2 Fundamentals

The main purpose of this chapter is to introduce notation used throughout this book, and review fundamental principles of signal processing, statistical analysis and modeling that are used in subsequent chapters. Readers who are familiar e.g. with principles of one- and multi-dimensional sampling, random signal analysis, linear prediction and linear transforms may browse quickly over these topics.

2.1 Signals and systems

2.1.1 Elementary signals

A two-dimensional cosine signal defined over continuous coordinates $\mathbf{t}=[t_1\ t_2]^{\mathrm{T}}$ is given as

$$s_{\cos}(t_1,t_2)=\cos\left[2\pi\left(F_1t_1+F_2t_2\right)\right]=\cos\left[2\pi\breve{\mathbf{f}}^T\mathbf{t}\right]\ \text{with}\ \breve{\mathbf{f}}=\left[F_1\quad F_2\right]^{\mathrm{T}}. \tag{2.1}$$

After applying a coordinate transformation

$$\begin{bmatrix}\tilde{t}_1\\ \tilde{t}_2\end{bmatrix}=\begin{bmatrix}\cos\varphi & \sin\varphi\\ -\sin\varphi & \cos\varphi\end{bmatrix}\begin{bmatrix}t_1\\ t_2\end{bmatrix}\ \text{with}\ \ \varphi=\arctan\frac{F_2}{F_1}\ \ \left(\text{for}\ F_1\geq 0\right), \tag{2.2}$$

only a one-dimensional dependency remains as

$$s_{\cos}(\tilde{t}_1,\tilde{t}_2)=\cos\left[2\pi F\tilde{t}_1\right]\ \ \text{with}\ \ F=\sqrt{F_1^{\ 2}+F_2^{\ 2}}=\left\|\breve{\mathbf{f}}\right\|_2=\frac{F_1}{\cos\varphi}=\frac{F_2}{\sin\varphi}. \tag{2.3}$$

(2.1) can be interpreted as a sinusoidal wave front with orientation by an angle φ relative to the t_1 axis. Sections of this wave front in parallel with one of the two axes are observed as sinusoids of frequencies F_1 or F_2, respectively. These correspond to the periods or wavelengths (measured along the coordinate axes, see Fig. 2.1a)

J.-R. Ohm, *Multimedia Signal Coding and Transmission*,
Signals and Communication Technology, DOI 10.1007/978-3-662-46691-9_2

$$T_1 = \frac{1}{F_1} \quad ; \quad T_2 = \frac{1}{F_2}. \tag{2.4}$$

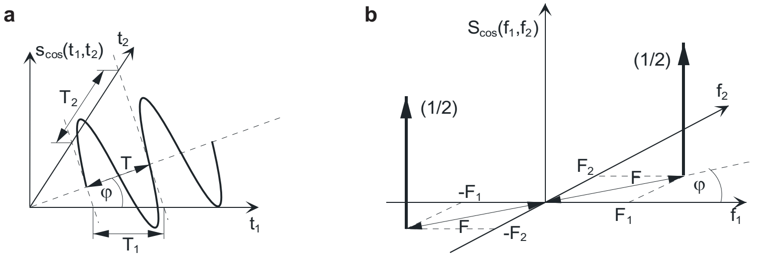

Fig. 2.1. **a** Directional orientation and wavelength of a sinusoid in a 2D plane **b** spectrum

As another interpretation, consider a cosine of period $T_1 = 1/F_1$ along the t_1 orientation, with phase shifted by $\phi(t_2)$ depending on the position in t_2,

$$s_{\cos}(t_1,t_2) = \cos\left[2\pi F_1 t_1 + \phi(t_2)\right]. \tag{2.5}$$

With linear dependency $\phi(t_2) = 2\pi F_2 t_2$, this is identical to (2.1). Then, for any $t_2 = k / F_2$ ($k \in \mathbb{Z}$), $\phi(t_2) = 2\pi k$. This determines distances where the signal has equal amplitude for a fixed t_1, i.e. $T_2 = 1/F_2$ is the period length along the t_2 orientation. Thus, the 2-dimensional cosine can also be interpreted as a sinusoid over one dimension which has a linear phase shift depending on the other dimension. This is illustrated in Fig. 2.2. Alternative formulations of the same signal would be

$$\begin{aligned} s_{\cos}(t_1,t_2) &= \cos\left[2\pi F_2 t_2 + \phi(t_1)\right] \text{ with } \phi(t_1) = 2\pi F_1 t_1 \text{ or} \\ s_{\cos}(t_1,t_2) &= \cos\left[2\pi F_1 t_1 + \phi(t_2)\right] \text{ with } \phi(t_2) = 2\pi F_2 t_2, \end{aligned} \tag{2.6}$$

such that any horizontal or vertical section over the different phase-shifted versions will also give a sinusoid of period T_2. Whereas T_1 and T_2 are the periods that can be measured w.r.t. to the coordinate axis orientations, the effective period of the 2-dimensional sinusoid, measured by the direction of wave front propagation, can be determined from (2.3) and (2.4) as

$$T = \frac{1}{F} = \frac{T_1 T_2}{\sqrt{T_1^2 + T_2^2}}. \tag{2.7}$$

Even though the example given here is based on a cosine function, a similar principle can be applied for any sinusoid. Likewise, it can be extended to a one- or multidimensional complex periodic exponential function

$$\begin{aligned} s_{\exp}(\mathbf{t}) &= \mathrm{e}^{\mathrm{j}2\pi F_1 t_1}\,\mathrm{e}^{\mathrm{j}2\pi F_2 t_2}\cdots\mathrm{e}^{\mathrm{j}2\pi F_\kappa t_\kappa} = \mathrm{e}^{\mathrm{j}2\pi\left[F_1 t_1 + F_2 t_2 + \ldots + F_\kappa t_\kappa\right]} = \mathrm{e}^{\mathrm{j}2\pi\breve{\mathbf{f}}^{\mathrm{T}}\mathbf{t}} \\ &= \cos\left(2\pi\breve{\mathbf{f}}^{\mathrm{T}}\mathbf{t}\right) + \mathrm{j}\sin\left(2\pi\breve{\mathbf{f}}^{\mathrm{T}}\mathbf{t}\right) \text{ with } \breve{\mathbf{f}} = \left[F_1 \quad \cdots \quad F_\kappa\right]^{\mathrm{T}}, \mathbf{t} = \left[t_1 \quad \cdots \quad t_\kappa\right]^{\mathrm{T}} \end{aligned} \tag{2.8}$$

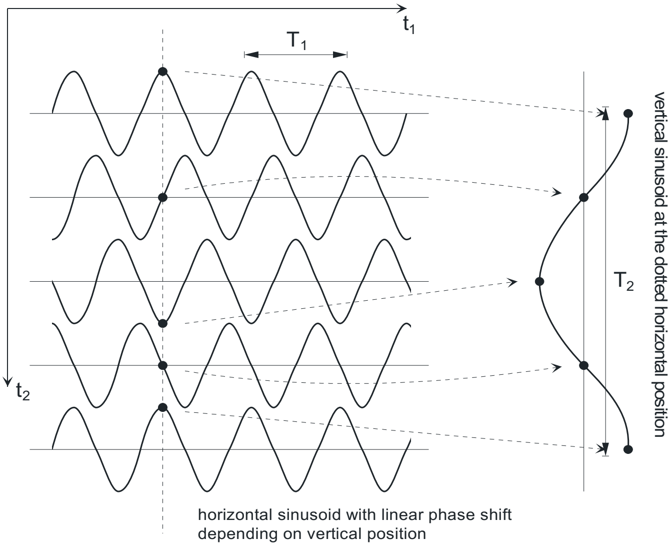
t₁
T₁
t₂
T₂
vertical sinusoid at the dotted horizontal position
horizontal sinusoid with linear phase shift
depending on vertical position

Fig. 2.2. Interpretation of a two-dimensional sinusoid: Linear phase shift depending on vertical position results in vertical wavelength and frequency

For the case κ=2, (2.1) is the real part of $s_{\exp}(\mathbf{t})$. For expression as a one-dimensional signal as in (2.3), $\kappa - 1$ rotations are necessary. If a signal can be defined by independent one-dimensional functions (as is the case with (2.8)), it is called *separable*, i.e.

$$s_{\text{sep}}(\mathbf{t}) = s(t_1)\cdot s(t_2)\cdot\cdots\cdot s(t_\kappa). \tag{2.9}$$

Some examples of aperiodic elementary 1D signals (which could be used to construct corresponding separable multi-dimensional signals) are the *sinc function*[1]

$$s(t) = \frac{\sin(\pi t)}{\pi t} = \mathrm{si}(\pi t)\,, \tag{2.10}$$

the *rectangular impulse*

$$\mathrm{rect}(t) = \begin{cases} 1, & |t| \le 1/2 \\ 0, & |t| > 1/2, \end{cases} \tag{2.11}$$

the *unit step function*

[1] sinc= *sin*us *c*ardinalis, si(x)=sin(x)/x with si(1)=1.

$$\varepsilon(t)=\begin{cases}1, & t\geq 0\\ 0, & t<0,\end{cases} \tag{2.12}$$

and the *Gaussian impulse*

$$s(t)=e^{-\pi t^2}. \tag{2.13}$$

2.1.2 Systems operations

A system generally performs a mapping (transfer) of an input $s(t)$ into an output $g(t)=\mathrm{Tr}\{s(t)\}$. A system is *linear*, if superposition using a weighted combination with constants a_i can be applied either at the input or at the output,

$$\mathrm{Tr}\left\{\sum_i a_i s_i(t)\right\}\overset{!}{=}\sum_i a_i\,\mathrm{Tr}\{s_i(t)\}=\sum_i a_i g_i(t). \tag{2.14}$$

Further, the system is *time invariant*, if for any shift t_0 of the input the output is shifted equally,

$$\mathrm{Tr}\{s(t-t_0)\}=g(t-t_0). \tag{2.15}$$

If a system fulfills both (2.14) and (2.15), it is called *linear time invariant* (LTI). The output signal of an LTI system fed by a *Dirac impulse* $\delta(t)$ as input is the *impulse response* $h(t)$. The transfer between input and output is given by the *convolution integrals*

$$s(t)=\int_{-\infty}^{\infty}s(\tau)\delta(t-\tau)\,\mathrm{d}\tau=s(t)*\delta(t), \tag{2.16}$$

$$g(t)=\int_{-\infty}^{\infty}s(\tau)h(t-\tau)\,\mathrm{d}\tau=s(t)*h(t). \tag{2.17}$$

The most important rules of convolution algebra are

a) The Dirac impulse is the *unity element* of convolution, according to (2.16).

b) *Commutative property*,

$$s(t)*h(t)=g(t)=\int_{+\infty}^{-\infty}s(t-\theta)h(\theta)\left(-\mathrm{d}\theta\right)=\int_{-\infty}^{+\infty}h(\theta)s(t-\theta)\,\mathrm{d}\theta=h(t)*s(t). \tag{2.18}$$

c) *Associative property*[2],

$$f(t)*s(t)*h(t)=\left[f(t)*s(t)\right]*h(t)=f(t)*\left[s(t)*h(t)\right]. \tag{2.19}$$

d) *Distributive property*,

$$f(t)*\left[s(t)+h(t)\right]=\left[f(t)*s(t)\right]+\left[f(t)*h(t)\right]. \tag{2.20}$$

Convolution can straightforwardly be extended to signals with multi-dimensional dependencies, e.g. image signals where an amplitude is defined for positions with horizontal/vertical coordinates (t_1,t_2). An example of a two-dimensional convolu-

[2] For combinations of convolution with other operations, in particular multiplication of functions, this is not true; the sequence of processing needs to be observed.

tion integral, with both the signal and the impulse response having two-dimensional dependencies, is defined as

$$g(t_1,t_2)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\tau_1,\tau_2)h(t_1-\tau_1,t_2-\tau_2)\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 = s(t_1,t_2)**h(t_1,t_2). \qquad (2.21)$$

If κ dimensions are combined into a vector $\mathbf{t}=[t_1,..., t_\kappa]^\mathrm{T}$, same with the variables of the convolution integral $\boldsymbol{\tau}=[\tau_1,....,\tau_\kappa]^\mathrm{T}$, multi-dimensional convolution is defined by[3]

$$s(\mathbf{t})=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} s(\boldsymbol{\tau})\delta(\mathbf{t}-\boldsymbol{\tau})\,\mathrm{d}^\kappa\boldsymbol{\tau} = s(\mathbf{t}) \ast \delta(\mathbf{t}). \qquad (2.22)$$

$$g(\mathbf{t})=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} s(\boldsymbol{\tau})h(\mathbf{t}-\boldsymbol{\tau})\,\mathrm{d}^\kappa\boldsymbol{\tau} = s(\mathbf{t}) \ast h(\mathbf{t}). \qquad (2.23)$$

(2.22) can be interpreted via the *sifting property* of the Dirac impulse, which contributes only the signal value $\boldsymbol{\tau}=\mathbf{t}$ to the result of the integration. The multi-dimensional Dirac impulse can be described as a separable combination of a series of 1D Dirac impulses[4], each of which performs sifting in one dimension. Therefore,

$$\delta(\mathbf{t})=\delta(t_1)\cdot\delta(t_2)\cdot\cdots \quad \text{with} \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\delta(\mathbf{t})\,\mathrm{d}^\kappa\mathbf{t}=\int_{-\infty}^{\infty}\delta(t_1)\,\mathrm{d}t_1\int_{-\infty}^{\infty}\delta(t_2)\,\mathrm{d}t_2\cdots=1 \qquad (2.24)$$

Properties (2.18)-(2.20) still hold for multi-dimensional convolution. An interesting class of two- and multi-dimensional LTI[5] systems are the *separable systems* with an impulse response that can be written as a multiplication of two or more functions, e.g. in the two-dimensional case[6]

$$h(t_1,t_2)=h_1(t_1)\cdot h_2(t_2)=\left[h_1(t_1)\cdot\delta(t_2)\right]**\left[\delta(t_1)\cdot h_2(t_2)\right]. \qquad (2.25)$$

Inserting (2.25) into (2.21) unveils that convolution in case of a 2D separable system can be implemented as a concatenation of two 1D convolutions to be performed at any position of the respective other dimension,

[3] The bold star symbol ($\ast$) expresses convolution over vector variables, to be performed by nested integrations.

[4] A 1D Dirac impulse $\delta(t_1)$ in a two- or multi-dimensional coordinate system can be interpreted as a line impulse, plane impulse or hyper-plane impulse (depending on the number of dimensions). It is zero for any $t_1\neq0$, but can be interpreted as an infinite-amplitude slice positioned at $t_1=0$ with infinite extension over the remaining dimension(s), with volume integration over the entire multi-dimensional space giving a value of 1.

[5] For sake of simplicity, the denotation *time invariant* is not changed, even though typically at most one of the dependencies in a multi-dimensional system is along the time axis.

[6] In the expression by 2D convolution the line impulses are needed to indicate the presence of the impulse response at any position of the other dimension(s).

$$\begin{aligned} g(t_1,t_2) &= s(t_1,t_2)**h(t_1,t_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\tau_1,\tau_2)h_1(t_1-\tau_1)h_2(t_2-\tau_2)\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \\ &= \int_{-\infty}^{\infty} h_2(t_2-\tau_2)\underbrace{\int_{-\infty}^{\infty} s(\tau_1,\tau_2)h_1(t_1-\tau_1)\,\mathrm{d}\tau_1}_{g_1(t_1,\tau_2)}\,\mathrm{d}\tau_2 \\ &= \underbrace{s(t_1,t_2)**\left[h_1(t_1)\cdot\delta(t_2)\right]}_{g_1(t_1,t_2)}**\left[\delta(t_1)\cdot h_2(t_2)\right]. \end{aligned} \tag{2.26}$$

Due to the associative property (2.19), the sequence of processing the dimensions is irrelevant in case of separable systems.

Eigenfunctions have the property that their shape is not changed when they are transmitted over an LTI system; the output can be computed by multiplication with a complex amplitude factor *H*, the related *eigenvalue*. A periodic 1D eigenfunction can be defined as a special case of (2.8),

$$s_{\mathrm{E}}(\mathbf{t}) = \mathrm{e}^{\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}} = \cos\left(2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}\right) + \mathrm{j}\sin\left(2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}\right). \tag{2.27}$$

Transmission over an LTI system gives

$$\begin{aligned} s_{\mathrm{E}}(\mathbf{t}) * h(\mathbf{t}) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h(\boldsymbol{\tau})\,\mathrm{e}^{\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}(\mathbf{t}-\boldsymbol{\tau})}\,\mathrm{d}^{\kappa}\boldsymbol{\tau} \\ &= \mathrm{e}^{\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}}\underbrace{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h(\boldsymbol{\tau})\,\mathrm{e}^{-\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\boldsymbol{\tau}}\,\mathrm{d}^{\kappa}\boldsymbol{\tau}}_{H(\mathbf{f})} = H(\mathbf{f})\,\mathrm{e}^{\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}}. \end{aligned} \tag{2.28}$$

The type of complex periodic eigenfunctions plays an important role in *Fourier analysis*, establishing a relation between signal (**t**) and Fourier (**f**) domains. Herein,

$$H(\mathbf{f}) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} h(\mathbf{t})\,\mathrm{e}^{-\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}}\,\mathrm{d}^{\kappa}\mathbf{t} \tag{2.29}$$

is the relation of Fourier transform of the impulse response *h*(t), giving the frequency-dependent Fourier transfer function *H*(**f**) of an LTI system. Feeding an eigenfunction into a series of two LTI systems with impulse responses $h_{\mathrm{A}}(\mathbf{t})$ and $h_{\mathrm{B}}(\mathbf{t})$ gives the result

$$s_{\mathrm{E}}(\mathbf{t}) * h_{\mathrm{A}}(\mathbf{t}) * h_{\mathrm{B}}(\mathbf{t}) = \left[H_{\mathrm{A}}(\mathbf{f})s_{\mathrm{E}}(\mathbf{t})\right] * h_{\mathrm{B}}(\mathbf{t}) = H_{\mathrm{A}}(\mathbf{f})\cdot H_{\mathrm{B}}(\mathbf{f})\cdot s_{\mathrm{E}}(\mathbf{t}) \tag{2.30}$$

It can be concluded that the convolution product in the time domain is mapped to an algebraic product in the frequency domain.

The Fourier transform is applicable not only for impulse responses *h*(**t**), but for any signals *s*(**t**), *g*(**t**) etc. into their corresponding Fourier spectra

$$S(\mathbf{f}) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} s(\mathbf{t})\,\mathrm{e}^{-\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{t}}\,\mathrm{d}^{\kappa}\mathbf{t}.$$

Two- and multi-dimensional extensions of eigenfunctions are straightforward and establish the basis of multi-dimensional Fourier spectra that are discussed in the subsequent sections. Due to the separable property of the multi-dimensional complex eigenfunctions, the multi-dimensional Fourier transform can be computed sequentially over the different dimensions, but the final result still provides an interpretation by directional orientation.

2.2 Signals and Fourier spectra

2.2.1 Spectra over two- and multi-dimensional coordinates

Rectangular coordinate systems. The amplitude of an image signal is dependent on the spatial position in two dimensions t_1 and t_2 – horizontally and vertically. Related frequency axis orientations shall be f_1 (horizontally) and f_2 (vertically). The two-dimensional Fourier transform of a spatially continuous signal is

$$S(f_1,f_2)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(t_1,t_2)\,\mathrm{e}^{-\mathrm{j}2\pi f_1 t_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2 t_2}\,\mathrm{d}t_1\,\mathrm{d}t_2. \tag{2.31}$$

(2.31) can be extended into a generic definition of κ-dimensional spectra associated with a κ-dimensional signal, where all frequency coordinates $\mathbf{f}=[f_1\ f_2\ ...\ f_\kappa]^\mathrm{T}$ and signal coordinates in space and time $\mathbf{t}=[t_1\ t_2\ ...\ t_\kappa]^\mathrm{T}$ are expressed as vectors. This gives

$$S(\mathbf{f})=\int_{-\infty}^{\infty}..\int_{-\infty}^{\infty} s(\mathbf{t})\,\mathrm{e}^{-\mathrm{j}2\pi\mathbf{f}^\mathrm{T}\mathbf{t}}\,\mathrm{d}^\kappa\,\mathbf{t}. \tag{2.32}$$

The complex spectrum can be interpreted by *magnitude* and *phase* of a contributing oscillation at a given frequency **f**,

$$\begin{aligned}|S(\mathbf{f})|&=\sqrt{\left[\mathrm{Re}\{S(\mathbf{f})\}\right]^2+\left[\mathrm{Im}\{S(\mathbf{f})\}\right]^2}=\sqrt{S(\mathbf{f})S^*(\mathbf{f})}\\ \varphi(\mathbf{f})&=\arctan\frac{\mathrm{Im}\{S(\mathbf{f})\}}{\mathrm{Re}\{S(\mathbf{f})\}}\pm\pi\cdot k(\mathbf{f})\ \text{ with }\ k(\mathbf{f})=\begin{cases}1 & \text{for } \mathrm{Re}\{S(\mathbf{f})\}<0\\ 0 & \text{else.}\end{cases}\end{aligned} \tag{2.33}$$

By *inverse Fourier transform*, the signal can be reconstructed from the Fourier spectrum:

$$s(\mathbf{t})=\int_{-\infty}^{\infty}...\int_{-\infty}^{\infty} S(\mathbf{f})\,\mathrm{e}^{\mathrm{j}2\pi\mathbf{f}^\mathrm{T}\mathbf{t}}\,\mathrm{d}^\kappa\,\mathbf{f}. \tag{2.34}$$

Coordinate system mapping. Rectangular (orthogonal) coordinate systems are only a special case for the description of two- and multidimensional signals. They allow expressing the multi-dimensional Fourier transform through eigenfunctions which are also orthogonal (i.e. independent in terms of signal analysis properties) between the different dimensions. Two unity vectors $\mathbf{e}_1 = [1\ \ 0]^{\mathrm{T}}$ and $\mathbf{e}_2 = [0\ \ 1]^{\mathrm{T}}$ define the orientations of the axes. Any coordinate pair (t_1,t_2) can then be expressed as a vector $\mathbf{t} = t_1\mathbf{e}_1 + t_2\mathbf{e}_2$. The relationship with frequency vectors $\mathbf{f} = f_1\mathbf{e}_1 + f_2\mathbf{e}_2$ is given by (2.32), using the same orientation. Now, a linear coordinate mapping $\tilde{\mathbf{t}} = t_1\mathbf{t}_1 + t_2\mathbf{t}_2 = \mathbf{Tt}$ shall be applied to the signal (leaving the coordinate origin unchanged), which can be expressed through the mapping matrix[7]

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 \end{bmatrix}. \tag{2.35}$$

The vectors $\mathbf{t}_1$ und $\mathbf{t}_2$ are the *basis vectors* of this mapping. A complementary mapping of frequency coordinates shall exist, expressed similarly as $\tilde{\mathbf{f}} = \mathbf{Ff}$ by using a mapping matrix

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 \end{bmatrix}. \tag{2.36}$$

Unless the determinants of matrices **T** or **F** are zero, the mappings must be reversible, such that $\mathbf{t} = \mathbf{T}^{-1}\tilde{\mathbf{t}}$ and $\mathbf{f} = \mathbf{F}^{-1}\tilde{\mathbf{f}}$. The relations are given by biorthogonality (A.25) of **T** and **F** [see e.g. OHM 2004],

$$\mathbf{T}^{-1} = \mathbf{F}^{\mathrm{T}}; \quad \mathbf{F}^{-1} = \mathbf{T}^{\mathrm{T}} \quad \Rightarrow \quad \mathbf{F} = \left[\mathbf{T}^{-1}\right]^{\mathrm{T}}; \quad \mathbf{T} = \left[\mathbf{F}^{-1}\right]^{\mathrm{T}}. \tag{2.37}$$

The Fourier transform in the mapped coordinate system can then be expressed as follows, assuming amplitude invariance of the mapped samples,

$$\tilde{S}(\tilde{\mathbf{f}}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} s(\tilde{\mathbf{t}})\,\mathrm{e}^{-\mathrm{j}2\pi\tilde{\mathbf{f}}^{\mathrm{T}}\tilde{\mathbf{t}}}\,\mathrm{d}^{\kappa}\,\tilde{\mathbf{t}} = |\mathbf{T}|\,S(\mathbf{f}). \tag{2.38}$$

2.2.2 Spatio-temporal signals

In a video signal, two-dimensional pictures vary over time. The time dependency t is mapped into a 'temporal' frequency f_3, where the Fourier spectrum is

$$S(f_1,f_2,f_3) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(t_1,t_2,t_3)\,\mathrm{e}^{-\mathrm{j}2\pi f_1 t_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2 t_2}\,\mathrm{e}^{-\mathrm{j}2\pi f_3 t_3}\,\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3. \tag{2.39}$$

For the case of sinusoids, the spectral property resulting by temporal changes can

[7] It is assumed here that the origin of the coordinate transform is not changed by the mapping. A more general form is the mapping $\tilde{\mathbf{t}} = \mathbf{Tt} + \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ expresses a shift of the origin. This is also denoted as *affine mapping*. Regarding the Fourier spectrum, the additional translation only effects a linear phase shift $\mathrm{e}^{\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\boldsymbol{\tau}}$.

be interpreted similarly to Fig. 2.3. In particular, if motion is constant (without local variations of shift and without acceleration) and the amplitude of the signal is only changing by motion, the behavior of the signal can be expressed by a linear phase shift in t_1 and t_2, depending on time t_3. Consider first the case of zero motion, $s(t_1,t_2,t_3) = s(t_1,t_2,0)$. Then, the three-dimensional Fourier spectrum (2.39) is

$$\begin{aligned} S(f_1,f_2,f_3) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(t_1,t_2,0)\,\mathrm{e}^{-\mathrm{j}2\pi f_1 t_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2 t_2}\,\mathrm{d}t_1\,\mathrm{d}t_2 \cdot \int_{-\infty}^{\infty} \mathrm{e}^{-\mathrm{j}2\pi f_3 t_3}\,\mathrm{d}t_3 \\ &= S(f_1,f_2)\big|_{t_3=0} \cdot \delta(f_3). \end{aligned} \tag{2.40}$$

The Dirac impulse $\delta(f_3)$ indicates that the 3D spectrum in case of unchanged signals is a sampled plane, with non-zero components only at f_3=0:

$$S(f_1,f_2,f) \sim \begin{cases} S(f_1,f_2)\big|_{t_3=0} & \text{when} \quad f_3 = 0 \\ 0 & \text{when} \quad f_3 \neq 0. \end{cases} \tag{2.41}$$

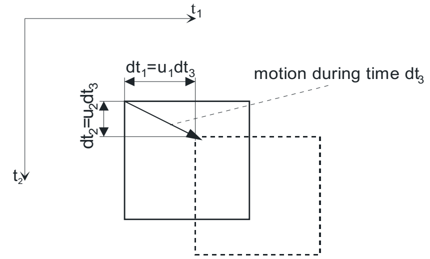

Fig. 2.3. Spatial shift caused by translational motion of velocity **u**

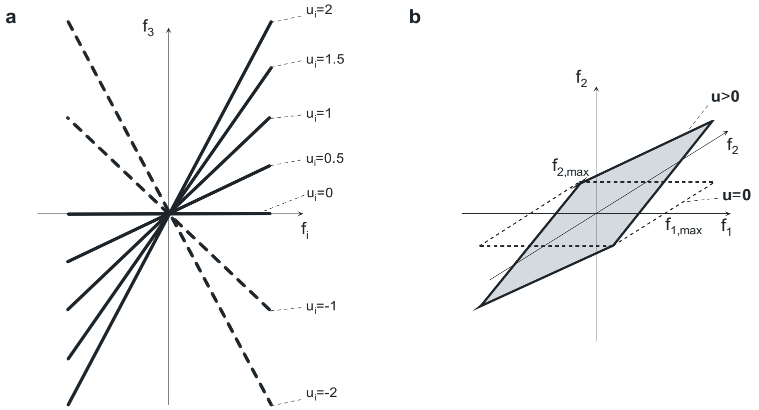

Fig. 2.4. **a** Shear of the non-zero spectral components by different translational motion velocities, shown in an (f_i,f_3) section (i=1,2) of the 3D frequency domain **b** Position of the non-zero spectral components in cases of zero and non-zero 2D translational motion

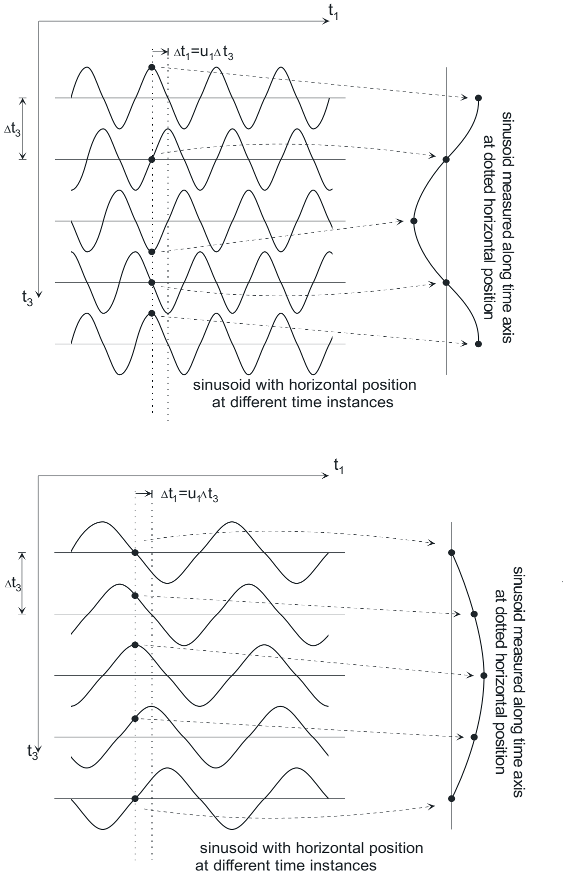
t1
Δt1=u1Δt3
Δt3
t3
sinusoid measured along time axis at dotted horizontal position
sinusoid with horizontal position at different time instances
t1
Δt1=u1Δt3
Δt3
t3
sinusoid measured along time axis at dotted horizontal position
sinusoid with horizontal position at different time instances

Fig. 2.5. Interpretation of the frequency in f_3 for two sinusoids of different spatial frequencies, which are subject to the same translational motion

If constant-velocity translation motion is present in the signal, a spatial shift $\mathrm{d}t_1=u_1\mathrm{d}t_3$ in horizontal direction and $\mathrm{d}t_2=u_2\mathrm{d}t_3$ in vertical direction occurs within time interval $\mathrm{d}t_3$, which is linearly dependent on the velocity vector $\mathbf{u}=[u_1,u_2]^{\mathrm{T}}$ (see Fig. 2.3). Taking reference to the signal for $t_3=0$, this gives

$$s(t_1,t_2,t_3)=s(t_1+u_1t_3,t_2+u_2t_3,0) \tag{2.42}$$

and

$$S(f_1,f_2,f_3)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}s(t_1+u_1t_3,t_2+u_2t_3,0)\,\mathrm{e}^{-\mathrm{j}2\pi f_1t_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2t_2}\,\mathrm{e}^{-\mathrm{j}2\pi f_3t_3}\,\mathrm{d}t_1\,\mathrm{d}t_2\,\mathrm{d}t_3. \tag{2.43}$$

By replacing $\tau_i=t_i+u_it_3\Rightarrow\mathrm{d}\tau_i=\mathrm{d}t_i, t_i=\tau_i-u_it_3$, the temporal dependency can be separated in the Fourier integration

$$\begin{aligned} S(f_1,f_2,f_3) &= \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} s(\tau_1,\tau_2)\,\mathrm{e}^{-\mathrm{j}2\pi f_1\tau_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2\tau_2}\,\mathrm{d}\tau_1\,\mathrm{d}\tau_2 \cdot \int\limits_{-\infty}^{\infty} e^{-\mathrm{j}2\pi(f_3-f_1u_1-f_2u_2)t_3}\,d\tau_3 \\ &= S(f_1,f_2)\big|_{t_3=0}\cdot\delta(f_3-f_1u_1-f_2u_2). \end{aligned} \tag{2.44}$$

Thus,

$$S(f_1,f_2,f_3)\sim\begin{cases} S(f_1,f_2)\big|_{t=0} & \text{when } f_3=f_1u_1+f_2u_2 \\ 0 & \text{when } f_3\neq f_1u_1+f_2u_2. \end{cases} \tag{2.45}$$

The spectrum $S(f_1,f_2)$ is now sampled on a plane $f_3=f_1u_1+f_2u_2$ in the 3D frequency domain. Fig. 2.4a shows positions of non-zero spectrum planes for different normalized velocities u_i, where the (f_i,f_3) section is shown for $f_j=0$, $i\neq j,(i,j)\in[1,2]$. Fig. 2.4b shows qualitatively the behavior in the full (f_1,f_2,f_3) space for the zero-motion case and for motion by one constant velocity $\mathbf{u}>\mathbf{0}$, where further the spectrum is assumed to be band-limited in f_1 and f_2.

Fig. 2.4 illustrates that the positions of non-zero spectral values in case of constant velocity are found via a linear relationship between f_3 and the frequencies relating to the spatial coordinates. This effect can also be interpreted in the signal domain. Fig. 2.5 shows two sinusoids of different frequencies f_1, both moving by the same velocity. The phase shift occurring due to the constant-velocity motion linearly depends on the given spatial frequency.

2.3 Sampling of multimedia signals

Ideal sampling describes the multiplication (modulation) of a signal by a regular (equidistant) train of Dirac impulses. In the 1D case, this gives

$$s_{\delta_T}(t)=s(t)\sum_{n=-\infty}^{\infty}\delta\left(t-nT\right)=\sum_{n=-\infty}^{\infty}s(nT)\delta\left(t-nT\right). \tag{2.46}$$

An example is shown in Fig. 2.6.

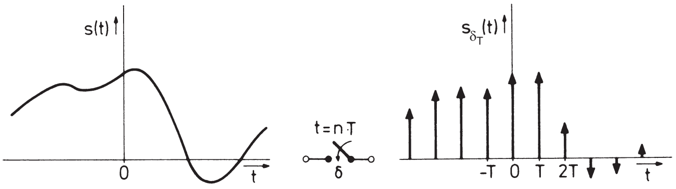

Fig. 2.6. Output $s_{\delta_T}(t)$ of an ideal sampling unit

The *ideal sampler* generates a discrete-time, equidistant series of weighted Dirac impulses from the continuous-time signal $s(t)$. The weights are the *samples* $s(nT)$. The Fourier spectrum of the sampled signal is

$$\begin{array}{ccccc} s_{\delta_T}(t) & = & s(t) & \cdot & \sum\limits_{n=-\infty}^{\infty} \delta\left(t-nT\right) \\ \circ\!\!-\!\!\bullet & & \circ\!\!-\!\!\bullet & \circ\!\!-\!\!\bullet & \circ\!\!-\!\!\bullet \\ S_{\delta_T}(f) & = & S(f) & * & \frac{1}{\left|T\right|}\sum\limits_{n=-\infty}^{\infty} \delta\left(f-\frac{n}{T}\right) \end{array} \tag{2.47}$$

$S\delta_T(f)$ is periodic by the *sampling rate* $1/T$, with spectral copies scaled in amplitude,

$$S_{\delta_T}(f)=\frac{1}{\left|T\right|}\sum_{k=-\infty}^{\infty} S\left[f-k/T\right]. \tag{2.48}$$

Fig. 2.7 shows this relation for a real-valued band-limited *lowpass signal* with zero-valued spectrum at any $|f| \geq f_c$ with cut-off frequency f_c.

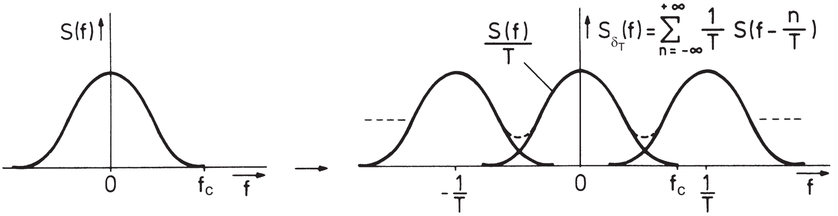

Fig. 2.7. Periodic components in the Fourier spectrum of a sampled signal $s_{\delta_T}(t)$

When sampling is performed using a sampling period

$$T \leq \frac{1}{2f_c}, \tag{2.49}$$

the periodic copies of the spectrum $S(f)$ in $S\delta_T(f)$ are not overlapping, such that the original $S(f)$ can be perfectly reconstructed by suitable lowpass filtering from $S\delta_T(f)$. This basic idea of sampling is shown in Fig. 2.8. If (2.49) is violated, frequency components from the periodic copies may appear in the baseband after the lowpass filtering, which is denoted as *aliasing*.

The lowpass shall have a transfer function which is flat in the range $|f| < f_c$ of the pass band and shall perfectly discard frequencies $|f| > 1/T - f_c$ from $S\delta_T(f)$. Assuming that an *ideal lowpass* is used, the reconstruction of the continuous-time signal from the sampled signal can be formulated in the frequency and time domains as (see Fig. 2.8)

$$\begin{array}{ccccccc} S(f) & = & S_{\delta_T}(f) & \cdot & T\,\mathrm{rect}\left(\dfrac{f}{2f_\mathrm{c}}\right) \\ \bullet\!\!-\!\!\circ & & \bullet\!\!-\!\!\circ & \bullet\!\!-\!\!\circ & \bullet\!\!-\!\!\circ \\ s(t) & = & s_{\delta_T}(t) & * & 2f_\mathrm{c}T\,\mathrm{si}\left(\pi 2 f_\mathrm{c} t\right). \end{array} \tag{2.50}$$

If the largest possible sampling period $T = 1/(2f_\mathrm{c})$ is used, (2.50) gives

$$s(t) = \left[\sum_{n=-\infty}^{\infty} s(nT)\delta(t-nT)\right] * \mathrm{si}\left(\pi\frac{t}{T}\right) = \sum_{n=-\infty}^{\infty} s(nT)\,\mathrm{si}\left(\pi\frac{t-nT}{T}\right). \tag{2.51}$$

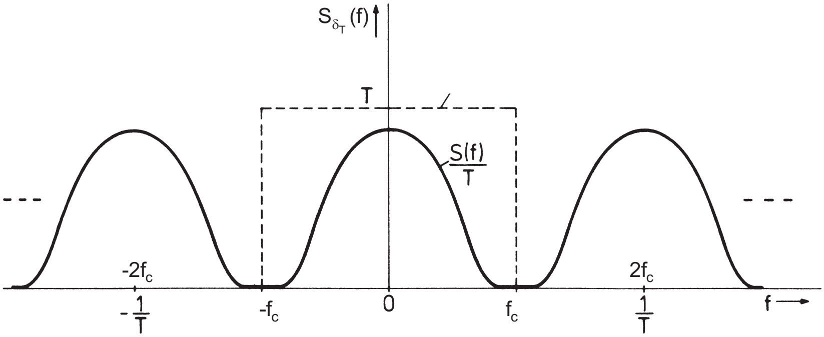

Fig. 2.8. Reconstruction of the Fourier spectrum $S(f)$ from $S_{\delta_T}(f)$ by using an ideal lowpass filter of cut-off frequency f_c

This formulation of the *sampling theorem* shows, that a real-valued signal which is band limited within a given lowpass range limited by f_c can be described without any errors by an equidistant series of weighted sinc functions. This is also denoted as the *cardinal series* of $s(t)$. The weights are equal to the samples of the signal as extracted with distances $T = 1/(2f_\mathrm{c})$ from the signal. Fig 2.9 shows this principle.

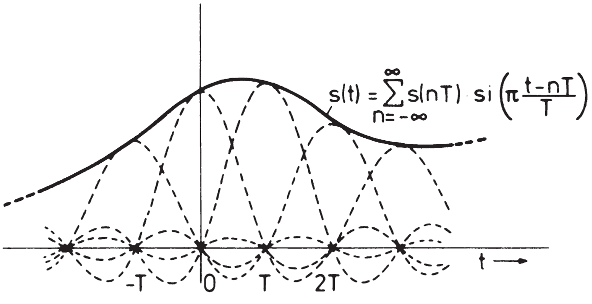
$s(t) = \sum_{n=-\infty}^{\infty} s(nT)\ \mathrm{si}\left(\pi\frac{t-nT}{T}\right)$
-T
0
T
2T
t

Fig. 2.9. Band limited real-valued lowpass signal $s(t)$ reconstructed by superposition of weighted sinc functions with distances $T=1/(2f_\mathrm{g})$

2.3.1 Separable two-dimensional sampling

Separable two- or multidimensional sampling is independent in the respective dimensions. This can be expressed from 1D *Dirac impulse trains* (refer to (2.47))

$$\delta_{T_i}(t_i)=\sum_{n_i=-\infty}^{\infty}\delta(t_i-n_iT_i)\,. \tag{2.52}$$

By multiplication of two impulse trains, which are separable on a rectangular grid, a two-dimensional ideal sampling function is defined as

$$\delta_{T_1,T_2}(t_1,t_2)=\delta_{T_1}(t_1)\cdot\delta_{T_2}(t_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}\delta(t_1-n_1T_1,t_2-n_2T_2)\,. \tag{2.53}$$

Due to separable property, the rectangular impulse grid has a 2D spectrum

$$\begin{aligned}\frac{1}{|T_1|}\delta_{1/T_1}(f_1)\cdot\frac{1}{|T_2|}\delta_{1/T_2}(f_2)&=\frac{1}{|T_1T_2|}\delta_{1/T_1,1/T_2}(f_1,f_2)\\&=\frac{1}{|T_1T_2|}\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}\delta(f_1-k_1/T_1,f_2-k_2/T_2).\end{aligned} \tag{2.54}$$

The operation of ideal rectangular-grid sampling of a spatially-continuous 2D signal $s(t_1,t_2)$ is expressed by multiplication with $\delta_{T_1,T_2}(t_1,t_2)$. The sample aspect ratio is defined as T_1/T_2. The discrete signal $s(n_1,n_2)$ consists of amplitude samples $s(n_1T_1,n_2T_2)$. Its spectrum is

$$\begin{aligned}S_{\delta_{T_1T_2}}(f_1,f_2)&=\frac{1}{|T_1T_2|}S(f_1,f_2)**\delta_{1/T_1,1/T_2}(f_1,f_2)\\&=\frac{1}{|T_1T_2|}\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}S\big(f_1-k_1/T_1,f_2-k_2/T_2\big).\end{aligned} \tag{2.55}$$

It is also possible to compute the periodic spectrum directly from the discrete series of samples:

$$\begin{aligned}S_{\delta_{T_1T_2}}(f_1,f_2)&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}s_{\delta_{T_1,T_2}}(t_1,t_2)\mathrm{e}^{-\mathrm{j}2\pi f_1t_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2t_2}\,\mathrm{d}t_1\,\mathrm{d}t_2\\&=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}s(n_1T_1,n_2T_2)\delta(t_1-n_1T_1,t_2-n_2T_2)\mathrm{e}^{-\mathrm{j}2\pi f_1t_1}\,\mathrm{e}^{-j2\pi f_2t_2}\,\mathrm{d}t_1\,\mathrm{d}t_2\\&=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}s(n_1T_1,n_2T_2)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(t_1-n_1T_1,t_2-n_2T_2)\mathrm{e}^{-\mathrm{j}2\pi f_1t_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2t_2}\,\mathrm{d}t_1\,\mathrm{d}t_2\\&=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}s(n_1T_1,n_2T_2)\mathrm{e}^{-\mathrm{j}2\pi f_1n_1T_1}\,\mathrm{e}^{-\mathrm{j}2\pi f_2n_2T_2}\,,\end{aligned} \tag{2.56}$$

or by performing normalization by setting $T_1=T_2=1$,

$$S_\delta(f_1,f_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}s(n_1,n_2)\mathrm{e}^{-\mathrm{j}2\pi f_1 n_1}\mathrm{e}^{-\mathrm{j}2\pi f_2 n_2}\,. \tag{2.57}$$

2D pulse grid sampling generates periodic copies of the spectrum along *both directions*. Examples of Fourier amplitude spectra $|S(f_1,f_2)|$ and $|S_\delta(f_1,f_2)|$ in case of rectangular sampling are shown in Fig. 2.10.

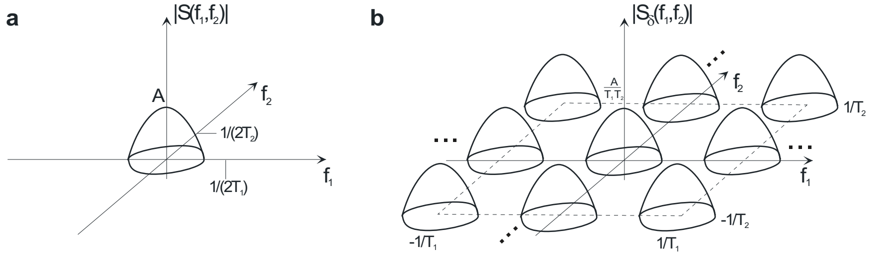

Fig. 2.10. Spectra of 2D image signals: **a** Continuous signal **b** sampled signal

To allow reconstruction by a 2D lowpass filter, $s(t_1,t_2)$ has to be band limited before sampling. 2D separable sampling allows perfect reconstruction by a lowpass interpolation filter if

$$S(f_1,f_2)\overset{!}{=}0 \quad \text{for} \quad |f_1|\geq\frac{1}{2T_1} \quad \text{or} \quad |f_2|\geq\frac{1}{2T_2}\,, \tag{2.58}$$

such that

$$\begin{aligned} S(f_1,f_2) &= T_1T_2S_{\delta_{T_1T_2}}(f_1,f_2)\cdot\mathrm{rect}\left(T_1f_1\right)\cdot\mathrm{rect}\left(T_2f_2\right) \\ &\quad\begin{matrix}\bullet\\ \circ\end{matrix} \\ s(t_1,t_2) &= s_{\delta_{T_1T_2}}(t_1,t_2)**\mathrm{si}\left(\pi\frac{t_1}{T_1}\right)\cdot\delta(t_2)**\mathrm{si}\left(\pi\frac{t_2}{T_2}\right)\cdot\delta(t_1) \\ &= \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}s(n_1T_1,n_2T_2)\,\mathrm{si}\left[\pi\left(\frac{t_1}{T_1}-n_1\right)\right]\mathrm{si}\left[\pi\left(\frac{t_2}{T_2}-n_2\right)\right]. \end{aligned} \tag{2.59}$$

This method of separable sampling can be straightforwardly extended to an arbitrary number of dimensions.

2.3.2 Non-separable two-dimensional sampling

Equidistant one-dimensional sampling and separable multi-dimensional sampling have only one degree of freedom (per dimension) in varying the sampling distance T. In case of non-separable sampling, sampling positions are still following a regular pattern, but need to be formulated with mutual dependency. Different regular grids of 2D sampling are illustrated in Fig. 2.11. Regularity means a systematic periodicity of a basic structure, which can be expressed by a system of *basis vectors* $\mathbf{t}_1 = [t_{11}\ t_{21}]^{\mathrm{T}}$, $\mathbf{t}_2 = [t_{12}\ t_{22}]^{\mathrm{T}}$. Linear combinations of these vectors, when multiplied by the integer vector index $\mathbf{n} = [n_1,n_2]^{\mathrm{T}}$, point to the effective positions $\mathbf{t}(\mathbf{n}) = n_1\mathbf{t}_1 + n_2\mathbf{t}_2$, which could be interpreted as 'centers of sampling cells'. The basis vectors are the columns of a coordinate transformation matrix **T**, which in this context is also denoted as *sampling matrix*:

$$\underbrace{\begin{bmatrix} t_1(n_1,n_2) \\ t_2(n_1,n_2) \end{bmatrix}}_{\mathbf{t}(\mathbf{n})} = \underbrace{\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}}_{\mathbf{T}} \cdot \underbrace{\begin{bmatrix} n_1 \\ n_2 \end{bmatrix}}_{\mathbf{n}} . \tag{2.60}$$

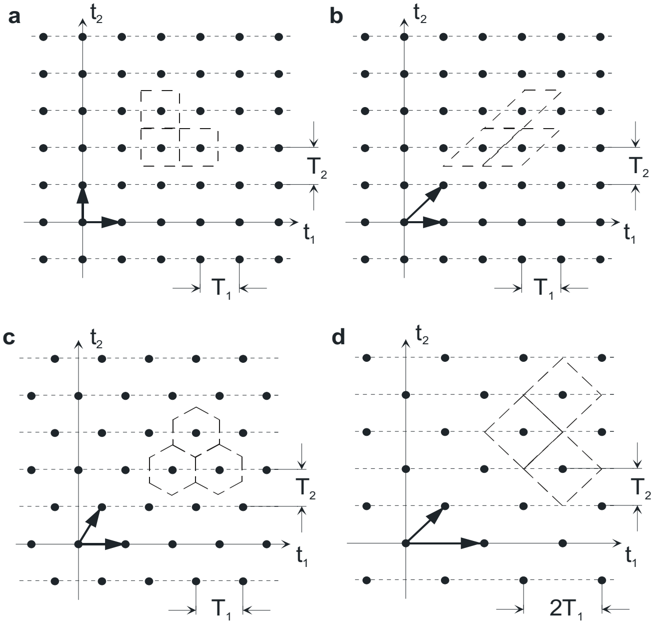

Fig. 2.11. 2D sampling grids: **a** rectangular **b** horizontal shear, v=1 **c** hexagonal **d** quincunx

For the separable case, the sampling distances T_1 in horizontal and T_2 in vertical direction are independent of each other. The corresponding sampling matrix is diagonal, with a frequency matrix according to (2.37),

$$\mathbf{T}_{\text{rect}} = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} \Rightarrow \mathbf{F}_{\text{rect}} = \begin{bmatrix} \frac{1}{T_1} & 0 \\ 0 & \frac{1}{T_2} \end{bmatrix}. \tag{2.61}$$

For the case of *shear sampling* (horizontal or vertical shear alternatively)

$$\mathbf{T}_{\text{shear}} = \begin{bmatrix} T_1 & v \cdot T_1 \,|\, 0 \\ 0 \,|\, v \cdot T_2 & T_2 \end{bmatrix} \Rightarrow \mathbf{F}_{\text{shear}} = \begin{bmatrix} \frac{1}{T_1} & 0 \,|\, -\frac{v}{T_1} \\ -\frac{v}{T_2} \,|\, 0 & \frac{1}{T_2} \end{bmatrix}, \tag{2.62}$$

the effective sampling grid would still appear as rectangular when v is an integer value (see Fig. 2.11b). Shear sampling can be interpreted as an alternative approach of adapting the sampling process by directional signal characteristics, where one axis of the coordinate system is tilted by the propagation direction of the signal. Such an approach may be useful when in a multidimensional sampling process the sampling positions in some dimensions cannot be changed due to system restrictions, e.g. when an image is scanned line-wise, or with fixed temporal sampling positions in case of video sampling.

