The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.

Chapter 2.1

Texture Analysis

Mihran Tuceryan
*Department of Computer and Information Science,
Indiana University -Purdue University at Indianapolis,
723 W. Michigan St. Indianapolis, IN 46202-5132*

and

Anil K. Jain
*Computer Science Department, Michigan State University
East Lansing, MI 48824-1027 USA
Internet: jain@cps.msu.edu*

This chapter reviews and discusses various aspects of texture analysis. The concentration is on the various methods of extracting textural features from images. The geometric, random field, fractal, and signal processing models of texture are presented. The major classes of texture processing problems such as segmentation, classification, and shape from texture are discussed. The possible application areas of texture such as automated inspection, document processing, and remote sensing are summarized. A bibliography is provided at the end for further reading.

Keywords: Texture, segmentation, classification, shape, signal processing, fractals, random fields, Gabor filters, wavelet transform, gray level dependency matrix.

1. Introduction

In many machine vision and image processing algorithms, simplifying assumptions are made about the uniformity of intensities in local image regions. However, images of real objects often do not exhibit regions of uniform intensities. For example, the image of a wooden surface is not uniform but contains variations of intensities which form certain repeated patterns called *visual texture*. The patterns can be the result of physical surface properties such as roughness or oriented strands which often have a tactile quality, or they could be the result of reflectance differences such as the color on a surface.

We recognize texture when we see it but it is very difficult to define. This difficulty is demonstrated by the number of different texture definitions attempted by vision researchers. Coggins [1] has compiled a catalogue of texture definitions in the computer vision literature and we give some examples here.

- “We may regard texture as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule.” [2]

- “A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic.” [3]

- “The image texture we consider is nonfigurative and cellular... An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of its (tonal) primitives... A fundamental characteristic of texture: it cannot be analyzed without a frame of reference of tonal primitive being stated or implied. For any smooth gray-tone surface, there exists a scale such that when the surface is examined, it has no texture. Then as resolution increases, it takes on a fine texture and then a coarse texture.” [4]

- “Texture is defined for our purposes as an attribute of a field having no components that appear enumerable. The phase relations between the components are thus not apparent. Nor should the field contain an obvious gradient. The intent of this definition is to direct attention of the observer to the global properties of the display — i.e., its overall “coarseness,” “bumpiness,” or “fineness.” Physically, nonenumerable (aperiodic) patterns are generated by stochastic as opposed to deterministic processes. Perceptually, however, the set of all patterns without obvious enumerable components will include many deterministic (and even periodic) textures.” [5]

- “Texture is an apparently paradoxical notion. On the one hand, it is commonly used in the early processing of visual information, especially for practical classification purposes. On the other hand, no one has succeeded in producing a commonly accepted definition of texture. The resolution of this paradox, we feel, will depend on a richer, more developed model for early visual information processing, a central aspect of which will be representational systems at many different levels of abstraction. These levels will most probably include actual intensities at the bottom and will progress through edge and orientation descriptors to surface, and perhaps volumetric descriptors. Given these multi-level structures, it seems clear that they should be included in the definition of, and in the computation of, texture descriptors.” [6]

- “The notion of texture appears to depend upon three ingredients: (i) some local ‘order’ is repeated over a region which is large in comparison to the order’s size, (ii) the order consists in the nonrandom arrangement of elementary parts, and (iii)

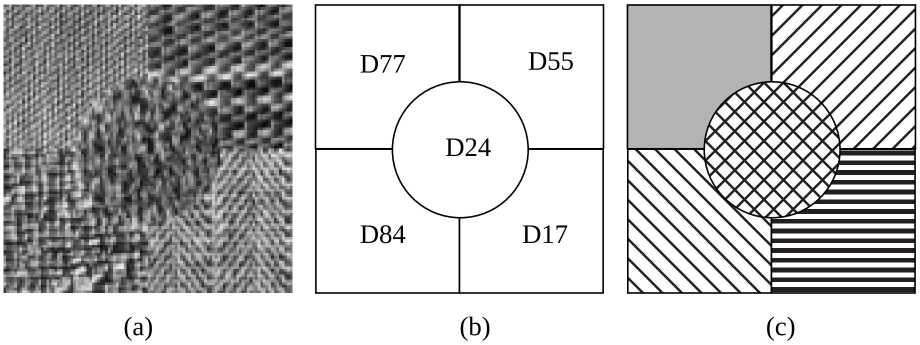


FIGURE 1. (a) An image consisting of five different textured regions: cotton canvas (D77), straw matting (D55), raffia (D84), herringbone weave (D17), and pressed calf leather. [8]. (b) The goal of texture classification is to label each textured region with the proper category label: the identities of the five texture regions present in (a). (c) The goal of texture segmentation is to separate the regions in the image which have different textures and identify the boundaries between them. The texture categories themselves need not be recognized. In this example, the five texture categories in (a) are identified as separate textures by the use of generic category labels (represented by the different fill patterns).

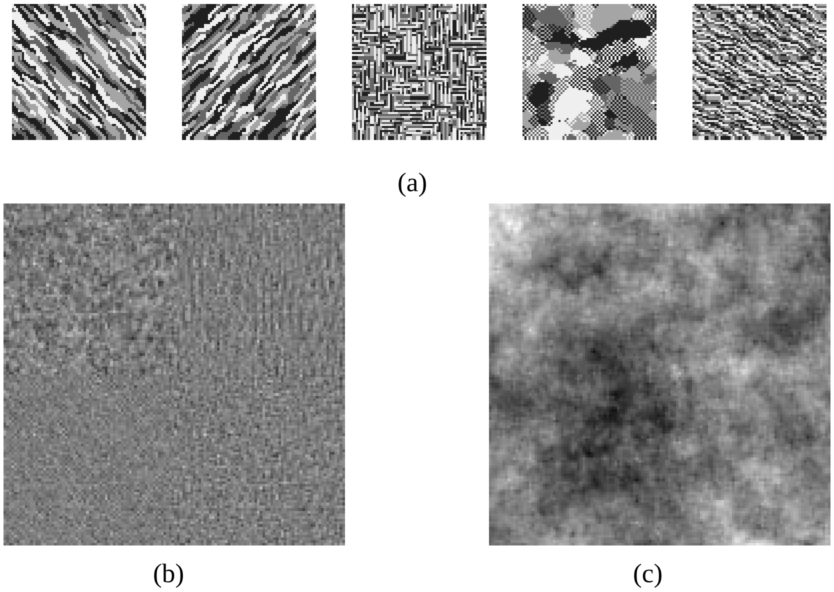


FIGURE 2. A set of example textures generated synthetically using only a small number of parameters. (a) Textures generated by discrete Markov random field models. (b) Four textures (in each of the four quadrants) generated by Gaussian Markov random field models. (c) Texture generated by fractal model.

> the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.” [7]

This collection of definitions demonstrates that the “definition” of texture is formulated by different people depending upon the particular application and that there is no generally agreed upon definition. Some are perceptually motivated, and others are driven completely by the application in which the definition will be used.

Image texture, defined as a function of the spatial variation in pixel intensities (gray values), is useful in a variety of applications and has been a subject of intense study by many researchers. One immediate application of image texture is the recognition of image regions using texture properties. For example, in Figure 1(a), we can identify the five different textures and their identities as cotton canvas, straw matting, raffia, herringbone weave, and pressed calf leather. Texture is the most important visual cue in identifying these types of homogeneous regions. This is called *texture classification*. The goal of texture classification then is to produce a classification map of the input image where each uniform textured region is identified with the texture class it belongs to as shown in Figure 1(b). We could also find the texture boundaries even if we could not classify these textured surfaces. This is then the second type of problem that texture analysis research attempts to solve — *texture segmentation*. The goal of texture segmentation is to obtain the boundary map shown in Figure 1(c). *Texture synthesis* is often used for image compression applications. It is also important in computer graphics where the goal is to render object surfaces which are as realistic looking as possible. Figure 2 shows a set of synthetically generated texture images using Markov random field and fractal models [9]. The *shape from texture* problem is one instance of a general class of vision problems known as “shape from X”. This was first formally pointed out in the perception literature by Gibson [10]. The goal is to extract three-dimensional shape information from various cues such as shading, stereo, and texture. The texture features (texture elements) are distorted due to

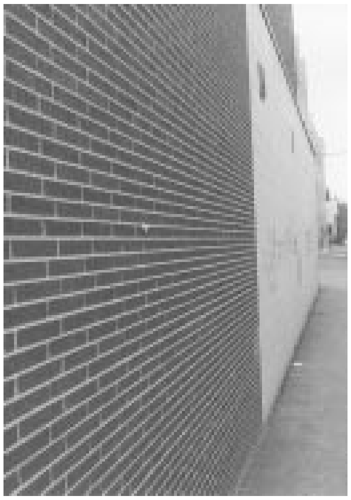


FIGURE 3. We can extract the orientation of the surface from the variations of texture (defined by the bricks) in this image.

the imaging process and the perspective projection which provide information about surface orientation and shape. An example of shape from texture is given in Figure 3.

2. Motivation

Texture analysis is an important and useful area of study in machine vision. Most natural surfaces exhibit texture and a successful vision system must be able to deal with the textured world surrounding it. This section will review the importance of texture perception from two viewpoints — from the viewpoint of human vision or psychophysics and from the viewpoint of practical machine vision applications.

2.1. Psychophysics

The detection of a tiger among the foliage is a perceptual task that carries life and death consequences for someone trying to survive in the forest. The success of the tiger in camouflaging itself is a failure of the visual system observing it. The failure is in not being able to separate *figure* from *ground*. Figure-ground separation is an issue which is of intense interest to psychophysicists. The figure-ground separation can be based on various cues such as brightness, form, color, texture, etc. In the example of the tiger in the forest, texture plays a major role. The camouflage is successful because the visual system of the observer is unable to discriminate (or segment) the two textures of the foliage and the tiger skin. What are the visual processes that allow one to separate figure from ground using the texture cue? This question is the basic motivation among psychologists for studying texture perception.

Another reason why it is important to study the psychophysics of texture perception is that the performance of various texture algorithms is evaluated against the performance of the human visual system doing the same task. For example, consider the texture pair in Figure 4(a), first described by Julesz [11]. The image consists of two regions each of which is made up of different texture tokens. Close scrutiny of the texture image will indicate this fact to the human observer. The immediate perception of the image, however, does not result in the perception of two different textured regions; instead only one uniformly textured region is perceived. Julesz says that such a texture pair is not “effortlessly discriminable” or “preattentively discriminable.” Such synthetic textures help us form hypotheses about what image properties are important in human texture perception. In addition, this example raises the question of how to evaluate the performance of computer algorithms that analyze textured images. For example, suppose we have an algorithm that can discriminate the texture pair in Figure 4(a). Is this algorithm “correct?” The answer, of course, depends on the goal of the algorithm. If it is a very special purpose algorithm that should detect such scrutably different regions, then it is performing correctly. On the other hand, if it is to be a computational model of how the human visual system processes texture, then it is performing incorrectly.

Julesz has studied texture perception extensively in the context of texture discrimination [11,12,13]. The question he posed was “When is a texture pair discriminable, given that they had the same brightness, contrast, and color?” Julesz concentrated on the spatial statistics of the image gray levels that are inherent in the definition of texture by keeping other illumination-related properties the same.

To discuss Julesz’s pioneering work, we need to define the concepts of first- and second-order spatial statistics.

(i) *First-order statistics* measure the likelihood of observing a gray value at a randomly-chosen location in the image. First-order statistics can be computed from the histogram of pixel intensities in the image. These depend only on individual pixel values and not on the interaction or co-occurrence of neighboring pixel values. The average intensity in an image is an example of the first-order statistic.

(ii) *Second-order statistics* are defined as the likelihood of observing a pair of gray values occurring at the endpoints of a dipole (or needle) of random length placed in the image at a random location and orientation. These are properties of pairs of pixel values.

Julesz conjectured that two textures are not preattentively discriminable if their second-order statistics are identical. This is demonstrated by the example in Figure 4(a). This image consists of a pair of textured regions whose second-order statistics are identical. The two textured regions are not preattentively discriminable. His later counter-examples to this conjecture were the result of a careful construction of texture pairs that have identical second-order statistics (see Figure 4(b)) [14,15,16].

Julesz proposed the “theory of textons” to explain the preattentive discrimination of texture pairs. Textons are visual events (such as collinearity, terminations, closure, etc.) whose presence is detected and used in texture discrimination. Terminations are endpoints of line segments or corners. Using his theory of textons, Julesz explained the examples in Figure 4 as follows. Recall that both texture images in Figure 4 have two regions that have identical second-order statistics. In Figure 4(a), the number of terminations in both the upper and lower regions is the same (i.e., the texton information in the two regions is not different), therefore the visual system is unable to preattentively discriminate the two textures. In Figure 4(b), on the other hand, the number of terminations in the upper half is three, whereas the number of terminations in the lower half is four. The difference in this texton makes the two textured regions discriminable. Caelli has also proposed the existence of perceptual analyzers by the visual system for detecting textons [17]. Beck *et al.* [18] have conducted experiments and argue that the perception of texture segmentation in certain types of patterns is primarily a function of spatial frequency analysis and not the result of higher level symbolic grouping processes.

