15 Texture

15.1 Introduction

In Chapters 11 and 12 we studied smoothing and edge detection and in Chapter 13 simple neighborhoods. In this chapter, we will take these important building blocks and extend them to analyze complex patterns, known as *texture* in image processing. Actually, textures demonstrate the difference between an artificial world of objects whose surfaces are only characterized by their color and reflectivity properties to that of real-world imagery.

Our visual system is capable of recognizing and distinguishing texture with ease, as can be seen from Fig. 15.1. It appears to be a much more difficult task to characterize and distinguish the rather “diffuse” properties of the texture with some precisely defined parameters that allow a computer to perform this task.

In this chapter we systematically investigate operators to analyze and differentiate between textures. These operators are able to describe even complex patterns with just a few characteristic figures. We thereby reduce the texture recognition problem to the simple task of distinguishing gray values.

How can we define a texture? An arbitrary pattern that extends over a large area in an image is certainly not recognized as a texture. Thus the basic property of a texture is a small elementary pattern which is repeated periodically or quasi-periodically in space like a pattern on a wall paper. Thus, it is sufficient to describe the small elementary pattern and the repetition rules. The latter give the characteristic scale of the texture.

Texture analysis can be compared to the analysis of the structure of solids, a research area studied in solid state physics, chemistry, and mineralogy. A solid state physicist must find out the repetition pattern and the distribution of atoms in the elementary cell. Texture analysis is complicated by the fact that both the patterns and periodic repetition may show significant random fluctuation, as shown in Fig. 15.1.

Textures may be organized in a *hierarchical* manner, i. e., they may look quite different at different scales. A good example is the curtain shown in Fig. 15.1a. On the finest scale our attention is focused on the individual threads (Fig. 15.2a). Then the characteristic scale is the thickness of the threads. They also have a predominant local orientation. On