Two other cases which can be interpreted as special cases of shear sampling (using non-integer shear factors v) are the hexagonal sampling scheme (Fig. 2.11c) and the quincunx sampling scheme (Fig. 2.11d). The basis vectors are tilted such that each sample has same distances towards its six or four nearest neighbors, respectively. To achieve this, a common scaling of sampling distance T (equal to the vertical distance between lines in these two cases) is used for both basis vectors.

$$\mathbf{T}_{\text{hex}} = T \cdot \begin{bmatrix} \frac{2}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 1 \end{bmatrix} \Rightarrow \mathbf{F}_{\text{hex}} = \frac{1}{T} \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 \\ -\frac{1}{2} & 1 \end{bmatrix};$$

$$\mathbf{T}_{\text{quin}} = T \cdot \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \Rightarrow \mathbf{F}_{\text{quin}} = \frac{1}{T} \begin{bmatrix} \frac{1}{2} & 0 \\ -\frac{1}{2} & 1 \end{bmatrix}. \tag{2.63}$$

To determine the positions of periodic spectral copies in the case of non-separable sampling, a non-separable 2D Dirac impulse grid with sampling positions defined by $\mathbf{T}$ is mapped by a coordinate transformation into a separable, unity-distance

Dirac impulse grid[8] $Ш(\mathbf{t}) \equiv \delta_\mathbf{I}(\mathbf{t}) \circ\!\!-\!\!\bullet\, \delta_\mathbf{I}(\mathbf{f}) \equiv Ш(\mathbf{f})$, with T_1=T_2=...=1. This gives, using (2.38):

$$Ш(\mathbf{T}^{-1}\mathbf{t}) \circ\!\!-\!\!\bullet\, \left|\mathbf{T}\right| Ш(\mathbf{F}^{-1}\mathbf{f}) \;\; \text{with } \mathbf{F} = \left[\mathbf{T}^{-1}\right]^\mathrm{T} \tag{2.64}$$

It should however be observed, that by applying the coordinate transformation to the ‘sheh’ function, the Dirac impulses are scaled reciprocally, following the determinant of the respective coordinate transformation matrix (in both **t** and **f** domains). Therefore, explicitly expressed by sums of non-scaled Dirac impulses,

$$Ш(\mathbf{T}^{-1}\mathbf{t}) = \sum_\mathbf{n} \left|\mathbf{T}\right| \delta(\mathbf{t} - \mathbf{Tn}) \quad \text{and} \quad Ш(\mathbf{T}^{-1}\mathbf{f}) = \sum_\mathbf{k} \left|\mathbf{F}\right| \delta(\mathbf{f} - \mathbf{Fk}) \,, \tag{2.65}$$

and finally

$$\underbrace{\sum_\mathbf{n} \delta(\mathbf{t} - \mathbf{Tn})}_{\delta_\mathbf{T}(\mathbf{t})} \circ\!\!-\!\!\bullet\, \frac{1}{\left|\mathbf{T}\right|} \underbrace{\sum_\mathbf{k} \delta(\mathbf{f} - \mathbf{Fk})}_{\delta_\mathbf{F}(\mathbf{f})} \tag{2.66}$$

The spectrum of a multi-dimensional signal being ideally sampled with the scheme defined by the sampling matrix **T** then is

$$s_{\delta_\mathbf{T}}(\mathbf{t}) = s(\mathbf{t}) \cdot \delta_\mathbf{T}(\mathbf{t}) \circ\!\!-\!\!\bullet\, S_{\delta_\mathbf{T}}(\mathbf{f}) = S(\mathbf{f}) * \frac{1}{\left|\mathbf{T}\right|} \delta_\mathbf{F}(\mathbf{f}) = \frac{1}{\left|\mathbf{T}\right|} \sum_\mathbf{k} S(\mathbf{f} - \mathbf{Fk}) \,, \tag{2.67}$$

which gives specifically for the 2-dimensional case

$$S_{\delta_\mathbf{T}}(f_1, f_2) = \frac{1}{\left|\mathbf{T}\right|} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} S(f_1 - k_1 f_{11} - k_2 f_{12}, f_2 - k_1 f_{21} - k_2 f_{22}) \,. \tag{2.68}$$

(2.67) and (2.68) can be interpreted in a way that each κ-tuple of integer values in **k** points to one copy of the spectrum by the corresponding linear combination of the basis vectors, **Fk.** Again, direct computation of S_δ(**f**) as in (2.56) would be possible from the series of samples,

$$S_{\delta_\mathbf{T}}(\mathbf{f}) = \sum_\mathbf{n} s(\mathbf{Tn}) \mathrm{e}^{-\mathrm{j}2\pi \mathbf{f}^\mathrm{T} \mathbf{Tn}} = \sum_\mathbf{n} s(\mathbf{Tn}) \mathrm{e}^{-\mathrm{j}2\pi \left[\mathbf{F}^{-1}\mathbf{f}\right]^\mathrm{T} \mathbf{n}} \,. \tag{2.69}$$

$\mathbf{F}^{-1}\mathbf{f}$ in (2.64) could be interpreted as normalized frequency where spectral copies are at integer vector positions **k**, and $\mathbf{n}=\mathbf{T}^{-1}\mathbf{t}$ would describe a discrete signal over integer vector indices **n**, corresponding to a normalization of the **t** coordinates by **T**. Then, a separable Fourier sum over a normalized frequency could be computed directly from the signal samples s(**n**), regardless of the actual sampling structure, as a generalization of (2.57),

$$S_\delta(\mathbf{f}) = \sum_\mathbf{n} s(\mathbf{n}) \mathrm{e}^{-\mathrm{j}2\pi \mathbf{f}^\mathrm{T} \mathbf{n}} \,. \tag{2.70}$$

[8] Using the letter ‘sheh‘ from the Russian alphabet as a symbolic expression of a unity distance Dirac impulse grid

It must however be noted that the normalization assumed in (2.70) may be misleading because reasonable conditions about band limitation, as necessary for alias-free sampling and reconstruction, are not fully reflected here. A mapping of conditions in (2.58) would unnecessarily restrict the degrees of freedom in defining band limitation in the non-rectangular sampling case, because the matrix **F** only allows describing a linear coordinate transformation of the baseband boundaries from a square-shaped lowpass. Two examples of such a mapping for the case of quincunx sampling (2.63) are shown in Fig. 2.12. The resulting limitation of the base band in Fig. 2.12b would be asymmetric, giving different preference to orientations. Moreover, other grids with identical sampling points (though differently indexed in **k**) can be defined using alternative sampling matrices **T**. For example, two possible definitions of a position-wise identical quincunx grid as in (2.63) would be (case II is based on a vertical shear, case III is a rotation of coordinates)

$$\mathbf{T}_{\text{quin-II}} = T\begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix} \quad ; \quad \mathbf{T}_{\text{quin-III}} = T\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{2.71}$$

These different definitions would have a significant impact on the band limitation, when back-projection from the unity grid (2.64) is used as reference. In the specific case of quincunx, the rotation would give the somewhat optimum omnidirectional packing as discussed below, but it may not be possible to make such definition from the sampling matrix for any sampling structure, particularly in higher dimensions.

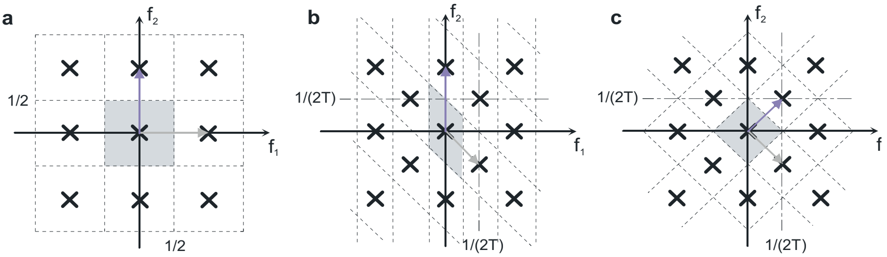

Fig. 2.12. **a** Base band and its periodic copies for rectangular-grid sampling in the normalized frequency plane **b/c** corresponding reverse mapping $\mathbf{F}^{-1}\mathbf{f}$ for the case of quincunx sampling according to (2.63) (**b**) and version III from (2.71) (**c**).

Best omnidirectional lowpass band limitation for non-rectangular sampling can be derived from the theory of dense packing of identically shaped cells (areas or volumes) in multiple dimensions. For this, the position of the center of the base band at the origin of the frequency plane is regarded in relation to the positions of centers of directly neighbored spectral copies. In order to make the shapes identical and symmetric for different directional orientations, the cut-off frequency should be at half distance between the zero frequency and the centers of closest spectral copies. This can be determined by drawing interconnection lines (called

Delaunay lines) from $\mathbf{f}=\mathbf{0}$ to those center points. In the 2D case, *Voronoi lines* (which would become *planes* or *hyper planes* in higher dimensions) are intersecting at the mid position of the respective Delaunay line with perpendicular orientation (i.e. the orientation of the Delaunay line could be interpreted as normal vector of the Voronoi boundary). The connection of all Voronoi lines closest to frequency zero establishes the boundary of the base band. For the examples of quincunx and hexagonal sampling matrices in (2.63), the shapes of base bands and the periodic copies thereof are illustrated in Fig. 2.13a/b, and conditions are explicitly determined in the following paragraphs. In the quincunx case, this becomes indeed identical to the mapping of conditions (2.58) by $\mathbf{F}_{\text{quin-III}}$ of (2.71); in the hexagonal case, no such direct mapping is possible.

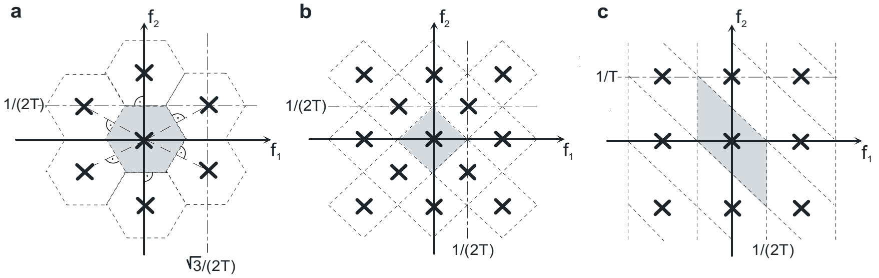

Fig. 2.13. Positions of base band and spectral copies for different 2D sampling grids. **a** Hexagonal **b** Quincunx **c** Shear, $v = 1$ [– – – Delaunay line - - - - - Voronoi line]

Hexagonal sampling. The hexagonal shape of the base band requires piecewise definition, but is symmetric over all four quadrants. When $|f_1| \leq \sqrt{3}/6T$, the boundary of the base band is parallel with the f_1 axis, while for higher frequencies $|f_1|$, four lines with slopes $a = \pm\sqrt{3}$ and intercepts $b = \pm 1/T$ define the boundary. This results in sampling conditions (see problem 2.1)

$$S(f_1,f_2)\overset{!}{=}0 \quad \text{for} \quad |f_2| \geq \frac{1}{2T} \quad \text{or} \quad |f_1| + \frac{|f_2|}{\sqrt{3}} \geq \frac{1}{\sqrt{3}T}. \tag{2.72}$$

Quincunx sampling. The boundary of the base band is described by four lines of slopes $a=\pm 1$ and intercepts $b=\pm 1/(2T)$. This gives the condition

$$S(f_1,f_2)\overset{!}{=}0 \quad \text{for} \quad |f_1| + |f_2| \geq \frac{1}{2T}. \tag{2.73}$$

In quincunx sampling, pure horizontal or vertical sinusoids can be reconstructed up to the same frequency as with quadrangular (i.e. rectangular where $T=T_1=T_2$) sampling, though the number of samples is reduced by a factor of two. For sinusoids of diagonal orientation, the maximum allowable frequency is however lower by a factor $\sqrt{2}$. Quincunx sampling better matches human perception which is less sensitive to fine detail in the diagonal directions. In the Bayer pattern (Fig. 1.5), a quincunx sampling grid is therefore applied to the G (green) component. For interpolation into full resolution, a 2D sinc function rotated by 45° (or an approximation thereof) can be applied:

$$h(t_1,t_2)=\mathrm{si}\left(\pi\frac{t_1-t_2}{4T}\right)\mathrm{si}\left(\pi\frac{t_1+t_2}{4T}\right). \tag{2.74}$$

For any two- and multi-dimensional sampling system, the allowable bandwidth of the signal (area or volume covered by the base bands in Fig. 2.13) is identical to the determinant of the frequency sampling matrix **F**. Likewise, the area or volume of each 'sampling cell' is the determinant of the sampling matrix **T**. Due to (2.37), the density of samples and the alias-free signal bandwidth are mutually reciprocal. The definition of the base band allows certain degrees of freedom, in trading the resolution ranges between the different dimensions. As an example for this, alias-free quincunx sampling could also be realized using a separable reconstruction filter of horizontal pass-band cut-off $\pm 1/(4T)$, vertical cut-off $\pm 1/(2T)$ or vice versa. The question whether this makes sense can only be answered by an analysis of signal characteristics, and by the actual goal of sampling, e.g. the effective bandwidth of signals along each of the dimensions.

The theory of densest packing as mentioned above can not only be used for determining the boundaries of the baseband (respectively the cut-off characteristics of the lowpass interpolation filter), but also to determine the best two- or multi-dimensional sampling grid. Assume that a goal would be to represent directional sinusoids such as (2.1),(2.3) with highest possible frequency F regardless of the orientation. From that point of view, the optimum shape of the baseband would be a circle, in higher dimensions, it becomes a sphere or hyper sphere. If a circle or sphere of given radius r (e.g. $r=1/2$) is fitted with the minimum baseband cut-off, the determinant $|\mathbf{F}|$ of the related matrix is a criterion for the necessary number of samples per unit to allow a cut-off at $f=1/2$ at minimum. For example, in the case of separable 2D sampling (2.61) (see Fig. 2.12a), dense packing of circles with $r=1/2$ is possible when $T_1=T_2=1$, $|\mathbf{F}|=1$. In hexagonal sampling (2.63) (see Fig. 2.12c), this is possible using $T=1$, which gives $|\mathbf{F}|=\sqrt{3}/2\approx 0.866$. This is denoted as *sphere packing advantage* of the hexagonal structure, meaning that sinusoids of arbitrary directional orientation with a given maximum frequency can be sampled using less than 87% of the samples that would be necessary for the separable case. Alternatively, using the same number of samples, the cut-off frequency can be increased by the reciprocal square root of that factor. In 2D, the hexagonal scheme provides the best possible sampling in that sense. The quincunx scheme does not provide a sphere packing advantage.

Two- and multi-dimensional sampling structures can also be constructed by superimposing different systems of basis vectors. For example, a quincunx scheme as in Fig. 2.11c can be interpreted as a superposition of two rectangular schemes (see problem 2.2). Similarly, a grid of equal-sized triangle cells can be formed by a superposition of two hexagonal grids of Fig. 2.11c, where the second is vertically offset by $2T_2/3$. However in this case, the cells corresponding to the two sub-grids have different orientation, and each point has only three nearest neighbors with equal distance, which indicates that the packing would be less dense than with a single hexagonal grid.

2.3.3 Sampling of video signals

A video sequence of pictures can be interpreted as a three-dimensional (2D spatial+temporal) signal (see Fig. 2.14). Let the time distance between subsequent sampled pictures be T_3. An extension of separable sampling (2.60) to the third dimension then leads to the following mapping of sampling positions in the spatio-temporal continuum:

$$\begin{bmatrix} t_1(n_1,n_2,n_3) \\ t_2(n_1,n_2,n_3) \\ t_3(n_1,n_2,n_3) \end{bmatrix} = \begin{bmatrix} n_1T_1 \\ n_2T_2 \\ n_3T_3 \end{bmatrix} = \mathbf{T}_{\text{prog}} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}. \tag{2.75}$$

For the example of Fig. 2.14a, samples have identical spatial positions in any picture. Such a configuration is denoted as *progressive sampling*, which is shown in Fig. 2.14b over the vertical and temporal directions. The sampling matrix related to fully-separable progressive sampling is given as

$$\mathbf{T}_{\text{prog}} = \begin{bmatrix} T_1 & 0 & 0 \\ 0 & T_2 & 0 \\ 0 & 0 & T_3 \end{bmatrix} \Rightarrow \mathbf{F}_{\text{prog}} = \begin{bmatrix} 1/T_1 & 0 & 0 \\ 0 & 1/T_2 & 0 \\ 0 & 0 & 1/T_3 \end{bmatrix}, \tag{2.76}$$

however, any sampling that handles the temporal dimension independent of the two spatial dimensions could also be entitled as progressive (e.g. quincunx or hexagonal only in the two spatial dimensions). In analog video, *interlaced sampling* was typically used, and interlaced formats still exist in some digital video cameras. Even and odd lines are sampled in a time-interleaved fashion, such that for each time instance, only half of the lines is sampled and available for subsequent processing. The resulting pictures consisting of either even or odd lines are called the even and odd *fields*, respectively (see. Fig. 2.14c). The sampling matrix in this case can be defined as[9]

[9] In Fig. 2.14c and in the sampling matrix (2.77) a configuration is shown where the top field (lines 0,2,4,..) is the field which is sampled first within the frame. In NTSC TV and digital 60 Hz interlaced video derived thereof, the bottom field is sampled first. This is however only relevant if field pictures are grouped together as a ‘frame’, e.g. when absolute timing information is assigned.

$$\mathbf{T}_{\text{inter}} = \begin{bmatrix} T_1 & 0 & 0 \\ 0 & 2T_2 & T_2 \\ 0 & 0 & T_3 \end{bmatrix} \Rightarrow \mathbf{F}_{\text{inter}} = \begin{bmatrix} 1/T_1 & 0 & 0 \\ 0 & 1/(2T_2) & 0 \\ 0 & -1/(2T_3) & 1/T_3 \end{bmatrix}. \tag{2.77}$$

This could be interpreted as a quincunx sampling grid[10] applied to the vertical/temporal continuum in 3D. By this, higher vertical frequencies can be supported only when no significant temporal changes (e.g. caused by motion) are present.

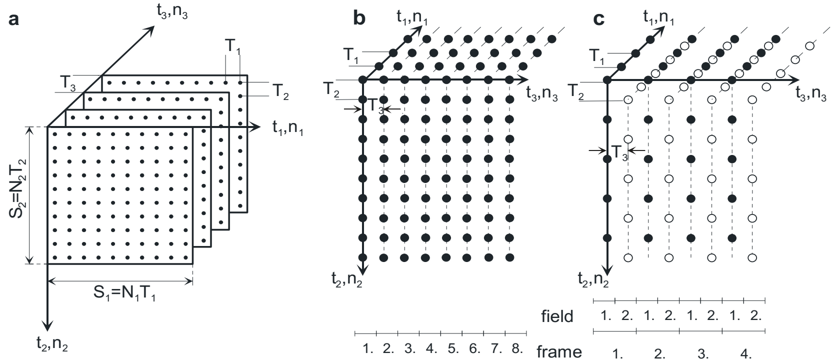

Fig. 2.14. **a** Progressively sampled image sequence **b**/**c** Video sampling in vertical/temporal directions: Progressive (**b**) and interlaced (**c**) schemes.

In progressive sampling – which is the 3D version of separable sampling – the conditions of the sampling theorem for avoiding alias can be formulated independently in each dimension. In this case the sampling matrix is diagonal, such that no interrelationships occur:

$$S(f_1,f_2,f_3)\overset{!}{=}0 \quad \text{when } |f_1| \geq \frac{1}{2T_1} \text{ or } |f_2| \geq \frac{1}{2T_2} \text{ or } |f_3| \geq \frac{1}{2T_3}. \tag{2.78}$$

In interlaced sampling, only the condition for the first dimension can be separated, since the horizontal sampling positions are independent,

$$S(f_1,f_2,f_3)\overset{!}{=}0 \quad \text{when} \quad |f_1| \geq \frac{1}{2T_1} \quad \text{or} \quad \frac{|f_2|}{T_3} + \frac{|f_3|}{T_2} \geq \frac{1}{2T_2T_3}. \tag{2.79}$$

In video acquisition, spatial sampling is often assumed to be alias free, as the elements of the acquisition system (lenses etc.) naturally have a lowpass effect. As was shown in (2.45), the frequency f_3 depends on spatial frequency and the strength of motion. Assume that the signal could contain sinusoids of almost the

[10] The bottom-right 2x2 sub-matrix in (2.77) is indeed similar to (2.63) except for the fact that T_2 and T_3 actually express different physical units (space and time). Therefore, setting T_1=T_2 (as it was done in the 2D case) is not meaningful here.

maximum allowed spatial frequencies ($F_1 \approx 1/(2T_1)$, $F_2 \approx 1/(2T_2)$). Substituting the condition for f_3 from (2.78) into (2.45), the following limiting condition must then be imposed on the velocity to achieve alias-free sampling:

$$|u_1| \cdot \frac{T_3}{T_1} + |u_2| \cdot \frac{T_3}{T_2} \overset{!}{<} 1 \text{ resp. } \quad |k_1| + |k_2| \overset{!}{<} 1 \quad \text{with} \quad k_i = u_i \cdot \frac{T_3}{T_i}. \tag{2.80}$$

Herein, the k_i express the horizontal/vertical displacements in units of samples from one picture to the next, if the velocity u_i is observed in the continuous signal in the respective dimension. The strict limitation, disallowing shifts larger than one spatial sample per time unit, appears surprising at first sight, as humans usually are capable to watch moving pictures of much higher motion without any problem. However, the limitation in (2.80) assumes that only *one* sinusoid of close-to-highest allowable spatial frequency would be sampled. Spectra of natural video signals are non-sparse with high energy in low-frequency ranges, which allow perceiving the motion reliably and alias-free. Particularly, the observer's eyes can track the motion which compensates alias by projecting the spectrum towards frequency f_3=0 (or alternatively could be interpreted as using a shear of the reconstruction filter pass-band).

To illustrate the effects of alias occurring in the case of progressive sampling, Fig. 2.15 shows a vertical/temporal section (f_2,f_3) of the 3D frequency space. A spatial sinusoid of close to half vertical sampling frequency is assumed which has a spectrum consisting of two Dirac impulses (⦿). Centers of periodic spectral copies are marked by '**x**'. Fig. 2.15a shows the spectrum of the signal without motion. Fig. 2.15b indicates skewing of the position in direction of f_3, when the signal is moved by half a special sampling unit per time unit ($u_2 = -0.5T_2/T_3$) upwards, Fig. 2.15c illustrates the case of motion by 1.5 units ($u_2 = -1.5T_2/T_3$). In the latter case, alias components appear in the base band, such that a viewer could interpret this as a motion by half a unit downwards ($u_2 = 0.5T_2/T_3$). The spatial frequency of the signal remains unchanged in any case, i.e. aliasing in f_3 only causes wrong interpretation of motion in the case of progressive sampling. In cinema, this is known as the 'stage coach effect', where rotating wheels equipped with periodic spokes seem to move slower, stand still or turn backwards, depending on the combined effect of temporal sampling distance, angular distance between the spokes and the speed of the wheel.

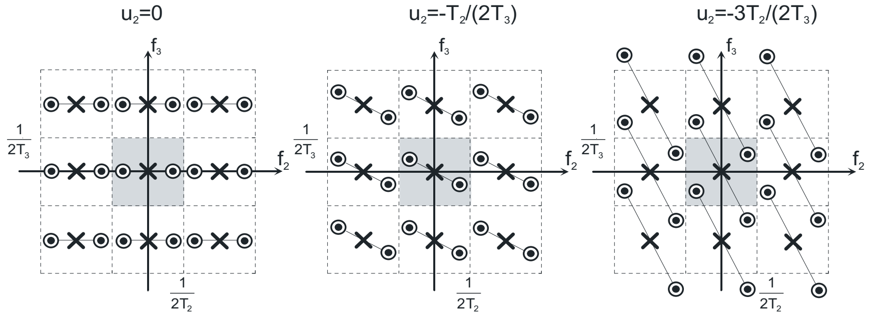

Fig. 2.15. Effect of alias, vertical motion of a progressively-sampled sinusoid.

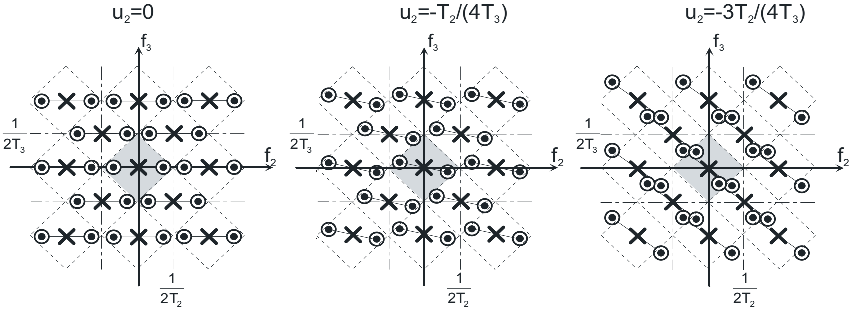

Fig. 2.16. Effect of alias, vertical motion of an interlaced-sampled sinusoid.

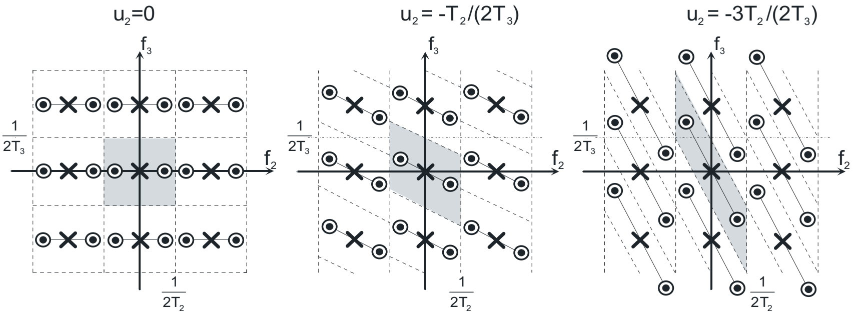

Fig. 2.17. Avoidance of alias by adaptation of the human visual system; tracking by the eyes effects correct reconstruction in sheared sampling.

Fig. 2.16 shows the effect for the case of interlaced sampling of the same signal[11]. First, it is obvious that aliasing already occurs with lower motion than in the pro-

[11] Note that the temporal sampling distance T_3 according to the sampling matrix in (2.77) refers to frame units, i.e. the sampling distance between fields is $T_3/2$. Likewise, the vertical sampling distance between the adjacent lines of each field is $2T_2$ by this definition.

gressive sampling case. Second, if alias spectra originate from diagonally-adjacent spectral copies and with vertical frequency of the sinusoid as F_2, an alias component of frequency $\tilde{F}_2 = 1/(2T_2) - F_2$ appears in the base band. In particular when highly-detailed periodic stripes are present in the scene and moving, this can result in appearance of strange sinusoidal components, typically also having different orientations than the original, as the horizontal frequency component F_1 would not be affected and orientation follows from (2.2).

As motion causes a tilt of spectra towards positions $f_3 \neq 0$, but does not cause a spreading of spectra, perfect reconstruction and correct perception would in principle be possible when the motion is known to the observer. This can either be interpreted to relate to the case of shear sampling (where the spectral shape of the reconstruction filter is aligned towards $f_3 = u_1 f_1 + u_2 f_2$), or as motion compensation (where the observer 'transforms' the reference coordinate system according to the motion). Fig. 2.17 illustrates that a single sinusoid moving by higher velocity can still be interpreted correctly; however, from a *single* sinusoid it is usually not possible to determine the actual motion, as the signal is periodic and multiple correspondences are detected between the subsequent pictures (a typical observer would assume the lowest possible velocity, which means that the displacement should not be larger than half a period in any direction). However, for structured signals which contain salient points, edges etc., the true motion can be tracked accordingly, as all frequency components are identically shifted (consistent linear phase shift). Motion-compensated processing in video compression performs a similar task, allowing to compress signals based on their actual redundancy along the temporal axis, thus avoiding alias components.

2.4 Discrete signal processing

2.4.1 LSI systems

The one- or multidimensional operation[12]

$$g(\mathbf{n}) = \sum_{\mathbf{m} \in \mathbf{Z}_\kappa} s(\mathbf{m}) h(\mathbf{n} - \mathbf{m}) = s(\mathbf{n}) * h(\mathbf{n}) \tag{2.81}$$

is denoted as *discrete convolution*. Its properties are similar to the continuous-time convolution integral, e.g. the associative, commutative and distributive properties apply. The *unit impulse*

[12] The Z-lattice $\mathbf{Z}_\kappa$ is an infinite set of vectors consisting of all possible integer number combinations over κ dimensions.

$$\delta(\mathbf{n}) = \begin{cases} 1 & \text{für } \mathbf{n} = \mathbf{0} \\ 0 & \text{für } \mathbf{n} \neq \mathbf{0}, \end{cases} \tag{2.82}$$

also denoted as *Kronecker impulse*, is the unity element,

$$s(\mathbf{n}) = \delta(\mathbf{n}) * s(\mathbf{n}) = \sum_{\mathbf{m}\in\mathbf{Z}_\kappa} s(\mathbf{m})\delta(\mathbf{n}-\mathbf{m}) . \tag{2.83}$$

Discrete convolution (2.81) is *linear* (2.14) and *shift invariant*, the latter property being equivalent with time invariance (2.15). Therefore, a system performing the discrete convolution operation is denoted as LSI system, for which (2.81) provides the unique mapping between input and output, with behaviour fully described by the impulse response $h(\mathbf{n})$. The operation of certain classes of LSI systems can be interpreted by finite order *difference equations*, for which a causal form[13] is

$$\sum_{\mathbf{p}\in\mathcal{N}_\mathbf{p}^{0+}} \tilde{b}_\mathbf{p} g(\mathbf{n}-\mathbf{p}) = \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{q}^{0+}} \tilde{a}_\mathbf{q} s(\mathbf{n}-\mathbf{q}). \tag{2.84}$$

This gives the input/output relation (simplified when normalizing $\tilde{b}_\mathbf{0} = 1$)

$$g(\mathbf{n}) = \underbrace{\sum_{\mathbf{q}\in\mathcal{N}_\mathbf{q}^{0+}} a_\mathbf{q} s(\mathbf{n}-\mathbf{q})}_{\text{FIR part}} + \underbrace{\sum_{\mathbf{p}\in\mathcal{N}_\mathbf{p}^{+}} b_\mathbf{p} g(\mathbf{n}-\mathbf{p})}_{\text{IIR part}} \quad \text{with } a_\mathbf{q} = -\frac{\tilde{a}_\mathbf{q}}{\tilde{b}_\mathbf{0}},\ b_\mathbf{p} = -\frac{\tilde{b}_\mathbf{p}}{\tilde{b}_\mathbf{0}}. \tag{2.85}$$

The corresponding digital filters consist of an FIR (*Finite Impulse Response*) part taking reference to $|\mathcal{N}_\mathbf{q}^+|$ previous samples of the input, and an IIR (*Infinite Impulse Response*) part using feedback from $|\mathcal{N}_\mathbf{p}^+|$ previously processed output samples.

2.4.2 Discrete Fourier transform

Similar to (2.47), a spectrum $S(\mathbf{f})$ shall be represented by samples which have distances that are expressed by a separable (diagonal) sampling matrix **F** on the frequency axis[14]:

$$S_\mathrm{p}(\mathbf{f}) = \sum_{\mathbf{k}\in\mathbf{Z}_\kappa} S\left(\mathbf{Fk}\right)\delta(\mathbf{f}-\mathbf{Fk}) = S(\mathbf{f})\sum_{\mathbf{k}\in\mathbf{Z}_\kappa} \delta(\mathbf{f}-\mathbf{Fk}) . \tag{2.86}$$

Applying the inverse Fourier transform gives

[13] Herein, $\mathcal{N}^{0+}$ is a finite set of integer index vectors $\mathbf{p}|\mathbf{q}$ corresponding to a neighbourhood of previously available input samples, including the current sample with $\mathbf{p}|\mathbf{q}=\mathbf{0}$. For example, in 1D, the range of values is q=0…Q. Similarly, $\mathcal{N}^+$ is excluding the current sample, e.g. in 1D, with range of values p=1…P

[14] In principle, the following considerations are extensible to non-separable spectrum sampling, which for simplicity is omitted here.

$$\begin{array}{ccccc} S_\mathrm{p}(\mathbf{f}) & = & S(\mathbf{f}) & \cdot & \sum\limits_{\mathbf{k}\in\mathbf{Z}_\kappa}\delta(\mathbf{f}-\mathbf{F}\mathbf{k}) \\ \substack{\bullet\\|\\\circ} & & \substack{\bullet\\|\\\circ} & \substack{\bullet\\|\\\circ} & \substack{\bullet\\|\\\circ} \\ s_\mathrm{p}(\mathbf{t}) & = & s(\mathbf{t}) & * & \frac{1}{|\mathbf{F}|}\sum\limits_{\mathbf{n}\in\mathbf{Z}_\kappa}\delta(\mathbf{t}-\mathbf{T}\mathbf{n}) \end{array} \quad \text{with } \mathbf{T}=\left[\mathbf{F}^{-1}\right]^\mathrm{T}. \tag{2.87}$$

Spectrum sampling described by **F** effects a periodic repetition of the **t**-domain function described by **T**. If the duration of $s(\mathbf{t})$ fits into one ‘periodic cell’ of **T**, it can be reconstructed from $s_\mathrm{p}(\mathbf{t})$ by multiplying it with a separable rectangular window function that has the shape of the cell, which in the frequency domain corresponds to a separable sinc function:

$$\begin{array}{ccccc} s(\mathbf{t}) & = & s_\mathrm{p}(\mathbf{t}) & \cdot & |\mathbf{F}|\,\mathrm{rect}(\mathbf{T}\mathbf{t}) \\ \substack{\circ\\|\\\bullet} & & \substack{\circ\\|\\\bullet} & \substack{\circ\\|\\\bullet} & \substack{\circ\\|\\\bullet} \\ S(\mathbf{f}) & = & S_\mathrm{p}(\mathbf{f}) & * & \mathrm{si}(\pi\mathbf{F}\mathbf{f}). \end{array} \tag{2.88}$$

From these considerations, periodic signals possess discrete spectra, but also signals that are time limited to a range that is equivalent to one period of **T** are completely represented by spectral samples over **F**.

As band-limited signals can be described from a series of samples over time, it can further be concluded that a signal which is considered as limited and could therefore be equivalently periodic *in both time and frequency domains* can also be perfectly represented by *finite series of samples* in any of the two domains. A signal $s_\mathrm{d}(\mathbf{n})$ shall be nonzero only in ranges $[0;M_i-1]$ within all of its κ dimensions ($i=1\ldots\kappa$), or equivalently be periodic over M_i samples. Then, samples of the periodic Fourier spectrum taken at distances $F_i=1/M_i$ are giving a unique representation, where in the two subsequent equations $\mathbf{F}=[\mathbf{M}^{-1}]^\mathrm{T}$ is a diagonal matrix with the F_i values of the different dimensions as entries (and **M** similarly holding the M_i values). This gives the *Discrete Fourier Transform* (DFT) over κ dimensions,

$$S_\mathrm{a}(\mathbf{F}\mathbf{k})=S_\mathrm{d}(\mathbf{k})=\sum_{n_1=0}^{M_1-1}\cdots\sum_{n_\kappa=0}^{M_\kappa-1}s_\mathrm{d}(\mathbf{n})\mathrm{e}^{-\mathrm{j}2\pi\mathbf{n}^\mathrm{T}\mathbf{F}\mathbf{k}};\quad k_i=0,\ldots,M_i-1, \tag{2.89}$$

with the inverse DFT allowing reconstruction of all $|\mathbf{M}|$ samples,

$$s_\mathrm{d}(\mathbf{n})=\sum_{n_1=0}^{M_1-1}\cdots\sum_{n_\kappa=0}^{M_\kappa-1}S_\mathrm{d}(\mathbf{k})\mathrm{e}^{\mathrm{j}2\pi\mathbf{n}^\mathrm{T}\mathbf{F}\mathbf{k}};\quad n_i=0,\ldots,M_i-1. \tag{2.90}$$

2.4.3 *z* transform

A condition for existence of the Fourier sum (2.57) of a discrete-time signal is finite absolute summation

$$\sum_{\mathbf{n}\in\mathbf{Z}_\kappa}|s(\mathbf{n})|<\infty\,. \tag{2.91}$$

An exception is established for periodic signals which have Fourier spectra $S_\delta(f)$

containing Dirac impulses. Otherwise, for a larger class of signals that do not grow stronger than exponentially on at most one side, convergence can be achieved by an exponential weighting $\mathrm{e}^{-\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{n}} = \mathrm{e}^{-\sigma_1 n_1}\cdots\mathrm{e}^{-\sigma_\kappa n_\kappa}$ (σ_i values real),

$$e^{-\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{n}}s(\mathbf{n}) \circ\!\!-\!\!\bullet \sum_{\mathbf{n}\in\mathbf{Z}_\kappa}\left(s(\mathbf{n})\mathrm{e}^{-\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{n}}\right)\mathrm{e}^{-\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{n}} = \sum_{\mathbf{n}\in\mathbf{Z}_\kappa} s(\mathbf{n})\mathrm{e}^{-(\boldsymbol{\sigma}+\mathrm{j}2\pi\mathbf{f})^{\mathrm{T}}\mathbf{n}}. \tag{2.92}$$

Substituting z_i=$\mathrm{e}^{(\sigma_i+\mathrm{j}2\pi f_i)}$ by polar coordinates z_i=$\rho_i\mathrm{e}^{\mathrm{j}2\pi f_i}$ with ρ_i=$\mathrm{e}^{\sigma_i} \geq 0$ (ρ_i >0 and σ_i real valued, $\rho_i \to 0$ for $\sigma_i \to -\infty$) and defining

$${}^{\kappa}\mathbf{z}^{(\mathbf{l})} = \prod_{i=1}^{\kappa} z_i^{l_i}, \tag{2.93}$$

the two-sided κ-dimensional *z*-transform of the signal $s(\mathbf{n})$ is

$$S(\mathbf{z}) = \sum_{\mathbf{n}\in\mathbf{Z}_\kappa} s(\mathbf{n})\,{}^{\kappa}\mathbf{z}^{(-\mathbf{n})}. \tag{2.94}$$

Values of $\mathbf{z}$ where a solution exists are contained within the *region of convergence* (RoC) of the complex *z* hyperspace. The *z*-transform is particularly useful in LSI system analysis and synthesis. Convolution in the time domain can again be expressed by multiplication in the *z* domain,

$$\begin{aligned} g(\mathbf{n}) = s(\mathbf{n}) * h(\mathbf{n}) \overset{z}{\circ\!\!-\!\!\bullet} G(\mathbf{z}) = S(\mathbf{z})\cdot H(\mathbf{z}) \\ \text{with } \mathrm{RoC}\{G\} = \mathrm{RoC}\{S\} \cap \mathrm{RoC}\{H\}, \end{aligned} \tag{2.95}$$

and a delay by $\mathbf{k}$ samples can be expressed as

$$s(\mathbf{n}-\mathbf{k}) = s(\mathbf{n}) * \delta(\mathbf{n}-\mathbf{k}) \overset{z}{\circ\!\!-\!\!\bullet} S(\mathbf{z})\cdot \sum_{\mathbf{n}\in\mathbf{Z}_\kappa} \delta(\mathbf{n}-\mathbf{k})\,{}^{\kappa}\mathbf{z}^{(-\mathbf{n})} = S(\mathbf{z})\,{}^{\kappa}\mathbf{z}^{(-\mathbf{k})}. \tag{2.96}$$

A causal FIR/IIR filter with difference equation (2.84), where the *z*-transform is separately applied to the left and right sides, gives

$$\begin{aligned} &\sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{q}}^{0+}} a_{\mathbf{q}} s(\mathbf{n}-\mathbf{q}) \overset{z}{\circ\!\!-\!\!\bullet} S(\mathbf{z})\cdot A(\mathbf{z}) \;\text{ with }\; A(\mathbf{z}) = \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{q}}^{0+}} a_{\mathbf{q}}\,{}^{\kappa}\mathbf{z}^{(-\mathbf{q})} \\ &\sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} b_{\mathbf{q}}\cdot g(\mathbf{n}-\mathbf{p}) \overset{z}{\circ\!\!-\!\!\bullet} G(\mathbf{z})\cdot B(\mathbf{z}) \;\text{ with }\; B(\mathbf{z}) = \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} b_{\mathbf{p}}\,{}^{\kappa}\mathbf{z}^{(-\mathbf{p})} \end{aligned} \tag{2.97}$$

and therefore

$$G(\mathbf{z})\cdot\left[1-B(\mathbf{z})\right] = S(\mathbf{z})\cdot A(\mathbf{z}) \quad \Rightarrow H(\mathbf{z}) = \frac{G(\mathbf{z})}{S(\mathbf{z})} = \frac{A(\mathbf{z})}{1-B(\mathbf{z})} = \frac{\displaystyle\sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{q}}^{0+}} a_{\mathbf{q}}\,{}^{\kappa}\mathbf{z}^{(-\mathbf{q})}}{1-\displaystyle\sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} b_{\mathbf{p}}\,{}^{\kappa}\mathbf{z}^{(-\mathbf{p})}}. \tag{2.98}$$

The FIR part of the filter corresponds to the numerator polynomial and the zero locations of the *z* transform, whereas the IIR part relates to the denominator and its singularities (poles). From (2.98) it is straightforward to design an inverse filter which performs *de-convolution*, i.e. reproduces $s(\mathbf{n})$ from $g(\mathbf{n})$,

$$S(\mathbf{z}) = \frac{G(\mathbf{z})}{H(\mathbf{z})} = G(\mathbf{z}) \cdot H^{(-1)}(\mathbf{z})$$

$$\Rightarrow H^{(-1)}(\mathbf{z}) = \frac{S(\mathbf{z})}{G(\mathbf{z})} = \frac{1-B(\mathbf{z})}{A(\mathbf{z})} = \frac{1 - \sum\limits_{\mathbf{p}\in\mathcal{N}_\mathbf{p}^{+}} b_\mathbf{p} \, {}^{\kappa}\mathbf{z}^{(-\mathbf{p})}}{\sum\limits_{\mathbf{q}\in\mathcal{N}_\mathbf{q}^{0+}} a_\mathbf{q} \, {}^{\kappa}\mathbf{z}^{(-\mathbf{q})}} = \frac{\frac{1}{a_0} - \sum\limits_{\mathbf{p}\in\mathcal{N}_\mathbf{p}^{+}} \frac{b_\mathbf{p}}{a_0} \, {}^{\kappa}\mathbf{z}^{(-\mathbf{p})}}{1 - \sum\limits_{\mathbf{q}\in\mathcal{N}_\mathbf{q}^{+}} \frac{a_\mathbf{q}}{a_0} \, {}^{\kappa}\mathbf{z}^{(-\mathbf{q})}} . \tag{2.99}$$

Properties of the multi-dimensional *z* transform. Properties of the multidimensional *z* transform are very similar to those of the Fourier transform:

Linearity:
$$\sum_i a_i s_i(\mathbf{n}) \;\overset{z}{\circ\!\!-\!\!\bullet}\; \sum_i a_i S_i(\mathbf{z}) \tag{2.100}$$

Shift:
$$s(\mathbf{n}-\mathbf{k}) \;\overset{z}{\circ\!\!-\!\!\bullet}\; {}^{\kappa}\mathbf{z}^{(-\mathbf{k})} S(\mathbf{z}) \tag{2.101}$$

Convolution:
$$g(\mathbf{n}) = s(\mathbf{n}) * h(\mathbf{n}) \;\overset{z}{\circ\!\!-\!\!\bullet}\; G(\mathbf{z}) = S(\mathbf{z}) \cdot H(\mathbf{z}) \tag{2.102}$$

Inversion[15]:
$$S(-\mathbf{n}) \;\overset{z}{\circ\!\!-\!\!\bullet}\; S(\mathbf{z}^{(-\mathbf{I})}) \tag{2.103}$$

Scaling[16]:
$$s_{\mathbf{U}\downarrow}(\mathbf{n}) = s(\mathbf{U}\mathbf{n}) \;\overset{z}{\circ\!\!-\!\!\bullet}\; S\left(\mathbf{z}^{(\mathbf{U}^{-1})}\right) \tag{2.104}$$

Expansion:
$$s_{\mathbf{U}\uparrow}(\mathbf{n}) = \begin{cases} s(\mathbf{m}), \; \mathbf{n} = \mathbf{U}\mathbf{m} \\ 0, \; \text{else} \end{cases} \;\overset{z}{\circ\!\!-\!\!\bullet}\; S_{\mathbf{U}\uparrow}(\mathbf{z}) = S(\mathbf{z}^{(\mathbf{U})}) \tag{2.105}$$

Modulation:
$$s(\mathbf{n}) \cdot \mathrm{e}^{\mathrm{j}2\pi\mathbf{F}\mathbf{n}} \;\overset{z}{\circ\!\!-\!\!\bullet}\; S(\mathbf{z}\mathrm{e}^{-\mathrm{j}2\pi\mathbf{F}}) . \tag{2.106}$$

2.4.4 Multi-dimensional LSI systems

The set of samples accessed by a two- and multi-dimensional system is entitled as ‘support region’ or neighborhood $\mathcal{N}$. An interesting class of symmetric 2D support regions is established by a *homogeneous neighborhood*, where signal samples at positions (m_1,m_2) belong to the neighborhood of a sample at position $\mathbf{n}=[n_1\; n_2]^\mathrm{T}$ according to a maximum distance norm of order P[17]:

[15] $\mathbf{z}^{(\mathbf{A})}$ expresses a coordinate mapping in the multi-dimensional $\boldsymbol{z}$ domain such that in the i^{th} dimension $z_i^{(\mathbf{A})}=\Pi z_j^{a_{ji}}$. With $z_i=\mathrm{e}^{\mathrm{j}2\pi f_i}$, the equivalent mapping in the Fourier domain is $\mathbf{Af}$.

[16] Scaling is a sub-sampling operation with integer values $U > 1$. The z transform mapping as expressed in (2.104) is strictly valid when no information loss occurs, i.e. where only samples in $s(n_1,n_2,...)$ which are at positions n_iU_i were non-zero.

[17] Homogeneous neighbourhood systems are *symmetric* in terms of shape, but also in terms of mutual relationship of samples, which means that the current sample at position (n_1,n_2) is

$$\mathcal{N}_C^{(P)}(\mathbf{n}) = \left\{ \mathbf{m} \;:\; 0 < \sum_i \left| m_i - n_i \right|^P \leq C \right\}. \tag{2.107}$$

The parameter $C \geq 0$ influences the size of the neighborhood support region, whereas $P \geq 0$ influences the shape. The discrete *multi-dimensional convolution* of a signal $s(\mathbf{n})$ by the impulse response $h(\mathbf{n})$ is then defined as a finite-neighborhood operation

$$g(\mathbf{n}) = \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{0})} s(\mathbf{m}) \cdot h(\mathbf{n}-\mathbf{m}) = \sum_{\mathbf{m} \in \mathcal{N}(\mathbf{0})} h(\mathbf{m}) \cdot s(\mathbf{n}-\mathbf{m}). \tag{2.108}$$

The support region $\mathcal{N}$ in (2.108) can specify impulse responses which have either finite or infinite extension.

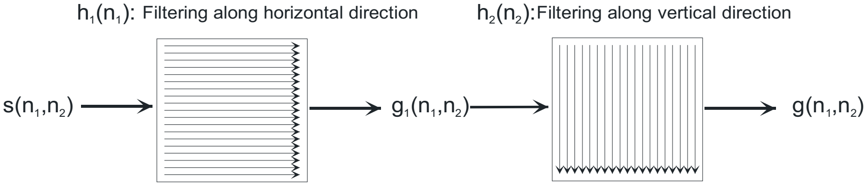
$h_1(n_1)$: Filtering along horizontal direction
$h_2(n_2)$: Filtering along vertical direction
$s(n_1,n_2)$
$g_1(n_1,n_2)$
$g(n_1,n_2)$

Fig. 2.18. Principle of a separable 2D LSI system with horizontal filter step first

Separable 2D LSI systems can be implemented in a similar fashion as per (2.26). Fig. 2.18 shows the principle, where first a horizontal 1D convolution is performed along each row, resulting in $g_1(n_1,n_2)$. In the next step, $g(n_1,n_2)$ is computed by convolving each column of $g_1(n_1,n_2)$. *Infinite Impulse Response* (IIR) filters are not realized by direct implementation of the convolution equation (2.108), but use feedback from previous output values $g(n_1,n_2)$. A given sequence of processing has to be obeyed due to the recursive relationship. For a 2D geometry, all positions which need to be previously processed to provide the input for the current position establish the support region $\mathcal{N}$. Fig. 2.19 shows three different causal IIR filter geometries with their respective $\mathcal{N}$ geometries: The *wedge plane filter*, the *quarter plane filter* and the *asymmetric half plane filter*. For the cases of quarter-plane and wedge-plane filter masks, either row-wise or column-wise recursion scans are possible; these filters also allow processing sequences with diagonal scans, or computation of all samples positioned on a diagonal in parallel (denoted as *wavefront processing*). For the asymmetric half plane filter, row-wise processing (starting at the top left position) is the only possible sequence of recursion. On a rectangular grid, only quarter-plane filters can be defined from separable causal 1D filters.

also a member of the same neighbourhood systems when applied to any of its neighbours (m_1,m_2). The neighbourhood can also be infinitely extended, e.g. for $P=0$ and $C \geq 2$.

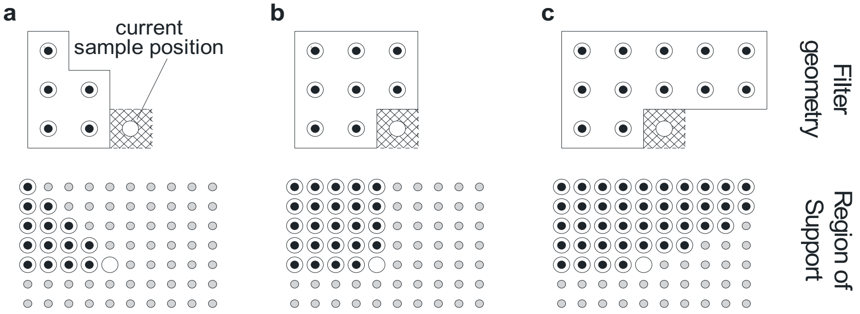

Fig. 2.19. Causal 2D filter masks and geometries of their support regions: **a** Wedge plane **b** Quarter plane **c** Asymmetric half plane

A recursive 2D quarter-plane filter, where the filter geometry defines the feedback from $(P_1+1)(P_2+1)-1$ previously filtered samples, generates the output signal

$$g(n_1,n_2)=s(n_1,n_2)+\sum_{\substack{m_1=0\\(m_1,m_2)\neq(0,0)}}^{P_1}\sum_{m_2=0}^{P_2} b(m_1,m_2)\cdot g(n_1-m_1,n_2-m_2)\,. \tag{2.109}$$

In case of separable recursive filtering, lines and columns of a picture can be processed sequentially, such that the result of filtering along one of the dimensions is input to the filter along the other dimension, e.g. with horizontal processing first as

$$\begin{aligned} g_1(n_1,n_2)&=s(n_1,n_2)+\sum_{m_1=1}^{P_1} b_1(m_1)\cdot g_1(n_1-m_1,n_2)\ \text{ (for all } n_2\text{)}\\ g(n_1,n_2)&=g_1(n_1,n_2)+\sum_{m_2=1}^{P_2} b_2(m_2)\cdot g(n_1,n_2-m_2)\ \text{ (for all } n_1\text{).}\end{aligned} \tag{2.110}$$

The actual relation between the recursive coefficients in 1D and 2D can be determined by the difference equation (2.84) and its modification (2.85), such that

$$b(m_1,m_2)=-\tilde{b}_1(m_1)\tilde{b}(m_2)\ \text{ with } \tilde{b}_i(0)=1,\ \tilde{b}_i(m_i)=-b_i(m_i) \text{ for } 1\leq m_i\leq P_i\,. \tag{2.111}$$

Pictures are finite, where the output of filtering shall usually have the same size as the input e.g. for display purposes. Indices **n**−**m** in (2.108) can however have values less than zero or larger than the maximum coordinates M_1-1 or M_2-1, when samples close to the image boundaries shall be processed. Hence, it is necessary to define a signal extension beyond the boundaries of the input signal to consistently compute the convolution. Zero-setting of values is not useful, as pictures typically have non-zero mean. Three methods copying samples from the picture beyond the boundary, and therefore applicable for FIR filtering, are shown in Figs. 2.20a-c[18].

