and we end up with the well-known binomial smoothing mask and the Sobel operator for the estimate of the mean and slopes of a local plane fit, respectively.

Thus, the close relationship between LSI operators and least squares fits is helpful in determining what kind of properties an LSI operator is filtering out from a signal.

The case with nonorthogonal fit functions is slightly more complex. As the matrix $\boldsymbol{M}$ (Eq. (9.32)) depends only on the fit functions and the chosen window and not on the signal $g(\boldsymbol{x})$, the matrix $\boldsymbol{M}$ can be inverted once for a given fit. Then the fit coefficients are given as a linear combination of the results from the convolutions with all P fit functions:

$$a_p(\boldsymbol{x}) = \sum_{p'=0}^{P-1} M_{p,p'}^{-1} \int_{-\infty}^{\infty} w(\boldsymbol{x}') f_{p'}(-\boldsymbol{x}') g(\boldsymbol{x} - \boldsymbol{x}')\, \mathrm{d}^D x' \qquad (9.40)$$

9.3 Recursive filters

9.3.1 Definition

Recursive filters are a special form of the linear convolution filters. This type of filter includes results from previous convolutions at neighboring pixels into the convolution sum. In this way, the filter becomes directional. Recursive filters can most easily be understood if we apply them first to a 1-D discrete signal, a *time series*. Then we can write

$$g'_n = -\sum_{n''=1}^{S} a_{n''} g'_{n-n''} + \sum_{n'=-R}^{R} h_{n'} g_{n-n'} \qquad (9.41)$$

While the neighborhood of the nonrecursive part (coefficients h) is symmetric around the central point, the recursive part is asymmetric, using only previously computed values. A filter that contains only such a recursive part is called a *causal filter*. If we put the recursive part on the left hand side of the equation, we observe that the recursive filter is equivalent to the following difference equation, also known as an *ARMA*(S,R) process (*autoregressive-moving average process*):

$$\sum_{n''=0}^{S} a_{n''} g'_{n-n''} = \sum_{n'=-R}^{R} h_{n'} g_{n-n'} \quad \text{with} \quad a_0 = 1 \qquad (9.42)$$

9.3.2 Transfer function and z-transform

The transfer function of such a filter with a recursive and a nonrecursive part can be computed by applying the *discrete-space Fourier transform*

(Section 8.5.1, Table 8.3) to Eq. (9.42). In the Fourier space the convolution of $\boldsymbol{g}'$ with $\boldsymbol{a}$ and of $\boldsymbol{g}$ with $\boldsymbol{h}$ is replaced by a multiplication of the corresponding Fourier transforms:

$$\hat{g}'(k)\sum_{n''=0}^{S} a_{n''}\exp(-2\pi \mathrm{i} n''k) = \hat{g}(k)\sum_{n'=-R}^{R} h_{n'}\exp(-2\pi \mathrm{i} n'k) \qquad (9.43)$$

Thus the transfer function is

$$\hat{h}(k) = \frac{\hat{g}'(k)}{\hat{g}(k)} = \frac{\sum\limits_{n'=-R}^{R} h_{n'}\exp(-2\pi \mathrm{i} n'k)}{\sum\limits_{n''=0}^{S} a_{n''}\exp(-2\pi \mathrm{i} n''k)} \qquad (9.44)$$

The nature of the transfer function of a recursive filter becomes more evident if we consider that both the numerator and the denominator can have zeros. Thus the nonrecursive part of the transfer function may cause zeros and the recursive part poles.

A deeper analysis of the zeros and thus the structure of the transfer function is not possible in the form as Eq. (9.44) is written. It requires an extension similar to the extension from real numbers to complex numbers that was necessary to introduce the Fourier transform (Section 8.5.3). We observe that the expressions for both the numerator and the denominator are polynomials in the *complex exponential* $\exp(2\pi \mathrm{i} k)$. The complex exponential has a magnitude of one and thus covers the unit circle in the complex plane. It covers the whole complex plane if we add a radius r to the expression: $z = r\exp(2\pi \mathrm{i} k)$.

With this extension, the expressions become polynomials in z. As such we can apply the fundamental law of algebra that any *complex polynomial* of degree n can be factorized in n factors containing the roots or zeros of the polynomial. Thus we can write a new expression in z, which becomes the transfer function for $z = \exp(2\pi \mathrm{i} k)$:

$$\hat{h}(z) = \frac{\prod\limits_{n'=-R}^{R}(1 - c_{n'}z^{-1})}{\prod\limits_{n''=0}^{S}(1 - d_{n''}z^{-1})} \qquad (9.45)$$

Each of the factors $c_{n'}$ and $d_{n''}$ is a zero of the corresponding polynomial ($z = c_{n'}$ or $z = d_{n''}$).