B. Jähne, Digital Image Processing
ISBN 3-540-24035-7

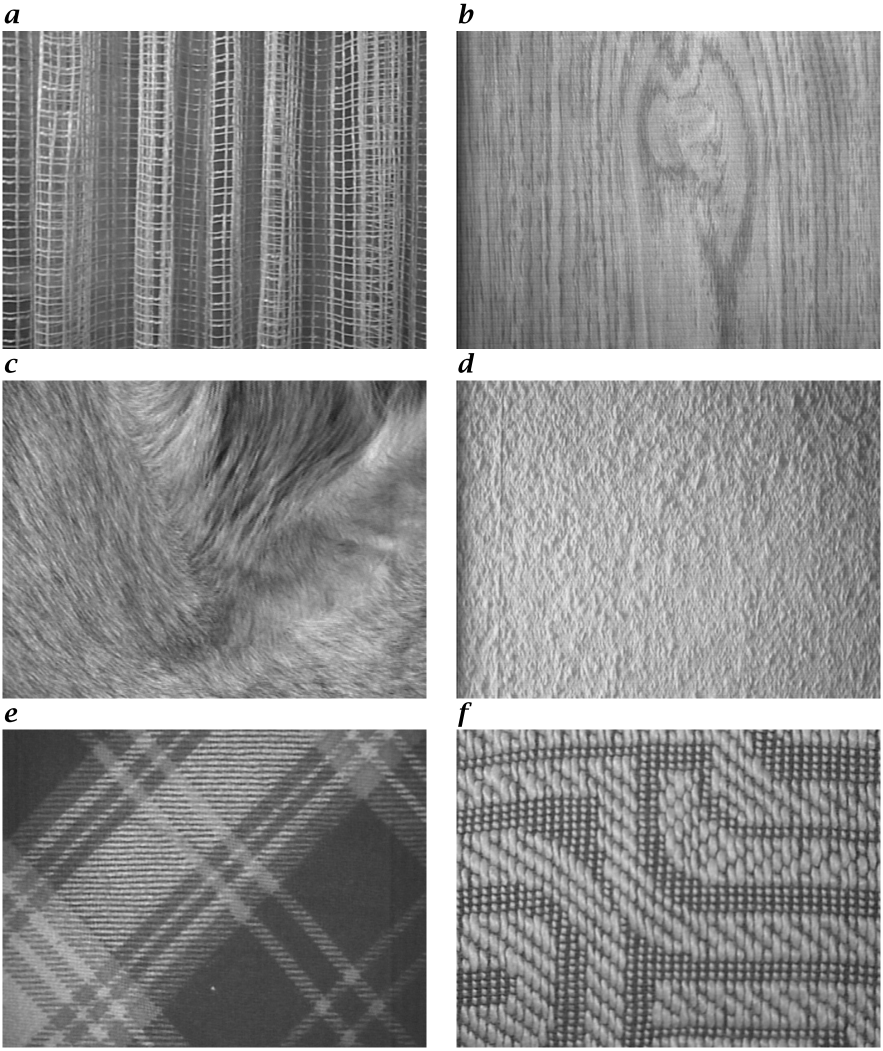

Figure 15.1: *Examples of textures:* ***a*** *curtain;* ***b*** *wood;* ***c*** *dog fur;* ***d*** *woodchip paper;* ***e, f*** *clothes.*

the next coarser level, we will recognize the meshes of the net (Fig. 15.2b). The characteristic scale here shows the size of the meshes. At this level, the local orientation is well distributed. Finally, at an even coarser level, we no longer recognize the individual meshes, but observe the folds of the curtain (Fig. 15.2c). They are characterized by yet another characteristic scale, showing the period of the folds and their orientation. These considerations emphasize the importance of *multiscale texture analysis*.

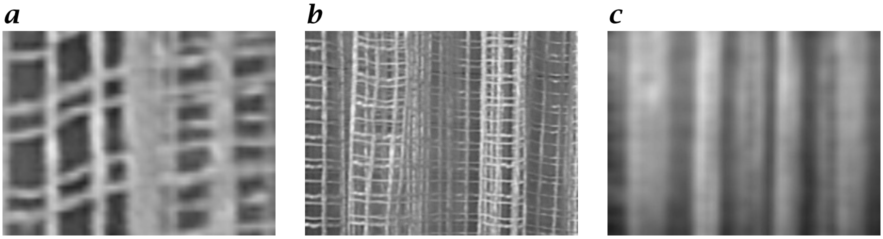

Figure 15.2: *Hierarchical organization of texture demonstrated by showing the image of the curtain in Fig. 15.1a at different resolutions.*

Thus multiscale data structures as discussed in the first part of this book (Chapter 5) are essential for texture analysis.

Generally, two classes of texture parameters are of importance. Texture parameters may or may not be rotation and scale invariant. This classification is motivated by the task we have to perform.

Imagine a typical industrial or scientific application in which we want to recognize objects that are randomly oriented in the image. We are not interested in the orientation of the objects but in their distinction from each other. Therefore, texture parameters that depend on orientation are of no interest. We might still use them but only if the objects have a characteristic shape which then allows us to determine their orientation. We can use similar arguments for scale-invariant features. If the objects of interest are located at different distances from the camera, the texture parameter used to recognize them should also be scale invariant. Otherwise the recognition of the object will depend on distance. However, if the texture changes its characteristics with the scale — as in the example of the curtain in Fig. 15.1a — the scale-invariant texture features may not exist at all. Then the use of textures to characterize objects at different distances becomes a difficult task.

In the examples above, we were interested in the objects themselves but not in their orientation in space. The orientation of surfaces is a key feature for another image processing task, the reconstruction of a three-dimensional scene from a two-dimensional image. If we know the surface of an object shows a uniform texture, we can analyze the orientation and scales of the texture to find the orientation of the surface in space. For this, the characteristic scales and orientations of the texture are needed.

Texture analysis is one of those areas in image processing which still lacks fundamental knowledge. Consequently, the literature contains many different empirical and semiempirical approaches to texture analysis. Here these approaches are not reiterated. In contrast, a rather simple approach to texture analysis is presented which builds complex texture operators from elementary operators.

