combinations of linear filter operators and point operators are very attractive as they can be composed of very simple and elementary operations that are very well understood and for which analytic expressions are available. Thus, these operations in contrast to many others can be the subject of a detailed mathematical analysis. Many advanced signal and image-processing techniques are of that type. This includes operators to compute local structure in images and various operations for texture analysis.

9.5 Local averaging

Averaging is an elementary neighborhood operation for multidimensional signal processing. Averaging results in better feature estimates by including more data points. It is also an essential tool to regularize otherwise ill-defined quantities such as derivatives (Chapters 10 and 12). Convolution provides the framework for all elementary averaging filters. In this chapter averaging filters are considered for continuous signals and for discrete signals on square, rectangular and hexagonal lattices. The discussion is not restricted to 2-D signals. Whenever it is possible, the equations and filters are given for signals with arbitrary dimension.

The common properties and characteristics of all averaging filters are discussed in Section 9.5.1. On lattices two types of averaging filters are possible [3, Section 5.7.3]. Type I filters generate an output on the same lattice. On a rectangular grid such filters are of odd length in all directions. Type II filters generate an output on a grid with lattice points between the original lattice points (intermediate lattice). On a rectangular grid such filters are of even length in all directions. In this chapter two elementary averaging filters for digital multidimensional signals are discussed—box filters (Section 9.5.3) and binomial filters (Section 9.5.4). Then we will deal with techniques to cascade these elementary filters to large-scale averaging filters in Section 9.5.5, and filters with weighted signals (normalized convolution) in Section 9.5.6.

9.5.1 General properties

Transfer function. Any averaging filter operator must preserve the mean value. This condition means that the transfer function for zero wave number is 1 or, equivalently, that the sum of all coefficients of the mask is 1:

$$\hat{h}(\mathbf{0}) = 1 \iff \int\limits_{-\infty}^{\infty} h(\boldsymbol{x})\,\mathrm{d}^D x = 1 \quad \text{or} \quad \sum_{\boldsymbol{n}\in\text{mask}} H_{\boldsymbol{n}} = 1 \qquad (9.68)$$

Intuitively, we expect that any smoothing operator attenuates smaller scales more strongly than coarser scales. More specifically, a smoothing operator should not completely annul a certain scale while smaller scales still remain in the image. Mathematically speaking, this means that the transfer function decreases monotonically with the wave number. Then for any direction, represented by a unit vector $\bar{\boldsymbol{r}}$

$$\hat{h}(k_2\bar{\boldsymbol{r}}) \leq \hat{h}(k_1\bar{\boldsymbol{r}}) \quad \text{if} \quad k_2 > k_1 \tag{9.69}$$

We may impose the more stringent condition that the transfer function approaches zero in all directions,

$$\lim_{k\to\infty} \hat{h}(k\bar{\boldsymbol{r}}) = 0 \tag{9.70}$$

On a discrete lattice the wave numbers are limited by the *Nyquist* condition, that is, the wave number must lay within the first Brillouin zone (Section 8.4.2). Then it makes sense to demand that the transfer function of an averaging filter is zero at the border of the Brillouin zone. On a rectangular lattice this means

$$\hat{h}(\boldsymbol{k}) = 0 \quad \text{if} \quad \boldsymbol{k}\hat{\boldsymbol{b}}_d = |\hat{\boldsymbol{b}}_d|/2 \tag{9.71}$$

where $\hat{\boldsymbol{b}}_d$ is any of the D-basis vectors of the reciprocal lattice (Section 8.4.2). Together with the monotonicity condition and the preservation of the mean value, this means that the transfer function decreases monotonically from one to zero for each averaging operator.

For a 1-D filter we can easily use Eq. (9.24) to relate the condition in Eq. (9.71) to a condition for the coefficients of type I filters:

$$\hat{h}(1) = 0 \Longleftrightarrow h_0 + 2\sum_{r\ \text{even}} h_r = 2\sum_{r\ \text{odd}} h_r \tag{9.72}$$

One-dimensional type II filters are, according to Eq. (9.24), always zero for $\tilde{k} = 1$.

Even filters in continuous space. With respect to object detection, the most important feature of an averaging operator is that it must not shift the object position. Any shift introduced by a preprocessing operator would cause errors in the estimates of the position and possibly other geometric features of an object. In order not to cause a spatial shift, a filter must not induce any phase shift in the Fourier space. A filter with this property is known as a *zero-phase filter*. This implies that the transfer function is real and this is equivalent with an *even symmetry* of the filter mask (Section 8.6.3):

$$h(-\boldsymbol{x}) = h(\boldsymbol{x}) \Longleftrightarrow \hat{h}(\boldsymbol{k}) \quad \text{real} \tag{9.73}$$

Averaging filters normally meet a stronger symmetry condition in the sense that each axis is a symmetry axis. Then Eq. (9.73) is valid for each component of $\boldsymbol{x}$:

$$h([x_1, \ldots, -x_d, \ldots, x_D]^T) = h([x_1, \ldots, x_d, \ldots, x_D]^T) \tag{9.74}$$