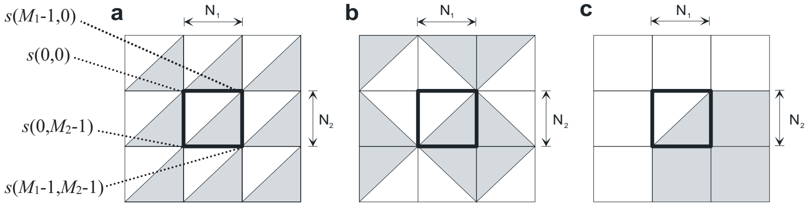

Fig. 2.20. Boundary extensions of finite image signals.
a periodic **b** symmetric (antiperiodic) **c** constant value

Fourier transfer functions of multi-dimensional filters. The multi-dimensional Fourier transform of the discrete impulse response is

$$H_\delta(\mathbf{f}) = \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_\kappa=-\infty}^{\infty} h(\mathbf{n}) \cdot \mathrm{e}^{-\mathrm{j}2\pi\mathbf{f}^{\mathrm{T}}\mathbf{n}} \ . \tag{2.112}$$

If the system has FIR or causal IIR property, the summation limits can be bounded, such that the complex transfer function can directly be determined. For example, a 2D FIR system with a symmetric neighborhood of size $(Q_1+1)(Q_2+1)$ (Q_1 and Q_2 even) gives

$$H_\delta(f_1,f_2) = \sum_{n_1=-Q_1/2}^{Q_1/2} \sum_{n_2=-Q_2/2}^{Q_2/2} a(n_1,n_2)\ \mathrm{e}^{-\mathrm{j}2\pi n_1 f_1} \mathrm{e}^{-\mathrm{j}2\pi n_2 f_2} \ , \tag{2.113}$$

or for the case of a 2D quarter-plane IIR system, the Fourier transfer function is

$$H_\delta(f_1,f_2) = \frac{1}{1-\sum\limits_{\substack{n_1=0\\(n_1,n_2)\neq(0,0)}}^{P_1}\sum\limits_{n_2=0}^{P_2} b(n_1,n_2)\ \mathrm{e}^{-\mathrm{j}2\pi n_1 f_1}\mathrm{e}^{-\mathrm{j}2\pi n_2 f_2}} \ . \tag{2.114}$$

The filter types and geometries of (2.113) and (2.114) are often used in the context of spatial prediction and interpolation of pictures.

[18] In case of IIR filters, it is necessary to define start values for the recursion from values $g(\mathbf{n}-\mathbf{m})$ which would be outside of the picture. Usually this should reflect the mean expectation, e.g. zero for audio/speech, mean gray value for pictures.

2.5 Statistical analysis

Statistical analysis is mainly discussed here for *sampled* multimedia signals $s(\mathbf{n})$, however similar properties hold for continuous signals $s(\mathbf{t})$. An ideal assumption would be *stationarity*, i.e. statistical properties not dependent on the position in time or space. For multimedia signals this does usually not hold; however, similar methods of analysis can be applied on local groups of samples assuming that the properties are invariant there, sufficient for giving reliable empirical measurements. To avoid differentiation between such cases, statistical parameters throughout this chapter are discussed as if they were independent of measurement time and place, and of the data set's size.

It should be observed that in the design of multimedia compression technology, it is normally necessary to use test data sets that exhibit all possible variety. It is even useful to augment test sets by more 'untypical' data which put challenges to the compression algorithm. Even though in adaptive methods usually local statistical properties are exploited, it is still necessary to allow possible adaptation states which give support to the whole variety of data that are expected to be fed into a coder.

2.5.1 Sample statistics

Statistical properties of samples from signals can be characterized by the *Probability Density Function* (PDF) $p_s(x)$, interpreting observed signal amplitudes as instantiations of a *random variable* x of an underlying random process $s(\mathbf{n})$. For the case of continuous amplitudes, the PDF provides information about expected occurrences of certain ranges of amplitude. The probability of a value observation $s(\mathbf{n})\leq x$ is given by the *Cumulative Distribution Function* (CDF)

$$P_s(x) \equiv \Pr\left[s(\mathbf{n}) \leq x\right] = \int_{-\infty}^{x} p_s(\xi)\,\mathrm{d}\xi\ . \tag{2.115}$$

The CDF is monotonically increasing and has a value in the range $P_s(-\infty) = 0 \leq P_s(x) \leq 1 = P_s(\infty)$. The probability of a signal amplitude to be within an interval range $[x_\mathrm{a};x_\mathrm{b}]$ is therefore

$$\Pr\left[x_\mathrm{a} < s(\mathbf{n}) \leq x_\mathrm{b}\right] = \int_{x_\mathrm{a}}^{x_\mathrm{b}} p_s(\xi)\,\mathrm{d}\xi = P_s(x_\mathrm{b}) - P_s(x_\mathrm{a}) \geq 0\ . \tag{2.116}$$

Furthermore,

$$\int_{-\infty}^{\infty} p_s(x)\,\mathrm{d}x = P_s(\infty) - P_s(-\infty) = 1\,. \tag{2.117}$$

The *expected value* $\mathcal{E}\{f[x]\}$ is the mean over a set of signal observations with a function $f[x]$ applied to the samples; it is related to the PDF by[19]

$$\mathcal{E}\left\{f\left[s(\mathbf{n})\right]\right\} = \lim_{N\to\infty}\frac{1}{N}\sum_{\mathbf{n}} f\left[s(\mathbf{n})\right] = \int_{-\infty}^{\infty} f(x)p_s(x)\,\mathrm{d}x\,. \tag{2.118}$$

From these definitions, the following important parameters are defined describing *sample statistics*:

- $f(x)=x$: Mean value $m_s = \int_{-\infty}^{\infty} x\cdot p_s(x)\,\mathrm{d}x = \mathcal{E}\left\{s(\mathbf{n})\right\}$ (2.119)
- $f(x)=x^2$: Quadratic mean (power) $Q_s = \int_{-\infty}^{\infty} x^2\cdot p_s(x)\,\mathrm{d}x = \mathcal{E}\left\{s^2(\mathbf{n})\right\}$ (2.120)
- Variance $\sigma_s{}^2 = \int_{-\infty}^{\infty} (x-m_s)^2\, p_s(x)\,\mathrm{d}x = \mathcal{E}\left\{(s(\mathbf{n})-m_s)^2\right\} = Q_s - m_s{}^2$ (2.121)

For numeric (digital) processing, signal samples are quantized, which means they are mapped into a set of discrete amplitudes (see section 4.1). The mapping function is the *quantization characteristic*, which is a staircase function (see Fig. 2.21 for a case of uniform quantization of a finite positive amplitude range using a step size Δ). The value of the discrete *probability mass function* (PMF) of the quantized process can then be determined from the areas under the PDF of the unquantized process within the respective quantization intervals j with lower boundary x_j, upper boundary x_{j+1} and reconstruction[20] y_j,

$$p_{s_\mathrm{Q}}(y_j) = \int_{x_j}^{x_{j+1}} p_s(x)\,\mathrm{d}x\,. \tag{2.122}$$

[19] The terminology ‘expected value’ is used here both for cases of finite and infinite data sets. Only the latter is mathematically precise. If a finite set of N measurements is used, the expected value is *empirical*, but could be regarded as reliable if it is not significantly changing when N would be further increased.

[20] Typically in uniform quantization of step size Δ, the reconstruction value is placed at the center of the interval, i.e. $x_j=y_j-\Delta/2$ and $x_{j+1}=y_j+\Delta/2$ (see section 4.1). Note that in the context of quantization we will typically assume that representation (encoding) by a finite alphabet is possible. In general, a PMF can also consist of an infinite number of discrete values. This is of no harm if the probability of values converges towards zero at both ends of the amplitude range.

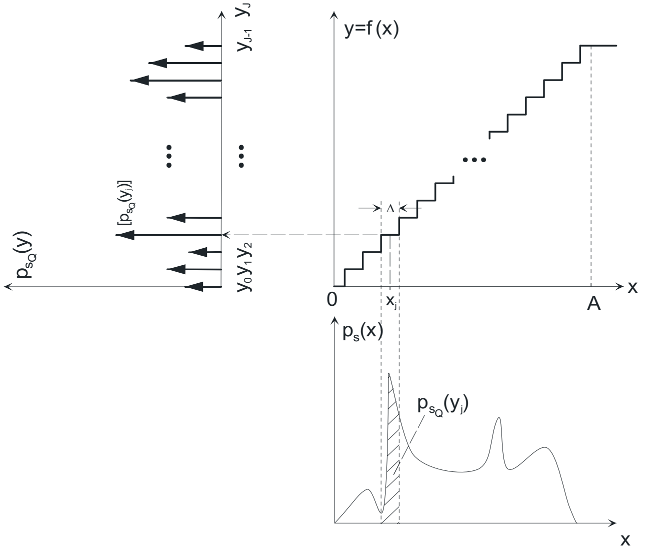

Fig. 2.21. Quantization characteristic and mapping of the PDF $p_s(x)$ of the continuous-amplitude signal to the probability mass function $p_{sQ}(y_j)$ of the quantized (discrete-amplitude) signal.

The PMF expresses the probability of the quantized (discrete) amplitude values y_j. The related PDF consists of a weighted sum of Dirac impulses[21]

$$p_{s,\delta}(x) = \sum_j p_{s_Q}(y_j)\delta(x - y_j) \tag{2.123}$$

where further from (2.117),

$$\sum_j p_{s_Q}(y_j) = 1 . \tag{2.124}$$

and

$$\mathcal{E}\left\{f\left[s_Q(\mathbf{n})\right]\right\} = \int_{-\infty}^{\infty} f(x)\sum_j p_{s_Q}(y_j)\delta(x - y_j)\,\mathrm{d}\,x = \sum_j p_{s_Q}(y_j)f(y_j) . \tag{2.125}$$

PDF models are useful to characterize the statistical behavior of a random process. For example, mean value and variance could be measured and used as parameters

[21] In the sequel, the subscript ‘Q’ is usually omitted, as the fact that the signal has been quantized is obvious from the context. Discrete probability functions (PMF) are written as $p_s(\mathrm{y}_j)$. In the case of finite alphabets, this can also be expressed as $\Pr(S_j)$, where S_j is one discrete state with index j (without explicitly expressing an amplitude value).

under the assumption that a certain PDF shape is given. For multimedia signals, the generalized Gaussian distribution is often useful to express sample statistics[22]:

$$p_s(x) = a\,\mathrm{e}^{-\left|b(x-m_s)\right|^{\gamma}} \quad \text{with } a = \frac{b\gamma}{2\Gamma\left(\frac{1}{\gamma}\right)} \quad \text{and } b = \frac{1}{\sigma_s}\sqrt{\frac{\Gamma\left(\frac{3}{\gamma}\right)}{\Gamma\left(\frac{1}{\gamma}\right)}}\,. \tag{2.126}$$

For $\gamma = 2$, (2.126) gives the Gaussian normal PDF

$$p_s(x) = \frac{1}{\sqrt{2\pi\sigma_s^2}}\mathrm{e}^{-\frac{(x-m_s)^2}{2\sigma_s^2}}\,, \tag{2.127}$$

for which many optimization problems can be solved analytically. The normal PDF plays an important role, as according to the central limit theorem, it is the result of the superposition of a large number of statistically independent random signals. For $\gamma = 1$, (2.126) gives the Laplacian PDF:

$$p_s(x) = \frac{1}{\sqrt{2\sigma_s^2}}\mathrm{e}^{-\frac{\sqrt{2}|x-m_s|}{\sigma_s}}\,. \tag{2.128}$$

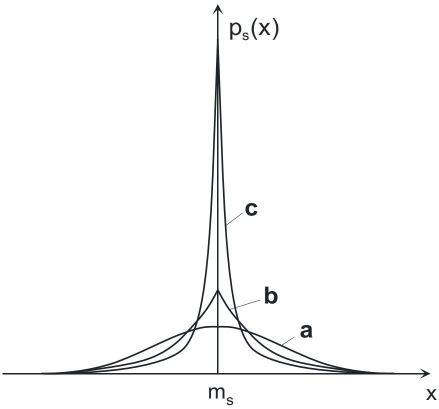

Fig. 2.22. Generalized Gaussian PDF for different values of γ: γ=2, Gaussian (**a**); γ=1, Laplacian (**b**); γ=0.5 (**c**)

Both, Gaussian and Laplacian cases are shown in Fig. 2.22, as well as a more narrow case ($\gamma = 0.5$). The Laplacian PDF has been reported to be a suitable mod-

[22] The function Γ(·) which influences the shape of the PDF via the parameter γ, is defined as $\Gamma(u) = \int_0^\infty \mathrm{e}^{-x} x^{u-1}\,\mathrm{d}x$.

el for the probability distribution of DCT block transform coefficients extracted from still images [Reininger, Gibson 1983] [Lam, Goodman 2000], and from motion compensated residual signals [Bellifemine et al. 1992] as used in video coding. Finally, for $\gamma \to \infty$, (2.126) also expresses a uniform distribution (see problem 2.4),

$$p_s(x) = \frac{1}{\sqrt{12\sigma_s^{\;2}}} \operatorname{rect}\left(\frac{x - m_s}{\sqrt{12\sigma_s^{\;2}}} \right). \tag{2.129}$$

Models for discrete PMFs can be derived from analytic PDF models by applying an appropriate quantization in (2.122). Direct *sampling* of a PDF might give similar results in the case of small quantization step size Δ, but would typically lead to violation of (2.124) which would require re-normalization of the values. Another approach is representation of a continuous PDF by a mixture distribution – mixtures of Gaussians are often used for this purpose,

$$p_s(x) = \sum_i w_i \frac{1}{\sqrt{2\pi\sigma_{s_i}^{\;2}}} \mathrm{e}^{-\frac{(x-m_{s_i})^2}{2\sigma_{s_i}^{\;2}}} . \tag{2.130}$$

The parameters m_{s_i}, σ_{s_i} and the weights w_i of the different contributing Gaussian hulls, as well as the number of hulls have to be estimated. This can be achieved by initially identifying local peaks in the PDF to be described, analyze the slopes around the peaks, and then refine the match by algorithms such as expectation maximization or kernel density estimation (see MCA, Ch. 5).

Models of PMFs can also be formulated directly in a finite discrete number space. As an example, the Bernoulli or binomial PMF defines probabilities of J discrete values, such that the jth value state occurs by probability

$$\Pr(S_j) = \binom{J-1}{j-1} p^{j-1} (1-p)^{J-j}; \quad 1 \le j \le J . \tag{2.131}$$

Alternatively, the probability values of the Bernoulli distribution in the J discrete states can be obtained by convolutions involving J–1 subsequent [p 1–p] FIR filter kernels. The symmetric case of the Bernoulli distribution (p=0.5) with increasing J can also be interpreted as discrete counterpart of the Gaussian normal PDF, which would be approached by iterative convolution of narrow continuous rectangular pulses.

2.5.2 Joint statistical properties

Joint probability functions (CDF, PDF or PMF) are used to express statistics about joint observations of two or multiple random values. Herein, the values can either stem from the same or from different signals, and/or from same or different locations in time and space. Therefore, joint probability functions express dependencies that exist either between the samples of only one or of different random signals. Joint probability functions have a K-dimensional dependency when K values are observed jointly. For the following paragraphs, the case K=2 is discussed, assuming $s_1(\mathbf{n})$ and $s_2(\mathbf{n}+\mathbf{k})$ are two observations with a relative shift of $\mathbf{k}$ samples. The concepts straightforwardly extend to higher K when additional observations are made.

The joint PDF $p_{s_1s_2}(x_1,x_2;\mathbf{k})$ is a 2-dimensional function (for one value of $\mathbf{k}$). The basic rules which are given in this section are applicable likewise to the discrete PMF or other discrete joint probability functions. Firstly, the joint functions are symmetric,

$$p_{s_1s_2}(x_1,x_2;\mathbf{k}) = p_{s_2s_1}(x_2,x_1;\mathbf{k})\,. \tag{2.132}$$

In the hypothetical case that the observed samples were generally identical,

$$p_{s_1s_2}(x_1,x_2;\mathbf{k}) = p_{s_1}(x_1)\delta(x_2-x_1) = p_{s_2}(x_2)\delta(x_1-x_2)\,, \tag{2.133}$$

whereas for statistical independence,

$$p_{s_1s_2}(x_1,x_2;\mathbf{k}) = p_{s_1}(x_1)p_{s_2}(x_2)\,. \tag{2.134}$$

Conditional probabilities allow to express an expectation about the probability of random variables x_1 for the first observation, if it is already known that the other observation came as x_2, expressing the '*probability of* x_1 *given* x_2'. No uncertainty about the conditioning event exists, such that the conditional probabilities can be gained from the joint probability, normalized by the probability of the condition,

$$p_{s_1s_2}(x_1|x_2;\mathbf{k}) = \frac{p_{s_1s_2}(x_1,x_2;\mathbf{k})}{p_{s_2}(x_2)}\;;\quad p_{s_2s_1}(x_2|x_1;\mathbf{k}) = \frac{p_{s_1s_2}(x_1,x_2;\mathbf{k})}{p_{s_1}(x_1)}\,. \tag{2.135}$$

For statistically independent processes, (2.134) and (2.135) give $p_{s_1s_2}(x_1|x_2;\mathbf{k})=p_{s_1}(x_1)$ and $p_{s_2s_1}(x_2|x_1;\mathbf{k})=p_{s_2}(x_2)$, i.e. the given condition does not help to decrease uncertainty.

These concepts can likewise be extended to joint statistics of more than two signals or more than two samples from one signal. If e.g. K values from one or several continuous-amplitude signal(s) are combined into a vector $\mathbf{s} = [s_1,s_2, \ldots , s_K]^{\mathrm{T}}$, the joint probability density becomes also K-dimensional and is denoted as *vector PDF*[23]

$$p_{\mathbf{s}}(\mathbf{x}) = p_{s_1s_2\ldots s_K}(x_1,x_2,\ldots,x_K)\,, \tag{2.136}$$

where specifically for the case of statistical independency of the vector elements

[23] For simplicity, it is not explicitly expressed here that the samples of the vector can stem from various locations; in principle, individual shift parameters $\mathbf{k}$ would optionally need to be specified for the elements of the vector.

$$p_{\mathbf{s}}(\mathbf{x}) = p_{s_1}(x_1) \cdot p_{s_2}(x_2) \cdot \ldots \cdot p_{s_K}(x_K)\,. \tag{2.137}$$

The conditional PDF of a sample $s(\mathbf{n})$, provided that a conditioning vector $\mathbf{s}$ is given (which shall not include the sample itself), is defined as

$$p_{s|\mathbf{s}}(x\,|\,\mathbf{x}) = \frac{p_{s\mathbf{s}}(x,\mathbf{x})}{p_{\mathbf{s}}(\mathbf{x})}\,, \tag{2.138}$$

which for each given $\mathbf{x}$ is a one-dimensional PDF over variable x. In the context of joint analysis, also the definition of the joint expected value has to be extended to functions over several variables which are taken from distant positions in the signal, such as

$$\begin{aligned}\mathcal{E}\left\{f\left[s_1(\mathbf{n}),s_2(\mathbf{n}+\mathbf{k}),\ldots\right]\right\} &= \lim_{N\to\infty}\frac{1}{N}\sum_{\mathbf{n}} f\left[s_1(\mathbf{n}),s_2(\mathbf{n}+\mathbf{k}),\ldots\right] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{s_1 s_s ..}(x_1,x_2,\ldots;\mathbf{k}) f(x_1,x_2,\ldots)\,\mathrm{d}\,x_2\,\mathrm{d}\,x_1.\end{aligned} \tag{2.139}$$

The joint PDF $p_{s_1s_2}(x_1,x_2;\mathbf{k})$ expresses the probability of a constellation where one random sample $s_1(\mathbf{n})$ has a value x_1, while the other sample $s_2(\mathbf{n}+\mathbf{k})$ has a value x_2. From this, linear statistical dependencies between the two samples are expressed by the *correlation function*[24]:

$$\begin{aligned}\varphi_{s_1s_2}(\mathbf{k}) = \mathcal{E}\left\{s_1(\mathbf{n})s_2(\mathbf{n}+\mathbf{k})\right\} &= \lim_{N\to\infty}\frac{1}{N}\sum_{\mathbf{n}} s_1(\mathbf{n})s_2(\mathbf{n}+\mathbf{k}) \\ = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1x_2\,p_{s_1s_2}(x_1,x_2;\mathbf{k})\,\mathrm{d}\,x_1\,\mathrm{d}\,x_2.&\end{aligned} \tag{2.140}$$

For the case $s_1=s_2=s$ (samples for correlation calculation taken from the same signal $s(\mathbf{n})$), (2.140) is an *autocorrelation function* (*ACF*), otherwise a *cross correlation function* (*CCF*). The *covariance function* is similarly computed by separating the mean values:

$$\mu_{s_1s_2}(\mathbf{k}) = \mathcal{E}\left\{\left[s_1(\mathbf{n})-m_{s_1}\right]\left[s_2(\mathbf{n}+\mathbf{k})-m_{s_2}\right]\right\} = \varphi_{s_1s_2}(\mathbf{k}) - m_{s_1}m_{s_2}\,. \tag{2.141}$$

The autocorrelation (2.140) and autocovariance (2.141) for $\mathbf{k}=\mathbf{0}$ give the power (2.120) and the variance (2.121), respectively. These are the maximum values of these functions. When normalized by their respective maxima, the resulting *standardized autocorrelation* and *autocovariance functions* have values between −1 and +1:

[24] For quantized signals, the expected value can be computed from the PMF by applying (2.125) analogously, which is used here.

$$\alpha_{ss}(\mathbf{k}) = \frac{\varphi_{ss}(\mathbf{k})}{\varphi_{ss}(\mathbf{0})} = \frac{\varphi_{ss}(\mathbf{k})}{Q_s} \quad ; \quad \rho_{ss}(\mathbf{k}) = \frac{\mu_{ss}(\mathbf{k})}{\mu_{ss}(\mathbf{0})} = \frac{\mu_{ss}(\mathbf{k})}{\sigma_s{}^2}. \tag{2.142}$$

A similar normalization by the cross power and cross variance (values for **k**=**0**) is applicable to the cross correlation and covariance functions,

$$\alpha_{s_1s_2}(\mathbf{k}) = \frac{\varphi_{s_1s_2}(\mathbf{k})}{\sqrt{Q_{s_1}Q_{s_2}}} \quad ; \quad \rho_{s_1s_2}(\mathbf{k}) = \frac{\mu_{s_1s_2}(\mathbf{k})}{\sigma_{s_1}\sigma_{s_2}}. \tag{2.143}$$

Correlation and covariance functions analyze *linear statistical dependencies*. If two signals are *uncorrelated*, $\varphi_{s_1s_2}(\mathbf{k})=m_{s_1}m_{s_2}$ and $\mu_{s_1s_2}(\mathbf{k})=0$ over all **k**. Unless periodic components are present in a signal, the following conditions hold for the ACF and covariance if |**k**| grows large[25]:

$$\lim_{|\mathbf{k}|\to\infty} \varphi_{ss}(\mathbf{k}) = m_s{}^2 \quad ; \quad \lim_{|\mathbf{k}|\to\infty} \mu_{ss}(\mathbf{k}) = 0. \tag{2.144}$$

It should be observed that 'uncorrelated' signals or signal samples are not necessarily statistically independent. More general *nonlinear dependencies* cannot be identified by correlation functions. Cases of such nonlinear dependencies are real-valued signals that are similar by amplitude but have random sign, or complex-valued signals that are similar in amplitude but have random phase properties compared to each other.

Two correlated or uncorrelated, zero-mean stationary Gaussian processes $s_1(\mathbf{n})$ and $s_2(\mathbf{n})$ shall be given. After normalizing their amplitudes by the standard deviations, a sum process and a difference process are established as follows:

$$\Sigma(\mathbf{n},\mathbf{k}) = \frac{s_1(\mathbf{n})}{\sigma_{s_1}} + \frac{s_2(\mathbf{n}+\mathbf{k})}{\sigma_{s_2}}; \; \Delta(\mathbf{n},\mathbf{k}) = \frac{s_1(\mathbf{n})}{\sigma_{s_1}} - \frac{s_2(\mathbf{n}+\mathbf{k})}{\sigma_{s_2}}. \tag{2.145}$$

Sum and difference processes are zero-mean Gaussian as well, having the following variances:

$$\sigma_\Sigma^2(\tau) = \mathcal{E}\left\{\left[\frac{s_1(\mathbf{n})}{\sigma_{s_1}} + \frac{s_2(\mathbf{n}+\mathbf{k})}{\sigma_{s_2}}\right]^2\right\} = 2[1+\rho_{s_1s_2}(\mathbf{k})], \tag{2.146}$$

and similarly

$$\sigma_\Delta^2(\tau) = \mathcal{E}\left\{\left[\frac{s_1(\mathbf{n})}{\sigma_{s_1}} - \frac{s_2(\mathbf{n}+\mathbf{k})}{\sigma_{s_2}}\right]^2\right\} = 2[1-\rho_{s_1s_2}(\mathbf{k})], \tag{2.147}$$

where $\rho_{s_1s_2}(\tau)$ is the standardized cross covariance following the principle of (2.142). The correlation between the sum and difference processes is

[25] In case of multi-dimensional correlation functions It is sufficient when one of the values in the vector **k** grows large.

$$\mathcal{E}\{\Sigma(\mathbf{n},\mathbf{k})\Delta(\mathbf{n},\mathbf{k})\} = \mathcal{E}\left\{\left[\frac{s_1(\mathbf{n})}{\sigma_{s_1}} + \frac{s_2(\mathbf{n}+\mathbf{k})}{\sigma_{s_2}}\right]\left[\frac{s_1(\mathbf{n})}{\sigma_{s_1}} - \frac{s_2(\mathbf{n}+\mathbf{k})}{\sigma_{s_2}}\right]\right\}$$
$$= \frac{\mathcal{E}\{s_1{}^2(\mathbf{n})\}}{\sigma_{s_1}^2} - \frac{\mathcal{E}\{s_2{}^2(\mathbf{n}+\mathbf{k})\}}{\sigma_{s_2}^2} = 0. \tag{2.148}$$

Due to Gaussian property, the uncorrelated sum and difference processes are furthermore statistically independent. The joint PDF therefore is

$$p_{\Sigma\Delta}(y_1, y_2, \mathbf{k}) = \underbrace{\frac{1}{\sqrt{4\pi[1+\rho_{s_1s_2}(\mathbf{k})]}}\mathrm{e}^{-\frac{y_1{}^2}{4[1+\rho_{s_1s_2}(\mathbf{k})]}}}_{p_\Sigma(y_1)} \cdot \underbrace{\frac{1}{\sqrt{4\pi[1-\rho_{s_1s_2}(\mathbf{k})]}}\mathrm{e}^{-\frac{y_2{}^2}{4[1-\rho_{s_1s_2}(\mathbf{k})]}}}_{p_\Delta(y_2)}$$
$$= \frac{1}{4\pi\sqrt{1-\rho_{s_1s_2}{}^2(\mathbf{k})}}\mathrm{e}^{-\frac{y_1{}^2[1-\rho_{s_1s_2}(\mathbf{k})]+y_2{}^2[1+\rho_{s_1s_2}(\mathbf{k})]}{4[1-\rho_{s_1s_2}{}^2(\mathbf{k})]}} \tag{2.149}$$

Reverse mapping from y_1 and y_2 to the random variables x_1 and x_2 of the original processes $s_1(\mathbf{n})$ and $s_2(\mathbf{n})$ gives

$$y_1 = \frac{x_1}{\sigma_{s_1}} + \frac{x_2}{\sigma_{s_2}} = \frac{\sigma_{s_2}x_1 + \sigma_{s_1}x_2}{\sigma_{s_1}\sigma_{s_2}}; \; y_2 = \frac{x_1}{\sigma_{s_1}} - \frac{x_2}{\sigma_{s_2}} = \frac{\sigma_{s_2}x_1 - \sigma_{s_1}x_2}{\sigma_{s_1}\sigma_{s_2}} \tag{2.150}$$

such that

$$p_{s_1s_2}(x_1, x_2; \mathbf{k}) = \frac{1}{2\pi\sigma_{s_1}\sigma_{s_2}\sqrt{1-\rho_{s_1s_2}{}^2(\mathbf{k})}}\mathrm{e}^{-\frac{\sigma_{s_2}^2 x_1{}^2 + \sigma_{s_1}^2 x_2{}^2 - 2\sigma_{s_1}\sigma_{s_2}\rho_{s_1s_2}(\mathbf{k})x_1x_2}{2\sigma_{s_1}^2\sigma_{s_2}^2(1-\rho_{s_1s_2}{}^2(\mathbf{k}))}}. \tag{2.151}$$

Generalization to the case of non-zero mean processes further gives

$$p_{s_1s_2}(x_1, x_2; \mathbf{k}) = \frac{1}{2\pi\sigma_{s_1}\sigma_{s_2}\sqrt{1-\rho_{s_1s_2}{}^2(\mathbf{k})}}\mathrm{e}^{-\frac{\sigma_{s_2}^2(x_1-m_{s_1})^2 + \sigma_{s_1}^2(x_2-m_{s_2})^2 - 2\sigma_{s_1}\sigma_{s_2}\rho_{s_1s_2}(\mathbf{k})(x_1-m_{s_1})(x_2-m_{s_2})}{2\sigma_{s_1}^2\sigma_{s_2}^2(1-\rho_{s_1s_2}{}^2(\mathbf{k}))}}. \tag{2.152}$$

A more compact expression of (2.152) is possible by the following matrix notation using a *covariance matrix* $\mathbf{C}_{s_1s_2}$,

$$p_{s_1s_2}(x_1, x_2; \mathbf{k}) = \frac{1}{\sqrt{(2\pi)^2 \cdot |\mathbf{C}_{s_1s_2}(\mathbf{k})|}} \cdot e^{-\frac{1}{2}\xi^{\mathrm{T}}\mathbf{C}_{s_1s_2}(\mathbf{k})^{-1}\xi} \quad \text{with } \xi = \begin{bmatrix} x_1 - m_{s_1} \\ x_2 - m_{s_2} \end{bmatrix}$$
$$\text{and } \mathbf{C}_{s_1s_2}(\mathbf{k}) = \mathcal{E}\{\xi \cdot \xi^{\mathrm{T}}\} = \begin{bmatrix} \sigma_{s_1}^2 & \mu_{s_1s_2}(\mathbf{k}) \\ \mu_{s_1s_2}(\mathbf{k}) & \sigma_{s_2}^2 \end{bmatrix} \tag{2.153}$$
$$\Rightarrow \mathbf{C}_{s_1s_2}(\mathbf{k})^{-1} = \frac{1}{\underbrace{\sigma_{s_1}^2\sigma_{s_2}^2\left(1-\rho_{s_1s_2}{}^2(\mathbf{k})\right)}_{|\mathbf{C}_{s_1s_2}(\mathbf{k})|}}\begin{bmatrix} \sigma_{s_2}^2 & -\sigma_{s_1}\sigma_{s_2}\rho_{s_1s_2}(\mathbf{k}) \\ -\sigma_{s_1}\sigma_{s_2}\rho_{s_1s_2}(\mathbf{k}) & \sigma_{s_1}^2 \end{bmatrix}.$$

The transformation (2.145) into sum and difference processes can be interpreted as a coordinate transformation from a Cartesian (x_1,x_2) coordinate space into the rotated (y_1,y_2) coordinate space, where the axes y_1 and y_2 are still orthogonal. Equal values of the PDF, according to the exponent in (2.149), can be found on ellipses with principal axes along the y_1 and y_2 axes[26], scaled by $\sqrt{1+\rho_{s_1s_2}(\mathbf{k})}$ and $\sqrt{1-\rho_{s_1s_2}(\mathbf{k})}$, respectively.

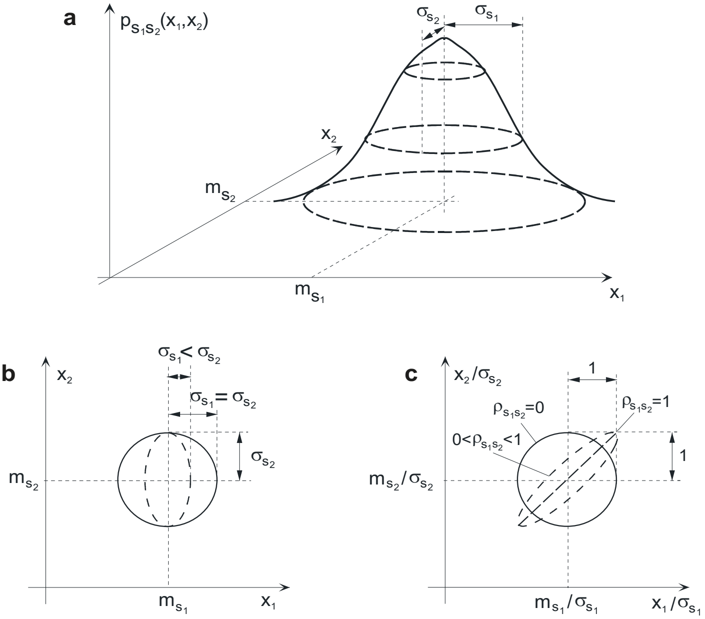

Fig. 2.23. Joint 2D Gaussian PDF $p_{s_1s_2}(x,y)$ **a** for case of statistically independent signals **b** modification of the shape by different variances **c** modification of the shape by different co-variances

Fig. 2.23a shows the shape of the 2D Gaussian PDF for the case of statistically independent signals. Figs. 2.23b/c illustrate the influence of variance and covariance. For the case of negative covariance, the longer axis (higher variance) of the ellipse would follow the y_2 axis of the difference process.

(2.153) straightforwardly extends to the general case where the correlation properties between measurements of K random values combined in a vector notation (2.136) are formulated in a covariance matrix

$$\mathbf{C}_{\mathbf{ss}} = \mathcal{E}\left\{\mathbf{s}\mathbf{s}^{\mathrm{T}}\right\} - \mathbf{m}_{\mathbf{s}}\mathbf{m}_{\mathbf{s}}^{\mathrm{T}} = \left[\mathcal{E}\left\{s_i s_j\right\} - m_{s_i} m_{s_j}\right] . \qquad (2.154)$$

using the vector of linear mean values

[26] Note that the axes y_1 and y_2 are defined after normalization of the x and y axes by the standard deviations of the respective processes. The ellipse's orientation therefore is 45 degrees relative to the normalized x and y axes.

$$\mathbf{m}_\mathbf{s} = \mathcal{E}\{\mathbf{s}\} = \left[\mathcal{E}\{s_i\}\right] \text{ with } 1 \le i \le K \, . \tag{2.155}$$

The joint PDF in this case can be expressed as *vector Gaussian PDF*

$$p_\mathbf{s}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^K \cdot \left|\mathbf{C}_\mathbf{ss}\right|}} \cdot e^{-\frac{1}{2}[\mathbf{x}-\mathbf{m}_\mathbf{s}]^\mathrm{T} \mathbf{C}_\mathbf{ss}^{-1} [\mathbf{x}-\mathbf{m}_\mathbf{s}]} \, . \tag{2.156}$$

Again, to show its properties, it would be necessary to find an alternative representation by applying a linear transformation to the normalized combination of random samples (as in the case above by the sum and difference operations). After this, K statistically independent output processes are available, which in case of Gaussian processes means they are uncorrelated. In (2.279)-(2.282) it will be shown that this is possible by computing the set of eigenvectors of the covariance matrix, which establish a new orthogonal coordinate system, on which the amplitudes of random vectors **s** are projected. In case of a Gaussian PDF, equal values are then found on the hull of a K-dimensional hyper-ellipsoid, with principal axes having same orientations as the corresponding eigenvectors, and widths of the ellipsoid axes proportional to the square roots of the related eigenvalues.

$$\begin{aligned} \mathbf{C}_\mathbf{ss} &= \begin{bmatrix} \mu_{ss}(0) & \mu_{ss}(1) & \mu_{ss}(2) & \cdots & \mu_{ss}(K-1) \\ \mu_{ss}(1) & \mu_{ss}(0) & \mu_{ss}(1) & \ddots & \vdots \\ \mu_{ss}(2) & \mu_{ss}(1) & \mu_{ss}(0) & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \mu_{ss}(1) \\ \mu_{ss}(K-1) & \cdots & & \mu_{ss}(1) & \mu_{ss}(0) \end{bmatrix} \\ &= \sigma_s^2 \cdot \begin{bmatrix} 1 & \rho_{ss}(1) & \rho_{ss}(2) & \cdots & \rho_{ss}(K-1) \\ \rho_{ss}(1) & 1 & \rho_{ss}(1) & \ddots & \vdots \\ \rho_{ss}(2) & \rho_{ss}(1) & 1 & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \rho_{ss}(1) \\ \rho_{ss}(K-1) & \cdots & & \rho_{ss}(1) & 1 \end{bmatrix}, \end{aligned} \tag{2.157}$$

A special case applies, if the observations combined in the vector **s** are K samples from one single stationary Gaussian process, which are taken at equidistant time positions. In this case, the covariance matrix becomes an autocovariance matrix which has the following Toeplitz structure of (2.157)[27], where the mean vector is filled by a constant mean value,

$$\mathbf{m}_\mathbf{s} = m_s \cdot \mathbf{1} = \begin{bmatrix} m_s & m_s & \cdots & m_s \end{bmatrix}^\mathrm{T} \tag{2.158}$$

[27] In case of stationarity, variance and covariance values only depend on the distance, i.e. $\mathcal{E}\{s(0)s(1)\}=\mathcal{E}\{s(1)s(2)\}=\ldots$, which leads to this structure.

2.5.3 Spectral properties of random signals

The Fourier transform of the correlation function gives the *power density spectrum*[28]

$$\varphi_{ss}(\mathbf{k}) = \mathcal{E}\left\{s(\mathbf{n})s(\mathbf{n}+\mathbf{k})\right\} \quad \circ\!\!-\!\!\bullet \quad \Phi_{ss,\delta}(\mathbf{f}) = \mathcal{E}\left\{\left|S_\delta(\mathbf{f})\right|^2\right\} \tag{2.159}$$

The relationship between the power (quadratic mean) value and the power density spectrum is expressed by *Parseval's theorem*,

$$Q_s = \varphi_{ss}(\mathbf{0}) = \int_{-1/2}^{1/2} \cdots \int_{-1/2}^{1/2} \Phi_{ss,\delta}(\mathbf{f})\mathrm{d}^\kappa \mathbf{f}\,. \tag{2.160}$$

If a random process is zero-mean, its autocorrelation and autocovariance functions are identical. Otherwise, the autocorrelation is increased by m_s^2. Likewise, for non-zero mean processes, a Dirac impulse is contained in the power density spectrum at $\mathbf{f}=\mathbf{0}$ (and at all periodic copies) with a weight m_s^2, corresponding to the power of the mean value (DC component). With presence of periodic components, Dirac impulses would be contained in the power density spectrum at the corresponding frequency locations.

Estimation of power spectra is often done via the DFT (2.89), i.e. a sampled frequency axis is used in computing the expected value in the right part of (2.159). For this purpose, blocks of M samples (ΠM_i samples for two- and multidimensional finite signals) are transformed into instantaneous DFT energy spectra $|S_\mathrm{d}(k)|^2$. To minimize the effect of the inherent periodic continuation of the DFT, window functions can be used to let the signal decay towards zero at the boundaries of the analysis block. An alternative way to estimate power density spectra can be achieved via autoregressive (AR) modeling (see section 2.6.1). Both DFT-based spectral estimation, as well as AR modeling can be applied locally over a finite number of samples e.g. with the goal to adapt a compression algorithm by instantaneous (local) signal properties, or globally by computing expected values (power density spectrum or ACF) over a sufficiently large number of samples of a random process, which could be used to tune the general properties of a compression algorithm by the typical statistics of the given class of multimedia signals.

2.5.4 Markov chain models

The *state change behavior* of random processes with rather discrete appearance needs to be modeled for binary signals $b(n) \in \{0,1\}$ (e.g. two-tone images), bit streams, or for features on a more abstract level, e.g. segment transitions in space or time, where a segment relates to a semantic unit (spoken word, video scene,

[28] Note that the power spectrum of sampled random signals is periodic. Furthermore, in the formulation of the expected value over spectra from random signals in (2.159), a normalization by the time span of the Fourier transform must be performed to get an expression about the average spectral power density within a given frequency range.

region in an image). A simple model to define finite states of signals with memory is the *Markov chain*, in simplest case a 2-state (binary) model as shown in Fig. 2.24a[29]. As $b(n)$ has only two states S_0='0' and S_1='1', the model is fully defined by transition probabilities of its temporal sequence[30] $\Pr(S_0|S_1)$ (S_0 follows S_1), and $\Pr(S_1|S_0)$ (S_1 follows S_0). The remaining probabilities $\Pr(S_0|S_0)$ and $\Pr(S_1|S_1)$, which express occurrence of sequences with two equal values, can in case of the two-state chain be derived as

$$\Pr(S_i \mid S_i) = 1 - \Pr(S_j \mid S_i)\,. \tag{2.161}$$

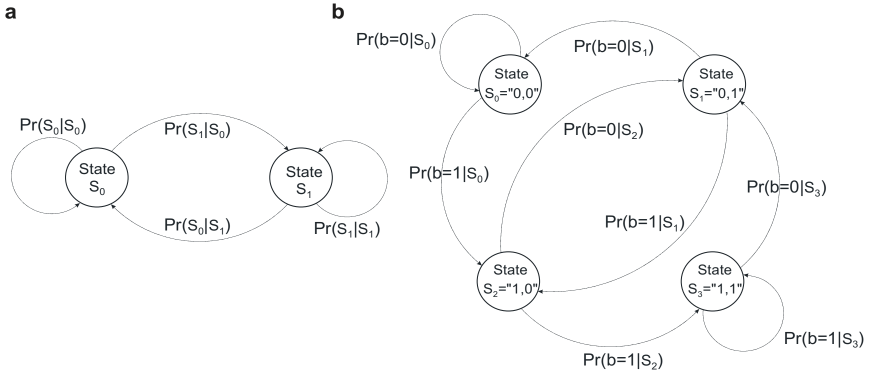

Fig. 2.24. Binary sequences modeled by Markov chain of **a** two states **b** four states giving dependency on two previous binary states

The 'Markov property' of the model process shall fulfill two conditions:

- The probability to be in a state is only dependent on the transition probabilities leading to this state, coupled with the respective probability of the state from which the transition is made;
- The model shall be stationary, the probability of states shall be independent of time or location of observation.

This can be formulated as follows for the two-state model, based on a state transition matrix **P**:

$$\begin{bmatrix}\Pr(S_0)\\ \Pr(S_1)\end{bmatrix} = \underbrace{\begin{bmatrix}\Pr(S_0 \mid S_0) & \Pr(S_0 \mid S_1)\\ \Pr(S_1 \mid S_0) & \Pr(S_1 \mid S_1)\end{bmatrix}}_{\mathbf{P}} \begin{bmatrix}\Pr(S_0)\\ \Pr(S_1)\end{bmatrix}. \tag{2.162}$$

From this, the global probabilities of two states can be determined as

[29] The problem will be discussed here mainly for binary sequences $b(n)$, but it can formally be extended to any sequences of discrete events $\mathscr{s}(n) \in \{S_j;\, j=1,\ldots,J\}$. Extensions to continuous-amplitude signals $s(n)$ are also made by *Markov Random Fields* (cf. Sec. 6.6.2).

[30] For simple notation, $\Pr(S_i|S_j) \equiv \mathrm{Prob}[b(n)=S_i|b(n-1)=S_j]$.

$$\Pr(S_i) = \frac{\Pr(S_i \mid S_j)}{\Pr(S_i \mid S_j) + \Pr(S_j \mid S_i)} = 1 - \Pr(S_j)\,; \quad [i,j] \in \{0,1\}. \tag{2.163}$$

Once a state is given, the probability of '0'- or '1'-sequences of remaining length l can be determined by concatenating the probabilities that the model rests in the state for another $l-1$ cycles and then changes,

$$\begin{aligned} \mathrm{Prob}[b(n) = \{..\underbrace{00..0}_{\text{length } l}1..\}] &= \Pr(S_1 \mid S_0)\cdot\left[1 - \Pr(S_1 \mid S_0)\right]^{l-1} \\ \mathrm{Prob}[b(n) = \{..\underbrace{11..1}_{\text{length } l}0..\}] &= \Pr(S_0 \mid S_1)\cdot\left[1 - \Pr(S_0 \mid S_1)\right]^{l-1}. \end{aligned} \tag{2.164}$$

These probabilities decay exponentially by increasing length l. Successive binary samples would be statistically independent for the case where $\Pr(S_0|S_1)=\Pr(S_0)$ and $\Pr(S_1|S_0)=\Pr(S_1)$. Markov chains with more than two states can be defined accordingly, where again the full set of transition probabilities between all states suffices to define the model. This general formulation of the Markov chain transitions for a model of J states can be written as an extension of (2.162)

$$\begin{bmatrix} \Pr(S_0) \\ \Pr(S_1) \\ \vdots \\ \Pr(S_{J-1}) \end{bmatrix} = \begin{bmatrix} \Pr(S_0 \mid S_0) & \Pr(S_0 \mid S_1) & \cdots & \Pr(S_0 \mid S_{J-1}) \\ \Pr(S_1 \mid S_0) & \Pr(S_1 \mid S_1) & & \vdots \\ \vdots & & \ddots & \\ \Pr(S_{J-1} \mid S_0) & \cdots & & \Pr(S_{J-1} \mid S_{J-1}) \end{bmatrix} \cdot \begin{bmatrix} \Pr(S_0) \\ \Pr(S_1) \\ \vdots \\ \Pr(S_{J-1}) \end{bmatrix}. \tag{2.165}$$

Due to the Markov property, the probability of transition into a state only depends on *one previous state*, such that for the binary 2-state model

$$\begin{aligned} &\mathrm{Prob}\left[b(n) = S_i \mid b(n-1) = S_j, b(n-2) = S_k, b(n-3) = S_l, ...\right] \\ &\quad = \Pr\left(S_i \mid S_j, S_k, S_l, ...\right) = \Pr\left(S_i \mid S_j\right). \end{aligned} \tag{2.166}$$

If a binary sequence $b(n)$ shall be defined where the state of a sample depends on *two previous samples*, the transition probabilities have to be expressed as $\Pr(S_i|(S_j, S_k))$. It is then necessary to define a Markov chain with four states, relating to the four configurations of [$b(n-1)=S_j$, $b(n-2)=S_k$]. As however the current [$b(n)$, $b(n-1)$] will become [$b(n-1)$, $b(n-2)$] in the follow-up state, certain state transitions are impossible, which can be implemented by assigning zero as transition probability. A state diagram related to this given case is shown in Fig. 2.24b. This model straightforwardly extends to the case where a sample $b(\mathbf{n})$ is conditioned by a K-dimensional vector $\mathbf{b}$ of previous values, which can be established from a one- or multi-dimensional neighborhood context $\mathcal{C}(\mathbf{n})$ of K members, not including the current position. The model will then be based on 2^K different states, and is fully described by 2^{K+1} state transitions

$$\Pr\left(b(\mathbf{n})=\beta\,|\,\mathbf{b}\right) \quad ; \quad \mathbf{b}=\left\{\left[b(\mathbf{i})\right]\,|\,\mathbf{i}\in\mathcal{C}(\mathbf{n});\mathbf{i}\neq\mathbf{n}\right\} \quad ; \quad \beta\in\{0,1\}\,. \tag{2.167}$$

However, only 2^K state transitions are freely selectable, as in the example above, if the current sample would become member of **b** in the next step.

$$\Pr\left(b(\mathbf{n})=0\,|\,\mathbf{b}\right)=1-\Pr\left(b(\mathbf{n})=1\,|\,\mathbf{b}\right). \tag{2.168}$$

The follow-up state can also be constrained by zero-probability transitions, as in the example above, ruled by the fact that certain values are not independent.

Even though in the cases discussed so far the number of states is finite, the complete sequences $b(n)$ or $b(\mathbf{n})$ can be regarded as infinite. If a Markov chain model allows a transition with a non-zero probability from any state to any other state within a finite number of steps, it is said to be *irreducible*. This would not be the case for chains where one or several states S_i exist with *all* outgoing transition probabilities $\Pr(S_j|S_i)=0$, but at least one incoming transition probability $\Pr(S_i|S_j)>0$. This S_i will be a terminating state which once reached can never again be left. Such models can be useful in cases where finite sequences with expected termination shall be modeled.