For texture analysis only four fundamental texture operators are used:

- *mean*,
- *variance*,
- *orientation*,
- *scale*,

which are applied at different levels of the hierarchy of the image processing chain. Once we have, say, computed the local orientation and the local scale, the mean and variance operators can be applied again, now not to the mean and variance of the gray values but to the local orientation and local scale.

These four basic texture operators can be grouped in two classes. The mean and variance are rotation and scale independent, while the orientation and scale operators just determine the orientation and scale, respectively. This important separation between parameters invariant and variant to scale and rotation significantly simplifies texture analysis. The power of this approach lies in the simplicity and orthogonality of the parameter set and the possibility of applying it hierarchically.

15.2 First-Order Statistics

15.2.1 Basics

All texture features based on first-order statistics of the gray value distributions are by definition invariant on any permutation of the pixels. Therefore they do not depend on the orientation of objects and — as long as fine-scale features do not disappear at coarse resolutions — on the scale of the object. Consequently, this class of texture parameter is rotation and scale invariant.

The invariance of first-order statistics to pixel permutations has, however, a significant drawback. Textures with different spatial arrangements but the same gray value distribution cannot be distinguished. Here is a simple example. A texture with equally wide black and white stripes and a texture with a black and white chess board have the same bimodal gray value distribution but a completely different spatial arrangement of the texture.

Thus many textures cannot be distinguished by parameters based on first-order statistics. Other classes of texture parameter are required in addition for a better distinction of different textures.

15.2.2 Local Variance

All parameters that are deviating from the statistics of the gray values of individual pixels are basically independent of the orientation of the objects. In Section 3.2.2 we learnt to characterize the gray value distribution by the mean, variance, and higher moments. To be suitable for texture analysis, the estimate of these parameters requires to be averaged over a local neighborhood. This leads to a new operator estimating the *local variance.*

In the simplest case, we can select a mask and compute the parameters only from the pixels contained in this window M. The *variance operator*, for example, is then given by

$$v_{mn} = \frac{1}{P-1} \sum_{m',n' \in M} \left(g_{m-m',n-n'} - \overline{g}_{mn}\right)^2. \tag{15.1}$$

The sum runs over the P image points of the window. The expression $\overline{g}_{mn}$ denotes the mean of the gray values at the point $[m, n]^T$, computed over the same window M:

$$\overline{g}_{mn} = \frac{1}{P} \sum_{m',n' \in M} g_{m-m',n-n'}. \tag{15.2}$$

It is important to note that the variance operator is nonlinear. However, it resembles the general form of a neighborhood operation — a convolution. Combining Eqs. (15.1) and (15.2), we can show the variance operator is a combination of linear convolution and nonlinear point operations

$$v_{mn} = \frac{1}{P-1} \left[\sum_{m',n' \in M} g^2_{m-m',n-n'} - \left(\frac{1}{P} \sum_{m',n' \in M} g_{m-m',n-n'} \right)^2 \right], \tag{15.3}$$

or, in operator notation,

$$\mathcal{V} = \mathcal{R}(\mathcal{I} \cdot \mathcal{I}) - (\mathcal{R} \cdot \mathcal{R}). \tag{15.4}$$

The operator $\mathcal{R}$ denotes a smoothing over all the image points with a box filter of the size of the window M. The operator $\mathcal{I}$ is the identity operator. Therefore the operator $\mathcal{I} \cdot \mathcal{I}$ performs a nonlinear point operation, namely the squaring of the gray values at each pixel. Finally, the variance operator subtracts the square of a smoothed gray value from the smoothed squared gray values. From discussions on smoothing in Section 11.3 we know that a box filter is not an appropriate smoothing filter. Thus we obtain a better variance operator if we replace the box filter $\mathcal{R}$ with a binomial filter $\mathcal{B}$

$$\mathcal{V} = \mathcal{B}(\mathcal{I} \cdot \mathcal{I}) - (\mathcal{B} \cdot \mathcal{B}). \tag{15.5}$$