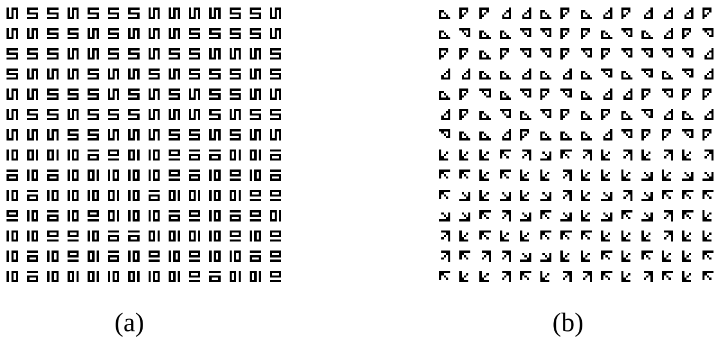
Studies in psychophysiology have suggested that a multi-channel, frequency and orientation analysis of the visual image formed on the retina is performed by the brain. Campbell and Robson [19] performed psychophysical experiments using various grating patterns. They suggested that the visual system decomposes the image into filtered images of various frequencies and orientations. De Valois *et al.* [20] have studied the brain of the macaque monkey which is assumed to be close to the human brain in its visual processing. They recorded the response of the simple cells in the visual cortex of the monkey to sinusoidal gratings of various frequencies and orientations and concluded that these cells are tuned to narrow ranges of frequency and orientation. These studies have motivated vision researchers to apply multi-channel filtering approaches to texture analysis.

2.2. Applications

Texture analysis methods have been utilized in a variety of application domains. In some of the mature domains (such as remote sensing) texture already has played a major role, while in other disciplines (such as surface inspection) new applications of texture are being found. We will briefly review the role of texture in automated inspection, medical image processing, document processing, and remote sensing. Images from two application domains are shown in Figure 5. The role that texture plays in these examples varies depending upon the application. For example, in the SAR images of Figures 5(b) and (c) texture is defined to be the local scene heterogeneity and this property is used for classification of land use categories such as water, agricultural areas, etc. In the ultrasound image of the heart in Figure 5(a), texture is defined as the amount of randomness which has a lower value in the vicinity of the border between the heart cavity and the inner wall than in the blood filled cavity. This fact can be used to perform segmentation and boundary detection using texture analysis methods.

2.2.1. Inspection

There has been a limited number of applications of texture processing to automated inspection problems. These applications include defect detection in images of textiles and automated inspection of carpet wear and automobile paints.



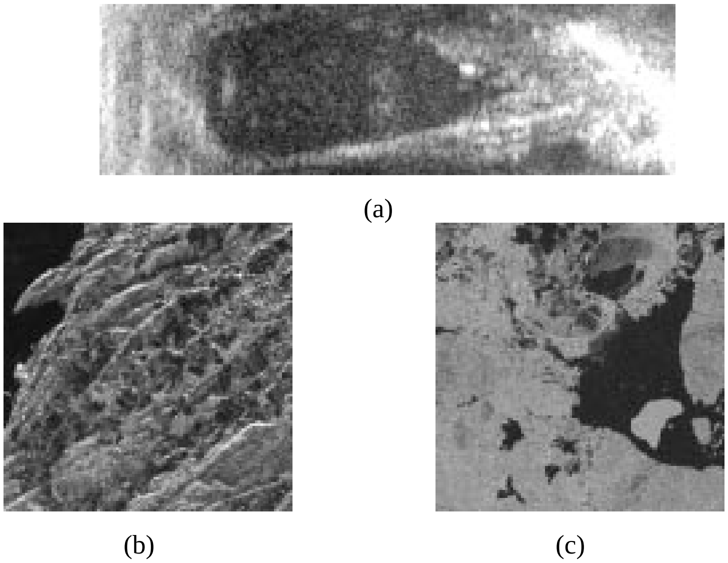
(a)

(b)

FIGURE 4. Texture pairs with identical second-order statistics. The bottom halves of the images consist of texture tokens that are different from the ones in the top half. (a) Humans cannot perceive the two regions without careful scrutiny. (b) The two different regions are immediately discriminable by humans.

In the detection of defects in texture images, most applications have been in the domain of textile inspection. Dewaele *et al.* [21] used signal processing methods to detect point defects and line defects in texture images. They have sparse convolution masks in which the bank of filters are adaptively selected depending upon the image to be analyzed. Texture features are computed from the filtered images. A Mahalanobis distance classifier is used to classify the defective areas. Chetverikov [22] defined a simple window differencing operator to the texture features obtained from simple filtering operations. This allows one to detect the boundaries of defects in the texture. Chen and Jain [23] used a structural approach to defect detection in textured images. They extract a skeletal structure from images, and by detecting anomalies in certain statistical features in these skeletons, defects in the texture are identified. Conners *et al*. [24] utilized texture analysis methods to detect defects in lumber wood automatically. The defect detection is performed by dividing the image into subwindows and classifying each subwindow into one of the defect categories such as knot, decay, mineral streak, etc. The features they use to perform this classification is based on tonal features such as mean, variance, skewness, and kurtosis of gray levels along with texture features computed from gray level co-occurrence matrices in analyzing pictures of wood. The combination of using tonal features along with textural features improves the correct classification rates over using either type of feature alone.

In the area of quality control of textured images, Siew *et al.* [25] proposed a method for the assessment of carpet wear. They used simple texture features that are computed from second-order gray level dependency statistics and from first-order gray level difference statistics. They showed that the numerical texture features obtained from these techniques can characterize the carpet wear successfully. Jain *et al*. [26] used the texture features computed from a bank of Gabor filters to automatically classify the quality of painted metallic surfaces. A pair of automotive paint finish images is shown in Figure 6 where the image in (a) has uniform coating of paint, but the image in (b) has “mottle” or “blotchy” appearance.



(a)

(b) (c)

FIGURE 5. Examples of images from various application domains in which texture analysis is important. (a) The ultrasound images of a heart. (b), (c) example aerial images using SAR sensors .

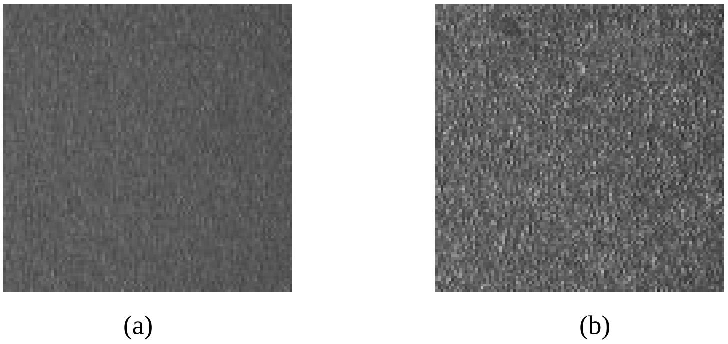


FIGURE 6. Example images used in paint inspection. (a) A non-defective paint which has a smooth texture. (b) A defective paint which has a mottled look.

2.2.2. Medical Image Analysis

Image analysis techniques have played an important role in several medical applications. In general, the applications involve the automatic extraction of features from the image which are then used for a variety of classification tasks, such as distinguishing normal tissue from abnormal tissue. Depending upon the particular classification task, the extracted features capture morphological properties, color properties, or certain textural properties of the image.

The textural properties computed are closely related to the application domain to be used. For example, Sutton and Hall [27] discuss the classification of pulmonary disease using texture features. Some diseases, such as interstitial fibrosis, affect the lungs in such a manner that the resulting changes in the X-ray images are texture changes as opposed to clearly delineated lesions. In such applications, texture analysis methods are ideally suited for these images. Sutton and Hall propose the use of three types of texture features to distinguish normal lungs from diseased lungs. These features are computed based on an isotropic contrast measure, a directional contrast measure, and a Fourier domain energy sampling. In their classification experiments, the best classification results were obtained using the directional contrast measure.

Harms *et al.* [28] used image texture in combination with color features to diagnose leukemic malignancy in samples of stained blood cells. They extracted texture micro-edges and “textons” between these micro-edges. The textons were regions with almost uniform color. They extracted a number of texture features from the textons including the total number of pixels in the textons which have a specific color, the mean texton radius and texton size for each color and various texton shape features. In combination with color, the texture features significantly improved the correct classification rate of blood cell types compared to using only color features.

Landeweerd and Gelsema [29] extracted various first-order statistics (such as mean gray level in a region) as well as second-order statistics (such as gray level co-occurrence matrices) to differentiate different types of white blood cells. Insana *et al.* [30] used textural features in ultrasound images to estimate tissue scattering parameters. They made significant use of the knowledge about the physics of the ultrasound imaging process and tissue characteristics to design the texture model. Chen *et al.* [31] used fractal texture features to classify ultrasound images of livers, and used the fractal texture features to do edge enhancement in chest X-rays.

Lundervold [32] used fractal texture features in combination with other features (such as response to edge detector operators) to analyze ultrasound images of the heart (see Figure 7). The ultrasound images in this study are time sequence images of the left ventricle of the heart. Figure 7 shows one frame in such a sequence. Texture is represented as an index at each pixel, being the local fractal dimension within an 11×11 window estimated according to the fractal Brownian motion model proposed by Chen *et al.* [31]. The texture feature is used in addition to a number of other traditional features, including the response to a Kirsch edge operator, the gray level, and the result of temporal operations. The fractal dimension is expected to be higher on an average in blood than in tissue due to the noise and backscatter characteristics of the blood which is more disordered than that of solid tissue. In addition, the fractal dimension is low at non-random blood/tissue interfaces representing edge information.

2.2.3. Document Processing

One of the useful applications of machine vision and image analysis has been in the area of document image analysis and character recognition. Document processing has applications ranging from postal address recognition to analysis and interpretation of maps. In many postal document processing applications (such as the recognition of destination address and zip code information on envelopes), the first step is the ability to separate the regions in the image which contain useful information from the background.

Most image analysis methods proposed to date for document processing are based upon the characteristics of printed documents and try to take advantage of these properties. For example, generally newspaper print is organized in rectangular blocks and this fact is used

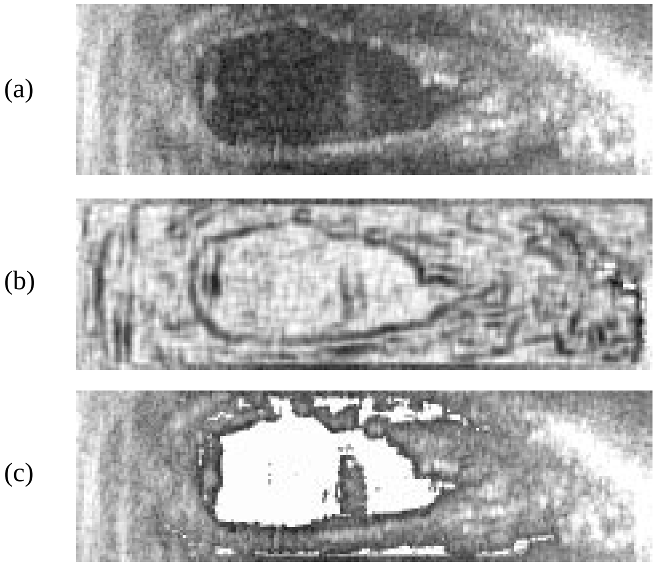
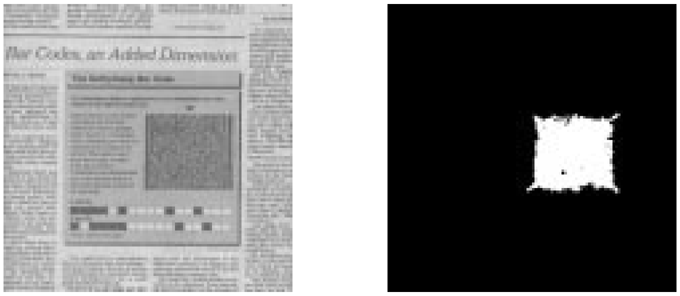


FIGURE 7. The processing of the ultrasound images of the heart using textural features. (a) A 128×432 ultrasound image from the left ventricle of the heart. (b) The fractal dimension used as the texture feature. The fractal dimension is lower at the walls of the ventricle. (c) Image segmentation from a k-means clustering algorithm. The white region is cluster 2 which corresponds to the blood. The clustering uses four features, one of which is the fractal texture feature.

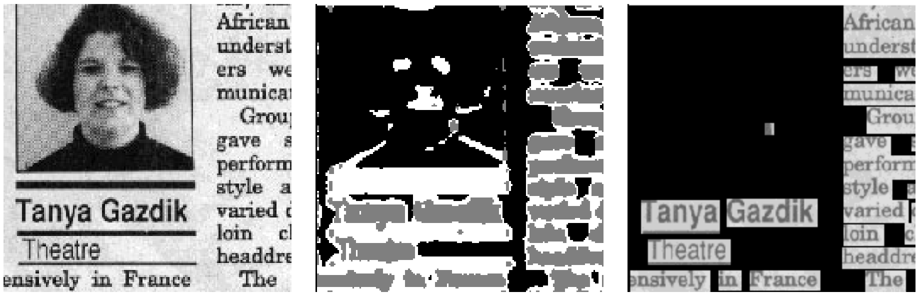


(a) (b)

FIGURE 8. Locating bar code in a newspaper image. (a) A scanned image of a newspaper that contains a bar code. (b) The two-class segmentation using Gabor filter features in [40]. The bar code region in the image has a distinct texture.

in a segmentation algorithm proposed in [33]. Many methods work on images based on precise algorithms which one might consider as having morphological characteristics. For example, Wang and Srihari [33] used projection profiles of the pixel values to identify large "text" blocks by detecting valleys in these profiles. Wahl *et al.* [34] used constrained run lengths and connected component analysis to detect blocks of text. Fletcher and Kasturi [35] used the fact that most text blocks lie in a straight line, and utilized Hough transform techniques to detect collinear elements. Taxt *et al.* [36] view the identification of print in document images as a two-category classification problem, where the categories are print and background. They use various classification methods to compute the segmentation including contextual classification and relaxation algorithms.