Even filters on 1-D lattices. For digital signals we distinguish filters with odd and even numbers of coefficients in *all* directions (Section 9.2.5). For both cases, we can write the symmetry condition for a filter with $R_d + 1$ coefficients in the direction d as

$$H_{r_0,r_1,\ldots,R_d-r_d,\ldots,r_D} = H_{r_0,r_1,\ldots,r_d,\ldots,r_D} \quad \forall d \in [1, D] \tag{9.75}$$

when we count the coefficients in each direction from left to right from 0 to R_d. This is not the usual counting but it is convenient as only one equation is required to express the evenness for filters with even and odd numbers of coefficients. For a 1-D filter the symmetry conditions reduce to

$$H_{R-r} = H_r \tag{9.76}$$

The symmetry relations significantly ease the computation of the transfer functions because for real transfer functions only the cosine term of the complex exponential from the Fourier transform remains in the equations (Sections 8.6 and 9.2.5). The transfer function for 1-D even masks with either $2R + 1$ (type I filter) or $2R$ coefficients (type II filter) is

$$\begin{aligned} {}^{I}\hat{h}(\tilde{k}) &= h_0 + 2\sum_{r=1}^{R} h_r \cos(r\pi\tilde{k}) \\ {}^{II}\hat{h}(\tilde{k}) &= 2\sum_{r=1}^{R} h_r \cos((r - 1/2)\pi\tilde{k}) \end{aligned} \tag{9.77}$$

Note that in these equations only pairs of coefficients are counted from 1 to R. The central coefficient of a filter with an odd number of coefficients has the index zero. As discussed in Section 9.2.5, filters with an odd number of coefficients output the filter results to the same lattice while filters with an even number of coefficients output the filter result to the intermediate lattice. A further discussion of the properties of symmetric filters up to three dimensions can be found in Jähne [4].

Isotropic filters. In most applications, the averaging should be the same in all directions in order not to prefer any direction. Thus, both the filter mask and the transfer function should be isotropic. Consequently, the filter mask depends only on the magnitude of the distance from the center pixel and the transfer function on the magnitude of the wave number:

$$H(\boldsymbol{x}) = H(|\boldsymbol{x}|) \iff \hat{H}(\tilde{\boldsymbol{k}}) = \hat{H}(|\tilde{\boldsymbol{k}}|) \tag{9.78}$$

This condition can also be met easily in discrete space. It means that the coefficients at lattice points with an equal distance from the center point are the same. However, the major difference now is that a filter whose coefficients meet this condition does not necessarily have an isotropic transfer function. The deviations from the isotropy are stronger the smaller is the filter mask. We will discuss the deviations from isotropy in detail for specific filters.

9.5.2 Separable averaging filters

The importance of separable filters for higher-dimensional signals is related to the fact that they can be computed much faster than nonseparable filters [CVA2, Section 5.6]. The symmetry conditions for separable averaging filters are also quite simple because only the symmetry condition Equation (9.76) must be considered. Likewise, the equations for the transfer functions of separable filters are quite simple. If we apply the same 1-D filter in all directions, the resulting transfer function of a D-dimensional filter is given after Eq. (9.77) by

$$\begin{aligned} {}^{I}\hat{h}(\tilde{\boldsymbol{k}}) &= \prod_{d=1}^{D}\left(h_0 + 2\sum_{r=1}^{R} h_r \cos(r\pi\tilde{k}_d)\right) \\ {}^{II}\hat{h}(\tilde{\boldsymbol{k}}) &= \prod_{d=1}^{D}\left(2\sum_{r=1}^{R} h_r \cos((r-1/2)\pi\tilde{k}_d)\right) \end{aligned} \tag{9.79}$$

With respect to isotropy, there exists only a single separable filter that is also isotropic, that is, the Gaussian function

$$\begin{aligned} \frac{1}{a^D}\exp(-\pi\boldsymbol{x}^T\boldsymbol{x}/a^2) &= \frac{1}{a^D}\prod_{d=1}^{D}\exp(-\pi x_d^2/a^2) \iff \\ \exp(-\pi a^2\tilde{\boldsymbol{k}}^T\tilde{\boldsymbol{k}}/4) &= \prod_{d=1}^{D}\exp(-\pi a^2\tilde{k}_d^2/4) \end{aligned} \tag{9.80}$$

This feature shows the central importance of the Gaussian function for signal processing from yet another perspective.

To a good approximation, the Gaussian function can be replaced on orthogonal discrete lattices by the binomial distribution. The coefficients of a 1-D binomal filter with $R+1$ coefficients and its transfer

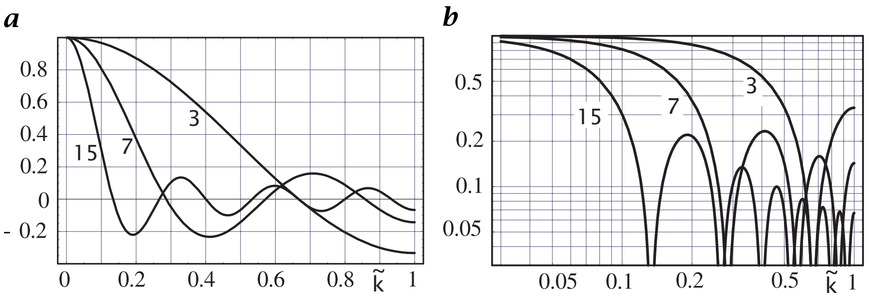

Figure 9.5: *Transfer functions of type I box filters with 3, 7, and 15 coefficients in* ***a*** *a linear plot; and* ***b*** *a log-log plot of the absolute value.*

function are given by

$$B^R = \frac{1}{2^R}\left[b_0 = 1, \ldots, b_r = \binom{R}{r}, \ldots, b_{R+1} = 1\right] \Longleftrightarrow \hat{B}^R(\tilde{k}) = \cos^R(\pi\tilde{k}/2) \tag{9.81}$$

With the comments on the isotropy of discrete filters in mind (Section 9.5.1), it is necessary to study the deviation of the transfer function of binomial filters from an isotropic filter.

