13 Simple Neighborhoods

13.1 Introduction

In the last two chapters we became acquainted with neighborhood operations for performing averaging and detecting edges. In fact, we only studied the very simplest structures in a local neighborhood: constant areas and discontinuities. However, a local neighborhood could also contain patterns. In this chapter, we discuss the simplest class of such patterns, which we will call simple neighborhoods. As an introduction, we examine what types of simple patterns can be used to make an object distinguishable from a background for the human visual system.

Our visual system can easily recognize objects that do not differ from a background by their mean gray value but only by the orientation or scale of a pattern, as demonstrated in Fig. 13.1. To perform this recognition task with a digital image processing system, we need operators that determine the orientation and the scale of the pattern. After such an operation, a gray scale image is converted into a *feature image*. In the feature image, we can distinguish patterns that differ by orientation or scale in the same way we can distinguish gray values.

We denote local neighborhoods that can be described by an orientation as simple neighborhoods. The development of suitable operators for orientation and scale is an important and necessary requirement for analysis of more complex structures. It is interesting to observe that the meaning of one and the same local structure may be quite different, as illustrated in Fig. 13.2 for 2-D images:

- In the simplest case, the observed scene consists of objects and a background with uniform radiance (Fig. 13.2a). Then, a gray value change in a local neighborhood indicates that an edge of an object is encountered and the analysis of orientation yields the orientation of the edge.
- In Fig. 13.2b, the objects differ from the background by the orientation of the *texture*. Now, the local spatial structure does not indicate an edge but characterizes the texture of the objects. The analysis of texture will be discussed in Chapter 15.
- In image sequences, the local structure in the space-time domain is determined by motion, as illustrated by Fig. 13.2c for a 2-D space-

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

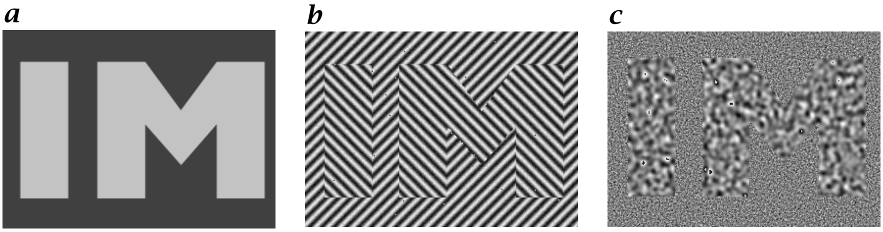

***Figure 13.1:** An object can be distinguished from the background because it differs in **a** gray value, **b** the orientation of a pattern, or **c** the scale of a pattern.*

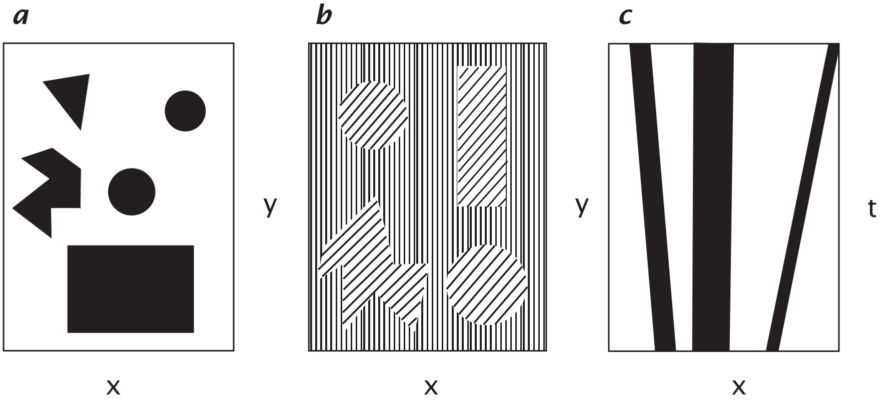

***Figure 13.2:** Three different interpretations of local structures in 2-D images: **a** edge between uniform object and background; **b** orientation of pattern; **c** orientation in a 2-D space-time image indicating the velocity of 1-D objects.*

time image. Motion is an important feature, just like any other, for identifying objects and will be treated in detail in Chapter 14.

Although the three examples refer to entirely different image data, they have in common that the local structure is characterized by an orientation, i. e., the gray values change locally only in one direction. In this sense, the concept of orientation is an extension of the concept of edges.

13.2 Properties of Simple Neighborhoods

13.2.1 Representation in the Spatial Domain

The mathematical description of a local neighborhood is best done with continuous functions. This approach has two significant advantages. First, it is much easier to formulate the concepts and to study their properties analytically. As long as the corresponding discrete image satisfies the sampling theorem, all the results derived from continuous functions remain valid, as the sampled image is an exact representation of the continuous gray value function. Second, we can now distinguish between errors inherent to the chosen approach and those that are only introduced by the discretization.

A local neighborhood with ideal local orientation is characterized by the fact that the gray value only changes in one direction. In all other directions it is constant. As the gray values are constant along lines, local orientation is also denoted as *linear symmetry* [9]. More recently, the term *simple neighborhood* has been coined by Granlund and Knutsson [64]. If we orient the coordinate system along the principal directions, the gray values become a 1-D function of only one coordinate. Generally, we will denote the direction of local orientation with a unit vector $\bar{\boldsymbol{n}}$ perpendicular to the lines of constant gray values. Then, a simple neighborhood is mathematically represented by

$$g(\boldsymbol{x}) = g(\boldsymbol{x}^T\bar{\boldsymbol{n}}), \tag{13.1}$$

where we denote the scalar product simply by $\boldsymbol{x}^T\bar{\boldsymbol{n}}$. We will use this simplified notation throughout this chapter. Equation (13.1) is also valid for image data with more than two dimensions. The projection of the vector $\boldsymbol{x}$ onto the unit vector $\bar{\boldsymbol{n}}$ makes the gray values depend only on a scalar quantity, the coordinate in the direction of $\bar{\boldsymbol{n}}$ (Fig. 13.3). It is easy to verify that this representation is correct by computing the gradient:

$$\nabla g(\boldsymbol{x}^T\bar{\boldsymbol{n}}) = \begin{bmatrix} \dfrac{\partial g(\boldsymbol{x}^T\bar{\boldsymbol{n}})}{\partial x_1} \\ \vdots \\ \dfrac{\partial g(\boldsymbol{x}^T\bar{\boldsymbol{n}})}{\partial x_W} \end{bmatrix} = \begin{bmatrix} \bar{n}_1 g'(\boldsymbol{x}^T\bar{\boldsymbol{n}}) \\ \vdots \\ \bar{n}_W g'(\boldsymbol{x}^T\bar{\boldsymbol{n}}) \end{bmatrix} = \bar{\boldsymbol{n}} g'(\boldsymbol{x}^T\bar{\boldsymbol{n}}). \tag{13.2}$$

With g' we denote the derivative of g with respect to the scalar variable $\boldsymbol{x}^T\bar{\boldsymbol{n}}$. In the hyperplane perpendicular to the gradient, the values remain locally constant. Equation Eq. (13.2) proves that the gradient lies in the direction of $\bar{\boldsymbol{n}}$.

13.2.2 Representation in the Fourier Domain

A simple neighborhood also has a special form in Fourier space. In order to derive it, we first assume that the whole image is described by Eq. (13.1), i. e., $\bar{\boldsymbol{n}}$ does not depend on the position. Then — from the very fact that a simple neighborhood is constant in all directions except $\bar{\boldsymbol{n}}$ — we infer that the Fourier transform must be confined to a line. The direction of the line is given by $\bar{\boldsymbol{n}}$:

$$g(\boldsymbol{x}^T\bar{\boldsymbol{n}}) \quad \circ\!\!-\!\!\!-\!\!\bullet \quad \hat{g}(k)\delta(\boldsymbol{k} - \bar{\boldsymbol{n}}(\boldsymbol{k}^T\bar{\boldsymbol{n}})), \tag{13.3}$$

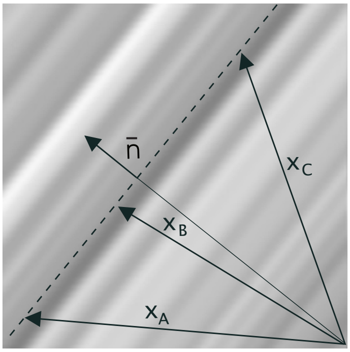

***Figure 13.3:** Illustration of a linear symmetric or simple neighborhood. The gray values depend only on a coordinate given by a unit vector $\bar{\boldsymbol{n}}$.*

where k denotes the coordinate in the Fourier domain in the direction of $\bar{\boldsymbol{n}}$. The argument in the δ function is only zero when $\boldsymbol{k}$ is parallel to $\bar{\boldsymbol{n}}$. In a second step, we now restrict Eq. (13.3) to a local neighborhood by multiplying $g(\boldsymbol{x}^T\bar{\boldsymbol{n}})$ with a window function $w(\boldsymbol{x}-\boldsymbol{x}_0)$ in the spatial domain. Thus, we select a local neighborhood around $\boldsymbol{x}_0$. The size and shape of the neighborhood is determined by the window function. A window function that gradually decreases to zero diminishes the influence of pixels as a function of their distance from the outer pixel. Multiplication in the space domain corresponds to a convolution in the Fourier domain (Section 2.3). Thus,

$$\boxed{w(\boldsymbol{x}-\boldsymbol{x}_0)\cdot g(\boldsymbol{x}^T\bar{\boldsymbol{n}}) \quad \circ\!\!-\!\!\bullet \quad \hat{w}(\boldsymbol{k}) * \hat{g}(k)\delta(\boldsymbol{k}-\bar{\boldsymbol{n}}(\boldsymbol{k}^T\bar{\boldsymbol{n}})),} \tag{13.4}$$

where $\hat{w}(\boldsymbol{k})$ is the Fourier transform of the window function.

The limitation to a local neighborhood, thus, blurs the line in Fourier space to a “sausage-like” shape. Because of the reciprocity of scales between the two domains, its thickness is inversely proportional to the size of the window. From this elementary relation, we can already conclude qualitatively that the accuracy of the orientation estimate is directly related to the ratio of the window size to the wavelength of the smallest structures in the window.

13.2.3 Vector Representation of Local Neighborhoods

For an appropriate representation of simple neighborhoods, it is first important to distinguish *orientation* from *direction*. The direction is defined over the full angle range of 2π (360°). Two vectors that point in opposite directions, i. e., differ by 180°, are different. The gradient vector, for example, always points into the direction into which the gray

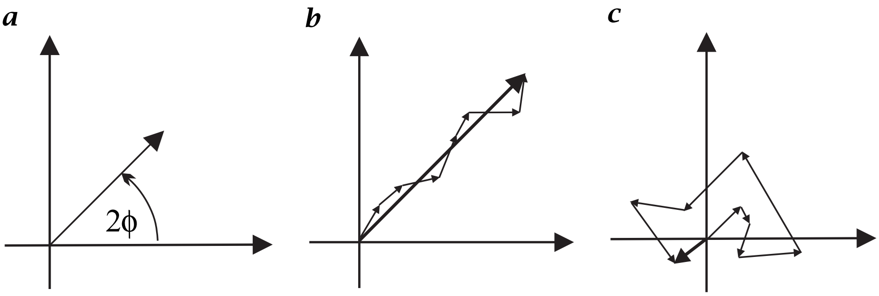

Figure 13.4: *Representation of local orientation as a vector:* ***a*** *the orientation vector;* ***b*** *averaging of orientation vectors from a region with homogeneous orientation;* ***c*** *same for a region with randomly distributed orientation.*

values are increasing. With respect to a bright object on a dark background, this means that the gradient at the edge is pointing towards the object. In contrast, to describe the direction of a local neighborhood, an angle range of 360° makes no sense. We cannot distinguish between patterns that are rotated by 180°. If a pattern is rotated by 180°, it still has the same direction. Thus, the direction of a simple neighborhood is different from the direction of a gradient. While for the edge of an object, gradients pointing in opposite directions are conflicting and inconsistent, for the direction of a simple neighborhood this is consistent information.

In order to distinguish the two types of "directions", we will speak of *orientation* in all cases where an angle range of only 180° is required. Orientation is still, of course, a *cyclic* quantity. Increasing the orientation beyond 180° flips it back to 0°. Therefore, an appropriate representation of orientation requires an angle doubling.

After this discussion of the principles of representing orientation, we are ready to think about an appropriate representation of simple neighborhoods. Obviously, a scalar quantity with just the doubled orientation angle is not appropriate. It seems to be useful to add a certainty measure that describes how well the neighborhood approximates a simple neighborhood. The scalar quantity and the certainty measure can be put together to form a vector. We set the magnitude of the vector to the certainty measure and the direction of the vector to the doubled orientation angle (Fig. 13.4a). This vector representation of orientation has two significant advantages.

First, it is more suitable for further processing than a separate representation of the orientation by two scalar quantities. Take, for example, averaging. Vectors are summed up by chaining them together, and the resulting sum vector is the vector from the starting point of the first vector to the end point of the last vector (Fig. 13.4b). The weight of an individual vector in the vector sum is given by its length. In this way, the certainty of the orientation measurement is adequately taken into account. The vectorial representation of local orientation shows suitable averaging properties. In a region with homogeneous orientation the vectors line up to a large vector (Fig. 13.4b), i. e., a certain orientation estimate. In a region with randomly distributed orientation, however, the resulting vector remains small, indicating that no significant local orientation is present (Fig. 13.4c).