The inclusion of the factor r in the extended transfer function results in an extension of the Fourier transform, the *z-transform* that is defined as

$$\hat{g}(z) = \sum_{n=-\infty}^{\infty} g_n z^{-n} \qquad (9.46)$$

The z-transform of the series g_n can be regarded as the Fourier transform of the series $g_n r^{-n}$ [5]. The z-transform is the key mathematical tool to understand recursive filters. Detailed accounts of the z-transform are given by Oppenheim and Schafer [3] and Poularikas [6]; the 2-D z-transform is discussed by Lim [5].

The factorization of the z-transform of the filter in Eq. (9.45)—and in turn of the transfer function—is an essential property. Multiplication of the individual factors of the transfer function means that we can decompose any filter into elementary filters containing only one factor because multiplication of transfer functions is equivalent to cascaded convolution in the spatial domain (Section 8.6.3). The basic filters that are equivalent to a single factor in Eq. (9.45) will be discussed further in Section 9.3.6.

Recursive filters can also be defined in higher dimensions with the same type of equations as in Eq. (9.42); also the transfer function and z-transform of higher-dimensional recursive filters can be written in the very same way as in Eq. (9.44). However, it is generally *not* possible to factorize the z-transform as in Eq. (9.45) [5]. From Eq. (9.45) we can immediately conclude that it will be possible to factorize a *separable* recursive filter because then the higher-dimensional polynomials can be factorized into 1-D polynomials. Given these inherent difficulties of higher-dimensional recursive filters we will restrict the further discussion on 1-D recursive filters that can be extended by cascaded convolution into higher-dimensional filters.

9.3.3 Infinite and unstable response

The *impulse response* or *point spread function* of a recursive filter is no longer identical to the filter coefficients as for nonrecursive filters (Section 9.2.3). It must rather be computed as the inverse Fourier transform of the transfer function. The impulse response of nonrecursive filters has only a finite number of nonzero samples. A filter with this property is called a *finite-duration impulse response* or FIR filter. In contrast, recursive filters have an *infinite-duration impulse response* (IIR).

The *stability* of the filter response is not an issue for nonrecursive filters but of central importance for recursive filters. A filter is said to be *stable* if and only if *each* bound input sequence generates a bound output sequence. In terms of the impulse response this means that a filter is stable if and only if the impulse response is absolutely summable [3]. For 1-D filters the analysis of the stability is straightforward because the conditions are well established by the same basic algebraic theorems. A filter is stable and causal if and only if all poles and zeros of the z-transform $\hat{h}(z)$ (Eq. (9.45)) are inside the unit circle [3].

9.3.4 Relation between recursive and nonrecursive filters

Any stable recursive filter can be replaced by a nonrecursive filter, in general, with an infinite-sized mask. Its mask is given by the point spread function of the recursive filter. In practice, the masks cannot be infinite and also need not be infinite. This is due to the fact that the envelope of the impulse response of any recursive filter decays exponentially (Section 9.3.6).

Another observation is of importance. From Eq. (9.44) we see that the transfer function of a recursive filter is the ratio of its nonrecursive and recursive part. This means that a purely recursive and a nonrecursive filter with the same coefficients are inverse filters to each other. This general relation is a good base to construct inverse filters from nonrecursive filters.

9.3.5 Zero-phase recursive filtering

The causal 1-D recursive filters are of not much use for processing of higher-dimensional spatial data. While a filter that uses only previous data is natural and useful for real-time processing of time series, it makes not much sense for spatial data. There is no “before” and “after” in spatial data. Even worse, the spatial shift (delay) associated with recursive filters is not acceptable because it causes phase shifts and thus objects to be shifted depending on the filters applied.

With a single recursive filter it is impossible to construct a zero-phase filter. Thus it is required to combine multiple recursive filters. The combination should either result in a zero-phase filter suitable for smoothing operations or a derivative filter that shifts the phase by 90°. Thus the transfer function should either be purely real or purely imaginary (Section 8.6.3).