9.5.3 Box filters

The simplest method is to average pixels within the filter mask and to divide the sum by the number of pixels. Such a simple filter is called a *box filter*. It is also known under the name *running mean*. In this section, only type I box filters are discussed. For type II box filters and box filters on hexagonal lattices see CVA2 [Section 7.3].

The simplest type I 1-D box filter is

$${}^3\boldsymbol{R} = \frac{1}{3}[1, 1, 1] \Longleftrightarrow {}^3\hat{R}(\tilde{k}) = \frac{1}{3} + \frac{2}{3}\cos(\pi\tilde{k}) \tag{9.82}$$

The factor 1/3 scales the result of the convolution sum in order to preserve the mean value (see Eq. (9.68) in Section 9.5.1). Generally, a type I 1-D box filter with $2R+1$ coefficients has the transfer function

$$\begin{aligned} {}^I\hat{R}(\tilde{k}) &= \frac{1}{2R+1} + \frac{2}{2R+1}\sum_{r=1}^{R}\cos(\pi r\tilde{k}) \\ &= \frac{1}{2R+1}\,\frac{\cos(\pi R\tilde{k}) - \cos(\pi(R+1)\tilde{k})}{1-\cos(\pi\tilde{k})} \end{aligned} \tag{9.83}$$

For small wave numbers the transfer function can be approximated by

$${}^I\hat{R}(\tilde{k}) \approx 1 - \frac{R(R+1)}{6}(\pi\tilde{k})^2 + \frac{R(R+1)(3R^2+3R-1)}{360}(\pi\tilde{k})^4 \tag{9.84}$$

Figure 9.5 shows that the box filter is a poor averaging filter. The transfer function is not monotonical and the envelope of the transfer function is only decreasing with k^{-1} (compare Section 8.6.3). The highest wave number is not completely suppressed even with large filter masks. The box filter also shows significant oscillations in the transfer function. The filter ${}^{2R+1}\mathcal{R}$ completely eliminates the wave numbers $\tilde{k} = 2r/(2R+1)$ for $1 \leq r \leq R$. In certain wave-number ranges, the transfer function becomes negative. This corresponds to a 180° phase shift and thus a contrast inversion.

Despite all their disadvantages, box filters have one significant advantage. They can be computed very fast with only one addition, subtraction, and multiplication independent of the size of the filter, that is, $O(R^0)$. Equation (9.83) indicates that the box filter can also be understood as a filter operation with a recursive part according to the following relation:

$$g'_n = g'_{n-1} + \frac{1}{2R+1}(g_{n+R} - g_{n-R-1}) \tag{9.85}$$

This recursion can easily be understood by comparing the computations for the convolution at neighboring pixels. When the box mask is moved one position to the right, it contains the same weighting factor for all pixels except for the last and the first pixel. Thus, we can simply take the result of the previous convolution (g'_{n-1}), subtract the first pixel that just moved out of the mask (g_{n-R-1}), and add the gray value at the pixel that just came into the mask (g_{n+R}). In this way, the computation of a box filter does not depend on its size.

Higher-dimensional box filters can simply be computed by cascading 1-D box filters running in all directions, as the box filter is separable. Thus the resulting transfer function for a D-dimensional filter is

$${}^{2R+1}\hat{R}(\tilde{k}) = \frac{1}{(2R+1)^D}\prod_{d=1}^{D}\frac{\cos(\pi R\tilde{k}_d) - \cos(\pi(R+1)\tilde{k}_d)}{1-\cos(\pi\tilde{k}_d)} \tag{9.86}$$

For a 2-D filter, we can approximate the transfer function for small wave numbers and express the result in cylinder coordinates by using $k_1 = k\cos\phi$ and $k_2 = k\sin\phi$ and obtain

$$\begin{aligned} {}^{I}\hat{R}(\tilde{\boldsymbol{k}}) \quad &\approx \quad 1 - \frac{R(R+1)}{6}(\pi\tilde{k})^2 + \frac{R(R+1)(14R^2+14R-1)}{1440}(\pi\tilde{k})^4 \\ &- \quad \frac{R(R+1)(2R^2+2R+1)}{1440}\cos(4\phi)(\pi\tilde{k})^4 \end{aligned} \tag{9.87}$$

This equation indicates that—although the term with $\tilde{k}^2$ is isotropic—the term with $\tilde{k}^4$ is significantly anisotropic. The anisotropy does not

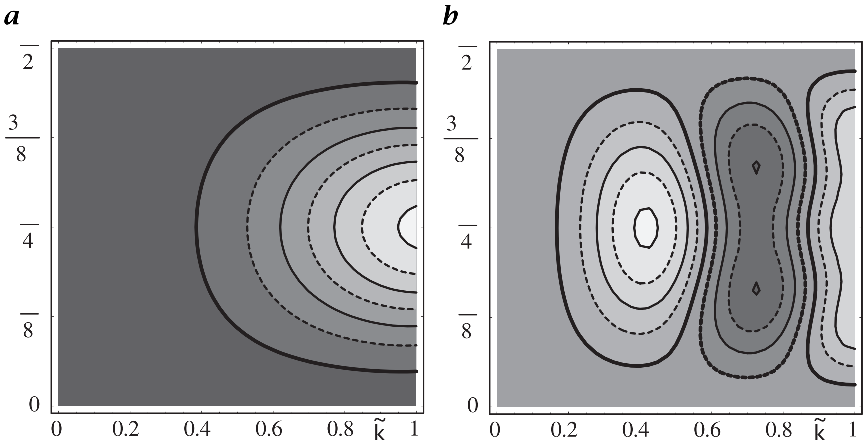

__Figure 9.6:__ Absolute deviation of the 2-D transfer functions of type I 2-D box filters from the transfer function along the x axis (1-D transfer function shown in Fig. 9.5) for a __a__ 3 × 3 and __b__ 7 × 7 filter. The distance of the contour lines is 0.05. The area between the thick contour lines marks the range around zero.

improve for larger filter masks because the isotropic and anisotropic terms in $\tilde{k}^4$ grow with the same power in R.