Second, it is difficult to display orientation as a gray scale image. While orientation is a cyclic quantity, the gray scale representation shows an unnatural jump between the smallest angle and the largest one. This jump dominates the appearance of the orientation images and, thus, does not give a good impression of the orientation distribution. The orientation vector can be well represented, however, as a color image. It appears natural to map the certainty measure onto the luminance and the orientation angle as the hue of the color. Our attention is then drawn to the bright parts in the images where we can distinguish the colors well. The darker a color is, the more difficult it gets to distinguish the different colors visually. In this way, our visual impression coincides with the orientation information in the image.

13.3 First-Order Tensor Representation

13.3.1 The Structure Tensor

The vectorial representation discussed in Section 13.2.3 is incomplete. Although it is suitable for representing the orientation of simple neighborhoods, it cannot distinguish between neighborhoods with constant values and isotropic orientation distribution (e. g., uncorrelated noise). Both cases result in an orientation vector with zero magnitude.

Therefore, it is obvious that an adequate representation of gray value changes in a local neighborhood must be more complex. Such a representation should be able to determine a unique orientation (given by a unit vector $\bar{\boldsymbol{n}}$) and to distinguish constant neighborhoods from neighborhoods without local orientation.

A suitable representation can be introduced by the following optimization strategy to determine the orientation of a simple neighborhood. The optimum orientation is defined as the orientation that shows the least deviations from the directions of the gradient. A suitable measure for the deviation must treat gradients pointing in opposite directions equally. The squared scalar product between the gradient vector and the unit vector representing the local orientation $\bar{\boldsymbol{n}}$ meets this criterion:

$$(\nabla g^T \bar{\boldsymbol{n}})^2 = |\nabla g|^2 \cos^2\left(\angle(\nabla g, \bar{\boldsymbol{n}})\right). \qquad (13.5)$$

This quantity is proportional to the cosine squared of the angle between the gradient vector and the orientation vector and is thus maximal when ∇g and $\bar{\boldsymbol{n}}$ are parallel or antiparallel, and zero if they are perpendicular to each other. Therefore, the following integral is maximized in a W-dimensional local neighborhood:

$$\int w(\boldsymbol{x}-\boldsymbol{x}') \left(\nabla g(\boldsymbol{x}')^T \bar{\boldsymbol{n}}\right)^2 \mathrm{d}^W x', \tag{13.6}$$

where the window function w determines the size and shape of the neighborhood around a point $\boldsymbol{x}$ in which the orientation is averaged. The maximization problem must be solved for each point $\boldsymbol{x}$. Equation Eq. (13.6) can be rewritten in the following way:

$$\bar{\boldsymbol{n}}^T \boldsymbol{J} \bar{\boldsymbol{n}} \rightarrow \text{maximum} \tag{13.7}$$

with

$$\boldsymbol{J} = \int w(\boldsymbol{x}-\boldsymbol{x}') \left(\nabla g(\boldsymbol{x}') \nabla g(\boldsymbol{x}')^T\right) \mathrm{d}^W x',$$

where $\nabla g \nabla g^T$ denotes an outer (Cartesian) product. The components of this symmetric $W \times W$ tensor, named the *structure tensor*, are

$$\boxed{J_{pq}(\boldsymbol{x}) = \int\limits_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}') \left(\frac{\partial g(\boldsymbol{x}')}{\partial x'_p} \frac{\partial g(\boldsymbol{x}')}{\partial x'_q}\right) \mathrm{d}^W x'.} \tag{13.8}$$

These equations indicate that a tensor is an adequate first-order representation of a local neighborhood. The term first-order has a double meaning. First, only first-order derivatives are involved. Second, only simple neighborhoods can be described in the sense that we can analyze in which direction(s) the gray values change. More complex structures such as structures with multiple orientations cannot be distinguished.

The complexity of Eqs. (13.7) and (13.8) somewhat obscures their simple meaning. The tensor is symmetric. By a rotation of the coordinate system, it can be brought into a diagonal form. Then, Eq. (13.7) reduces in the 2-D case to

$$J' = [\bar{n}'_1, \bar{n}'_2] \begin{bmatrix} J'_{11} & 0 \\ 0 & J'_{22} \end{bmatrix} \begin{bmatrix} \bar{n}'_1 \\ \bar{n}'_2 \end{bmatrix} \quad \rightarrow \quad \text{maximum.} \tag{13.9}$$

A unit vector $\bar{\boldsymbol{n}}' = [\cos\theta \;\; \sin\theta]$ in the direction θ gives the values

$$J' = J'_{11} \cos^2\theta + J'_{22} \sin^2\theta.$$

Without loss of generality, we assume that $J'_{11} \geq J'_{22}$. Then, it is obvious that the unit vector $\bar{\boldsymbol{n}}' = [1 \;\; 0]^T$ maximizes Eq. (13.9). The maximum value is J'_{11}. In conclusion, this approach not only yields a tensor

Table 13.1: *Eigenvalue classification of the structure tensor in 2-D images.*

Condition	rank($\boldsymbol{J}$)	Description
$\lambda_1 = \lambda_2 = 0$	0	Both eigenvalues are zero. The mean squared magnitude of the gradient $(\lambda_1 + \lambda_2)$ is zero. The local neighborhood has constant values.
$\lambda_1 > 0, \lambda_2 = 0$	1	One eigenvalue is zero. The values do not change in the direction of the corresponding eigenvector. The local neighborhood is a simple neighborhood with ideal orientation.
$\lambda_1 > 0, \lambda_2 > 0$	2	Both eigenvalues are unequal to zero. The gray values change in all directions. In the special case of $\lambda_1 = \lambda_2$, we speak of an isotropic gray value structure as it changes equally in all directions.

representation for the local neighborhood but also shows the way to determine the orientation. Essentially, we have to solve what is known as an *eigenvalue problem*. The eigenvalues λ_w and eigenvectors $\boldsymbol{e}_w$ of a $W \times W$ matrix are defined by:

$$\boldsymbol{J}\boldsymbol{e}_w = \lambda_w \boldsymbol{e}_w. \tag{13.10}$$

An eigenvector $\boldsymbol{e}_w$ of $\boldsymbol{J}$ is thus a vector that is not turned in direction by multiplication with the matrix $\boldsymbol{J}$ but is only multiplied by a scalar factor, the eigenvalue λ_w. This implies that the structure tensor becomes diagonal in a coordinate system that is spanned by the eigenvectors Eq. (13.9). For our further discussion it is important to keep the following basic facts about the eigenvalues of a symmetric matrix in mind:

1. The eigenvalues are all real and non-negative.
2. The eigenvectors form an orthogonal basis.

According to the *maximization problem* formulated here, the eigenvector to the maximum eigenvalue gives the orientation of the local neighborhood.

13.3.2 Classification of Eigenvalues

The power of the tensor representation becomes apparent if we classify the eigenvalues of the structure tensor. The classifying criterion is the number of eigenvalues that are zero. If an eigenvalue is zero, this means that the gray values in the direction of the corresponding eigenvector do not change. The number of zero eigenvalues is also closely related to the rank of a matrix. The *rank* of a matrix is defined as the dimension of the subspace for which $\boldsymbol{J}\boldsymbol{k} \neq \boldsymbol{0}$. The space for which is $\boldsymbol{J}\boldsymbol{k} = \boldsymbol{0}$ is denoted

Table 13.2: *Eigenvalue classification of the structure tensor in 3-D (volumetric) images.*

Condition	rank($\boldsymbol{J}$)	Description
$\lambda_1 = \lambda_2 = \lambda_3 = 0$	0	The gray values do not change in any direction; constant neighborhood.
$\lambda_1 > 0, \lambda_2 = \lambda_3 = 0$	1	The gray values change only in one direction. This direction is given by the eigenvector to the non-zero eigenvalue. The neighborhood includes a boundary between two objects or a layered texture. In a space-time image, this means a constant motion of a spatially oriented pattern ("planar wave").
$\lambda_1 > 0, \lambda_2 > 0, \lambda_3 = 0$	2	The gray values change in two directions and are constant in a third. The eigenvector to the zero eigenvalue gives the direction of the constant gray values.
$\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0$	3	The gray values change in all three directions.

as the *null space*. The dimension of the null space is the dimension of the matrix minus the rank of the matrix and equal to the number of zero eigenvalues. We will perform an analysis of the eigenvalues for two and three dimensions. In two and three dimensions, we can distinguish the cases summarized in Tables 13.1 and 13.2, respectively.

In practice, it will not be checked whether the eigenvalues are zero but below a critical threshold that is determined by the noise level in the image.

13.3.3 Orientation Vector

With the simple convolution and point operations discussed in the previous section, we computed the components of the structure tensor. In this section, we solve the eigenvalue problem to determine the orientation vector. In two dimensions, we can readily solve the eigenvalue problem. The orientation angle can be determined by rotating the inertia tensor into the principal axes coordinate system:

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} J_{11} & J_{12} \\ J_{12} & J_{22} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

Using the trigonometric identities $\sin 2\theta = 2\sin\theta\cos\theta$ and $\cos 2\theta = \cos^2\theta - \sin^2\theta$, the matrix multiplications result in

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} =$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} J_{11}\cos\theta - J_{12}\sin\theta & J_{11}\sin\theta + J_{12}\cos\theta \\ -J_{22}\sin\theta + J_{12}\cos\theta & J_{22}\cos\theta + J_{12}\sin\theta \end{bmatrix} =$$

$$\begin{bmatrix} J_{11}\cos^2\theta + J_{22}\sin^2\theta\text{-}J_{12}\sin 2\theta & 1/2(J_{11}\text{-}J_{22})\sin 2\theta + J_{12}\cos 2\theta \\ 1/2(J_{11}\text{-}J_{22})\sin 2\theta + J_{12}\cos 2\theta & J_{11}\sin^2\theta + J_{22}\cos^2\theta + J_{12}\sin 2\theta \end{bmatrix}$$

Now we can compare the matrix coefficients on the left and right side of the equation. Because the matrices are symmetric, we have three equations with three unknowns, θ, λ_1, and λ_2. Although the equation system is nonlinear, it can readily be solved for θ.

A comparison of the off-diagonal elements on both sides of the equation

$$1/2(J_{11} - J_{22})\sin 2\theta + J_{12}\cos 2\theta = 0 \tag{13.11}$$

yields the orientation angle as

$$\boxed{\tan 2\theta = \frac{2J_{12}}{J_{22} - J_{11}}.} \tag{13.12}$$

Without defining any prerequisites, we have obtained the anticipated angle doubling for orientation. Since $\tan 2\theta$ is gained from a quotient, we can regard the dividend as the y and the divisor as the x component of a vector and can form the *orientation vector* $\boldsymbol{o}$, as introduced by Granlund [63]:

$$\boldsymbol{o} = \begin{bmatrix} J_{22} - J_{11} \\ 2J_{12} \end{bmatrix}. \tag{13.13}$$

The argument of this vector gives the orientation angle and the magnitude a certainty measure for local orientation.

The result of Eq. (13.13) is remarkable in that the computation of the components of the orientation vector from the components of the orientation tensor requires just one subtraction and one multiplication by two. As these components of the orientation vector are all we need for further processing steps we do not need the orientation angle or the magnitude of the vector. Thus, the solution of the eigenvalue problem in two dimensions is trivial.

13.3.4 Coherency

The orientation vector reduces local structure to local orientation. From three independent components of the symmetric tensor still only two are used. When we fail to observe an orientated structure in a neighborhood, we do not know whether no gray value variations or distributed orientations are encountered. This information is included in the not yet used component of the tensor, $J_{11} + J_{22}$, which gives the mean square magnitude of the gradient. Consequently, a well-equipped structure operator needs to include also the third component. A suitable linear combination is

$$\boxed{\boldsymbol{s} = \begin{bmatrix} J_{11} + J_{22} \\ J_{22} - J_{11} \\ 2J_{12} \end{bmatrix}.} \tag{13.14}$$

This structure operator contains the two components of the orientation vector and, as an additional component, the mean square magnitude of the gradient, which is a rotation-invariant parameter. Comparing the latter with the magnitude of the orientation vector, a constant gray value area and an isotropic gray value structure without preferred orientation can be distinguished. In the first case, both squared quantities are zero, in the second only the magnitude of the orientation vector. In the case of a perfectly oriented pattern, both quantities are equal. Thus their ratio seems to be a good *coherency measure* c_c for local orientation:

$$c_c = \frac{\sqrt{(J_{22} - J_{11})^2 + 4J_{12}^2}}{J_{11} + J_{22}} = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}. \tag{13.15}$$

The coherency ranges from 0 to 1. For ideal local orientation ($\lambda_2 = 0$, $\lambda_1 > 0$) it is one, for an isotropic gray value structure ($\lambda_1 = \lambda_2 > 0$) it is zero.

13.3.5 Color Coding of the 2-D Structure Tensor

In Section 13.2.3 we discussed a color representation of the orientation vector. The question is whether it is also possible to represent the structure tensor adequately as a color image. A symmetric 2-D tensor has three independent pieces of information Eq. (13.14), which fit well to the three degrees of freedom available to represent color, for example luminance, hue, and saturation.

A color represention of the structure tensor requires only two slight modifications as compared to the color representation for the orientation vector. First, instead of the length of the orientation vector, the squared magnitude of the gradient is mapped onto the intensity. Second, the coherency measure Eq. (13.15) is used as the saturation. In the color representation for the orientation vector, the saturation is always one. The angle of the orientation vector is still represented as the hue.