We know the variance operator to be isotropic. It is also scale independent if the window is larger than the largest scales in the textures a b

c d

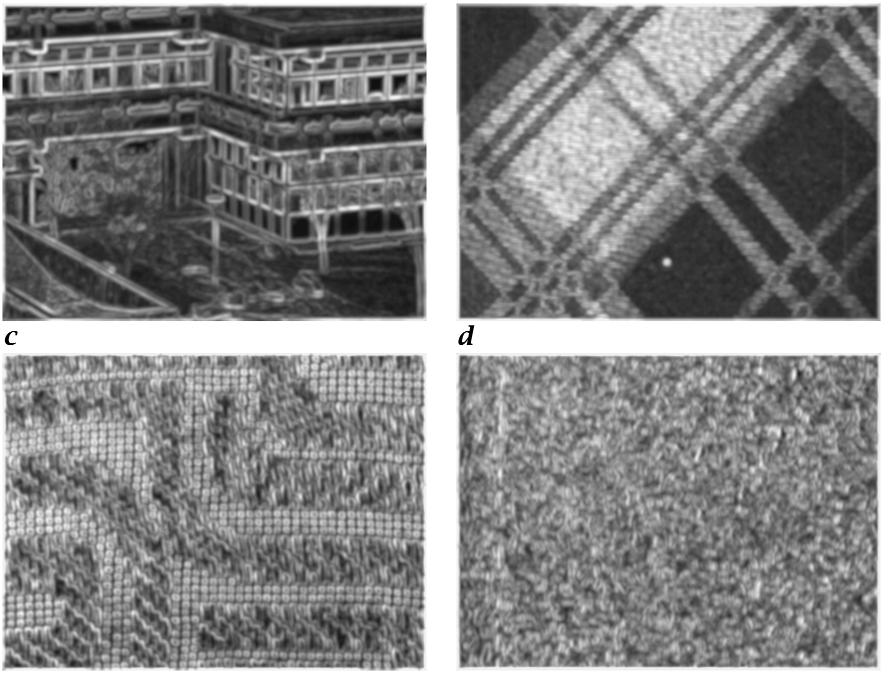

Figure 15.3: *Variance operator applied to different images:* ***a*** *Fig. 11.6a;* ***b*** *Fig. 15.1e;* ***c*** *Fig. 15.1f;* ***d*** *Fig. 15.1d.*

and if no fine scales of the texture disappear because the objects are located further away from the camera. This suggests that a scale-invariant texture operator only exists if the texture itself is scale invariant.

The application of the variance operator Eq. (15.5) with $\mathcal{B}^{16}$ to several images is shown in Fig. 15.3. In Fig. 15.3a, the variance operator turns out to be an isotropic edge detector, because the original image contains areas with more or less uniform gray values.

The other three examples in Fig. 15.3 show variance images from textured surfaces. The variance operator can distinguish the areas with the fine horizontal stripes in Fig. 15.1e from the more uniform surfaces. They appear as uniform bright areas in the variance image (Fig. 15.3b). The variance operator cannot distinguish between the two textures in Fig. 15.3c. As the resolution is still finer than the characteristic repetition scale of the texture, the variance operator does not give a uniform estimate of the variance in the texture. The chipwood paper (Fig. 15.3d) also gives a non-uniform response to the variance operator because the pattern shows significant random fluctuations.

