematical Society*, 12:18–25, 1937.

Bovik A. C., Clark M., and Geisler W. S. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.

Brodatz P. *Textures: A Photographic Album for Artists and Designers*. Dover, Toronto, 1966.

Carlson G. E. and Ebel W. J. Co-occurrence matrix modification for small region texture measurement and comparison. In *IGARSS'88—Remote Sensing: Moving towards the 21st Century,* Edinburgh, Scotland, pages 519–520, Piscataway, NJ, 1988. IEEE.

Castleman K. R. *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1996.

Chang T. and Kuo C. C. Texture analysis and classification with tree-structure wavelet transform. *IEEE Transactions on Image Processing*, 2:429–441, 1993.

Coggins J. M. and Jain A. K. A spatial filtering approach to texture analysis. *Pattern Recognition Letters*, 3:195–203, 1985.

Conners R. W. and Harlow C. A. Toward a structural textural analyser based on statistical methods. *Computer Graphics and Image Processing*, 12:224–256, 1980a.

Conners R. W. and Harlow C. A. A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3):204–222, 1980b.

Crouse M. S., Nowak R. D., and Baraniuk R. G. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46:886–902, 1998.

Davis L. S. and Mitiche A. Edge detection in textures. *Computer Graphics and Image Processing*, 12:25–39, 1980.

Davis L. S., Janos L., and Dunn S. M. Efficient recovery of shape from texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(5):485–492, 1983.

Deguchi K. and Morishita I. Texture characterization and texture-based partitioning using two-dimensional linear estimation. *IEEE Transactions on Computers*, 27:739–745, 1978.

Derin H. and Elliot H. Modelling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55, 1987.

Dougherty E. R., Kraus E. J., and Pelz J. B. Image segmentation by local morphological granulometries. In *Proceedings of IGARSS '89 and Canadian Symposium on Remote Sensing,* Vancouver, Canada, pages 1220–1223, New York, 1989. IEEE.

Du Buf J. M. H., Kardan M., and Spann M. Texture feature performance for image segmentation. *Pattern Recognition*, 23(3–4):291–309, 1990.

Ehrick R. W. and Foith J. P. A view of texture topology and texture description. *Computer Graphics and Image Processing*, 8:174–202, 1978.

Fan G. and Xia X. G. Wavelet-based texture analysis and synthesis using hidden Markov models. *IEEE Trans. Circuits and Systems*, 50:106–120, 2003.

Fan Z. Edge-based hierarchical algorithm for textured image segmentation. In *International Conference on Acoustics, Speech, and Signal Processing,* Glasgow, Scotland, pages 1679–1682, Piscataway, NJ, 1989. IEEE.

Fu K. S. *Syntactic Methods in Pattern Recognition.* Academic Press, New York, 1974.

Fu K. S. *Syntactic Pattern Recognition and Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

Fung P. W., Grebbin G., and Attikiouzel Y. Contextual classification and segmentation of textured images. In *Proceedings of the 1990 International Conference on Acoustics, Speech, and Signal Processing—ICASSP 90,* Albuquerque, NM, pages 2329–2332, Piscataway, NJ, 1990. IEEE.

Gagalowicz A. Stochatic texture fields synthesis from a priori given second order statistics. In *Proceedings, Pattern Recognition and Image Processing,* Chicago, IL, pages 376–381, Piscataway, NJ, 1979. IEEE.

Gagalowicz A., Graffigne C., and Picard D. Texture boundary positioning. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, pages 16–19, Beijing/Shenyang, China, 1988. IEEE.

Galloway M. M. Texture classification using gray level run length. *Computer Graphics and Image Processing*, 4:172–179, 1975.

Gotlieb C. C. and Kreyszig H. E. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51(1):70–86, 1990.

Haralick R. M. Statistical and structural approaches to texture. *Proceedings IEEE*, 67(5): 786–804, 1979.

Haralick R. M., Shanmugam K., and Dinstein I. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3:610–621, 1973.

Hausdorff F. Dimension und ausseres Mass. *Mathematische Annalen*, 79:157–179, 1919.