In practice, a slight modification of this color representation is useful. The squared magnitude of the gradient shows variations too large to be displayed in the narrow dynamic range of a display screen with only 256 luminance levels. Therefore, a suitable normalization is required. The basic idea of this normalization is to compare the squared magnitude of the gradient with the noise level. Once the gradient is well above the noise level it is regarded as a significant piece of information. This train of thoughts suggests the following normalization for the intensity I:

$$I = \frac{J_{11} + J_{22}}{(J_{11} + J_{22}) + \gamma \sigma_n^2}, \qquad (13.16)$$

where σ_n is an estimate of the standard deviation of the noise level. This normalization provides a rapid transition of the luminance from one, when the magnitude of the gradient is larger than σ_n, to zero when the gradient is smaller than σ_n. The factor γ is used to optimize the display.

13.3.6 Implementation

The structure tensor (Section 13.3.1) or the inertia tensor (Section 13.5.1) can be computed straightforwardly as a combination of *linear convolution* and *nonlinear point operations*. The partial derivatives in Eqs. (13.8) and (13.64) are approximated by discrete derivative operators. The integration weighted with the window function is replaced by a convolution with a smoothing filter which has the shape of the window function. If we denote the discrete partial derivative operator with respect to the coordinate p by the operator $\mathcal{D}_p$ and the (isotropic) smoothing operator by $\mathcal{B}$, the local structure of a gray value image can be computed with the structure tensor operator

$$\mathcal{J}_{pq} = \mathcal{B}(\mathcal{D}_p \cdot \mathcal{D}_q). \qquad (13.17)$$

The equation is written in an operator notation. Pixelwise multiplication is denoted by $\cdot$ to distinguish it from successive application of convolution operators. Equation Eq. (13.17) says, in words, that the $\mathcal{J}_{pq}$ component of the tensor is computed by convolving the image independently with $\mathcal{D}_p$ and $\mathcal{D}_q$, multiplying the two images pixelwise, and smoothing the resulting image with $\mathcal{B}$.

These operators are valid in images of any dimension $W \geq 2$. In a W-dimensional image, the structure tensor has $W(W+1)/2$ independent components, hence 3 in 2-D, 6 in 3-D, and 10 in 4-D images. These components are best stored in a multichannel image with $W(W+1)/2$ components.

The smoothing operations consume the largest number of operations. Therefore, a fast implementation must, in the first place, apply a fast smoothing algorithm. A fast algorithm can be established based on the general observation that higher-order features always show a lower resolution than the features they are computed from. This means that the structure tensor can be stored on a coarser grid and thus in a smaller image. A convenient and appropriate subsampling rate is to reduce the scale by a factor of two by storing only every second pixel in every second row.

These procedures lead us in a natural way to multigrid data structures which are discussed in detail in Chapter 5. Multistep averaging is discussed in detail in Section 11.5.1.

Storing higher-order features on coarser scales has another significant advantage. Any subsequent processing is sped up simply by the fact that many fewer pixels have to be processed. A linear scale reduction by a factor of two results in a reduction in the number of pixels and the number of computations by a factor of 4 in two and 8 in three dimensions.

Figure 13.5 illustrates all steps to compute the structure tensor and derived quantities using the ring test pattern. This test pattern is particularly suitable for orientation analysis since it contains all kinds of orientations and wave numbers in one image.

The accuracy of the orientation angle strongly depends on the implementation of the derivative filters. The straightforward implementation of the algorithm using the standard derivative filter mask $1/2\,[1\ 0\ -1]$ (Section 12.4.3) or the *Sobel operator* (Section 12.7.3) results in surprisingly high errors (Fig. 13.6b), with a maximum error in the orientation angle of more than 7° at a wave number of $\tilde{k} = 0.7$. The error depends on both the wave number and the orientation of the local structure. For orientation angles in the direction of axes and diagonals, the error vanishes. The high error and the structure of the error map result from the transfer function of the derivative filter. The transfer function shows significant deviation from the transfer function for an ideal derivative filter for high wave numbers (Section 12.3). According to Eq. (13.12), the orientation angle depends on the ratio of derivatives. Along the axes, one of the derivatives is zero and, thus, no error occurs. Along the diagonals, the derivatives in the x and y directions are the same. Consequently, the error in both cancels in the ratio of the derivatives as well.

The error in the orientation angle can be significantly suppressed if better derivative filters are used. Figure 13.6 shows the error in the orientation estimate using two examples of the optimized Sobel operator (Section 12.7.5) and the least-squares optimized operator (Section 12.6).

The little extra effort in optimizing the derivative filters thus pays off in an accurate orientation estimate. A residual angle error of less than 0.5° is sufficient for almost all applications. The various derivative filters discussed in Sections 12.4 and 12.7 give the freedom to balance computational effort with accuracy.

An important property of any image processing algorithm is its *robustness*. This term denotes the sensitivity of an algorithm against noise.

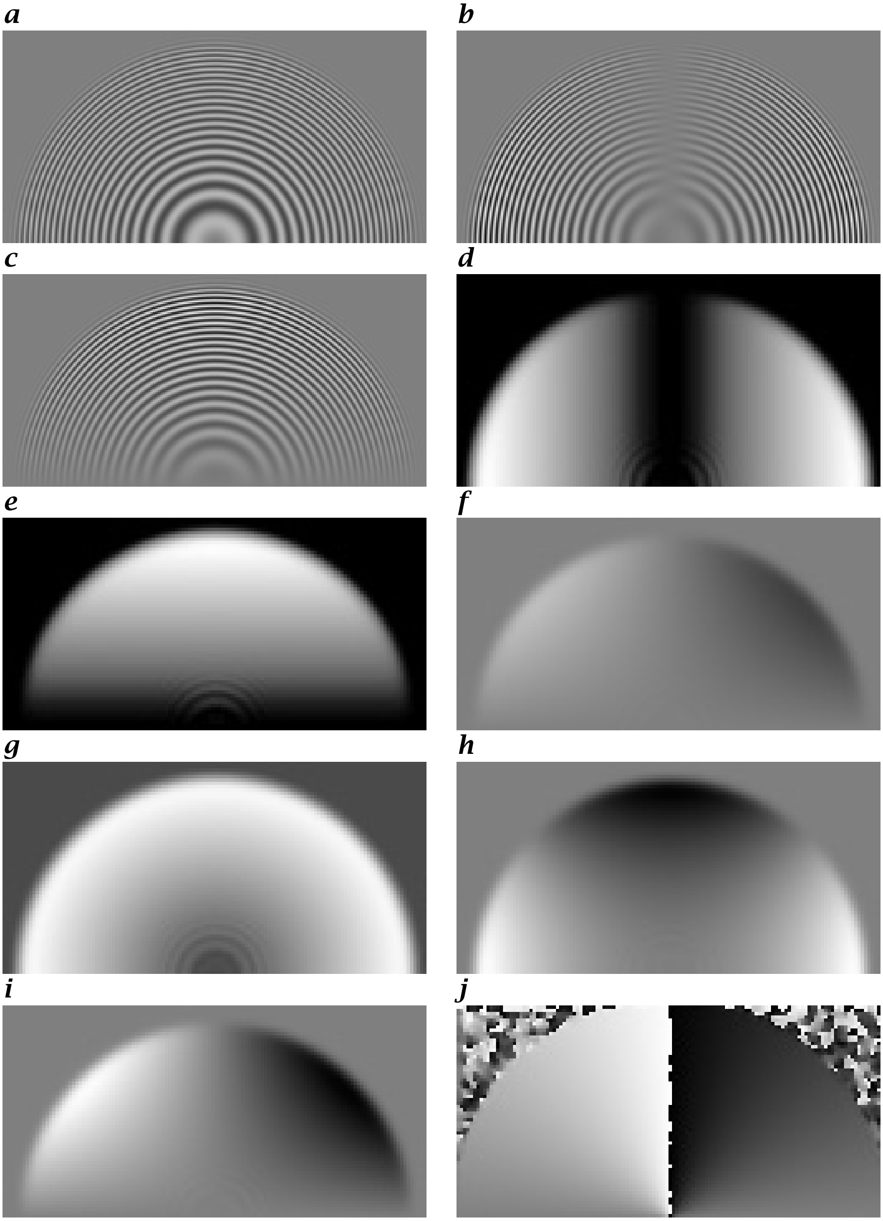

Figure 13.5: *Steps to compute the structure tensor:* ***a*** *original ring test pattern;* ***b*** *horizontal derivation* $\mathcal{D}_x$*;* ***c*** *vertical derivation* $\mathcal{D}_y$*;* ***d–f*** *averaged components for the structure tensor* $J_{11} = \mathcal{B}(\mathcal{D}_x \cdot \mathcal{D}_x), J_{22} = \mathcal{B}(\mathcal{D}_y \cdot \mathcal{D}_y), J_{12} = \mathcal{B}(\mathcal{D}_x \cdot \mathcal{D}_y)$*;* ***g*** *squared magnitude of gradient* $J_{11} + J_{22}$*;* ***h*** x *component of orientation vector* $J_{11} - J_{22}$*;* ***i*** y *component of orientation vector* $2J_{12}$*;* ***j*** *orientation angle from* $[-\pi/2, \pi/2]$ *mapped to a gray scale interval from [0, 255].*

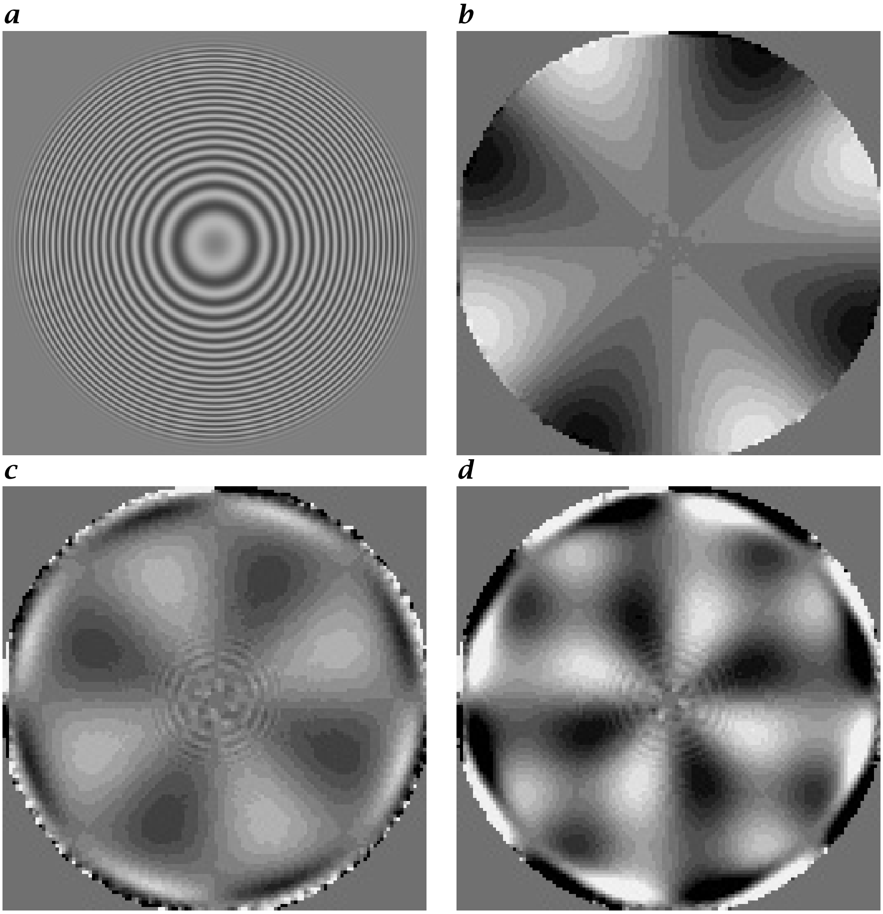

***Figure 13.6:** Systematic errors for the orientation angle estimate using different derivative operators: **a** original ring test pattern with a maximum normalized wave number $\tilde{k} = 0.7$; error maps for **b** the Sobel operator (angle range ±7° in 16 discrete steps), **c** the optimized Sobel operator, and **d** the least squares optimized operator (angle range ±0.7° in 16 discrete steps) with $r = 3$.*

Two questions are important. First, how large is the error of the estimated features in noisy images? To answer this question, the laws of statistics are used to study error propagation. In this context, noise makes the estimates only uncertain but not erroneous. The mean — if we make a sufficient number of estimates — is still correct. However, a second question arises. In noisy images an operator can also give results that are biased, i. e., the mean can show a significant deviation from the correct value. In the worst case, an algorithm can even become unstable and deliver meaningless results.