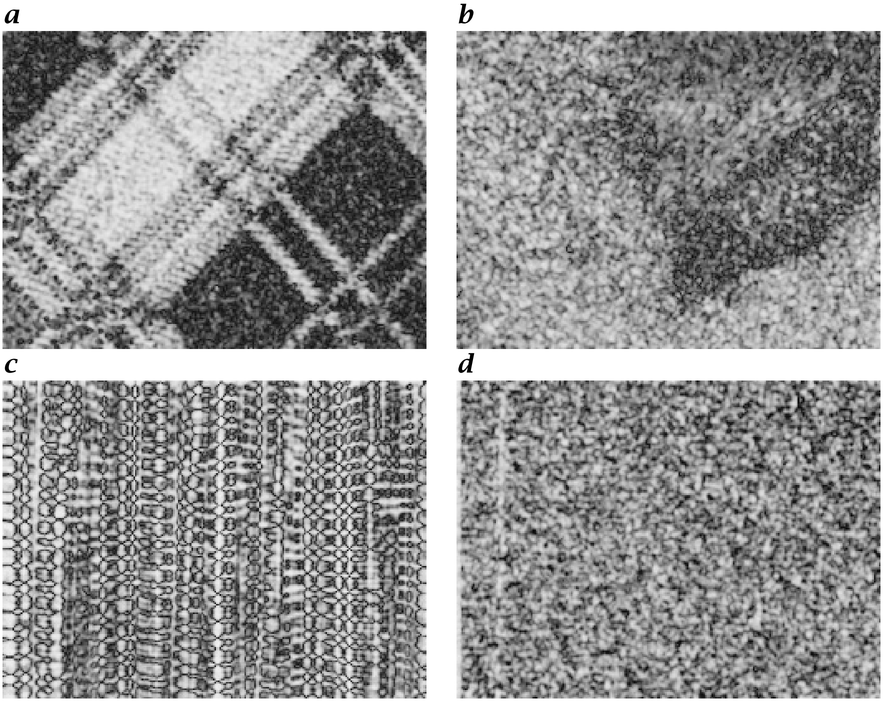

Figure 15.4: *Coherence of local orientation of* ***a*** *piece of cloth with regions of horizontal stripes (Fig. 15.1e),* ***b*** *dog fur (Fig. 15.1c),* ***c*** *curtain (Fig. 15.1a), and* ***d*** *woodchip wall paper.*

15.2.3 Higher Moments

Besides the variance, we could also use the higher moments of the gray value distribution as defined in Section 3.2.2 for a more detailed description. The significance of this approach may be illustrated with examples of two quite different gray value distributions, a normal and a bimodal distribution:

$$p(g) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{g-\overline{g}}{2\sigma^2}\right), \quad p'(g) = \frac{1}{2}\left(\delta(\overline{g}+\sigma) + \delta(\overline{g}-\sigma)\right).$$

Both distributions show the same mean and variance. Because both distributions are of even symmetry, all odd moments are zero. Thus the third moment (skewness) is also zero. However, the forth and all higher-order even moments of the two distributions are different.

15.3 Rotation and Scale Variant Texture Features

15.3.1 Local Orientation

As local orientation has already been discussed in detail in Chapter 13, we now only discuss some examples to illustrate the significance of local orientation for texture analysis. As this book contains only gray scale images, we only show coherence images of the local orientation.

Figure 15.4 shows the coherence measure for local orientation as defined in Section 13.3. This measure is one for an ideally oriented texture where the gray values change only in one direction, and zero for a distributed gray value structure. The coherency measure is close to one in the areas of the piece of shirt cloth with horizontal stripes (Fig. 15.4a) and in the dense parts of the dog fur (Fig. 15.4b).

The orientation analysis of the curtain (Fig. 15.1a) results in an interesting coherency pattern (Fig. 15.4c). The coherency is high along the individual threads, but not at the corners where two threads cross each other, or in most of the areas in between. The coherency of the local orientation of the woodchip paper image (Fig. 15.1d) does not result in a uniform coherence image as this texture shows no predominant local orientation.

15.3.2 Local Wave Number

In Section 13.4 we discussed in detail the computation of the *local wave number* from a *quadrature filter pair* by means of either a *Hilbert filter* (Section 13.4.2) or quadrature filters (Section 13.4.5). In this section we apply these techniques to compute the characteristic scale of a texture using a *directiopyramidal decomposition* as a directional bandpass filter followed by Hilbert filtering.

The piece of shirt cloth in Fig. 15.5a shows distinct horizontal stripes in certain parts. This image is first bandpass filtered using the levels one and two of the vertical component of a directiopyramidal decomposition of the image (Fig. 15.5b). Figure 15.5c shows the estimate of the local wave number (component in vertical direction).

All areas are masked out in which the amplitude of the corresponding structure (Fig. 15.5d) is not significantly higher than the noise level. In all areas with the horizontal stripes, a local wave number was computed. The histogram in Fig. 15.5e shows that the peak local wave number is about 0.133.