One can also use texture segmentation methods for preprocessing document images to identify regions of interest [37,38,39]. An example of this can be seen in Figure 8. The texture segmentation algorithm described in [40] was used to segment a newspaper image. In the resulting segmentation, one of the regions identified as having a uniform texture, which is different from its surrounding texture, is the bar code block. A similar method is used for locating text blocks in newspapers. A segmentation of the document image is obtained using three classes of textures: one class for the text regions, a second class for the uniform regions that form the background or images where intensities vary slowly, and a third class for the transition areas between the two types of regions (see Figure 9). The text regions are characterized by their high frequency content.



(a) (b) (c)

FIGURE 9. Text/graphics separation using texture information. (a) An image of a newspaper captured by a flatbed scanner. (b) The three-class segmentation obtained by the Gabor filter based texture segmentation algorithm. (c) The regions identified as text.

2.2.4. Remote Sensing

Texture analysis has been extensively used to classify remotely sensed images. Land use classification where homogeneous regions with different *types* of terrains (such as wheat, bodies of water, urban regions, etc.) need to be identified is an important application. Haralick *et al.* [41] used gray level co-occurrence features to analyze remotely sensed images. They computed gray level co-occurrence matrices for a distance of one with four directions ($0°$, $45°$, $90°$, and $135°$). For a seven-class classification problem, they obtained approximately 80% classification accuracy using texture features.

Rignot and Kwok [42] have analyzed SAR images using texture features computed from gray level co-occurrence matrices. However, they supplement these features with knowledge about the properties of SAR images. For example, image restoration algorithms were used to eliminate the specular noise present in SAR images in order to improve classification results. The use of various texture features was studied for analyzing SAR images by Schistad and Jain [43]. SAR images shown in Figures 5(b) and (c) were used to identify land use categories of water, agricultural areas, urban areas, and other areas. Fractal dimension, autoregressive Markov random field model, and gray level co-occurrence texture features were used in the classification. The classification errors ranged from 25% for the fractal based models to as low as 6% for the MRF features. Du [44] used texture features derived from Gabor filters to segment SAR images. He successfully segmented the SAR images into categories of water, new forming ice, older ice, and multi-year ice. Lee and Philpot [45] also used spectral texture features to segment SAR images.

3. A Taxonomy of Texture Models

Identifying the perceived qualities of texture in an image is an important first step towards building mathematical models for texture. The intensity variations in an image which characterize texture are generally due to some underlying physical variation in the scene (such as pebbles on a beach or waves in water). Modelling this physical variation is very difficult, so texture is usually characterized by the two-dimensional variations in the intensities present in the image. This explains the fact that no precise, general definition of texture exists in the computer vision literature. In spite of this, there are a number of intuitive properties of texture which are generally assumed to be true.

- Texture is a property of areas; the texture of a point is undefined. So, texture is a contextual property and its definition must involve gray values in a spatial neighborhood. The size of this neighborhood depends upon the texture type, or the size of the primitives defining the texture.

- Texture involves the spatial distribution of gray levels. Thus, two-dimensional histograms or co-occurrence matrices are reasonable texture analysis tools.

- Texture in an image can be perceived at different scales or levels of resolution [10]. For example, consider the texture represented in a brick wall. At a coarse resolution, the texture is perceived as formed by the individual bricks in the wall; the interior details in the brick are lost. At a higher resolution, when only a few bricks are in the field of view, the perceived texture shows the details in the brick.

- A region is perceived to have texture when the number of primitive objects in the region is large. If only a few primitive objects are present, then a group of countable objects is perceived instead of a textured image. In other words, a texture is perceived when significant individual “forms” are not present.

Image texture has a number of perceived qualities which play an important role in describing texture. Laws [47] identified the following properties as playing an important role in describing texture: uniformity, density, coarseness, roughness, regularity, linearity, directionality, direction, frequency, and phase. Some of these perceived qualities are not independent. For example, frequency is not independent of density and the property of direction only applies to directional textures. The fact that the perception of texture has so many different dimensions is an important reason why there is no single method of texture representation which is adequate for a variety of textures.

3.1. Statistical Methods

One of the defining qualities of texture is the spatial distribution of gray values. The use of statistical features is therefore one of the early methods proposed in the machine vision literature. In the following, we will use $\{I(x, y), 0 \le x \le N-1, 0 \le y \le N-1\}$ to denote an $N \times N$ image with G gray levels. A large number of texture features have been proposed. But, these features are not independent as pointed out by Tomita and Tsuji [46]. The relationship between the various statistical texture measures and the input image is summarized in Figure 10 [46]. Picard [48] has also related the gray level co-occurrence matrices to the Markov random field models.

3.1.1. Co-occurrence Matrices

Spatial gray level co-occurrence estimates image properties related to second-order statistics. Haralick [10] suggested the use of gray level co-occurrence matrices (GLCM) which have become one of the most well-known and widely used texture features. The $G \times G$ gray level co-occurrence matrix $P_{\mathbf{d}}$ for a displacement vector $\mathbf{d} = (dx, dy)$ is defined as follows. The entry (i, j) of $P_{\mathbf{d}}$ is the number of occurrences of the pair of gray levels i and j which are a distance $\mathbf{d}$ apart. Formally, it is given as

$$P_{\mathbf{d}}(i, j) = |\{((r, s), (t, v)) : I(r, s) = i, I(t, v) = j\}| \tag{11.1}$$

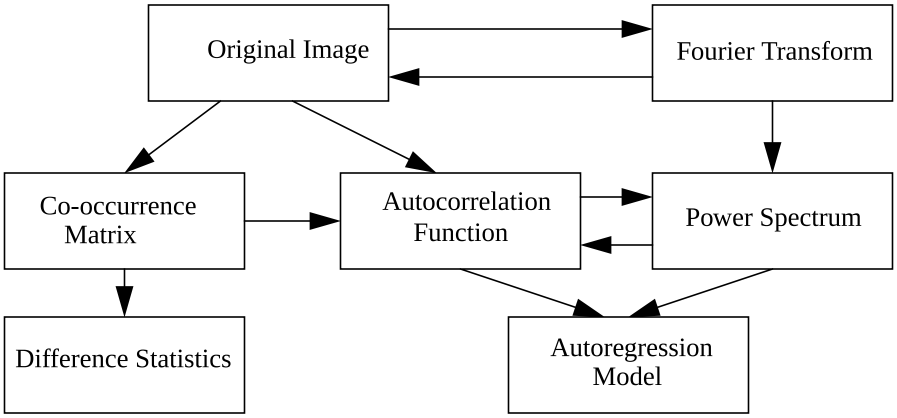


FIGURE 10. The interrelation between the various second-order statistics and the input image [46]. © Reprinted by permission of Kluwer Academic Publishers.

where $(r, s), (t, v) \in N \times N$, $(t, v) = (r + dx, s + dy)$, and $|.|$ is the cardinality of a set.

As an example, consider the following 4×4 image containing 3 different gray values:

$$\begin{matrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{matrix}$$

The 3×3 gray level co-occurrence matrix for this image for a displacement vector of $\mathbf{d} = (1, 0)$ is given as follows:

$$P_{\mathbf{d}} = \begin{bmatrix} 4 & 0 & 2 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Here the entry $(0, 0)$ of $P_{\mathbf{d}}$ is 4 because there are four pixel pairs of $(0, 0)$ that are offset by $(1, 0)$ amount. Examples of $P_{\mathbf{d}}$ for other displacement vectors is given below.

$\mathbf{d} = (0, 1)$	$\mathbf{P_d} = \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$
$\mathbf{d} = (1, 1)$	$\mathbf{P_d} = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

Notice that the co-occurrence matrix so defined is not symmetric. But a symmetric co-occurrence matrix can be computed by the formula $P = P_{\mathbf{d}} + P_{-\mathbf{d}}$. The co-occurrence matrix reveals certain properties about the spatial distribution of the gray levels in the texture image. For example, if most of the entries in the co-occurrence matrix are concentrated along the diagonals, then the texture is coarse with respect to the displacement vector **d**. Haralick has proposed a number of useful texture features that can be computed from the co-occurrence matrix. Table 1 lists some of these features.

Here μ_x and μ_y are the means and σ_x and σ_y are the standard deviations of $P_{\mathbf{d}}(x)$ and $P_{\mathbf{d}}(y)$, respectively, where $P_{\mathbf{d}}(x) = \sum_j P_{\mathbf{d}}(x, j)$ and $P_{\mathbf{d}}(y) = \sum_i P_{\mathbf{d}}(i, y)$.

The co-occurrence matrix features suffer from a number of difficulties. There is no well established method of selecting the displacement vector $\mathbf{d}$ and computing co-occurrence matrices for different values of $\mathbf{d}$ is not feasible. For a given $\mathbf{d}$, a large number of features can be computed from the co-occurrence matrix. This means that some sort of feature selection method must be used to select the most relevant features. The co-occurrence matrix-based texture features have also been primarily used in texture classification tasks and not in segmentation tasks.

3.1.2. Autocorrelation Features

An important property of many textures is the repetitive nature of the placement of texture elements in the image. The autocorrelation function of an image can be used to assess the amount of regularity as well as the fineness/coarseness of the texture present in the image. Formally, the autocorrelation function of an image $I(x, y)$ is defined as follows:

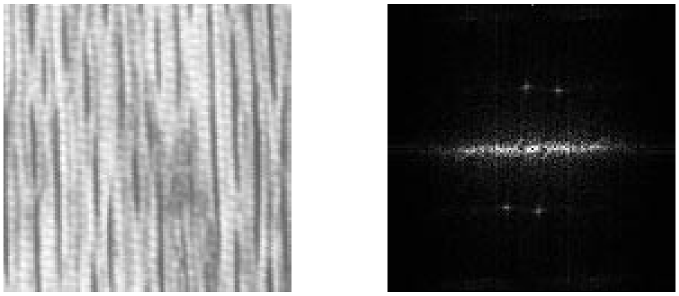
$$\rho(x, y) = \frac{\sum_{u=0}^{N} \sum_{v=0}^{N} I(u, v) I(u + x, v + y)}{\sum_{u=0}^{N} \sum_{v=0}^{N} I^2(u, v)} \tag{11.2}$$

The image boundaries must be handled with special care but we omit the details here. This function is related to the size of the texture primitive (i.e., the fineness of the texture). If the texture is coarse, then the autocorrelation function will drop off slowly; otherwise, it will drop off very rapidly. For regular textures, the autocorrelation function will exhibit peaks and valleys.

The autocorrelation function is also related to the power spectrum of the Fourier transform (see Figure 10). Consider the image function in the spatial domain $I(x, y)$ and its Fourier transform $F(u, v)$. The quantity $|F(u, v)|^2$ is defined as the power spectrum where $|.|$ is the modulus of a complex number. The example in Figure 11 illustrates the effect of the directionality of a texture on the distribution of energy in the power spectrum. Early

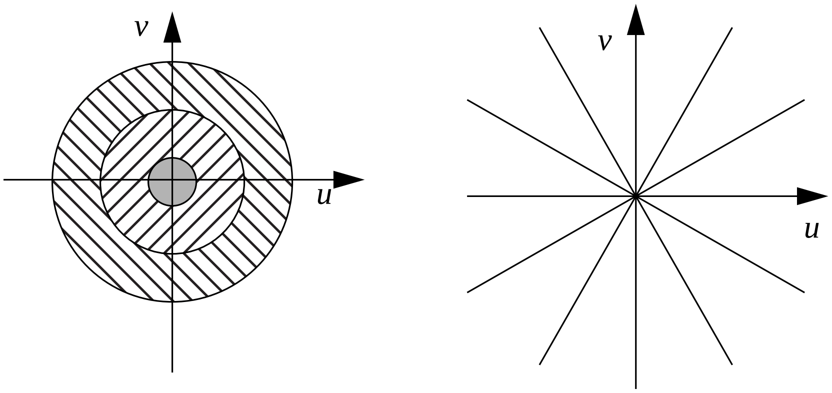
TABLE 1. Some texture features extracted from gray level co-occurrence matrices.

Texture Feature	**Formula**
Energy	$\sum_i \sum_j P_{\mathbf{d}}^2(i, j)$
Entropy	$-\sum_i \sum_j P_{\mathbf{d}}(i, j) \log P_{\mathbf{d}}(i, j)$
Contrast	$\sum_i \sum_j (i - j)^2 P_{\mathbf{d}}(i, j)$
Homogeneity	$\sum_i \sum_j \frac{P_{\mathbf{d}}(i, j)}{1 + \lvert i - j \rvert}$
Correlation	$\frac{\sum_i \sum_j (i - \mu_x)(j - \mu_y) P_{\mathbf{d}}(i, j)}{\sigma_x \sigma_y}$



(a) (b)

FIGURE 11. Texture features from the power spectrum. (a) A texture image, and (b) its power spectrum. The directional nature of this texture is reflected in the directional distribution of energy in the power spectrum.



$$f_{r_1, r_2} = \int_0^{2\pi}\int_{r_2}^{r_1} |F(u, v)|^2 dr d\theta \qquad f_{\theta_1, \theta_2} = \int_{\theta_1}^{\theta_2}\int_0^{\infty} |F(u, v)|^2 dr d\theta$$

$$r = \sqrt{u^2 + v^2} \qquad \theta = \operatorname{atan}(v/u) \qquad r = \sqrt{u^2 + v^2} \qquad \theta = \operatorname{atan}(v/u)$$

(a) (b)

FIGURE 12. Texture features computed from the power spectrum of the image. (a) The energy computed in each shaded band is a texture feature indicating coarseness/fineness, and (b) the energy computed in each wedge is a texture feature indicating directionality.

approaches using such spectral features would divide the frequency domain into rings (for frequency content) and wedges (for orientation content) as shown in Figure 12. The frequency domain is thus divided into regions and the total energy in each of these regions is computed as texture features.