2.5.5 Statistical foundation of information theory

Considerations about certainty and uncertainty of an information establish the foundations of *information theory*. In general, sending an information intends reducing the *uncertainty* about an event, a letter from a text, the state of a signal etc. Assume a discrete set $\boldsymbol{S}$ is given, characterizing J possible states S_j of an event. Each state shall have a probability $\Pr(S_j)$. The goal is to define a measure for the information $I(S_j)$ which is related to the knowledge that the event would be in state S_j. Consequently, the mean of information over all states will be $H(\boldsymbol{S})=\Sigma\,\Pr(\mathrm{S}_j)I(S_j)=\mathcal{E}\{I(S_j)\}$. Availability of *complete information* means that any uncertainty is removed about the state of the event. The function $H(\boldsymbol{S})$ shall retain its consistency if the amount of certainty is varied, e.g. if it stays uncertain whether the state is S_0 or S_1, while it is already certain that the state will not be $S_2\ldots S_{J-1}$. Assuming that the $I(S_j)$ shall be related to the probabilities of the states, the following condition must be observed:

$$\begin{aligned} H(S) &= H\left\{\Pr(S_0),\Pr(S_1),\Pr(S_2),\ldots,\Pr(S_{J-1})\right\} \\ &\overset{!}{=} H\left\{\Pr(S_0)+\Pr(S_1),\Pr(S_2),\ldots,\Pr(S_{J-1})\right\} \\ &\quad +\left(\Pr(S_0)+\Pr(S_1)\right)H\left\{\frac{\Pr(S_0)}{\Pr(S_0)+\Pr(S_1)},\frac{\Pr(S_1)}{\Pr(S_0)+\Pr(S_1)}\right\}. \end{aligned} \tag{2.169}$$

If (2.169) is valid, an arbitrary separation of the information into certainty and uncertainty about any of the states of the event is possible. It can be shown that the only function fulfilling (2.169) is the *self information* of a discrete event of state S_j from the set $\boldsymbol{S}$ defined as[31]

$$I(S_j) = \log_2 \frac{1}{\Pr(S_j)} = -\log_2 \Pr(S_j)\,. \tag{2.170}$$

The mean value of the self information over all possible states is denoted as the *entropy*

$$H(\boldsymbol{S}) = -\sum_{j=0}^{J-1} \Pr(S_j) \log_2 \Pr(S_j)\,. \tag{2.171}$$

If two distinct events defined over sets $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ occur, their *joint information* and *joint entropy* can be defined via the joint probability

$$H(\boldsymbol{S}_1,\boldsymbol{S}_2) = \sum_{j_1=0}^{J_1-1}\sum_{j_2=0}^{J_2-1} \Pr(S_{j_1},S_{j_2}) I(S_{j_1},S_{j_2}) \quad \text{with } I(S_{j_1},S_{j_2}) = -\log_2 \Pr(S_{j_1},S_{j_2}). \tag{2.172}$$

The joint entropy is lower and upper bounded by

$$\max\{H(\boldsymbol{S}_1),H(\boldsymbol{S}_2)\} \le H(\boldsymbol{S}_1,\boldsymbol{S}_2) \le H(\boldsymbol{S}_1)+H(\boldsymbol{S}_2)\,, \tag{2.173}$$

where the upper bound is valid in the case of statistical independence of the two events, and the lower bound applies if they always come with identical joint occurrence of states. The concept of conditional probability defines the probability of an event in $\boldsymbol{S}_2$ to be in state S_{j_2}, provided that the state S_{j_1} of the other event in $\boldsymbol{S}_1$ is given. This allows reflecting the statistical dependency of state S_{j_2} from S_{j_1} in terms of the remaining uncertainty in the *conditional information*

$$I(S_{j_2}\,|\,S_{j_1}) = -\log_2 \Pr(S_{j_2}\,|\,S_{j_1}) = -\log_2 \frac{\Pr(S_{j_1},S_{j_2})}{\Pr(S_{j_1})}\ . \tag{2.174}$$

For statistically independent events, due to (2.134) and (2.135) $\Pr(S_{j_2}|S_{j_1})=\Pr(S_{j_2})$, which makes the conditional information identical to the self information $I(S_{j_2})$. The difference between the self information and the conditional information is the *mutual information*. It signifies the amount of information in state S_{j_2}, which was already provided by the state S_{j_1}. Likewise, this can be interpreted as the amount of information which could possibly be saved (e.g. needs not to be encoded or transmitted) when the statistical dependency is exploited:

[31] In (2.170), any base of the logarithm can be selected, where the unit of information is 'bit' in case of base 2 (counting the amount of binary digits). A probability $P(S_j)$=0 would lead to an infinite self information; in the subsequent definitions of entropy this is not a problem, as $\lim_{x\to 0}\left(x\cdot\log\frac{1}{x}\right)=0$.

$$I(S_{j_2};S_{j_1}) = I(S_{j_2}) - I(S_{j_2}\left|S_{j_1}\right.) . \tag{2.175}$$

Combining (2.170) and (2.174) into (2.175), further considering (2.135) gives

$$I(S_{j_2};S_{j_1}) = \log_2 \frac{\Pr(S_{j_2}\left|S_{j_1}\right.)}{\Pr(S_{j_2})} = \log_2 \frac{\Pr(S_{j_1},S_{j_2})}{\Pr(S_{j_1})\cdot\Pr(S_{j_2})} = \log_2 \frac{\Pr(S_{j_1}\left|S_{j_2}\right.)}{\Pr(S_{j_1})} = I(S_{j_1};S_{j_2}) . \tag{2.176}$$

This shows the symmetry property of mutual information. If two events are statistically independent, the mutual information becomes zero in all states. This is an ultimate condition for statistical independency, which even allows to test for presence or absence of nonlinear dependencies, being a more rigid criterion than the cross correlation. The mean of conditional information over all state combinations S_{j_1} and S_{j_2} is the *conditional entropy*[32],

$$\begin{aligned} H(\boldsymbol{S}_2\left|\boldsymbol{S}_1\right.) &= -\sum_{j_1=0}^{J_1-1}\sum_{j_2=0}^{J_2-1} \Pr(S_{j_1})\Pr(S_{j_2}\left|S_{j_1}\right.)\log_2 \Pr(S_{j_2}\left|S_{j_1}\right.) \\ &= -\sum_{j_1=0}^{J_1-1}\sum_{j_2=0}^{J_2-1} \Pr(S_{j_1},S_{j_2})\log_2 \Pr(S_{j_2}\left|S_{j_1}\right.). \end{aligned} \tag{2.177}$$

The *mean of mutual information* can also be expressed from (2.175) and (2.176) as follows[33]:

$$\begin{aligned} H(\boldsymbol{S}_2;\boldsymbol{S}_1) = H(\boldsymbol{S}_1;\boldsymbol{S}_2) &= \sum_{j_1=0}^{J_1-1}\sum_{j_2=0}^{J_2-1} \Pr(S_{j_1},S_{j_2})\log_2 \frac{\Pr(S_{j_1},S_{j_2})}{\Pr(S_{j_1})\Pr(S_{j_2})} \\ &= H(\boldsymbol{S}_2) - H(\boldsymbol{S}_2\left|\boldsymbol{S}_1\right.) = H(\boldsymbol{S}_1) - H(\boldsymbol{S}_1\left|\boldsymbol{S}_2\right.). \end{aligned} \tag{2.178}$$

The general relationships between entropy, conditional entropy and mean of mutual information are shown in Fig. 2.25a by a diagram of information flow. In principle, the whole schema is invertible, i.e. the states S_{j_1} and S_{j_2} can change their roles, while the mutual information will not change. In addition, Fig. 2.25b shows an interpretation borrowed from set algebra, where the circles indicate the total amount of information from the events defined by $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. The intersection is the mean of mutual information which is shared, such that at least some statistical dependency between the two events must be in effect.

Entropy, conditional entropy and mutual information can be used to express the problem of encoding information by *discrete alphabets*. Typical examples of

[32] If $H(\boldsymbol{S}_2|\boldsymbol{S}_1) < H(\boldsymbol{S}_2)$ and the state of $\boldsymbol{S}_1$ is known at the decoder, it is usually possible to reduce the data rate by utilizing this prior information. This is the basis of predictive coding (see Sec. 5.2) and context-dependent entropy coding (see Sec. 4.4.5).

[33] (2.178) also is often by itself denoted as *mutual information.* Consequently, it should be called mutual entropy, but this is hardly established.

discrete alphabets are finite sets of alphanumeric letters, or sets of reconstruction values in case of signal quantization. Let a source alphabet $\mathcal{A}$ be defined, which contains all distinct letters that a discrete source could ever produce. Further, a reconstruction alphabet $\mathcal{B}$ is given. Both alphabets need not necessarily be identical (however only if $\mathcal{A}$ is identical with $\mathcal{B}$ or a subset thereof, it is possible at all to perform lossless coding and decoding). The mapping of values from $\mathcal{A}$ into values from $\mathcal{B}$ is defined by a code $\mathcal{C}$. Then,

$$H(\mathcal{A};\mathcal{B})\big|_{e} = H(\mathcal{A}) - H(\mathcal{A}\,|\,\mathcal{B})\big|_{e} \tag{2.179}$$

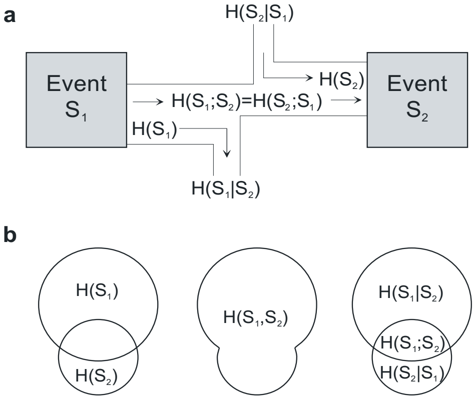

Fig. 2.25. Graphical interpretation of information-theoretic statistical parameters: **a** in terms of information flow **b** in terms of set algebra

As the mutual information cannot become negative,

$$0 \le H(\mathcal{A}\,|\,\mathcal{B})\big|_{e} \le H(\mathcal{A}), \tag{2.180}$$

where for $H(\mathcal{A}\,|\,\mathcal{B})\big|_{e} = 0$ it is possible to perform lossless decoding, while for $H(\mathcal{A}\,|\,\mathcal{B})\big|_{e} = H(\mathcal{A})$ *nothing is known* after decoding about the state of the source. For any values of $H(\mathcal{A}\,|\,\mathcal{B})\big|_{e}$ between these extremes, lossy decoding will be in effect, such that *distortion* occurs. Let $\mathcal{C}_D$ define the set of all codes, which are capable to perform the mapping from $\mathcal{A}$ onto $\mathcal{B}$ by effecting a given value of distortion D[34]. The best possible code among all $\mathcal{C}_D$ is the one which needs lowest rate for its representation, which is the code requiring least mutual information when the mapping from $\mathcal{A}$ into $\mathcal{B}$ is performed with that distortion. The lowest bound for the rate by a given distortion D will then be

[34] At this point, D shall be introduced in a quite abstract way, more concrete definitions will be used in Ch. 4.

$$R(D) = \min_{e \in \mathcal{e}_D} H(\mathcal{A};\mathcal{B})\big|_{e} \; . \tag{2.181}$$

This definition, however, does not indicate a direct method to design codes, only allows judging their performance. $R(D)$ is the *rate distortion function* (RDF), which defines an interrelationship between rate R and distortion D. In this abstract form, the definition is valid for arbitrary source alphabets and arbitrary definitions of distortion. From (2.179)-(2.181) and the related reasoning, the following conclusions can be drawn:

- Lossless coding of a source generating letters from a discrete source alphabet $\mathcal{A}$, can only be achieved by investing a minimum rate $R_{\min} = H(\mathcal{A})$.
- The minimum rate is zero, where at the decoder *nothing* would be known about the state of the source. In this case, a maximum distortion $D_{\max}$ occurs which should never be superseded at any positive rate.
- If the source has continuous amplitude, the number of letters in the source alphabet $\mathcal{A}$ would grow towards infinity. Hence, it is not possible to achieve zero distortion (lossless encoding) using a finite rate. If the reconstruction alphabet $\mathcal{B}$ is sufficiently large, the distortion may however become negligibly small.

Qualitative graphs of rate distortion functions for both cases of continuous-amplitude and discrete-amplitude sources are shown in Fig. 2.26[35]. Typically, the rate distortion function is convex and continuously decreasing until the maximum distortion (for rate zero) is reached.

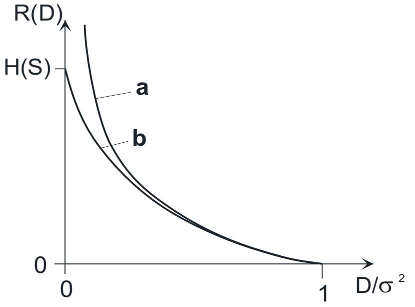

Fig. 2.26. Examples of $R(D)$ for sampled continuous (**a**) and discrete (**b**) sources

Example: Entropy of a Markov process. A Markov chain of J states is defined by the transition probability definitions in (2.165). Due to the property that the probabilities of next-state transitions are independent of history, the entropy of each

[35] Here, the distortion is expressed in terms of squared error, i.e. Euclidean distance, and normalized by $D_{\max} = \sigma_s^2$, which occurs in case of zero reconstruction.

state can first be defined independently by the respective probabilities of the next-state transitions

$$H(S_j) = -\sum_{i=0}^{J-1} \Pr(S_i \mid S_j)\log_2 \Pr(S_i \mid S_j) \quad ; \quad j = 0,...,J-1 . \tag{2.182}$$

The overall entropy of the Markov process can then be computed as the probability-weighted average over all states,

$$H(S) = \sum_{j=0}^{J-1} \Pr(S_j)H(S_j) = -\sum_{i=1}^{J}\sum_{j=1}^{J} \Pr\left(S_i, S_j\right)\log_2 \Pr\left(S_i \mid S_j\right) . \tag{2.183}$$

Differential entropy and entropy of Gaussian processes. The concept of entropy can be extended in relation to the PDF of continuous-amplitude sources. However, in principle the number of bits necessary to represent a continuous source (and therefore its entropy) would be infinite. With uniform quantization using intervals $\left[(i-1/2)\Delta;(i+1/2)\Delta\right]$, the PMF $p_s(i)$ is according to (2.122)

$$p_s(i) = \int_{(i-1/2)\Delta}^{(i+1/2)\Delta} p_s(x)\,\mathrm{d}x \approx \Delta p_s(i\Delta) . \tag{2.184}$$

The entropy of the discrete distribution for Δ→0 becomes

$$\begin{aligned} H_s &= \lim_{\Delta\to 0}\left(-\sum_{i=-\infty}^{\infty} p_s(i)\log p_s(i) \right) = \lim_{\Delta\to 0}\left(-\sum_{i=-\infty}^{\infty} p_s(i)\log\left(p_s(i\Delta)\Delta\right) \right) \\ &= \lim_{\Delta\to 0}\left(-\sum_{i=-\infty}^{\infty} p_s(i\Delta)\Delta\log\left(p_s(i\Delta)\right) - \sum_{i=-\infty}^{\infty} p_s(i)\log\left(\Delta\right) \right) \\ &= -\int_{-\infty}^{\infty} p_s(x)\log p_s(x)\,\mathrm{d}x - \lim_{\Delta\to 0}\log(\Delta). \end{aligned} \tag{2.185}$$

Whereas the term –log(Δ) converges towards infinity for Δ→0 and is independent of the PDF, the left term is denoted as the *differential entropy*[36],

$$H_s = -\int_{-\infty}^{\infty} p_s(x)\log p_s(x)\,\mathrm{d}x . \tag{2.186}$$

(2.186) cannot be used as an information-theoretic criterion about the quantitative amount of information contained in a source, and the value could even become negative. It is however useful for comparing properties of PDFs or in optimization. All other variants such as joint, vector and conditional entropies can be defined similarly.

Specifically taking the natural logarithm (base e), the differential entropy of a zero-mean Gaussian process is (using the unit '*nat*' which refers to a symbol count based on Euler's number instead of the binary number system)

[36] Note that H_s as defined in (2.186) cannot quantitatively be compared against (2.171).

$$\begin{aligned} H_s &= -\int_{-\infty}^{\infty} p_s(x)\left[-\frac{x^2}{2\sigma_s^2} - \log\sqrt{2\pi\sigma_s^2}\right]\mathrm{d}x \\ &= \mathcal{E}\left\{\frac{x^2}{2\sigma_s^2}\right\} + \frac{1}{2}\log\left(2\pi\sigma_s^2\right) = \frac{1}{2}\log\left(2\pi\mathrm{e}\,\sigma_s^2\right) \quad \text{[nats]} \end{aligned} . \qquad (2.187)$$

Similarly, the extension to a K-dimensional vector Gaussian process gives

$$H_{\mathbf{s}} = \frac{1}{2}\log\left(\left(2\pi\mathrm{e}\right)^K \left|\mathbf{C}_{\mathbf{ss}}\right|\right) \le KH_s \quad \text{[nats]}, \qquad (2.188)$$

which provides a quantitative expectation that the entropy for the correlated source will be lower than the K-fold entropy of single samples.

2.6 Linear prediction

2.6.1 Autoregressive models

Algorithms of multimedia signal processing often require a model about the statistical properties of sources for analytic optimization (cf. Sec. 3.4). If statistical assumptions are made which go beyond sample statistics, modeling of statistical dependencies between samples is required. The autocovariance is usually sufficient to optimize *linear systems* for a given purpose, as it characterizes *linear statistical dependencies* between samples.

A random signal of an *autoregressive (AR) process* (Fig. 2.27) is generated by a recursive filter with $\mathbf{z}$ transfer function $B(\mathbf{z})=1/(1-H(\mathbf{z}))$ from a stationary white Gaussian noise process $v(\mathbf{n})$ as input. The process $s(\mathbf{n})$ at the filter output possesses spectral distribution properties which are only determined by the amplitude transfer function of the filter. The PDF of this stationary process is also Gaussian.

The property of stationarity does not usually apply to multimedia sources. Moreover, a high degree of variation is observed in the local properties of image, speech and audio signals, such that a local adaptation of model parameters is typically necessary. Even then, the AR model helps to simplify problems of optimization due to its simple analytic properties. If the AR model generates a stationary Gaussian process, it is indeed fully described by a covariance matrix. In this case, an AR process would perfectly follow the vector Gaussian PDF (2.156).

The autoregressive model of first order [AR(1)] is often used to model the global statistics of image signals. For the 1D case, a Gaussian zero-mean white-noise process $v(n)$ (*innovation*) of variance σ_v^2 is fed into a recursive filter with z transfer function

$$B(z)=\frac{1}{1-\rho z^{-1}}\,. \tag{2.189}$$

The computation of the output is

$$s(n)=\rho s(n-1)+v(n)\,. \tag{2.190}$$

The AR(1) process has an autocovariance function[37]

$$\mu_{ss}(k)=\sigma_s{}^2\rho^{|k|} \quad ; \quad \sigma_s{}^2=\frac{\sigma_v{}^2}{1-\rho^2}\,, \tag{2.191}$$

and a power density spectrum

$$\phi_{ss}(f)=\sigma_s{}^2\sum_{k=-\infty}^{\infty}\rho^{|k|}e^{-j2\pi fk}=\frac{\sigma_s{}^2(1-\rho^2)}{1-2\rho\cos\left(2\pi f\right)+\rho^2}\,. \tag{2.192}$$

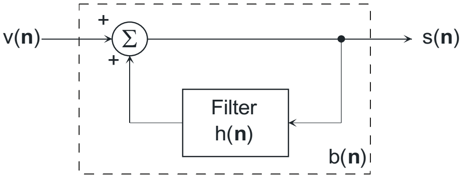

Fig. 2.27. System for generating samples from an autoregressive process

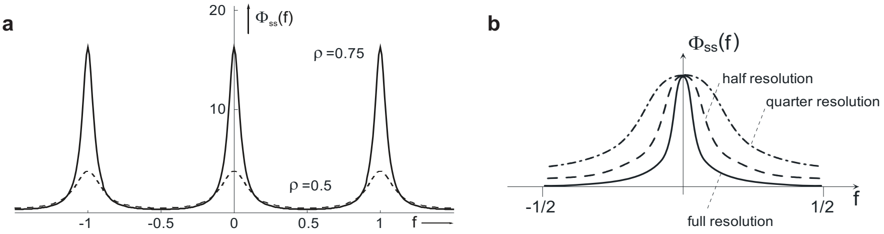

Fig. 2.28. a Power density spectra of AR(1) processes with $\sigma_s{}^2=1$, for two different values of $\rho(1)$ **b** Effect of decreasing sampling resolution by factors $U=2$ and $U=4$.

With zero-mean input, also the autoregressive output of the filter has zero-mean property[38]. Obviously, the AR(1) model is fully characterized by the filter parameter ρ, which is identical to the standardized autocovariance coefficient $\rho_{ss}(1)$, and one of the variances, σ_v^2 or σ_s^2. Typical values of $\rho(1)$ for natural images are between .85 and .99, which exhibit extreme concentration of spectral power

[37] For a proof on (2.191) and (2.192), see Problem 2.9.

[38] Optionally, a mean value can be added either at the input or at the output, with $m_s=m_v\cdot B(\mathbf{z})\big|_{\mathbf{z}=\mathbf{1}}$.

around the zero frequency. Examples with lower values of $\rho(1)$=.75 and .5 are shown in Fig. 2.28a. It should be observed that the measurement of the correlation parameter ρ which is used to adapt an AR(1) process also depends on the sample density (resolution of the signal). If possible alias effects are ignored, downsampling the ACF by a factor of U leads to a modification into $\rho_U(1)=\rho(U)=\rho(1)^U$. The effect of increasing high-frequency components in the power density spectrum is illustrated in Fig. 2.28b.

For simple extensions of the AR(1) model into two and multiple dimensions, expression by separate standardized autocovariance coefficients $\rho_1 \equiv \rho_1(1)$ and $\rho_2 \equiv \rho_2(1)$ for horizontal and vertical directions can be used. Properties of three 2D methods are illustrated in Fig. 2.29, showing lines of constant autocovariance in the (m_1,m_2) plane (only positive values of ρ are assumed here).

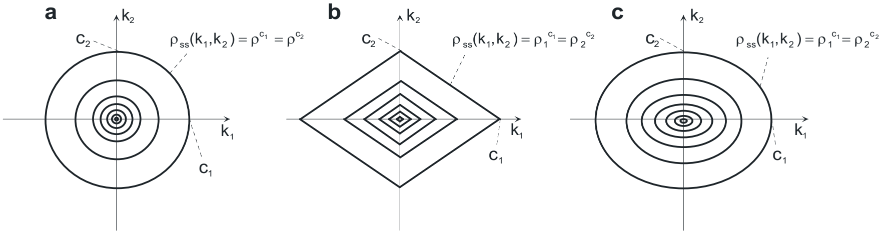
a
k_2
c_2
$\rho_{ss}(k_1,k_2)=\rho^{c_1}=\rho^{c_2}$
k_1
c_1
b
k_2
c_2
$\rho_{ss}(k_1,k_2)=\rho_1^{c_1}=\rho_2^{c_2}$
k_1
c_1
c
k_2
c_2
$\rho_{ss}(k_1,k_2)=\rho_1^{c_1}=\rho_2^{c_2}$
k_1
c_1

Fig. 2.29. Lines of constant autocovariance in 2D AR(1) models.
a isotropic **b** separable **c** elliptic

The *isotropic* model has an autocovariance function

$$\varphi_{ss}(m_1,m_2)=\sigma_s^{\,2}\rho^{\sqrt{m_1^{\,2}+m_2^{\,2}}}\,, \tag{2.193}$$

expressing circular-symmetric values independent of the direction, $\rho_1=\rho_2$ is inherently assumed. Constant values appear on circles of radius $|\mathbf{m}|=\sqrt{m_1^{\,2}+m_2^{\,2}}$ (see Fig. 2.29a). The two-dimensional power density spectrum of the isotropic model is then also circular-symmetric[39],

$$\phi_{ss}(f_1,f_2)=\frac{\sigma_s^{\,2}(1-\rho^2)}{1-2\rho\cos\left(2\pi\sqrt{f_1^2+f_2^2}\right)+\rho^2}\,. \tag{2.194}$$

For the remaining models, autocovariance values are defined differently for the horizontal and vertical directions. In natural images, it can be observed that autocovariance statistics sometimes differ per orientation. It is often found that the

[39] Note that this is not fully precise due to the fact that the nearest periodic copies of the spectrum are only present at some angular orientations. The best coincidence would be found for the case of hexagonal sampling, or for $\rho\to 1$.

covariance along the vertical axis is lower than along the horizontal axis. The *separable model* with autocovariance function

$$\varphi_{ss}(m_1,m_2)=\sigma_s^{\,2}\rho_1^{|m_1|}\rho_2^{|m_2|} \quad ; \quad \sigma_s^{\,2}=\frac{\sigma_v^{\,2}}{(1-\rho_1^{\,2})(1-\rho_2^{\,2})}\,, \tag{2.195}$$

shows straight lines of constant autocovariance[40]. These lines intersect with axes m_1 and m_2 at positions m_1' and m_2' where $\rho_1^{|m_1'|}=\rho_2^{|m_2'|}$ (see Fig. 2.29b). The generation of the discrete 2D signal can be implemented by a separable recursive filter, whose output is expressed by the equation

$$s(n_1,n_2)=\rho_1 s(n_1-1,n_2)+\rho_2 s(n_1,n_2-1)-\rho_1\rho_2 s(n_1-1,n_2-1)+v(n_1,n_2)\,. \tag{2.196}$$

The related power density spectrum is

$$\phi_{ss}(f_1,f_2)=\sigma_s^{\,2}\frac{1-\rho_1^{\,2}}{1-2\rho_1\left(\cos 2\pi f_1\right)+\rho_1^2}\cdot\frac{1-\rho_2^{\,2}}{1-2\rho_2\left(\cos 2\pi f_2\right)+\rho_2^2}\,. \tag{2.197}$$

The *elliptic model* has an autocovariance function (for $0<\rho_i<1$)

$$\varphi_{ss}(m_1,m_2)=\sigma_s^{\,2}\mathrm{e}^{-\sqrt{(\beta_1 m_1)^2+(\beta_2 m_2)^2}} \quad \text{with } \beta_i=-\ln\rho_i \quad . \tag{2.198}$$

It shows constant autocovariance values on elliptic shapes due to the ellipse equation in the exponent (Fig. 2.29c). This model can also be interpreted as an extension of the isotropic model, which would be a special case $\beta_1=\beta_2$. Intersections of constant autocovariance graphs with the coordinate axes are identical to the case of the separable model. All models introduced so far can be interpreted as special cases of a generalized 2D AR(1) model with autocovariance[41]

$$\varphi_{ss}(m_1,m_2)=\sigma_s^{\,2}\cdot e^{-\left[\left(\beta_1\cdot|m_1|\right)^{\gamma}+\left(\beta_2\cdot|m_2|\right)^{\gamma}\right]^{\frac{1}{\gamma}}}\,. \tag{2.199}$$

For the isotropic and for the elliptic model, $\gamma=2$; for the separable model $\gamma=1$. When $\gamma>1$, lines of constant autocovariance are convex over all the four quadrants of m_i coordinates. The intersections with the axes remain identical as in the case of separable and elliptic models, irrespective of γ. The relation of factors β_i with the horizontal/vertical autocovariance coefficients is equal to (2.198).

Autoregressive models of higher order (more than one autocovariance value per dimension) are frequently applied in speech analysis and for texture analysis of images. As those signals typically do not have the property of stationarity, the

[40] If the two exponential expressions in (2.195) are modified for a common basis, a line equation over absolute values appears in the exponent, see (2.199) with γ=1.

[41] This model is, like the elliptic model, only applicable for positive ρ_1 and ρ_2 values. In both cases, a precise analytic expression of the power density spectrum is difficult to define due to the directional dependent alias effect.

autocovariance function needs to be estimated over segments (finite time windows or 2D regions) of samples. A generic synthesis equation expressing AR filtering over a finite causal neighborhood $\mathcal{N}_{\mathbf{p}}^{+}$ is[42]

$$s(\mathbf{n}) = \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})s(\mathbf{n}-\mathbf{p}) + v(\mathbf{n})\,. \tag{2.200}$$

With white noise input, the output process has a power density spectrum

$$\phi_{ss}(\mathbf{f}) = \frac{\sigma_v^{\;2}}{\left|1-\sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})\,\mathrm{e}^{-j2\pi\mathbf{f}^{\mathrm{T}}\mathbf{p}}\right|^2}\,. \tag{2.201}$$

Next, a causal model shall be optimized under the assumption that the white-noise signal $v(n)$ which is to be fed into the AR synthesis filter shall have lowest possible variance:

$$\begin{aligned}\sigma_v^{\;2} &= \mathcal{E}\left\{v^2(\mathbf{n})\right\} = \mathcal{E}\left\{\left[s(\mathbf{n}) - \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})s(\mathbf{v}-\mathbf{p})\right]^2\right\} \\ &= \mathcal{E}\left\{s^2(\mathbf{n})\right\} - 2\mathcal{E}\left\{\left[s(\mathbf{n})\sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})s(\mathbf{n}-\mathbf{p})\right]\right\} + \mathcal{E}\left\{\left[\sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})s(\mathbf{n}-\mathbf{p})\right]^2\right\} \overset{!}{=} \min.\end{aligned} \tag{2.202}$$

The minimization is achieved by computing partial derivatives over each filter coefficient:

$$\frac{\partial\,\sigma_v^{\;2}}{\partial\,a(\mathbf{k})} \overset{!}{=} 0 \Rightarrow \mathcal{E}\left\{s(\mathbf{n})s(\mathbf{n}-\mathbf{k})\right\} = \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})\mathcal{E}\left\{s(\mathbf{n}-\mathbf{p})s(\mathbf{n}-\mathbf{k})\right\}. \tag{2.203}$$

This the linear *Wiener-Hopf equation system*, where the optimum filter coefficients fulfill the condition

$$\mu_{ss}(\mathbf{k}) = \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}^{+}} a(\mathbf{p})\mu_{ss}(\mathbf{k}-\mathbf{p})\,, \tag{2.204}$$

or specifically for the 1D case with order P:

[42] Observe that the definitions (2.201) and (2.202) do not implicitly postulate the *causality* of the AR synthesis filters. In fact, all following deductions can likewise be made for non-causal filter sets without any limitation. Non-causal recursive filtering is indeed practically applicable for signals of finite extension, e.g. image signals. Only the current position **n** must be excluded, which means that $a(\mathbf{0})=0$. For more detail on non-causal AR modeling of images, see e.g. [Jain 1989].

$$\mu_{ss}(k)=\sum_{p=1}^{P}a(p)\mu_{ss}(k-p) \quad ; \quad 1\le k\le P\,. \tag{2.205}$$

Due to the symmetry of the autocovariance, $\mu_{\mathrm{ss}}(k-p)=\mu_{\mathrm{ss}}(p-k)$, the problem can be simplified in a more regular matrix structure, and the Wiener-Hopf equation can be written as follows, where $\mathbf{C}_{ss}$ is the autocovariance matrix (2.157):

$$\underbrace{\begin{bmatrix}\mu_{ss}(1)\\ \mu_{ss}(2)\\ \vdots\\ \vdots\\ \mu_{ss}(P)\end{bmatrix}}_{\mathbf{c}_{ss}}=\underbrace{\begin{bmatrix}\mu_{ss}(0) & \mu_{ss}(1) & \cdots & \cdots & \mu_{ss}(P-1)\\ \mu_{ss}(1) & \mu_{ss}(0) & \mu_{ss}(1) & \cdots & \mu_{ss}(P-2)\\ \vdots & \mu_{ss}(1) & \mu_{ss}(0) & \ddots & \vdots\\ \vdots & \vdots & \ddots & \ddots & \mu_{ss}(1)\\ \mu_{ss}(P-1) & \mu_{ss}(P-2) & \cdots & \mu_{ss}(1) & \mu_{ss}(0)\end{bmatrix}}_{\mathbf{C}_{ss}}\underbrace{\begin{bmatrix}a(1)\\ a(2)\\ \vdots\\ \vdots\\ a(P)\end{bmatrix}}_{\mathbf{a}}\,. \tag{2.206}$$

The solution is obtained when the vector $\mathbf{c}_{ss}$ and the matrix $\mathbf{C}_{ss}$ are filled by the autocovariance estimates (computed either locally from a given signal or globally from an ensemble)[43], and then the matrix is inverted:

$$\mathbf{a}=\hat{\mathbf{C}}_{ss}{}^{-1}\hat{\mathbf{c}}_{ss}\,. \tag{2.207}$$

In the 1D case, $\mathbf{C}_{ss}$ has a *Toeplitz structure*, which means that it is diagonally symmetric (identical to its transpose), and values on each one of the diagonals (main diagonal and its off-diagonals) are identical. Even though the matrix is not sparse, it is highly regular, such that the problem of matrix inversion is simplified. This is mainly due to the fact that the lower and upper triangular matrices, as well as many sub-matrices are identical, such that sub-matrix inversions need to be computed only once. Furthermore, the matrix has full rank and is positive-definite (except for degenerate cases). Therefore, Cholesky decomposition or Levinson-Durbin recursion can be used for computationally more efficient solutions. Furthermore, the latter can be used to map the predictor into a ladder structure of subsequent first-order filters, where PARCOR (partial correlation) coefficients are determined step by step, guaranteeing stability of the synthesis filter under the simple rule that the absolute value of each coefficient in the structure must be smaller than one (see [RABINER, SCHAFER 1978] as reference for more detail on these approaches).

[43] When computing an autocovariance estimate for a finite segment, boundary conditions need to be observed. When using samples from the previous segment (as far as they are used to predict the samples from the current segment), this may lead to better prediction results, but also may cause instability of the resulting synthesis filter (see [RABINER, SCHAFER 1978], [MARAGOS, SCHAFER 1984] and the notes about positive-definiteness of autocovariance matrices under (2.210)).

The variance of the innovation signal for the case of a 1D AR(P) model is[44]

$$\begin{aligned}\sigma_v^2 &= \mathcal{E}\left\{v^2(n)\right\} = \mathcal{E}\left\{\left(s(n)-\sum_{p=1}^{P}a(p)s(n-p)\right)^2\right\} = \underbrace{\mathcal{E}\left\{\left(s(n)\right)^2\right\}}_{=\sigma_s^2} \\ &-2\sum_{p=1}^{P}a(p)\underbrace{\mathcal{E}\left\{s(n)s(n-p)\right\}}_{=\mu_{ss}(-p)}+\sum_{p=1}^{P}\sum_{q=1}^{P}a(p)a(q)\underbrace{\mathcal{E}\left\{s(n-p)s(n-q)\right\}}_{=\mu_{ss}(p-q)} \\ &= \sigma_s^2-2\sum_{p=1}^{P}a(p)\mu_{ss}(p)+\sum_{p=1}^{P}a(p)\underbrace{\sum_{q=1}^{P}a(q)\mu_{ss}(p-q)}_{=\mu_{ss}(p)\text{ acc. to W-H eq.}} = \sigma_s^2-\sum_{p=1}^{P}a(p)\mu_{ss}(p).\end{aligned} \tag{2.208}$$

This leads to an alternative formulation of the Wiener-Hopf equation, where the computation of the innovation signal variance is included by the first row of the matrix:

$$\sigma_v^2\delta(k)=\mu_{ss}(k)-\sum_{p=0}^{P}a(p)\cdot\mu_{ss}(k-p) \quad ; \quad 0\le k\le P$$

$$\underbrace{\begin{bmatrix}\sigma_v^2\\0\\ \vdots\\ \vdots\\0\end{bmatrix}}_{\mathbf{c}_{ss}}=\underbrace{\begin{bmatrix}\mu_{ss}(0)&\mu_{ss}(1)&\mu_{ss}(2)&\cdots&\mu_{ss}(P)\\ \mu_{ss}(1)&\mu_{ss}(0)&\mu_{ss}(1)&\cdots&\mu_{ss}(P-1)\\ \mu_{ss}(2)&\mu_{ss}(1)&\ddots&&\vdots\\ \vdots&\vdots&&\ddots&\vdots\\ \mu_{ss}(P)&\mu_{ss}(P-1)&\cdots&\cdots&\mu_{ss}(0)\end{bmatrix}}_{\mathbf{C}_{ss}}\underbrace{\begin{bmatrix}1\\-a(1)\\ \vdots\\ \vdots\\-a(P)\end{bmatrix}}_{\mathbf{a}}. \tag{2.209}$$

Furthermore, it can be concluded that

$$\sigma_v^2=\mathbf{a}^{\mathrm{T}}\mathbf{c}_{ss}=\mathbf{a}^{\mathrm{T}}\mathbf{C}_{ss}\mathbf{a}\,, \tag{2.210}$$

which means that the autocovariance matrix with Toeplitz structure has to be positive-definite, or at least positive semi-definite for the degenerate case of $\sigma_v^2=0$. This is guaranteed by the property $\mu_{ss}(0)\ge\left|\mu_{ss}(k)\right|$, which will also guarantee stability of the filter. When samples from a finite window are used to estimate the autocovariance values, the property of positive-definiteness would by guarantee be fulfilled if either a periodic extension of the window is used, or if zero values are padded outside of the window.

For a separable two- or multi-dimensional model, filter coefficients can be optimized independently by solving 1D Wiener-Hopf equations, using 1D autocovariance measurements over the different coordinate axes. However, separable models do not allow optimum adaptation for all properties of multidimensional signals. For example, in the two-dimensional case, characteristic diagonal orientations of autocovariance cannot be considered explicitly. When non-separable autocovariance functions are used, non-separable IIR filters must also be defined

[44] This is a generalization of (2.191) and also covers the AR(1) case.

as AR generator (or predictor) filters. Using the 2D autocovariance function, a 2D Wiener-Hopf equation can be defined as an extension of (2.209). For the case of a quarter-plane 2D filter, optimization gives

$$\sigma_v{}^2\delta(k_1,k_2) = \mu_{ss}(k_1,k_2) - \sum_{\substack{p_1=0 \\ (p_1,p_2)\neq(0,0)}}^{P_1} \sum_{p_2=0}^{P_2} a(p_1,p_2)\mu_{ss}(k_1-p_1,k_2-p_2)\,, \tag{2.211}$$

which can again be written as $\mathbf{c}_{ss} = \mathbf{C}_{ss}\mathbf{a}$. $\mathbf{C}_{ss}$ here is a *block Toeplitz matrix* [Dudgeon/Mersereau 1984]

$$\mathbf{C}_{ss} = \begin{bmatrix} \phi_0 & \phi_{-1} & \cdots & \cdots & \phi_{-P_2} \\ \phi_1 & \phi_0 & \cdots & \cdots & \phi_{1-P_2} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \phi_{P_2} & \phi_{P_2-1} & \cdots & \cdots & \phi_0 \end{bmatrix} \tag{2.212}$$

with associated sub-matrices

$$\phi_p = \begin{bmatrix} \mu_{ss}(0,p) & \mu_{ss}(-1,p) & \cdots & \cdots & \mu_{ss}(-P_1,p) \\ \mu_{ss}(1,p) & \mu_{ss}(0,p) & \cdots & \cdots & \mu_{ss}(-P_1+1,p) \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \mu_{ss}(P_1,p & \mu_{ss}(P_1-1,p) & \cdots & \cdots & \mu_{ss}(0,p) \end{bmatrix}. \tag{2.213}$$

The vector of coefficients is arranged by row-wise order

$$\mathbf{a} = \left[1,-a(1,0),..,-a(P_1,0),-a(0,1),..,-a(P_1,P_2)\right]^{\mathrm{T}}, \tag{2.214}$$

and the 'autocovariance vector' on the left side is

$$\mathbf{c}_{ss} = \left[\sigma_v{}^2,0,0,...,0\right]^{\mathrm{T}}. \tag{2.215}$$

The lengths of the vectors and the row/column lengths of the quadratic matrix are $(P_1+1)(P_2+1)$. The task is to determine the $(P_1+1)(P_2+1)-1$ unknown coefficients in $\mathbf{a}$. This is achieved as in (2.208), inverting the autocovariance matrix $\mathbf{C}_{ss}$. The full matrix of the 2D formulation does however no longer have a Toeplitz structure, because the sub-matrices (2.213) are not diagonally symmetric, since $\mu_{ss}(k,p)\neq\mu_{ss}(-k,p)$. As a consequence, the inversion cannot use the same efficient decomposition as in the 1D case, and also the number of covariance values to be used in the optimization is larger than the number of filter coefficients to be determined; therefore, a unique revertible mapping between model parameters and autocovariance does not longer exist. If positive-definiteness is violated, this may also lead to unstable synthesis filters. As an alternative, a 2D PARCOR structure was proposed in [Marzetta 1980]. However, it is reported that in the non-separable 2D case this does not guarantee stability, either.

2.6.2 Linear prediction

Autoregressive modeling of signals is closely related to *linear prediction*, where a *predictor filter* computes an estimate $\hat{s}(\mathbf{n})$ for the signal value $s(\mathbf{n})$. The difference is the *prediction error*

$$e(\mathbf{n}) = s(\mathbf{n}) - \hat{s}(\mathbf{n}) . \tag{2.216}$$

The signal can be reconstructed by using the prediction error and the estimate,

$$\tilde{s}(\mathbf{n}) \overset{!}{=} s(\mathbf{n}) = e(\mathbf{n}) + \hat{s}(\mathbf{n}) . \tag{2.217}$$

If estimates $\hat{s}(\mathbf{n})$ are exclusively computed by past values of the signal, the prediction error $e(\mathbf{n})$ also is a unique equivalent of $s(\mathbf{n})$[45]. The prediction is typically performed by an FIR filter with transfer function $H(\mathbf{z})$ (Fig. 2.30a); the *prediction error filter* (Fig. 2.30b), performing the operation described in (2.216), has a transfer function

$$A(\mathbf{z}) = 1 - H(\mathbf{z}) . \tag{2.218}$$

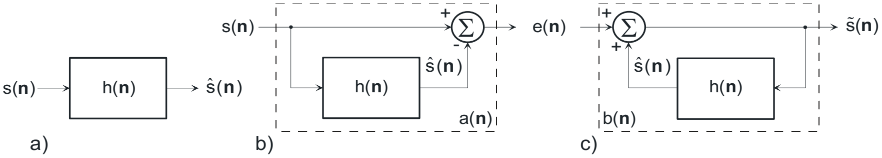

Fig. 2.30. System elements in linear prediction: **a** Predictor filter $h(\mathbf{n})$ **b** Prediction error filter (analysis filter) $a(\mathbf{n})$ **c** inverse prediction error filter (synthesis filter) $b(\mathbf{n})$

The *inverse prediction error filter* (synthesis filter, Fig. 2.30c) performs the operation (2.217). It is a recursive filter with transfer function

$$B(\mathbf{z}) = \frac{1}{A(\mathbf{z})} = \frac{1}{1 - H(\mathbf{z})} . \tag{2.219}$$

The filter (2.219) can be regarded equivalent to the synthesis filter of an AR model. Therefore, the prediction error signal would actually be Gaussian white noise if an AR process is optimally predicted (i.e. using a predictor which inverts the synthesis filter by which the process was generated). In the context of linear pre-

[45] In practical implementations, the equivalence may not be up to mathematical precision, when rounding errors occur. This can be avoided by performing systematic rounding as part of the prediction, which however would introduce a nonlinear element that can no longer be described as an LSI operation.

diction, the ratio of signal variance and prediction error variance is denoted as the *prediction gain*

$$G = \frac{\sigma_s^{\;2}}{\sigma_e^{\;2}}, \tag{2.220}$$

which can be determined from (2.208) for the case of an AR model.

Backward-adaptive prediction. Whereas the solution of the Wiener-Hopf equation assumes that autocovariance statistics either globally or of the current local segment is known, *backward-adaptive* methods of predictor filter adaptation use analysis of *past* samples under the assumption that the statistical properties are only slowly changing. The *least mean squares* algorithm (LMS) is often applied in this context [ALEXANDER, RAJALA 1985]. Predictor filter coefficients $a_{\mathbf{n}}(\mathbf{p})$ shall be used at the current position $\mathbf{n}$ in the prediction equation

$$\hat{s}(\mathbf{n}) = \sum_{\mathbf{p}} a_{\mathbf{n}}(\mathbf{p}) \cdot s(\mathbf{n}-\mathbf{p}) \quad \text{and} \quad e(\mathbf{n}) = s(\mathbf{n}) - \hat{s}(\mathbf{n}). \tag{2.221}$$

After computing the prediction error, it is evaluated how each filter coefficient would need to be modified to achieve a lower prediction error. The partial derivative of $e^2(\mathbf{n})$ over $a_{\mathbf{n}}(\mathbf{p})$ is

$$\frac{\partial e^2(\mathbf{n})}{\partial a_{\mathbf{n}}(\mathbf{p})} = -2e(\mathbf{n}) \cdot s(\mathbf{n}-\mathbf{p}), \tag{2.222}$$

such that an LMS update of coefficients to be used at the next position[46] reduces the prediction error by optimizing with regard to its negative gradient,

$$a_{\mathbf{n}+1}(p) = a_{\mathbf{n}}(\mathbf{p}) + \alpha\, e(\mathbf{n}) s(\mathbf{n}-\mathbf{p}). \tag{2.223}$$

The step size factor α influences the adaptation speed.

Two-dimensional prediction. The prediction equation in the case of a 2D quarter-plane predictor filter of order $(P_1+1)(P_2+1)-1$ is

$$\hat{s}(n_1,n_2) = \sum_{\substack{p_1=0 \\ (p_1,p_2)\neq(0,0)}}^{P_1} \sum_{p_2=0}^{P_2} a(p_1,p_2) s(n_1-p_1,n_2-p_2). \tag{2.224}$$

The z transfer function of this filter is

$$\begin{aligned} H(z_1,z_2) = {} & a(1,0)z_1^{\;-1} + \ldots + a(P_1,0)z_1^{\;-P_1} + a(0,1)z_2^{\;-1} + \ldots + a(P_1,1)z_1^{\;-P_1}z_2^{\;-1} \\ & + \ldots + a(0,P_2)z_2^{\;-P_2} + \ldots + a(P_1,P_2)z_1^{\;-P_1}z_2^{\;-P_2}. \end{aligned} \tag{2.225}$$

[46] The ‘next’ position at which the updated coefficient is used in 2D and multi-dimensional can be determined from the prediction direction, e.g. vertical down for a coefficient that performs vertical prediction from the sample above.

For the case of 2D signals, 2D prediction can be expected to better minimize the variance of the prediction error signal, as compared to 1D (horizontal or vertical) prediction.

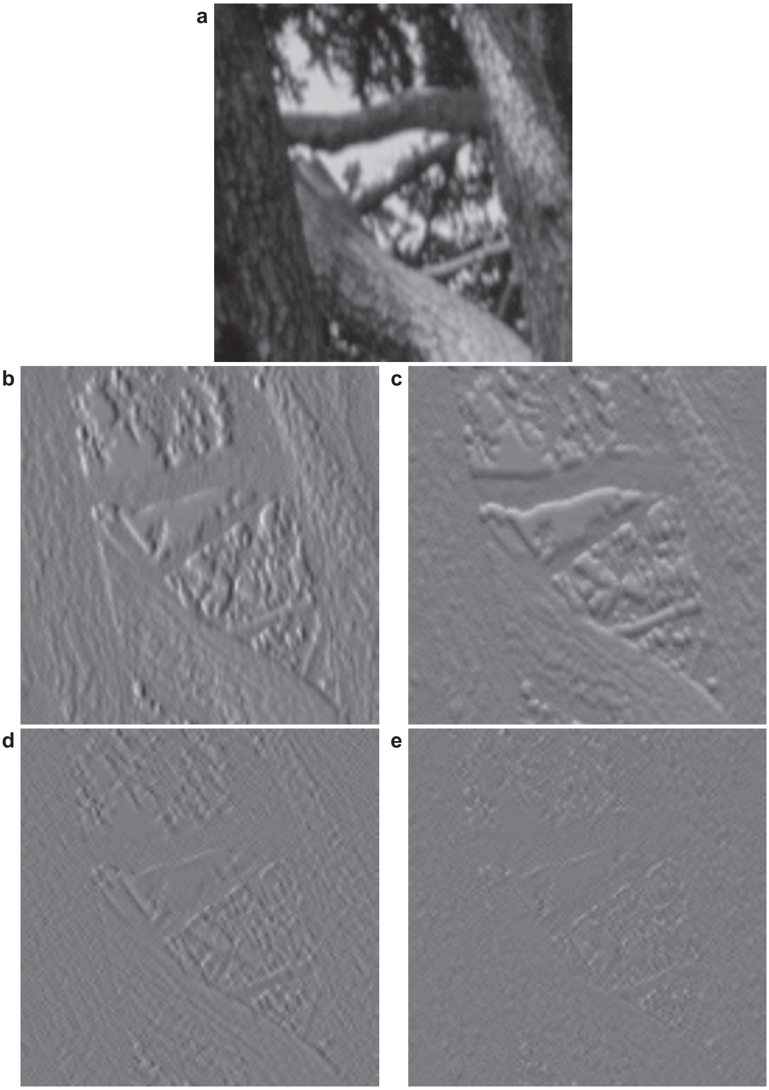

Fig. 2.31. Original image (**a**) and prediction error images: **b-c** 1D prediction row-wise, P_1=1, ρ_1=0.95 (b) 1D column-wise, P_2=1, ρ_2=0.95 (*c*) **d** 2D separable, fixed coefficients P_1=P_2=1, ρ_1=ρ_2=0.95, **e** 2D non-separable with local adaptation, quarter-plane P_1=P_2=2

Assume that 2D prediction is applied to a separable 2D AR(1) model, where the same prediction filter $H(z_1,z_2)$ is used as in the recursive loop of the model generator. Hence, the prediction error filter $A(z_1,z_2) = 1 - H(z_1,z_2)$ will exactly reproduce the Gaussian white noise fed into the generator of the AR process. For the 2D separable AR(1) model, the optimum predictor filter is constructed from the two (horizontal and vertical) 1D filters as follows:

$$\begin{aligned}&H(z_1)=\rho_1 z_1^{-1} \Rightarrow A(z_1)=1-\rho_1 z_1^{-1},\\&H(z_2)=\rho_2 z_2^{-1} \Rightarrow A(z_2)=1-\rho_2 z_2^{-1},\\&A(z_1,z_2)=A(z_1)A(z_2)=1-\rho_1 z_1^{-1}-\rho_2 z_2^{-1}+\rho_1\rho_2 z_1^{-1}z_2^{-1}\\&\Rightarrow H(z_1,z_2)=1-A(z_1,z_2)=\rho_1 z_1^{-1}+\rho_2 z_2^{-1}-\rho_1\rho_2 z_1^{-1}z_2^{-1}.\end{aligned} \tag{2.226}$$

Fig. 2.31 shows an original image (*a*), prediction error images obtained by horizontal (*b*) and vertical (*c*) 1D prediction, and by separable 2D prediction (*d*). While the horizontal prediction is not capable to predict vertical edges, the vertical filter fails at horizontal edges, but the 2D prediction performs reasonably well in both cases. Specifically in areas showing regular textured structures (e.g. grass, water, hairs, etc.), the usage of higher-order 2D predictors can be advantageous, if adapted properly to the signal (Fig. 2.31e). In the given example, the adaptation block size was 16x16 samples, quarter-plane prediction filters of size 3x3 were optimized by solving the Wiener-Hopf equation system (2.211).

Motion compensated prediction. When temporal prediction from previous picture(s) of a video signal shall be performed, an autoregressive model cannot reasonably capture the temporal changes occurring by object or camera motion, as it is not efficiently considering the sparseness of a moving video signal's spectrum from (2.44). In *motion compensated prediction*, predictor adaptation is rather performed by *motion estimation.* Samples in picture n_3 shall be predicted, and the best-matching position in a prediction reference picture (e.g. the previous picture n_3-1) is found to be displaced by k_1 samples horizontally and k_2 samples vertically. Then, the prediction equation is

$$\begin{aligned}&e(n_1,n_2,n_3)=s(n_1,n_2,n_3)-\hat{s}(n_1,n_2,n_3)\\&\qquad\text{with } \hat{s}(n_1,n_2,n_3)=s(n_1+k_1,n_2+k_2,n_3-1).\end{aligned} \tag{2.227}$$

This motion compensated predictor filter can be characterized by the 3D *z* transfer function[47]

$$H(z_1,z_2,z_3)=z_1^{\,k}z_2^{\,l}z_3^{-1}, \tag{2.228}$$

which describes a multi-dimensional shift (or delay); motion-compensated prediction therefore is a specific type of linear prediction. This simple type of filter uses a copy of samples from one previous picture and shifts them by an integer number of sample positions to generate the estimate. If the brightness of the signal changes, it could be more appropriate to multiply the amplitude by an additional factor, or shift it by an offset; in a more generalized approach, values from different ref-

[47] Basically, a random motion shift could also be included in an AR synthesis filter to obtain a reasonable video model.

erence pictures can be superimposed for prediction: Each of them may be weighted individually by a weighting factor $a(p_3)$, an offset c can optionally be added. If prediction shall further support sub-sample displacements, a spatial interpolation filter with impulse response $h_{\text{int}}(\mathbf{n})$ has to be included, with coefficients $a(p_1,p_2)$ in the convolution equation pending on the sub-sample phase $d_{1|2}$. The estimate is then computed by using up to P_3 reference pictures[48]

$$\begin{aligned}\hat{s}(n_1,n_2,n_3) &= c(n_1,n_2,n_3) + \sum_{p_3=1}^{P_3} a(p_3)\cdot \\ &\cdot \sum_{p_1=-Q_1/2}^{Q_1/2-1} \sum_{p_2=-Q_2/2}^{Q_2/2-1} a_{\text{int}}^{[d_1(p_3),d_2(p_3)]}(p_1,p_2)\, s\left[n_1+k_1(p_3)-p_1, n_2+k_2(p_3)-p_2, n_3+k_3(p_3)\right].\end{aligned} \tag{2.229}$$

With an offset $c(\mathbf{n})=0$, the z transfer function of the entire predictor filter can be described by

$$H(z_1,z_2,z_3) = \sum_{p_3=1}^{P_3} a_3(p_3) A_{\text{int}}^{[d_1(p_3),d_2(p_3)]}(z_1,z_2) z_1^{k_1(p_3)} z_2^{k_2(p_3)} z_3^{k_3(p_3)} \ . \tag{2.230}$$

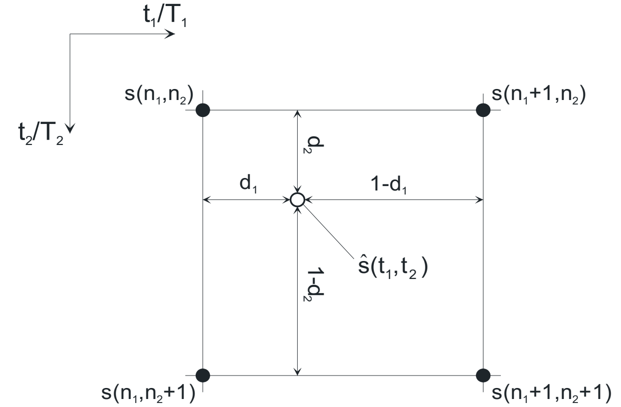

Fig. 2.32. Bilinear interpolation

The simplest approach of 2D interpolation is *bilinear interpolation*, which is separable, $h(t_1,t_2)=\Lambda(t_1)\Lambda(t_2)$ with $\Lambda(t)=\text{rect}(t)*\text{rect}(t)$. The principle is illustrated in Fig. 2.32. The value to be estimated at position (t_1,t_2) is computed from samples of

[48] 2D FIR interpolators with even impulse response lengths Q_1 and Q_2 are assumed for the horizontal and vertical sub-sample interpolations. As sample and sub-sample shifts can be different for each reference picture used in the prediction, the interpolation filter and sample motion shifts k_i are defined depending on the reference index p_3. Practically, displacements vary locally, such that the predictor filter is also a shift-variant system and the a, c and k_i parameters may also depend on n_1,n_2. In video coding, the reference pictures do not necessarily need to be ordered by their temporally sequence (see Sec. 7.2.4). This is expressed by the index n_3+$k_3(p_3)$ which defines an arbitrary list mapping.

four neighboring positions, which are weighted depending on the horizontal and vertical sub-sample phases d_1 and d_2 ($0 \le d_i < 1$):

$$\begin{aligned}\hat{s}(t_1,t_2) &= s(n_1,n_2)(1-d_1)(1-d_2) + s(n_1+1,n_2)d_1(1-d_2) \\ &\quad + s(n_1,n_2+1)(1-d_1)d_2 + s(n_1+1,n_2+1)d_1 d_2 .\end{aligned} \tag{2.231}$$

However, bilinear interpolation has a relatively strong lowpass effect, and also does not provide good alias suppression[49]. Therefore, in practice, higher-order interpolation filters are used for better performance in motion compensated prediction with sub-sample accuracy (cf. Sec. 7.2.5).