We start with a 1-D causal recursive filter that has the transfer function

$$ {}^{+}\hat{h}(\tilde{k}) = a(\tilde{k}) + \mathrm{i}b(\tilde{k}) \tag{9.47} $$

The superscript “+” denotes that the filter runs in positive coordinate direction. The transfer function of the same filter but running in the opposite direction has a similar transfer function. We replace $\tilde{k}$ by $-\tilde{k}$ and note that $a(-\tilde{k}) = a(+\tilde{k})$ and $b(-\tilde{k}) = -b(\tilde{k}))$ because the transfer function of a real PSF is Hermitian (Section 8.6.3) and thus obtain

$$ {}^{-}\hat{h}(\tilde{k}) = a(\tilde{k}) - \mathrm{i}b(\tilde{k}) \tag{9.48} $$

Thus, only the sign of the imaginary part of the transfer function changes when the filter direction is reversed.

We now have three possibilities to combine the two transfer functions (Eqs. (9.47) and (9.48)) either into a purely real or imaginary transfer function:

$$\begin{array}{ll} \text{Addition} & {}^{e}\hat{h}(\tilde{k}) = \frac{1}{2}\left({}^{+}\hat{h}(\tilde{k}) + {}^{-}\hat{h}(\tilde{k})\right) = a(\tilde{k}) \\ \text{Subtraction} & {}^{o}\hat{h}(\tilde{k}) = \frac{1}{2}\left({}^{+}\hat{h}(\tilde{k}) - {}^{-}\hat{h}(\tilde{k})\right) = \mathrm{i}b(\tilde{k}) \\ \text{Multiplication} & \hat{h}(\tilde{k}) = {}^{+}\hat{h}(\tilde{k})\,{}^{-}\hat{h}(\tilde{k}) = a^2(\tilde{k}) + b^2(\tilde{k}) \end{array} \quad (9.49)$$

Addition and multiplication (consecutive application) of the left and right running filter yields filters of even symmetry, while subtraction results in a filter of odd symmetry. This way to cascade recursive filters gives them the same properties as zero- or $\pi/2$-phase shift nonrecursive filters with the additional advantage that they can easily be tuned, and extended point spread functions can be realized with only a few filter coefficients.

9.3.6 Basic recursive filters

In Section 9.3.2 we found that the factorization of the generalized recursive filter is a key to analyze its transfer function and stability properties (Eq. (9.45)). The individual factors contain the poles and zeros. From each factor, we can compute the impulse response so that the resulting impulse response of the whole filter is given by a cascaded convolution of all components.

As the factors are all of the form

$$f_n(\tilde{k}) = 1 - c_n \exp(-2\pi \mathrm{i}\tilde{k}) \quad (9.50)$$

the analysis becomes quite easy. Still we can distinguish two basic types of partial factors. They result from the fact that the impulse response of the filter must be real. Therefore, the transfer function must be Hermitian, that is, $f^*(-k) = f(k)$. This can only be the case when either the zero c_n is real or a pair of factors exists with complex-conjugate zeros. This condition gives rise to two basic types of recursive filters, the relaxation filter and the resonance filter that are discussed in detail in what follows. As these filters are only useful for image processing if they are applied both in forward and backward direction, we discuss also the resulting symmetric transfer function and point spread function.

Relaxation filter. The transfer function of the relaxation filter running in forward or backward direction is

$$^{\pm}\hat{r}(\tilde{k}) = \frac{1-\alpha}{1-\alpha\exp(\mp\pi \mathrm{i}\tilde{k})} \quad \text{with} \quad \alpha \in \mathbb{R} \quad (9.51)$$

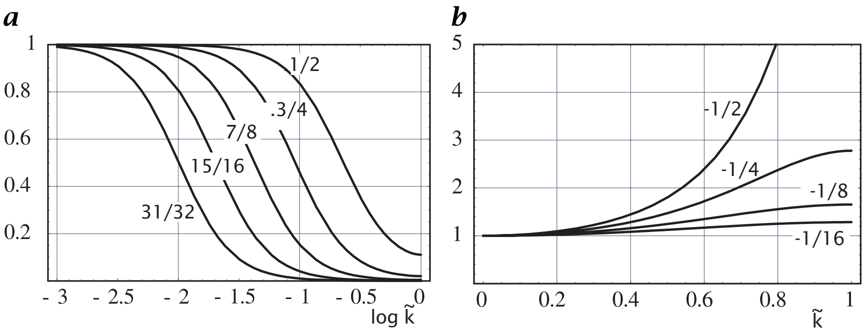

Figure 9.3: *Transfer function of the relaxation filter $g'_n = \alpha g'_{n\mp1} + (1-\alpha)g_n$ applied first in forward and then in backward direction for **a** positive; and **b** negative values of α as indicated.*