3.2. Geometrical Methods

The class of texture analysis methods that falls under the heading of geometrical methods is characterized by their definition of texture as being composed of “texture elements” or primitives. The method of analysis usually depends upon the geometric properties of these texture elements. Once the texture elements are identified in the image, there are two major approaches to analyzing the texture. One computes statistical properties from the extracted texture elements and utilizes these as texture features. The other tries to extract the placement rule that describes the texture. The latter approach may involve geometric or syntactic methods of analyzing texture.

3.2.1. Voronoi tessellation Features

Tuceryan and Jain [49] proposed the extraction of texture tokens by using the properties of the Voronoi tessellation of the given image. Voronoi tessellation has been proposed because of its desirable properties in defining local spatial neighborhoods and because the local spatial distributions of tokens are reflected in the shapes of the Voronoi polygons. First, texture tokens are extracted and then the tessellation is constructed. Tokens can be as simple as points of high gradient in the image or complex structures such as line segments or closed boundaries.

In computer vision, the Voronoi tessellation was first proposed by Ahuja as a model for defining “neighborhoods” [50]. Suppose that we are given a set S of three or more tokens (for simplicity, we will assume that a token is a point) in the Euclidean plane. Assume that these points are not all collinear, and that no four points are cocircular. Consider an arbitrary pair of points P and Q. The bisector of the line joining P and Q is the locus of points equidistant from both P and Q and divides the plane into two halves. The half plane H_P^Q (H_Q^P) is the locus of points closer to P (Q) than to Q (P). For any given point P, a set of such half planes is obtained for various choices of Q. The intersection $\bigcap_{Q \in S, Q \neq P} H_P^Q$ defines a polygonal region consisting of points closer to P than any other point. Such a region is called the Voronoi polygon [51] associated with the point. The set of complete polygons is called the *Voronoi diagram* of S [52]. The Voronoi diagram together with the incomplete polygons in the convex hull define a *Voronoi tessellation* of the entire plane. Two points are said to be *Voronoi neighbors* if the Voronoi polygons enclosing them share a common edge. The dual representation of the Voronoi tessellation is the *Delaunay graph* which is obtained by connecting all the pairs of points which are Voronoi neighbors as defined above. An optimal algorithm to compute the Voronoi tessellation for a point pattern is described by Preparata and Shamos [53]. A simple 2D dot pattern and its Voronoi tessellation are shown in Figure 13.

The neighborhood of a token P is defined by the Voronoi polygon containing P. Many of the perceptually significant characteristics of a token’s environment are manifest in the geometric properties of the Voronoi neighborhoods (see Figure 13). The geometric properties of the Voronoi polygons are used as texture features.

In order to apply geometrical methods to gray level images, we need to first extract tokens from images. We use the following simple algorithm to extract tokens from input gray level textural images.

1. Apply a Laplacian-of-Gaussian (LoG or $\nabla^2 G$) filter to the image. For computational efficiency, the $\nabla^2 G$ filter can be approximated with a difference of Gaussians (DoG) filter. The size of the DoG filter is determined by the sizes of the two Gaussian filters. Tuceryan and Jain used $\sigma_1 = 1$ for the first Gaussian and $\sigma_2 = 1.6\sigma_1$ for the second. According to Marr, this is the ratio at which a DoG filter best approximates the corresponding $\nabla^2 G$ filter [54].

2. Select those pixels that lie on a local intensity maximum in the filtered image. A pixel in the filtered image is said to be on a local maximum if its magnitude is larger than six or more of its eight nearest neighbors. This results in a binary image. For example, applying steps 1 and 2 to the image in Figure 14(a) yields the binary image in Figure 14(b).

3. Perform a connected component analysis on the binary image using eight nearest neighbors. Each connected component defines a texture primitive (token).

The Voronoi tessellation of the resulting tokens is constructed. Features of each Voronoi cell are extracted and tokens with similar features are grouped to construct uniform texture regions. Moments of area of the Voronoi polygons serve as a useful set of features that reflect both the spatial distribution and shapes of the tokens in the textured image. The $(p+q)^{th}$ order moments of area of a closed region R with respect to a token with coordinates (x_0, y_0) are defined as [55]:

$$m_{pq} = \iint_R (x - x_0)^p (y - y_0)^q dxdy \tag{11.3}$$

where $p + q = 0, 1, 2, \ldots$. A description of the five features used is given in Table 2 where $(\bar{x}, \bar{y})$ are the coordinates of the Voronoi polygon's centroid.

The texture features based on Voronoi polygons have been used for segmentation of textured images. The segmentation algorithm is edge based, using a statistical comparison of the neighboring collections of tokens. A large dissimilarity among the texture features is evidence for a texture edge. This algorithm has successfully segmented gray level texture images as well as a number of synthetic textures with identical second-order statistics.

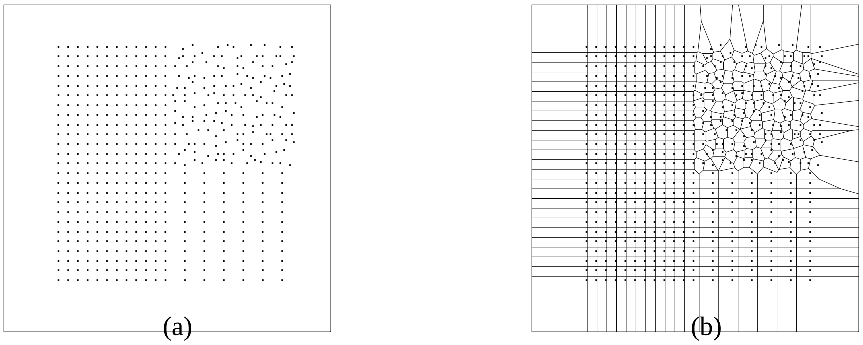


FIGURE 13. Voronoi tessellation: (a) An example dot pattern, and (b) its Voronoi tessellation.

TABLE 2. Voronoi polygon features used by the texture segmentation algorithm [49]. Here, f_2 gives the magnitude of the vector from the token to the polygon centroid, f_3 gives its direction, f_4 gives the overall elongation of the polygon ($f_4 = 0$ for a circle), and f_5 gives the orientation of its major axis. $(\bar{x}, \bar{y})$ are the coordinates of the Voronoi polygon's centroid.

Texture Feature	**Computation**
f_1	m_{00}
f_2	$\sqrt{\bar{x}^2 + \bar{y}^2}$
f_3	$\mathrm{atan}(\bar{y}/\bar{x})$
f_4	$\dfrac{\sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}$
f_5	$\mathrm{atan}\left(\dfrac{2m_{11}}{m_{20} - m_{02}}\right)$

Figure 14(a) shows an example texture pair and Figure 14(c) shows the resulting segmentation.

3.2.2. Structural Methods

The structural models of texture assume that textures are composed of texture primitives. The texture is produced by the placement of these primitives according to certain placement rules. This class of algorithms, in general, is limited in power unless one is dealing with very regular textures. Structural texture analysis consists of two major steps: (a) extraction of the texture elements, and (b) inference of the placement rule.

There are a number of ways to extract texture elements in images. It is useful to define what is meant by texture elements in this context. Usually texture elements consist of regions in the image with uniform gray levels. Voorhees and Poggio [56] argued that blobs are important in texture perception. They have proposed a method based on filtering the image with Laplacian of Gaussian (LoG) masks at different scales and combining this information to extract the blobs in the image. Blostein and Ahuja [57] perform similar processing in order to extract texture tokens in images by examining the response of the LoG filter at multiple scales. They integrate their multi-scale blob detection with surface shape computation in order to improve the results of both processes. Tomita and Tsuji [46] also suggest a method of computing texture tokens by doing a medial axis transform on the connected components of a segmented image. They then compute a number of properties such as intensity and shapes of these detected tokens.

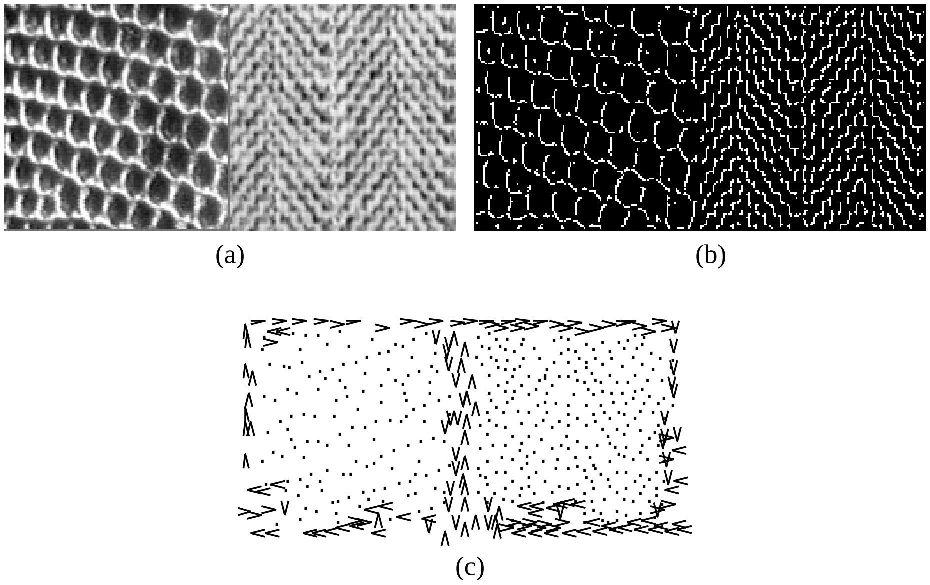


FIGURE 14. Texture segmentation using the Voronoi tessellation. (a) An example texture pair from Brodatz's album [92], (b) the peaks detected in the filtered image, and (c) the segmentation using the texture features obtained from Voronoi polygons [49]. The arrows indicate the border direction. The interior is on the right when looking in the direction of the arrow.

Zucker [58] has proposed a method in which he regards the observable textures (real textures) as distorted versions of ideal textures. The placement rule is defined for the ideal texture by a graph that is isomorphic to a regular or semiregular tessellation. These graphs are then transformed to generate the observable texture. Which of the regular tessellations is used as the placement rule is inferred from the observable texture. This is done by computing a two-dimensional histogram of the relative positions of the detected texture tokens.

Another approach to modeling texture by structural means is described by Fu [59]. In this approach the texture image is regarded as texture primitives arranged according to a placement rule. The primitive can be as simple as a single pixel that can take a gray value, but it is usually a collection of pixels. The placement rule is defined by a tree grammar. A texture is then viewed as a string in the language defined by the grammar whose terminal symbols are the texture primitives. An advantage of this method is that it can be used for texture generation as well as texture analysis. The patterns generated by the tree grammars could also be regarded as ideal textures in Zucker's model.

3.3. Model Based Methods

Model based texture analysis methods are based on the construction of an image model that can be used not only to describe texture, but also to synthesize it. The model parameters capture the essential perceived qualities of texture.

3.3.1. Random Field Models

Markov random fields (MRFs) have been popular for modeling images. They are able to capture the local (spatial) contextual information in an image. These models assume that the intensity at each pixel in the image depends on the intensities of only the neighboring pixels. MRF models have been applied to various image processing applications such as texture synthesis [60], texture classification [61,62], image segmentation [63,64], image restoration [65], and image compression.

The image is usually represented by an $N \times N$ lattice denoted by $L = \{(i, j) | 1 \le i \le M, 1 \le j \le N\}$. $I(i, j)$ is a random variable which represents the gray level at pixel (i, j) on lattice *L*. The indexing of the lattice is simplified for mathematical convenience to I_t with $t = (i-1)N + j$. Let *A* be the range set common to all random variables I_t and let $\Omega = \{(x_1, x_2, \ldots, x_{MN}) \middle| x_t \in A, \forall t\}$ denote the set of all labellings of *L*. Note that *A* is specified according to the application. For instance, for an image with 256 different gray levels *A* may be the set $\{0, 1, \ldots, 255\}$. The random vector $I = (I_1, I_2, \ldots, I_{MN})$ denotes a coloring of the lattice. A discrete Markov random field is a random field whose probability mass function has the properties of positivity, Markovianity, and homogeneity.

The neighbor set of a site *t* can be defined in different ways. The first-order neighbors of *t* are its four-connected neighbors and the second-order neighbors are its eight-connected neighbors. Within these neighborhoods, sets of neighbors which form cliques (single site, pairs, triples, and quadruples) are usually used in the definition of the conditional probabilities.

A discrete Gibbs random field (GRF) assigns a probability mass function to the entire lattice:

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} e^{-U(\mathbf{x})} \qquad \forall \mathbf{x} \in \Omega \tag{11.4}$$

where $U(\mathbf{x})$ is an energy function and Z is a normalizing constant called the partition function. The energy function is usually specified in terms of cliques formed over neighboring pixels. For the second-order neighbors the possible cliques are given in Figure 15. The energy function then is expressed in terms of potential functions $V_C(.)$ over the cliques Q:

$$U(\mathbf{x}) = \sum_{c \in Q} V_C(\mathbf{x}) \tag{11.5}$$

We have the property that with respect to a neighborhood system, there exists a unique Gibbs random field for every Markov random field and there exists a unique Markov random field for every Gibbs random field [66]. The consequence of this theorem is that one can model the texture either globally by specifying the total energy of the lattice or model it locally by specifying the local interactions of the neighboring pixels in terms of the conditional probabilities.

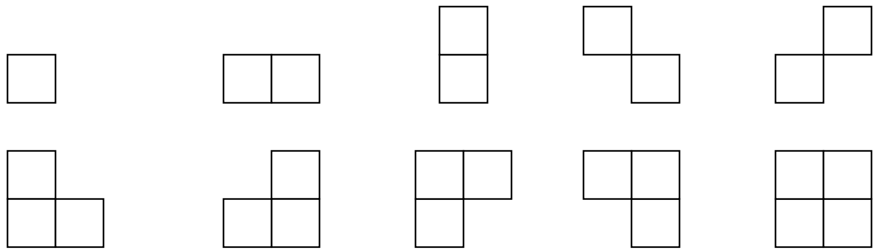


FIGURE 15. The clique types for the second-order neighborhood.