A useful measure for the anisotropy is the deviation of the 2-D filter response from the response in the direction of the x_1 axis:

$$\Delta\hat{R}(\tilde{\boldsymbol{k}}) = \hat{R}(\tilde{\boldsymbol{k}}) - \hat{R}(\tilde{k}_1) \tag{9.88}$$

For an isotropic filter, this deviation is zero. Again in an approximation for small wave numbers we obtain by Taylor expansion

$$\Delta^I\hat{R}(\tilde{\boldsymbol{k}}) \approx \frac{2R^4 + 4R^3 + 3R^2 + R}{720}\sin^2(2\phi)(\pi\tilde{k})^4 \tag{9.89}$$

The anisotropy for various box filters is shown in Fig. 9.6. Clearly, the anisotropy does not become weaker for larger box filters. The deviations are significant and easily reach 0.25. This figure means that the attenuation for a certain wave number varies up to 0.25 with the direction of the wave number.

9.5.4 Binomial filters

In Section 9.5.2 we concluded that only the Gaussian function meets the most desirable features of an averaging filter: separability and isotropy. In this section we will investigate to which extent the binomial filter, which is a discrete approximation to the Gaussian filter, still meets these criteria. The coefficients of the one-dimensional binomial filter

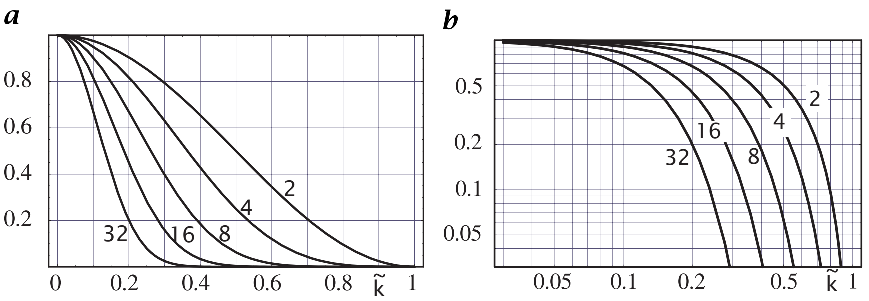

Figure 9.7: *Transfer functions of binomial filters $\mathcal{B}^R$ in **a** a linear plot and **b** a log-log plot of the absolute value with values of R as indicated.*

can be generated by repeatedly convolving the simple $1/2\,[1\ 1]$ mask:

$$\boldsymbol{B}^R = \underbrace{1/2\,[1\ 1] * \ldots * 1/2\,[1\ 1]}_{R \text{ times}} \tag{9.90}$$

This cascaded convolution is equivalent to the scheme in *Pascal's triangle*. The transfer function of the elementary $\boldsymbol{B} = 1/2\,[1\ 1]$ filter is

$$\hat{\boldsymbol{B}} = \cos(\pi\tilde{k}/2) \tag{9.91}$$

There is no need to distinguish type I and type II binomial filters in the equations because they can be generated by cascaded convolution as in Eq. (9.90). Therefore, the transfer function of the $\boldsymbol{B}^R$ binomial filter is

$$\hat{\boldsymbol{B}}^R = \cos^R(\pi\tilde{k}/2) \tag{9.92}$$

The most important features of binomial averaging filters are:

Monotonic transfer function. The transfer function decreases monotonically from 1 to 0 (Fig. 9.7).

Spatial variance. The coefficients of the binomial filter quickly approach with increasing mask size a sampled normal distribution. The spatial variance is

$$\sigma_x^2 = R/4 \tag{9.93}$$

A binomial filter effectively averages over a width of $2\sigma_x$. In contrast to the box filters, the effective averaging width increases only with the square root of the filter length.

Variance. Also the transfer function of the binomial filter quickly approaches the Gaussian function with increasing mask size (Fig. 9.7a). It is instructive to compare the Taylor expansion of the Gaussian

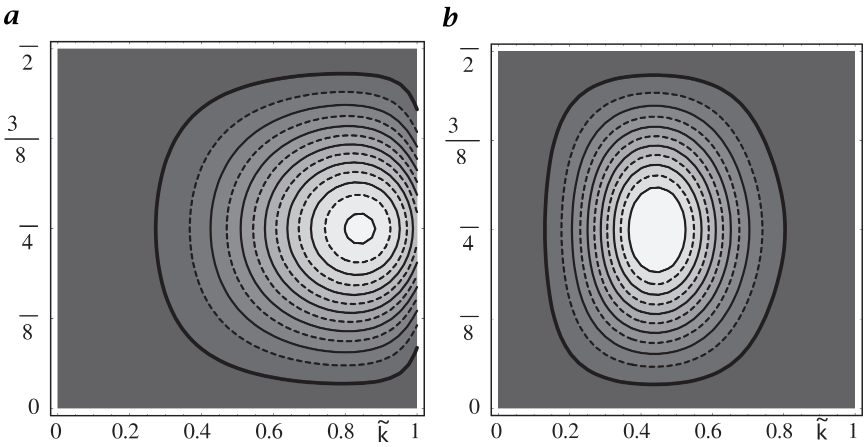

Figure 9.8: *Absolute deviation of the 2-D transfer functions of binomial filters from the transfer function along the x axis (1-D transfer function shown in Fig. 9.7) for a **a** 3×3 ($\mathcal{B}^2$) and **b** 9×9 ($\mathcal{B}^8$) filter. The distance of the contour lines is 0.005 in **a** and 0.001 in **b**. The area between the thick contour lines marks the range around zero.*