This structure is sampled about 7.5 times per wavelength. Note the long tail of the distribution towards short wave numbers. Thus a secondary larger-scale structure is contained in the texture. This is indeed given by the small diagonal stripes.

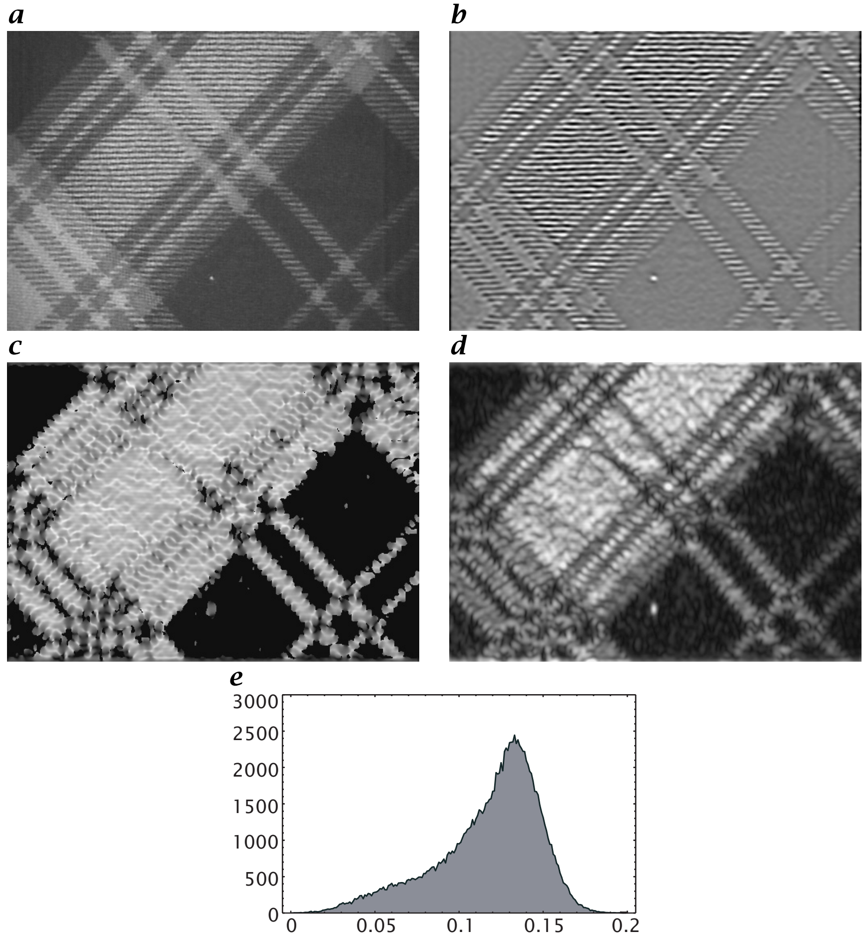

Figure 15.5: *Determination of the characteristic scale of a texture by computation of the local wave number:* ***a*** *original texture,* ***b*** *directional bandpass using the levels one and two of the vertical component of a directiopyramidal decomposition,* ***c*** *estimate of the local wave number (all structures below a certain threshold are masked to black),* ***d*** *amplitude of the local wave number, and* ***e*** *histogram of the local wave number distribution (units: number of periods per pixel).*

Figure 15.6 shows the same analysis for a textured wood surface. This time the texture is more random. Nevertheless, it is possible to determine the local wave number. It is important, though, to mask out the areas in which no significant amplitudes of the bandpass filtered image are present. If the masking is not performed, the estimate of the local wave number will be significantly distorted. With the masking a quite narrow distribution of the local wave number is found with a peak at a wave number of 0.085.