In video coding, *block matching* is often used for motion estimation. Let Λ express a partition, for which a common horizontal/vertical displacement shift vector $\mathbf{k}+\mathbf{d} = [k_1+d_1,\ k_2+d_2]$ shall be determined for a given reference picture with distance k_3 from the current picture n_3. Π describes a set of candidate displacement shifts. *Cost functions* based on minimization of difference criteria of norm Q are often used for this purpose[50],

$$\left[\mathbf{k}+\mathbf{d}\right]_{\text{opt}}(k_3) = \underset{[k_1,k_2]\in\Pi}{\arg\min} \left| \frac{1}{|\Lambda|} \sum_{(n_1,n_2)\in\Lambda} \left| s(n_1,n_2,n_3) - \hat{s}(n_1+k_1+d_1, n_2+k_2+d_2, n_3+k_3) \right|^Q \right|^{\frac{1}{Q}} . \tag{2.232}$$

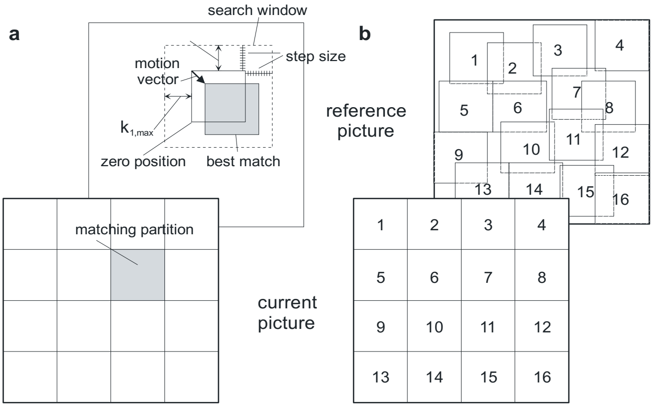

Fig. 2.33. Block matching motion estimation. **a** Definition of matching partition, search range and step size in the current picture **b** Possible overlaps of best-matching blocks in the reference picture

[49] Due to the triangular impulse response of the underlying 1D filter, its spectral transfer function is sinc^2, which has its first zero at $|f|=1/2$, the first two side lobe in the first alias band $1/2 \le |f| \le 3/2$, and further side lobes in higher-frequency alias bands.

[50] Q=1 for *sum of absolute difference* (SAD), Q=2 for *sum of squared difference* (SSD). Sub-sample accurate shift parameters $l_i = k_i + d_i$ are used here, which means that for the case $d_i \neq 0$ it is necessary to compute $\hat{s}$ by interpolation filtering, cf. (2.229)/(2.230).

Fig. 2.33 illustrates the method. In Fig. 2.33a, all samples of a given partition in the current picture are subject to the same horizontal/vertical shift, and the sample pattern of the partition is compared against patterns from the set of candidate positions in the given reference picture. As a result, the displacement vector corresponding to the best pattern match is selected. Fig. 2.33b indicates an inconsistency of rigid block partitioning, as this may cause unreasonable overlaps or gaps between adjacent blocks in the reference picture, at positions where the motion is discontinuous (e.g. at object boundaries). Generally, the partitions may either be of equal size (as shown in the figure) or of variable size. As an example, with full search, scanning over all possible shift positions within a 2D search window, the total number of positions to be compared is growing linearly with the area of the search window and with the density (reciprocal squared value of step size Δ, which is the distance between adjacent shifted candidate positions).

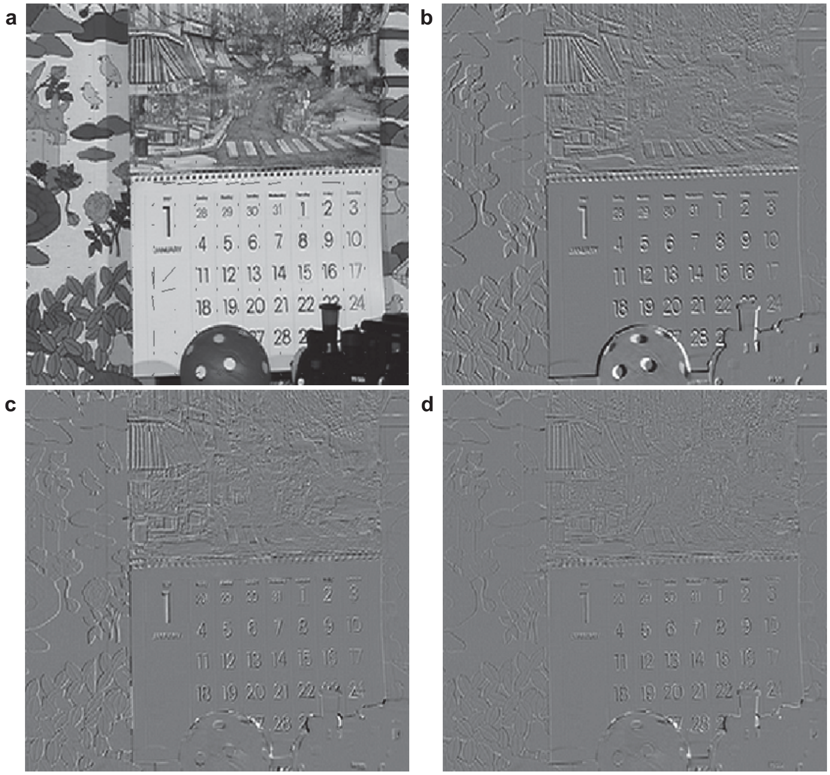

Fig. 2.34. a Picture (with MVs) from a video sequence, and prediction error pictures: **b** without motion compensation **c** with motion compensation, full-sample shift accuracy **d** with motion compensation, half-sample shift accuracy (both motion compensated examples with constant block grid of size 16x16, half-sample shift by bilinear interpolation)

Fig. 2.34 shows results of a picture predicted without and with motion compensation, the latter case also with bilinear interpolation filtering for half-sample accuracy (all with block grid of size 16x16). Fig. 2.35 shows corresponding results with quarter-sample accuracy, with bilinear and higher-quality (8-tap filter) interpolation, the latter as well with reduced sizes of the block grid, 8x8 and 4x4 samples (contrast enhanced 1.5x in residual pictures for better visibility).

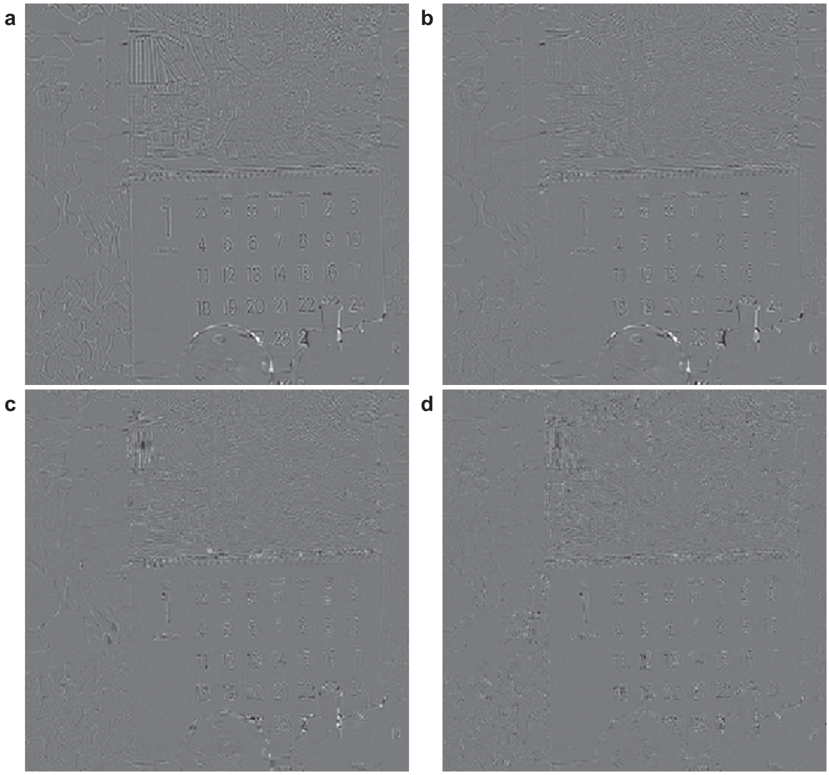

Fig. 2.35. Examples with quarter-sample accuracy in motion compensation: **a** bilinear interpolation, 16x16 block grid, and further examples with 8-tap interpolation filter: **b** 16x16 block grid **c** 8x8 block grid **d** 4x4 block grid

The true motion shift between two pictures will typically be by sub-sample units. It is however not useful to test all possible sub-sample positions over the entire search window range, as it can be expected that the cost criterion in (2.232) varies smoothly over **k**. Therefore, strategies for fast search are used which start by larger Δ values and refine the estimated motion vector into sample or sub-sample accuracy only by the last few steps.

Fast search algorithms do not test all possible candidates and therefore may no longer guarantee that the global optimum over the **k**+**d** parameter space is reached. However, with same complexity, fast algorithms can often achieve even better results compared to exhaustive (full search) approaches, since they avoid testing unreasonable candidates and can instead investigate an extended parameter space. In one or the other way, fast motion estimation algorithms inherently exploit

- the smoothness of cost functions in dependency of **k**+**d**, due to the fact that the sample patterns at adjacent candidate positions in the reference picture are almost identical when the step size Δ is small; this allows optimizing the result by iterative steps;
- the smoothness of displacement vector fields, both over the spatial coordinate (e.g. consistent motion of larger objects) and the temporal coordinate (along the motion trajectory), which allows predicting initial candidates from previous estimates;
- The joint scaling property of picture and motion vector field, where for spatially downsampled signals the number of sample-wise operations, as well as the size of the search range can be reduced[51].

Furthermore, it is possible to apply early termination of the search, when a sufficiently good displacement (in terms of cost function) has already been found. All previously mentioned approaches for search speedup are complementary and can be combined.

Multi-step search. Two principles of fast motion estimation algorithms are shown in Fig. 2.36a/b. Both are based on testing only a subset of search positions out of the entire set of parameters, where the favorable direction of changing **k** is traced for optimization of the cost criterion. In Fig. 2.36a/b, all positions tested in the particular steps are drawn as black dots, the steps are referenced by numbers, and the optimum as found in the respective step is marked by a circle. In both examples, the motion vector is finally found as $k_1=-5$, $k_2=2$. These two algorithms are typical representatives for a variety of similar approaches, one of the first was suggested in [KOGA ET AL 1981].

In the method of Fig. 2.36a, originally denoted as *three-step search* in [MUSMANN ET AL. 1985], only a small set of 9 candidate positions is evaluated in each step. Simultaneously, the search step size is decreased gradually ($\Delta=3$, 2, 1 sample width for the three iterations in the example shown). The center of the search range in iteration step r is selected from the best-matching position of the previous iteration $r-1$, such that cost criteria need to be computed only for 8 new positions in iterations 2 and 3. In the example shown, a total of $9+8+8=25$ candidate posi-

[51] The scaling property also imposes an interesting relationship between the picture size and the complexity of motion estimation. If the picture size is doubled horizontally and vertically, the density of samples is increased likewise. However, the size of the search range has also to be doubled horizontally and vertically, as now the related displacement maps into a motion vector **k** of double length. Considering exhaustive search, this leads to a complexity increase by a factor of 16 when doubling the picture size. It could be argued that when downsampling the step size Δ should be decreased (e.g. going from half sample to quarter sample precision of motion compensation), which would diminish this benefit. This is however not the case, if proper lowpass filtering is applied in the context of downsampling, which loses spatial detail to some extent.

tions are compared in the three iterations; the maximum range in the given example is $k_{1,\max}=k_{2,\max}=\pm 6$ samples. A full search with same range would require testing of $13^2=169$ positions. The factor of reducing computational complexity increases with larger search range (more iteration steps). Typically, a complexity dependency on $k_{\max}$ or $\log(k_{\max})$ (instead of $k_{\max}^2$ for full search) can be achieved.

In the search method shown in Fig. 2.36b, 5 different positions in $\mathcal{N}_1$ arrangement are compared in the first iteration step. In the example, a step size $\Delta=2$ is used. After finding the best match among these, only three more positions adjacent to the previous optimum need to be compared in any remaining step. This process is continued until the best-match position remains unchanged, which indicates that a local minimum over the cost function has been approached. Then, in a final step, all 8 shift positions around this optimum, or additional sub-sample positions, are checked as candidates. In the example shown, only 5+2·3+8=19 candidate positions have to be tested in total.

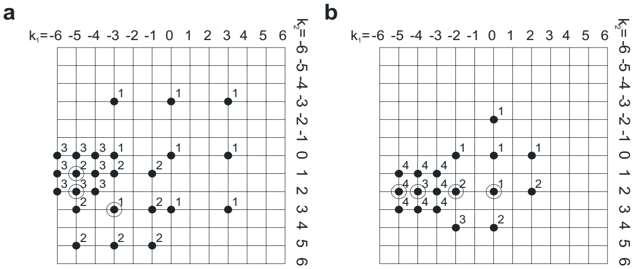
a
b
k1=-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
k2=-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
1
2
3
4

Fig. 2.36. Multi-step block-matching estimation methods – examples of **a** 'three-step search' **b** 'logarithmic search'

These concepts will approach the *globally-optimum* result (as in full search), if the cost function is truly convex and therefore improving over the motion parameter space towards the optimum position. If local extremes of the cost function exist, it is possible to get stuck in such a position. This can be the case when several similar structures (e.g. periodicities) are present.

Displacement vector predictors. As continuity of the motion vector field can be assumed, reasonable predictions for correct displacement vectors are often available from previous estimates in the spatial or temporal neighborhood. Fig. 2.37 shows possible candidate vectors from adjacent partitions (here: blocks), which can be used to predict the displacement for the current partition. The temporal predictor can be selected from the 'collocated' position in the reference picture, or as a vector which points from a location in the reference picture into the current block partition[52].

[52] It may be necessary to scale the candidate vectors based on the time distance between the current picture and the reference picture in comparison to the time distance that is in

Different approaches are possible to determine the final estimate for the displacement vector of the current partition:

- The mean or median from a set of previously-estimated vectors (candidates) is computed as starting point, and the new value is optimized by testing candidates within an additional search range around this initial hypothesis[53];
- The search range is determined from the range between minimum and maximum displacement values found in the set of predictor candidates;
- Different search ranges are tested around the values of several predictor candidate's displacement vectors (if they are not identical).

In this context, it is also common practice to terminate the estimation without further refinement when one of the initial candidates already provides a good estimate, which further speeds up the search on average [DE HAAN ET AL. 1993].

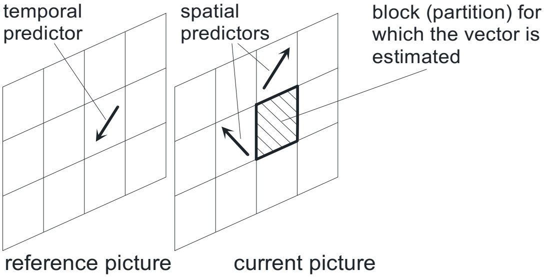

Fig. 2.37. Examples of temporally and spatially adjacent displacement candidates in predictive motion estimation for block matching

Multi-resolution motion estimation. Multi-resolution estimation determines candidates from downsampled pictures. For the same content, the displacement shift is down-scaled as well with the picture size/resolution; therefore, the search range can be down-scaled as well, whereas the motion is still captured [NAM ET AL. 1995]. In subsequent steps, the picture resolution is increased, but the estimation starts from the result of the previous step, which can be expected to be already close to the true motion, such that the search range can again be small. Furthermore, also the size of the partitions for which a common motion displacement is estimated can be decreased, in which case the spatial resolution of available displacement vectors (i.e. the density of the vector field) also increases with the picture resolution. In terms of the hierarchical representation of the pictures, such an approach can be interpreted as a Gaussian pyramid (Sec. 2.8); with decreased

effect in the motion compensation where they are used. The number of candidates can also be variable, depending on the local variation of motion.

[53] The median value can be computed separately for the horizontal and vertical displacements, or jointly for both, depending on the vector length; this may however generate a combined displacement which effectively does not exist.

partition sizes, also a *pyramid of motion vector fields* with both increasing resolution and precision is generated.

In full-search motion estimation, the size of the search range per direction $k_{i,\max}$ has the most important effect on complexity. With this regard, multi-resolution estimation has a similar benefit as multi-step search[54], but due to the additional subsampling of the pictures reduces the complexity even further and is eventually more stable due to the lowpass characteristics of downsampled pictures.

When up-scaled displacement vectors from a lower resolution level are used as starting points for estimating in several adjacent partitions in the next higher resolution level, and the differences between them are small due to using small search ranges around the candidate, hierarchical estimation implicitly can generate spatially more continuous motion vector fields. The relationships between matching areas and estimated motion vectors at two different levels are illustrated in Fig. 2.38.

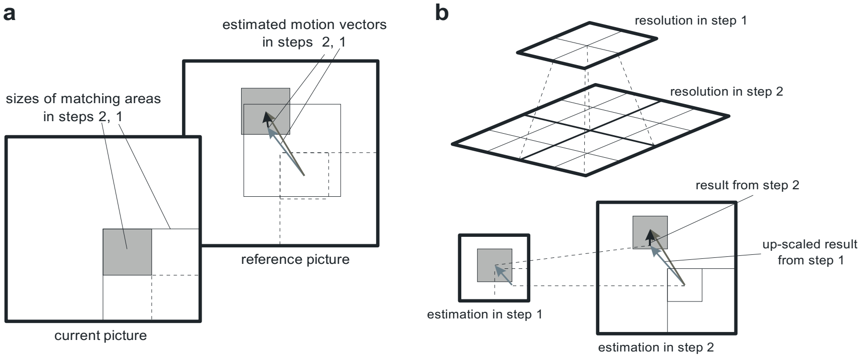

Fig. 2.38. Hierarchical motion estimation over two steps **a** Interpretation at full resolution **b** Principle of reduced resolution in the first step, and up-scaling of the resulting displacement vector

Variable block-size estimation. Displacement vector fields representing motion shifts are continuous (with only small amount of changes) within areas of background or larger moving objects, but discontinuous at object boundaries. Both properties can best be reflected when variable-size partitions are used for regions that are assigned to a common displacement vector. Typical strategies in optimizing displacement estimation for variable block sizes start from larger partitions and performs splitting into smaller partitions in cases where this has advantages w.r.t. the cost function. However, it should be observed that generally for areas with less detail estimated vectors could be ambiguous (this is denoted as *aperture problem*, see [JÄHNE 2005]). Therefore, splitting should be justified by a significant benefit in the cost function, and eventually large deviations in the displacements

[54] Both multi-step and multi-resolution methods are sometimes entitled as *hierarchical motion estimation*.

of adjacent split partitions may be inhibited by additional constraints (e.g. smoothness criteria, see subsequent section). Beyond the splitting strategy, another approach would be to start with smaller partitions and merge them, if the same displacement can be applied without significant disadvantage to the cost function. Again here, a smoothness constraint can be used as part of the cost function.

Constrained estimation. *Additional constraints* are often introduced in block matching, where the cost function of a given estimate is modified by a *penalty term* $\lambda\mathscr{P}$ e.g.

- establishing interrelationships between motion vectors estimated in adjacent blocks by a smoothness constraint that takes into account motion vector differences,
- regularizing estimates in low-detail regions where no unique motion vector can be determined, by aligning them with the displacement of adjacent higher-detail regions,
- taking into account the rate that would be required to encode the displacement vector [GIROD 1994].

An example for a constrained optimization criterion in analogy with (2.232) is

$$\mathbf{k}_{\text{opt}} = \arg\min_{\mathbf{k}\in\mathbf{\Pi}} \left[\frac{1}{|\mathbf{\Lambda}|} \sum_{\mathbf{n}\in\mathbf{\Lambda}} \left| s(\mathbf{n}) - s(\mathbf{n}+\mathbf{k}) \right|^P + \lambda\mathscr{P}(\mathbf{k}) \right] \tag{2.233}$$

State of the art fast motion estimation algorithms used in video coding are often using combinations of the aforementioned approaches.

2.7 Linear block transforms

2.7.1 Orthogonal basis functions

The Discrete Fourier Transform (DFT) (2.89) is computed by multiplying M samples from a signal by an orthogonal set of complex basis functions. In general, two finite discrete 1D (real or complex) functions $t_i(n)$ and $t_j(n)$, each of length M, are *orthogonal*, if their linear combination gives zero,

$$\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_j^* = \sum_{n=0}^{M-1} t_i(n)t_j^*(n) = \varphi_{t_i t_j}(0) = 0 \quad \text{with} \quad \mathbf{t}_k = \left[t_k(0) \quad \cdots \quad t_k(M-1) \right]^{\mathrm{T}} . \tag{2.234}$$

If the functions $t_k(n)$ are interpreted as impulse responses of linear filters, and if the operations $c_k(n)=s(n)*t_k(n)$ are performed over all n, it can be shown that the cross correlation between any two resulting outputs $\varphi_{c_ic_j}(0)=0$ for $i\neq j$. Therefore, the usage of orthogonal basis functions can provide a *de-correlated representation* of a signal[55].

For an *orthogonal set* of basis functions, each member of the set is orthogonal with any other. The computation of the transform coefficient $c_k = \mathbf{s}^{\mathrm{T}}\mathbf{t}_k$ is a mapping from the signal domain into a transformed domain (which could be interpreted as a sampled frequency domain, provided that the basis functions have an appropriately ordered frequency transfer behaviour). If reconstruction of the signal samples is possible, the discrete set of transform coefficients establishes an equivalent representation. For processing of longer-duration signals, *local* or *short-time transforms* are often applied, in simplest case processing non-overlapping block segments (vectors) **s** of length M from the signal in a *block transform*. In the following, this problem will first be discussed for the case of one-dimensional transforms, from which two- and multidimensional transforms can straightforwardly be constructed by separable processing over the different coordinate axes. A segment from the signal $s(n)$, consisting of M subsequent samples and starting at position mM, shall be mapped into a set of U transform coefficients

$$c_k(m) = \sum_{n=0}^{M-1} s(mM+n)t_k(n) \quad ; \quad 0 \le k < U \,. \tag{2.235}$$

It shall be possible to reconstruct this segment of the signal by a complementary set of synthesis functions (inverse transform), such that

$$s(mM+n) = \sum_{k=0}^{U-1} c_k(m) r_k(n) \; ; \quad 0 \le n < M \,. \tag{2.236}$$

Substituting (2.236) into (2.235) gives

$$\sum_{n=0}^{M-1} t_k(n) \sum_{l=0}^{U-1} \frac{c_l(m)}{c_k(m)} r_l(n) = 1 \quad ; \quad 0 \le (k,l) < U \,. \tag{2.237}$$

This condition can only hold for all k, if the factor c_l / c_k is zero for $l \neq k$, such that

$$\sum_{n=0}^{M-1} t_k(n) r_l(n) = \begin{cases} 1 & \text{for} \quad k = l \\ 0 & \text{for} \quad k \neq l \end{cases} \quad ; \quad 0 \le (k,l) < U \,. \tag{2.238}$$

In the special case of an orthogonal set $\{\mathbf{t}_k\}$[56], this is fulfilled by choosing the matching analysis and synthesis bases $\mathbf{t}_k$ and $\mathbf{r}_k$ as complex conjugates, by which

[55] If however the sequences $c_k(n)$ are subsampled, as often applied in the context of transform coding to avoid an overcomplete representation, correlation may occur partially due to aliasing.

[56] Observe that the fulfillment of (2.238) does not necessarily require that the analysis basis functions $\mathbf{t}_k$ or synthesis basis functions $\mathbf{r}_l$ by themselves establish orthogonal sets; it is only necessary that function k from one set is orthogonal with function $l \neq k$ from the other set. This joint property of two sets is called *bi-orthogonality*; the choice of an orthogonal set $\{\mathbf{t}_k\}$ and $\{\mathbf{r}_k\}=\{\mathbf{t}_k^*\}$ is a special case thereof.

$\mathbf{t}_k$ is implicitly orthogonal with any other synthesis basis $\mathbf{r}_l$. The further constraint $\mathbf{t}_k{}^{\mathrm{T}}\mathbf{r}_k=1$ can be avoided by the generalization

$$\mathbf{r}_k = \frac{\mathbf{t}_k^*}{A_k} \quad \text{s. t.} \quad \mathbf{t}_k{}^{\mathrm{T}}\mathbf{t}_k^* = A_k \ \text{(real, positive)}\,. \tag{2.239}$$

By combining (2.238) and (2.239), a more general orthogonality condition for the set $\{\mathbf{t}_k\}$ is

$$\mathbf{t}_k{}^{\mathrm{T}}\mathbf{t}_l^* = \sum_{n=0}^{M-1} t_k(n)t_l^*(n) = \begin{cases} A_k & \text{for} \quad k=l \\ 0 & \text{for} \quad k \neq l \end{cases} \quad ; \quad 0 \leq (k,l) < U\,. \tag{2.240}$$

For the example of the DFT (2.89) and IDFT (2.90), $t_k(n)=\mathrm{e}^{-\mathrm{j}2\pi nk/M}$, $r_k(n)=\mathrm{e}^{\mathrm{j}2\pi nk/M}/M$ and $A_k = M$ fulfills these conditions. A general transform from M signal values into U coefficients can also be formulated by the matrix notation

$$\underbrace{\begin{bmatrix} c_0(m) \\ c_1(m) \\ \vdots \\ \vdots \\ c_{U-1}(m) \end{bmatrix}}_{\mathbf{c}(m)} = \underbrace{\begin{bmatrix} t_0(0) & t_0(1) & \cdots & \cdots & t_0(M-1) \\ t_1(0) & t_1(1) & \cdots & \cdots & t_1(M-1) \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ t_{U-1}(0) & t_{U-1}(1) & \cdots & \cdots & t_{U-1}(M-1) \end{bmatrix}}_{\mathbf{T}} \cdot \underbrace{\begin{bmatrix} s(mN+N_0) \\ s(mN+N_0+1) \\ \vdots \\ \vdots \\ s((m+1)N+N_0-1) \end{bmatrix}}_{\mathbf{s}(m)}, \tag{2.241}$$

where the signal vector $\mathbf{s}$ consists of M samples, the transform matrix $\mathbf{T}$ has size MxU with rows establishing *basis vectors* and the result $\mathbf{c}$ includes U transform coefficients c_k. As a minimum condition for reconstruction, the transformed representation $\{\mathbf{c}(m)\}$ over all blocks shall have the same number of samples as the signal $s(n)$. This can be achieved when the starting positions of subsequent vectors $\mathbf{s}(m)$ are $N=U$ samples apart, where N_0 is an optional constant offset. In simplest case of a non-overlapping block transform, $N=M=U$ and $M_0=0$. Then, since the rows of $\mathbf{T}=[\mathbf{t}_0\ \mathbf{t}_1\ \ldots\ \mathbf{t}_{U\text{-}1}]^{\mathrm{T}}$ are the *basis functions* from an orthogonal set, they are linearly independent, $\mathbf{T}$ is a square matrix, will have full rank and is invertible. Following (2.241), the values of $s(n)$ in $\mathbf{s}(m)$ can uniquely be reconstructed from the coefficients c_k in $\mathbf{c}(m)$,

$$\mathbf{s}(m) = \mathbf{T}^{-1}\mathbf{c}(m)\,. \tag{2.242}$$

The transform is *orthonormal*, if $A_k = 1$ in (2.240). Analysis and synthesis vectors in (2.239) are identical for a real-valued orthonormal transform basis. More generally, for a complex orthonormal transform from (A.26)[57]

$$\mathbf{T}^{-1} = \left[\mathbf{T}^*\right]^{\mathrm{T}} = \mathbf{T}^{\mathrm{H}}\,. \tag{2.243}$$

The synthesis functions $\mathbf{r}_l$ from (2.238) are the columns of $\mathbf{T}^{\mathrm{H}}$; in combination

[57] $\mathbf{T}^{\mathrm{H}}$ is the Hermitian matrix (conjugate transpose) of $\mathbf{T}$.

with (2.243), this gives $\mathbf{T}\mathbf{T}^{-1} = \mathbf{I}$. In orthonormal linear transforms, the quadratic norm (energy) of signal vectors can directly be computed without any normalization from the coefficient vectors,

$$\mathbf{s}^{\mathrm{T}}\mathbf{s} = \mathbf{s}^{\mathrm{T}} \underbrace{\mathbf{T}^{\mathrm{H}}\mathbf{T}}_{\mathbf{I}} \mathbf{s} = \left[\mathbf{c}^*\right]^{\mathrm{T}} \mathbf{c} \quad \text{or} \quad \left\|\mathbf{s}\right\|^2 = \left\|\mathbf{c}\right\|^2 . \tag{2.244}$$

Otherwise, if basis vector norms are different from unity, an equivalence is still found when the values in $||\mathbf{c}||^2$ are scaled by the individual A_k values.

The series of transform vectors $\mathbf{c}(m)$ is computed from signal vectors $\mathbf{s}(m)$ with starting positions $n_0(m)=mN+N_0$. With hop size $N>U$, reconstruction cannot be guaranteed, with $N<U$, the result of the transform would be over-complete. For the purpose of coding, $N=U$ is most appropriate. In case of block-overlapping transforms (Sec. 2.7.4), the vectors $\mathbf{s}$ are longer than vectors $\mathbf{c}$, i.e. $M>U$. In this case, though a single $\mathbf{c}(m)$ can uniquely be computed from the corresponding $\mathbf{s}(m)$, reconstruction may require involvement of other vectors $\mathbf{c}(m)$ that also depend on samples in $\mathbf{s}(m)$, which can be achieved by a weighted *overlap-and-add* procedure as a secondary step. In the remaining part of the current section, $N=U=M$ is assumed.

A *separable two-dimensional transform* can be expressed as concatenation of two matrix multiplications using a horizontal transform $\mathbf{T}_{\mathrm{h}}$ and a vertical transform $\mathbf{T}_{\mathrm{v}}$,

$$\underbrace{\begin{bmatrix} c_{0,0} & c_{1,0} & \cdots & c_{U_1-1,1} \\ c_{0,1} & c_{1,1} & \cdots & c_{U_1-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{0,U_2-1} & c_{1,U_2-1} & \cdots & c_{U_1-1,U_2-1} \end{bmatrix}}_{\mathbf{C}} = \underbrace{\begin{bmatrix} t_0(0) & t_0(1) & \cdots & t_0(M_2-1) \\ t_1(0) & t_1(1) & \cdots & t_1(M_2-1) \\ \vdots & \vdots & \ddots & \vdots \\ t_{U_2-1}(0) & t_{U_2-1}(1) & \cdots & t_{U_2-1}(M_2-1) \end{bmatrix}}_{\mathbf{T}_{\mathrm{v}}} \cdot \cdots \tag{2.245}$$

$$\cdots \cdot \underbrace{\begin{bmatrix} s(0,0) & s(1,0) & \cdots & s(M_1-1,0) \\ s(0,1) & s(1,1) & \cdots & s(M_1-1,1) \\ \vdots & \vdots & \ddots & \vdots \\ s(0,M_2-1) & s(1,M_2-1) & \cdots & s(M_1-1,M_2-1) \end{bmatrix}}_{\mathbf{S}} \cdot \underbrace{\begin{bmatrix} t_0(0) & t_1(0) & \cdots & t_{U_1-1}(0) \\ t_0(1) & t_1(1) & \cdots & t_{U_1-1}(1) \\ \vdots & \vdots & \ddots & \vdots \\ t_0(M_1-1) & t_1(M_1-1) & \cdots & t_{U_1-1}(M_1-1) \end{bmatrix}}_{\mathbf{T}_{\mathrm{h}}^{\mathrm{T}}} .$$

In a first step, all columns (length M_2) of the image matrix $\mathbf{S}$ are transformed separately giving $\mathbf{C}_{\mathrm{v}}=\mathbf{T}_{\mathrm{v}}\mathbf{S}$, the result of the vertical transform applied separately over all columns. The subsequent horizontal transform of $\mathbf{C}_{\mathrm{v}}$ is performed by using the transposed transform matrix $\mathbf{T}_{\mathrm{h}}^{\mathrm{T}}$ rather than transposing the matrix $\mathbf{C}_{\mathrm{v}}$[58]. The

[58] Alternatively, the second step could be $\mathbf{C}=[\mathbf{T}_{\mathrm{h}}\mathbf{C}_{\mathrm{v}}^{\mathrm{T}}]^{\mathrm{T}}$, however the above formulation gives the output in correct (not transposed) order right away. It is of course also possible to perform the horizontal transform first, where mathematically the final result is identical.

matrix equations for the separable 2D transform and the related inverse transform are as follows[59]:

$$\mathbf{C} = \underbrace{\left[\mathbf{T}_\mathrm{v}\mathbf{S}\right]}_{\mathbf{C}_\mathrm{v}}\mathbf{T}_\mathrm{h}^{\mathrm{T}} \quad \Rightarrow \quad \mathbf{S} = \left[\mathbf{T}_\mathrm{v}^{-1}\mathbf{C}\right]\left[\mathbf{T}_\mathrm{h}^{-1}\right]^{\mathrm{T}} . \tag{2.246}$$

The basis functions relating to U_1U_2 coefficients of the separable 2D transform are

$$t_{k_1,k_2}(n_1,n_2) = t_{k_1}(n_1)t_{k_2}(n_2) \quad ; \quad 0 \le k_i < U_i \quad ; \quad \mathbf{T}_{k_1,k_2} = \mathbf{t}_{k_1}\mathbf{t}_{k_2}^{\mathrm{T}} \quad . \tag{2.247}$$

The 2D basis matrices $\mathbf{T}_{k_1,k_2}$ are also denoted as *basis images*. A two- or multi-dimensional expression can generally be written as $t_\mathbf{k}(\mathbf{n})$ and $\mathbf{T}_\mathbf{k}$, where the related (scalar) transform coefficient can be expressed as the Frobenius product (A.10) of matrices or tensors,

$$c_\mathbf{k} = \mathbf{T}_\mathbf{k} : \mathbf{S} . \tag{2.248}$$

2.7.2 Types of orthogonal transforms

In this section, basis functions of some important transforms are introduced mostly by their one-dimensional versions. They extend to the case of two-dimensional separable transforms according to (2.246).

$$\mathbf{T}^{\mathrm{Haar}}(8) = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} = \begin{bmatrix} \mathbf{t}_0^{\mathrm{T}} \\ \mathbf{t}_1^{\mathrm{T}} \\ \vdots \\ \vdots \\ \\ \\ \vdots \\ \mathbf{t}_7^{\mathrm{T}} \end{bmatrix} . \tag{2.249}$$

Rectangular basis functions. The analysis block lengths M of the following rectangular basis function transforms are typically dyadic (M=2^l, $l \in \mathbb{N}$). Typically (except for scaling necessary to achieve orthonormality), these transforms can be computed without multiplications. The Haar transform uses basis functions of non-constant length[60], where identical elementary functions (performing difference analysis over neighbored samples) are re-used at different positions of the

[59] In case of orthonormality, $\mathbf{S}$=$\mathbf{T}_\mathrm{v}^{\mathrm{H}}\mathbf{C}\mathbf{T}_\mathrm{h}^{*}$.

[60] A more systematic construction of the Haar and Walsh transforms can be found in the formulation of Problem 2.13. The Haar transform can also be defined as a discrete wavelet transform (see Sec. 2.8.4) from the filter basis (2.312).

block. As an example, the transform matrix of an orthonormal Haar transform with $U=M=8$ is shown in (2.249). For the orthonormal transform, the scaling factors for the different basis types vary. Basis functions for the case $M=8$ are shown in Fig. 2.39a. The *Walsh basis* consists of $U=M$ basis functions, the set for the case $M=8$ is shown in Fig. 2.39b. The corresponding transform matrix is

$$\mathbf{T}^{\text{Walsh}}(8)=\frac{1}{2\sqrt{2}}\begin{bmatrix}1&1&1&1&1&1&1&1\\1&1&1&1&-1&-1&-1&-1\\1&1&-1&-1&-1&-1&1&1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&-1&-1&1&-1&1&1&-1\\1&-1&1&-1&-1&1&-1&1\\1&-1&1&-1&1&-1&1&-1\end{bmatrix}=\begin{bmatrix}\mathbf{t}_0^{\text{T}}\\\mathbf{t}_1^{\text{T}}\\\vdots\\\vdots\\ \\ \\\vdots\\\mathbf{t}_7^{\text{T}}\end{bmatrix}. \tag{2.250}$$

The Walsh transform can be interpreted to be analyzing 'frequency' (based on toggling rectangles rather than oscillating sinusoids), as the number of zero crossings is steadily increasing with index *k*.

The *Hadamard transform* has the same set of functions as the Walsh basis, however the ordering (index numbering of basis functions) is different, not allowing interpretation by 'increasing frequency'. The rule for recursive construction implicitly guarantees orthogonality. Starting from a 1x1 identity matrix with $M'=1$, the recursion doubles the block length by each step,

$$\begin{aligned}&\mathbf{T}^{\text{Had}}(1)=[1],\\&\mathbf{T}^{\text{Had}}(2M')=\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{T}^{\text{Had}}(M') & \mathbf{T}^{\text{Had}}(M')\\\mathbf{T}^{\text{Had}}(M') & -\mathbf{T}^{\text{Had}}(M')\end{bmatrix}\quad\text{for}\quad M'=1,2,4,\ldots,M/2.\end{aligned} \tag{2.251}$$

The Hadamard transform matrix for the case *M*=8 then is (see also Fig. 2.39c):

$$\mathbf{T}^{\text{Had}}(8)=\frac{1}{2\sqrt{2}}\begin{bmatrix}1&1&1&1&1&1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&1&1&1&-1&-1&-1&-1\\1&-1&1&-1&-1&1&-1&1\\1&1&-1&-1&-1&-1&1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix}=\begin{bmatrix}\mathbf{t}_0^{\text{T}}\\\mathbf{t}_1^{\text{T}}\\\vdots\\\vdots\\ \\ \\\vdots\\\mathbf{t}_7^{\text{T}}\end{bmatrix}. \tag{2.252}$$

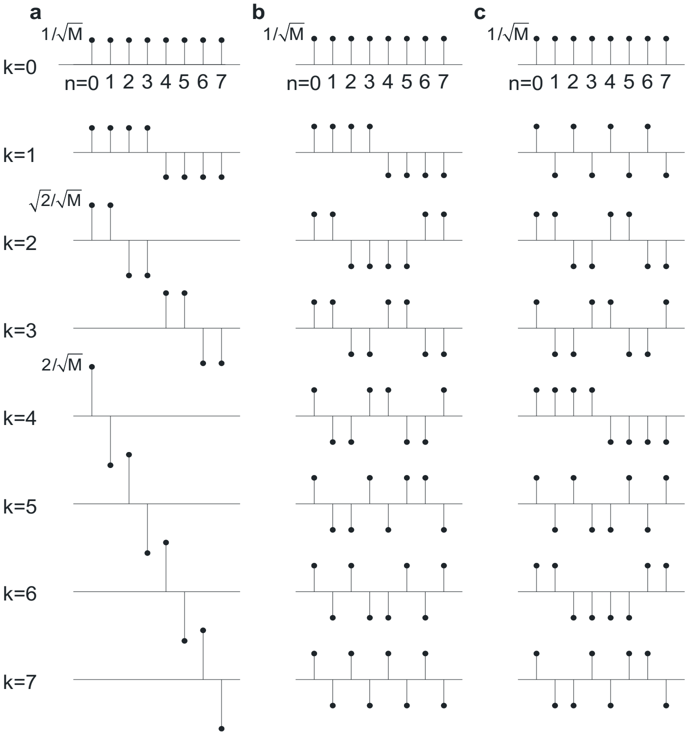

Fig. 2.39. Rectangular basis function systems **a** Haar **b** Walsh **c** Hadamard

$$\mathbf{T}^{\mathrm{DFT}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \mathrm{e}^{-\mathrm{j}\frac{2\pi}{M}} & \mathrm{e}^{-\mathrm{j}\frac{4\pi}{M}} & \mathrm{e}^{-\mathrm{j}\frac{6\pi}{M}} & \cdots & & \mathrm{e}^{-\mathrm{j}\frac{2\pi(M-1)}{M}} \\ 1 & \mathrm{e}^{-\mathrm{j}\frac{4\pi}{M}} & \mathrm{e}^{-\mathrm{j}\frac{8\pi}{M}} & & \ddots & & \mathrm{e}^{-\mathrm{j}\frac{4\pi(M-1)}{M}} \\ 1 & \mathrm{e}^{-\mathrm{j}\frac{6\pi}{M}} & & & & & \\ 1 & \vdots & \ddots & & \ddots & & \vdots \\ \vdots & & & & & & \\ 1 & \mathrm{e}^{-\mathrm{j}\frac{2\pi(U-1)}{M}} & \mathrm{e}^{-\mathrm{j}\frac{4\pi(U-1)}{M}} & & \cdots & & \mathrm{e}^{-\mathrm{j}\frac{2\pi(U-1)(M-1)}{M}} \end{bmatrix}. \qquad (2.253)$$

Sinusoidal basis functions. The Discrete Fourier Transform (DFT) is defined as

$$c_k = \sum_{n=0}^{M-1} s(n) W_M{}^{-mk} \qquad s(n) = \sum_{k=0}^{M-1} c_k W_M{}^{mk} \qquad \text{with} \;\; W_M = \mathrm{e}^{\mathrm{j}\frac{2\pi}{M}}. \qquad (2.254)$$

The complex exponential basis can be interpreted as harmonic sinusoids of specific frequency and phase. The transform matrix of the DFT is shown in (2.253). Note that this version of the DFT is not orthonormal. From (2.240), $A_k = M$ for a 1D transform and $A_{k_1k_2} = M_1M_2$ for a separable 2D transform[61]. Further, the DFT implicitly interprets a series of samples as periodic, even if they only represent a segment from a longer signal. Therefore, occasional amplitude differences between the left and right boundaries of the analysis segment are interpreted as discontinuity (see Fig. 2.40a), and spectral energy appears over broad frequency ranges. Further, when the signal is locally periodic, but the wave length (or a multiple thereof) does not match with M, energy is also spread over a certain range of the spectrum. Therefore, the DFT possesses undesirable properties with the threat of producing artifacts both in picture/video and audio compression. One approach of avoiding this is usage of window functions with roll-off towards the ends, typically used with overlap of adjacent blocks.

The amplitude discontinuity can also be avoided, if a (mirror) symmetric extension of the signal is constructed, which leads to an even symmetry and a real-valued DFT spectrum. In a first approach, even symmetry can be implemented around the points n=0 and n=M−1, with a period length of 2M−2 (from M independent samples), as shown in Fig. 2.40b. Computing a DFT

$$c_k = \sum_{n=-M+1}^{M-2} s(n)\,\mathrm{e}^{-\mathrm{j}2\pi\frac{kn}{2M-2}} \quad \text{with} \quad s(n) = s(-n) \quad \text{for} \quad n<0\,, \tag{2.255}$$

gives real-valued coefficients as

$$\begin{aligned} c_k &= s(0) + (-1)^k s(M-1) + \sum_{n=1}^{M-2} s(n)\left[\mathrm{e}^{\mathrm{j}2\pi\frac{kn}{2M-2}} + \mathrm{e}^{-\mathrm{j}2\pi\frac{kn}{2M-2}}\right] \\ &= s(0) + (-1)^k s(M-1) + 2\sum_{n=1}^{M-2} s(n)\cos\left[\frac{\pi}{M-1}nk\right]. \end{aligned} \tag{2.256}$$

The values c_k are periodic over k with 2M−2, and are again even symmetric around k=0 and k=M−1. Therefore, the inverse transform is identical, except for a normalization factor 1/(2M−2)[62]. This real-valued transform is entitled as DCT type-I (for a detailed description of the different types and their implementation, see [Britanak et al. 2010] [Chen, Smith, Fralick 1977]).

For applications in data compression, the DCT-I is however not best suitable due to the property that all basis functions have a maximum amplitude value at n=0, which can lead to high errors at the left block edge when transform coeffi-

[61] For an orthonormal version of the 1D DFT, normalization by a factor $1/\sqrt{M}$ has to be applied both in the analysis and synthesis (IDFT). Likewise, for a 2D transform, the normalization must use a factor $1/\sqrt{M_1M_2}$.

[62] Alternatively, forward and inverse transform are identical in case of orthonormality, where a normalization $1/\sqrt{1/(2M-2)}$ is applied in (2.256).

cients are discarded or heavily quantized. Second, the basis functions do not have symmetry properties themselves due to the misalignment between the length of M samples and the cosine which is periodic over $M-1$ or a multiple thereof. Third, the lowest frequency is representing approximately a full cosine period over the length of the basis function, such that signals with a slower increasing amplitude are not efficiently presented (which are often observed particularly in image signals).

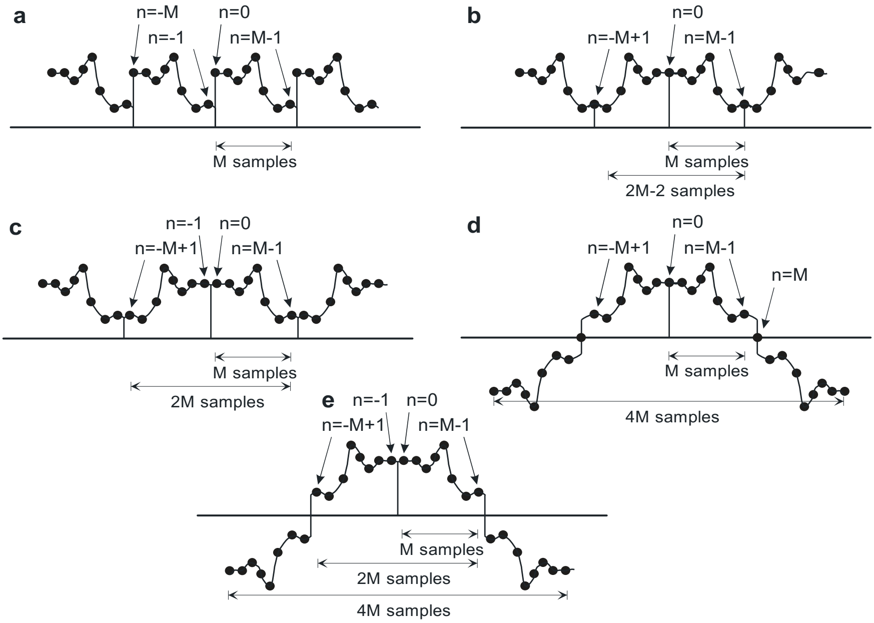

Fig. 2.40. Extension of the signal at boundaries of a finite analysis segment of length M **a** periodic (DFT case) **b** DCT type I **c** DCT type II **d** DCT type III **e** DCT type IV

To overcome these problems, the points of even symmetry could also be put at $n=-½$ and $n=M-½$, such that these two points are duplicated as well, and the even Fourier transform has to be computed over a length of $2M$, where now exactly half of the samples is redundant (see Fig. 2.40c). This can be realized by a modification of the DFT basis function, through a shift by half a sample in the complex exponent. Then,

$$c_k=\sum_{n=-M}^{M-1}s(n)\mathrm{e}^{-\mathrm{j}2\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)} \quad \text{with} \quad s(n)=s(-n-1) \quad \text{for} \quad n<0\,, \tag{2.257}$$

which can be re-written as DCT type II:

$$c_k=\sum_{n=0}^{M-1}s(n)\left[\mathrm{e}^{-\mathrm{j}2\pi\frac{k}{2M}\left(-n-1+\frac{1}{2}\right)}+\mathrm{e}^{-\mathrm{j}2\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)}\right]=2\sum_{n=0}^{M-1}s(n)\cos\left[k\left(n+\frac{1}{2}\right)\frac{\pi}{M}\right]. \tag{2.258}$$

In the coefficient domain, the following observations are made: Coefficients c_M and c_{-M} are zero; otherwise, the following symmetries apply:

$$c_{-k} = c_k \quad \text{for } 0 < k < M; \quad c_{|k|} = -c_{|2M-k|} \quad \text{for } M < |k| < 2M\,. \tag{2.259}$$

This means that the series of coefficients has an odd symmetry around k=±M, and is periodic in k over a length of 4M, where however still only M independent coefficients exist, all other are redundant. This can be explained by the fact that by introducing the shift by half a sample virtually the sampling rate is doubled and the block length of the DCT would also be 4M, where however each second sample is implicitly zero. Therefore, alias spectra appear within the spectrum period (cf. Sec. 2.8.1). In terms of the inverse DCT, the following computation is necessary (formally, the sum should run over 4M samples with normalization by 1/(4M), but the coefficients for |k| ≥ M with corresponding complex exponentials would give exactly the same contribution and can therefore be omitted):

$$\begin{aligned} s(n) &= \frac{1}{2M}\left[\sum_{k=0}^{M-1} c_k\,\mathrm{e}^{\mathrm{j}2\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)} + \sum_{k=-M+1}^{1} c_k\,\mathrm{e}^{\mathrm{j}2\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)}\right] \text{ with } c_{-k} = c_k \\ &= \frac{1}{M}\left(\frac{c_0}{2} + \sum_{k=1}^{M-1} c_k \cos\left[k\left(n+\frac{1}{2}\right)\frac{\pi}{M}\right]\right). \end{aligned} \tag{2.260}$$

The transform of (2.260), when applied to a signal, is also entitled as DCT type III. Its symmetry properties (even around n=0 and odd around n=M) are shown in Fig. 2.40d. The corresponding inverse transform is the DCT-II.

Due to the shift by half a sample and usage of symmetric basis functions, the combination of DCT-II and DCT-III can also be used for linear interpolation (upsampling) of signals. In case of upsampling by a factor of 2, this can be achieved replacing the sign-inverted coefficients at positions k=M+1…2M−1 by zero values[63], and extending the inverse transform to the full length of 4M samples, such that 2M samples are generated by the inverse transform. Similarly, filling more zeroes and extending the block length of the inverse transform allows higher upsampling ratios.