function for small wave numbers with those of the transfer functions of binomial filters:

$$\begin{array}{lcl}\exp(-\tilde{k}^2/(2\sigma_k^2)) & \approx & 1 - \dfrac{1}{2\sigma_k^2}\tilde{k}^2 + \dfrac{1}{8\sigma_k^4}\tilde{k}^4 \\ \hat{\boldsymbol{B}}^R(\tilde{k}) & \approx & 1 - \dfrac{R\pi^2}{8}\tilde{k}^2 + \left(\dfrac{R^2\pi^4}{128} - \dfrac{R\pi^4}{192}\right)\tilde{k}^4\end{array} \tag{9.94}$$

For large R both expansions are the same with

$$\sigma_k = \frac{2}{\sqrt{R}\pi} \tag{9.95}$$

Higher-dimensional binomial filters can be composed from 1-D binomial filters in all directions:

$$\mathcal{B}^R = \prod_{d=1}^{D} \mathcal{B}_d^R \tag{9.96}$$

Thus the transfer function of the multidimensional binomial filter $\mathcal{B}^R$ with $(R+1)^D$ coefficients is given by

$$\hat{B}^R = \prod_{d=1}^{D} \cos^R(\pi\tilde{k}_d/2) \tag{9.97}$$

The isotropy of binomial filters can be studied by expanding Eq. (9.97) in a Taylor series using cylindrical coordinates $\tilde{\boldsymbol{k}} = [\tilde{k}, \phi]^T$:

$$\hat{B}^R \approx 1 - \frac{R}{8}(\pi\tilde{k})^2 + \frac{2R^2 - R}{256}(\pi\tilde{k})^4 - \frac{R\cos 4\phi}{768}(\pi\tilde{k})^4 \qquad (9.98)$$

Only the second-order term is isotropic. In contrast, the fourth-order term contains an anisotropic part, which increases the transfer function in the direction of the diagonals. A larger filter (larger R) is less anisotropic as the isotropic term with $\tilde{k}^4$ increases quadratically with R while the anisotropic term with $\tilde{k}^4\cos 4\theta$ increases only linearly with R. The anisotropy deviation according to Eq. (9.88) is given by

$$\Delta\hat{B}^R \approx \frac{R}{384}\sin^2(2\phi)(\pi\tilde{k})^4 + \frac{5R^2 - 4R}{15360}\sin^2(2\phi)(\pi\tilde{k})^6 \qquad (9.99)$$

and shown in Fig. 9.8.

Three-dimensional binomial filters and binomial filters on hexagonal grids are discussed in CVA2 [Section 7.4.3].

9.5.5 Cascaded averaging

The approaches discussed so far for local averaging are no solution if the averaging should cover large neighborhoods for the following reasons: First, binomial filters are not suitable for large-scale averaging—despite their efficient implementation by cascaded convolution with $\mathcal{B}$—because the averaging distance increases only with the square root of the mask size (see Eq. (9.93) in Section 9.5.4). Second, box filters and recursive filters are, in principle, suitable for large-scale averaging because the number of operations does not increase with the size of the point spread function (operation of the order $O(R^0)$). However, both types of filters have a nonideal transfer function. The transfer function of the box filter is not monotonically decreasing with the wave number (Section 9.5.3) and both filters show overly large deviations from an isotropic response. In this section, several techniques are discussed for large-scale averaging that overcome these deficits and limitations.

Multistep averaging. The problem of slow large-scale averaging originates from the small distance between the pixels averaged by small masks. In order to overcome this problem, we may use the same elementary averaging process but with more distant pixels. As the box, binomial and recursive averaging filters are separable and thus are applied as cascaded filter operations running one after the other in all coordinate directions through a multidimensional signal, it is sufficient to discuss increasing the step width for 1-D filter operations. A 1-D convolution with a mask that operates only with every S-th pixel can

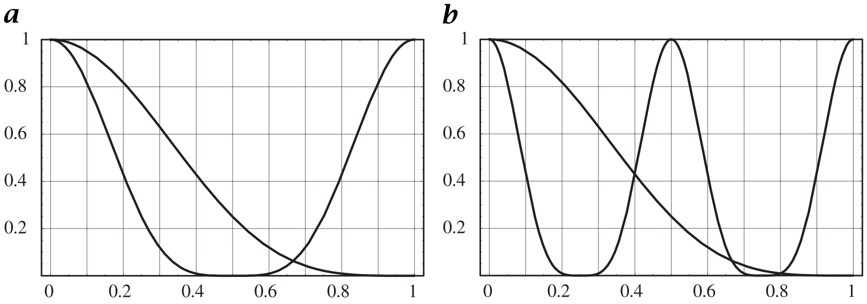

Figure 9.9: *Transfer functions of the binomial filter $\mathcal{B}^4$ ($\boldsymbol{B} = 1/16[1\ 4\ 6\ 4\ 1]$) and the same filter stretched by **a** a factor of two, $\mathcal{B}_2^4$ ($\boldsymbol{B}_2 = 1/16[1\ 0\ 4\ 0\ 6\ 0\ 4\ 0\ 1]$), and **b** a factor of four, $\mathcal{B}_4^4$.*

be written as a stretched mask

$$(h_S)_n = \begin{cases} h_{n'} & n = Sn' \\ 0 & \text{else} \end{cases} \iff \hat{h}_S(\tilde{k}) = \hat{h}(\tilde{k}/S) \tag{9.100}$$