There are a number of ways in which textures are modeled using Gibbs random fields. Among these are the Derin-Elliot model [67] and the auto-binomial model [66, 60] which are defined by considering only the single pixel and pairwise pixel cliques in the second-order neighbors of a site. In both models the conditional probabilities are given by expressions of the following form:

$$P(x_t|R_t) = \frac{1}{Z_t}e^{-\mathbf{w}(x_t|R_t)^T\theta} \tag{11.6}$$

where $Z_t = \sum_{g \in A} e^{-\mathbf{w}(g, R_t)^T\theta}$ is the normalization constant. The energy of the Gibbs random field is given by:

$$U(\mathbf{x}) = \frac{1}{2}\sum_{t=1}^{MN} \mathbf{w}(x_t, R_t)^T\theta \tag{11.7}$$

where $\mathbf{w}(x_t, R_t) = \left[w_1(x_t)\ w_2(x_t)\ w_3(x_t)\ w_4(x_t)\right]^T$ and $\theta = \left[\theta_1\ \theta_2\ \theta_3\ \theta_4\right]^T$. The two models define the components of the $\mathbf{w}$ vector, $w_r(x_t)$, differently as follows:

Derin-Elliott model: $w_r(x_t) = I(x_t, x_{t-r}) + I(x_t, x_{t+r}) \qquad 1 \le r \le 4$.

Auto-binomial model: $w_r(x_t) = x_t(x_{t-r} + x_{t+r}) \qquad 1 \le r \le 4$.

Here r is the index that defines the set of neighboring pixels of a site t, and $I(a, b)$ is an indicator function as follows:

$$I(a, b) = \begin{cases} -1 & \text{if } a = b \\ 1 & \text{otherwise} \end{cases} \tag{11.8}$$

The vector θ is the set of parameters that define and model the textural properties of the image. In texture synthesis problems, the values are set to control the type of texture to be generated. In the classification and segmentation problems, the parameters need to be estimated in order to process the textures images. Textures were synthesized using this method by Cross and Jain [60]. Model parameters were also estimated for a set of natural textures. The estimated parameters were used to generate synthetic textures and the results were compared to the original images. The models captured microtextures well, but they failed with regular and inhomogeneous textures.

3.3.2. Fractals

Many natural surfaces have a statistical quality of roughness and self-similarity at different scales. Fractals are very useful and have become popular in modeling these properties in image processing. Mandelbrot [68] proposed fractal geometry and is the first one to notice its existence in the natural world.

We first define a deterministic fractal in order to introduce some of the fundamental concepts. Self-similarity across scales in fractal geometry is a crucial concept. A deterministic fractal is defined using this concept of self-similarity as follows. Given a bounded set *A* in a Euclidean n-space, the set *A* is said to be self-similar when *A* is the union of *N* distinct (non-overlapping) copies of itself, each of which has been scaled down by a ratio of *r*. The fractal dimension *D* is related to the number *N* and the ratio *r* as follows:

$$D = \frac{\log N}{\log(1/r)} \tag{11.9}$$

The fractal dimension gives a measure of the roughness of a surface. Intuitively, the larger the fractal dimension, the rougher the texture is. Pentland [69] has argued and given evidence that images of most natural surfaces can be modeled as spatially isotropic fractals. Most natural surfaces and in particular textured surfaces are not deterministic as described above but have a statistical variation. This makes the computation of fractal dimension more difficult.

There are a number of methods proposed for estimating the fractal dimension *D*. One method is the estimation of the box dimension as follows [70]. Given a bounded set *A* in Euclidean n-space, consider boxes of size L_{max} on a side which cover the set *A*. A scaled down version of the set A by ratio *r*, will result in $N = 1/r^D$ similar sets. This new set can be covered by boxes of size $L = rL_{max}$. The number of such boxes then is related to the fractal dimension by

$$N(L) = \frac{1}{r^D} = \left[\frac{L_{max}}{L}\right]^D \tag{11.10}$$

The fractal dimension is then estimated from Equation (11.10) by the following procedure. For a given *L*, divide the n-space into a grid of boxes of size *L* and count the number of boxes covering *A*. Repeat this procedure for different values of *L*. Then estimate the value of the fractal dimension *D* from the slope of the line

$$\ln(N(L)) = -D\ln(L) + D\ln(L_{max}) \tag{11.11}$$

This can be accomplished by computing the least squares linear fit to the data, namely, a plot of $\ln(L)$ vs. $-\ln(N(L))$.

An improved method of estimating the fractal dimension was proposed by Voss [71]. Assume we are estimating the fractal dimension of an image surface *A*. Let $P(m, L)$ be the probability that there are *m* points within a box of side length *L* centered at an arbitrary point on the surface *A*. Let *M* be the total number of points in the image. When one overlays the image with boxes of side length *L*, then the $(M/m)P(m, L)$ is the expected number of boxes with *m* points inside. The expected total number of boxes needed to cover the whole image is

$$E[N(L)] = M\sum_{m=1}^{N}(1/m)P(m, L) \tag{11.12}$$

The expected value of $N(L)$ is proportional to L^{-D} and thus can be used to estimate the fractal dimension *D*. Other methods have also been proposed for estimating the fractal dimension. For example, Super and Bovik [72] have proposed using Gabor filters and signal processing methods to estimate the fractal dimension in textured images.

The fractal dimension is not sufficient to capture all textural properties. It has been shown [70] that there may be perceptually very different textures that have very similar fractal dimensions. Therefore, another measure, called *lacunarity* [68,71,70], has been suggested in order to capture the textural property that will let one distinguish between such textures. Lacunarity is defined as

$$\Lambda = E\left[\left(\frac{M}{E(M)} - 1\right)^2\right] \tag{11.13}$$

where *M* is the mass of the fractal set and $E(M)$ is the expected value of the mass. This measures the discrepancy between the actual mass and the expected value of the mass. Lacunarity is small when texture is fine and it is large when the texture is coarse. The mass of the fractal set is related to the length *L* by the power law:

$$M(L) = KL^D \tag{11.14}$$

Voss [71] suggested computing lacunarity from the probability distribution $P(m, L)$ as follows. Let $M(L) = \sum_{m=1}^{N} mP(m, L)$ and $M^2(L) = \sum_{m=1}^{N} m^2P(m, L)$. Then lacunarity Λ is defined as:

$$\Lambda(L) = \frac{M^2(L) - (M(L))^2}{(M(L))^2} \tag{11.15}$$

where $E[M(L)]$ is the expected value of $M(L)$. This measure of the image is then used as a texture feature in order to perform texture segmentation or classification.

Ohanian and Dubes [73] have studied the performance of various texture features. They studied the texture features with the performance criteria “which features optimized the classification rate?” They compared four fractal features, sixteen co-occurrence features, four Markov random field features, and Gabor features. They used Whitney’s forward selection method for feature selection. The evaluation was done on four classes of images: Gauss Markov random field images, fractal images, leather images, and painted surfaces.

The co-occurrence features generally outperformed other features (88% correct classification) followed by fractal features (84% classification). Using both fractal and co-occurrence features improved the classification rate to 91%. Their study did not compare the texture features in segmentation tasks. It also used the energy from the raw Gabor filtered images instead of using the empirical nonlinear transformation needed to obtain the texture features as suggested in [40] (see also Section 3.4.3).

3.4. Signal Processing Methods

Psychophysical research has given evidence that the human brain does a frequency analysis of the image [19,74]. Texture is especially suited for this type of analysis because of its properties. This section will review the various techniques of texture analysis that rely on signal processing techniques. Most techniques try to compute certain features from filtered images which are then used in either classification or segmentation tasks.

3.4.1. Spatial Domain Filters

Spatial domain filters are the most direct way to capture image texture properties. Earlier attempts at defining such methods concentrated on measuring the edge density per unit area. Fine textures tend to have a higher density of edges per unit area than coarser textures. The measurement of edgeness is usually computed by simple edge masks such as the Robert's operator or the Laplacian operator [10,47]. The two orthogonal masks for the Robert's operator and one digital realization of the Laplacian are given below.

Roberts Operators

Laplacian Operator

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

The edgeness measure can be computed over an image area by computing a magnitude from the responses of Roberts masks or from the response of the Laplacian mask.

Malik and Perona [75] proposed spatial filtering to model the preattentive texture perception in human visual system. Their proposed model consists of three stages: (i) convolution of the image with a bank of even-symmetric filters followed by half-wave rectification, (ii) inhibition of spurious responses in a localized area, and (iii) detection of the boundaries between the different textures. The even-symmetric filters they used consist of differences of offset Gaussian (DOOG) functions. The half-wave rectification and inhibition (implemented as leaders-take-all strategy) are methods of introducing a nonlinearity into the computation of texture features. A nonlinearity is needed in order to discriminate texture pairs with identical mean brightness and identical second-order statistics. The texture boundary detection is done by a straightforward edge detection method applied to the feature images obtained from stage (ii). This method works on a variety of texture examples and is able to discriminate natural as well as synthetic textures with carefully controlled properties. Unser and Eden [76] have also looked at texture features that are obtained from spatial filters and a nonlinear operator. Reed and Wechsler [77] review a number of spatial/spatial frequency domain filter techniques for segmenting textured images.

Another set of spatial filters are based on spatial moments [47]. The $(p+q)^{th}$ moments over an image region *R* are given by the formula

$$m_{pq} = \sum_{(x,y) \in R} x^p y^q I(x,y) \tag{11.16}$$

If the region *R* is a local rectangular area and the moments are computed around each pixel in the image, then this is equivalent to filtering the image by a set of spatial masks. The resulting filtered images that correspond to the moments are then used as texture features. The masks are obtained by defining a window of size $W \times W$ and a local coordinate system centered within the window. Let (i, j) be the image coordinates at which the moments are computed. For pixel coordinates (m, n) which fall within the $W \times W$ window centered at (i, j), the normalized coordinates (x_m, y_n) are given by:

$$x_m = \frac{(m-i)}{(W/2)} \qquad y_n = \frac{(n-j)}{(W/2)} \tag{11.17}$$

Then the moments within a window centered at pixel (i, j) are computed by the sum in Equation (11.16) that uses the normalized coordinates.

$$m_{pq} = \sum_{n=-W/2}^{W/2} \sum_{m=-W/2}^{W/2} I(m,n) x_m^p y_n^q \tag{11.18}$$

The coefficients for each pixel within the window to evaluate the sum is what defines the mask coefficients. If *R* is a 3x3 region, then the resulting masks are given below:

$$M_{00} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad M_{10} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad M_{01} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$M_{20} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad M_{11} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \qquad M_{02} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

The moment-based features have been used successfully in texture segmentation [78]. An example texture pair and the segmentation are shown in Figure 16.

3.4.2. Fourier domain filtering

The frequency analysis of the textured image is best done in the Fourier domain. As the psychophysical results indicated, the human visual system analyzes the textured images by decomposing the image into its frequency and orientation components [19]. The multiple channels tuned to different frequencies are also referred as multi-resolution processing in the literature. The concept of multi-resolution processing is further refined and developed in the wavelet model described below. Along the lines of these psychophysical results, texture analysis systems have been developed that perform filtering in the Fourier domain to obtain feature images. The idea is similar to the features computed from the rings and wedges as described in Section 3.1.2, except that the phase information is kept. Coggins and Jain [79] used a set of frequency and orientation selective filters in multichannel filtering approach. Each filter is either frequency selective *or* orientation selective. There are four orientation filters centered at 0, 45, 90, and 135. The number of frequency selective filters depends on the image size. For an image of size 128×128 six filters with center frequencies at 1, 2, 4, 8, 16, 32, and 64 cycles/image were used. They were able to successfully segment and classify a variety of natural images as well as synthetic texture pairs described by Julesz with identical second-order statistics (see Figure 4).

3.4.3. Gabor and Wavelet models

The Fourier transform is an analysis of the global frequency content in the signal. Many applications require the analysis to be localized in the spatial domain. This is usually handled by introducing spatial dependency into the Fourier analysis. The classical way of doing this is through what is called the window Fourier transform. The window Fourier transform (or short-time Fourier Transform) of a one-dimensional signal $f(x)$ is defined as:

$$F_w(u, \xi) = \int_{-\infty}^{\infty} f(x) w(x-\xi) e^{-j2\pi ux} dx \qquad (11.19)$$

When the window function $w(x)$ is Gaussian, the transform becomes a Gabor transform. The limits on the resolution in the time and frequency domain of the window Fourier transform are determined by the *time-bandwidth product* or the *Heisenberg uncertainty inequality* given by:

$$\Delta t \Delta u \geq \frac{1}{4\pi} \qquad (11.20)$$

Once a window is chosen for the window Fourier transform, the time-frequency resolution is fixed over the entire time-frequency plane. To overcome the resolution limitation of the window Fourier transform, one lets the Δt and Δu vary in the time-frequency domain. Intuitively, the time resolution must increase as the central frequency of the analyzing filter is increased. That is, the relative bandwidth is kept constant in a logarithmic scale. This is accomplished by using a window whose width changes as the frequency changes.