Huang Q., Lorch J. R., and Dubes R. C. Can the fractal dimension of images be measured? *Pattern Recognition*, 27:339–349, 1994.

Hurst H. E. Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770–808, 1951.

Ip H. H. S. and Lam S. W. C. Three-dimensional structural texture modeling and segmentation. *Pattern Recognition*, 28:1299–1319, 1995.

Iversen H. and Lonnestad T. An evaluation of stochastic models for analysis and synthesis of gray-scale texture. *Pattern Recognition Letters*, 15:575–585, 1994.

Jin X. C., Ong S. H., and Jayasooriah. A practical method for estimating fractal dimension. *Pattern Recognition Letters*, 16:457–464, 1995.

Julesz B. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981.

Julesz B. and Bergen J. R. Textons, the fundamental elements in preattentive vision and perception of textures. In *Readings in Computer Vision*, pages 243–256. Morgan Kaufmann Publishers, Los Altos, CA, 1987.

Julesz B. and Caelli T. On the limits of Fourier decompositions in visual texture perception. *Perception*, 8:69–73, 1979.

Keller J. M., Chen S., and Crownover R. M. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing*, 45(2):150–166, 1989.

Kjell B. P. and Wang P. Y. Noise-tolerant texture classification and image segmentation. In *Intelligent Robots and Computer Vision IX: Algorithms and Techniques,* Boston, pages 553–560, Bellingham, WA, 1991. SPIE.

Laws K. I. Texture energy measures. In *DARPA Image Understanding Workshop,* Los Angeles, CA, pages 47–51, Los Altos, CA, 1979. DARPA.

Liu S. S. and Jernigan M. E. Texture analysis and discrimination in additive noise. *Computer Vision, Graphics, and Image Processing*, 49:52–67, 1990.

Lu S. Y. and Fu K. S. A syntactic approach to texture analysis. *Computer Graphics and Image Processing*, 7:303–330, 1978.

Lundahl T., Ohley W. J., Kay S. M., and Siffert R. Fractional Brownian motion: A maximum likelihood estimator and its application to image texture. *IEEE Transactions on Medical Imaging*, 5:152–161, 1986.

Mallat S. G. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

Mandelbrot B. B. *The Fractal Geometry of Nature.* Freeman, New York, 1982.

Miller P. and Astley S. Classification of breast tissue by texture analysis. *Image and Vision Computing*, 10(5):277–282, 1992.

Mitchell O. R., Myer C. R., and Boyne W. A max-min measure for image texture analysis. *IEEE Transactions on Computers*, 26:408–414, 1977.

Mojsilovic A., Popovic M. V., and Rackov D. M. On the selection of an optimal wavelet basis for texture charactrization. *IEEE Transactions on Image Processing*, 9:2043–2050, 2000.

Oppenheim A. V., Schafer R. W., and Buck J. R. *Discrete-Time Signal Processing*. Prentice Hall, New York, 2nd edition, 1999.

Pentland A. P. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:661–674, 1984.

Perry A. and Lowe D. G. Segmentation of non-random textures using zero-crossings. In *1989 IEEE International Conference on Systems, Man, and Cybernetics,* Cambridge, MA, pages 1051–1054, Piscataway, NJ, 1989. IEEE.

Pratt W. K. and Faugeras O. C. Development and evaluation of stochastic-based visual texture features. *IEEE Transactions on Systems, Man and Cybernetics*, 8:796–804, 1978.

Rao A. R. Identifying high level features of texture perception. *CVGIP – Graphical Models and Image Processing*, 55:218–233, 1993.

Reed T. R., Wechsler H., and Werman M. Texture segmentation using a diffusion region growing technique. *Pattern Recognition*, 23(9):953–960, 1990.

Rioul O. A discrete-time multiresolution theory. *IEEE Trans on Signal Proc.*, 41:2591–2606, 1993.

Rosenfeld A., editor. *Digital Picture Analysis*. Springer Verlag, Berlin, 1976.