Because of the reciprocity between the spatial and Fourier domains the stretching of the filter mask by a factor S results in a corresponding shrinking of the transfer function. This shrinking goes—because of the periodicity of the transfer function of discrete samples—along with an S-fold replication of the transfer function as illustrated in Fig. 9.9.

An averaging filter that is used with a larger step width is no longer a good averaging filter for the whole wave-number range but only for wave numbers up to $\tilde{k} = 1/S$. Used individually, these filters are thus not of much help. But we can use them in cascade in such a way that previous smoothing has already removed all wave numbers beyond $\tilde{k} = 1/S$. This is the basic principle for the design of cascaded filters.

For practical design there are two degrees of freedom. First, we can select the basic filter that is used repeatedly with different step widths. Here, box, binomial and relaxation filters are investigated. Second, we can choose the way in which the step width is increased. We will consider both a linear and an exponential increase in the step width. Generally, a cascaded filter operation consists of the following chain of P operations with the filter operation $\mathcal{B}$:

$$\underbrace{\mathcal{B}_{a_P} \ldots \mathcal{B}_{a_p} \ldots \mathcal{B}_{a_2}\mathcal{B}_{a_1}}_{P \text{ times}} \tag{9.101}$$

where a_p consists of a sequence of step widths. Whereas in each step the same operator $\mathcal{B}$ with the spatial variance σ^2 is used and only the

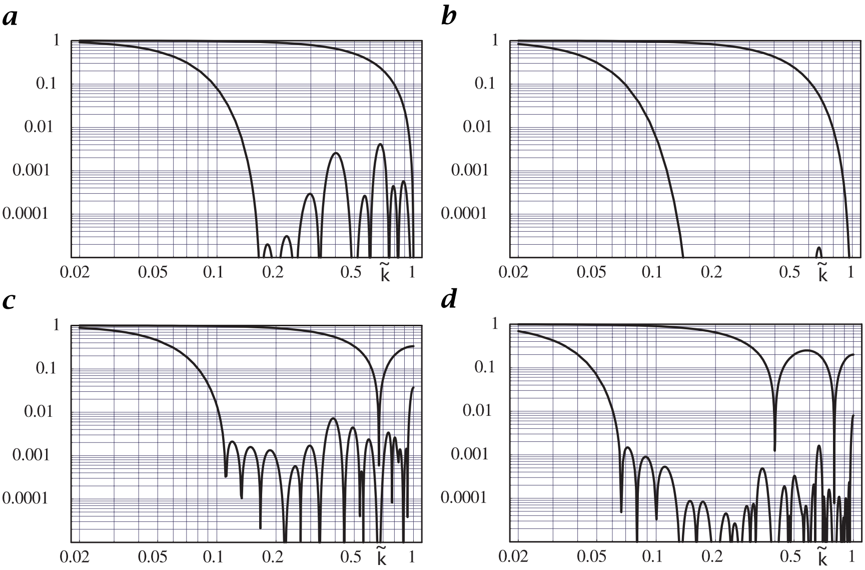

***Figure 9.10:** Transfer functions of cascaded filtering with linear increase in step width with **a** $\mathcal{B}^2$, **b** $\mathcal{B}^4$, **c** ${}^3\mathcal{R}$, and **d** ${}^5\mathcal{R}$. Shown are the transfer functions of the original filters and of the cascaded filtering up to the six-fold step size with a resulting averaging width $\sqrt{91} \approx 9.54$ times larger than the original filter.*

step width is changed, the resulting step width can be computed by

$$\sigma_c^2 = \sigma^2 \sum_{p=1}^{P} a_p^2 \qquad (9.102)$$

From this equation it is also obvious that efficient filter cascading requires an increasing step width. If we keep the step width constant, the averaging width given by σ_c increases only with $\sqrt{P}$ and not linearly with P.

Linearly increasing step width. In the simplest case, the step width is increased linearly, that is, $a_p = p$. This results in the following sequence of P step widths: $1, 2, 3, 4, \ldots, P$. According to Eq. (9.102), the resulting series of variances is

$$\sigma_c^2 = \sigma^2 \sum_{p=1}^{P} p^2 = \frac{P(P+1)(2P+1)}{6}\sigma^2 \qquad (9.103)$$

For large P, $\sigma_c = P^{3/2}\sigma/\sqrt{3}$. Thus the averaging width increases even stronger than linear with the number of steps. With only six steps, the resulting averaging width is $\sqrt{91} \approx 9.54$ times larger than that of the