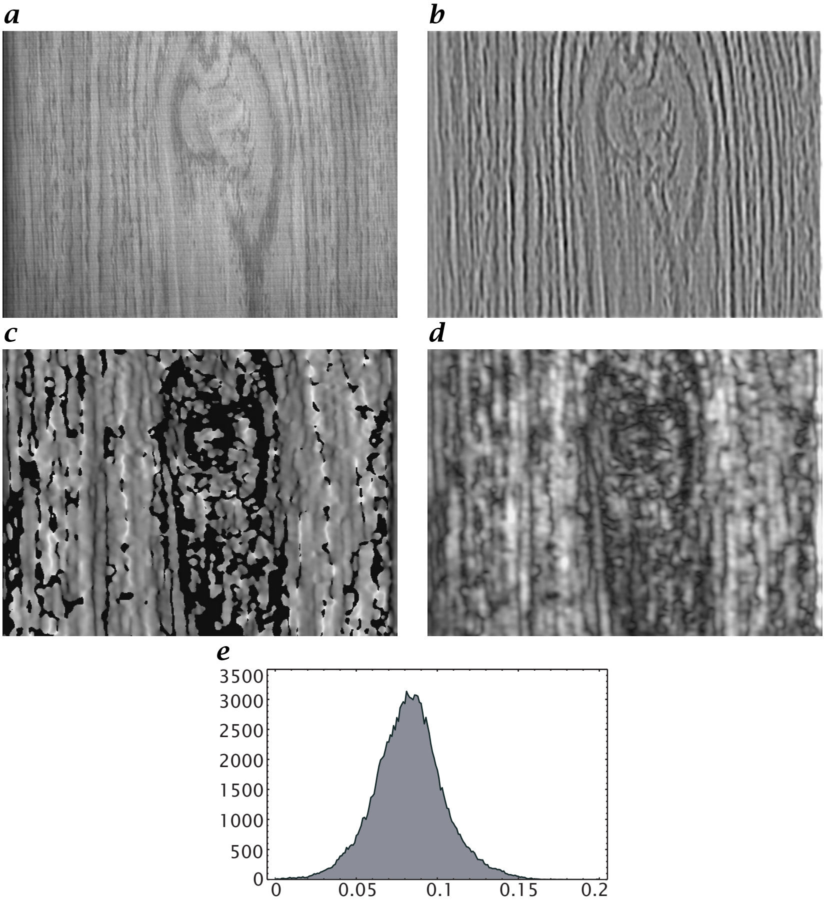

Figure 15.6: *Same as Fig. 15.5 applied to a textured wood surface.*

15.3.3 Pyramidal Texture Analysis

The Laplace pyramid is an alternative to the local wave number operator, because it results in a bandpass decomposition of the image. This decomposition does not compute a local wave number directly, but we can obtain a series of images which show the texture at different scales.

The variance operator takes a very simple form with a Laplace pyramid, as the mean gray value, except for the coarsest level, is zero:

$$\mathcal{V} = \mathcal{B}(\mathcal{L}^{(p)} \cdot \mathcal{L}^{(p)}). \tag{15.6}$$

Figure 15.7 demonstrates how the different textures from Fig. 15.1f appear at different levels of the Laplacian pyramid. In the two finest scales at the zero and first level of the pyramid (Fig. 15.7a, b), the variance is dominated by the texture itself. The most pronounced feature