Sarkar N. and Chaudhuri B. B. An efficient differential box-counting approach to compute fractal dimension of image. *IEEE Transactions on Systems, Man and Cybernetics*, 24:115–120, 1994.

Serra J. and Verchery G. Mathematical morphology applied to fibre composite materials. *Film Science Technology*, 6:141–158, 1973.

Shulman A. R. *Optical Data Processing*. Wiley, New York, 1970.

Sklansky J. Image segmentation and feature extraction. *IEEE Transactions on Systems, Man and Cybernetics*, 8(4):237–247, 1978.

Soh Y., Murthy S. N. J., and Huntsberger T. L. Development of criteria to compare model-based texture analysis methods. In *Intelligent Robots and Computer Vision IX: Algorithms and Techniques,* Boston, pages 561–573, Bellingham, WA, 1991. SPIE.

Sonka M. A new texture recognition method. *Computers and Artificial Intelligence*, 5(4):357–364, 1986.

Sutton R. and Hall E. Texture measures for automatic classification of pulmonary diseases. *IEEE Transactions on Computers*, C–21(1):667–678, 1972.

Tomita F. and Tsuji S. *Computer Analysis of Visual Textures*. Kluwer, Norwell, MA, 1990.

Tomita F., Shirai Y., and Tsuji S. Description of textures by a structural analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(2):183–191, 1982.

Toulson D. L. and Boyce J. F. Segmentation of MR images using neural nets. *Image and Vision Computing*, 10(5):324–328, 1992.

Tsuji S. and Tomita F. A structural analyser for a class of textures. *Computer Graphics and Image Processing*, 2:216–231, 1973.

Tuceryan M. and Jain A. K. Texture segmentation using Voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):211–216, 1990.

Unser M. Local inear transforms for texture measurements. *Signal Processing*, 11:61–79, 1986.

Unser M. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4:1549–1560, 1995.

Unser M. and Eden M. Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):717–728, 1989.

Urquhart R. Graph theoretical clustering based on limited neighbourhood sets. *Pattern Recognition*, 15(3):173–187, 1982.

Vafaie H. and Bourbakis N. G. Tree grammar scheme for generation and recognition of simple texture paths in pictures. In *Third International Symposium on Intelligent Control 1988,* Arlington, VA, pages 201–206, Piscataway, NJ, 1988. IEEE.

Van de Wouwer G., Scheunders P., and Van Dyck D. Statistical texture characterization from wavelet representations. *IEEE Transactions on Image Processing*, 8:592–598, 1999.

Vautrot P., Bonnet N., and Herbin M. Comparative study of different spatial/spatial frequency methods for texture segmentation/classification. In *Proceedings of the IEEE International Conference on Image Processing,* Lausanne, Switzerland, pages III:145–148, Piscataway, NJ, 1996. IEEE.

Voss R. Random fractals: Characterization and measurement. In *Scaling Phenomena in Disordered Systems*. Plenum Press, New York, 1986.

Wang S. and Rosenfeld A. A relative effectiveness of selected texture primitive. *IEEE Transactions on Systems, Man and Cybernetics*, 11:360–370, 1981.

Wang Z., Guerriero A., and Sario M. D. Comparison of several approaches for the segmentation of texture images. *Pattern Recognition Letters*, 17:509–521, 1996.

Weszka J. S., Dyer C., and Rosenfeld A. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6(4):269–285, 1976.

Wood E. J. Applying Fourier and associated transforms to pattern characterization in textiles. *Textile Research Journal*, 60(4):212–220, 1990.

Wu C. M., Chen Y. C., and Hsieh K. S. Texture features for classification of ultrasonic liver images. *IEEE Transactions on Medical Imaging*, 11:141–152, 1992.

Xu Y., Sonka M., McLennan G., Guo J., and Hoffman E. MDCT-based 3-D texture classification of emphysema and early smoking related pathologies. *IEEE Transactions on Medical Imaging*, 25:464–475, 2006.

Zhu Y. M. and Goutte R. A comparison of bilinear space/spatial-frequency representations for texture discrimination. *Pattern Recognition Letters*, 16:1057–1068, 1995.