In this equation, the wave number has been replaced by the wave number normalized with the Nyquist limit (see Section 8.4.2, Eq. (8.34)). It also has been normalized so that $\hat{r}(0) = 1$. Comparing Eqs. (9.42) and (9.43) it is evident that the transfer function Eq. (9.51) belongs to the simple recursive filter

$$g'_n = \alpha g'_{n\mp1} + (1-\alpha)g_n = g_n + \alpha(g'_{n\mp1} - g_n) \tag{9.52}$$

with the point spread function

$$^{\pm}r_{\pm n} = \begin{cases} (1-\alpha)\alpha^n & n \geq 0 \\ 0 & \text{else} \end{cases} \tag{9.53}$$

This filter takes the fraction α from the previously calculated value and the fraction $1-\alpha$ from the current pixel.

The transfer function Eq. (9.51) is complex and can be divided into its real and imaginary parts as

$$^{\pm}\hat{r}(\tilde{k}) = \frac{1-\alpha}{1 - 2\alpha\cos\pi\tilde{k} + \alpha^2}\left[(1-\alpha\cos\pi\tilde{k}) \mp \mathrm{i}\alpha\sin\pi\tilde{k}\right] \tag{9.54}$$

From this transfer function, we can compute the multiplicative ($\hat{r}$) application of the filters by running it successively in positive and negative direction; see Eq. (9.49):

$$\hat{r}(\tilde{k}) = \frac{(1-\alpha)^2}{1 - 2\alpha\cos\pi\tilde{k} + \alpha^2} = \frac{1}{1 + \beta - \beta\cos\pi\tilde{k}} \tag{9.55}$$

with

$$\beta = \frac{2\alpha}{(1-\alpha)^2} \quad \text{and} \quad \alpha = \frac{1 + \beta - \sqrt{1 + 2\beta}}{\beta}$$

From Eq. (9.53) we can conclude that the relaxation filter is stable if $|\alpha| < 1$, which corresponds to $\beta \in]-1/2, \infty[$. As already noted, the transfer function is one for small wave numbers. A Taylor series in $\tilde{k}$ results in

$$\hat{r}(\tilde{k}) \approx= 1 - \frac{\alpha}{(1-\alpha)^2}(\pi\tilde{k})^2 + \frac{\alpha((1+10\alpha+\alpha^2)}{12(1-\alpha^2)^2}(\pi\tilde{k})^4 \tag{9.56}$$

If α is positive, the filter is a low-pass filter (Fig. 9.3a). It can be tuned by adjusting α. If α is approaching 1, the averaging distance becomes infinite. For negative α, the filter enhances high wave numbers (Fig. 9.3b).

This filter is the discrete analog to the first-order differential equation $\dot{y} + \tau y = 0$ describing a relaxation process with the relaxation time $\tau = -\Delta t / \ln \alpha$ [4].

Resonance filter. The transfer function of a filter with a pair of complex-conjugate zeros running in forward or backward direction is

$$\begin{aligned} {}^{\pm}\hat{s}(\tilde{k}) &= \frac{1}{1-r\exp(\mathrm{i}\pi\tilde{k}_0)\exp(\mp\mathrm{i}\pi\tilde{k})} \cdot \frac{1}{1-r\exp(-\mathrm{i}\pi\tilde{k}_0)\exp(\mp\mathrm{i}\pi\tilde{k})} \\ &= \frac{1}{1-2r\cos(\pi\tilde{k}_0)\exp(\mp\mathrm{i}\pi\tilde{k})+r^2\exp(\mp 2\mathrm{i}\pi\tilde{k})} \end{aligned} \tag{9.57}$$

The second row of the equation shows that this is the transfer function of the recursive filter

$$g'_n = g_n + 2r\cos(\pi\tilde{k}_0)g'_{n\mp 1} - r^2 g'_{n\mp 2} \tag{9.58}$$

The impulse response of this filter is [3]

$$h_{\pm n} = \begin{cases} \dfrac{r^n}{\sin\pi\tilde{k}_0}\sin[(n+1)\pi\tilde{k}_0] & n \geq 0 \\ 0 & n < 0 \end{cases} \tag{9.59}$$