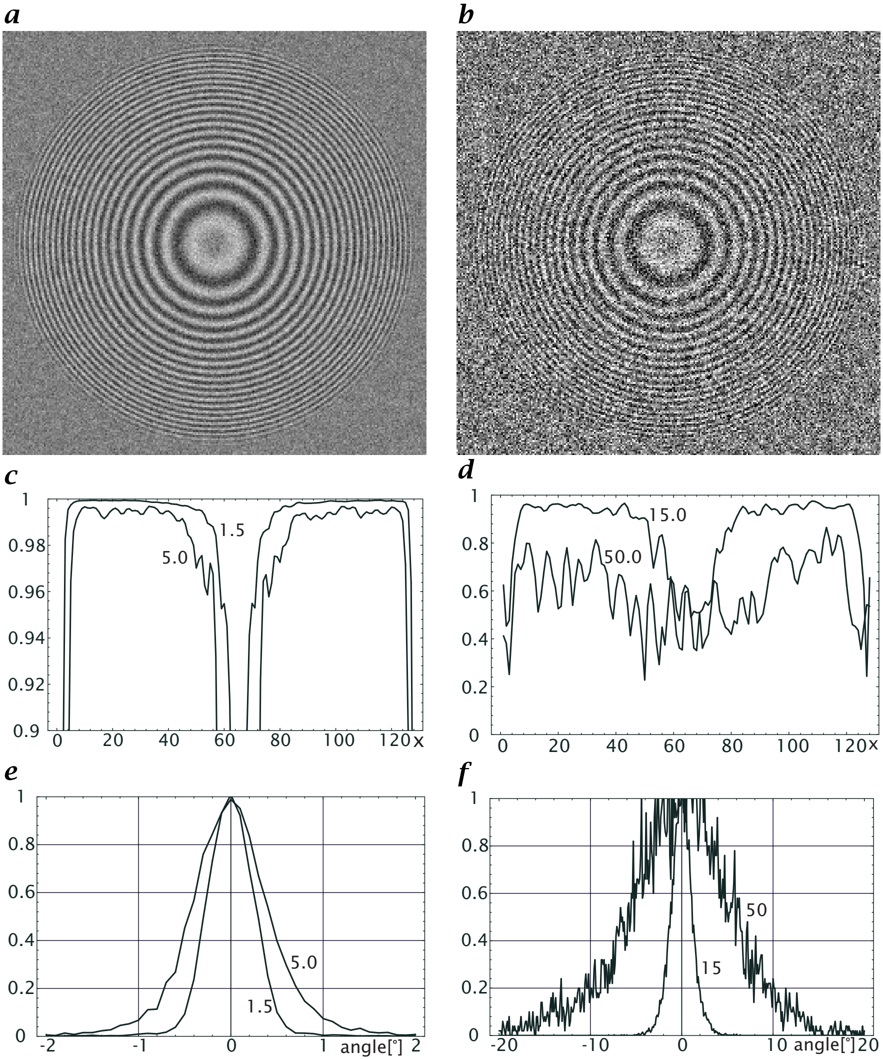

Figure 13.7: *Orientation analysis with a noisy ring test pattern using the optimized Sobel operator: ring pattern with amplitude 50, standard deviation of normal distributed noise* ***a*** *15, and* ***b*** *50;* ***c*** *and* ***d*** *radial cross section of the coherency measure for standard deviations of the noise level of 1.5 and 5, 15 and 50, respectively;* ***e*** *and* ***f*** *histograms of angle error for the same conditions.*

Figure 13.7 demonstrates that the estimate of orientation is also a remarkably robust algorithm. Even with a low signal-to-noise ratio, the orientation estimate is still correct if a suitable derivative operator is used. With increasing noise level, the coherency (Section 13.3.4) decreases and the statistical error of the orientation angle estimate increases (Fig. 13.7).

13.4 *Local Wave Number and Phase*

13.4.1 *Phase*

So far in this chapter we have discussed in detail the analysis of simple neighborhoods with respect to their orientation. In this section we proceed with another elementary property of simple neighborhoods. In Chapter 5 we stressed the importance of the scale for image processing. Thus we must not only ask in which directions the gray values change. We must also ask how fast the gray values change. This question leads us to the concept of the *local wave number*.

The key to determining the local wave number is the *phase* of the signal. As an introduction we discuss a simple example and consider the one-dimensional periodic signal

$$g(x) = g_0 \cos(kx). \tag{13.18}$$

The argument of the cosine function is known as the phase of the periodic signal:

$$\phi(x) = kx. \tag{13.19}$$

The equation shows that the phase is a linear function of the position and the wave number. Thus we obtain the wave number of the periodic signal by computing the first-order spatial derivative of the phase signal

$$\frac{\partial \phi(x)}{\partial x} = k. \tag{13.20}$$

These simple considerations re-emphasize the significant role of the phase in image processing that we discussed already in Section 2.3.5. We will discuss two related approaches for determining the phase of a signal, the *Hilbert transform* (Section 13.4.2) and the *quadrature filter* (Section 13.4.5) before we introduce efficient techniques to compute the local wave number from phase gradients.

13.4.2 *Hilbert Transform and Hilbert Filter*

In order to explain the principle of computing the phase of a signal, we take again the example of the simple periodic signal from the previous section. We suppose that an operator is available to delay the signal by a phase of 90°. This operator would convert the $g(x) = g_0 \cos(kx)$ signal into a $g'(x) = -g_0 \sin(kx)$ signal as illustrated in Fig. 13.8. Using both signals, the phase of $g(x)$ can be computed by

$$\phi(g(x)) = \arctan\left(\frac{-g'(x)}{g(x)}\right). \tag{13.21}$$

As only the ratio of $g'(x)$ and $g(x)$ goes into Eq. (13.21), the phase is indeed independent of amplitude. If we take the signs of the two functions $g'(x)$ and $g(x)$ into account, the phase can be computed over the full range of 360°.

Thus all we need to determine the phase of a signal is a linear operator that shifts the phase of a signal by 90°. Such an operator is known as the *Hilbert*

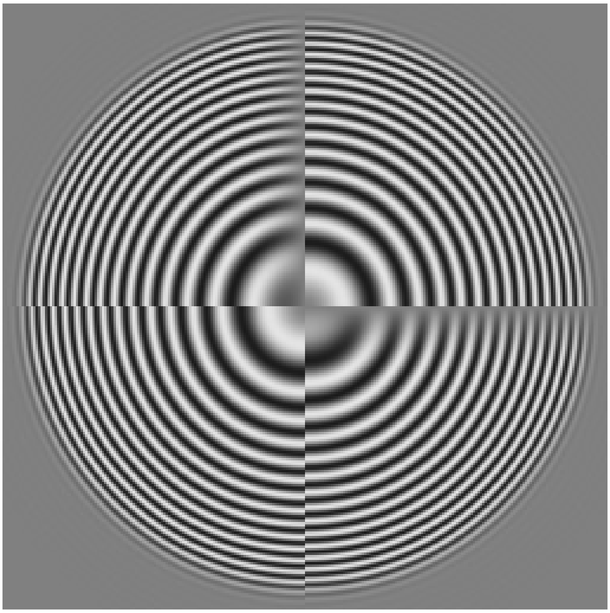

Figure 13.8: *Application of the Hilbert filter to the ring test pattern: upper left quadrant: in the horizontal direction; lower right quadrant: in the vertical direction.*

filter ***H*** or *Hilbert operator* $\mathcal{H}$ and has the transfer function

$$\hat{h}(k) = \begin{cases} \mathrm{i} & k > 0 \\ 0 & k = 0 \\ -\mathrm{i} & k < 0 \end{cases}. \tag{13.22}$$

The magnitude of the transfer function is one as the amplitude remains unchanged. As the Hilbert filter has a purely imaginary transfer function, it must be of odd symmetry to generate a real-valued signal. Therefore positive wave numbers are shifted by 90° ($\pi/2$) and negative wave numbers by $-90°$ ($-\pi/2$). A special situation is given for the wave number zero where the transfer function is also zero. This exception can be illustrated as follows. A signal with wave number zero is a constant. It can be regarded as a cosine function with infinite wave number sampled at the phase zero. Consequently, the Hilbert filtered signal is the corresponding sine function at phase zero, that is, zero.

Because of the discontinuity of the transfer function of the Hilbert filter at the origin, its point spread function is of infinite extent

$$h(x) = -\frac{1}{\pi x}. \tag{13.23}$$

The convolution with Eq. (13.23) can be written as

$$g_h(x) = \frac{1}{\pi} \int\limits_{-\infty}^{\infty} \frac{g(x')}{x' - x} \mathrm{d}x'. \tag{13.24}$$

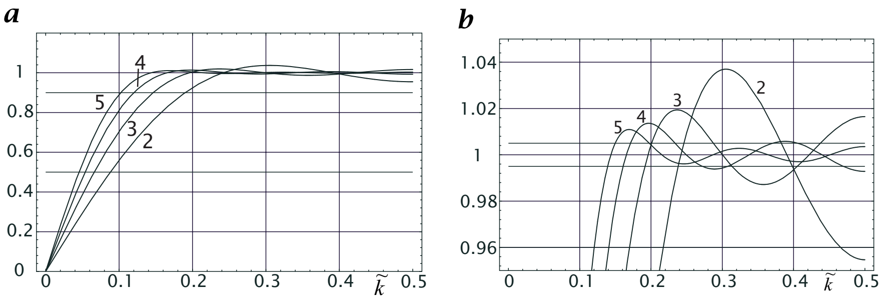

Figure 13.9: ***a*** *Transfer functions of a family of least-squares optimized Hilbert operators according to Eq. (13.25) for the four filter coefficients* $R = 2, 3, 4, 5$*.* ***b*** *sector of* ***a*** *to better show the deviations from an ideal Hilbert filter. As the filters are symmetric around* $\tilde{k} = 0.5$ *only a wave number range from 0–0.5 is shown.*

This integral transform is known as the *Hilbert transform* [128].

Because the convolution mask of the Hilbert filter is infinite, it is impossible to design an exact discrete Hilbert filter for arbitrary signals. This is only possible if we restrict the class of signals to which it is applied. Thus the following approach is taken to design an effective implementation of a Hilbert filter.

First, the filter should precisely shift the phase by $\pi/2$. This requirement comes from the fact that we cannot afford an error in the phase because it includes the position information. A wave-number dependent phase shift would cause wave-number dependent errors. This requirement is met by any convolution kernel of odd symmetry.

Second, requirements for a magnitude of one can be relaxed if the Hilbert filter is applied to a bandpassed signal, e. g., the Laplace pyramid. Then, the Hilbert filter must only show a magnitude of one in the passband range of the bandpass filter used. This approach avoids the discontinuities in the transfer function at the wave number zero and thus results in finite-sized convolution kernels.

Optimized Hilbert filters are generated with the same least-squares techniques used above for interpolation filters (Section 10.6.2) and first-order derivative filters (Section 12.6).

Because of the odd symmetry of the Hilbert filter, the following formulation is used

$$\hat{h}(\tilde{k}) = 2\mathrm{i}\sum_{v=1}^{R} h_v \sin\left((2v-1)\pi\tilde{k}\right). \tag{13.25}$$

Note that we have only used sine functions with odd wave numbers. This causes the transfer function also to become symmetric around $\tilde{k} = 1/2$ and leads to a filter mask with alternating zeros

$$[h_R, 0, \cdots, h_2, 0, h_1, 0, -h_1, 0, -h_2, \cdots, 0, -h_R]. \tag{13.26}$$

The mask has $4R-1$ coefficients, $2R-1$ of which are zero. Figure 13.9 shows the transfer functions optimized with the least squares technique for $R = 2, 3, 4, 5$. The filter with $R = 4$ (a mask with 15 coefficients)

$$\boldsymbol{h} = \{0.6208, 0.1683, 0.0630, 0.0191\}, \tag{13.27}$$

for instance, has an amplitude error of only slightly larger than 1.0 % in the wave number range [0.16, 0.84] and by design no phase error. The convolution with this mask requires 4 multiplications and 7 additions/subtractions.

13.4.3 *Analytic Signal*

A real-valued signal and its Hilbert transform can be combined into a complex-valued signal by

$$g_a = g - \mathrm{i}g_h. \qquad (13.28)$$

This complex-valued signal is denoted as the *analytic function* or *analytic signal*. According to Eq. (13.28) the analytic filter has the point spread function

$$a(x) = 1 + \frac{\mathrm{i}}{\pi x} \qquad (13.29)$$

and the transfer function

$$\hat{a}(k) = \begin{cases} 2 & k > 0 \\ 1 & k = 0 \\ 0 & k < 0 \end{cases}. \qquad (13.30)$$

Thus all negative wave numbers are suppressed. Although the transfer function of the analytic filter is real, it results in a complex signal because it is asymmetric. For a real signal no information is lost by suppressing the negative wave numbers. They can be reconstructed as the Fourier transform of a real signal is Hermitian (Section 2.3.4). The analytic signal can be regarded as just another representation of a real signal with two important properties. The magnitude of the analytic signal gives the *local amplitude*

$$|\mathcal{A}|^2 = \mathcal{I} \cdot \mathcal{I} + \mathcal{H} \cdot \mathcal{H}. \qquad (13.31)$$

and the argument the *local phase*

$$\arg(\mathcal{A}) = \arctan\left(\frac{-\mathcal{H}}{\mathcal{I}}\right), \qquad (13.32)$$

using $\mathcal{A}$ and $\mathcal{H}$ for the analytic and Hilbert operators, respectively.