Both DCT-II (2.258) and DCT-III (2.260) have orthogonal basis vectors, but do not fulfill (2.240), as the norm of $\mathbf{t}_0$ is different. By the following modification, orthonormality is achieved, and the basis vectors of the DCT-II can be written as

$$\begin{aligned} t_k^{\text{DCT-II}}(n) &= \sqrt{\frac{2}{M}}C_0 \cos\left[k\left(n+\frac{1}{2}\right)\frac{\pi}{M}\right] \quad \text{for } 0 \le \{n,k\} < M \\ &\text{with} \quad C_0 = \frac{1}{\sqrt{2}} \text{ for } k=0 \quad ; \quad C_0 = 1 \text{ for } k \ne 0. \end{aligned} \tag{2.261}$$

Another concept is the DCT type IV, which somewhat combines the properties of DCT-II and DCT-III by using even symmetry around n = −½ and odd symmetry

[63] This is equivalent to suppression of alias spectra in interpolation filtering, cf. Sec. 2.8.1.

around $n = \pm M-½$ (see Fig. 2.40e). The period over n is $4M$ (again with only M independent samples), where however now the contributions of the values beyond the odd symmetry point contribute differently. Since $s(n)$ is even, values for $n<0$ would contribute as complex conjugate such that the DFT over length $4M$ gives

$$c_k = 2\mathrm{Re}\left\{\sum_{n=0}^{2M-1} s(n)\,\mathrm{e}^{-\mathrm{j}2\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)}\right\} \text{with } s(n) = -s(2M-n-1) \;\text{ for }\; n \geq M\,, \tag{2.262}$$

which can be re-written as

$$\begin{aligned} c_k &= 2\mathrm{Re}\left\{\sum_{n=0}^{M-1} s(n)\left[\mathrm{e}^{-\mathrm{j}2\pi\frac{k}{4M}\left(n+\frac{1}{2}\right)} - \mathrm{e}^{-\mathrm{j}2\pi\frac{k}{4M}\left(2M-n-1+\frac{1}{2}\right)}\right]\right\} \\ &= 2\mathrm{Re}\left\{\sum_{n=0}^{M-1} s(n)\left[\mathrm{e}^{-\mathrm{j}\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)} - \mathrm{e}^{-\mathrm{j}\pi k}\,\mathrm{e}^{\mathrm{j}\pi\frac{k}{2M}\left(n+\frac{1}{2}\right)}\right]\right\}. \end{aligned} \tag{2.263}$$

The result is zero for even values of k. Replacing $k \rightarrow 2k+1$ for considering only the non-zero coefficients at odd positions finally gives

$$c_k = 4\sum_{n=0}^{M-1} s(n)\cos\left[\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\frac{\pi}{M}\right]. \tag{2.264}$$

In the DCT-IV, n and k have identical influences on the basis function. Furthermore, it can be concluded that virtual zero samples should exist both over n and k; including the frequency zero which would be at $k = -½$ in (2.264). Due to the symmetry of the coefficient series which is even around $k = -½$ and odd around $k = \pm M-½$, the inverse transform is identical to (2.264), except the need for amplitude scaling by $1/(8M)$[64]. The lowest discrete frequency comes with the basis function $\mathbf{t}_0$ which is a cosine with one full period over $4M$ (or a quarter of a period over M).

Due to these properties, the DCT-IV is most suitable for compression of zero-mean signals, such as audio. It is also used as a basis for block-overlapping transforms (see Sec. 2.7.4), where typically the length of the basis function is extended to an equivalent of $2M$ but multiplied by a window function that decays towards the tails and completely chops off the negative mirrored parts. By this, any undesirable effects of the odd symmetry at $n = \pm(M-½)$ are avoided, which is denoted as 'time domain alias cancellation' [PRINCEN, BRADLEY 1986].

As further variants of symmetric extensions with extended DFT basis functions, it is also possible to apply an *odd symmetry* around $n=0$ or $n=-½$. In this case, the real part of the DFT will be zero, but the values of the imaginary part can be used as if they were real valued coefficients. It should however be considered that odd symmetry requires the symmetry point itself to be zero, which has to

[64] $8M$ is the actual period over k when the zero coefficients would be included. Alternatively, both forward and inverse transforms can be scaled by $\sqrt{2/M}$.

apply whenever n=0 or n=±M are used as symmetry points. When there are M non-zero values, nevertheless the extended DFT length needs to include the zero values. Since odd signal components relate to the (imaginary) sine component of the DFT's complex exponential, this class of transforms is categorized as *Discrete Sine Transform* (DST). Similar to the discussion above, there are mainly four types:

- DST type I: Odd symmetries both at n=0 and n=±M, effective DFT period (including two zero samples) 2M+2; it has symmetric basis functions.
- DST type II: Odd symmetries both at n=–½ and n=M–½, no zero samples, effective DFT period 2M;
- DST type III: Odd symmetry at n=0, even symmetry at n=±M, effective DFT period (including two zero samples at n=0 and n=–2M) 4M;
- DST type IV: Odd symmetries at n=–½, even symmetry at n=M–½, no zero samples, effective DFT period 4M;

Again, DST-II and DST-III are inverses of each other, whereas DST-I and DST-IV are identical with their inverse transforms. Relevant in terms of data compression are DST-I and DST-IV, which match the properties of boundary prediction problems, where a set of M subsequent samples is predicted from the same boundary sample, such that the prediction error increases with larger distance from the boundary (see section 5.2.4) – in case of an AR(1) process, the DST in an optimum way removes correlation from the prediction error, specifically

- The DST-I is best suitable in case of two-sided prediction (block of M samples predicted from boundary samples at both ends) [Jain 1976]. The first basis function from the following set of length M is a half sine wave with maximum in the center of the block; this matches the properties of the prediction error which can be expected to be maximum around position M/2 (farthest from the boundary samples used for prediction)

$$t_k^{\text{DST-I}}(n)=\sqrt{\frac{2}{M+1}}\cdot\sin\left[\frac{\pi}{M+1}(k+1)(n+1)\right] \quad ; \quad 0\le\{n,k\}<M \; ; \qquad (2.265)$$

- The DST-IV is best suitable for one-sided prediction (block of M samples predicted from boundary samples at the beginning); The first basis function from the following set of length M is a quarter sine wave with maximum by the end of the block,

$$t_k^{\text{DST-IV}}(n)=\sqrt{\frac{2}{M}}\cdot\sin\left[\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\right] \quad ; \quad 0\le\{n,k\}<M \, . \qquad (2.266)$$

The two-dimensional DCT is widely used in image and video compression (e.g. in standards like MPEG, JPEG, H.261/2/3). Mathematically precise formulations of the 2D DCT-II and its inverse 2D DCT-III over a rectangular signal block of size M_1M_2 are (with factors C_0 defined separately for the two dimensions, following (2.261))

$$c_{k_1k_2} = \frac{2}{\sqrt{M_1M_2}} C_0^{(k_1)} C_0^{(k_2)} \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{M_2-1} s(n_1,n_2) \cos\left[k_1\left(n_1+\frac{1}{2}\right)\frac{\pi}{M_1}\right] \cos\left[k_2\left(n_2+\frac{1}{2}\right)\frac{\pi}{M_2}\right]; \tag{2.267}$$

$$s(n_1,n_2) = \frac{2}{\sqrt{M_1M_2}} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} C_0^{(k_1)} C_0^{(k_2)} c_{k_1k_2} \cos\left[k_1\left(n_1+\frac{1}{2}\right)\frac{\pi}{M_1}\right] \cos\left[k_2\left(n_2+\frac{1}{2}\right)\frac{\pi}{M_2}\right]. \tag{2.268}$$

Fig. 2.41 shows *basis images* of different separable 2D transforms, as defined by (2.247).

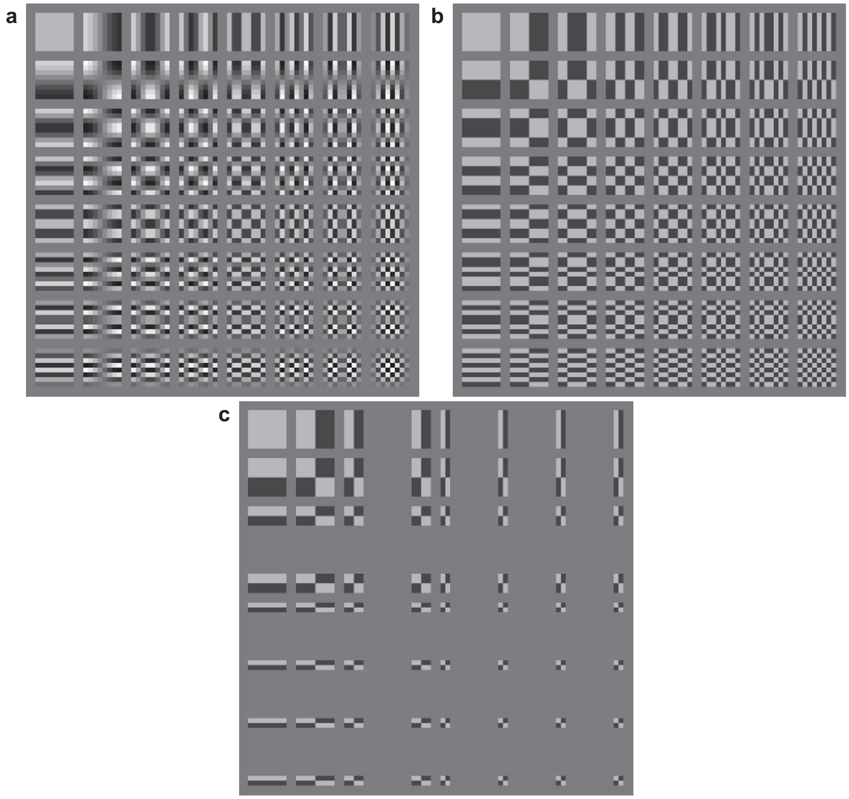

Fig. 2.41. 2D basis images of transforms. **a** DCT **b** Walsh **c** Haar

Integer transforms. Rectangular (binary) basis transforms allow to be computed without multiplications. Rectangular-basis transforms also guarantee perfect reconstruction of signals from a transform coefficient representation of finite bit precision[65]. Furthermore, the basis functions of rectangular transforms allow

[65] However, it should be observed that the necessary bit precision for lossless representation increases by $\log_2 M$ bits compared to the original signal representation, even in the case of Haar and Walsh transforms

efficient representation of discontinuities such as sharp edges, whereas they give only a poor representation of smoothly increasing amplitudes or smooth periodic structures. Sinusoidal transforms are better capable to approximate the latter types of signals by minimum error, but the trigonometric functions cannot be implemented up to full mathematical precision; rounding errors may occur, which could even affect the property of orthogonality. As a compromise, (non-binary) integer transform bases can be designed, which capture smoothly varying signal behavior better than rectangular basis functions, and retain orthogonality properties even with low word length integer arithmetic. One example for this class is the following transform of length $M = 4$, which is used in the Advanced Video Coding standard (cf. Sec. 7.8) [MALVAR ET AL. 2003],

$$\mathbf{T}^{\text{int}}(4) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}. \tag{2.269}$$

Different normalization factors $1/2, 1/\sqrt{10}, 1/2, 1/\sqrt{10}$ have to be applied for orthonormality of the respective basis vectors of (2.269), but the necessary scaling can be combined with quantization (if used for compression). A truly orthonormal/identical-norm integer transform (the square root scaling factor could be transferred to the inverse transform), approximating a length-4 DCT is defined by the following matrix

$$\mathbf{T}^{\text{int}}(4) = \frac{1}{\sqrt{676}} \begin{bmatrix} 13 & 13 & 13 & 13 \\ 17 & 7 & -7 & -17 \\ 13 & -13 & -13 & 13 \\ 7 & -17 & 17 & -7 \end{bmatrix}. \tag{2.270}$$

$$\mathbf{T}^{\text{int}}(8) = \frac{1}{\sqrt{1352}} \begin{bmatrix} 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\ 19 & 15 & 9 & 3 & -3 & -9 & -15 & -19 \\ 17 & 7 & -7 & -17 & -17 & -7 & 7 & 17 \\ 9 & 3 & -19 & -15 & 15 & 19 & -3 & -9 \\ 13 & -13 & -13 & 13 & 13 & -13 & -13 & 13 \\ 15 & -19 & -3 & 9 & -9 & 3 & 19 & -15 \\ 7 & -17 & 17 & -7 & -7 & 17 & -17 & 7 \\ 3 & -9 & 15 & -19 & 19 & -15 & 9 & -3 \end{bmatrix}. \tag{2.271}$$

A truly orthonormal integer transform of block length $M = 8$, with a similar construction as (2.270) proposed in [WIEN 2003], is defined by the transform matrix in (2.271). Herein, some of the basis functions can be constructed as mirror-symmetric extensions of the transform (2.270) with $M = 4$, or the latter can be generated by using first halves of each second basis function in (2.271), as indicated by the boxes.

Similar constructions of ‘nested’ sets of transform basis functions for different transform block sizes have not been found yet beyond M=8, if the property of strict orthogonality shall be retained, However, if the orthogonality constraint is slightly released, such that $0\neq\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_j << \mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i$ for $i\neq j$, similar constructions are possible. The HEVC standard contains the definition of integer approximations of the DCT for block sizes $M=4,8,16,32$, where the different-length basis functions are nested exactly the same way as above; also the norms of the basis vectors are almost equal and consistent over the various transform block sizes, such that no specific quantization needs to be employed. As an example, the transform matrix for $M=16$ is shown here, and the nested DCT functions for $M=8$ are highlighted in the leftmost eight columns [BUDAGAVI ET AL. 2013]:

$$\mathbf{T}^{\text{int}}(16)=\begin{bmatrix}
64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64\\
90 & 87 & 80 & 70 & 57 & 43 & 25 & 9 & -9 & -25 & -43 & -57 & -70 & -80 & -87 & -90\\
89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 & -89 & -75 & -50 & -18 & 18 & 50 & 75 & 89\\
87 & 57 & 9 & -43 & -80 & -90 & -70 & -25 & 25 & 70 & 90 & 80 & 43 & -9 & -57 & -87\\
83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 & 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83\\
80 & 9 & -70 & -87 & -25 & 57 & 90 & 43 & -43 & -90 & -57 & 25 & 87 & 70 & -9 & -80\\
75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 & -75 & 18 & 89 & 50 & -50 & -89 & -18 & 75\\
70 & -43 & -87 & 9 & 90 & 25 & -80 & -57 & 57 & 80 & -25 & -90 & -9 & 87 & 43 & -70\\
64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64\\
57 & -80 & -25 & 90 & -9 & -87 & 43 & 70 & -70 & -43 & 87 & 9 & -90 & 25 & 80 & -57\\
50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 & -50 & 89 & -18 & -75 & 75 & 18 & -89 & 50\\
43 & -90 & 57 & 25 & -87 & 70 & 9 & -80 & 80 & -9 & -70 & 87 & -25 & -57 & 90 & -43\\
36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 & 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36\\
25 & -70 & 90 & -80 & 43 & 9 & -57 & 87 & -87 & 57 & -9 & -43 & 80 & -90 & 70 & -25\\
18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 & -18 & 50 & -75 & 89 & -89 & 75 & -50 & 18\\
9 & -25 & 43 & -57 & 70 & -80 & 87 & -90 & 90 & -87 & 80 & -70 & 57 & -43 & 25 & -9
\end{bmatrix} \tag{2.272}$$

From the matrix (2.272) and the construction of the nested shorter transforms, it can be recognized that the basis functions of the DCT alternate as even and odd symmetric functions with amplitude-identical coefficients at both sides (as can be expected from the type-II construction, see above). Only the even functions are useful for the construction of the shorter DCT transforms, as they again give an alternating even/odd set of functions. Likewise, the even functions of the next larger transform can be constructed by symmetric extension of all functions.

Another example for an almost orthogonal transform based on integer coefficients of basis functions is the approximation of the type-IV DST (2.266) which is used as alternative transform for this purpose with block length M=4 in the HEVC standard,

$$\mathbf{T}^{\text{int}}(4)=\begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}. \tag{2.273}$$

An optimum transform – KLT. Linear transforms for multimedia signal compression should approximate a signal as accurate as possible, using lowest possible number of transform coefficients. Often, the energy of the reconstruction error is used as criterion for optimality. Assuming that only T out of U coefficients shall be retained to represent the signal, the reconstruction error of a 1D transform is

$$e(n)=s(n)-\frac{1}{A}\sum_{k=0}^{T-1}c_k t_k^*(n). \tag{2.274}$$

This gives an energy of the error over the complete block

$$\begin{aligned}\|\mathbf{e}\|^2 &= \sum_{n=0}^{M-1} e^2(n) \\ &= \sum_{n=0}^{M-1}\left[s^2(n)-2s(n)\frac{1}{A}\sum_{k=0}^{T-1}c_k t_k^*(m)+\left|\frac{1}{A}\sum_{k=0}^{T-1}c_k t_k^*(n)\right|^2\right].\end{aligned} \tag{2.275}$$

Substituting in (2.275) under the condition that $s(n)$ is real-valued

$$s(n)=\frac{1}{A}\sum_{l=0}^{U-1}c_l t_l^*(m)=\left[\frac{1}{A}\sum_{l=0}^{U-1}c_l^* t_l(n)\right]^*=\frac{1}{A}\sum_{l=0}^{U-1}c_l^* t_l(n) \tag{2.276}$$

gives

$$\|\mathbf{e}\|^2=\sum_{n=0}^{M-1}\left[s^2(n)-\frac{2}{A^2}\sum_{l=0}^{U-1}\sum_{k=0}^{T-1}c_k t_k^*(n)c_l^* t_l(n)+\left|\frac{1}{A}\sum_{k=0}^{T-1}c_k^* t_k(n)\right|^2\right], \tag{2.277}$$

by which, using the orthogonality condition (2.240),

$$\|\mathbf{e}\|^2=\sum_{n=0}^{M-1}s^2(n)-\frac{1}{A}\sum_{k=0}^{T-1}|c_k|^2. \tag{2.278}$$

(2.278) shows that for $T=U=M$, $\|\mathbf{e}\|^2=0$, such that again the energy of the signal over M values can be determined from the coefficients of the linear orthogonal transform. This corresponds to the condition (2.244), however it is more general now as it shows that by omitting transform coefficients, the error energy in the reconstruction is exactly the energy of these coefficients, scaled by the normalization factor A.

Consequently, the optimum transform can be derived under the condition that the energy of the error shall be *minimized* if reconstruction from a finite number of coefficients is performed. This is equivalent to *maximizing* the energy over the first T coefficients. Now assume that samples from a stationary random process shall be transformed, expected squared values are taken of the first T coefficients, such that the following condition can be formulated (for simplicity, the orthonormal case is considered):

$$\begin{aligned}\sum_{k=0}^{T-1}\mathcal{E}\left\{\left|c_k\right|^2\right\}&=\sum_{k=0}^{T-1}\mathcal{E}\left\{c_k c_k^*\right\}=\sum_{k=0}^{T-1}\mathcal{E}\left\{\sum_{n=0}^{M-1}s(n)t_k(n)\sum_{m=0}^{M-1}s(m)t_k^*(m)\right\}\\&=\sum_{k=0}^{T-1}\mathcal{E}\left\{\sum_{m=0}^{M-1}\left(\sum_{n=0}^{M-1}s(n)s(m)t_k(n)\right)t_k^*(m)\right\}\\&=\sum_{k=0}^{T-1}\left[\sum_{m=0}^{M-1}\left(\sum_{n=0}^{M-1}\mu_{ss}(\left|m-n\right|)t_k(n)\right)t_k^*(m)\right].\end{aligned}\qquad(2.279)$$

The maximum of this expression is approached if the innermost parenthesis fulfills the condition

$$\sum_{n=0}^{M-1}\mu_{ss}(\left|m-n\right|)t_k(n)=\lambda_k t_k(m),\qquad(2.280)$$

which is achieved by establishing the transform basis as the set of eigenvectors of the discrete autocovariance sequence. Substituting (2.280) into (2.279) further gives

$$\sum_{k=0}^{T-1}\mathcal{E}\left\{\left|c_k\right|^2\right\}=\sum_{k=0}^{T-1}\lambda_k\underbrace{\left[\sum_{m=0}^{M-1}t_k(m)t_k^*(m)\right]}_{=1},\qquad(2.281)$$

which shows that the related eigenvalue λ_k represents the power of coefficient c_k. The basis functions of this *Karhunen-Loève transform* (KLT) need specific adaptation by the autocovariance statistics of a given signal, and would be globally optimum for Gaussian (e.g. autoregressive) stationary processes. Formulating (2.280) for the entire set of basis functions, these can be computed as eigenvectors $\boldsymbol{\phi}_k$ of the autocovariance matrix (2.157)[66], which must then be constructed as an MxM matrix containing the covariance function samples $\mu_{ss}(0)$ ... $\mu_{ss}(M-1)$:

$$\begin{aligned}&\mathbf{C}_{ss}\boldsymbol{\phi}_k=\lambda_k\boldsymbol{\phi}_k\\&\quad\text{with}\quad\boldsymbol{\phi}_k=\left[\phi_k(0)\quad\phi_k(1)\quad\cdots\quad\phi_k(M-1)\right]^{\mathrm{T}}\quad;\quad 0\le k<U.\end{aligned}\qquad(2.282)$$

Alternatively, the conjugates[67] of the eigenvectors $\boldsymbol{\phi}_k^*$ can be defined to establish the rows of transform matrix $\mathbf{T}^{\mathrm{KLT}}$ [68]

[66] We have assumed zero-mean property here and in the subsequent equations; however, the power component related to the mean would typically concentrate in coefficient c_0, such that the basic proof for the optimality of the KLT does not change.

[67] Definition by conjugates of eigenvectors makes the KLT consistent with the DFT definition and with correlation analysis; actually, the DFT is the optimum transform for perfect

$$\left[\boldsymbol{\phi}_k^{\,*}\right]^{\mathrm{T}} \mathbf{C}_{ss}{}^{\mathrm{T}} = \lambda_k \left[\boldsymbol{\phi}_k^{\,*}\right]^{\mathrm{T}} . \tag{2.283}$$

(2.283) can be written in matrix notation

$$\begin{bmatrix} \phi_0^*(0) & \phi_0^*(1) & \cdots & \phi_0^*(M-1) \\ \phi_1^*(0) & \ddots & \ddots & \\ \vdots & \ddots & & \vdots \\ & & & \\ \phi_{U-1}^*(0) & & \cdots & \phi_{U-1}^*(M-1) \end{bmatrix} \begin{bmatrix} \mu_{ss}(0) & \mu_{ss}(1 & \cdots & & \mu_{ss}(M-1) \\ \mu_{ss}(1) & \mu_{ss}(0) & \mu_{ss}(1) & & \\ \vdots & \mu_{ss}(1) & \ddots & \ddots & \vdots \\ & & \ddots & & \\ \mu_{ss}(M-1) & & \cdots & & \mu_{ss}(0) \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & 0 & & \\ 0 & 0 & \lambda_2 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & \cdots & 0 & \lambda_{U-1} \end{bmatrix} \begin{bmatrix} \phi_0^*(0) & \phi_0^*(1) & \cdots & \phi_0^*(M-1) \\ \phi_1^*(0) & \ddots & \ddots & \\ \vdots & \ddots & & \vdots \\ & & & \\ \phi_{U-1}^*(0) & & \cdots & \phi_{U-1}^*(M-1) \end{bmatrix}$$

$$\Leftrightarrow \mathbf{T}^{\mathrm{KLT}}\mathbf{C}_{ss} = \boldsymbol{\Lambda}\mathbf{T}^{\mathrm{KLT}} . \tag{2.284}$$

Multiplying both sides of (2.284) by the inverse transform matrix retains the diagonal matrix $\boldsymbol{\Lambda}$ on the right side, which is populated by the eigenvalues λ_k:

$$\mathbf{T}^{\mathrm{KLT}}\mathbf{C}_{ss}\left[\mathbf{T}^{\mathrm{KLT}}\right]^{\mathrm{H}} = \boldsymbol{\Lambda} = \mathrm{Diag}\left\{\lambda_k\right\} \quad \Rightarrow \sum_{k=0}^{U-1} \mathcal{E}\left\{\left|c_k\right|^2\right\} = \mathrm{tr}\left\{\boldsymbol{\Lambda}\right\} = \mathcal{E}\left\{\sum_{n=0}^{M-1} s^2(n)\right\} . \tag{2.285}$$

$\boldsymbol{\Lambda}$ in (2.285) can also be regarded as the ‘optimum transform’ of the autocovariance matrix, whereby a statistical representation of the discrete spectral samples c_k is generated. While the correlation inherent in the signal is indicated by the fact that $\mathbf{C}_{ss}$ is not a diagonal matrix, the diagonal shape of $\boldsymbol{\Lambda}$ indicates that no correlation is present anymore between the spectral samples.

2.7.3 Efficiency of transforms

An important criterion for judging the efficiency of a transform is concentration of as much signal energy as possible in as few transform coefficients as possible [Clarke 1985]. The related *energy packing efficiency* η_e is the normalized ratio of energy, contained within the first T out of U transform coefficients:

cyclic signals, such as signals composed from sinusoids, each with a period being an exact fraction of the analysis block length.

[68] This provides consistency with transform basis definitions used so far, which could be interpreted as a correlation test between the signal samples and the respective basis function. In case of a complex basis, it is necessary to test against the complex conjugate.

$$\eta_e(T) = \frac{\sum_{l=0}^{T-1} \mathcal{E}\left\{c_l^{\,2}\right\}}{\sum_{k=0}^{U-1} \mathcal{E}\left\{c_k^{\,2}\right\}} . \tag{2.286}$$

The KLT maximizes the energy packing efficiency, as it is optimized using this criterion, see (2.279). Another aspect is the *decorrelation efficiency* η_c, determined from the autocovariance matrix $\mathbf{C}_{ss}$ and its transform[69],

$$\mathbf{C}_{cc} = \mathcal{E}\left\{\mathbf{cc}^{\mathrm{H}}\right\} = \mathcal{E}\left\{(\mathbf{Ts})(\mathbf{Ts})^{\mathrm{H}}\right\} = \mathbf{T}\mathcal{E}\left\{\mathbf{ss}^{\mathrm{T}}\right\}\mathbf{T}^{\mathrm{H}} = \mathbf{TC}_{ss}\mathbf{T}^{\mathrm{H}} . \tag{2.287}$$

The decorrelation efficiency is then defined as[70]

$$\eta_c = 1 - \frac{\sum_{k=0}^{U-1} \sum_{\substack{l=0 \\ (k \neq l)}}^{U-1} \left|\mu_{cc}(k,l)\right|}{\sum_{k=0}^{M-1} \sum_{\substack{l=0 \\ (k \neq l)}}^{M-1} \left|\mu_{ss}(k,l)\right|} . \tag{2.288}$$

For the case of the KLT, $\mathbf{C}_{cc}$ is the eigenvalue matrix $\mathbf{\Lambda}$ in (2.285) where all entries with $k \neq l$ are zero, hence η_c has a maximum of 1. This means that the KLT achieves optimum decorrelation when optimized for a signal that posesses certain autocovariance statistics. For other transforms than the KLT, also linear statistical dependencies (non-zero correlation) between coefficients of the discrete transform representation may be present.

2.7.4 Transforms with block overlap

A linear transform can be interpreted as a convolution of the signal, using impulse responses which are the time reversed and complex conjugate basis functions. Unlike conventional convolution, the computation of transform coefficients needs only to be performed at each M^{th} position in the case of transforms without block overlap, which can also be interpreted as subsampling of the convolution output. In the spectral domain, the generation of the transform coefficient can therefore be interpreted as multiplication of the signal spectrum by the Fourier transfer functions of the respective basis vectors. The transform coefficients are carrying information related to *all frequencies* which are passing through their respective Fourier-domain transfer functions. Fig. 2.42 shows the Amplitude transfer func-

[69] When the number of basis functions equals the number of samples, the square matrices $\mathbf{C}_{ss}$ and $\mathbf{C}_{cc}$ are both of same size.

[70] An exception is the case of zero-correlation (white noise), where a 0/0 division would happen. Formally, the decorrelation efficiency would be 0 in that case.

tions computed from several basis vectors of the DCT, block length $M = 8$. Beneath a pass-band, each of the functions has significant side lobes, indicating that the frequency separation property of the DCT in this case is rather poor.

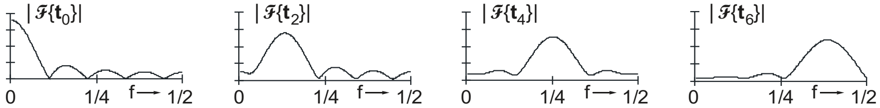

Fig. 2.42. Fourier amplitude spectra of DCT basis vectors $\mathbf{t}_0$, $\mathbf{t}_2$, $\mathbf{t}_4$, $\mathbf{t}_6$

Longer impulse responses (or basis functions) can improve the spectral cut-off and stop-band suppression. Windowing enforces the basis functions to roll off smoothly towards the tails and provides spectra with less energy in the side lobes compared to hard truncated functions. To prevent loss of information which is close to the tails of the window, basis functions of neighbored blocks need to overlap, such that synthesis can be performed by an overlap-add approach. The hop size between the start positions of two subsequent windows must not be larger than the number of transform coefficients U, such that the number of samples in the signal is not larger than the number of transform coefficients. With a hop size smaller than U, the transform would be over-complete; therefore, typically hop size U is used. This principle of *block-overlapping transforms* can still establish an orthogonal system of basis functions. Real-valued cosine-modulated functions combined with an appropriate weighting window are e.g. used in the *lapped orthogonal transform* (LOT) [MALVAR, STAELIN 1989] and in the TDAC transform (*time domain aliasing cancellation*) [PRINCEN, BRADLEY 1986]; more generally, this family of transforms is denoted as *cosine modulated filter banks* or *modified DCT*; a prominent application domain is audio signal compression (cf. Sec. 8.2.1).

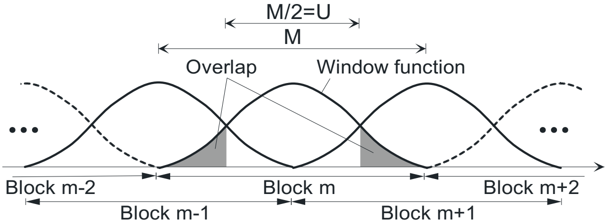

Fig. 2.43. Positions of analysis blocks with their overlapping window functions in a block-overlapping transform, M=2U

Example: TDAC transform [PRINCEN, BRADLEY 1986]. Here, decomposition is performed into U=M/2 frequency bands, the basis functions are based on an orthonormal version of the type-IV DCT (2.264), have a length M=2U and are defined as[71]

[71] Other overlap factors are possible, as long as the condition (2.290) holds.

$$t_k(n) = \sqrt{\frac{4}{M}} w(n) \cos\left[\frac{2\pi}{M}(k+0.5)(n+0.5-U/2)\right]. \qquad (2.289)$$

The orthonormality of the underlying DCT is not changed when multiplying the even or odd symmetric basis functions with an even-symmetric window function $w(n)$, and therefore the synthesis again uses the same set of basis functions. By this, the corresponding value of the window function is multiplied twice to a sample position in the overall signal flow. Therefore, when the entire sequence of window functions from all blocks is superimposed, their squared values must sum up to unity to achieve perfect reconstruction (see example from Fig. 2.44). Assuming a window function which is nonzero for $0 \le n < M-1$, this is fulfilled when the following conditions hold true in cases $M \le 2U$[72] and transition width $M-U$:

$$w^2(n) + w^2(n+U) = 1 \text{ for } 0 \le n < M-U \text{ and } w(n) = 1 \text{ for } M-U \le n < U\,. \qquad (2.290)$$

This also guarantees that the cosine basis functions are still orthonormal when considered jointly across all blocks,

$$\sum_{m=-M/U}^{M/U} \sum_{n=0}^{M-1} t_k(n+mU) t_l(n+mU) = \begin{cases} 1, \; k = l \\ 0, \; k \ne l. \end{cases} \qquad (2.291)$$

Typically, symmetric windows $w(n)$ are used, where $w(n)=w(M-1+n)$[73]. An example fulfilling (2.290) for the case $U=M/2$ (even M) is the sine window

$$w(m) = \sin\left(\frac{\pi}{M}\cdot(m+0{,}5)\right). \qquad (2.292)$$

Fig. 2.43 shows the Fourier-domain amplitudes of different TDAC basis functions using the sine window in case $U=8$, $M=16$. The side lobes of the spectra are largely reduced as compared to the DCT case in Fig 2.42, whereas the main lobes have become broader, which effects spectral overlap to occur mainly between directly neighbored frequency bands.

In the time domain, the block overlap causes smooth transitions between adjacent blocks instead of discontinuities in case of non-overlapping transforms, when coarse quantization or discarding of coefficients occurs during coding. This is particularly beneficial in case of periodic signals extending over block boundaries

[72] For $M>2U$, the hop size would be so small that more than two blocks overlap; in that case, the sum of squares from all window functions has to be constant.

[73] This symmetry is reasonable when the same window function is used over all transform blocks. This is however not necessary; moreover, orthonormality is still achieved whenever the entire time sequence over all squared window functions sums up to one. Furthermore, it is also possible to apply the squared window function *only during analysis* or *only during synthesis*, and apply a flat weighting (rectangular window with same overlap) at the other end. These properties allow adaptive switching of window lengths depending on signal properties, as often used in audio compression, or adaptive switching between overlapping and non-overlapping transforms, as used for image compression in the JPEG-XR standard.

(where phase discontinuities are avoided) and for signals with constant or smoothly increasing amplitude (avoiding unnatural amplitude discontinuities). On the other hand, the block overlap can be disadvantageous when the signal has discontinuities of the amplitude.

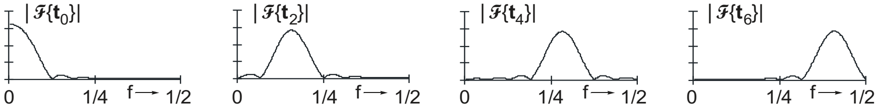

Fig. 2.44. Fourier amplitude spectra of TDAC basis vectors $\mathbf{t}_0$, $\mathbf{t}_2$, $\mathbf{t}_4$, $\mathbf{t}_6$

In principle, the transition shape of the window at the left and right boundary can also be different, still achieving perfect reconstruction under the condition that the complementary shape (squares summing up to unity) is used in the corresponding adjacent block. This enables switching between windows/transforms of different length M or overlapping and non-overlapping transform basis functions locally, still retaining perfect reconstruction.

2.8 Filterbank transforms

The general principle of a *filterbank transform*, with linear block and overlapping transforms as special cases, is shown in Fig. 2.45. Interpretation of a linear transform analysis as parallel convolution operation with sub-sampling was given in Sec. 2.7.4; the inverse transform (synthesis) can be interpreted in a similar way. Generalizing this principle without explicit consideration of block segmentation and analysis hop sizes allows to formulate properties of the basis functions (filter impulse responses) with even more flexibility.

The frequency analysis is performed using U parallel filters. Direct usage of the filter output samples would give a representation which is *over-complete* by a factor of U. Therefore, the output signals of the different frequency bands are subsampled (decimated) and those retained are used as transform coefficients. The maximum (critical) decimation factor providing a complete representation of an arbitrary signal and thus enabling perfect reconstruction is equal to the number of subbands (U:1), such that the total number of coefficients equals the number of samples in $s(n)$. During synthesis, the signal is reconstructed by *interpolation* of the different subband signals and subsequent superposition of all components.

When comparing the DCT and its overlapping variants in Sec. 2.7.4, the aspect of spectral separation properties was discussed. With a set of ideal equal-bandwidth filters, U non-overlapping frequency bands cover a bandwidth of $f_\Delta = 1/(2U)$ each, such that alias-free critical sub-sampling and reconstruction could be applied according to sampling theory. This is however not possible if causal filters or filters with finite impulse response shall be used. With non-ideal filters and critical sub-sampling, overlaps of frequency bands occur, as schematically shown in Fig. 2.46. Fig. 2.46a shows the amplitude transfer function of a lowpass filter[74], which is shifted in frequency to provide the transfer functions of modulated bandpass and highpass filters. The corresponding layout of the spectrum (up to half sampling rate of the original signal) is shown in Fig. 2.46b.

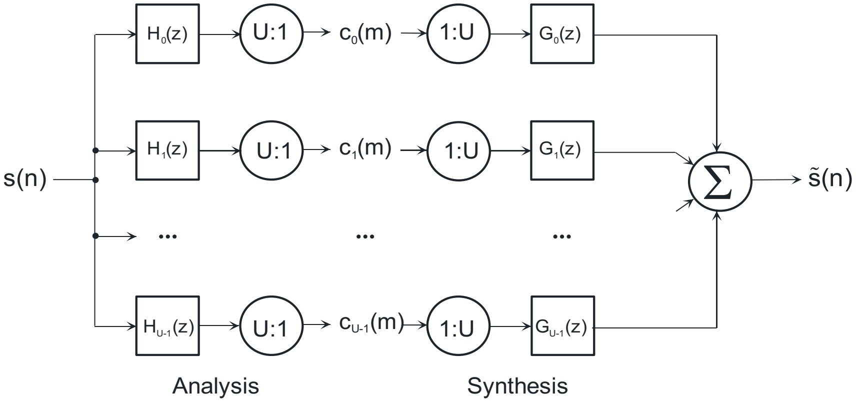

Fig. 2.45. Subband analysis and synthesis system, *U* frequency bands

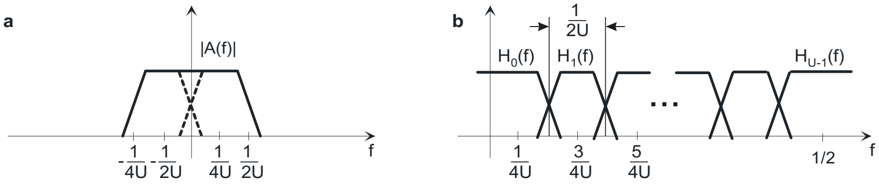

Fig. 2.46. a Lowpass filter **b** overlapping modulated bandpass filters

2.8.1 Decimation and interpolation

In case of discrete-time signals, scaling of the time axis has to be combined with downsampling (decimation) or upsampling (interpolation). The generation of a discrete signal $s_U(n)$, which is decimated by a factor U compared to the sampling of $s(n)$, is performed by discarding samples. The first step can be described as a multiplication by a train of Kronecker impulses

[74] This lowpass filter can actually be interpreted as a superposition of two complex-conjugate bandpass filter transfer functions at centre frequencies ±1/(4*U*). This allows to define bands of equal width in the range $0 \leq |f| < 1/2$. If the positive and negative parts of the spectra were regarded separately, the corresponding impulse responses would be complex.

$$s_{\delta_U}(n) = s(n) \sum_{m=-\infty}^{\infty} \delta(n - mU). \tag{2.293}$$

Subsequently, only each U^{th} value (i.e. one of the non-zero values) is retained without further information loss,

$$s_U(m) = s(mU) = s_{\delta_U}(mU). \tag{2.294}$$

The signals, $s(n)$, $s_{\delta_U}(n)$ and $s_U(m)$ are shown for the case U=2 in Fig. 2.47 left. The Fourier spectrum of the discrete Kronecker impulse sequence is a periodic sequence of Dirac impulses,

$$\sum_{m=-\infty}^{\infty} \delta(n - mU) \circ\!\!-\!\!\bullet \frac{1}{|U|} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{U}\right). \tag{2.295}$$

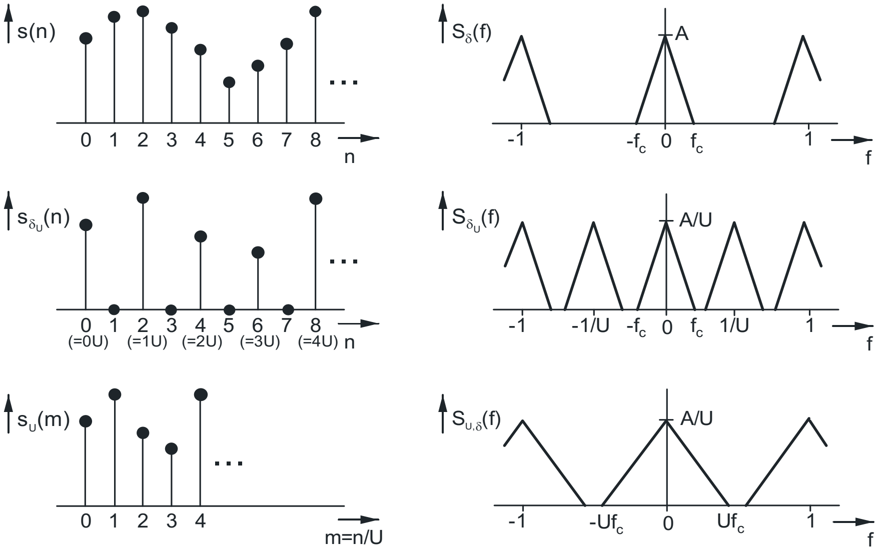

Fig. 2.47. Signals $s(n)$, $s_{\delta_U}(n)$, $s_U(m)$ and their spectra for the case U=2

The spectrum of the signal $s_{\delta_U}(n)$, which is sampled by rate $1/U$, can be expressed via the spectrum $S_\delta(f)$ of the signal $s(n)$, that was sampled from $s(t)$ with spectrum $S(f)$ with normalized rate f= 1, as

$$\begin{aligned} S_{\delta_U}(f) &= S_\delta(f) * \frac{1}{|U|} \sum_{k=0}^{U-1} \delta\left(f - \frac{k}{U}\right) \\ &= \frac{1}{|U|} \sum_{k=0}^{U-1} S_\delta\left(f - \frac{k}{U}\right) = \frac{1}{|U|} \sum_{k=-\infty}^{\infty} S\left(f - \frac{k}{U}\right). \end{aligned} \tag{2.296}$$

An identical spectrum $S_{\delta_U}(f)$ would show when the signal had originally been sampled by a rate $1/U$ (relative to f=1). If the signal was band limited to a maximum frequency $f_c = 1/(2U)$ before subsampling, no alias occurs. Computation of the spectrum is also possible directly from the subsampled signal,

$$\begin{aligned} S_{U,\delta}(f) &= \sum_{m=-\infty}^{\infty} s_U(m)\mathrm{e}^{-\mathrm{j}2\pi mf} = \sum_{m=-\infty}^{\infty} s_{\delta_U}(mU)\mathrm{e}^{-\mathrm{j}2\pi mf} \\ &= \sum_{n=-\infty}^{\infty} s_{\delta_U}(n)\mathrm{e}^{-\mathrm{j}2\pi n\frac{f}{U}} = S_{\delta_U}\left(\frac{f}{U}\right), \end{aligned} \tag{2.297}$$

and therefore

$$S_{U,\delta}(f) = \frac{1}{|U|}\sum_{k=-\infty}^{\infty} S\left(\frac{f-k}{U}\right). \tag{2.298}$$

In (2.297) and (2.298), the frequency is re-normalized by the new sampling rate $1/U$ which means that the frequency axis of $S_{U,\delta}(f)$ is scaled by a factor U compared to the frequency axis of $S_{\delta_U}(f)$. Fig. 2.47 shows the respective spectra next to the corresponding signals.

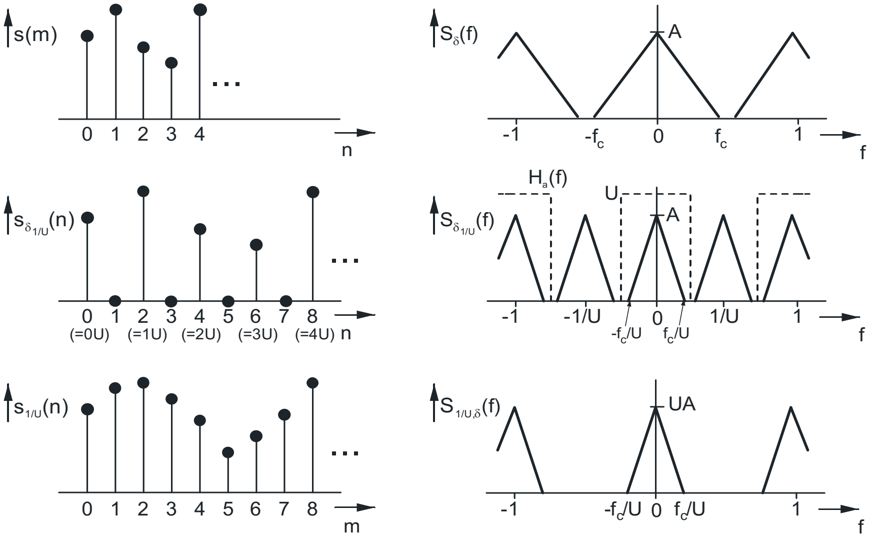

Fig. 2.48. Signals $s(m)$, $s_{\delta_{1/U}}(n)$, $s_{1/U}(n)$ and their spectra, example of upsampling by U=2

In interpolation, the increase of sampling rate by a factor U is achieved by inserting U–1 zero values between the available samples (Fig. 2.48):

$$s_{\delta_{1/U}}(n) = \begin{cases} s\left(\dfrac{n}{U}\right) & \text{for} \quad m = \dfrac{n}{U} \in \mathbb{Z} \\ 0 & \text{else.} \end{cases} \tag{2.299}$$

The related spectrum is scaled by a factor $1/U$ compared to the original spectrum $S_\delta(f)$,

$$S_{\delta_{1/U}}(f)=\sum_{n=-\infty}^{\infty} s_{\delta_{1/U}}(n)\mathrm{e}^{-\mathrm{j}2\pi nf}\ , \tag{2.300}$$

or alternatively

$$\begin{aligned}S_{\delta_{1/U}}(f)&=\sum_{m=-\infty}^{\infty} s_{\delta_{1/U}}(mU)\mathrm{e}^{-\mathrm{j}2\pi mUf}=\sum_{m=-\infty}^{\infty} s(m)\mathrm{e}^{-\mathrm{j}2\pi mUf}\\ &=S_{\delta}(Uf)=\sum_{k=-\infty}^{\infty} S\left(Uf-k\right)=\sum_{k=-\infty}^{\infty} S\left[U\left(f-\frac{k}{U}\right)\right].\end{aligned} \tag{2.301}$$

When the sampling rate is re-normalized to f=1, U spectral copies (including the original baseband) appear in the range $-1/2 \leq f < 1/2$. Lowpass filtering with cut-off frequency f_{c}=1/(2U) has to be applied to eliminate the U–1 alias copies and to generate the interpolated signal $s_{1/U}(n)$. Amplitude scaling by a factor of U is further necessary,

$$S_{1/U,\delta}(f)=S_{\delta_{1/U}}(f)H_{\mathrm{a}}(f)\ \text{ with }\ H_{\mathrm{a}}(f)=U\mathrm{rect}(Uf)*\sum_{k=-\infty}^{\infty}\delta(f-k)\,. \tag{2.302}$$

In the time domain, the impulse response of the lowpass filter (in ideal case a discrete-time sinc function) interpolates the missing values, leaving the originally available sampling positions m from (2.299) unchanged:

$$h(n)=\mathrm{si}\left(\frac{\pi n}{U}\right). \tag{2.303}$$

The spectrum of the interpolated signal $s_{1/U}(n)$ is

$$S_{1/U,\delta}(f)=|U|\sum_{k=-\infty}^{\infty} S\left[(f-k)U\right]=|U|S(Uf)*\sum_{k=-\infty}^{\infty}\delta(f-k)\,, \tag{2.304}$$

being identical to the spectrum of a signal that would have been originally sampled with a rate which is higher by a factor U,

$$s_{\delta_{1/U}}(t)=s(t)\sum_{n=-\infty}^{\infty}\delta\left(t-\frac{n}{U}\right)=\sum_{n=-\infty}^{\infty} s_{1/U}(n)\delta\left(t-\frac{n}{c}\right)\ \text{ with }\ s_{1/U}(n)=s\left(\frac{n}{U}\right). \tag{2.305}$$

The operations of decimation and interpolation described so far are only applicable with integer factors U. By combinations it is however possible to implement down- and upsampling by any rational factors, e.g. sampling rate conversion by a factor U_1/U_2 can be achieved by performing interpolation by a factor of U_1 followed by decimation by a factor U_2.

Interpolation in discrete time can be interpreted similar to continuous-time interpolation (2.51), it is however only performed at pre-defined positions. Before decimation, it is usually necessary to perform lowpass filtering to avoid alias, unless the signal is already appropriately band limited for the new sampling rate.

In case of re-sampling with rational factors, it is only necessary to compute those samples which would be retained after the second (decimation) step. This can be achieved by defining a set of interpolation filters typically having a target cut-off frequency

$$f_c = \min\left\{\frac{1}{2}, \frac{U_1}{2U_2}\right\}, \tag{2.306}$$

where the filters in the set have to be designed to support all phase shifts that can occur between the existing sampling positions. The number of filters to be defined for the non-existing re-sampling phase positions is $N_{\text{Ph}}=\max\{U_1-1,U_2-1\}$. However, due to the fact that in case of rational re-sampling factors always two of the phase positions are mirror symmetric relative to the original sampling grid (e.g. 1/4 and 3/4=1–1/4), it is usually only necessary to design $\lfloor N_{\text{Ph}}/2\rfloor+1$ different filters and re-use them with mirrored impulse response for the corresponding other position (an example with the interpolation filters of HEVC can be found in Tab. 7.1).

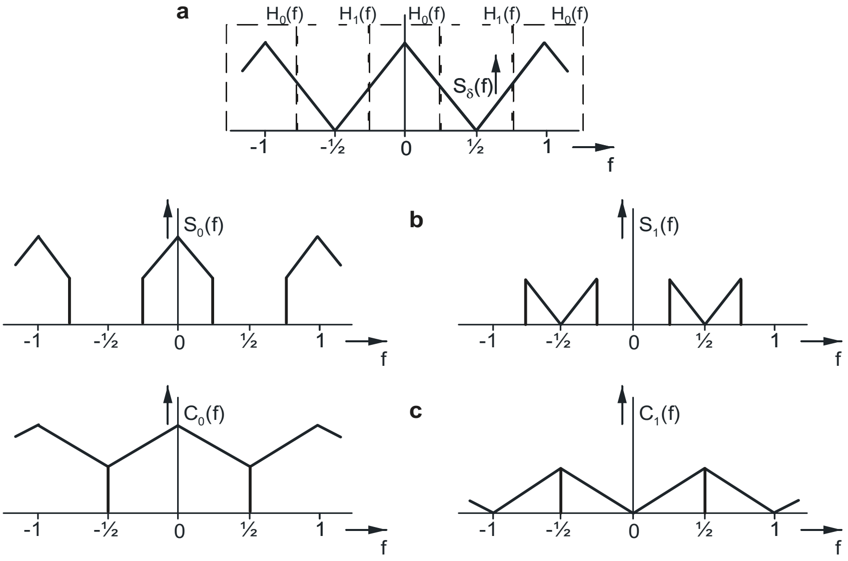

Fig. 2.49. Decomposition of a signal into decimated lowpass and highpass components **a** Signal spectrum **b** Spectra after lowpass/highpass filtering and multiplication by Kronecker impulse train **c** Spectra after sub-sampling

Further, the processes of decimation and interpolation are not restricted to lowpass signals as discussed so far, but can be applied to any appropriately band limited signals (e.g. bandpass outputs from the filter bank), such that no spectral overlaps occur. Fig. 2.49 illustrates decimation with U=2 applied in parallel to the *low* and *high frequency bands*, separated under assumption of ideal filters here. Fig. 2.49a shows the spectrum of the original signal, Fig. 2.49b the results after filtering and multiplication by the discrete sampling function (2.293), denoted as $S_k(f)=S_\delta(f)H_k(f)$, k=0 and k=1 for low and high components, respectively. By discarding the zero samples (Fig. 2.49c), the spectra $C_k(f)$ are expanded by the factor U=2. Observe that after sub-sampling the highpass spectra $C_1(f)$ appear over an *inverted frequency axis* around frequency zero, i.e. spectral components which originally were close to $f=1/2$ now appear around $f=0$, whereas components which were originally around $f=1/4$ are mapped into proximity of $f=1/2$ after sub-sampling. This phenomenon of frequency inversion likewise occurs in case of multiple-band filter banks within each *odd-indexed* band[75]

In the following sub-sections, it will be shown that perfect reconstruction can indeed be achieved even if the sub-sampling of the particular bands cannot be performed alias-free, i.e. different from the concept of Fig. 2.49, non-ideal filters are used for the separation. It is however then necessary to design the filter banks for the analysis and synthesis stages jointly, such that alias components are eliminated when the interpolated signals are superimposed; standalone alias-free interpolation of the different frequency band signals is no longer possible.