If we run the filter back and forth, the resulting transfer function is

$$\hat{s}(\tilde{k}) = \frac{1}{\left(1-2r\cos[\pi(\tilde{k}-\tilde{k}_0)]+r^2\right)\left(1-2r\cos[\pi(\tilde{k}+\tilde{k}_0)]+r^2\right)} \tag{9.60}$$

From this equation, it is evident that this filter is a bandpass filter with a center wave number of $\tilde{k}_0$. The parameter r is related to the width of the bandpass. If $r = 1$, the transfer function has two *poles* at $\tilde{k} = \pm\tilde{k}_0$. If $r > 1$, the filter is unstable; even the slightest excitement will cause infinite amplitudes of the oscillation. The filter is only stable for $r \leq 1$.

The response of this filter can be normalized to obtain a bandpass filter with a unit response at the center wave number. The transfer function of this normalized filter is

$$\hat{s}(\tilde{k}) = \frac{(1-r^2)^2\sin^2(\pi\tilde{k}_0)}{(1+r^2)^2+2r^2\cos(2\pi\tilde{k}_0)-4r(1+r^2)\cos(\pi\tilde{k}_0)\cos(\pi\tilde{k})+2r^2\cos(2\pi\tilde{k})} \tag{9.61}$$

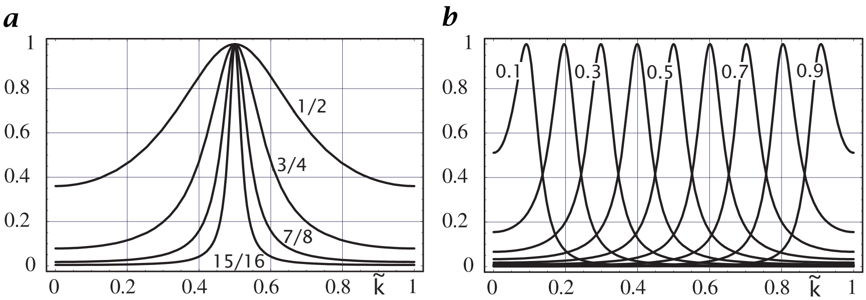

Figure 9.4: *Transfer function of the zero-phase recursive resonance filter for* ***a*** $\tilde{k}_0 = 1/2$ *and values of* r *as indicated; and* ***b*** $r = 7/8$ *and values of* $\tilde{k}_0$ *as indicated.*

The denominator in Eq. (9.61) is still the same as in Eq. (9.60); it has only been expanded in terms with $\cos(n\pi\tilde{k}_0)$. The corresponding recursive filter coefficients are:

$$g'_n = (1 - r^2)\sin(\pi\tilde{k}_0)g_n + 2r\cos(\pi\tilde{k}_0)g'_{n\mp1} - r^2 g'_{n\mp2} \qquad (9.62)$$

Figure 9.4 shows the transfer function of this filter for values of $\tilde{k}_0$ and r as indicated.

For symmetry reasons, the factors become most simple for a resonance wave number of $\tilde{k}_0 = 1/2$. Then the recursive filter is

$$g'_n = (1 - r^2)g_n - r^2 g'_{n\mp2} = g_n - r^2(g_n + g'_{n\mp2}) \qquad (9.63)$$

with the transfer function

$$\hat{s}(\tilde{k}) = \frac{(1 - r^2)^2}{1 + r^4 + 2r^2\cos(2\pi\tilde{k})} \qquad (9.64)$$

The maximum response of this filter at $\tilde{k} = 1/2$ is one and the minimum response at $\tilde{k} = 0$ and $\tilde{k} = 1$ is $((1 - r^2)/(1 + r^2))^2$.

This resonance filter is the discrete analog to a linear system governed by the second-order differential equation $\ddot{y} + 2\tau\dot{y} + \omega_0^2 y = 0$, the damped harmonic oscillator. The circular eigenfrequency ω_0 and the time constant τ of a real-world oscillator are related to the parameters of the discrete oscillator, r and $\tilde{k}_0$ by [4]

$$r = \exp(-\Delta t/\tau) \quad \text{and} \quad \tilde{k}_0 = \omega_0 \Delta t/\pi \qquad (9.65)$$

9.4 Classes of nonlinear filters

9.4.1 Limitations of linear filters

In the previous sections, the theory of linear shift-invariant filters was discussed in detail. Although the theory of these filters is well estab-