The original signal and its Hilbert transform can be obtained from the analytic signal using Eq. (13.28) by

$$\begin{array}{rcl} g(x) & = & (g_a(x) + g_a^*(x))/2 \\ g_h(x) & = & \mathrm{i}(g_a(x) - g_a^*(x))/2. \end{array} \qquad (13.33)$$

The concept of the analytic signal also makes it easy to extend the ideas of local phase into multiple dimensions. The transfer function of the analytic operator uses only the positive wave numbers, i. e., only half of the Fourier space. If we extend this partitioning to multiple dimensions, we have more than one choice to partition the Fourier space into two half spaces. Instead of the wave number, we can take the scalar product between the wave number vector $\boldsymbol{k}$ and any unit vector $\bar{\boldsymbol{n}}$ and suppress the half space for which the scalar product $\boldsymbol{k}\bar{\boldsymbol{n}}$ is negative:

$$\hat{a}(\boldsymbol{k}) = \begin{cases} 2 & \boldsymbol{k}\bar{\boldsymbol{n}} > 0 \\ 1 & \boldsymbol{k}\bar{\boldsymbol{n}} = 0 \\ 0 & \boldsymbol{k}\bar{\boldsymbol{n}} < 0 \end{cases}. \qquad (13.34)$$

The unit vector $\bar{\boldsymbol{n}}$ gives the direction in which the Hilbert filter is to be applied. The definition Eq. (13.34) of the transfer function of the analytic signal implies that the Hilbert operator can only be applied to directionally filtered signals. This results from the following considerations. For one-dimensional signals we have seen that a discrete Hilbert filter does not work well for small wave numbers (Fig. 13.9). In multiple dimensions this means that a Hilbert filter does not work well if $\tilde{\boldsymbol{k}}\bar{\boldsymbol{n}} \ll 1$. Thus no wave numbers near an orthogonal to the direction of the Hilbert filter may exist, in order to avoid errors.

This fact makes the application of Hilbert filters and thus the determination of the local phase in higher-dimensional signals significantly more complex. It is not sufficient to use bandpass filtered images, e. g., a Laplace pyramid (Section 5.2.3). In addition, the bandpass filtered images must be further decomposed into directional components. At least as many directional components as the dimensionality of the space are required.

13.4.4 *Monogenic Signal*

The extension of the Hilbert transform from a 1-D signal to higher-dimensional signals is not satisfactory because it can only be applied to directionally filtered signals. For wave numbers close to the separation plane, the Hilbert transform does not work. What is really required is an isotropic extension of the Hilbert transform. It is obvious that no scalar-valued transform for a multidimensional signal can be both isotropic and of odd symmetry.

A vector-valued extension of the analytic signal meets both requirements. It is known as the *monogenic signal* and was introduced to image processing by Felsberg and Sommer [44]. The monogenic signal is constructed from the original signal and its *Riesz transform.* The transfer function of the Riesz transform is given by

$$\hat{\boldsymbol{h}}(\boldsymbol{k}) = \mathrm{i}\frac{\boldsymbol{k}}{|\boldsymbol{k}|}. \tag{13.35}$$

The magnitude of the vector $\boldsymbol{h}$ is one for all values of $\boldsymbol{k}$. The Riesz transform is thus isotropic. It is also of odd symmetry because

$$\hat{\boldsymbol{h}}(-\boldsymbol{k}) = -\hat{\boldsymbol{h}}(\boldsymbol{k}). \tag{13.36}$$

The Riesz transform can be applied to a signal of any dimension. For a 1-D signal it reduces to the Hilbert transform.

For a 2-D signal the transfer function of the Riesz transform can be written using polar coordinates as

$$\hat{\boldsymbol{h}}(\boldsymbol{k}) = \mathrm{i}\left[\frac{k\cos\theta}{|k|}, \frac{k\sin\theta}{|k|}\right]^T. \tag{13.37}$$

The transfer function is similar to the transfer function for the gradient operator (Section 12.2.1, Eq. (12.2)). It differs by the fact that the transfer function for the Riesz transform is divided by the magnitude of the wavenumber.

The convolution mask or PSF of the Riesz transform is given by

$$\boldsymbol{h}(\boldsymbol{x}) = -\frac{\boldsymbol{x}}{2\pi\,|\boldsymbol{x}|^3}. \tag{13.38}$$

The original signal and the signal convolved by the Riesz transform can be combined for a 2-D signal to the 3-D monogenic signal as

$$\boldsymbol{g}_m(\boldsymbol{x}) = [p, q_1, q_2]^T \quad \text{with} \quad p = g, q_1 = h_1 * g, q_2 = h_2 * g. \tag{13.39}$$

The local amplitude of the monogenic signal is given as the norm of the vector of the monogenic signal as in the case of the analytic signal (Eq. (13.31)):

$$|\boldsymbol{g}_m|^2 = p^2 + q_1^2 + q_2^2. \tag{13.40}$$

The monogenic signal does not only give an estimate for the *local phase* ϕ as the *analytic signal* does. The monogenic signal gives also an estimate of the *local orientation* θ by the following relations:

$$p = a\cos\phi, \quad q_1 = a\sin\phi\cos\theta, \quad q_2 = a\sin\phi\sin\theta. \tag{13.41}$$

We can thus conclude that the monogenic signal combines an estimate of local orientation and local phase. This is of high significance for image processing because the two most important features of a local neighborhood, the local orientation and the local wave number can be estimated in a unified way.

13.4.5 *Quadrature Filters*

Quadrature filters are an alternative approach to getting a pair of signals that differ only by a phase shift of 90° ($\pi/2$). It is easiest to introduce the complex form of the quadrature filters. Essentially, the transfer function of a *quadrature filter* is also zero for $\boldsymbol{k}\bar{\boldsymbol{n}} < 0$, like the transfer function of the analytic filter. However, the magnitude of the transfer function is not one but can be any arbitrary real-valued function $h(\boldsymbol{k})$:

$$\hat{q}(\boldsymbol{k}) = \begin{cases} 2h(\boldsymbol{k}) & \boldsymbol{k}\bar{\boldsymbol{n}} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{13.42}$$

The quadrature filter thus also transforms a real-valued signal into an analytical signal. In contrast to the analytical operator, a wave number weighting is applied. From the complex form of the quadrature filter, we can derive the real quadrature filter pair by observing that they are the part of Eq. (13.42) with even and odd symmetry. Thus

$$\begin{aligned} \hat{g}_+(\boldsymbol{k}) &= (\hat{q}(\boldsymbol{k}) + \hat{q}(-\boldsymbol{k}))/2, \\ \hat{g}_-(\boldsymbol{k}) &= (\hat{q}(\boldsymbol{k}) - \hat{q}(-\boldsymbol{k}))/2. \end{aligned} \tag{13.43}$$

The even and odd part of the quadrature filter pair show a phase shift of 90° and can thus can also be used to compute the local phase.

Quadrature filters can also be designed on the basis of the monogenic signal (Section 13.4.4). These quadrature filters have one component more than the dimension of the signal. The transfer function is

$$\hat{\boldsymbol{h}}(\boldsymbol{k}) = \left[\hat{q}_+(\boldsymbol{k}), \quad \mathrm{i}\boldsymbol{k}\hat{q}_+(\boldsymbol{k})/|\boldsymbol{k}|\right]^T. \tag{13.44}$$

The best-known quadrature filter pair is the *Gabor filter*. A Gabor filter is a bandpass filter that selects a certain wavelength range around the center wavelength $\boldsymbol{k}_0$ using the Gauss function. The complex transfer function of the Gabor filter is

$$\hat{g}(\boldsymbol{k}) = \begin{cases} \exp\left(|\boldsymbol{k} - \boldsymbol{k}_0|^2 \sigma_x^2/2\right) & \boldsymbol{k}\boldsymbol{k}_0 > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{13.45}$$

If $|\boldsymbol{k}_0|\sigma_x > 3$, Eq. (13.45) reduces to

$$\hat{g}(\boldsymbol{k}) = \exp\left(-|\boldsymbol{k} - \boldsymbol{k}_0)|^2 \sigma_x^2/2\right). \tag{13.46}$$

Using the relations in Eq. (13.43), the transfer function for the even and odd component are given by

$$\begin{aligned} \hat{g}_+(\boldsymbol{k}) &= \frac{1}{2}\left[\exp\left(-|\boldsymbol{k} - \boldsymbol{k}_0|^2\sigma_x^2/2\right) + \exp\left(-|\boldsymbol{k} + \boldsymbol{k}_0|^2\sigma_x^2/2\right)\right], \\ \hat{g}_-(\boldsymbol{k}) &= \frac{1}{2}\left[\exp\left(-|\boldsymbol{k} - \boldsymbol{k}_0|^2\sigma_x^2/2\right) - \exp\left(-|\boldsymbol{k} + \boldsymbol{k}_0|^2\sigma_x^2/2\right)\right]. \end{aligned} \tag{13.47}$$

The point spread function of these filters can be computed easily with the shift theorem (Theorem 2.3, p. 54, $\succ$R4):

$$\begin{aligned} g_+(\boldsymbol{x}) &= \cos(\boldsymbol{k}_0\boldsymbol{x}) \exp\left(-\frac{|\boldsymbol{x}|^2}{2\sigma_x^2}\right), \\ g_-(\boldsymbol{x}) &= \mathrm{i}\sin(\boldsymbol{k}_0\boldsymbol{x}) \exp\left(-\frac{|\boldsymbol{x}|^2}{2\sigma_x^2}\right), \end{aligned} \tag{13.48}$$

or combined into a complex filter mask:

$$g(\boldsymbol{x}) = \exp(\mathrm{i}\boldsymbol{k}_0\boldsymbol{x}) \exp\left(-\frac{|\boldsymbol{x}|^2}{2\sigma_x^2}\right). \tag{13.49}$$

Gabor filters are useful for bandpass-filtering images and performing image analysis in the space/wave number domain. Figure 13.10 illustrates an application [*Riemer*, 1991; *Riemer et al.*, 1991]. An image with short wind-generated water surface waves is decomposed by a set of Gabor filters. The center wavelength $\boldsymbol{k}_0$ was set in the x direction, parallel to the wind direction. The filters had the center wavelength in octave distances at 1.2, 2.4, and 4.8 cm wavelengths. The bandwidth was set proportional to the center wave number.

The left column of images in Fig. 13.10 shows the filtering with the even Gabor filter, the right column the local amplitude, which is directly related to the energy of the waves. The filtered images show that waves with different wavelength are partly coupled. In areas where the larger waves have large amplitudes, also the small-scale waves (capillary waves) have large amplitudes. The energy of waves is not equally distributed over the water surface.

An extension of this analysis to image sequences gives a direct insight into the nonlinear wave-wave interaction processes. Figure 13.11 shows the temporal evolution of one row of images from Fig. 13.10. As we will discuss in detail in Section 14.2.4, the slope of the structures in these space-time images towards the time axis is directly proportional to the speed of the moving objects.

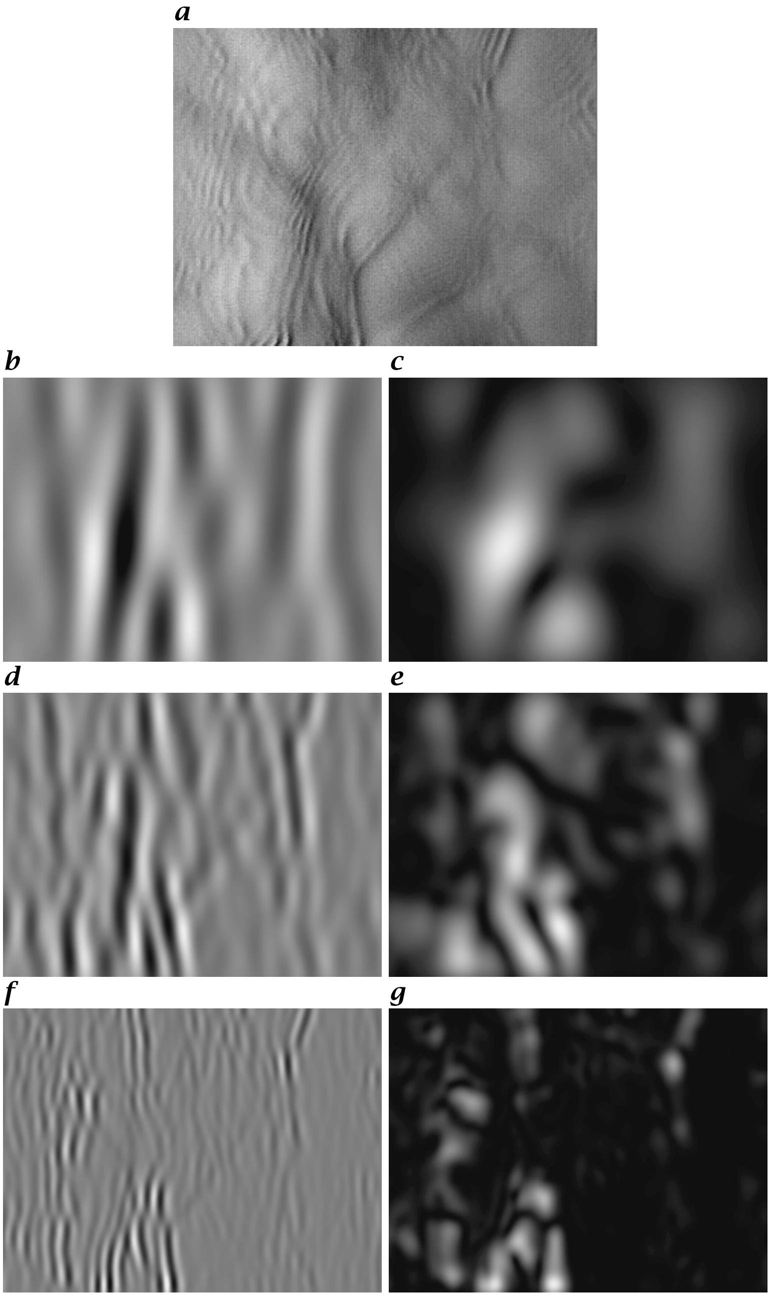

Figure 13.10: *Analysis of an image (**a**, 40 cm × 30 cm) from wind-generated water surface waves. The intensity is proportional to the along-wind component of the slope of the waves. The even part (**b**, **d**, **f**) and squared magnitude (energy, **c**, **e**, **g**) of the Gabor-filtered images with center wavelength at 48, 24, and 12 mm, respectively.*