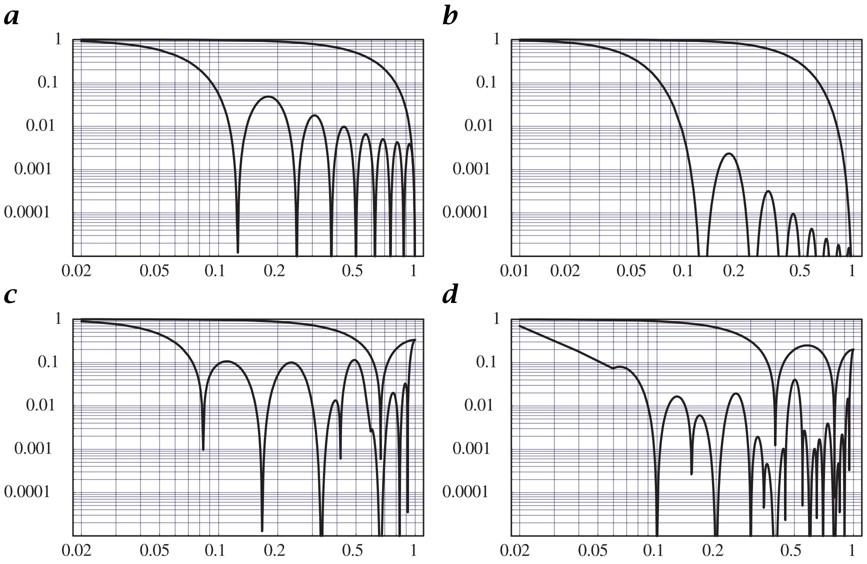

Figure 9.11: *Transfer functions of cascaded filtering with exponential increase in step width with* ***a*** $\mathcal{B}^2$, ***b*** $\mathcal{B}^4$, ***c*** ${}^3\mathcal{R}$, *and* ***d*** ${}^5\mathcal{R}$. *Shown are the transfer functions of the original filters and of four cascaded filters (up to step size 8) with a resulting averaging width* $\sqrt{85} \approx 9.22$ *times larger than the original filter.*

original filter (Fig. 9.10). To achieve this averaging width, the same filter would have to be applied 91 times.

The quality of the cascaded filtering, that is, the degree of deviation from a monotonic transfer function, is determined by the basic filter. Figure 9.10 shows the transfer functions for a number of different filters in a double-logarithmic plot. Only the binomial filter $\mathcal{B}^4$ shows negligible secondary peaks well beyond 10^{-4}. The other filters in Fig. 9.10 have significantly more pronounced secondary peaks in the 10^{-4} to 10^{-2} range.

Exponentially increasing step width. A linear increase in the step width is still too slow to achieve averaging over very large scales. It is also disadvantageous in that the increase in the averaging width is of the odd order $P^{3/2}$. This means that filtering does not increase the width of the averaging linearly. The increase is slightly stronger. Both difficulties are overcome with an exponential increase in the step width. The easiest way is to increase the step width by a factor of two from filtering to filtering. The resulting mask has the standard deviation

$$\sigma_c^2 = \sigma^2 \sum_{p=1}^{P} 2^{2p-2} = \frac{2^{2P}-1}{3}\sigma^2 \qquad (9.104)$$

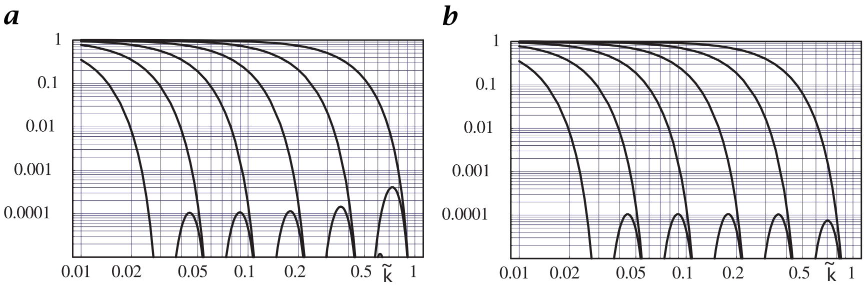

Figure 9.12: ***a*** *Sequence of transfer functions of cascaded filtering with exponential increase in step width using the $\mathcal{B}^6$ binomial filter.* ***b*** *Shows the same sequence except that the first filter with step width 1 is $\mathcal{B}^8$.*

Thus the standard deviation grows exponentially to $\approx (2^P/\sqrt{3})\sigma$ with only P filtering steps. In other words, the number of computations increases only logarithmically with the averaging width.

As for the linear increase of the step width, the basic filter determines the quality of the resulting transfer function of the cascaded filtering. Figure 9.11 shows that only the binomial filter $\mathcal{B}^4$ results in an acceptable transfer function of the cascaded filtering. All other filters show too high secondary peaks.

Figure 9.12a shows a sequence of transfer functions for the cascading of the binomial filter $\mathcal{B}^6$. It can be observed that the filters are not of exactly the same shape but that the secondary peak is higher for the first steps and only gradually levels off to a constant value. This effect is caused by the constant term in Eq. (9.104). It can be compensated if the first filter ($p = 1$) does not have variance σ^2 but has variance $4/3\sigma^2$. Indeed, if a $\mathcal{B}^8$ filter is used instead of the $\mathcal{B}^6$ filter in the first step, the filters in the different steps of the filter cascade are much more similar (Fig. 9.12b).

For higher-dimensional signals the isotropy of the averaging is of significance. As we already know that all filters except for the binomial filters are significantly anisotropic, only binomial filters are discussed. While the $\mathcal{B}^2$ filter still shows a pronounced anisotropy of several percent (Fig. 9.13a), the anisotropy is already just slightly more than 0.01 for a $\mathcal{B}^4$ filter (Fig. 9.13b).