2.8.2 Properties of subband filters

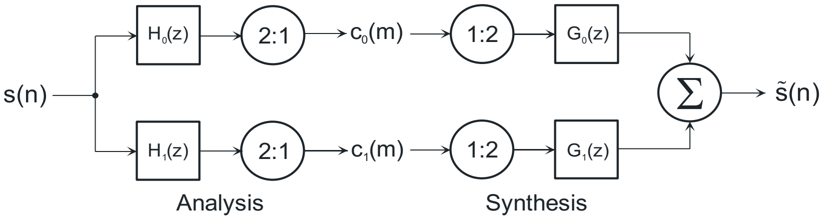

Fig. 2.50. Subband analysis system with U=2 frequency bands

Frequency transfer functions of the analysis filters from a critically-sampled filterbank overlap in case of non-ideal filters with finite impulse responses, such that alias can occurs by sub-sampling. Let $H_k(z)$ express the z-domain transfer functions of the analysis filters, $G_k(z)$ those of synthesis (interpolation) filters. For the case of $U=2$, which applies to the subband system in Fig. 2.50, only one lowpass band (k=0) and one highpass band (k=1) are generated. Hence, from (2.296), only one additional spectrum appears at $f=1/2$ or $z=-1$ due to subsampling after the filter operations $H_k(z)S(z)$. Over the complete system chain, the following spectrum appears after synthesis:

$$\tilde{S}(z)=\underbrace{\frac{1}{2}\left[H_0(z)G_0(z)+H_1(z)G_1(z)\right]S(z)}_{\text{baseband components}}+\underbrace{\frac{1}{2}\left[H_0(-z)G_0(z)+H_1(-z)G_1(z)\right]S(-z)}_{\text{alias components}}. \tag{2.307}$$

[75] This gives ground for yet another interpretation about the correlations between pairs of even-indexed or odd-indexed coefficients that can often be observed in block transforms. The bands overlap in frequency, which is one of the causes for correlation between formally orthogonal components after sub-sampling. On the other hand, as even and odd bands appear by original and reversed frequency order, linear relations are lost, such that the correlation is cancelled out again.

Quadrature mirror filters (QMF). In (2.307), the upper part expresses the components from the baseband spectrum, while the lower term contains alias components, which shall be eliminated. This can be achieved if the lower term has a value of zero. In the QMF construction, the highpass analysis filter $H_1(f)$ is derived from the lowpass filter $H_0(f)$ by time reverting, modulating by a discrete cosine with $f = 1/2$ and shifting the impulse response. Due to the symmetry in the lowpass transfer function around $f = 0$, lowpass and highpass functions are symmetric around the point $f = 1/4$ after the modulation. Modulation, time reversal and shift establish them as a system of *orthogonal functions*. Typical QMF relationships of the different filters in the signal domain and the spectral domains of f- and z-transfer functions are listed in Table 2.1. The relevant mapping relationships for the impulse responses and the Fourier and z transfer functions are also given in the lower part of the table.

Table 2.1. Definition of quadrature mirror filters (QMF): Relationships of orthogonal lowpass and highpass analysis and synthesis filters, expressed by impulse responses, z- and Fourier spectra

	$a(n)$	$A(f)$	$A(z)$
H_0	$h_0(n)=a(n)$	$H_0(f)=A(f)$	$H_0(z)=A(z)$
H_1	$h_1(n)=(-1)^{-1-n}\cdot a(-1-n)$	$H_1(f)=e^{-j2\pi f}\cdot A(1/2-f)$	$H_1(z)=z^{-1}\cdot A(-z^{--1})$
G_0	$g_0(n)=a(-n)$	$G_0(f)=A(-f)$	$G_0(z)=A(z^{-1})$
G_1	$g_1(n)=(-1)^{n+1}\cdot a(n+1)$	$G_1(f)=e^{j2\pi f}\cdot A(f-1(2)$	$G_1(z)=z\cdot A(-z)$
Inversion	$h(-n)$	$H(-f)= H^*(f)$	$H(z^{-1})$
Modulation	$(-1)^n h(n)$	$H(f-1/2)=H^*(1/2-f)$	$H(-z)=H(z\cdot e^{-j\pi})$

Substituting $z=\exp(j2\pi f)$ in (2.307) gives

$$\begin{aligned}\tilde{S}(f) &= \frac{1}{2}\left[H_0(f)G_0(f)+H_1(f)G_1(f)\right]S(f)\\ &+\frac{1}{2}\left[H_0(f-1/2)G_0(f)+H_1(f-1/2)G_1(f)\right]S(f-1/2).\end{aligned} \tag{2.308}$$

If the common model filter $A(f)$ as defined in Table 2.1 is used,

$$\begin{aligned}\tilde{S}(f) &= \frac{1}{2}\left[A(f)A(-f)+A(1/2-f)A(f-1/2)\right]S(f)\\ &+\frac{1}{2}\left[A(f-1/2)A(-f)+e^{j\pi}A(-f)A(f-1/2)\right]S(f-1/2).\end{aligned} \tag{2.309}$$

The alias component at $f=1/2$ is eliminated, and the condition

$$A(f)A^*(f)+A(1/2-f)A^*(1/2-f)=\left|A(f)\right|^2+\left|A(1/2-f)\right|^2=2 \qquad (2.310)$$

gives perfect reconstruction at the output. (2.310) is generalized to the case of an arbitrary number of U subbands by

$$\sum_{k=0}^{U-1}\left|H_k(f)\right|^2=U. \qquad (2.311)$$

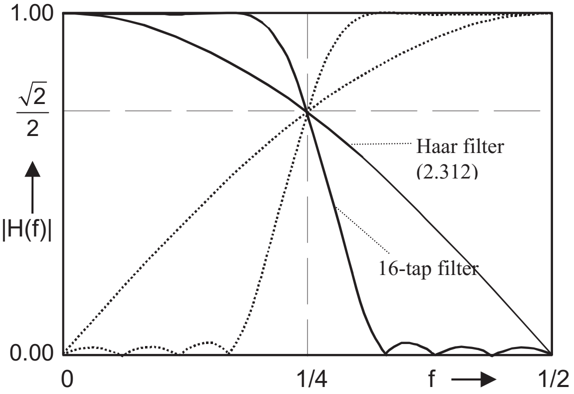

Fig. 2.51. Fourier magnitude transfer functions of filters from (2.312) and a 16-tap filter[76] [— lowpass ··· highpass]

Whereas the alias components are eliminated perfectly, the condition (2.311) for mathematically perfect reconstruction of the signal can only be fulfilled for two specific cases of QMF:

- Impulse response lengths are identical to the number of subbands U (which is the special case of block transforms, e.g. for U=2 the Haar filter basis);
- Ideal pass/stop band filters, which would require infinitely extended impulse responses (i.e. sinc function or modulated versions thereof).

Example: Haar filter basis. For the case U=M=2, the Haar filter defines the basis functions of almost any orthonormal block transforms, including DCT, Walsh, Hadamard, Haar transforms and the KLT as optimized for an AR(1) process. The z transfer functions according to Table 2.1 are[77]

[76] The 16-tap lowpass FIR filter has the z transfer function

$$H_0(z)=0.007\cdot z^7-0.02\cdot z^6+0.002\cdot z^5+0.046\cdot z^4-0.026\cdot z^3-0.099\cdot z^2+0.118\cdot z+0.472$$
$$+0.472\cdot z^{-1}+0.118\cdot z^{-2}-0.099\cdot z^{-3}-0.026\cdot z^{-4}+0.046\cdot z^{-5}+0.002\cdot z^{-6}-0.02\cdot z^{-7}+0.007\cdot z^{-8}.$$

[77] Another definition $H_1(z)=\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}\cdot z^{-1}$ (sign permutation) may alternatively be used.

$$
\begin{aligned}
H_0(z) &= F(z) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \cdot z^{-1} \\
H_1(z) &= z^{-1} \cdot F(-z^{-1}) = -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \cdot z^{-1}.
\end{aligned}
\tag{2.312}
$$

The Fourier spectrum gives with (2.311)

$$
|H_0(f)|^2 + |H_1(f)|^2 = \left(\frac{\sqrt{2}}{2}\right)^2 \cdot (2\cos\pi f)^2 + \left(\frac{\sqrt{2}}{2}\right)^2 \cdot (2\sin\pi f)^2 = 2\,. \tag{2.313}
$$

The disadvantage however is the flat decay of the amplitude transfer function, due to the short length of the filters, causing poor frequency separation property and eventually strong alias in the sub-sampled signals. Other finite-length filters constructed by the conditions of Table 2.1 will not fulfill (2.311) perfectly. The design of such filters is made as a compromise between frequency separation properties for alias suppression in the subbands, and a reconstruction error which should be kept as low as possible, such that

$$
U - \sum_{k=0}^{U-1} |H_k(f)|^2 \overset{!}{=} \min\,. \tag{2.314}
$$

Fig. 2.51 shows the amplitude transfer functions of the filters (2.312) and a pair of 16 tap QMF filters originally suggested in [Johnston 1980], where for the latter the value that can be computed from (2.314) is in the range of 10^{-4}.

To define more general conditions for alias-free *and* lossless reconstruction, the constraint of QMF, where H_0 and H_1 are mirror-symmetric, can be released. From (2.307), the elimination of the alias component is also achieved if the following conditions are met[78]:

$$
\begin{aligned}
G_0(z) &= \pm z^m \cdot H_1(-z), \\
G_1(z) &= \mp z^m \cdot H_0(-z).
\end{aligned}
\tag{2.315}
$$

Substituting (2.315) into (2.307) gives

$$
\tilde{S}(z) = \frac{1}{2}\left[H_0(z)H_1(-z) - H_1(z)H_0(-z)\right] S(z) z^m\,, \tag{2.316}
$$

which gives as condition for perfect reconstruction

$$
K(z) - K(-z) = 2z^{-m} \qquad \text{with} \qquad K(z) = H_0(z)H_1(-z)\,. \tag{2.317}
$$

Two types of filters, which are determined from (2.315)-(2.317), are introduced in

[78] Both combinations (± / ∓) are possible. The following equations in the explanation of PRF use the first option (±), whereas e.g. (2.325) uses (∓).

the following sub-sections. The term z^{-m} expresses an arbitrary shift which may occur anywhere in the analysis/synthesis chain. In image processing, filtering is often performed such that the current sample of the signal is weighted by the center sample of the impulse response (in case of odd length) or by one of the two center samples (in case of even length). As all filters introduced here are of FIR type[79], this has the effect that the reconstructed pictures are not spatially shifted.

Perfect reconstruction filters (PRF). For this type of filter, the basis functions of lowpass and highpass can be orthogonal, but the highpass impulse response may no longer be a modulated version of the lowpass response, no mirror symmetry exists. Typically, the resulting frequency bands have unequal widths. Besides the property of guaranteed perfect reconstruction, the filters have linear phase property, and with appropriate selection of the filter coefficients can be implemented using integer computations. (2.317) can be expressed by the following condition,

$$\det\left(\mathbf{K}(z)\right)=2z^{-m} \quad \text{with} \quad \mathbf{K}(z)=\begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix}. \tag{2.318}$$

The factorization of $P(z)$ into $H_0(z)$ and $H_1(-z)$ is now reduced into a problem to factorize the matrix $K(z)$, which shall have a determinant expressing a shift by m samples and multiplication by a factor of 2. The factorization is simplified, if the z polynomials are decomposed into polyphase components, where sub-responses of subscripts A and B contain only the even and odd samples of the impulse response, respectively:

$$H_k(z)=H_{k,\mathrm{A}}(z^2)+z^{-1}H_{k,\mathrm{B}}(z^2)\ . \tag{2.319}$$

Writing the polyphase components of the filter pair into the following polyphase matrix[80],

$$\mathbf{H}(z)=\begin{bmatrix} H_{0,\mathrm{A}}(z) & H_{0,\mathrm{B}}(z) \\ H_{1,\mathrm{A}}(z) & H_{1,\mathrm{B}}(z) \end{bmatrix}, \tag{2.320}$$

(2.318) will be fulfilled if (2.320) has $\det(\mathbf{H}(z^2))=z^{1-m}$. The following construction of polyphase matrices was suggested in [Vetterli, LeGall 1989]; observe that the leftmost matrix is the polyphase matrix of the Haar filter (2.312) which is further extended by the z polynomials expressed in the matrix product,

$$\mathbf{H}(z)=\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\cdot\prod_{p=1}^{P-1}\begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}\cdot\begin{bmatrix} 1 & \alpha_p \\ \alpha_p & 1 \end{bmatrix}. \tag{2.321}$$

[79] For IIR subband filters, see e.g. [Smith 1991].

[80] For a deeper discussion of polyphase systems, see Sec. 2.8.3.

H(*z*) is complemented by its inverse, which represents the polyphase components of the synthesis filters,

$$\mathbf{G}(z)=\begin{bmatrix} G_{0,A}(z) & G_{1,A}(z) \\ G_{0,B}(z) & G_{1,B}(z) \end{bmatrix}$$
$$=\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\cdot\prod_{p=1}^{P-1}\left[\begin{bmatrix} z^{-1} & 0 \\ 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} 1 & -\alpha_p \\ -\alpha_p & 1 \end{bmatrix}\cdot\frac{1}{1-\alpha_p{}^2}\right]. \tag{2.322}$$

The impulse response length of the filters $H_k(z)$ and $G_k(z)$ will then be $2P$.

Examples. For $P = 1$, the result from (2.321) and (2.322) is the Haar filter pair (2.323), using the alternative form of $H_1(z)$ as defined in the footnote on p. 123. For $P = 2$, the following set of filters is computed [LeGall, Tabatabai 1988):

$$\begin{aligned}
H_0(z) &= \frac{1}{\sqrt{2\left(\alpha^2-1\right)}}(1+\alpha z^{-1}+\alpha z^{-2}+z^{-3}),\\
H_1(z) &= \frac{1}{\sqrt{2\left(\alpha^2-1\right)}}(1+\alpha z^{-1}-\alpha z^{-2}-z^{-3}),\\
G_0(z) &= -H_1(-z)=\frac{1}{\sqrt{2\left(\alpha^2-1\right)}}(-1+\alpha z^{-1}+\alpha z^{-2}-z^{-3}),\\
G_1(z) &= H_0(-z)=\frac{1}{\sqrt{2\left(\alpha^2-1\right)}}(1-\alpha z^{-1}+\alpha z^{-2}-z^{-3}).
\end{aligned} \tag{2.324}$$

As an example, the normalization factor is 1/4 for $\alpha = 3$, which enables a division-free integer implementation. With (2.317), $K(z)-K(-z)=2z^{-3}$.

Biorthogonal filters. In the PRF construction described above, lowpass and highpass filter kernels are always of same length, and orthogonality still applies due to the Haar polyphase matrix in combination with the other symmetric matrix entries in (2.321). Even this relationship between the bases H_0 and H_1 can be waived[81]; (2.316) only requires the analysis highpass H_1 to be a '$-z$'-modulated version of the synthesis lowpass G_0, and the synthesis highpass G_1 shall be a '$-z$'-modulated version of the analysis lowpass H_0. Hence, a *bi-orthogonal* relationship shall exist between the pairs of analysis highpass / synthesis lowpass filters and the analysis lowpass / synthesis highpass filters. In this case, as for the following example sets of filters, lowpass and highpass impulse responses can also have

[81] Orthogonality is however an important property regarding encoding of frequency coefficients, see (5.47).

different lengths. Linear-phase properties are retained when the filters themselves have symmetric impulse responses, but are not required in general[82],

$$\begin{aligned} &H_0^{(5/3)}(z)=\frac{1}{8}\left(-z^2+2z+6+2z^{-1}-z^{-2}\right);\quad H_1^{(5/3)}(z)=\frac{1}{2}\left(-1+2z^{-1}-z^{-2}\right),\\ &G_0^{(5/3)}(z)=\frac{1}{2}\left(z+2+z^{-1}\right);\quad G_1^{(5/3)}(z)=\frac{1}{8}\left(-z^3-2z^2+6z-2-z^{-1}\right);\end{aligned} \tag{2.325}$$

$$\begin{aligned} H_0^{(9/7)}(z)&=0.027z^4-0.016z^3-0.078z^2+0.267z+0.603\\ &\quad+0.267z^{-1}-0.078z^{-2}-0.016z^{-3}+0.027z^{-4},\\ H_1^{(9/7)}(z)&=0.091z^2-0.057z-0.591+1.115z^{-1}\\ &\quad-0.591z^{-2}-0.057z^{-3}+0.091z^{-4}.\end{aligned} \tag{2.326}$$

Biorthogonal filters are often employed in the *Discrete Wavelet transform* (see Sec. 4.4.4). Certain constraints should be observed in the design, in particular that an iterative application of the lowpass filter on scaled (sub-sampled) signals shall still have the effect of a (stronger) lowpass filter.

2.8.3 Implementation of filterbank structures

If filterbanks are implemented according to the direct structures introduced so far, the complexity of realization for subband analysis and synthesis is considerably higher than with block transforms using fast transform algorithms. Methods which reduce the computational complexity are introduced here.

Cascaded two-band systems. If two-band systems from Fig. 2.50 are configured in a cascaded tree consisting of T subsequent stages, each output signal from a preceding stage of the cascade is again decomposed into two more narrow subbands, and a complete decomposition into $U=2^T$ subbands can be realized as shown in Fig. 2.52. Intermediate results are used as input to several filters at the subsequent stage, and the later stages use increasingly sub-sampled signals, which significantly reduces operations compared to a system with parallel filters. Due to the frequency inversion occurring in highpass band sub-sampling (see Fig. 2.49), any frequency band that stems from an odd number of highpass filter / decimation steps will be frequency inverted. For the subsequent level, it is therefore necessary to exchange the sequence of filters H_0 and H_1 if an arrangement of subbands by increasing frequency order is desirable[83].

[82] Both filters are sometimes modified, multiplying H_0 by $\sqrt{2}$ and dividing H_1 by $\sqrt{2}$, which almost approaches orthonormality at least for the case of the 9/7 filter. A shift m=1 is used w.r.t. (2.315). By the lengths of their lowpass/highpass analysis filter kernels, these two filter pairs are denoted as 5/3 and 9/7, respectively.

[83] This has an analogy with the distinction between Walsh and Hadamard transforms, where the iterative Hadamard development ignores the frequency reversion.

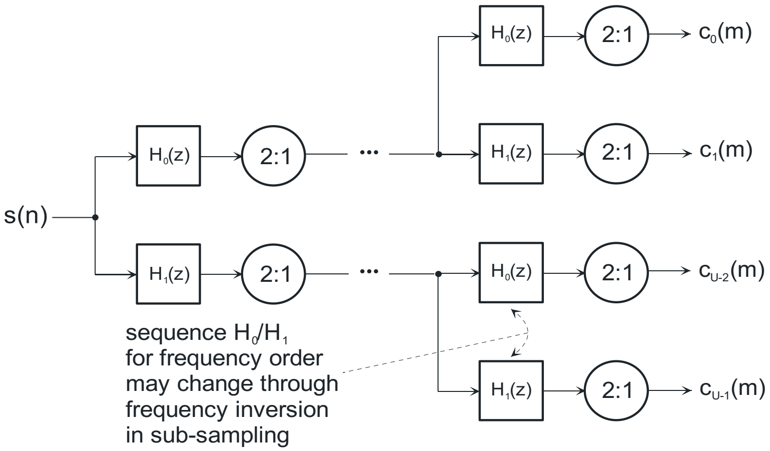

Fig. 2.52. Realization of subband analysis filter in a cascade from 2-band systems

Exploitation of filter symmetries. If symmetric (linear phase) filters are used, duplicate multiplications can be avoided, where samples have to be multiplied several times by identical factors. If the highpass basis function is a modulated version of the lowpass or uses the same multiplication factors (as in the cases of QMF and PRF types), yet another degree of freedom exists to reduce multiplications further by a factor of up to 2 by using results jointly (see also Problem 2.18).

Polyphase systems. Only each U^{th} sample will be retained by subband analysis after the filtering and sub-sampling steps. Hence, the convolution does not need to be performed at positions which are discarded anyway. This leads to a reduction of operations by a factor of U. The structure of a *polyphase system* is shown in Fig. 2.53 using an example of $U=2$. Sub-sampling is performed *prior to filtering*, whereby the signal is decomposed into U *polyphase components*, which establish a set of sample sequences each sub-sampled at a different phase position. Further, it is necessary to decompose the filter impulse responses into polyphase components, such that instead of a length-P filter, U partial filters of lengths either $\lfloor P/U \rfloor$ or $\lfloor P/U{+}1 \rfloor$ are obtained. If the subband filter impulse response h_k is decomposed into U polyphase components $h_{k,\mathrm{A}}(m)$, $h_{k,\mathrm{B}}(m)$, ... , U partial filters of transfer functions $H_{k,\mathrm{A}}(z)$, $H_{k,\mathrm{B}}(z)$, ... are given (see Fig. 2.53a for the case $U=2$). Similarly, it is not necessary to apply multiplications on zero values inserted for interpolation filtering at the synthesis stage. This can be realized by performing the interpolation filtering step within the polyphase components, and compose the different phase positions into the reconstructed signal only in a last step. In fact, the expansion of the signal is performed *after filtering*, but no zero values are actually inserted, as the polyphase components from all partial filters fill the corresponding gaps. The same reduction of the number of multiply/add operations by a factor of U is also achieved in synthesis (Fig. 2.53b).

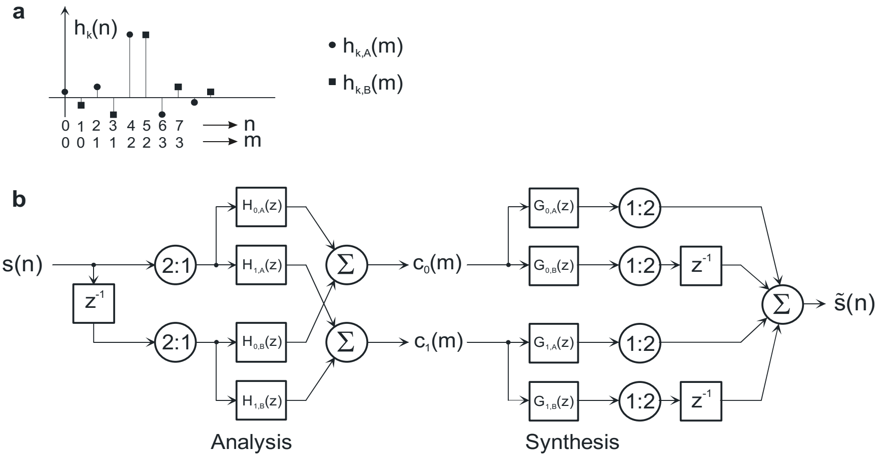

Fig. 2.53. Realization of subband analysis and synthesis by polyphase systems, U=2
a Separation of impulse response into partial terms $h_{\mathrm{A}}(n)$ und $h_{\mathrm{B}}(n)$
b Structure of the overall polyphase system

For a system with $U = 2$, the polyphase components of a signal $s(n)$ are sequences of even samples $s(2m)$ and odd samples $s(2m+1)$. In the z transform domain, the following relationships apply:

$$\begin{aligned} s(2m) &= s_{\mathrm{A}}(m) \overset{z}{\circ\!\!-\!\!\bullet} S_{\mathrm{A}}(z) = \frac{1}{2}\left[S\left(z^{1/2}\right) + S\left(-z^{1/2}\right)\right], \\ s(2m+1) &= s_{\mathrm{B}}(m) \overset{z}{\circ\!\!-\!\!\bullet} S_{\mathrm{B}}(z) = \frac{1}{2}\left[z^{1/2}S\left(z^{1/2}\right) - z^{1/2}S\left(-z^{1/2}\right)\right], \\ s(n) &\overset{z}{\circ\!\!-\!\!\bullet} S(z) = S_{\mathrm{A}}(z^2) + z^{-1}S_{\mathrm{B}}(z^2) \quad ; \quad m = \left\lfloor \frac{n}{2} \right\rfloor. \end{aligned} \tag{2.327}$$

Here, the subscripts A and B relate to the even and odd polyphase components, respectively. Formally, the components of the z transform can be combined in the following vector notation,

$$\mathbf{S}(z) = \begin{bmatrix} S_{\mathrm{A}}(z) \\ z^{-1}S_{\mathrm{B}}(z) \end{bmatrix}. \tag{2.328}$$

The same procedure can be applied to the z polynomials of the filter impulse responses. As the convolution in the signal domain corresponds to a multiplication in the z domain, the filtering of the even/odd signal spectra by the respective filter transfer functions can be expressed as

$$\underbrace{\begin{bmatrix} C_0(z) \\ C_1(z) \end{bmatrix}}_{\mathbf{C}(z)} = \underbrace{\begin{bmatrix} H_{0,\mathrm{A}}(z) & H_{0,\mathrm{B}}(z) \\ H_{1,\mathrm{A}}(z) & H_{1,\mathrm{B}}(z) \end{bmatrix}}_{\mathbf{H}(z)} \underbrace{\begin{bmatrix} S_{\mathrm{A}}(z) \\ z^{-1}S_{\mathrm{B}}(z) \end{bmatrix}}_{\mathbf{S}(z)}. \tag{2.329}$$

For the synthesis part, a similar principle applies. Writing the reconstructed signal by

$$\tilde{\mathbf{S}}(z) = \begin{bmatrix} \tilde{S}_{\mathrm{A}}(z) \\ z^{-1}\tilde{S}_{\mathrm{B}}(z) \end{bmatrix} \quad ; \quad \tilde{S}(z) = 2\left[\tilde{S}_{\mathrm{A}}(z^2) + z^{-1}\tilde{S}_{\mathrm{B}}(z^2)\right], \tag{2.330}$$

the synthesis filter step can be expressed as

$$\underbrace{\begin{bmatrix} \tilde{S}_{\mathrm{A}}(z) \\ z^{-1}\tilde{S}_{\mathrm{B}}(z) \end{bmatrix}}_{\tilde{\mathbf{S}}(z)} = \underbrace{\begin{bmatrix} G_{0,\mathrm{A}}(z) & G_{1,\mathrm{A}}(z) \\ G_{0,\mathrm{B}}(z) & G_{1,\mathrm{B}}(z) \end{bmatrix}}_{\mathbf{G}(z)} \underbrace{\begin{bmatrix} C_0(z) \\ C_1(z) \end{bmatrix}}_{\mathbf{C}(z)}. \tag{2.331}$$

Combining (2.329) and (2.331), the condition for perfect reconstruction is

$$\begin{bmatrix} G_{0,\mathrm{A}}(z) & G_{1,\mathrm{A}}(z) \\ G_{0,\mathrm{B}}(z) & G_{1,\mathrm{B}}(z) \end{bmatrix} \begin{bmatrix} H_{0,\mathrm{A}}(z) & H_{0,\mathrm{B}}(z) \\ H_{1,\mathrm{A}}(z) & H_{1,\mathrm{B}}(z) \end{bmatrix} = \mathbf{G}(z)\mathbf{H}(z) = \mathbf{I}\,, \tag{2.332}$$

from which the following relationships are determined:

$$\begin{aligned} G_{0,\mathrm{A}}(z)H_{0,\mathrm{A}}(z) + G_{1,\mathrm{A}}(z)H_{1,\mathrm{A}}(z) &= 1 \\ G_{0,\mathrm{A}}(z)H_{0,\mathrm{B}}(z) + G_{1,\mathrm{A}}(z)H_{1,\mathrm{B}}(z) &= 0 \\ G_{0,\mathrm{B}}(z)H_{0,\mathrm{A}}(z) + G_{1,\mathrm{B}}(z)H_{1,\mathrm{A}}(z) &= 0 \\ G_{0,B}(z)H_{0,\mathrm{B}}(z) + G_{1,\mathrm{B}}(z)H_{1,\mathrm{B}}(z) &= 1. \end{aligned} \tag{2.333}$$

These are fulfilled by the following conditions,

$$\begin{aligned} H_{0,\mathrm{A}}(z) = G_{1,\mathrm{B}}(z) \quad &; \quad H_{0,\mathrm{B}}(z) = -G_{1,\mathrm{A}}(z)\,; \\ H_{1,\mathrm{A}}(z) = -G_{0,\mathrm{B}}(z) \quad &; \quad H_{1,\mathrm{B}}(z) = G_{0,\mathrm{A}}(z), \end{aligned} \tag{2.334}$$

which by substitution into (2.333) gives the additional condition

$$H_{0,\mathrm{A}}(z)H_{1,\mathrm{B}}(z) - H_{0,\mathrm{B}}(z)H_{1,\mathrm{A}}(z) = \det\left(\mathbf{H}(z)\right) = 1\,. \tag{2.335}$$

Using (2.328) to express the polyphase filters in the (not downsampled) z domain, (2.333) is equivalent to (2.315), while (2.335) is equivalent to (2.317). A special case of the polyphase transform is observed for $\mathbf{H}(z) = \mathbf{G}(z) = \mathbf{I}$, which is the so-called *lazy transform* where the 'subband' signals $c_0(m)$ and $c_1(m)$ would simply be the polyphase components generated without any lowpass or highpass filtering.

Lifting implementation. Subband filters described by polyphase components can be implemented in a *lifting structure* [DAUBECHIES, SWELDENS 1998] as shown in Fig.

2.54. The first step of the lifting filter is a decomposition of the signal into its even- and odd-indexed polyphase components by the lazy transform. Then, the two basic operations are *prediction steps* $P(z)$ and *update steps* $U(z)$. The prediction and update filters have simple impulse responses typically of length 2 or 3; the number of steps necessary and the values of coefficients in each step are determined by a factorization of biorthogonal filter pairs. Finally, normalization by factors a_{Low} and a_{High} is applied.

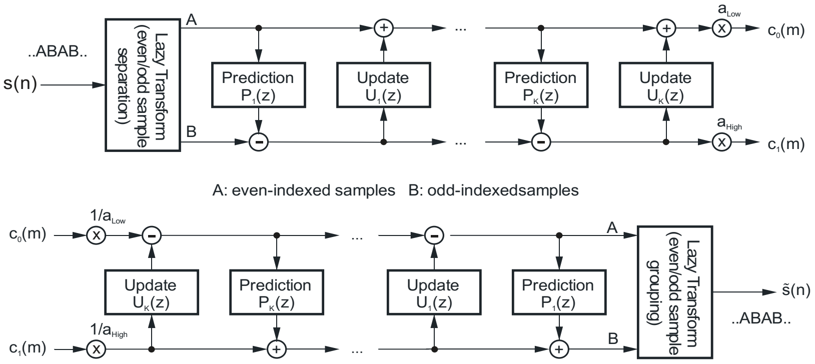

Fig. 2.54. Lifting structure of a subband filter (top: analysis; bottom: synthesis).

The construction of the prediction and update filter kernels can best be started from the polyphase representation. Assume that decomposition of a signal has been performed by a polyphase filter matrix $\mathbf{H}^0(z)$ (which could be the identity matrix $\mathbf{I}$ for the lazy transform in the beginning). If a *prediction step* is performed using the filter transfer function $P(z)$, the result is identical to a filter expressed by the polyphase matrix

$$\begin{aligned}\mathbf{H}^{\mathrm{pr}}(z) &= \underbrace{\begin{bmatrix} 1 & 0 \\ -P(z) & 1 \end{bmatrix}}_{\mathbf{P}(z)} \cdot \mathbf{H}^0(z) \\ &= \begin{bmatrix} H_{0,\mathrm{A}}(z) & H_{0,\mathrm{B}}(z) \\ H_{1,\mathrm{A}}(z) - P(z)H_{0,\mathrm{A}}(z) & H_{1,\mathrm{B}}(z) - P(z)H_{0,\mathrm{B}}(z) \end{bmatrix}.\end{aligned} \tag{2.336}$$

The complementary synthesis filter guarantees perfect reconstruction, such that $\mathbf{G}^{\mathrm{pr}}(z)\mathbf{H}^{\mathrm{pr}}(z) = \mathbf{I}$ when $\mathbf{G}^0(z)\mathbf{H}^0(z) = \mathbf{I}$,

$$\mathbf{G}^{\mathrm{pr}}(z) = \mathbf{G}^0(z) \cdot \begin{bmatrix} 1 & 0 \\ P(z) & 1 \end{bmatrix} = \begin{bmatrix} G_{0,\mathrm{A}}(z) + P(z)G_{1,\mathrm{A}}(z) & G_{1,\mathrm{A}}(z) \\ G_{0,\mathrm{B}}(z) + P(z)G_{1,\mathrm{B}}(z) & G_{1,\mathrm{B}}(z) \end{bmatrix}. \tag{2.337}$$

Similarly, a single *update step* can be formulated as

$$\begin{aligned}\mathbf{H}^{\mathrm{up}}(z)&=\underbrace{\begin{bmatrix}1 & U(z)\\ 0 & 1\end{bmatrix}}_{\mathbf{U}(z)}\cdot\mathbf{H}^{0}(z)\\ &=\begin{bmatrix}H_{0,\mathrm{A}}(z)+U(z)H_{1,\mathrm{A}}(z) & H_{0,\mathrm{B}}(z)+U(z)H_{1,\mathrm{B}}(z)\\ H_{1,\mathrm{A}}(z) & H_{1,\mathrm{B}}(z)\end{bmatrix},\end{aligned}\qquad(2.338)$$

where the complementary synthesis filter is

$$\mathbf{G}^{\mathrm{up}}(z)=\mathbf{G}^{0}(z)\cdot\begin{bmatrix}1 & -U(z)\\ 0 & 1\end{bmatrix}=\begin{bmatrix}G_{0,\mathrm{A}}(z) & G_{1,\mathrm{A}}(z)-U(z)G_{0,\mathrm{A}}(z)\\ G_{0,\mathrm{B}}(z) & G_{1,\mathrm{B}}(z)-U(z)G_{0,\mathrm{B}}(z)\end{bmatrix}.\qquad(2.339)$$

Using (2.336)-(2.339) iteratively starting by a lazy transform, the equivalent polyphase matrix after a number of subsequent prediction and update steps is the concatenated product of all matrices, e.g. for a number of L subsequent prediction and update steps

$$\mathbf{H}(z)=\begin{bmatrix}a_{\mathrm{Low}} & 0\\ 0 & a_{\mathrm{High}}\end{bmatrix}\prod_{l=1}^{L}\begin{bmatrix}1 & U_l(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -P_l(z) & 1\end{bmatrix}.\qquad(2.340)$$

Vice versa, it is possible to factorize a given polyphase matrix containing higher-order z polynomials into a series of prediction/update matrices with only simple, low-order polynomials. Separation of single prediction and update steps from a given (complete) polyphase matrix **H**(z) will result in the following expression according to (2.336) and (2.338):

$$\mathbf{H}(z)=\begin{bmatrix}1 & 0\\ -P(z) & 1\end{bmatrix}\cdot\mathbf{H}^{-\mathrm{pr}}(z)\quad;\quad\mathbf{H}(z)=\begin{bmatrix}1 & U(z)\\ 0 & 1\end{bmatrix}\cdot\mathbf{H}^{-\mathrm{up}}(z)\,.\qquad(2.341)$$

The factorization is always possible, as the determinant of any of the single prediction and update matrices is one, and hence inversion is possible. By polynomial division, the result can be computed step by step, and the factorization typically terminates when only a diagonal matrix with normalization factors a_{Low} and a_{High} is left.

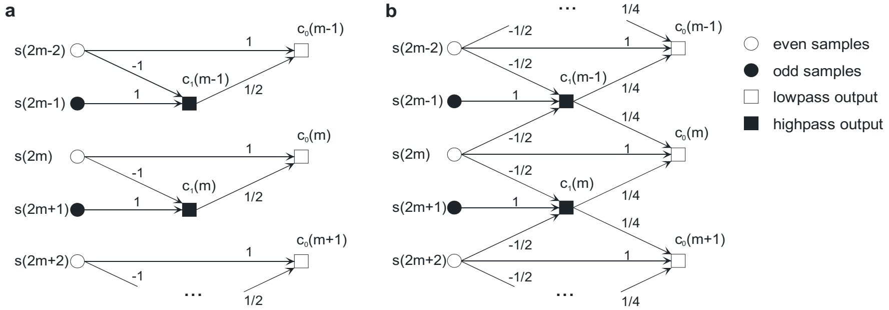

Fig. 2.55. Lifting flows of **a** Haar basis (2.312) **b** biorthogonal 5/3 filter (2.325)

Examples. The biorthogonal 5/3 filter from (2.325) can be expressed by the following polyphase matrix, which is further factorized into one normalization, one prediction and one update matrix

$$\mathbf{H}(z)=\begin{bmatrix}-\tfrac{1}{8}z+\tfrac{3}{4}-\tfrac{1}{8}z^{-1} & \tfrac{1}{4}z+\tfrac{1}{4}\\ -\tfrac{1}{2}-\tfrac{1}{2}z^{-1} & 1\end{bmatrix}=\underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{\mathbf{A}}\cdot\underbrace{\begin{bmatrix}1 & \tfrac{1}{4}z+\tfrac{1}{4}\\ 0 & 1\end{bmatrix}}_{\mathbf{U}(z)}\cdot\underbrace{\begin{bmatrix}1 & 0\\ -\tfrac{1}{2}-\tfrac{1}{2}z^{-1} & 1\end{bmatrix}}_{\mathbf{P}(z)}. \qquad (2.342)$$

Here, $a_{\text{High}}=a_{\text{Low}}=1$, $P(z)=½(z^{-1}+1)$ and $U(z)=¼(1+z)$. Another example is for the Haar filter[84], where $a_{\text{Low}}=\sqrt{2}$, $a_{\text{High}}=\sqrt{2}/2$, $P(z)=-1$ and $U(z)=1/2$:

$$\mathbf{H}(z)=\frac{\sqrt{2}}{2}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}=\underbrace{\begin{bmatrix}\sqrt{2} & 0\\ 0 & \sqrt{2}/2\end{bmatrix}}_{\mathbf{A}}\underbrace{\begin{bmatrix}1 & \tfrac{1}{2}\\ 0 & 1\end{bmatrix}}_{\mathbf{U}(z)}\underbrace{\begin{bmatrix}1 & 0\\ -1 & 1\end{bmatrix}}_{\mathbf{P}(z)}. \qquad (2.343)$$

The lifting structure can also be interpreted by a signal flow diagram, which is shown in Fig. 2.55 for the examples of a Haar filter (2.343) (without considering the normalization factors) and the biorthogonal 5/3 filter (2.342).

The lifting structure further allows definition of *nonlinear subband filters*. A simple example is usage of rank-order filters like median or weighted median filters in prediction and update steps [CLAYPOOLE ET AL. 1997]

2.8.4 Wavelet transform

The continuous-time *wavelet transform* (WT) is defined by the convolution equation

$$\mathcal{W}_s(t,f)=\int_{-\infty}^{\infty} s(\tau)\psi_f(t-\tau)\mathrm{d}\tau\,, \qquad (2.344)$$

being based on bandpass filter kernels

$$\psi_f(t)=\frac{1}{\sqrt{\alpha}}\cdot\psi\left(\frac{t}{\alpha}\right) \quad \text{with} \quad \alpha=\frac{f_0}{f}\,. \qquad (2.345)$$

The function $\psi(\cdot)$ is the *mother wavelet*, which is a bandpass filter of center fre-

[84] In the case of the Haar filter, the usage of the lifting approach does not to give an advantage in terms of complexity for signal decomposition, which is due to the fact that the polyphase polynomials already are of order zero before the factorization. This method is however relevant in motion-compensated temporal-axis wavelet filtering, cf. Sec. 7.3.2, and can also be used to avoid bit-depth extension of the transformed representation.

quency f_0, which is time-scaled by the factor α when intended to operate at a different frequency [RIOUL, VETTERLI 1991].

The continuous WT in (2.344) is not useful for practical signal analysis. It is highly overcomplete, being defined for an infinite number of instances both of time and frequency positions. In the *discrete wavelet transform* (DWT), the analysis shall only be performed for discrete (sampled) signal positions, and only for a discrete set of frequencies. The commonly used method is defining a set of basis functions by a *dyadic frequency sampling scheme*, where the upper band limits f_k and the distances of sampling positions t_k used for the respective frequency bands are defined over power-of-two relationships, such that the frequency partitioning has octave-band style. Assume that U frequency bands are defined[85] by

$$\alpha_k = 2^{U-k}, \quad f_k = \frac{1}{\alpha_k T}, \quad t_k(n) = \alpha_k nT \quad \text{with} \quad 0 \le k < U. \tag{2.346}$$

The distances between discrete center frequencies of the analysis are no longer constant, and the effective bandwidth[86] $\Delta f_k = [\, f_k - f_{k-1} \,]$ of the frequency bands is increased by a factor of 2 when incrementing k. Simultaneously, the distance between analysis positions $\Delta t_\mathrm{k} = [\, t_k(n) - t_k(n-1) \,]$ decreases by a factor of two. This means that for higher frequency bands (higher k), the temporal resolution becomes more precise, while less precision in the resolution of the frequency axis is achieved. This is illustrated in Fig. 2.56 for both cases of an idealized DWT and a discrete short time Fourier transform (STFT), which is typically implemented via windowed DFT or DCT analysis. Using the definitions in (2.346), the DWT coefficient of discrete frequency k and position n is defined as

$$c_k(n) = \frac{1}{\sqrt{\alpha_k}} \int_{-\infty}^{\infty} s(\tau)\,\psi\!\left(\frac{\tau - nT}{\alpha_k}\right) d\tau\,. \tag{2.347}$$

Remark that the basis functions defined here to compute the DWT are time-continuous and have the purpose to perform filtering for band limitation, whereas the convolution is only defined at discrete positions, such that sampling is implicitly included. As in (2.346), T is the sampling distance corresponding to the reso-

[85] In principle, the number U could become arbitrarily high, however for discrete signals of finite length N, at least one sampling position $t_u(m)$ should be retained in the last step. The condition $t_1(1)-t_1(0)\le NT$ gives e.g. $U_\mathrm{max}=\log_2 N$ for cases where N is a power of 2, or $\lfloor \log_2 N \rfloor+1$ otherwise. For practical applications, a much lower (pre-defined) number of bands is used for discrete wavelet decomposition. To be consistent with previous notation, we use the variable k as an index that increases with the frequency (starting with $k=0$ for the lowest frequency, whereas $k=U$ would be the original signal (without wavelet decomposition).

[86] The term ‘effective bandwidth’ is not precisely defined, except for the case of ideal filters. One possible way of interpretation is the width of a rectangular function with identical maximum amplitude and total integration area as the filter’s Fourier transfer function has.

lution accuracy when all frequency bands are used (i.e. original sampling before DWT decomposition is applied).

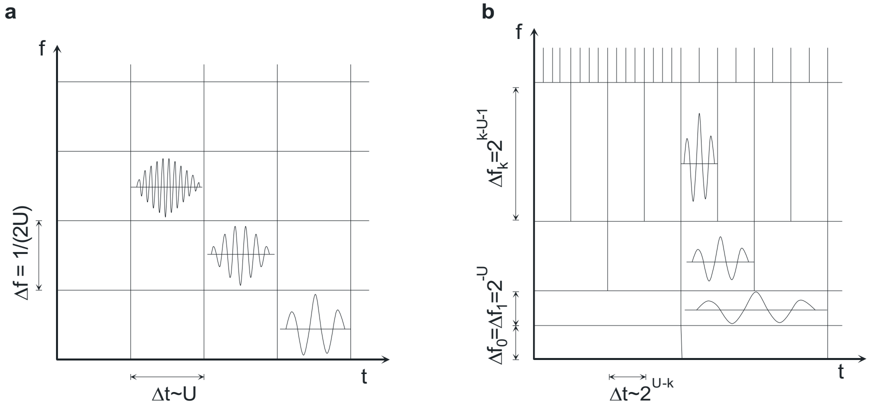

Fig. 2.56. Resolution accuracy in signal and frequency domains **a** for STFT **b** for DWT

The DWT allows reconstructing the signal by different resolution levels (scales). In a more abstract sense, the frequency domain representation up to half sampling rate can be constructed from a set of *scale spaces* and a set of *wavelet spaces*, each of which is related to one of the dyadic resolution levels (see Fig. 2.57). When the scale space $\mathcal{V}_k$ represents a certain bandwidth resolution of a (sampled) signal $s_k(n)$, the next-lower scale space $\mathcal{V}_{k-1}$ represents a signal $s_{k-1}(n)$ with half number of samples and half bandwidth. The scale space $\mathcal{V}_U$ represents the signal $s(n)=s_U(n)$ with maximum possible resolution, relating to a sampling distance T=1, which then corresponds to the frequency cut-off $|f|=1/2$. To achieve the perfect approximation, the wavelet space $\mathcal{W}_k$ must be an orthogonal complement which contains the residual between two adjacent scale spaces:

$$\mathcal{V}_{k+1}=\mathcal{V}_k\oplus\mathcal{W}_k \quad \text{and} \quad \mathcal{V}_k\perp\mathcal{W}_k \tag{2.348}$$

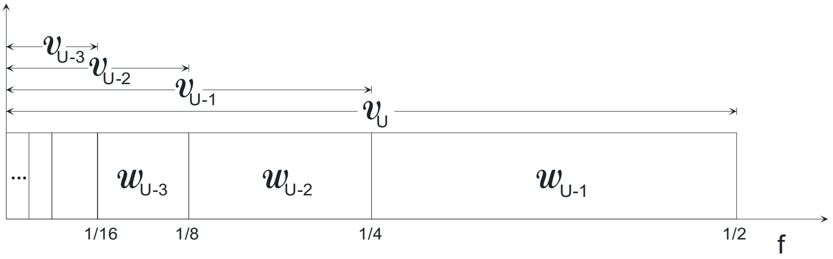

Fig. 2.57. Layout of dyadic scale and wavelet spaces by partitioning of the frequency axis

If the conditions in (2.348) hold true, all lower-frequency wavelet spaces must be orthogonal as well. All details which are lost when reducing the resolution from $\mathcal{V}_k$ to $\mathcal{V}_{k-1}$ are found in $\mathcal{W}_{k-1}$. Iteratively, an arbitrary scale space can be expressed as a direct sum of all lower-indexed wavelet spaces, where the summation is terminated by the lowest-resolution scale space[87]:

$$\mathcal{V}_k = \mathcal{W}_{k-1} \oplus \mathcal{W}_{k-2} \oplus \ldots \oplus \mathcal{W}_1 \oplus \mathcal{V}_1 \,. \tag{2.349}$$

The analysis of the signal, i.e. the decomposition into components which relate to the respective scale and wavelet spaces, is performed by *scaling functions* $\varphi(\tau)$ and *wavelet functions* $\psi(\tau)$. The scaling function is in principle a lowpass filter which is used to generate a lower-resolution representation, e.g. to construct $\mathcal{V}_{k-1}$ out of $\mathcal{V}_k$.

As $\mathcal{V}_{k-1} \subset \mathcal{V}_k$, any function in $\mathcal{V}_{k-1}$ can be expressed as a linear combination of basis functions $\varphi_k(\tau)$ related to the scale space $\mathcal{V}_k$. Therefore, also the scaling function in $\mathcal{V}_{k-1}$ can be described by the *refinement equation* expressing a superposition of scaling functions in $\mathcal{V}_k$:

$$\varphi_{k-1}(\tau) = \sum_m h_0(m)\varphi_k(\tau - m\alpha_k T) \,. \tag{2.350}$$

As for the wavelet space $\mathcal{W}_{k-1} \subset \mathcal{V}_k$ is also valid, an associated wavelet function can be generated similarly by the *wavelet equation*

$$\psi_{k-1}(\tau) = \sum_m h_1(m)\varphi_k(\tau - m\alpha_k T) \,. \tag{2.351}$$

Likewise, the operations (2.350)/(2.351) can be reversed, such that the next-higher scaling function (representing a signal of higher resolution) shall be reconstructed from a current level's scaling and wavelet functions as

$$\begin{aligned} \varphi_k(\tau) &= \sum_m g_{0,\mathrm{A}}(m)\cdot\varphi_{k-1}(\tau - m\alpha_{k-1}T) + \sum_m g_{1,\mathrm{A}}(m)\cdot\psi_{k-1}(\tau - m\alpha_{k-1}T) \\ \varphi_k(\tau - \alpha_k T) &= \sum_m g_{0,\mathrm{B}}(m)\cdot\varphi_{k-1}(\tau - m\alpha_{k-1}T) + \sum_m g_{1,\mathrm{B}}(m)\cdot\psi_{k-1}(\tau - m\alpha_{k-1}T), \end{aligned} \tag{2.352}$$

where A and B denote the even and odd polyphase components of the discrete filter functions.

The iterative development of scaling and wavelet functions shall now be illustrated for the simplest possible orthogonal wavelet basis, which is the Haar basis. The refinement and wavelet equations to perform the mapping from $\mathcal{V}_{k+1}$ into $\mathcal{V}_k$ and $\mathcal{W}_k$, using the discrete filter coefficients (2.312), give

[87] The termination by a scale space is necessary if the analyzed signal is finite, or if the delay occurring by the analysis shall be finite, as is typically always the case in multimedia signal processing and analysis. Theoretically, a scale space could also be established from an infinite series of sub-ordinate wavelet spaces. To make the notation consistent with the previous frequency representations that are introduced, the signal in the lowest-resolution scale space $\mathcal{V}_1$ is either denoted as $s_1(n)$ or $c_0(n)$.

$$\begin{aligned}\varphi_{k-1}(\tau)&=\underbrace{\frac{\sqrt{2}}{2}}_{h_0(0)}\varphi_k(\tau)+\underbrace{\frac{\sqrt{2}}{2}}_{h_0(1)}\varphi_k(\tau-\alpha_k T)\\ \psi_{k-1}(\tau)&=\underbrace{\frac{\sqrt{2}}{2}}_{h_1(0)}\varphi_k(\tau)-\underbrace{\frac{\sqrt{2}}{2}}_{h_1(1)}\varphi_k(\tau-\alpha_k T)\end{aligned}\quad. \qquad (2.353)$$

The scaling function $\varphi_U(\tau)$ in $\mathcal{V}_U$ is a rectangle ('hold element' in sampling) of length T and amplitude 1. Fig. 2.58 shows the weighted superposition of two copies of this scaling function, resulting in the scaling and wavelet functions in $\mathcal{V}_{U-1}$ and $\mathcal{W}_{U-1}$, respectively. If this is performed iteratively, both functions are scaled to double width and are amplitude-scaled by another factor of $\sqrt{2}$ with each subsequent iteration step. For this case, the convergence into the final shape of scaling and wavelet functions is already achieved after one iteration (which is due to the fact that the shape of the scaling function will always remain the rectangle, regardless how wide it may become).