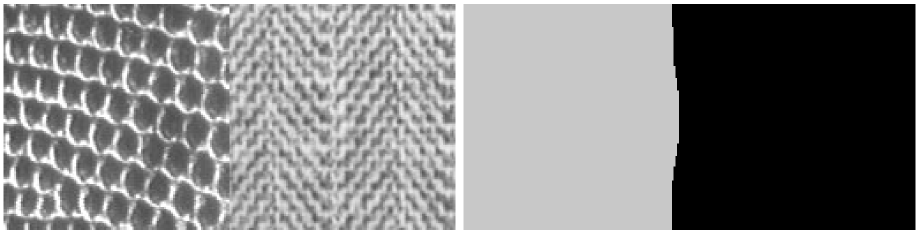


FIGURE 16. The segmentation results using moment based texture features. (a) A texture pair consisting of reptile skin and herringbone pattern from the Brodatz album [92]. (b) The resulting segmentation.

Recall that when a function $f(t)$ is scaled in time by a which is expressed as $f(at)$, the function is contracted if $a > 1$ and it is expanded when $a < 1$. Using this fact, the wavelet transform can be written as:

$$W_{f,a}(u, \xi) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) h^*\left(\frac{t - \xi}{a}\right) dt \tag{11.21}$$

Here, the impulse response of the filter bank is defined to be scaled versions of the same prototype function $h(t)$. Now, setting in Equation (11.21)

$$h(t) = w(t) e^{-j2\pi ut} \tag{11.22}$$

we obtain the wavelet model for texture analysis. Usually the scaling factor a will be based on the frequency of the filter.

Daugman [80] proposed the use of Gabor filters in the modeling of the receptive fields of simple cells in the visual cortex of some mammals. The proposal to use the Gabor filters in texture analysis was made by Turner [81] and Bovik *et al.* [82]. Later Farrokhnia and Jain used it successfully in segmentation and classification of textured images [40,83]. Gabor filters have some desirable optimality properties. Daugman [84] showed that for two-dimensional Gabor functions, the uncertainty relations $\Delta x \Delta u \geq \pi/4$ and $\Delta y \Delta v \geq \pi/4$ attain the minimum value. Here Δx and Δy are effective widths in the spatial domain and Δu and Δv are effective bandwidths in the frequency domain.

A two-dimensional Gabor function consists of a sinusoidal plane wave of a certain *frequency* and *orientation* modulated by a Gaussian envelope. It is given by:

$$f(x, y) = \exp(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]) \cos(2\pi u_0 x + \phi) \tag{11.23}$$

where u_0 and ϕ are the frequency and phase of the sinusoidal wave. The values σ_x and σ_y are the sizes of the Gaussian envelope in the x and y directions, respectively. The Gabor function at an arbitrary orientation θ_0 can be obtained from Equation (11.23) by a rigid rotation of the *x*-*y* plane by θ_0 .

The Gabor filter is a frequency and orientation selective filter. This can be seen from the Fourier domain analysis of the function. When the phase ϕ is 0, the Fourier transform of the resulting even-symmetric Gabor function $f(x, y)$ is given by

$$F(u, v) = A\left(\exp(-\frac{1}{2}\left[\frac{(u - u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]) + \exp(-\frac{1}{2}\left[\frac{(u + u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right])\right) \tag{11.24}$$

where $\sigma_u = 1/(2\pi\sigma_x)$, $\sigma_v = 1/(2\pi\sigma_y)$, and $A = 2\pi\sigma_x\sigma_y$. This function is real-valued and has two lobes in the spatial frequency domain, one centered around u_0 and another centered around $-u_0$. For a Gabor filter of a particular orientation, the lobes in the frequency domain are also appropriately rotated.

Jain and Farrokhnia [40] used a version of the Gabor transform in which window sizes for computing the Gabor filters are selected according to the central frequencies of the filters. The texture features were obtained as follows:

(a) Use a bank of Gabor filters at multiple scales and orientations to obtain filtered images. Let the filtered image for the i^{th} filter be $r_i(x, y)$.

(b) Pass each filtered image through a sigmoidal nonlinearity. This nonlinearity $\psi(t)$ has the form of $\tanh(\alpha t)$. The choice of the value of α is determined empirically.

(c) The texture feature for each pixel is computed as the absolute average deviation of the transformed values of the filtered images from the mean within a window W of size $M \times M$. The filtered images have zero mean, therefore, the i^{th} texture feature image $e_i(x, y)$ is given by the equation:

$$e_i(x, y) = \frac{1}{M^2} \sum_{(a, b) \in W} \left|\psi(r_i(a, b))\right| \tag{11.25}$$

The window size M is also determined automatically based on the central frequency of the filter. An example texture image and some intermediate results are shown in Figure 17. Texture features using Gabor filters were used in texture segmentation and texture classification tasks successfully. An example of the resulting segmentation is shown in Figure 18. Further details of the segmentation algorithm are explained in Section 4.1.

4. Texture Analysis Problems

The various methods for modeling textures and extracting texture features can be applied in four broad categories of problems: texture segmentation, texture classification, texture synthesis, and shape from texture. We now review these four areas.

4.1. Texture Segmentation

Texture segmentation is a difficult problem because one usually does not know *a priori* what types of textures exist in an image, how many different textures there are, and what regions in the image have which textures. In fact, one does not need to know which specific textures exist in the image in order to do texture segmentation. All that is needed is a way to tell that two textures (usually in adjacent regions of the images) are different.

The two general approaches to performing texture segmentation are analogous to methods for image segmentation: region-based approaches or boundary-based approaches. In a

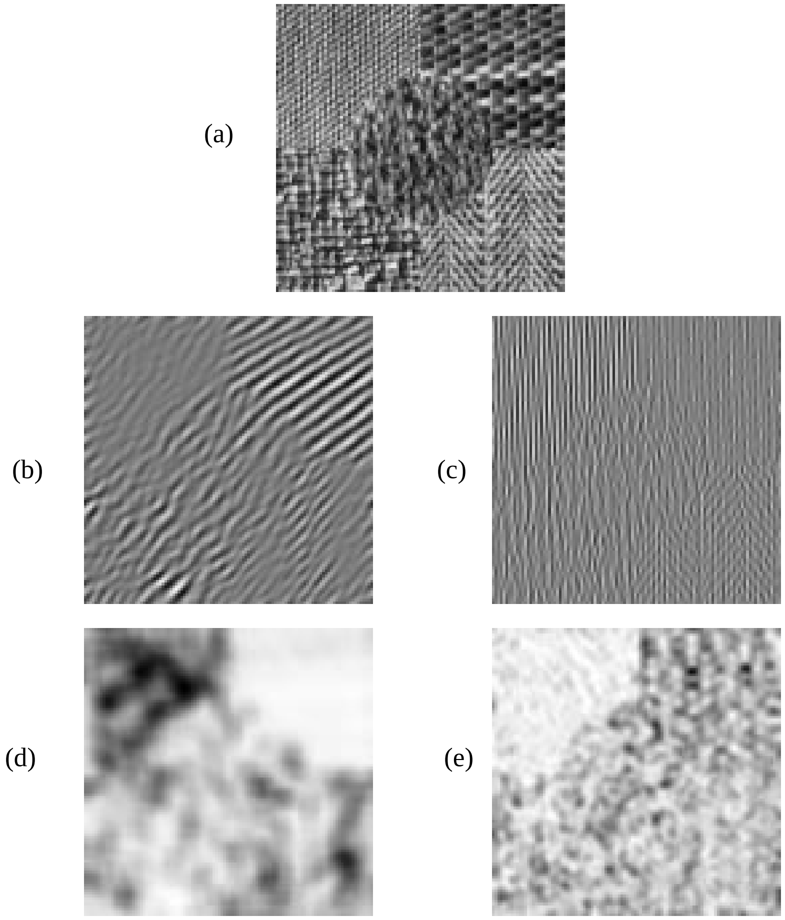


FIGURE 17. Filtering results on an example texture image. (a) Input image of five textures. (b), (c) A subset of the filtered images each filtered with a Gabor filter with the following parameters. The filter in (b) has a central frequency at 16 cycles/image-width and 135˚ orientation. The filter in (c) has a central frequency of 32 cycles/image-width and 0˚ orientation. (d), (e) The feature images obtained corresponding to the filtered images in (b) and (c). The filtered image in (b) shows a lot of activity in the textured region of the top right quadrant and the image in (c) shows activity in the textured region of the top left quadrant. These are reflected in the feature images in (d) and (e).

region-based approach, one tries to identify regions of the image which have a uniform texture. Pixels or small local regions are merged based on the similarity of some texture property. The regions having different textures are then considered to be segmented regions. This method has the advantage that the boundaries of regions are always closed and therefore, the regions with different textures are always well separated. It has the disadvantage, however, that in many region-based segmentation methods, one has to specify the number of distinct textures present in the image in advance. In addition, thresholds on similarity values are needed.

The boundary-based approaches are based upon the detection of differences in texture in adjacent regions. Thus boundaries are detected where there are differences in texture. In this method, one does not need to know the number of textured regions in the image in advance. However, the boundaries may have gaps and two regions with different textures are not identified as separate closed regions. Strictly speaking, the boundary based methods result in segmentation only if all the boundaries detected form closed curves.

Boundary based segmentation of textured images have been used by Tuceryan and Jain [49], Voorhees and Poggio [56], and Eom and Kashyap [85]. In all cases, the edges (or texture boundaries) are detected by taking two adjacent windows and deciding whether the textures in the two windows belong to the same texture or to different textures. If it is decided that the two textures are different, the point is marked as a boundary pixel. Du Buf and Kardan [86] studied and compared the performance of various texture segmentation techniques and their ability to localize the boundaries.

Tuceryan and Jain [49] use the texture features computed from the Voronoi polygons in order to compare the textures in the two windows. The comparison is done using a Kolmogorov-Smirnoff test. A probabilistic relaxation labeling, which enforces border smoothness, is used to remove isolated edge pixels and fill boundary gaps. Voorhees and Poggio extract blobs and elongated structures from images (they suggest that these correspond to Julesz's textons). The texture properties are based on blob characteristics such as their sizes, orientations, etc. They then decide whether the two sides of a pixel have the same texture using a statistical test called maximum frequency difference (MFD). The pixels where this statistic is sufficiently large are considered to be boundaries between different textures.

Jain and Farrokhnia [40] give an example of integrating a region-based and a boundary-based method to obtain a cleaner and more robust texture segmentation method. They use the texture features computed from the bank of Gabor filters to perform a region-based segmentation. This is accomplished by the following steps:

(a) Gabor features are calculated from the input image, yielding several feature images.

(b) A cluster analysis is performed in the Gabor feature space on a subset of randomly selected pixels in the input image (this is done in order to increase computational efficiency. About 6% of the total number of pixels in the image are selected). The number *k* of clusters is specified for doing the cluster analysis. This is set to a value larger than the true number of clusters and thus the image is oversegmented.

(c) Step (b) assigns a cluster label to the pixels (pattern) involved in cluster analysis. These labelled patterns are used as the training set and all the pixels in the image are classified into one of the *k* clusters. A minimum distance classifier is used. This results in a complete segmentation of the image into uniform textured regions.

(d) A connected component analysis is performed to identify each segmented region.

(e) A boundary-based segmentation is performed by applying the Canny edge detector on each feature image. The magnitude of the Canny edge detector for each feature image is summed up for each pixel to obtain a total edge response. The edges are then detected based on this total magnitude.

(f) The edges so detected are then combined with the region-based segmentation results to obtain the final texture segmentation.

The integration of the boundary-based and region-based segmentation results improve the resulting segmentation in most cases. For an example of this improvement see Figure 18.

4.2. Texture Classification

Texture classification involves deciding what texture category an observed image belongs to. In order to accomplish this, one needs to have an *a priori* knowledge of the classes to be recognized. Once this knowledge is available and the texture features are extracted, one then uses classical pattern classification techniques in order to do the classification.

Examples where texture classification was applied as the appropriate texture processing method include the classification of regions in satellite images into categories of land use [41]. Texture classification was also used in automated paint inspection by Farrokhnia [83]. In the latter application, the categories were ratings of the quality of paints obtained from human experts. These quality rating categories were then used as the training samples for supervised classification of paint images using texture features obtained from multi-channel Gabor filters.

4.3. Texture Synthesis

Texture synthesis is a problem which is more popular in computer graphics. It is closely tied to some of the methods discussed above, so we give only a brief summary here. Many of the modeling methods are directly applicable to texture synthesis. Markov random field models discussed in Section 3.3.1 can be directly used to generate textures by specifying the parameter vector θ and sampling from the probability distribution function [62, 60]. The synthetic textures in Figure 2(b) are generated using a gaussian Markov random field model and the algorithm in [87].

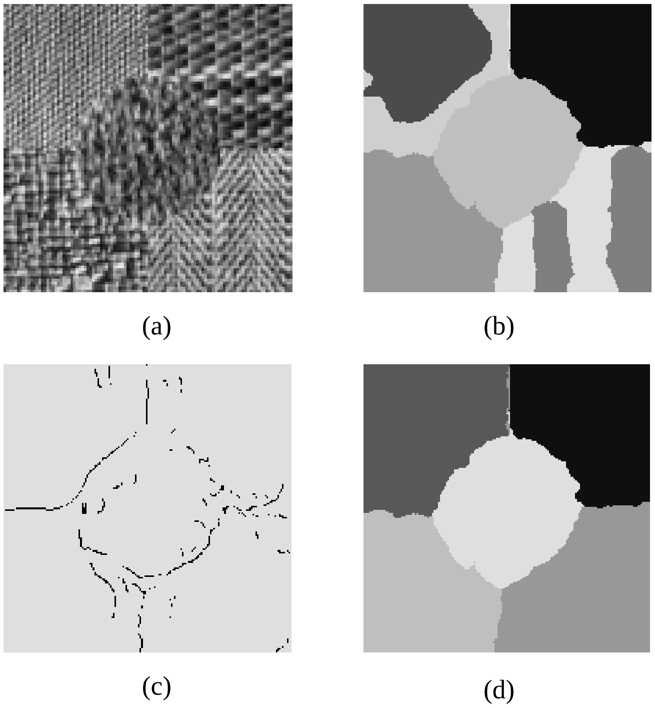


FIGURE 18. The results of integrating region-based and boundary-based processing using the multi-scale Gabor filtering method. (a) Original image consisting of five natural textures. (b) Seven category region-based segmentation results. (c) Edge-based processing and texture edges detected. (d) New segmentation after combining region-based and edge-based results.

Fractals have become popular recently in computer graphics for generating realistic looking textured images [88]. A number of different methods have been proposed for synthesizing textures using fractal models. These methods include midpoint displacement method and Fourier filtering method. The midpoint displacement method has become very popular because it is a simple and fast algorithm yet it can be used to generate very realistic looking textures. Here we only give the general outline of the algorithm. A much more detailed discussion of the algorithm can be found in [88]. The algorithm starts with a square grid representing the image with the four corners set to 0. It then displaces heights at the midpoints of the four sides and the center point of the square region by random amounts and repeats the process recursively. The iteration $n + 1$ uses the grid consisting of the midpoints of the squares in the grid for iteration n. The height at the midpoint is first interpolated between the endpoints and a random value is added to this value. The amount added is chosen from a normal distribution with zero-mean and variance σ_n^2 at iteration n. In order to keep the self-similar nature of the surface, the variance is changed as a function of the iteration number. The variance at iteration n is given by

$$\sigma_n^2 = r^{2nH} \qquad \text{where } r = 1/\sqrt{2} \tag{11.26}$$

This results in a fractal surface with fractal dimension $(3 - H)$. The heights of the fractal surface can be mapped onto intensity values to generate the textured images. The example image in Figure 2(c) was generated using this method.

Other methods include mosaic models [89,90]. This class of models can in turn be divided into subclasses of cell structure models and coverage models. In cell structure models the textures are generated by tessellating the plane into cells (bounded polygons) and assigning each cell gray levels according to a set of probabilities. The type of tessellation determines what type of textures are generated. The possible tessellations include triangular pattern, checkerboard patterns, Poisson line model, Delaunay model, and occupancy model. In coverage models, the texture is obtained by a random arrangement of a set of geometric figures in the plane. The coverage models are also referred to as bombing models.

4.4. Shape from Texture

There are many cues in images that allow the viewer to make inferences about the three-dimensional shapes of objects and surfaces present in the image. Examples of such cues include the variations of shading on the object surfaces or the relative configurations of boundaries and the types of junctions that allow one to infer three-dimensional shape from the line drawings of objects. The relation between the variations in texture properties and surface shape was first pointed out by Gibson [41].

Stevens observed that certain properties of texture are perceptually significant in the extraction of surface geometry [91]. There are three effects that surface geometry has on the appearance of texture in images: foreshortening and scaling of texture elements, and a change in their density. The foreshortening effect is due to the orientation of the surface on which the texture element lies. The scaling and density changes are due to the distance of the texture elements from the viewer. Stevens argued that texture density is not a useful measure for computing distance or orientation information because the density varies both with scaling and foreshortening. He concluded that the more perceptually stable property that allows one to extract surface geometry information is the direction in the image which is not foreshortened, called the *characteristic dimension*. Stevens suggested that one can compute relative depth information using the reciprocal of the scaling in the characteristic dimension. Using the relative depths, surface orientation can be estimated.

Bajcsy and Lieberman [92] used the gradient in texture element sizes to derive surface shape. They assumed a uniform texture element size on the three-dimensional surface in the scene. The relative distances are computed based on a gradient function in the image which was estimated from the texture element sizes. The estimation of the relative depth was done without using knowledge about the camera parameters and the original texture element sizes.

Witkin [93] used the distribution of edge orientations in the image to estimate the surface orientation. The surface orientation is represented by the slant (σ) and tilt (τ) angles. Slant is the angle between a normal to the surface and a normal to the image plane. Tilt is the angle between the surface normal's projection onto the image plane and a fixed coordinate axis in the image plane. He assumed an isotropic texture (uniform distribution of edge orientations) on the original surface. As a result of the projection process, the textures are foreshortened in the direction of steepest inclination (slant angle). Note that this idea is related to Stevens' argument because the direction of steepest inclination is perpendicular to the characteristic dimension. Witkin formulated the surface shape recovery by relating the slant and tilt angles to the distribution of observed edge directions in the image. Let β be the original edge orientation (the angle between the tangent and a fixed coordinate axis on the plane S containing the tangent). Let α^* be the angle between the x-axis in the image plane and the projected tangent. The α^* is related to the slant and tilt angles by the following expression:

$$\alpha^* = \operatorname{atan}\left(\frac{\tan\beta}{\cos\sigma}\right) + \tau \qquad (11.27)$$

Here α^* is an observable quantity in the image and (σ, τ) are the quantities to be computed. Witkin derived the expression for the conditional probabilities for the slant and tilt angles given the measured edge directions in the image and then used a maximum likelihood estimation method to compute the (σ, τ). Let $A^* = \{\alpha_1^*, \ldots, \alpha_n^*\}$ be a set of observed edge directions in the image. Then the conditional probabilities are given as:

$$P(\sigma, \tau | A^*) = \frac{P(\sigma, \tau)P(A^* | \sigma, \tau)}{\int\int P(\sigma, \tau)P(A^* | \sigma, \tau)d\sigma d\tau} \qquad (11.28)$$

where $P(\sigma, \tau) = \frac{\sin\sigma}{\pi^2}$. The maximum likelihood estimate of $P(\sigma, \tau | A^*)$ gives the desired surface orientation.

Blostein and Ahuja [57] used the scaling effect to extract surface information. They integrated the process of texture element extraction with the surface geometry computation. Texture element extraction is performed at multiple scales and the subset that yields a good surface fit is selected. The surfaces are assumed planar for simplicity. Texture elements are defined to be circular regions of uniform intensity which are extracted by filter-

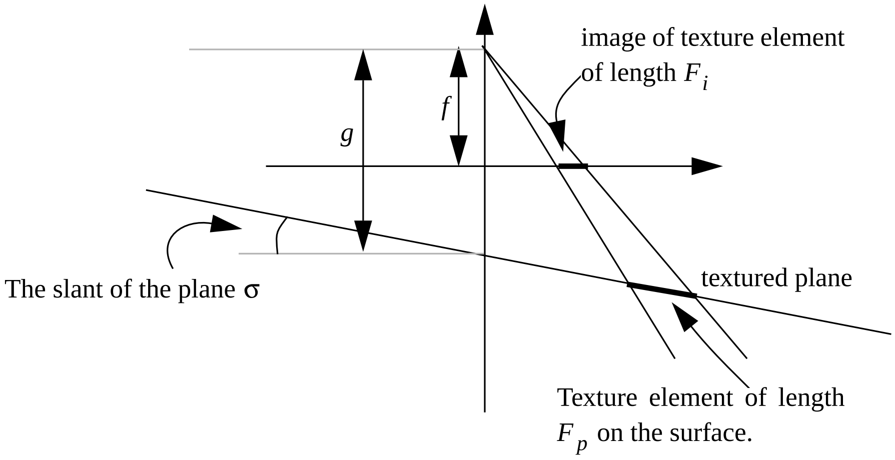
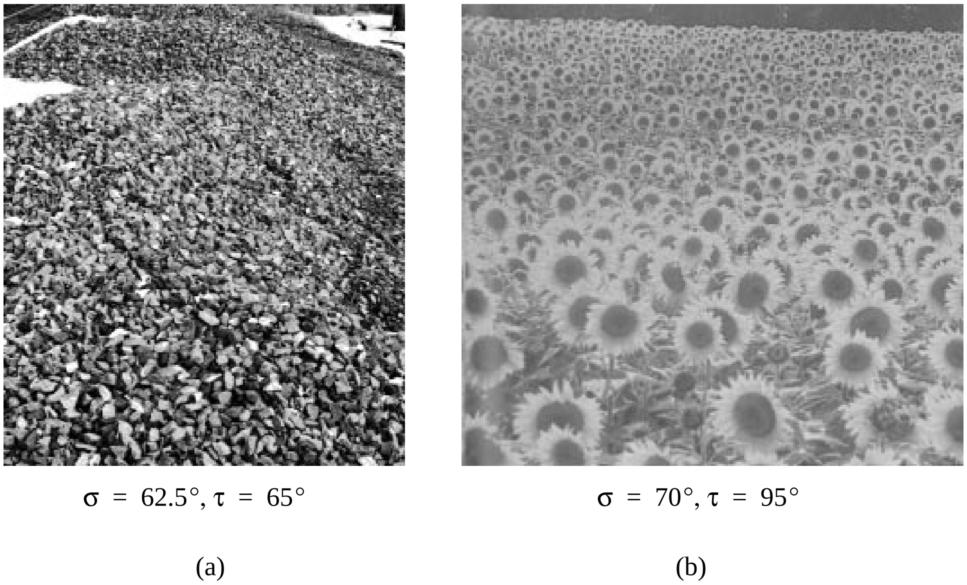


image of texture element of length F_i
f
g
textured plane
The slant of the plane σ
Texture element of length F_p on the surface.

FIGURE 19. The projective distortion of a texture element in the image.



$\sigma = 62.5°, \tau = 65°$

(a)

$\sigma = 70°, \tau = 95°$

(b)

FIGURE 20. Examples of shape from texture computation using Blostein and Ahuja’s algorithm [57]. (a) An image of a field of rocks and the computed slant and tilt of the plane. (b) An image of a sunflower field and the extracted slant and tilt values.

ing the image with $\nabla^2 G$ and $\frac{\partial}{\partial\sigma}(\nabla^2 G)$ operators and comparing the filter responses to those of an ideal disk (here σ is the size of the Gaussian, G). At the extremum points of the image filtered by $\nabla^2 G$, the diameter (D) and contrast (C) of the best fitting disks are computed. The convolution is done at multiple scales. Only those disks whose computed diameters are close to the size of the Gaussian are retained. As a result, blob-like texture elements of different sizes are detected.

The geometry of the projection is shown in Figure 19. Let σ and τ be the slant and tilt of the surface. The image of a texture element has the foreshortened dimension F_i and the characteristic dimension U_i. The area A_i of the image texel is proportional to the product $F_i U_i$ for compact shapes. The expression for the area, A_i, of the image of a texture element is given by:

$$A_i = A_C(1 - \tan\theta\tan\sigma)^3 \tag{11.29}$$

where A_C is the area that would be measured for the texel at the center of the image. The angle θ is given by the expression

$$\theta = \operatorname{atan}((x\cos\tau + y\sin\tau)(r/f)) \tag{11.30}$$

Here, r is the physical width of the image, r/f is a measure of field of view of the camera, and (x, y) denotes pixel coordinates in the image. A_i can be measured in the image. To find the surface orientation, an accumulator array consisting of the parameters (A_C, σ, τ) is constructed. For each combination of parameter values, a possible planar fit is computed. The plane with the highest fit rating is selected as the surface orientation and texture elements that support this fit are selected as the true texture elements. Some example images and the computed slant and tilt values are shown in Figure 20.

5. Summary

This chapter has reviewed the basic concepts and various methods and techniques for processing textured images. Texture is a prevalent property of most physical surfaces in the natural world. It also arises in many applications such as satellite imagery and printed documents. Many common low level vision algorithms such as edge detection break down when applied to images that contain textured surfaces. It is therefore crucial that we have robust and efficient methods for processing textured images. Texture processing has been successfully applied to practical application domains such as automated inspection and satellite imagery. It is also going to play an important role in the future as we can see from the promising application of texture to a variety of different application domains.

Acknowledgment

The support of the National Science Foundation through grants IRI-8705256 and CDA-8806599 is gratefully acknowledged. We thank the Norwegian Computing center for providing the SAR images shown in Figure 5. We also thank our colleagues Dr. Richard C. Dubes and Dr. Patrick J. Flynn for the invaluable comments and feedback they provided during the preparation of this document.

References

1. Coggins, J. M., “A Framework for Texture Analysis Based on Spatial Filtering,” Ph.D. Thesis, Computer Science Department, Michigan State University, East Lansing, Michigan, 1982.

2. Tamura, H., S. Mori, and Y. Yamawaki, “Textural Features Corresponding to Visual Perception,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-8**, pp. 460-473, 1978.

3. Sklansky, J., “Image Segmentation and Feature Extraction,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-8**, pp. 237-247, 1978.

4. Haralick, R.M., “Statistical and Structural Approaches to Texture,” *Proceedings of the IEEE*, **67**, pp. 786-804, 1979.

5. Richards, W. and A. Polit, “Texture matching,” *Kybernetic*, **16**, pp. 155-162, 1974.

6. Zucker, S. W. and K. Kant, “Multiple-level Representations for Texture Discrimination,” In *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, pp. 609-614, Dallas, TX, 1981.

7. Hawkins, J. K., “Textural Properties for Pattern Recognition,” In *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld (editors), Academic Press, New York, 1969.

8. Brodatz, P., *Textures: A Photographic Album for Artists and Designers*. New York, Dover Publications, 1966.

9. Chen, C. C., *Markov Random Fields in Image Analysis*, Ph.D. Thesis, Computer Science Department, Michigan State University, East Lansing, MI, 1988.

10. Gibson, J.J., *The perception of the visual world*, Boston, MA, Houghton Mifflin, 1950.

11. Julesz, B., E. N. Gilbert, L. A. Shepp, and H. L. Frisch, “Inability of humans to discriminate between visual textures that agree in second-order statistics - revisited,” *Perception*, **2**, pp. 391-405,1973.

12. Julesz, B., “Visual Pattern Discrimination,” *IRE Trans. on Information Theory*, **IT-8**, pp. 84-92, 1962.

13. Julesz, B., “Experiments in the visual perception of texture,” *Scientific American*, **232**, pp. 34-43, 1975.

14. Julesz, B., “Nonlinear and Cooperative Processes in Texture Perception,” In *Theoretical Approaches in Neurobiology,* T.P. Werner, E. Reichardt, (Editors). MIT Press, Cambridge, MA, pp. 93-108, 1981.

15. Julesz, B., “Textons, the Elements of Texture Perception, and Their Interactions,” *Nature*, **290**, pp. 91-97, 1981.

16. Julesz, B., “A Theory of Preattentive Texture Discrimination Based on First-Order Statistics of Textons,” *Biological Cybernetics*, **41**, pp. 131-138, 1981.

17. Caelli, T., *Visual Perception*, Pergamon Press, 1981.

18. Beck, J., A. Sutter, and R. Ivry, “Spatial Frequency Channels and Perceptual Grouping in Texture Segregation,” *Computer Vision, Graphics, and Image Processing*, **37**, pp. 299-325, 1987.

19. Campbell, F.W. and J.G. Robson, “Application of Fourier Analysis to the Visibility of Gratings,” *Journal of Physiology*, **197**, pp. 551-566, 1968.

20. Devalois, R.L., D.G. Albrecht, and L.G. Thorell, “Spatial -frequency selectivity of cells in macaque visual cortex,” *Vision Research*, **22**, pp. 545-559, 1982.

21. Dewaele, P., P. Van Gool, and A. Oosterlinck, “Texture Inspection with Self-Adaptive Convolution Filters,” In *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 56-60, Rome, Italy, Nov. 14-17, 1988.

22. Chetverikov, D., “Detecting Defects in Texture,” In *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 61-63, Rome, Italy, Nov. 14-17, 1988.

23. Chen, J. and A. K. Jain, “A Structural Approach to Identify Defects in Textured Images,” In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 29-32, Beijing, 1988.

24. Conners, R. W., C. W. McMillin, K. Lin, and R. E. Vasquez-Espinosa, “Identifying and Locating Surface Defects in Wood: Part of an Automated Lumber Processing System,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-5**, pp. 573-583, 1983.

25. Siew, L. H., R. M. Hodgson, and E. J. Wood, “Texture Measures for Carpet Wear Assessment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-10**, pp. 92-105, 1988.

26. Jain, A. K., F. Farrokhnia, and D. H. Alman, “Texture Analysis of Automotive Finishes,” In *Proceedings of SME Machine Vision Applications Conference*, pp. 1-16, Detroit, MI, Nov. 1990.

27. Sutton, R. and E. L. Hall, “Texture Measures for Automatic Classification of Pulmonary Disease,” *IEEE Transactions on Computers*, **C-21**, pp. 667-676, 1972.

28. Harms, H., U. Gunzer, and H. M. Aus, “Combined Local Color and Texture Analysis of Stained Cells,” *Computer Vision, Graphics, and Image Processing*, **33**, pp.364-376, 1986.

29. Landeweerd, G. H. and E. S. Gelsema, “The Use of Nuclear Texture Parameters in the Automatic Analysis of Leukocytes,” *Pattern Recognition*, **10**, pp. 57-61, 1978.

30. Insana, M. F., R. F. Wagner, B. S. Garra, D. G. Brown, and T. H. Shawker, “Analysis of Ultrasound Image Texture via Generalized Rician Statistics,” *Optical Engineering*, **25**, pp. 743-748, 1986.

31. Chen, C. C., J. S. Daponte, and M. D. Fox, “Fractal Feature Analysis and Classification in Medical Imaging,” *IEEE Transactions on Medical Imaging*, **8**, pp. 133-142, 1989.

32. Lundervold, A., *Ultrasonic Tissue Characterization - A Pattern Recognition Approach*, Technical Report, Norwegian Computing Center, Oslo, Norway, 1992.

33. Wang, D. and S. N. Srihari, “Classification of Newspaper Image Blocks Using Texture Analysis,” *Computer Vision, Graphics, and Image Processing*, **47**, pp. 327-352, 1989.

34. Wahl, F. M., K. Y. Wong, and R. G. Casey, “Block Segmentation and Text Extraction in Mixed Text/Image Documents,” *Computer Graphics and Image Processing*, **20**, pp. 375-390, 1982.

35. Fletcher, J. A. and R. Kasturi, “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images,” *IEEE Transactions on Image Analysis and Machine Intelligence*, **PAMI-10**, pp. 910-918, 1988.

36. Taxt, T., P. J. Flynn, and A. K. Jain, “Segmentation of Document Images,” *Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-11**, pp. 1322-1329, 1989.

37. Jain, A. K. and S. K. Bhattacharjee, “Text Segmentation Using Gabor Filters for Automatic Document Processing,” to appear in *Machine Vision and Applications*.

38. Jain, A. K. and S. K. Bhattacharjee, “Address Block Location on Envelopes Using Gabor Filters,” to appear in *Proceedings of 11th International Conference on Pattern Recognition*, The Hague, Netherlands, August 1992.

39. Jain, A. K., S. K. Bhattacharjee, and Y. Chen, “On Texture in Document Images,” to appear in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, IL, June 1992.

40. Jain, A.K. and F. Farrokhnia. “Unsupervised Texture Segmentation Using Gabor Filters,” *Pattern Recognition*, **24**, pp. 1167-1186, 1991.

41. Haralick, R. M., K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-3**, pp. 610-621, 1973.

42. Rignot, E. and R. Kwok, “Extraction of Textural Features in SAR Images: Statistical Model and Sensitivity,” In *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 1979-1982, Washington, DC, 1990.

43. Schistad, A. H. and A. K. Jain, “Texture Analysis in the Presence of Speckle Noise,” In *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, Houston, TX, May 1992.

44. Du, L. J., “Texture Segmentation of SAR Images Using Localized Spatial Filtering,” In *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 1983-1986, Washington, DC, 1990.

45. Lee, J. H. and W. D. Philpot, “A Spectral-Textural Classifier for Digital Imagery,” In *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 2005-2008, Washington, DC, 1990.

46. Tomita, Fumiaki and S. Tsuji, *Computer Analysis of Visual Textures*, Kluwer Academic Publishers, Boston, 1990.

47. Laws, K. I., *Textured Image Segmentation*. Ph.D. thesis, University of Southern California, 1980.

48. Picard, R., I. M. Elfadel, and A. P. Pentland, “Markov/Gibbs Texture Modeling: Aura Matrices and Temperature Effects,” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 371-377, Maui, Hawaii, 1991.

49. Tuceryan, M. and A. K. Jain, “Texture Segmentation Using Voronoi Polygons,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-12**, pp. 211-216, 1990.

50. Ahuja, N., “Dot Pattern Processing Using Voronoi Neighborhoods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-4**, pp. 336-343, 1982.

51. Voronoi, G., “Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire: Recherches sur les parallélloèdres primitifs,” *J. Reine Angew. Math.*, **134**, pp. 198-287,1908.

52. Shamos, M.I. and D. Hoey. “Closest-point Problems,” In *16th Annual Symposium on Foundations of Computer Science*, pp. 131-162, 1975.

53. Preparata, F. P. and M. I. Shamos, *Computational Geometry*, Springer-Verlag, New York, 1985.

54. Marr, D., *Vision*, Freeman, San Francisco, 1982.

55. Hu, M. K., “Visual Pattern Recognition by Moment Invariants,” *IRE Transactions on Information Theory*, **IT-8**, pp. 179-187, 1962.

56. Voorhees, H. and T. Poggio, “Detecting textons and texture boundaries in natural images,” In *Proceedings of the First International Conference on Computer Vision*, pp. 250-258, London, 1987.

57. Blostein, D. and N. Ahuja, “Shape from Texture: Integrating Texture-Element Extraction and Surface Estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-11**, pp. 1233-1251, 1989.

58. Zucker, S. W., “Toward a model of Texture,” *Computer Graphics and Image Processing*, **5**, pp. 190-202, 1976.

59. Fu, K.S., *Syntactic Pattern Recognition and Applications*, Prentice-Hall, New Jersey, 1982.

60. Cross, G.C. and A.K. Jain, “Markov Random Field Texture Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-5**, pp. 25-39, 1983.

61. Chellappa, R. and S. Chatterjee, “Classification of Textures Using Gaussian Markov Random Fields,” *IEEE Transactions on Acoustic, Speech, and Signal Processing*, **ASSP-33**, pp. 959-963, 1985.

62. Khotanzad, A. and R. Kashyap, “Feature Selection for Texture Recognition Based on Image Synthesis,” *IEEE Transactions on Systems, Man, and Cybernetics*, **17**, pp. 1087-1095, 1987.

63. Cohen, F.S. and D.B. Cooper, “Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-9**, pp. 195-219, 1987.

64. Therrien, C.W., “An Estimation-Theoretic Approach to Terrain Image Segmentation,” *Computer Vision, Graphics, and Image Processing*, **22**, pp. 313-326, 1983.

65. Geman, S. and D. Geman, “Stochastic Relaxation, Gibbs distributions, and the Bayesian Restoration of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**, pp. 721-741, 1984.

66. Besag, J., “Spatial Interaction and the Statistical Analysis of Lattice Systems,” *Journal of Royal Statistical Society*, **B-36**, pp. 344-348, 1974.

67. Derin, H. and H. Elliott, “Modeling and segmentation of noisy and textured images using Gibbs random fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-9**, pp. 39-55, 1987.

68. Mandelbrot, B. B., *The Fractal Geometry of Nature*, Freeman, San Francisco, 1983.

69. Pentland, A., “Fractal-based description of natural scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-9**, pp. 661-674, 1984.

70. Keller, J. M., S. Chen, and R. M. Crownover, “Texture Description and Segmentation through Fractal Geometry,” *Computer Vision, Graphics, and Image Processing*, **45**, pp. 150-166, 1989.

71. Voss, R., “Random fractals: Characterization and measurement,” In *Scaling Phenomena in Disordered Systems*, R. Pynn and A. Skjeltorp, (Editors), Plenum, New York, 1986.

72. Super, B. J. and A. C. Bovik, “Localized Measurement of Image Fractal Dimension Using Gabor Filters,” *Journal of Visual Communication and Image Representation*, **2**, pp. 114-128, 1991.

73. Ohanian, P. P. and R. C. Dubes, “Performance Evaluation for Four Classes of Textural Features,” submitted to *Pattern Recognition.*

74. Georgeson, M. A., “Spatial Fourier Analysis and Human Vision,” Chapter 2, in *Tutorial Essays in Psychology, A Guide to Recent Advances*, N. S. Sutherland (ed.), vol 2, Lawrence Earlbaum Associates, Hillsdale, N.J., 1979.

75. Malik, J. and P. Perona, “Preattentive Texture Discrimination with Early Vision Mechanisms,” *Journal of the Optical Society of America*, Series A, **7**, pp. 923-932, 1990.

76. Unser, M. and M. Eden, “Nonlinear Operators for Improving Texture Segmentation Based on Features Extracted by Spatial Filtering,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-20**, pp. 804-815, 1990.

77. Reed, T. R. and H. Wechsler, “Segmentation of Textured Images and Gestalt Organization Using Spatial/Spatial-Frequency Representations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-12**, pp. 1-12, 1990.

78. Tuceryan, M. “Moment Based Texture Segmentation,” In *Proceedings of 11th International Conference on Pattern Recognition*, The Hague, Netherlands, August 1992.

79. Coggins, J.M. and A.K. Jain, “A Spatial Filtering Approach to Texture Analysis,” *Pattern Recognition Letters*, **3**, pp. 195-203, 1985.

80. Daugman, J.G., “Two-dimensional spectral analysis of cortical receptive field profiles,” *Vision Research*, **20**, pp. 847-856, 1980.

81. Turner, M.R., “Texture Discrimination by Gabor Functions,” *Biological Cybernetics*, **55**, pp. 71-82, 1986.

82. Clark, M. and A.C. Bovik, “Texture segmentation using Gabor modulation/demodulation,” *Pattern Recognition Letters*, **6**, pp. 261-267, 1987.

83. Farrokhnia, F., *Multi-channel filtering techniques for texture segmentation and surface quality inspection*, Ph.D. thesis, Computer Science Department, Michigan State University, 1990.

84. Daugman, J.G., “Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters,” *Journal of Optical Society of America*, **2**, pp. 1160-1169, 1985.

85. Eom, Kie-Bum and R. L. Kashyap, “Texture and Intensity Edge Detection with Random Field Models,” In *Proceedings of the Workshop on Computer Vision*, pp. 29-34, Miami Beach, FL, 1987.

86. Du Buf, J. M. H. M. Kardan, and M. Spann, “Texture Feature Performance for Image Segmentation,” *Pattern Recognition*, **23**, pp. 291-309, 1990.

87. Chellappa, R., S. Chatterjee, and R. Bagdazian, “Texture Synthesis and Compression Using Gaussian-Markov Random Field Models,” *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-15**, pp. 298-303, 1985.

88. Peitgen, H. O. and D. Saupe, *The Science of Fractal Images*, Springer-Verlag, New York, 1988.

89. Ahuja, N. and A. Rosenfeld, “Mosaic Models for Textures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-3**, pp. 1-11, 1981.

90. Ahuja, N. “Texture,” In *Encyclopedia of Artificial Intelligence*, pp. 1101-1115, 1987.

91. Stevens, K. A., *Surface Perception from Local Analysis of Texture and Contour*, MIT Technical Report, Artificial Intelligence Laboratory, no. AI-TR 512, 1980.

92. Bajcsy, R. and L. Lieberman, “Texture Gradient as a Depth Cue,” *Computer Graphics and Image Processing*, **5**, pp. 52-67, 1976.

93. Witkin, A. P, “Recovering Surface Shape and Orientation from Texture,” *Artificial Intelligence*, **17**, pp. 17-45, 1981.