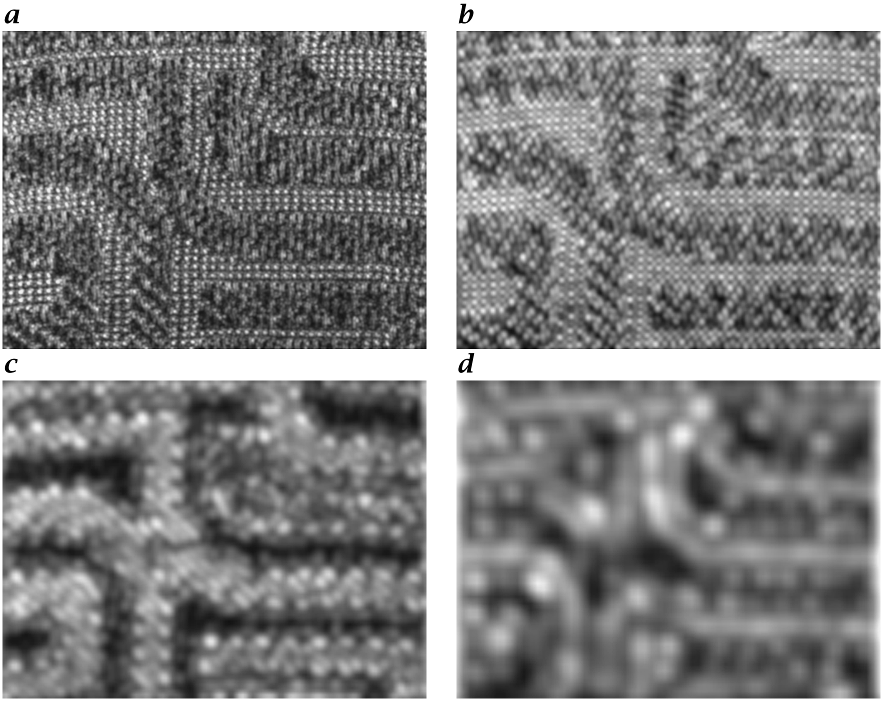

Figure 15.7: *Application of the variance operator to levels 0 to 3 of the Laplace pyramid of the image from Fig. 15.1f.*

is the variance around the dot-shaped stitches in one of the two textures. At the second level of the Laplacian pyramid (Fig. 15.7c), the dot-shaped stitches are smoothed away and the variance becomes small in this texture while the variance is still significant in the regions with the larger vertically and diagonally oriented stitches. Finally, the third level (Fig. 15.7d) is too coarse for both textures and thus dominated by the edges between the two texture regions because they have a different mean gray value.

The Laplace pyramid is a data structure well adapted for the analysis of hierarchically organized textures that may show different characteristics at different scales, as in the example of the curtain discussed in Section 15.1. In this way we can apply such operators as local variance and local orientation at each level of the pyramid. The simultaneous application of the variance and local orientation operators at multiple scales gives a rich set of features, which allows even complex hierarchically organized textures to be distinguished. It is important to note that application of these operations on all levels of the pyramid only increases the number of computations by a factor of 4/3 for 2-D images.

15.4 *Exercises*

15.1: Statistical parameters for texture analysis

Interactive demonstration of statistical parameters for texture analysis (dip6ex15.01)

15.2: Local orientation for texture analysis

Interactive demonstration of texture analysis using the structure tensor for orientation analysis (dip6ex15.02)

15.3: Texture analysis with pyramids

Interactive demonstration of texture analysis with a multiscale approach on pyramids (dip6ex15.03)

15.4: **Features for texture analysis**

Which features are suitable for texture analysis? Try to list the features in a systematic way starting from the simplest possible feature such a the mean gray value and continuing to more and more complex textures. Briefly explain your approach!

15.5: **Strukture tensor for texture analysis**

Which types of texture can be differentiated with the structure tensor and which types cannot?
(Hint: Use examples of patterns leading to the same features to explain which textures cannot be distinguished by the structure tensor.)

15.6: **Invariant texture features**

Show which of the listed texture features is invariant under a change of the scale, rotation, and a change of the brightness of the image:

1. Variance operator: $(\boldsymbol{G} - \boldsymbol{B}\boldsymbol{G})^2$
2. Local gray value histogram computed in a certain neighborhood
3. Local histogram of the first-order derivative in x direction
4. Magnitude of the gray value gradient
5. Angle of the orientation vector
6. Coherency of local orientation
7. Variance of the angle of the orientation vector

Is it possible to make the features, which depend on the brightness of the image, invariant against brightness changes? If yes, how?

15.5 *Further Readings*

The textbooks of Jain [97, Section 9.11] and Pratt [157, Chapter 17] deal also with texture analysis. Further references for texture analysis are the monography of Rao [161], the handbook by Jähne et al. [94, Vol. 2, Chapter 12], and the proceedings of the workshop on texture analysis, edited by Burkhardt [16].

Part IV

Image Analysis