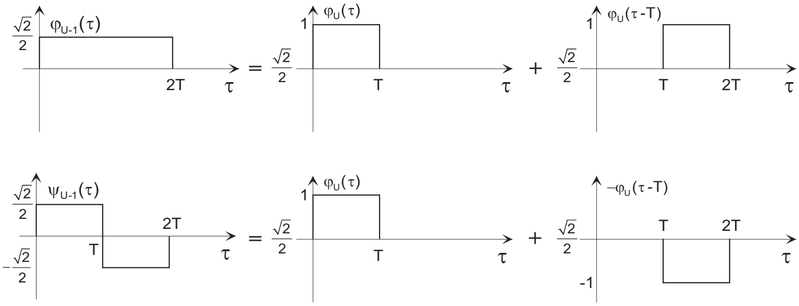

Fig. 2.58. Development of next-higher level scaling and wavelet functions for the Haar basis

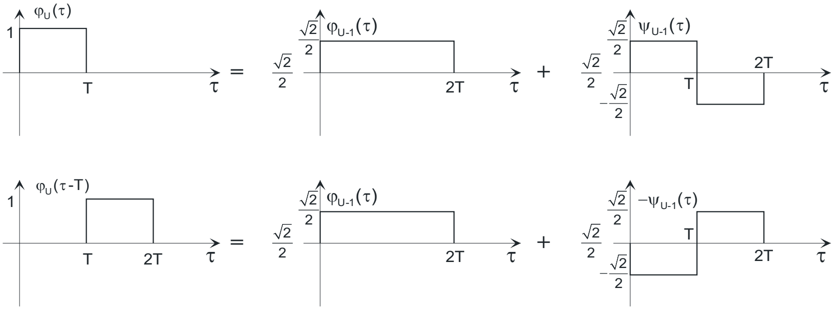

Fig. 2.59. Reconstruction of next-lower level scaling functions for the Haar basis

Now, reconstruction of the different copies of the scaling function in $\mathcal{V}_k$ shall be performed from the scaling and wavelet functions in $\mathcal{V}_{k-1}$ and $\mathcal{W}_{k-1}$. The related equations are

$$\varphi_k(\tau) = \underbrace{\frac{\sqrt{2}}{2}}_{g_0(0)}\varphi_{k-1}(\tau) + \underbrace{\frac{\sqrt{2}}{2}}_{g_1(0)}\psi_k(\tau)\,;\,\varphi_{k+1}(\tau-\alpha_k T) = \underbrace{\frac{\sqrt{2}}{2}}_{g_0(1)}\varphi_{k-1}(\tau) - \underbrace{\frac{\sqrt{2}}{2}}_{g_1(1)}\psi_{k-1}(\tau)\,. \quad (2.354)$$

This process of reconstruction is shown in Fig. 2.59.

(2.350) and (2.351) are the key equations of the DWT. They can be used to determine discrete lowpass and highpass analysis filter coefficients $h_0(k)$ and $h_1(k)$ of a filter bank system. Assume that continuous-time scaling and wavelet functions shall be orthogonal. If such functions can be constructed iteratively using discrete filter coefficients $h_0(k)$ and $h_1(k)$, any signal decomposition performed by these coefficients in a filterbank system would be orthogonal as well in case of large number of iterations, even if the impulse responses $h_0(k)$ and $h_1(k)$ may not be orthogonal. Furthermore, the scaling functions, even though they play conceptually a similar role as a band-limiting lowpass filter in conventional sampling, do not necessarily need to provide perfect band separation. When only the full set of spaces is relevant in a wavelet representation, orthogonality needs to be observed just between the underlying continuous scaling and wavelet functions, which is a much weaker condition than non-overlapping frequency bands. The orthogonality of the decomposition is guaranteed if the following condition holds true[88]:

$$\int_{-\infty}^{\infty} \varphi(\tau)\psi(\tau)\,d\tau = 0 \quad \text{where} \quad \varphi(\tau) = \lim_{U\to\infty}\varphi_1(\tau) \quad \text{and} \quad \psi(\tau) = \lim_{U\to\infty}\psi_1(\tau)\,. \quad (2.355)$$

This leads to conditions which can be used to design biorthogonal filter pairs. The continuous scaling and wavelet functions can be used to reconstruct (interpolate) continuous signals from the samples in the DWT domain. The discrete coefficients according to (2.350) and (2.351) can also directly be used to perform all underlying operations directly in the sampled signal domain (i.e. compute DWT from a sampled signal). Assume that a discrete approximation of the signal is available in some resolution scale k as $s_k(n)$. The scaling coefficients of the next-coarser approximation (representing a signal of half resolution or half number of samples) are then computed as

$$s_{k-1}(n) = \sum_m h_0(m)s_k(2n-m)\,, \quad (2.356)$$

and the complementary wavelet coefficients are

[88] The limit transitions in the following equation assume that the iterative construction of the scaling and wavelet functions could be continued ad infinitum (not stopping at k=0, which would be the case with a finite number of U bands, but rather continue with negative values). In contrast to the previous definitions in (2.350)-(2.354), time-axis scaling by a factor of 2 may be performed during each iteration of the continuous scaling and wavelet functions (corresponding to the subsampling in the discrete filterbank) to prevent infinite extension. This also implies, that even starting from an initial rectangular scaling function, final functions $\varphi(\tau)$ and $\psi(\tau)$ are becoming smooth with appropriate choice of the coefficients h and g, provided that the initial function has lowpass characteristics.

$$c_{k-1}(n)=\sum_{m}h_1(m)s_k(2n-m)\,. \tag{2.357}$$

This decomposition can be computed iteratively, starting by $s_U(n)\equiv s(n)$. Actually, each level of this decomposition is identical to the decomposition of a signal into low- and high-frequency subbands as introduced in Sec. 4.4.2. However, in contrast to the cascaded system from Fig. 2.52, only the low frequency output (the next lower scale signal s_{k-1}) is subject to further decomposition. Using the corresponding synthesis functions, it is possible to compute the reconstruction of the signal by inverting the sequence of recursion defining the *inverse DWT* (IDWT):

$$s_k(n)=\sum_{m}g_0(n-2m)s_{k-1}(m)+\sum_{m}g_1(n-2m)c_{k-1}(m)\,. \tag{2.358}$$

Note that (2.356)-(2.358) implicitly include polyphase operations in the expression of the discrete convolutions. As shown earlier, perfect reconstruction is possible if the synthesis coefficients $g_0(k)$ and $g_1(k)$ are related to $h_0(k)$ and $h_1(k)$ by bi-orthogonality (4.170). However, if $h_0(k)$ and $h_1(k)$ are chosen such that the continuous scaling and wavelet functions are orthogonal, it can be concluded that the sequences of discrete scaling and wavelet coefficients will also be orthogonal, even if the filter basis may only be biorthogonal (the latter being sufficient to achieve perfect reconstruction).

Fig. 2.60 shows the block diagram of a DWT analysis/synthesis filter bank, and a schematic layout of the resulting frequency decomposition, which can be described as an *octave-band structure*. For consistency with the notation used in case of other transforms, the signal relating to the scale space $\mathcal{V}_1$ is denoted as c_0 (instead s_1), while the designations of the wavelet coefficients relating to wavelet spaces $\mathcal{W}_k$, k= 1, … ,U–1, are retained as c_k as above.

If the Haar basis (2.312) is used, the resulting decomposition is exactly the same as for the Haar transform, cf. (2.249). However, longer filter impulse responses can provide a better frequency separation and also better alias suppression in the scaled signal versions.

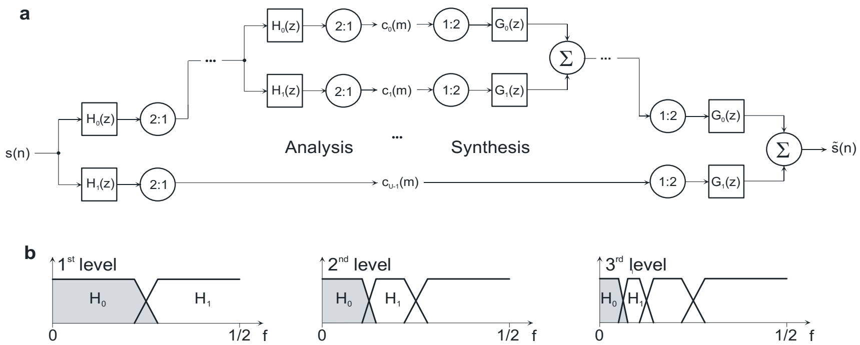

Fig. 2.60. **a** Octave-band filter bank system for DWT and IDWT **b** Octave-band frequency layout (3 levels of analysis)

For many classes of signals, in particular for natural image signals, the higher accuracy of frequency resolution for the lower-frequency bands provides a good fit with signal models. According to the AR(1) model with $\rho\rightarrow 1$, significantly more low-frequency than high-frequency components can be expected. For the high-frequency components, accurate frequency analysis is less important than an *accurate localization of detail*, in particular if a signal potentially exhibits discontinuities, as it is the case in edge areas (which are not adequately captured by the AR model). The fact that discontinuities in the signal appear at various resolution levels, and therefore also across wavelet bands at the same location, is denoted as *scaling property*.

2.8.5 Two- and multi-dimensional filter banks

The simplest realization of a two- or multi-dimensional filter bank is the *separable* method, where the analysis and synthesis filters are a product of horizontal and vertical filters. For the 2D case, the basis functions for the frequency band of index k_1 in horizontal and k_2 in vertical direction can be described as

$$h_{k_1,k_2}(n_1,n_2) = h_{k_1}(n_1)h_{k_2}(n_2) \text{ and } g_{k_1,k_2}(n_1,n_2) = g_{k_1}(n_1)g_{k_2}(n_2). \qquad (2.359)$$

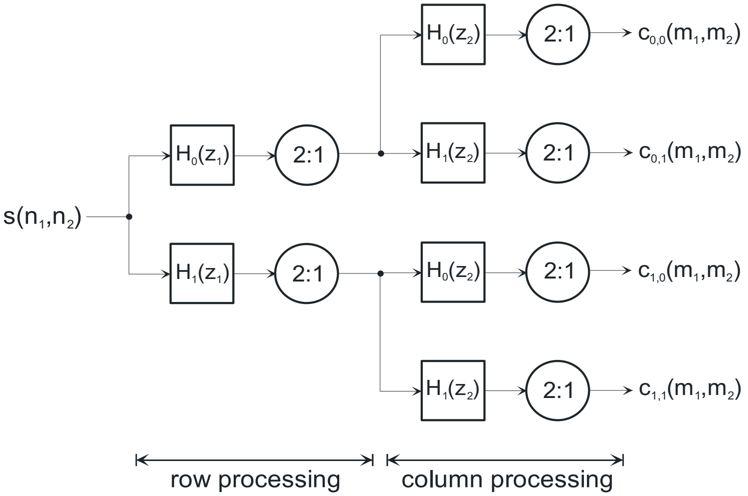

Fig 2.61. 2D system for decomposition into four frequency bands

With U_1U_2 bands, the total sub-sampling factor is $|\mathbf{U}| = U_1U_2$ in case of critical sampling. Separable 2D systems with 2-band decomposition structures per dimension can be realized sequentially, such that filtering and sub-sampling is at first performed over one dimension. Only a reduced number of samples then needs to be fed into the second directional decomposition stage. A block diagram with the case U_1=2, U_2=2, |U|=4, is shown in Fig. 2.61.

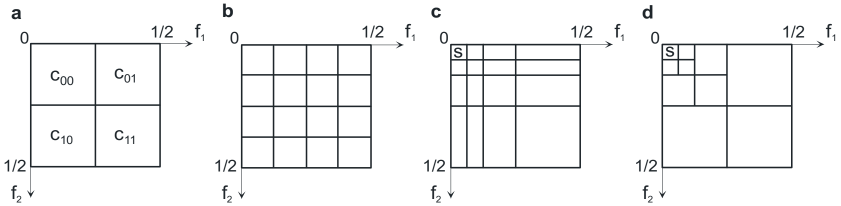

Fig. 2.62. Layout of 2D frequency bands. **a** 4 band elementary decomposition **b** 16 bands of equal bandwidth **c** Separable octave-band, 16 bands **d** 2D DWT, 10 bands (**S** = scaling band)

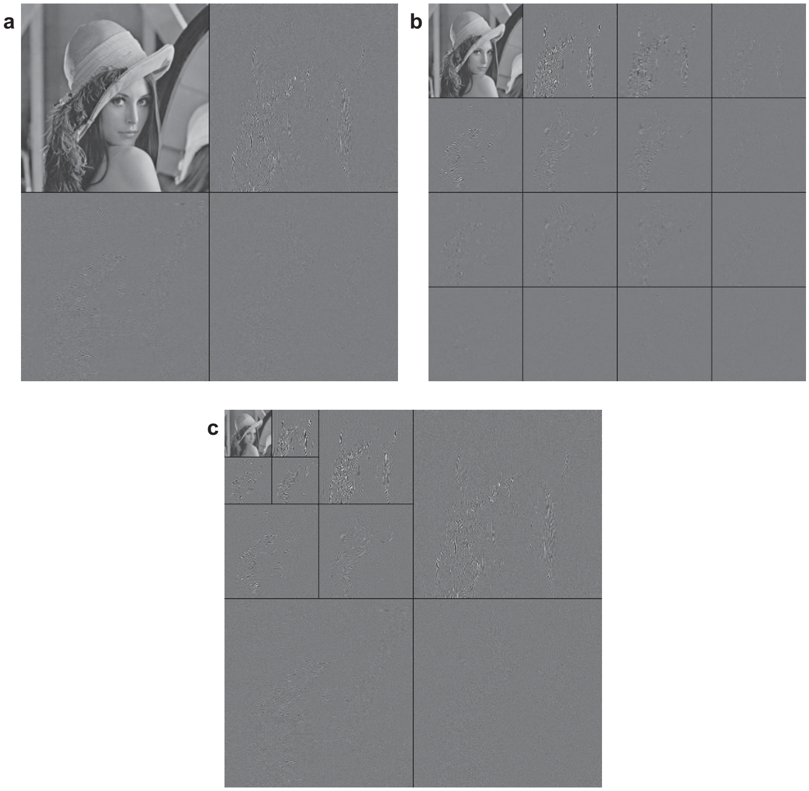

Fig. 2.63. Decomposition of an image into subband pictures (amplified by factor 4, except c_{00}) **a** relating to Fig. 2.62a **b** relating to Fig. 2.62b **c** relating to Fig. 2.62d

Fig. 2.62a depicts the related layout of subbands in the 2D frequency domain. This basic 4-band decomposition structure of Fig. 2.61 can then again be applied iteratively to respective (sub-sampled) outputs of the previous stage. For a case where all 4 subbands are equally decomposed in the next level, Fig. 2.62b shows a layout example with 16 bands. Fig. 2.62c is an example where a wavelet-style octave-band decomposition is applied fully separable over both dimensions, which is equivalent with the Haar transform scheme in Fig. 2.41c. Fig 2.62d shows the layout which is commonly denoted as *2D DWT*, where only the low-pass output c_{00} of the 4-band system is subject to further 4-band decomposition etc. In Fig. 2.62c/d, 'S' denotes the scaling band of lowest resolution, which represents a sub-sampled version of the picture.

Fig. 2.63 shows results of subband and wavelet decomposition applied to an image signal, where the different sub-sampled subband pictures are shown in the positions of their corresponding frequency partitions in Fig. 2.62.

It is also possible to realize non-separable 2D filter banks. Fig. 2.64 shows an example of a 2D decimation by a factor of 2, where a subband system decomposes a rectangular-grid (separable) sampled signal into two components of quincunx sampling. To describe such systems, the principles introduced in the context of multi-dimensional sampling can be used. If $s_k(\tilde{n}_1,\tilde{n}_2)$ is the original signal and $s_{k-1}(n_1,n_2)$ the sub-sampled signal, the relationship between the indices can be expressed by the sampling matrix **U** such that[89]

$$\begin{bmatrix}\tilde{n}_1\\ \tilde{n}_2\end{bmatrix}=\begin{bmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{bmatrix}\begin{bmatrix}n_1\\ n_2\end{bmatrix} \quad ; \quad \tilde{\mathbf{n}}=\mathbf{U}\mathbf{n}\,; \quad \mathbf{n}=\mathbf{U}^{-1}\tilde{\mathbf{n}}\in\mathbb{Z}\,. \tag{2.360}$$

The factor of sub-sampling, and hence the number of spectral copies (original plus alias spectra) is equal to the absolute determinant

$$|\mathbf{U}|=|u_{11}u_{22}-u_{21}u_{12}|\,. \tag{2.361}$$

The related frequency sampling matrix $\mathbf{F}=[\mathbf{U}^{-1}]^{\mathrm{T}}$ points to the positions of periodic spectral copies, where alias may occur. In analogy with (2.297), the *z* transform of the decimated signal is

$$S_{k-1}(z_1,z_2)=\frac{1}{|\mathbf{U}|}\sum_{k_1=0}^{U_1-1}\sum_{k_2=0}^{U_2-1}S\left(\mathrm{W}^{-(f_{11}k_1+f_{12}k_2)}\,z_1^{f_{11}}z_2^{f_{12}},\mathrm{W}^{-(f_{21}k_1+f_{22}k_2)}\,z_1^{f_{21}}z_2^{f_{22}}\right)$$
$$\text{with}\quad \mathrm{W}=\mathrm{e}^{\mathrm{j}2\pi}\quad\text{and}\quad U_i=\max_{j=1,2}\{|u_{ij}|\}. \tag{2.362}$$

The reverse operation is an interpolation by factor *U**, which is a generalization of (2.299) using the parameters in **U**

$$S_k(z_1,z_2)=S_{k-1}(z_1^{u_{11}}z_2^{u_{21}},z_1^{u_{12}}z_2^{u_{22}})\,. \tag{2.363}$$

[89] The following considerations are strictly valid for integer subsampling factors u_{ij}, as otherwise an additional sub-sample phase shift would be necessary which would require an additional interpolation step.

Example. The sampling matrix $\mathbf{T}_\mathrm{q}$ for the case of quincunx decimation and the related frequency sampling matrix $\mathbf{F}_\mathrm{q}$ in analogy with (2.63) and Fig. 2.64 are expressed as

$$\mathbf{U}_\mathrm{q}=\begin{bmatrix}2 & 1\\ 0 & 1\end{bmatrix} \quad ; \quad \mathbf{F}_\mathrm{q}=\left[\mathbf{U}_\mathrm{q}^{-1}\right]^\mathrm{T}=\begin{bmatrix}\tfrac{1}{2} & 0\\ -\tfrac{1}{2} & 1\end{bmatrix}. \tag{2.364}$$

The z transform of the decimated signal is

$$S_{k-1}(z_1,z_2)=\frac{1}{2}\sum_{k=0}^{1}S_k\left(\mathrm{W}^{-\frac{1}{2}k}z_1^{\frac{1}{2}},\mathrm{W}^{\frac{1}{2}k}z_1^{-\frac{1}{2}}z_2\right) \text{with} \quad \mathrm{W}=\mathrm{e}^{\mathrm{j}2\pi}. \tag{2.365}$$

To realize a non-separable decimation, it is typically necessary to use non-separable filters. The quincunx decimation can be performed using the following biorthogonal pair of 2D filter matrices [Kovacevic, Vetterli 1992], where according to the conditions of biorthogonal filters, the highpass $\mathbf{H}_1$ is operated with a one-sample delay either horizontally or vertically relative to the lowpass $\mathbf{H}_0$.

$$\mathbf{H}_0=\frac{1}{32}\begin{bmatrix}0 & 0 & -1 & 0 & 0\\ 0 & -2 & 4 & -2 & 0\\ -1 & 4 & 28 & 4 & -1\\ 0 & -2 & 4 & -2 & 0\\ 0 & 0 & -1 & 0 & 0\end{bmatrix} \quad ; \quad \mathbf{H}_1=\frac{1}{4}\begin{bmatrix}0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\end{bmatrix}. \tag{2.366}$$

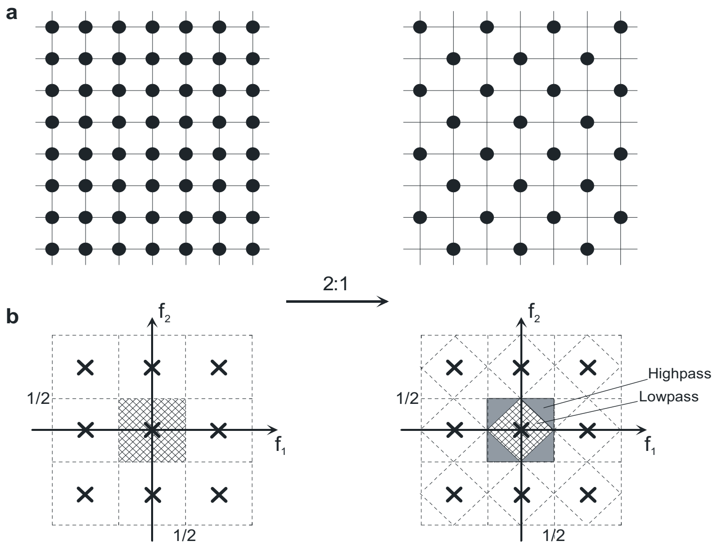

Fig. 2.64. Non-separable 2D system with 2:1 quincunx decimation
a Sub-sampling schema in the spatial domain **b** Layout of frequency bands

As in (2.325), the kernels of lowpass and highpass filters are of different size. Applying the relationships $G_0(\mathbf{z})=H_1(-\mathbf{z})$ and $G_1(\mathbf{z})=-H_0(-\mathbf{z})$ from (2.316), the synthesis filters are determined by multiplication (modulation) with alternating signs. For symmetric 2D filters, this is realized such that impulse response values with an odd sum of indices are multiplied by −1, i.e.

$$\begin{aligned} g_0(n_1,n_2) &= (-1)^{n_1+n_2} h_1(n_1,n_2), \\ g_1(n_1,n_2) &= (-1)^{n_1+n_2+1} h_0(n_1,n_2). \end{aligned} \tag{2.367}$$

The resulting synthesis filter matrices are

$$\mathbf{G}_0 = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad ; \quad \mathbf{G}_1 = \frac{1}{32}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 1 & 4 & -28 & 4 & 1 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{2.368}$$

2.8.6 Pyramid decomposition

The DWT is a *multi-resolution scheme* for signal representation. This means that by using more higher-frequency wavelet bands, the resolution of the reconstructed signal is increased; in a critically sampled (typically dyadic) wavelet representation, the total number of coefficient samples equals the number of samples in the full-resolution signal, regardless of the depth of the wavelet tree. An alternative type of multi-resolution methods are the *pyramid schemes*. U signal representations with different sampling resolutions are generated by filtering and downsampling, *in addition* to the original (full) resolution. In principle, arbitrary downsampling factors are possible, even though out of complexity reasons and to avoid excessive over-completeness, dyadic factors are often chosen when pyramid schemes are used in compression, unless different up/downsampling is needed e.g. to support multiple spatial resolutions.

Whereas the scheme of *Gaussian pyramid* generates the different resolution representations as independent entities (in case of dyadic resolutions this would correspond to the scale spaces in Fig. 2.57), the *Laplacian pyramid* establishes a differential representation, which can be interpreted as a set of bandpass channels (this would roughly correspond to the wavelet spaces in Fig. 2.57). However, in contrast to the DWT approach, no downsampling is applied to the bandpass components, which means that the representation (in terms of number of samples) is over-complete, as the spectrum below the respective pass band should be approximately void. On the other hand, alias and frequency reversion are avoided by omitting the downsampling, which can be beneficial in terms of coding, e.g. when shift invariance is required as in motion compensated prediction.

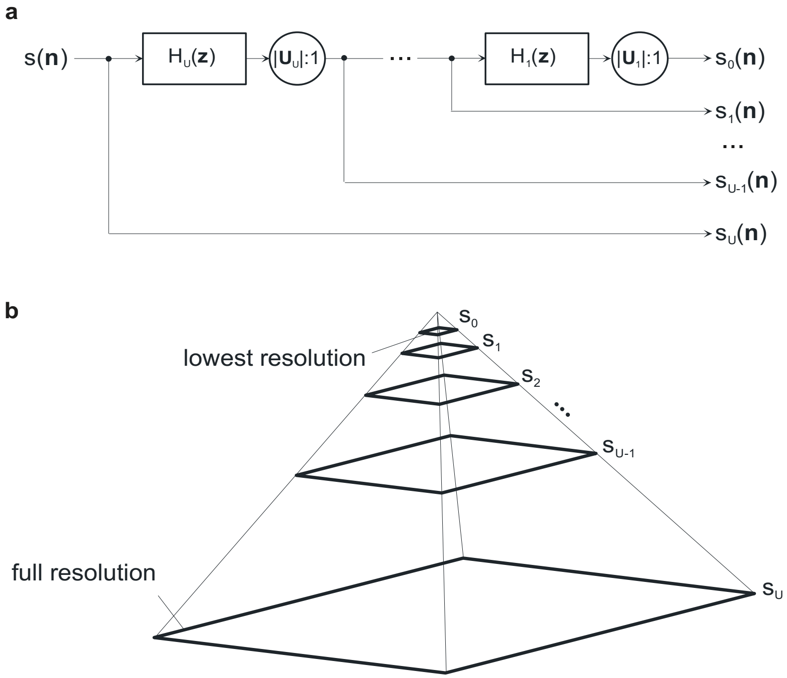

Fig. 2.65. **a** Generation of the Gaussian pyramid representation (typically identical filters and identical subsampling schemes described by **U** are used throughout the levels). **b** Illustration of images sizes (dyadic scheme) as levels of a pyramid

Gaussian pyramid. All resolution levels can be used independently, i.e. no lower resolution level is needed if a finer resolution level shall be used. The generation of the Gaussian pyramid representation is performed by elementary building blocks consisting of lowpass filtering followed by decimation described by a sampling matrix **U** (see Fig. 2.65a; as an example, for a 2D signal with horizontal/vertical subsampling factors $U_1=U_2=2$, the total subsampling ratio is 4:1 with $|\mathbf{U}|=4$). This is performed in an iterative cascade through all levels of the pyramid, starting from the base and terminating at the top (see Fig. 2.65b). By cascading U elementary building blocks, a total of $U+1$ resolution levels (including the original resolution $s(\mathbf{n})=s_U(\mathbf{n})$) are generated. The signal $s_{k-1}(\mathbf{U}^{-1}\mathbf{n})$ is obtained by lowpass filtering and subsampling, $s_{k-1}(\mathbf{U}^{-1}\mathbf{n})=s_k(\mathbf{n})*h(\mathbf{n})$[90]. The concept is similar to the processing of scale-space components in the wavelet transform (Fig. 2.57), but exhibits redundancy due to the fact that the coarser resolutions establish subspaces

[90] This convolution is to be performed with reference to **n** coordinates, but only at positions where $\mathbf{U}^{-1}\mathbf{n}$ consists of integer numbers (i.e. the positions still existing after subsampling). In case of non-dyadic subsampling, **U** itself could contain non-integer numbers. In that case, it would be necessary to include a position-dependent sub-sample phase shift (interpolation) in the lowpass filter impulse response.

of the finer resolutions instead of being orthogonal complements. The resulting representation is therefore significantly redundant and over-complete, and not as such very suitable for the purpose of compression.

The method is denoted as *Gaussian* pyramid, because filters approximating a Gaussian-shaped impulse response are often used as lowpass filters prior to decimation in this context. The convolution of two Gaussian functions results in a Gaussian of extended length. Hence, the effect of the cascaded system at a later stage is approximately equivalent to the usage of *one* Gaussian filter with a longer width of the impulse response (lowpass with lower cut-off frequency)[91]. The implementation complexity in the cascaded pyramid system is however much lower due to the intermediate sub-sampling operations.

An example for simple approximation of a non-separable 2D Gaussian with a short kernel is given by the filter matrix[92]

$$\mathbf{H}_G = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{2.369}$$

Laplacian pyramid. Each resolution level (except for the smallest scale image) is represented by a difference signal. The principle as applied for generation of the difference signals is shown in Fig. 2.66a. Firstly, the lower-resolution signal $s_{k-1}(\mathbf{n})$ is generated as in the case of the Gaussian pyramid. Then, it is upsampled and filtered by a lowpass interpolation filter to generate a prediction and compute the difference (prediction error) signal[93],

$$\hat{s}_k(\mathbf{n}) = s_{k-1}(\mathbf{U}^{-1}\mathbf{n} \uparrow \mathbf{n}) * g(\mathbf{n}), \quad e_k(\mathbf{n}) = s_k(\mathbf{n}) - \hat{s}_k(\mathbf{n}). \tag{2.370}$$

For reconstruction, the difference is added to the prediction from the next-coarser signal (Fig. 2.66b). If U elementary building blocks are arranged in a cascaded structure, a total of U+1 resolution levels is represented by U difference signals $e_1(\mathbf{n})\dots e_U(\mathbf{n})$ and one strongly-scaled signal $s_0(\mathbf{n})$. The reconstructed signals

[91] This statement ignores the alias which can occur due to the subsampling, depending on the spectrum of the signal. On the other hand, the Gaussian function is non-negative, which gives a penalty in terms of the sharpness of the frequency cut-off, but prevents from ringing at signal discontinuities (e.g. edges in images).

[92] A typical primitive 1D approximation of a Gaussian filter function is the binomial filter with the coefficient vector $\mathbf{h}=[\,¼\;\;½\;\;¼\,]^{\mathrm{T}}$. (2.369) represents a superposition of a horizontal and a vertical binomial filter. Iterated convolution gives longer binomial functions, which by tendency give an approximation of a sampled Gaussian according to the central limit theorem. For 2D, typically separable filters are used.

[93] This convolution is performed at all positions where $\mathbf{n}$ is integer, where $\mathbf{U}^{-1}\mathbf{n}\uparrow\mathbf{n}$ expresses that in the upsampled s_{k-1} zero values are inserted where $\mathbf{U}^{-1}\mathbf{n}$ is not an integer number. In case of non-dyadic subsampling, $\mathbf{U}$ may itself contain non-integer numbers and it may be necessary to additionally include a position-dependent sub-sample phase shift in the filter impulse response.

$s_1(\mathbf{n})\dots s_U(\mathbf{n})$ at the different pyramid levels are equivalent to the output of the Gaussian pyramid. Reconstruction always must start at the lowest resolution level and requires U sequential operations.

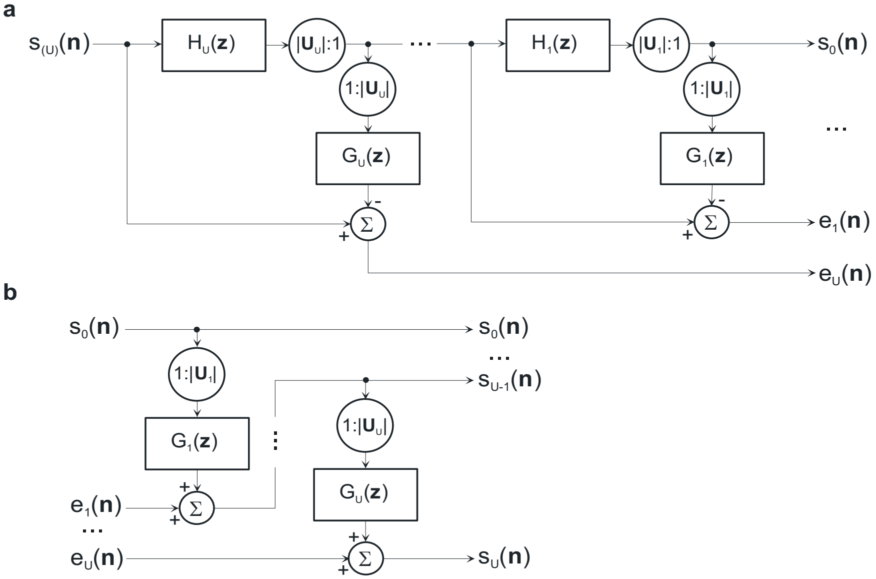

Fig. 2.66. Laplacian pyramid representation: **a** analysis **b** synthesis

Assuming (almost) alias-free subsampling and high-quality interpolation, the difference between the signal $s_k(\mathbf{n})$ and the output of the filter (2.369) would be close to the prediction $\hat{s}_k(\mathbf{n})$ from (2.370), which could then be generated directly by the filter

$$\mathbf{H}_{\mathrm{L}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \mathbf{H}_{\mathrm{G}} = \frac{1}{8}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{2.371}$$

This filter kernel provides an approximation of the local second derivative of the signal and is denoted as *Laplacian filter operator*. From this, the differential pyramid is also called *Laplacian* pyramid [Burt, Adelson 1983]. In principle, this pyramid represents second derivatives of the signal within different scale spaces, which could also be interpreted as bandpass-filtered (or highpass-filtered for $e_U(\mathbf{n})$) versions of $s(\mathbf{n})$[94].

[94] When subsampling is omitted and filtering is done by concatenating Gaussian impulses responses, the signals $e_k(n)$ are also entitled as *differences of Gaussians* (DoG), which is approximately equal to filtering by 2nd derivatives of Gaussians (Laplacian of Gaussian, LoG). Such representations are largely over-complete in terms of number of samples and

Unlike the wavelet transform, the differential signals $e_k(\mathbf{n})$ of the Laplacian pyramid are not orthogonal complements. Firstly, $e_k(\mathbf{n})$ and $s_k(\mathbf{n})$ must be correlated, as the prediction error contains all detail information that is not predictable from $s_{k-1}(\mathbf{n})$. Second, due to the usage of non-ideal filters the prediction errors $e_k(\mathbf{n})$ and $e_{k-1}(\mathbf{n})$ over the different levels may be correlated as well; furthermore, structures with wide spectra such as edges and pulses would also appear in the prediction errors over a variety of k values. In general however, this redundancy will be significantly lower than in the case of the Gaussian pyramid. Furthermore, an over-completeness is inherent to the pyramid schemes in terms of number of samples. For example, if U pyramid levels are used for a 2D (image) signal, the total number of samples to be represented grows by a factor of

$$\sum_{u=0}^{U}\left(\frac{1}{4}\right)^{u} < \frac{4}{3}, \tag{2.372}$$

as compared to the number of samples in the original signal. In contrast to that, block and block-overlapping transforms, filterbank and DWT transforms can use critical sampling, such that the overall number of frequency coefficients is identical to the number of signal samples. However, block and wavelet transform, though not over-complete, need to make trade-offs between lowpass and highpass filters in order to achieve perfect reconstruction. This can invoke other effects (in particular aliasing in bands and additionally frequency reversion in highpass bands), which may even be more severe than the disadvantage of increased number of samples. When using a pyramid representation in the context of encoding, the over-completeness seems to be a disadvantage in first place, but the aforementioned alias-bearing effects of critically sampled representations can be avoided. Furthermore, the redundancy between various components in the pyramid can be utilized and removed by coding. Therefore, the differential pyramid has turned out to be efficient as a compression method, particularly in the context of scalable (multi-resolution) representations of image and video signals, where additional methods such as prediction over another dimension can remove the redundancy. Nevertheless, it should be noted that the over-completeness causes a penalty in terms of larger complexity, as more samples need to be processed.

are not usually used in coding, but rather when using multi-resolution representations for feature analysis (see [MCA, SEC. 4.4])

2.9 Problems

Problem 2.1.

a) Determine a condition for alias-free hexagonal sampling (Fig. 2.11c).
b) What is the ratio of the area of the base band, as compared to rectangular sampling with horizontal and vertical sampling distances equal to the vertical distance in the hexagonal case?
c) What is the ratio of the horizontal sampling distance in hexagonal sampling, as compared to the rectangular sampling case of b) ?
d) Compute the determinant of the sampling matrix $\mathbf{T}_{\text{hex}}$, normalized by the vertical sampling distance. Discuss the relationship of this value with the results from parts b) and c).

Problem 2.2.

a) Show that the quincunx grid (Fig. 2.11d) can be constructed by superposition of two rectangular grids which are offset by $T_1|T_2$, and each having sampling distances $2T_1|2T_2$ horizontally | vertically.
b) Compute the periodic spectrum from this construction, and show that it is identical to the spectrum found via the sampling matrix (2.63).

Problem 2.3.

A two-dimensional cosine (2.1) of horizontal frequency $F_1=1/(3T)$ is sampled by a quincunx grid.
a) Compute the 2D Fourier spectrum.
b) Determine the upper limit for vertical frequency $|F_2|$ guaranteeing alias free sampling.
c) Which horizontal frequency becomes visible after ideal lowpass filter reconstruction from the sampled signal, if the vertical frequency is $F_2=1/(3T)$?

Problem 2.4.

For the generalized Gaussian PDF (2.126),
a) Show that $\gamma=2$ gives the Gaussian normal PDF (2.127).
b) Show that $\gamma=1$ gives the Laplacian PDF (2.128).
c) With $\Gamma(c)=\Gamma(c+1)/c$, which PDF can asymptotically be expected for $\gamma\rightarrow\infty$?
[use values $\Gamma(3)=2$; $\Gamma(1)=1$; $\Gamma(1.5)=\sqrt{\pi}/2$; $\Gamma(0.5)=\sqrt{\pi}$].

Problem 2.5.

A one-dimensional, stationary zero-mean process $s(n)$ with Gaussian PDF has an autocovariance function $\mu_{ss}(k)=\sigma_s^2\rho^{|k|}$.
a) Construct the autocovariance matrix $\mathbf{C}_{ss}$ of size 3x3 .
b) Show that for $\rho=0$: $p_3(\mathbf{x})=p_{\mathcal{N}}(x_1)p_{\mathcal{N}}(x_2)p_{\mathcal{N}}(x_3)$. Here, $p_3(\mathbf{x})$ is a vector Gaussian PDF (2.156) for vector random variables $\mathbf{x}=[x_1\, x_2\, x_3]^{\mathrm{T}}$, and $p_{\mathcal{N}}(x_i)$ shall be Gaussian normal distributions (2.127).

Problem 2.6.

a) Combined random instantiations from two event sets S_1 and S_2 shall be statistically independent, i.e. $\Pr(S_1,S_2)=\Pr(S_1)\Pr(S_2)$. Show the following relationships for this case: $H(S_1|S_2)=H(S_1)$; $H(S_2|S_1)=H(S_2)$; $I(S_1;S_2)=0$.
b) Instantiations drawn from the two event sets S_1 and S_2 shall now always be identical. Show that $H(S_1|S_2)=0$; $H(S_2|S_1)=0$; $I(S_1;S_2)=H(S_1)=H(S_2)$.

Problem 2.7.

The joint PDF of two Gaussian processes $s_1(n)$ and $s_2(n)$ shall be defined by (2.153). Further, $\sigma_{s_1}=\sqrt{2}\sigma_{s_2}$.

a) Determine the joint PDF for the cases of uncorrelated signals ($\rho_{s_1s_2}(\mathbf{0})=0$) and fully dependent signals ($\rho_{s_1s_2}(\mathbf{0})=1$).
b) Determine the conditional PDF $p_{s_2s_1}(x_2|x_1;\mathbf{0})$ for the general case first, then specifically for the two cases of a).

Problem 2.8.

The eigenvalues of the matrix $\mathbf{C}=\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}$ are $\lambda_1=1+\rho$ and $\lambda_2=1-\rho$.

a) Following (A.20), determine $\mathbf{\Phi}_1$ und $\mathbf{\Phi}_2$ of $\mathbf{C}$ such that they establish an orthonormal base $\mathbf{\Psi}=[\ \mathbf{\Phi}_1\ \mathbf{\Phi}_2\ ]$ according to (A.24).
b) Sketch the eigenvectors within a coordinate system of axes x_1, x_2.
c) Determine the inverse $\mathbf{\Psi}^{-1}$.
d) Compute the determinant of $\mathbf{C}$, and compare the result against the product of the eigenvalues.

Problem 2.9.

For the AR(1) model from (2.189) and (2.190), prove the validity of the autocorrelation and variance properties (2.191) and of the spectral properties (2.192).

Problem 2.10.

For statistical modeling of a 1D signal, an AR(1) model of variance σ_s^2 and correlation coefficient $\rho_{ss}(1)=0.95$ is used. By using the autocorrelation function of the model, a linear predictor shall be optimized. Determine the coefficients $a(1)$ and $a(2)$ for a predictor filter of order $P=2$ by solving the Wiener-Hopf equation (2.207).

Problem 2.11.

For linear 2D prediction of a separable AR(1) process with $\rho_1=\rho_2=0.95$, a non-separable predictor filter is used which implements $\hat{s}(n_1,n_2)=0.5\,s(n_1-1,n_2)+0.5\,s(n_1,n_2-1)$ as prediction equation. Determine the variance of the prediction error signal, and compare against the variance of the innovation signal $v(n_1,n_2)$.

Problem 2.12.

A video sequence consists of pictures which are unchanged except for global translation motion. The picture information is an output from a 2D separable AR(1) model generator. Parameters are $\rho_1=\rho_2=0.95$. From one picture to the next, translation shift by $k_1=7$ horizontally and $k_2=3$ is observed. Different methods of linear prediction shall be compared using the criterion of prediction error variance:

a) Spatial prediction, separable predictor according to (2.226);

b) Temporal prediction $\hat{s}(n_1,n_2,n_3)=s(n_1,n_2,n_3-2)$;

c) Motion-compensated temporal prediction $\hat{s}(n_1,n_2,n_3)=s(n_1-k_1,n_2-k_2,n_3-1)$.

Problem 2.13.

When defined as a block transform, the Haar transform has $U^*=\log_2 M+1$ 'basis types', which are subsequently described by an index $u^*=0,1,...,\log_2 M$. From each basis type, M^* basis functions are developed, each of which has only one internal flip of the sign for the cases $u^*>0$. M^* is 1 for $u^*=0,1$ and 2^{u^*-1} for the other basis types. Basis functions determined from the same basis type are indexed by $i=0,1,..., M^*-1$ subsequently, they are non-overlapping. The orthonormal transform basis set is

$$t_k^{\text{Haar}}(n)=\begin{cases}\text{ha}(n-i\frac{M}{M^*}) \text{ for } i\frac{M}{M^*}\leq n<\left(i+1\right)\frac{M}{M^*}\\ \qquad\qquad 0, \text{ else}\end{cases}$$

with

$$k=\begin{cases}k^* & \text{for} \quad k^*=0,1\\ M^*+i & \text{for} \quad k^*>1\end{cases} \qquad \text{and} \qquad \text{ha}(n)=\sqrt{\frac{M^*}{M}}\cdot(-1)^{\left\lfloor\frac{2^{k^*}n}{M}\right\rfloor} .$$

The *Walsh basis* consists of $K=M$ basis functions with constant length M. The kth function has k flips between positive and negative values. The function for $k=0$ consists of M positive constant values. The development of remaining Walsh functions is performed recursively, starting from $t_1(n)$. The number of recursions necessary to generate all basis functions is $\log_2 M-1$. During one recursion step, all basis functions developed in the previous step are scaled (which is done by eliminating each second sample), and then combined into new basis functions, once periodically and once 'antiperiodic', i.e. mirrored. The number of new basis functions is doubled by each iteration step, and with ν flips in the scaled function, two new functions can be generated, one of which has $2\nu-1$ sign flips, the other 2ν flips. The process of recursion can be described as follows, where the periodic/antiperiodic combinations are implemented by multiplying the scaled functions by ±1:

Let

$$t_k^{\text{Wal}}(n)=t_k^{\text{Rad}}(n) \qquad \text{for} \qquad 0\leq k<2, \quad 0\leq n<M$$

$k^*=1$, $M^*=2$, $K^*=\log_2 M$, $P(0)=-1$.

While $k^*<K^*$

{

For $0 \le i < \log_2 M^*$:

$$t_{\mathrm{scal}}(n,i) = t^{\mathrm{Wal}}_{(M^*+2i)/2}(2n,i)$$

$$t^{\mathrm{Wal}}_{M^*+2i+j}(n) = \begin{cases} t_{\mathrm{scal}}(n,i) & \text{for} \quad 0 \le n < M/2 \\ P(i)^{j+1} \cdot t_{\mathrm{scal}}(n - M/2, i) & \text{for} \quad n \ge M/2 \end{cases} \qquad \text{for} \quad j = 0,1$$

For the next step, set $P(2i+j) = -P(i)^{j+1}$, $M^* = 2M^*$, $k^* = k^* + 1$.

} (2.373)

The factor $P(i)$ has the effect that from a periodic basis function (having even number of flips), an antiperiodic function is generated at first in the next step and vice versa. This guarantees an ever increasing number of flips, equal to the index of the function. The recursion process is illustrated for the case $M = 8$ in Fig. 2.67

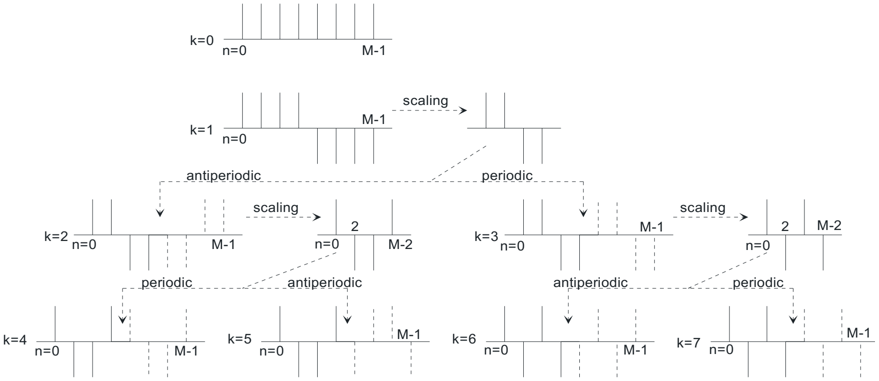

Fig. 2.67. Development of the Walsh transform basis

a) Construct the transform matrices of 1D Haar and Walsh transforms for M=4.

b) Transform the following image matrix by the related separable 2D transforms using (2.246):

$$\mathbf{S} = \begin{bmatrix} 18 & 4 & 2 & 4 \\ 18 & 4 & 2 & 4 \\ 2 & 4 & 2 & 4 \\ 2 & 4 & 2 & 4 \end{bmatrix}.$$

c) Interpret the results. Discuss in particular which transform better compacts the given image.

Problem 2.14.

a) Determine the transform matrix of a 1D DCT for M=3, and show that the transform is orthonormal.

b) Set up the autocorrelation matrix (2.157) of size 3x3 for an AR(1) model. The matrix shall then be transformed by the 1D DCT of a), using (2.287).

c) For a model of variance σ_s^2, two different cases ρ=0.9 and ρ=0.5 shall be considered to fill the matrix $\mathbf{C}_{cc}$. Give an interpretation of the differences you observe in the matrix entries.

d) For both cases from c), compute the trace (A.21) of the autocorrelation matrices and their transformed counterparts. Give an interpretation of the result.

Problem 2.15.

a) Determine the transform matrix of a 1D Haar transform for M=4.

b) Set up the autocorrelation matrix (2.157) of size 4x4 for an AR(1) model, and apply its transformation by the Haar transform of a), using (2.287).

c) Give an interpretation about the remaining correlations between transform coefficients.

Problem 2.16.

a) Determine the Fourier transfer functions $\mathcal{F}\{\mathbf{t}_k\}$ for the basis vectors of a transform

$$\mathbf{t}_0 = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}^{\mathrm{T}}, \qquad \mathbf{t}_1 = \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}^{\mathrm{T}}.$$

b) Prove the orthogonality of this basis system.

c) Show that the functions | $\mathcal{F}\{\mathbf{t}_k\}$ | of both basis vectors have a mirror symmetry around the frequency f= 1/4.

d) Show that $|\,\mathcal{F}\,\{\mathbf{t}_0\}\,|^2 + |\,\mathcal{F}\,\{\mathbf{t}_1\}\,|^2$ = const.

Problem 2.17.

a) Determine the basis vectors of the block-overlapping transform according to (2.289)-(2.292) with settings U=2, M=4. [Hint: To simplify expressions of the trigonometric functions, use constants cos(3π/8)=sin(π/8)=A and cos(π/8)=sin(3π/8)=B; consider for which other cases identical values ±A or ±B would appear.]

b) Show the orthogonality of the basis system.

c) Determine the Fourier transfer functions $\mathcal{F}\{\mathbf{t}_k\}$. Do the basis functions have a linear-phase property?

d) Would a realization of this transform by a fast algorithm be possible?

Problem 2.18.

a) Show the validity of the orthogonality property for linear-phase QMF systems for the following cases of filters:

$$i)\ H_0(z) = A\cdot z^2 + B\cdot z + C + C\cdot z^{-1} + B\cdot z^{-2} + A\cdot z^{-3}$$

$$ii)\ H_0(z) = A\cdot z^2 + B\cdot z + C + B\cdot z^{-1} + A\cdot z^{-2}$$

b) Determine the z transform representations of polyphase filters $H_{0,\mathrm{A}}$, $H_{0,\mathrm{B}}$, $H_{1,\mathrm{A}}$ and $H_{1,\mathrm{B}}$ according to Fig. 2.53 for both filter configurations from a). Which number of multiplications per sample would be necessary at minimum?

Problem 2.19.

A zero-mean random signal $s(n)$ shall be modeled by an AR(1) process. To describe the parameters of the process, the value of spectral power density $\Phi_{ss,\delta}(f=1/2)=\sigma_s^2/9$ is given.

a) Determine the correlation parameter ρ, and the variance σ_v^2 of the white-noise innovation in dependency of σ_s^2.

The signal shall be decomposed into two polyphase components, where the sequences of even- and odd-indexed samples shall be processed independently by predictor filters of first order, $H(z)=az^{-1}$, as shown in Fig. 2.68.

b) Determine the optimum predictor coefficient a.

c) Determine the variance of the prediction error signal $e_e(n)$ of the even-indexed samples when the optimum a is used. From this, compute the coding gain $G=\sigma_s^2/\sigma_{e_e}^2$. By which factor will this gain be smaller, as compared to the optimum case of prediction for the AR(1) model (without polyphase decomposition) ?

d) Compute the covariance between the signals $e_e(n)$ und $e_o(n)$ when the optimum a is used.

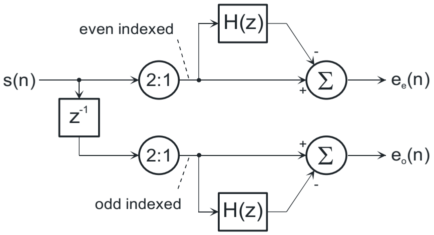

Fig. 2.68. Prediction within polyphase components

Problem 2.20.

An AR(1) process $s_{\text{AR}}(n)$ is characterized by the correlation parameter ρ=0.75 and the variance of the Gaussian innovation signal, σ_v^2=7.

a) Determine the variance of the AR process.

For linear prediction of $s_{\text{AR}}(n)$, a falsely adapted prediction error filter with transfer function $A(z) = 1 - z^{-1}$ is used (see Fig. 2.69).

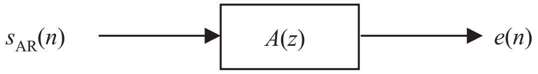

Fig. 2.69. Prediction of an AR(1) process

b) Compute the variance of the prediction error signal $e(n)$ and the coding gain.

c) Which would be the coding gain in case of optimum prediction? By which factor is the coding gain worse in the case of the falsely-adapted prediction from b)?

d) Determine the power density $\Phi_{ee,\delta}(f)$. Can $e(n)$ be a white noise process?

e) Determine a system (z transfer function and block diagram), which generates the optimum prediction error signal of lowest possible variance from $e(n)$.

Problem 2.21.

A binary signal shall be synthesized according to the block diagram given in Fig. 2.70. Herein, $v(n)$ is an uncorrelated zero-mean Gaussian process of variance σ_v^2. **T** is a threshold decision circuit with following characteristics:

$$b(n) = \begin{cases} 0 & \text{if } s(n) \le C \\ 1 & \text{if } s(n) > C \end{cases}$$

It is now to be assumed that $b(n)$ behaves as first order Markov process.

a) Determine the probabilities Pr(0) and Pr(1) depending on C.

b) For C=0, determine the probabilities Pr(1|0) and Pr(0|1) depending on ρ.

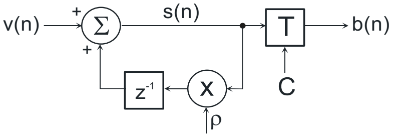

Fig. 2.70. Circuit for generating a binary signal

http://www.springer.com/978-3-662-46690-2

Multimedia Signal Coding and Transmission
Ohm, J.
2015, XI, 558 p. 232 illus., Hardcover
ISBN: 978-3-662-46690-2