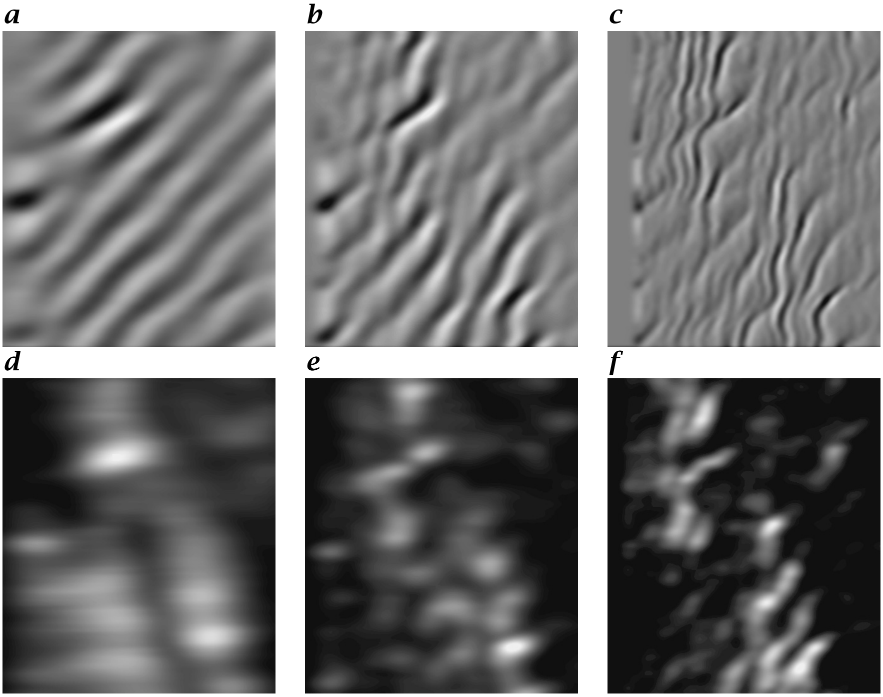

Figure 13.11: *Analysis of a 5 s long space-time slice in wind direction of an image sequence from short wind-generated water surface waves. The time axis is vertically oriented. Even part (**a**–**c**) and squared magnitude (energy, **d**–**f**) of the Gabor-filtered images with center wavelength at 48, 24, and 12 mm, respectively.*

It can be observed nicely that the small waves are modulated by the large waves and that the *group velocity* (speed of the wave energy) of the small waves is slower than the phase speed for the capillary waves.

13.4.6 *Local Wave Number Determination*

In order to determine the local wave number, we just need to compute the first spatial derivative of the phase signal (Section 13.4.1, Eq. (13.20)). This derivative has to be applied in the same direction as the Hilbert or quadrature filter has been applied. The phase is given by either

$$\phi(\boldsymbol{x}) = \arctan\left(\frac{-g_h(\boldsymbol{x})}{g(\boldsymbol{x})}\right) \tag{13.50}$$

or

$$\phi(\boldsymbol{x}) = \arctan\left(\frac{-q_+(\boldsymbol{x})}{q_-(\boldsymbol{x})}\right), \tag{13.51}$$

where q_+ and q_- denote the signals filtered with the even and odd part of the quadrature filter.

Direct computation of the partial derivatives from Eqs. (13.50) and (13.51) is not advisable, however, because of the inherent discontinuities in the phase signal. A phase computed with the inverse tangent restricts the phase to the main interval $[-\pi, \pi[$ and thus leads inevitably to a wrapping of the phase signal from π to $-\pi$ with the corresponding discontinuities.

As pointed out by Fleet [48], this problem can be avoided by computing the phase gradient directly from the gradients of $q_+(\boldsymbol{x})$ and $q_-(\boldsymbol{x})$. The result is

$$\boldsymbol{k} = \nabla\phi(\boldsymbol{x}) = \nabla \arctan(-p(\boldsymbol{x})/q(\boldsymbol{x})) = \frac{q\,\nabla p - p\,\nabla q}{p^2 + q^2} \tag{13.52}$$

This formulation of the phase gradient also eliminates the need for using trigonometric functions to compute the phase signal and is, therefore, significantly faster.

It is significantly more complex to computate the local wave number from the *monogic signal* (Section 13.4.4), because we need to use three signals for 2-D signals. From Eq. (13.41) we obtain two different equations for the phase:

$$\phi_1 = \operatorname{arccot}\left(\frac{p\cos\theta}{q_1}\right), \quad \phi_2 = \operatorname{arccot}\left(\frac{p\sin\theta}{q_2}\right). \tag{13.53}$$

It is necessary to combine these equations because each of them gives no result for certain directions. The solution is use the *directional derivative* (Section 12.2.1). When we differentiate the phase in the direction of the wave-number vector, we directly obtain the magnitude of the wave-number vector:

$$k = \frac{\partial\phi}{\partial\bar{\boldsymbol{k}}} = \cos\theta\frac{\partial\phi_1}{\partial x} + \sin\theta\frac{\partial\phi_2}{\partial y}. \tag{13.54}$$

The terms $\cos\theta$ and $\sin\theta$ can also be obtained from Eq. (13.41):

$$\cos^2\theta = \frac{q_1^2}{q_1^2 + q_1^2} \quad \text{and} \quad \sin^2\theta = \frac{q_2^2}{q_1^2 + q_1^2}. \tag{13.55}$$

Then the magnitude of the wave-number vector results in

$$k = \frac{p(q_{1x} + q_{2y}) - q_1 p_x - q_2 p_y}{p^2 + q_1^2 + q_1^2}. \tag{13.56}$$

The components of the vector $\boldsymbol{k} = [k\cos\theta, h\sin\theta]$ can be computed by combining Eqs. (13.56) and (13.54).

13.5 *Further Tensor Representations*

In this section we examine several alternative approaches to describe local structure with tensors. The method of the inertia tensor in Section 13.5.1 considers local structure in Fourier space. The main emphasis in this section is, however, the synthesis of tensor methods with quadrature filters. These are techniques that combine the analysis of local orientation and local wave number.

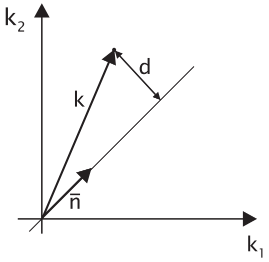

Figure 13.12: *Distance of a point in the wave number space from the line in the direction of the unit vector* $\bar{\boldsymbol{n}}$.

13.5.1 *The Inertia Tensor*

As a starting point, we consider what an ideally oriented gray value structure (Eq. (13.1)) looks like in the wave number domain. We can compute the Fourier transform of Eq. (13.1) more readily if we rotate the x_1 axis in the direction of $\bar{\boldsymbol{n}}$. Then the gray value function is constant in the x_2 direction. Consequently, the Fourier transform reduces to a δ line in the direction of $\bar{\boldsymbol{n}}$ (≻ R5).

It seems promising to determine local orientation in the Fourier domain, as all we have to compute is the orientation of the line on which the spectral densities are non-zero. Bigün and Granlund [9] devised the following procedure:

- Use a window function to select a small local neighborhood from an image.
- Fourier transform the windowed image. The smaller the selected window, the more blurred the spectrum will be (*uncertainty relation*, Theorem 2.7, p. 57). This means that even with an ideal local orientation we will obtain a rather band-shaped distribution of the spectral energy.
- Determine local orientation by fitting a straight line to the spectral density distribution. This yields the angle of the local orientation from the slope of the line.

The critical step of this procedure is fitting a straight line to the spectral densities in the Fourier domain. We cannot solve this problem exactly as it is generally *overdetermined*, but only minimize the measure of error. A standard error measure is the square of the magnitude of the vector (L2 norm; see Eq. (2.75) in Section 2.4.1). When fitting a straight line, we minimize the sum of the squares of the distances of the data points to the line:

$$\int_{-\infty}^{\infty} d^2(\boldsymbol{k}, \bar{\boldsymbol{n}}) |\hat{g}(\boldsymbol{k})|^2 \mathrm{d}^W k \rightarrow \text{minimum}. \tag{13.57}$$

The distance function is abbreviated using $d(\boldsymbol{k}, \bar{\boldsymbol{n}})$. The integral runs over the whole wave number space; the wave numbers are weighted with the spectral density $|\hat{g}(\boldsymbol{k})|^2$. Equation (13.57) is not restricted to two dimensions, but is generally valid for local orientation or linear symmetry in a W-dimensional space.

The distance vector $\boldsymbol{d}$ can be inferred from Fig. 13.12 to be

$$\boldsymbol{d} = \boldsymbol{k} - (\boldsymbol{k}^T \bar{\boldsymbol{n}})\bar{\boldsymbol{n}}. \tag{13.58}$$

The square of the distance is then given by

$$|\boldsymbol{d}|^2 = |\boldsymbol{k} - (\boldsymbol{k}^T\bar{\boldsymbol{n}})\bar{\boldsymbol{n}}|^2 = |\boldsymbol{k}|^2 - (\boldsymbol{k}^T\bar{\boldsymbol{n}})^2. \tag{13.59}$$

In order to express the distance more clearly as a function of the vector $\bar{\boldsymbol{n}}$, we rewrite it in the following manner:

$$|\boldsymbol{d}|^2 = \bar{\boldsymbol{n}}^T(\boldsymbol{I}(\boldsymbol{k}^T\boldsymbol{k}) - (\boldsymbol{k}\boldsymbol{k}^T))\bar{\boldsymbol{n}}, \tag{13.60}$$

where $\boldsymbol{I}$ is the unit diagonal matrix. Substituting this expression into Eq. (13.57) we obtain

$$\bar{\boldsymbol{n}}^T\boldsymbol{J}'\bar{\boldsymbol{n}} \to \text{minimum}, \tag{13.61}$$

where $\boldsymbol{J}'$ is a symmetric tensor with the diagonal elements

$$J'_{pp} = \sum_{q\neq p}\int_{-\infty}^{\infty} k_q^2|\hat{g}(\boldsymbol{k})|^2\mathrm{d}^W k \tag{13.62}$$

and the off-diagonal elements

$$J'_{pq} = -\int_{-\infty}^{\infty} k_p k_q|\hat{g}(\boldsymbol{k})|^2\mathrm{d}^W k, \quad p \neq q. \tag{13.63}$$

The tensor $\boldsymbol{J}'$ is analogous to a well-known physical quantity, the *inertia tensor*. If we replace the wave number coordinates by space coordinates and the spectral density $|\hat{g}(\boldsymbol{k})|^2$ by the specific density ρ, Eqs. (13.57) and (13.61) constitute the equation to compute the inertia of a rotary body rotating around the $\bar{\boldsymbol{n}}$ axis.

With this analogy, we can reformulate the problem of determining local orientation. We must find the axis about which the rotary body, formed from the spectral density in Fourier space, rotates with minimum inertia. This body might have different shapes. We can relate its shape to the different solutions we get for the eigenvalues of the inertia tensor and thus for the solution of the local orientation problem (Table 13.3).

We derived the inertia tensor approach in the Fourier domain. Now we will show how to compute the coefficients of the inertia tensor in the space domain.

The integrals in Eqs. (13.62) and (13.63) contain terms of the form

$$k_q^2|\hat{g}(\boldsymbol{k})|^2 = |\mathrm{i}k_q\hat{g}(\boldsymbol{k})|^2$$

and

$$k_pk_q|\hat{g}(\boldsymbol{k})|^2 = \mathrm{i}k_p\hat{g}(\boldsymbol{k})[\mathrm{i}k_q\hat{g}(\boldsymbol{k})]^*.$$

Integrals over these terms are *inner* or *scalar products* of the functions $\mathrm{i}k_p\hat{g}(\boldsymbol{k})$. Because the inner product is preserved under the Fourier transform (≻ R4), we can compute the corresponding integrals in the spatial domain as well. Multiplication of $\hat{g}(\boldsymbol{k})$ with $\mathrm{i}k_p$ in the wave number domain corresponds to performing the first spatial derivative in the direction of x_p in the space domain:

$$\begin{aligned} J'_{pp}(\boldsymbol{x}) &= \sum_{q\neq p}\int_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}')\left(\frac{\partial g}{\partial x_q}\right)^2 \mathrm{d}^W x' \\ J'_{pq}(\boldsymbol{x}) &= -\int_{-\infty}^{\infty} w(\boldsymbol{x}-\boldsymbol{x}')\frac{\partial g}{\partial x_p}\frac{\partial g}{\partial x_q}\mathrm{d}^W x'. \end{aligned} \tag{13.64}$$

Table 13.3: *Eigenvalue classification of the structure tensor in 3-D (volumetric) images.*

Condition	Explanation
Ideal local orientation	The rotary body is a line. For a rotation around this line, the inertia vanishes. Consequently, the eigenvector to the eigenvalue zero coincides with the direction of the line. The other eigenvector is orthogonal to the line, and the corresponding eigenvalue is unequal to zero and gives the rotation axis for maximum inertia.
Isotropic gray value structure	In this case, the rotary body is a kind of flat isotropic disk. A preferred direction does not exist. Both eigenvalues are equal and the inertia is the same for rotations around all axes. We cannot find a minimum.
Constant gray values	The rotary body degenerates to a point at the origin of the wave number space. The inertia is zero for rotation around any axis. Therefore both eigenvalues vanish.

In Eq. (13.64), we already included the weighting with the window function w to select a local neighborhood.

The structure tensor discussed in Section 13.3.1 Eq. (13.8) and the inertia tensor are closely related:

$$\boldsymbol{J}' = \mathrm{trace}(\boldsymbol{J})\boldsymbol{I} - \boldsymbol{J}. \tag{13.65}$$

From this relationship it is evident that both matrices have the same set of eigenvectors. The eigenvalues λ_p are related by

$$\lambda_p = \sum_{q=1}^{n} \lambda_q - \lambda'_p, \quad \lambda'_p = \sum_{q=1}^{n} \lambda_q - \lambda_p. \tag{13.66}$$

Consequently, we can perform the eigenvalue analysis with any of the two matrices. For the inertia tensor, the direction of local orientation is given by the minimum eigenvalue, but for the structure tensor it is given by the maximum eigenvalue.

13.5.2 *Further Equivalent Approaches*

In their paper on analyzing oriented patterns, Kass and Witkin [101] chose — at first glance — a completely different method. Yet it turns out to be equivalent to the tensor method, as will be shown in the following. They started with the idea of using *directional derivative* filters by differentiating a *difference of Gaussian* *filter* (*DoG*, Section 12.7.6) (written in operator notation)

$$\mathcal{R}(\Theta) = [\cos\Theta \;\; \sin\Theta] \begin{bmatrix} \mathcal{D}_x(\mathcal{B}_1 - \mathcal{B}_2) \\ \mathcal{D}_y(\mathcal{B}_1 - \mathcal{B}_2) \end{bmatrix} = [\cos\Theta \;\; \sin\Theta] \begin{bmatrix} \mathcal{R}_x \\ \mathcal{R}_y \end{bmatrix},$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ denote two Gaussian smoothing masks with different variances. The direction in which this directional derivative is maximal in a mean square sense gives the orientation normal to lines of constant gray values. This approach results in the following expression for the variance of the directional derivative:

$$\mathcal{V}(\Theta) = \mathcal{B}(\mathcal{R}(\Theta) \cdot \mathcal{R}(\Theta)). \tag{13.67}$$

The directional derivative is squared and then smoothed by a binomial filter. This equation can also be interpreted as the inertia of an object as a function of the angle. The corresponding inertia tensor has the form

$$\begin{bmatrix} \mathcal{B}(\mathcal{R}_y \cdot \mathcal{R}_y) & -\mathcal{B}(\mathcal{R}_x \cdot \mathcal{R}_y) \\ -\mathcal{B}(\mathcal{R}_x \cdot \mathcal{R}_y) & \mathcal{B}(\mathcal{R}_x \cdot \mathcal{R}_x) \end{bmatrix}. \tag{13.68}$$

Thus Kass and Witkin's approach is identical to the general inertia tensor method discussed in Section 13.5.1. They just used a special type of derivative filter.

Without being aware of either Bigün and Granlund [9] earlier or Knutsson [111] contemporary work, Rao and Schunck [162] and Rao [161] proposed the same structure tensor (denoting it as the moment tensor) as that we discussed in Section 13.3.1.

13.5.3 *Polar Separable Quadrature Filters*

Quadrature filters provide another way to analyze simple neighborhoods and to determine both the *local orientation* and the *local wave number*. Historically, this was the first technique for local structure analysis, pioneered by the work of Granlund [63]. The inertia and structure tensor techniques actually appeared later in the literature [9, 101, 161, 162].

The basic idea of the quadrature filter set technique is to extract structures in a certain wave number and direction range. In order to determine local orientation, we must apply a whole set of directional filters, with each filter being sensitive to structures of different orientation. We then compare the filter responses and obtain a maximum filter response from the directional filter whose direction coincides best with that of local orientation. Similarly, the quadrature filter set for different wave number ranges can be set up to determine the local wave number.

If we get a clear maximum in one of the filters but only little response in the others, the local neighborhood contains a locally oriented pattern. If the different filters give comparable responses, the neighborhood contains a distribution of oriented patterns.

So far, the concept seems to be straightforward, but a number of tricky problems needs to be solved. Which properties have to be met by the directional filters in order to ensure an exact determination of local orientation, if at all possible? For computational efficiency, we need to use a minimal number of filters to interpolate the angle of the local orientation. What is this minimal number?

The concepts introduced in this section are based on the work of Granlund [63], Knutsson [110], and Knutsson et al. [112], later summarized in a monograph by Granlund and Knutsson [64]. While the quadrature filter set techniques have been formulated by these authors for multiple dimensions, we will discuss here only the two-dimensional case.

We first discuss the design of quadrature filters that are suitable for the detection of both local orientation and local wave number. This leads to polar separable quadrature filters (Section 13.5.3). In a second step, we show how the orientation vector defined in Section 13.3.3 can be constructed by simple vector addition of the quadrature filter responses (Section 13.5.4). Likewise, in Section 13.5.5 we study the computation of the local wave number. Finally, Section 13.5.6 closes the circle by showing that the structure tensor can also be computed by a set of quadrature filters. Thus the tensor methods discussed in the first part of this chapter (Section 13.3) and the quadrature filter set technique differ only in some subtle points but otherwise give identical results.

For an appropriate set of directional filters, each filter should be a rotated copy of the others. This requirement implies that the transfer function of the filters can be separated into an angular part $d(\phi)$ and a wave number part $r(k)$. Such a filter is called *polar separable* and may be conveniently expressed in polar coordinates

$$\hat{q}(k,\phi) = \hat{r}(k)\hat{d}(\phi), \tag{13.69}$$

where $k = \sqrt{k_1^2 + k_2^2}$ and $\phi = \arctan(k_2/k_1)$ are the magnitude and argument of the wave number, respectively. For a set of directional filters, only the angular part of the transfer function is of importance, while the radial part must be the same for each filter but can be of arbitrary shape. The converse is true for a filter set to determine the local wave number.

Knutsson [110] suggested the following base quadrature filter

$$\begin{aligned} \hat{r}(k) &= \exp\left[-\frac{(\ln k - \ln k_0)^2}{(B/2)^2 \ln 2}\right] \\ \hat{d}(\phi) &= \begin{cases} \cos^{2l}(\phi - \phi_k) & |\phi - \phi_k| < \pi/2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{13.70}$$

In this equation, the complex notation for quadrature filters (Section 13.4.5) is used. The filter is directed into the angle ϕ_k. The unit vector in this direction is $\bar{\boldsymbol{d}}_k = [\cos\phi_k, \sin\phi_k]$.

The filter is continuous, since the cosine function is zero in the partition plane for the two half spaces ($|\phi - \phi_k| = \pi/2$ or $\bar{\boldsymbol{d}}_k\boldsymbol{k} = 0$). Using the unit vector $\bar{\boldsymbol{d}}_k$ in the direction of the filter, the angular part of the filter can also be written as:

$$\hat{d}(\boldsymbol{k}) = \begin{cases} (\boldsymbol{k}\bar{\boldsymbol{d}}_k)^{2l} & (\boldsymbol{k}\bar{\boldsymbol{d}}_k) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{13.71}$$

The constant k_0 in Eq. (13.70) denotes the peak wave number. The constant B determines the half-width of the wave number in number of octaves and l the angular resolution of the filter. In a logarithmic wave number scale, the filter has the shape of a Gaussian function. Therefore the radial part has a *lognormal* shape.

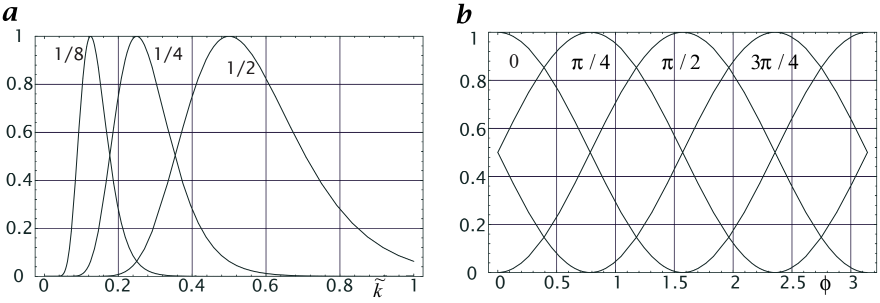

Figure 13.13: ***a*** *Radial and* ***b*** *angular part of quadrature filter according to Eq. (13.70) with $l = 1$ and $B = 2$ in different directions and with different peak wave numbers.*

For the real even and the imaginary odd filter of the quadrature filter pair, the radial part is the same and only the angular part differs:

$$\begin{aligned} \hat{d}_+(\phi) &= \cos^{2l}(\phi - \phi_k) \\ \hat{d}_-(\phi) &= \mathrm{i}\cos^{2l}(\phi - \phi_k)\ \mathrm{sign}(\cos(\phi - \phi_k)). \end{aligned} \tag{13.72}$$

Figure 13.13 shows the radial and angular part of the transfer function for different k_0 and ϕ_k. A set of directional filters is obtained by a suitable choice of different ϕ_k:

$$\phi_k = \frac{\pi k}{K} \quad k = 0, 1, \cdots, K - 1. \tag{13.73}$$

Knutsson used four filters with 45° increments in the directions 22.5°, 67.5°, 112.5°, and 157.5°. These directions have the advantage that only *one* filter kernel has to be designed. The kernels for the filter in the other directions are obtained by mirroring the kernels at the axes and diagonals.

These filters were designed in the wave number space. The filter coefficients are obtained by inverse Fourier transformation. If we choose a reasonably small filter mask, we will cut off a number of non-zero filter coefficients. This causes deviations from the ideal transfer function.

Therefore, Knutsson modified the filter kernel coefficient using an optimization procedure in such a way that it approaches the ideal transfer function as closely as possible. It turned out that at least a 15×15 filter mask is necessary to get a good approximation of the anticipated transfer function.

13.5.4 *Determination of the Orientation Vector*

The local orientation can be computed from the responses of the four quadrature filters by vector addition. The idea of the approach is simple. We assign to the individual directional filters an orientation vector. The magnitude of the vector corresponds to the response of the quadrature filter. The direction of the vector is given by the double angle of the filter direction (Section 13.3.3). In this representation each filter response shows how well the orientation of the

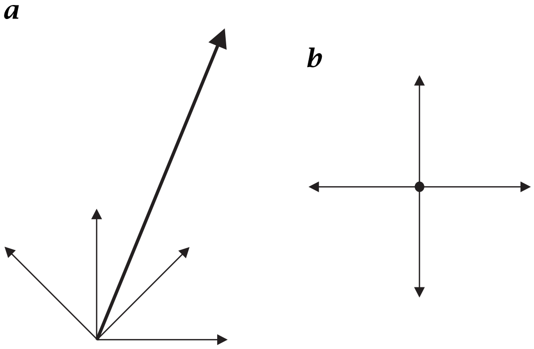

Figure 13.14: *Computation of local orientation by vector addition of the four filter responses. An example is shown where the neighborhood is isotropic concerning orientation: all four filter responses are equal. The angles of the vectors are equal to the filter directions in* ***a*** *and double the filter directions in* ***b*** *.*

pattern is directed in the direction of the filter. An estimate of the orientation vector is then given as the vector sum of the filter responses.

Using a represention with complex numbers for the orientation vector, we can write the filter response for the filter in ϕ_k direction as

$$\mathcal{Q}_{\phi_k} = |\mathcal{Q}| \exp(2\mathrm{i}\phi_k). \tag{13.74}$$

Then the orientation vector as the vector sum of the filter responses can be written as

$$\mathcal{O} = \sum_{k=0}^{K-1} \mathcal{Q}_{\phi_k}. \tag{13.75}$$

Figure 13.14 illustrates why an angle doubling is necessary for the vector addition to obtain the orientation vector. An example is taken where the responses from all four filters are equal. In this case the neighborhood contains structures in all directions. Consequently, we observe no local orientation and the vector sum of all filter responses vanishes. This happens if we double the orientation angle (Fig. 13.14b), but not if we omit this step (Fig. 13.14a).

After these more qualitative considerations, we will prove that we can compute the local orientation exactly when the local neighborhood is ideally oriented in an arbitrary direction ϕ_0. As a result, we will also learn the least number of filters we need. We can simplify the computations by only considering the angular terms, as the filter responses show the same wave number dependence. The quick reader can skip this proof.