Multigrid averaging. Multistep cascaded averaging can be further enhanced by converting it into a multiresolution technique. The idea of multigrid smoothing is very simple. If a larger step mask is involved, this operation can be applied on a correspondingly coarser grid. This means that the last operation before using the larger step mask needs to compute the convolution only at the grid points used by the following coarser grid operator. This sampling procedure is denoted by a special syntax in the operator index; $\mathcal{O}_{\downarrow 2}$ means: Apply the operator in

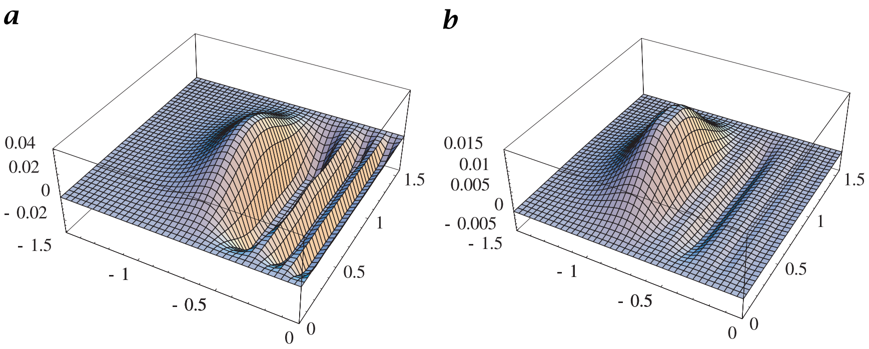

Figure 9.13: *Anisotropy of cascaded filtering with exponential increase of the step width in a log-polar plot. Shown is the deviation from the transfer function in the direction of the x axis for* **a** $\mathcal{B}_4^2\mathcal{B}_2^2\mathcal{B}_1^2$ *and* **b** $\mathcal{B}_4^4\mathcal{B}_2^4\mathcal{B}_1^4$.

all directions and advance the mask two pixels in all directions. Thus, the output of the filter operator has only half as many pixels in every direction as the input.

Multigrid smoothing makes the number of computations essentially independent of the standard deviation of the smoothing mask. We again consider a sequence of 1-D averaging filters:

$$\underbrace{\mathcal{B}_{\downarrow 2}\cdots\mathcal{B}_{\downarrow 2}\mathcal{B}_{\downarrow 2}}_{P\ \text{times}}$$

The standard deviation of the filter cascade is the same as for the multistep approach with exponential increase of the step width (Eq. (9.104)). Also, as long as the sampling condition is met, that is, $\hat{\boldsymbol{B}}^p(\tilde{k}) = 0 \ \forall \tilde{k} \geq 1/2$, the transfer functions of the filters are the same as for the multistep filters.

If $\mathcal{B}_{\downarrow 2}$ takes q operations, the operator sequence takes

$$q\sum_{p=1}^{P}\frac{1}{2^{p-1}} = 2q\left(1-\frac{1}{2^{P-1}}\right) < 2q \qquad (9.105)$$

Thus, smoothing to any degree takes no more than twice as many operations as smoothing at the first step.

9.5.6 Weighted averaging

Image data, just like any other experimental data, may be characterized by individual errors that have to be considered in any further processing. As an introduction, we first discuss the averaging of a set of N data g_n with standard deviations σ_n. From elementary statistics, it is known that appropriate averaging requires the weighting of each data point g_n with the inverse of the variance $w_n = 1/\sigma_n^2$. Then, an estimate of the mean value is given by

$$\langle g \rangle = \sum_{n=1}^{N} g_n/\sigma_n^2 \bigg/ \sum_{n=1}^{N} 1/\sigma_n^2 \tag{9.106}$$

while the standard deviation of the mean is

$$\sigma_{\langle g \rangle}^2 = 1 \bigg/ \sum_{n=1}^{N} 1/\sigma_n^2 \tag{9.107}$$

The application of *weighted averaging* to image processing is known as *normalized convolution* [9]. The averaging is now extended to a local neighborhood. Each pixel enters the convolution sum with a weighting factor associated with it. Thus, normalized convolution requires two signals. One is the image $\boldsymbol{G}$ to be processed, the other an image $\boldsymbol{W}$ with the weighting factors.

By analogy to Eqs. (9.106) and (9.107), normalized convolution with the mask $\boldsymbol{H}$ is defined as

$$\boldsymbol{G}' = \frac{\boldsymbol{H} * (\boldsymbol{W} \cdot \boldsymbol{G})}{\boldsymbol{H} * \boldsymbol{W}} \tag{9.108}$$

A normalized convolution with the mask $\boldsymbol{H}$ essentially transforms the image $\boldsymbol{G}$ and the weighting image $\boldsymbol{W}$ into a new image $\boldsymbol{G}'$ and a new weighting image $\boldsymbol{W}' = \boldsymbol{H} * \boldsymbol{W}$, which can undergo further processing.

Normalized convolution is just adequate consideration of pixels with spatially variable statistical errors. “Standard” convolution can be regarded as a special case of normalized convolution. Then all pixels are assigned the same weighting factor and it is not required to use a weighting image, because the factor remains a constant.

The flexibility of normalized convolution is given by the choice of the weighting image. The weighting image is not necessarily associated with an error. It can be used to select and/or amplify pixels with certain features. In this way, normalized convolution becomes a versatile nonlinear operator. The application of normalized convolution is discussed in a number of contributions in the application gallery: Sections A18, A20, A16, and A23.

9.6 Interpolation

Interpolation of digital signals is required for a wide range of signal-processing tasks whenever any operation shifts the digital points of the output signal so that they no longer coincide with the grid points of the input signal. This occurs, among others, with the following operations: