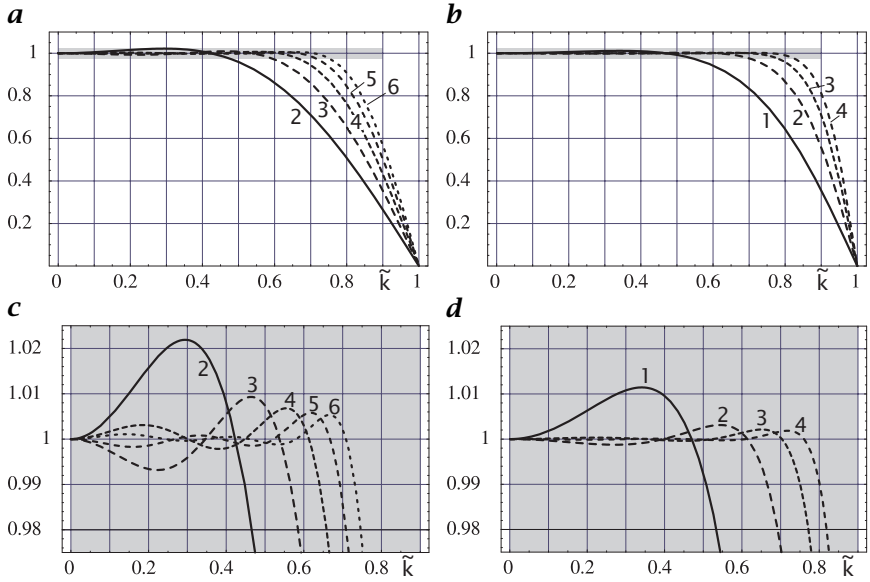

Figure 9.19: *Transfer function of interpolation kernels optimized with the weighted least squares technique of* ***a*** *Eq. (9.144) with R = 2 to 6 and* ***b*** *Eq. (9.145) with R = 1 to 4 (solid line).* ***c*** *and* ***d*** *show a narrow sector of the plots in* ***a*** *and* ***b*** *for a better estimation of small deviations from ideal values.*

9.7 Edge detection

Detection of edges is one of the most important tasks of low-level multidimensional signal processing. An edge marks the border of an object that is characterized by a different feature (gray value, color, or any other property) than the background. In the context of simple neighborhoods, an edge is a special type of simple neighborhood with a sharp transition. Low-level edge detection thus means to detect the strength of such a transition and the direction of the edge.

9.7.1 Edge detection by first-order derivatives

First-order derivative filters are one way for low-level edge detection. A first-order derivative operator corresponds to a multiplication by $2\pi i k_d$ in the wave-number space (Section 8.6.3). Thus, a first-order derivative operator in the direction d is represented by the following operations in the space and wave-number domain:

$$\frac{\partial}{\partial x_d} \;\Longleftrightarrow\; 2\pi i k_d \tag{9.146}$$

where $\tilde{k}$ is the dimensionless wave number normalized to the Nyquist limit Equation (8.34). One-dimensional first-order derivative operators are not sufficient for edge detection in higher-dimensional signals because they predominantly detect edges that are perpendicular to the direction of the operator. As shown with Eq. (9.186) in Section 9.8.1, the *gradient vector*

$$\nabla g = \left[\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, ..., \frac{\partial g}{\partial x_D} \right]^T \tag{9.147}$$

is parallel to the direction in which the gray values change. Thus it is a good low-level measure for edges. In the operator notation introduced in Section 9.1.3, the gradient can be written as a vector operator. In 2-D and 3-D space this is

$$\mathcal{D} = \begin{bmatrix} \mathcal{D}_x \\ \mathcal{D}_y \end{bmatrix} \quad \text{or} \quad \mathcal{D} = \begin{bmatrix} \mathcal{D}_x \\ \mathcal{D}_y \\ \mathcal{D}_z \end{bmatrix} \tag{9.148}$$

The magnitude of the gradient vector

$$|\nabla g| = \left(\sum_{d=1}^{D} \left(\frac{\partial g}{\partial x_d} \right)^2 \right)^{1/2} \tag{9.149}$$

is rotation-invariant and a measure for the edge strength. Because of the rotation invariance, this measure is isotropic. The computation of the magnitude of the gradient can be expressed in operator notation as

$$|\mathcal{D}| = \left[\sum_{d=1}^{D} \mathcal{D}_d \cdot \mathcal{D}_d \right]^{1/2} \tag{9.150}$$

The principal problem with all types of edge detectors is that a derivative operator can only be approximated on a discrete grid. This is one of the reasons why there is such a wide variety of solutions for edge detectors available.

General properties. With respect to object detection, the most important feature of a derivative convolution operator is that it must not shift the object position. For a first-order derivative filter, a real transfer function makes no sense, because extreme values should be mapped onto zero crossings and the steepest slopes to extreme values. This mapping implies a 90° phase shift, a purely imaginary transfer function and an antisymmetric or odd filter mask. According to the classification of *linear shift-invariant* (LSI) filters established in Section 9.2.5, first-order derivative filters are either type III or type IV filters. Thus

the simplified equations, Eqs. (9.25) and (9.27), can be used to compute the transfer function.

A derivative filter of any order must not show response to constant values or an offset in a signal. This condition implies that the sum of the coefficients must be zero and that the transfer function is zero for a zero wave number:

$$\sum_{\boldsymbol{n}} H_{\boldsymbol{n}} = 0 \Leftrightarrow \hat{H}(\boldsymbol{0}) = 0 \tag{9.151}$$

Intuitively, we expect that any derivative operator amplifies smaller scales more strongly than coarser scales, as the transfer function of a first-order derivative operator goes with k. However, this condition is too restrictive. Imagine that we first apply a smoothing operator to an image before we apply a derivative operator. Then the resulting transfer function would not increase monotonically with the wave number but decrease for higher wave numbers. We would, however, still recognize the joint operation as a derivation because the mean gray value is suppressed and the operator is only sensitive to spatial gray-value changes.

Thus a more general condition is required. Here we suggest

$$\hat{H}(\tilde{\boldsymbol{k}}) = \mathrm{i}\pi\tilde{k}_d\hat{B}(|\tilde{\boldsymbol{k}}|) \quad \text{with} \quad \hat{B}(0) = 1 \text{ and } \nabla\hat{B} = \boldsymbol{0} \tag{9.152}$$

This condition ensures that the transfer function is still zero for the wave number zero and increases in proportion to $\tilde{k}_d$ for small wave numbers. One can regard Eq. (9.152) as a first-order derivative filter regularized by an isotropic smoothing filter.

For good edge detection, it is important that the response of the operator does not depend on the direction of the edge. If this is the case, we speak of an isotropic edge detector. The isotropy of an edge detector can best be analyzed by its transfer function. Equation (9.152), which we derived from the condition of nonselective derivation, gives a general form for an isotropic first-order derivative operator.

First-order difference operators. This is the simplest of all approaches to compute a gradient vector. For the first partial derivative in the x direction, one of the following approximations may be used:

$$\begin{aligned}
\frac{\partial g(\boldsymbol{x})}{\partial x_d} &\approx \frac{g(\boldsymbol{x}) - g(\boldsymbol{x} - \Delta x_d\bar{\boldsymbol{e}}_d)}{\Delta x_d} && \text{backward} \\
&\approx \frac{g(\boldsymbol{x} + \Delta x_d\bar{\boldsymbol{e}}_d) - g(\boldsymbol{x})}{\Delta x_d} && \text{forward} \\
&\approx \frac{g(\boldsymbol{x} + \Delta x_d\bar{\boldsymbol{e}}_d) - g(\boldsymbol{x} - \Delta x_d\bar{\boldsymbol{e}}_d)}{2\Delta x_d} && \text{symmetric}
\end{aligned} \tag{9.153}$$

where $\bar{e}_d$ is a unit vector in the direction d. These approximations correspond to the filter masks

$$^{-}D_d = [1_\bullet \; -1], {}^{+}D_d = [1 \; -1_\bullet], D_{2d} = 1/2 \; [1 \; 0 \; -1] \tag{9.154}$$

The subscript "•" denotes the pixel of the asymmetric masks to which the result is written. The symmetric difference operator results in a type III operator (odd number of coefficients, odd symmetry, see Section 9.2.5). The forward and backward difference operators are asymmetric and thus not of much use in signal processing. They can be transformed in a type IV LSI operator if the result is not stored at the position of the right or left pixel but at a position half-way between the two pixels. This corresponds to a shift of the grid by half a pixel distance. The transfer function for the backward difference is then

$$^{-}\hat{D}_d = \exp(i\pi\tilde{k}_d/2)\left[1 - \exp(-i\pi\tilde{k}_d)\right] = i\sin(\pi\tilde{k}_d/2) \tag{9.155}$$

where the first term results from the shift by half a lattice point.

According to Eq. (9.25), the transfer function of the symmetric difference operator is given by

$$\hat{D}_{2d} = i\sin(\pi\tilde{k}_d) \tag{9.156}$$

This operator can also be computed from

$$D_{2d} = {}^{-}D_d B_d = [1_\bullet \; -1] * 1/2 \; [1 \; 1_\bullet] = 1/2\,[1 \; 0 \; -1]$$

Unfortunately, these simple difference filters are only poor approximations for an edge detector. From Eq. (9.156), we infer that the magnitude and direction of the gradient ϕ' are given by

$$|\nabla g| = \left[\sin^2(\pi\tilde{k}\cos\phi) + \sin^2(\pi\tilde{k}\sin\phi)\right]^{1/2} \tag{9.157}$$

and

$$\phi' = \arctan\frac{\sin(\pi\tilde{k}\sin\phi)}{\sin(\pi\tilde{k}\cos\phi)} \tag{9.158}$$

when the wave number is written in polar coordinates (k, ϕ). The magnitude of the gradient decreases quickly from the correct value. A Taylor expansion of Eq. (9.157) in $\tilde{k}$ yields for the anisotropy in the magnitude

$$\Delta|\nabla g| = |\nabla g(\phi)| - |\nabla g(0)| \approx \frac{(\pi\tilde{k})^3}{12}\sin^2(2\phi) \tag{9.159}$$

The resulting errors are shown in Fig. 9.20 as a function of the magnitude of the wave number and the angle to the x axis. The decrease

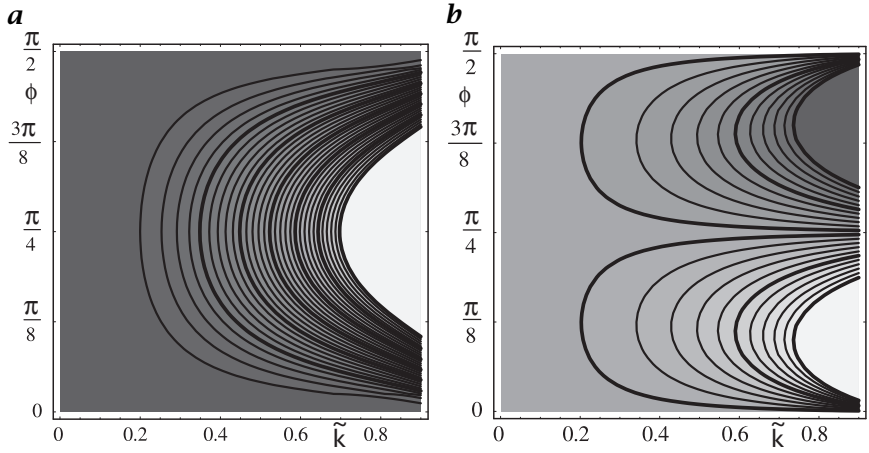

Figure 9.20: *Anisotropy of the* ***a*** *magnitude and* ***b*** *error in the direction of the gradient based on the symmetrical gradient operator* $[\mathcal{D}_{2x}, \mathcal{D}_{2y}]^T$. *The parameters are the magnitude of the wave number (0 to 0.9) and the angle to the* x *axis (0 to* $\pi/2$*). Distance of contour lines:* ***a*** *0.02 (thick lines 0.1);* ***b*** *2°.*

is also anisotropic; it is slower in the diagonal direction. The errors in the direction of the gradient are also large (Fig. 9.20b). While in the direction of the axes and diagonals the error is zero, in the directions inbetween it reaches values of about ± 10° already at $\tilde{k} = 0.5$. A Taylor expansion of Eq. (9.158) in $\tilde{k}$ gives in the approximation of small $\tilde{k}$ the angle error

$$\Delta\phi \approx \frac{(\pi\tilde{k})^2}{24} \sin 4\phi \tag{9.160}$$

From this equation, we see that the angle error is zero for $\phi = n\pi/4$ with $n \in \mathbb{Z}$, that is, for $\phi = 0°, 45°\ 90°, \ldots$.

Regularized difference operators. It is a common practice to regularize derivative operators by presmoothing the signal (see, e. g., Chapter 12). We will investigate here to what extent the direction and isotropy of the gradient is improved.

One type of regularized derivative filter is the derivate of a Gaussian. On a discrete lattice this operator is best approximated by the derivative of a binomial mask (Section 9.5.4) as

$$^{(B,R)}\mathcal{D}_d = \mathcal{D}_{2d}\mathcal{B}^R \tag{9.161}$$

with the transfer function

$$^{(B,R)}\hat{D}_d(\tilde{k}) = \mathrm{i}\sin(\pi\tilde{k}_d)\prod_{d=1}^{D}\cos^R(\pi\tilde{k}_d/2) \tag{9.162}$$

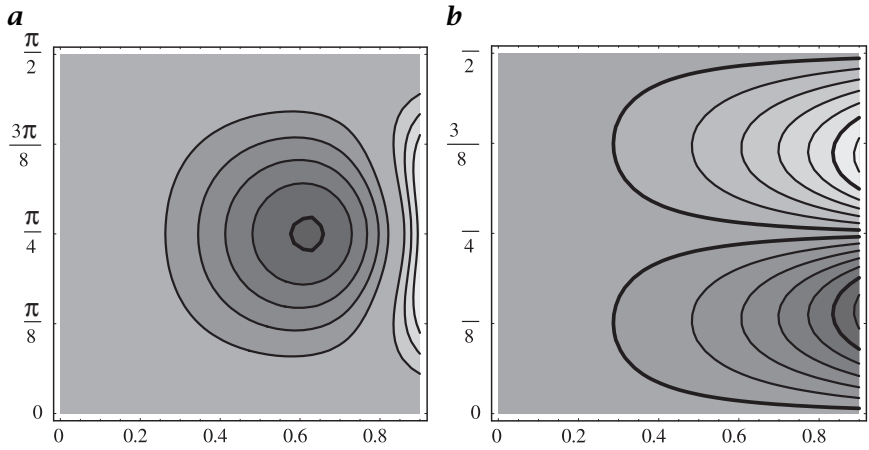

Figure 9.21: *Anisotropy of the* ***a*** *magnitude and* ***b*** *error in the direction of the gradient based on the Sobel operator Equation (9.166). Distance of contour lines as in Fig. 9.20.*

for even R. This approach leads to nonsquare masks and results in some improvement of the isotropy of the gradient magnitude. However, the error in the direction of the gradient is the same as for the symmetric difference operator because the smoothing terms in Eq. (9.162) cancel out in Eq. (9.158).

Slightly better are Sobel-type difference operators

$$^R S_d = \mathcal{D}_{2d} \mathcal{B}_d^{R-1} \prod_{d' \neq d} \mathcal{B}_{d'}^R \tag{9.163}$$

with the transfer function

$$^R \hat{S}_d(\tilde{k}) = \mathrm{i} \tan(\pi \tilde{k}_d/2) \prod_{d=1}^{D} \cos^R(\pi \tilde{k}_d/2) \tag{9.164}$$

that lead to square masks by reducing the smoothing in the direction of the derivation. The smallest operator of this type ($R = 1$) has in two dimensions the masks

$$^1\boldsymbol{S}_x = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad ^1\boldsymbol{S}_y = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \tag{9.165}$$

The best known example of this class of filters is the *Sobel operator*

$$^2\boldsymbol{S}_x = \boldsymbol{D}_x \boldsymbol{B}_x \boldsymbol{B}_y^2 = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad ^2\boldsymbol{S}_y = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{9.166}$$

The errors in the magnitude and direction of the gradient based on Eq. (9.164) are given by

$$\Delta|\nabla g| \approx -\frac{(\pi\tilde{k})^3}{24}\sin^2(2\phi) \tag{9.167}$$

and

$$\Delta\phi = \arctan\frac{\tan(\pi(\tilde{k}_d/2)\sin\phi)}{\tan(\pi(\tilde{k}_d/2)\cos\phi)} - \phi \approx -\frac{(\pi\tilde{k}_d)^2}{48}\sin 4\phi \tag{9.168}$$

and shown in Fig. 9.21. The results are remarkable in two respects. First, the error in the direction does not depend at all on the degree of smoothing as for the derivatives of Gaussians and is only about two times lower than that for the simple symmetric difference operator. Second, Fig. 9.21b shows that the anisotropy of the magnitude of the gradient is surprisingly low as compared to the symmetric difference filter in Fig. 9.20b. This could not be expected from the Taylor expansion because the term with $\tilde{k}^2$ is only a factor of two lower than that for the symmetric difference operator in Eq. (9.160). Thus the extrapolation of the transfer functions from small wave numbers to high wave numbers is not valid. The example of the Sobel operator shows that oscillating higher-order terms may cancel each other and lead to much better results as could be expected from a Taylor expansion.

The disadvantage of all approaches discussed so far is that they give no clear indication whether the achieved solution is good or whether any better exists. The filter design problem can be treated in a rigorously mathematical way as an optimization problem [CVA2, Chapter 6]. These techniques not only allow the design of optimal filters but they make it easier to decide precisely which criterion creates an optimal solution.

9.7.2 Edge detection by zero crossings

General properties. First-order derivative operators detect edges by maxima in the magnitude of the gradient. Alternatively, edges can be detected as zero crossings of second-order derivative operators. This technique is attractive because only *linear* operators are required to perform an isotropic detection of edges by zero crossings. In contrast, the magnitude of the gradient is only obtained after squaring and adding first-order derivative operators in all directions.

For an isotropic zero-crossing detector, only all second-order partial derivatives must be added up. The resulting operator is called the *Laplace operator* and denoted by Δ

$$\Delta = \sum_{d=1}^{D}\frac{\partial^2}{\partial x_w^2} \quad\Longleftrightarrow\quad -\sum_{d=1}^{D}4\pi^2 k_d^2 = -4\pi^2|\boldsymbol{k}|^2 \tag{9.169}$$

From this equation it is immediately evident that the Laplace operator is an isotropic operator.

A second-order derivative filter detects curvature. Extremes in function values should thus coincide with extremes in curvature. Consequently, a second-order derivative filter should be of even symmetry similar to a smoothing filter and all the properties for filters of even symmetry discussed in Sections 9.2.5 and 9.5.1 should also apply to second-order derivative filters. In addition, the sum of the coefficients must be zero as for first-order derivative filters:

$$\sum_{\boldsymbol{n}} H_{\boldsymbol{n}} = 0 \quad \Longleftrightarrow \quad \hat{H}(\mathbf{0}) = 0 \tag{9.170}$$

Also, a second-order derivative filter should not respond to a constant slope. This condition implies no further constraints as it is equivalent to the conditions that the sum of the coefficients is zero and that the filter is of even symmetry.

Laplace of Gaussian. The standard implementations for the Laplace operator are well known and described in many textbooks (see, e. g., [1]). Thus, we will discuss here only the question of an optimal implementation of the Laplacian operator. Because of a transfer function proportional to $\tilde{k}^2$ (Eq. (9.169)), Laplace filters tend to enhance the noise level in images considerably. Thus, a better edge detector may be found by first smoothing the image and then applying the Laplacian filter. This leads to a kind of regularized edge detection and to two classes of filters known as *Laplace of Gaussian* or LoG filters and *difference of Gaussian* or DoG filters. While these filters reduce the noise level it is not clear to what extent they improve or even optimize the isotropy of the Laplace operator.

In the discrete case, a LoG filter is approximated by first smoothing the image with a binomial mask and then applying the discrete Laplace filter. Thus we have the operator combination $\mathcal{L}\mathcal{B}^R$ with the transfer function

$$\hat{\text{LoG}} = \hat{L}\hat{B}^R = -4\sum_{d=1}^{D} \sin^2(\pi\tilde{k}_d/2) \prod_{d=1}^{D} \cos^R(\pi\tilde{k}_d/2) \tag{9.171}$$

For $R = 0$ this is the transfer function of the Laplace operator. In this equation, we used the standard implementation of the Laplace operator, which has in two dimensions the mask

$$\boldsymbol{L} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{9.172}$$

and the transfer function

$$\hat{L} = \sin^2(\pi\tilde{k}_1/2) + \sin^2(\pi\tilde{k}_2/2) \tag{9.173}$$

For small wave numbers, the 2-D transfer function in Eq. (9.171) can be approximated in polar coordinates by

$$\hat{\text{LoG}}(\tilde{k},\phi) \approx -(\pi\tilde{k})^2 + \left[\frac{1}{16} + \frac{R}{8} + \frac{1}{48}\cos(4\phi)\right](\pi\tilde{k})^4 \tag{9.174}$$

Difference of Gaussian filters. The multidimensional *difference of Gaussian* type of Laplace filter, or *DoG* filter, is defined as

$$\text{DoG} = 4(\mathcal{B}^2 - \mathcal{I})\mathcal{B}^R = 4(\mathcal{B}^{R+2} - \mathcal{B}^R) \tag{9.175}$$

and has the transfer function

$$\hat{\text{DoG}}(\tilde{\boldsymbol{k}}) = 4\prod_{d=1}^{D}\cos^{R+2}(\pi\tilde{k}_d/2) - 4\prod_{d=1}^{D}\cos^{R}(\pi\tilde{k}_d/2) \tag{9.176}$$

For small wave numbers it can be approximated by

$$\hat{\text{DoG}}(\tilde{k},\phi) \approx -(\pi\tilde{k})^2 + \left[\frac{3}{32} + \frac{R}{8} - \frac{1}{96}\cos(4\phi)\right](\pi\tilde{k})^4 \tag{9.177}$$

The transfer function of the LoG and DoG filters are quite similar. Both have a significant anisotropic term. Increased smoothing (larger R) does not help to decrease the anisotropy. It is obvious that the DoG filter is significantly more isotropic but neither of them is really optimal with respect to a minimal anisotropy. That second-order derivative operators with better isotropy are possible is immediately evident by comparing Eqs. (9.174) and (9.177). The anisotropic $\cos 4\phi$ terms have different signs. Thus they can easily be compensated by a mix of LoG and DoG operators of the form $2/3\text{DoG} + 1/3\text{LoG}$, which corresponds to the operator $(8/3\mathcal{B}^2 - 8/3\mathcal{I} - 1/3\mathcal{L})\mathcal{B}^p$.

This *ad hoc* solution is certainly not the best. Examples of optimized second-order differential operators are discussed in CVA2 [Chapter 6].

9.7.3 Edges in multichannel images

In multichannel images, it is significantly more difficult to analyze edges than to perform averaging, which simply can be performed channel by channel. The problem is that the different channels may contain conflicting information about edges. In channel A, the gradient can point to a different direction than in channel B. Thus a simple addition of the gradients in all channels

$$\sum_{p=1}^{P}\nabla g_p(\boldsymbol{x}) \tag{9.178}$$

is of no use. It may happen that the sum of the gradients over all channels is zero although the gradients themselves are not zero. Then we would be unable to distinguish this case from constant areas in all channels.

A more suitable measure of the total edge strength is the sum of the squared magnitudes of gradients in all channels

$$\sum_{p=1}^{P} |\nabla g_p|^2 = \sum_{p=1}^{P} \sum_{d=1}^{D} \left(\frac{\partial g_p}{\partial x_d} \right)^2 \tag{9.179}$$

While this expression gives a useful estimate of the overall edge strength, it still does not solve the problem of conflicting edge directions. An analysis of how edges are distributed in a D-dimensional multichannel image with P channels is possible with the following symmetric $\mathrm{D} \times \mathrm{D}$ matrix $\boldsymbol{S}$ (where D is the dimension of the image):

$$\boldsymbol{S} = \boldsymbol{J}^T \boldsymbol{J} \tag{9.180}$$

where $\boldsymbol{J}$ is known as the *Jacobian matrix*. This $\mathrm{P} \times \mathrm{D}$ matrix is defined as

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_D} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_D} \\ \vdots & & \ddots & \vdots \\ \frac{\partial g_P}{\partial x_1} & \frac{\partial g_P}{\partial x_2} & \cdots & \frac{\partial g_P}{\partial x_D} \end{bmatrix} \tag{9.181}$$

Thus the elements of the matrix $\boldsymbol{S}$ are

$$S_{kl} = \sum_{p=1}^{P} \frac{\partial g_p}{\partial x_k} \frac{\partial g_p}{\partial x_l} \tag{9.182}$$

Because $\boldsymbol{S}$ is a symmetric matrix, we can diagonalize it by a suitable coordinate transform. Then, the diagonals contain terms of the form

$$\sum_{p=1}^{P} \left(\frac{\partial g_p}{\partial x'_d} \right)^2 \tag{9.183}$$

In the case of an ideal edge, only one of the diagonal terms of the matrix will be nonzero. This is the direction perpendicular to the discontinuity. In all other directions it will be zero. Thus, $\boldsymbol{S}$ is a matrix of rank one in this case.

By contrast, if the edges in the different channels point randomly in all directions, all diagonal terms will be nonzero and equal. In this way, it is possible to distinguish random changes by noise from coherent edges. The trace of the matrix $\boldsymbol{S}$

$$\text{trace}(\boldsymbol{S}) = \sum_{d=1}^{D} S_{dd} = \sum_{d=1}^{D} \sum_{p=1}^{P} \left(\frac{\partial g_p}{\partial x_d} \right)^2 \tag{9.184}$$

gives a measure of the edge strength that we have already defined in Eq. (9.179). It is independent of the orientation of the edge because the trace of a symmetric matrix is invariant to a rotation of the coordinate system.

In conclusion, the matrix $\boldsymbol{S}$ is the key for edge detection in multichannel signals. Note that an arbitrary number of channels can be processed and that the number of computations increases only linearly with the number of channels. The analysis is, however, of order $O(D^2)$ in the dimension of the signal.

9.8 Tensor representation of simple neighborhoods

9.8.1 Simple neighborhoods

The mathematical description of a local neighborhood by continuous functions 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 because 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 simple local neighborhood is characterized by the fact that the gray value only changes in one direction. In all other directions it is constant. Because the gray values are constant along lines and form oriented structures this property of a neighborhood is denoted as *local orientation* [13] or *linear symmetry* [14]. Only more recently, the term *simple neighborhood* has been coined by Granlund and Knutsson [9].

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 $\bar{\boldsymbol{r}}$ perpendicular to the lines of constant gray values. Then, a simple neighborhood is mathematically represented by

$$g(\boldsymbol{x}) = g(\boldsymbol{x}^T \bar{\boldsymbol{r}}) \tag{9.185}$$