Using Eq. (13.74), Eq. (13.70), and Eq. (13.73) we can write the angular part of the filter response of the kth filter as

$$\hat{d}_k(\phi_0) = \exp\left(2\pi\mathrm{i}k/K\right)\cos^{2l}(\phi_0 - \pi k/K).$$

The cosine function is decomposed into the sum of two complex exponentials:

$$\begin{aligned}\hat{d}_k(\phi_0) &= \frac{1}{2^{2l}} \exp(2\pi \mathrm{i}k/K)\left[\exp(\mathrm{i}(\phi_0 - \pi k/K)) + \exp(-\mathrm{i}(\phi_0 - \pi k/K))\right]^{2l} \\ &= \frac{1}{2^{2l}} \exp(2\pi \mathrm{i}k/K) \sum_{j=0}^{2l} \binom{2l}{j} \exp(\mathrm{i}j(\phi_0 - \pi k/K)) \exp(-\mathrm{i}(2l-j)(\phi_0 - \pi k/K)) \\ &= \frac{1}{2^{2l}} \sum_{j=0}^{2l} \binom{2l}{j} \exp(\mathrm{i}(j-l)2\phi_0) \exp(2\pi \mathrm{i}(1+l-j)(k/K)).\end{aligned}$$

Now we sum up the vectors of all the K directional filters:

$$\sum_{k=0}^{K-1} \hat{d}_k = \frac{1}{2^{2l}} \sum_{j=0}^{2l} \binom{2l}{j} \exp(\mathrm{i}(j-l)2\phi_0) \sum_{k=0}^{K-1} \exp(2\pi \mathrm{i}(1+l-j)(k/K)).$$

The complex double sum can be solved if we carefully analyze the inner sum over k. If $j = l+1$ the exponent is zero. Consequently, the sum is K. Otherwise, the sum represents a geometric series with the factor $\exp(2\pi \mathrm{i}(1+l-j)(k/K))$ and the sum

$$\sum_{k=0}^{K-1} \exp(2\pi \mathrm{i}(1+l-j)(k/K)) = \frac{1-\exp(2\pi \mathrm{i}(1+l-j))}{1-\exp(2\pi \mathrm{i}(1+l-j)/K)}. \qquad (13.76)$$

We can use Eq. (13.76) only if the denominator $\neq 0 \ \forall j = 0, 1, \cdots, 2l$; consequently $K > 1 + l$. With this condition the sum vanishes. This result has a simple geometric interpretation. The sum consists of vectors which are equally distributed on the unit circle. The angle between two consecutive vectors is $2\pi k/K$.

In conclusion, the inner sum in Eq. (13.76) reduces to K for $j = l+1$, otherwise it is zero. Therefore the sum over j contains only the term with $j = l+1$. The final result

$$\sum_{k=0}^{K-1} \hat{d}_k = \frac{K}{2^{2l}} \binom{2l}{l+1} \exp(\mathrm{i}2\phi_0) \qquad (13.77)$$

shows a vector with the angle of the local orientation doubled. This concludes the proof. ■

The proof of the exactness of the vector addition techniques gives also the minimal number of directional filters required. From $l > 0$ and $K > l+1$ we conclude that at least $K = 3$ directional filters are necessary. We can also illustrate this condition intuitively. If we have only two filters ($K = 2$), the vector responses of these two filters lie on a line (Fig. 13.15a). Thus orientation determination is not possible. Only with three or four filters can the sum vector point in all directions (Fig. 13.15b, c).

With a similar derivation, we can prove another important property of the directional quadrature filters. The sum over the transfer functions of the K filters results in an isotropic function for $K > l$:

$$\sum_{k=0}^{K-1} \cos^{2l}(\phi - \pi k/K) = \frac{K}{2^{2l}} \binom{2l}{l} \frac{K}{2^{2l}} \frac{(2l)!}{l!^2}. \qquad (13.78)$$

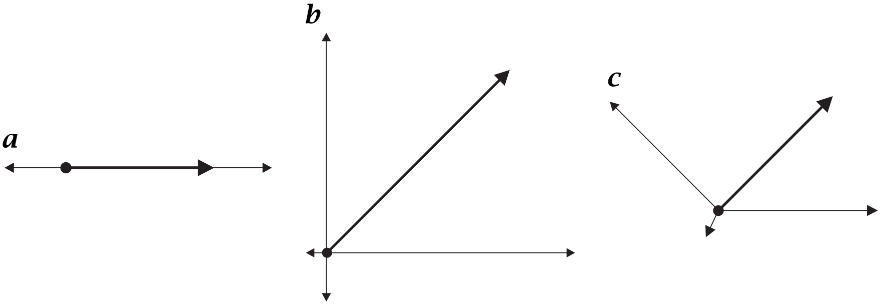

Figure 13.15: *Vector addition of the filter responses from K directional filters to determine local orientation;* ***a*** *$K = 2$;* ***b*** *$K = 3$;* ***c*** *$K = 4$; sum vector shown thicker.*

In other words, a preferred direction does not exist. The sum of all filter responses gives an *orientation invariant* response. This is also the deeper reason why we can determine local orientation exactly with a very limited number of filters and a simple linear procedure such as vector addition.

13.5.5 *Determination of the Local Wave Number*

The *lognormal* form of the radial part of the quadrature filter sets is the key for a direct estimate of the *local wave number* of a narrowband signal. According to Eq. (13.70), we can write the radial part of the transfer function of the quadrature filter sets as

$$\hat{r}_l(k) = \exp\left[-\frac{(\ln k - \ln k_l)^2}{2\sigma^2 \ln 2}\right]. \tag{13.79}$$

We examine the ratio of the output of two different radial center frequencies k_1 and k_2 and obtain:

$$\begin{aligned}
\frac{\hat{r}_2}{\hat{r}_1} &= \exp\left[-\frac{(\ln k - \ln k_2)^2 - (\ln k - \ln k_1)^2}{2\sigma^2 \ln 2}\right] \\
&= \exp\left[\frac{2(\ln k_2 - \ln k_1)\ln k + \ln^2 k_2 - \ln^2 k_1}{2\sigma^2 \ln 2}\right] \\
&= \exp\left[\frac{(\ln k_2 - \ln k_1)[\ln k - 1/2(\ln k_2 + \ln k_1)]}{\sigma^2 \ln 2}\right] \\
&= \exp\left[\frac{\ln(k/\sqrt{k_2 k_1})\ln(k_2/k_1)}{\sigma^2 \ln 2}\right] \\
&= \left(\frac{k}{\sqrt{k_1 k_2}}\right)^{\ln(k_2/k_1)/(\sigma^2 \ln 2)}
\end{aligned}$$

Generally, the ratio of two different radial filters is directly related to the local wave number. The relation becomes particularly simple if the exponent in the last expression is one. This is the case, for example, if the wave number ratio of the two filters is two ($k_2/k_1 = 2$ and $\sigma = 1$). Then

$$\frac{\hat{r}_2}{\hat{r}_1} = \frac{k}{\sqrt{k_1 k_2}}. \tag{13.80}$$

13.5.6 *Determination of the Structure Tensor*

In this final section we relate the quadrature filter set technique as discussed in Section 13.5 to the tensor technique (Section 13.3). It is shown that the structure tensor can be computed from the responses of these filters. Granlund and Knutsson [64] present the general equation to compute the structure tensor from the quadrature filter responses:

$$\boldsymbol{J}(\boldsymbol{x}) = \sum_{k=0}^{K-1} \mathcal{Q}_k g(\boldsymbol{x}) \left(\alpha \bar{\boldsymbol{d}}_k \otimes \bar{\boldsymbol{d}}_k - \beta \boldsymbol{I} \right), \tag{13.81}$$

where $\mathcal{Q}_k g(\boldsymbol{x})$ is the (amplitude) output of the kth quadrature filter and $\boldsymbol{I}$ the identity matrix. In the two-dimensional case, $\alpha = 4/3$ and $\beta = 1/3$.

We demonstrate this relationship with the quadrature filter set with (the minimum number of) three filters. The three filters point at 0°, 60°, and 120°. Thus the unit direction vectors are:

$$\begin{array}{rcl} \bar{\boldsymbol{d}}_0 & = & [1, 0]^T \\ \bar{\boldsymbol{d}}_1 & = & [1/2, \sqrt{3}/2]^T \\ \bar{\boldsymbol{d}}_2 & = & [-1/2, \sqrt{3}/2]^T \end{array} . \tag{13.82}$$

With these values for $\bar{\boldsymbol{d}}_k$, Eq. (13.81) can be written as

$$\begin{array}{rcl} \boldsymbol{J}(\boldsymbol{x}) & = & \mathcal{Q}_0 g(\boldsymbol{x}) \left[\begin{array}{cc} 1 & 0 \\ 0 & \text{-}1/3 \end{array} \right] \\ & + & \mathcal{Q}_1 g(\boldsymbol{x}) \left[\begin{array}{cc} 0 & 1/\sqrt{3} \\ 1/\sqrt{3} & 2/3 \end{array} \right] \\ & + & \mathcal{Q}_2 g(\boldsymbol{x}) \left[\begin{array}{cc} 0 & \text{-}1/\sqrt{3} \\ \text{-}1/\sqrt{3} & 2/3 \end{array} \right]. \end{array} \tag{13.83}$$

The matrices give the contribution of the individual quadrature filters to the corresponding elements of the structure tensor. For an isotropically oriented pattern, the output from all quadrature filters is the same. If we set the output to $q(\boldsymbol{x})$, Eq. (13.83) results in the correct structure tensor for an isotropically oriented pattern:

$$\boldsymbol{J}(\boldsymbol{x}) = \left[\begin{array}{cc} q(\boldsymbol{x}) & 0 \\ 0 & q(\boldsymbol{x}) \end{array} \right]. \tag{13.84}$$

Conversely, for an oriented pattern, the response is $q(\boldsymbol{x}) \cos^2(\phi_0 - \phi_k)$ and we obtain

$$\boldsymbol{J}(\boldsymbol{x}) = q(\boldsymbol{x}) \left[\begin{array}{cc} \cos^2(\phi_0) & \sin(2\phi_0)/2 \\ \sin(2\phi_0)/2 & \sin^2(\phi_0) \end{array} \right]. \tag{13.85}$$

This is the correct form of the structure tensor for an ideally oriented structure in the direction ϕ_0. (This can be shown for instance by checking that the determinant of the matrix is zero and by computing the orientation angle according to Eq. (13.12).)

There is one subtle but important difference between the quadrature filter technique and the structure tensor technique. The quadrature filter technique does not require any averaging to compute the elements of the structure tensor. However, the averaging is an essential element of the direct method. Without averaging, the coherency measure (see Eq. (13.15) in Section 13.3.4) would always be one.

13.6 *Exercises*

Problem 13.1: Analysis of local orientation

Interactive demonstration of the analysis of local orientation using various first-order derivative filters (dip6ex13.01)

Problem 13.2: Local orientation and noise

Interactive demonstration of the influence of noise on local orientation (dip6ex13.02)

Problem 13.3: ***Orientation and direction**

Explain the difference between orientation and direction and give at least one example for a vectorial image processing operator that constitute either a directional vector or an orientation vector.

Problem 13.4: ****Averaging of the structure tensor**

1. Why is it required to average the components of the structure tensor over a certain neighborhood (Eqs. (13.8) and (13.17))? Or asked the other way round: which information would the structure tensor deliver without averaging?
2. Do you know any tensorial image processing operators that require no averaging?

Problem 13.5: ****Analysis of local orientation with superimposing patterns**

In Section 13.3 we discussed in detail that with an ideally oriented structure the structure tensor is only of rank one. It is easy to compute the orientation vector (amplitude and angle of the structure), and the coherency is one. How does the structure tensor look like, if two ideally oriented structures with different directions superimpose? Without limitation of generality you can assume that the two structures are oriented with an angle $\pm\theta/2$ to the x axis. Let the amplitude be different. You can assume a sinusoidal signal.

1. Which orientation angle is computed by the structure tensor?
2. Which value has the coherency?
3. Analyze the results!

Problem 13.6: Hilbert filter

Interactive demonstration of various Hilbert filters (dip6ex13.03)

Problem 13.7: **Convolution mask for Hilbert filters

1. Which general conditions are required for a convolution mask that should be a Hilbert filter over a certain range of wave numbers?
2. Can an ideal Hilbert filter, i. e., a filter that has the ideal transfer function of a Hilbert filter for all wave numbers, be realized by a convolution mask with a a finite number of coefficients?
3. Is it possible to realize an ideal Hilbert filter with a recursive filter?

Problem 13.8: Local phase and wave number

Interactive demonstration of the determination of local phase and wave number using the Hilbert transform and quadrature filters (dip6ex13.04)

Problem 13.9: **Local amplitude, phase, and wave number

Local phase, amplitude, and wave numbers are features that are suitable to describe local properties of signals. Compute these three features for the following simple 1-D signals using the Hilbert transform

1. sine wave: $a_0 \sin kx$,
2. sine wave with harmonics: $a_0 \sin kx + a_1 \sin 2kx$ with $a_1 \ll a_0$, and
3. Superposition of two sine waves with equal amplitude and nearly equal wave numbers:
   $a \sin[(k_+ \Delta k/)2x] + a \sin[(k_- \Delta k/)2x]$ with $\Delta k \ll k$

Analyze the computed results!

Problem 13.10: Local phase and wave number with the Riesz transform

Interactive demonstration of the determination of local phase and wave number using the Riesz transform (dip6ex13.05)

Problem 13.11: **Simple 1-D quadrature filter

Is the simple filter pair

$$[-1 \quad 0 \quad 2 \quad 0 \quad -1]/4 \quad \text{and} \quad [1 \quad 0 \quad -1]/2$$

a useful quadrature filter pair?

1. Compute the transfer function of both filters.
2. Compute the phase difference between the two filters.
3. Compare the amplitudes of both transfer functions.

13.7 *Further Readings*

The quadrature filter approach (Section 13.5) is detailed in the monograph of Granlund and Knutsson [66], the inertia tensor method (Section 13.5.1) in a paper by Bigün and Granlund [10]. Poularikas [158] expounds the mathematics of the Hilbert transform. The extension of the analytical signal to higher-dimensional signals (Section 13.4.4) was published only recently by Felsberg and Sommer [46]. More mathematical background to the monogenic signal and geometric algebra for computer vision can be found in Sommer [195].