5 Multiscale Representation

5.1 Scale

5.1.1 Introduction

The neighborhood operations discussed in Chapter 4 can only be the starting point for image analysis. This class of operators can only extract local features at scales of at most a few pixels distance. It is obvious that images contain information also at larger scales. To extract object features at these larger scales, we need correspondingly larger filter masks. The use of large masks, however, results in a significant increase in computational costs. If we use a mask of size R^W in a W-dimensional image the number of operations is proportional to R^W. Thus a doubling of the scale leads to a four- and eight-fold increase in the number of operations in 2- and 3-dimensional images, respectively. For a ten times larger scale, the number of computations increases by a factor of 100 and 1000 for 2- and 3-dimensional images, respectively.

The explosion in computational cost is only the superficial expression of a problem with deeper roots. We illustrate it with a simple task, the detection of edges and lines at different resolutions. To this end, we use the same image row but blur it to different degrees (Fig. 5.1). We define the corresponding scale as the distance over which the image has been blurred and analyze the gray value differences over this distance.

We first investigate gray value differences at high resolution, a scale of just one pixel distance (Fig. 5.1a, b). At this fine scale, the change in gray values is dominated by the noisy background of the image. Any detection of gray value changes caused by the contrast between objects and background is inaccurate and erroneous. The problem is caused by a *scale mismatch*: the gray values only vary on larger scales than the operators used to detect them.

If we take instead a low resolution (Fig. 5.1e, f), the lines are blurred so much that the contrast has decreased significantlyd. Moreover, two closely spaced lines in the left part of the signal have merged into one object at this coarse resolution. Therefore the detection of edges and lines is suboptimal again. At a resolution comparable to the line width, however, the line detection seems to be optimal (Fig. 5.1c, d). Noise is significantly reduced compared to the finest scale (Fig. 5.1a) but the

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

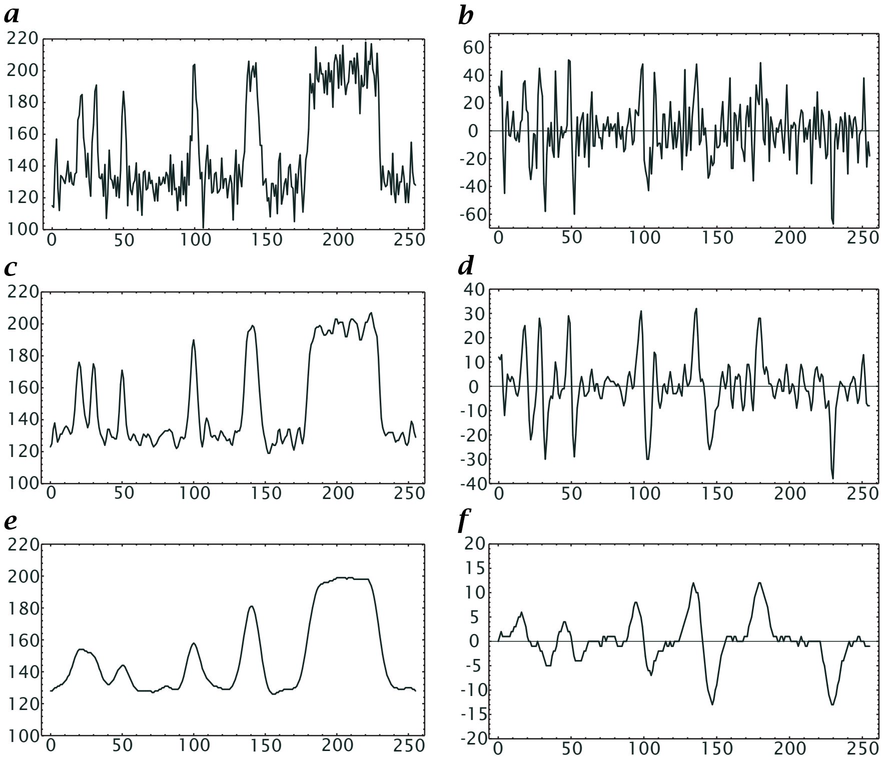

***Figure 5.1:** Lines and edges at **a** high, **c** medium, and **e** low resolution. **b**, **d**, and **f** Subtraction of neighboring pixels for edge detection for **a**, **c**, and **e**, respectively.*

contrast between the line and the background is not yet diminished as in Fig. 5.1e.

From the discussion of this example we can conclude that the detection of certain features in an image is optimal at a certain scale. This scale depends, of course, on the characteristic scales contained in the object to be detected. Optimal processing of an image thus requires the representation of an image at different scales. In order to meet this demand, we need a *multiscale representation* of images. In this chapter, we will first illuminate the relation between the spatial and wave number representation of images under this perspective (Section 5.1.2). Then we will turn to efficient *multigrid representations* such as the *Gaussian pyramid* (Section 5.2.2) and the *Laplacian pyramid* (Section 5.2.3). Finally, the *scale space* is introduced in Section 5.3 as an concept with a continuous scale parameter. We discuss how a diffusion process can generate it and describe its basic properties.

5.1.2 Spatial Versus Wave Number Representation

In Chapter 2 we discussed in detail the representation of images in the spatial and wave number domain. In this section we will revisit both representations under the perspective of how to generate a multiscale representation of an image.

If we represent an image on a grid in the spatial domain, we do not have any information at all about the wave numbers contained at that point in the image. We know the position with an accuracy of the grid constant Δx, but the local wave number at this position may be anywhere in the range of the possible wave numbers from 0 to $M\Delta k = 2\pi M/\Delta x$.

In the wave number representation, we have the reverse case. Each pixel in this domain represents one wave number with the highest wave number resolution possible for the given image size. But any positional information is lost, as one point in the wave number space represents a periodic structure that is spread over the whole image.

The above discussion shows that the representation of an image in either the spatial or wave number domain constitute two opposite extremes. We can optimize either the spatial or the wave number resolution but the resolution in the other domain is completely lost. What we need for a multiscale image representation is a type of joint resolution that allows for a separation into different wave number ranges (scales) but still preserves as much spatial resolution as possible.

5.1.3 Windowed Fourier Transform

One way to approach a joint space-wave number representation is the *windowed Fourier transform*. As the name says, the Fourier transform is not applied to the whole image but only to a section of the image that is formed by multiplying the image with a window function $w(\boldsymbol{x})$. The window function has a maximum at $\boldsymbol{x} = \boldsymbol{0}$ and decreases monotonically with $|\boldsymbol{x}|$ towards zero. The maximum of the window function is then put at each point $\boldsymbol{x}$ of the image to compute a windowed Fourier transform for each point:

$$\hat{g}(\boldsymbol{x}, \boldsymbol{k}_0) = \int_{-\infty}^{\infty} g(\boldsymbol{x}')w(\boldsymbol{x}' - \boldsymbol{x}) \exp\left(-2\pi i \boldsymbol{k}_0 \boldsymbol{x}'\right) dx'^2. \tag{5.1}$$

The integral in Eq. (5.1) almost looks like a convolution integral (Eq. (2.54), ≻ R4). To convert it into a convolution integral we observe that $w(-\boldsymbol{k}) = w(\boldsymbol{k})$ and rearrange the second part of Eq. (5.1):

$$\begin{aligned} & w(\boldsymbol{x}' - \boldsymbol{x}) \exp\left(-2\pi i \boldsymbol{k}_0 \boldsymbol{x}'\right) \\ = \quad & w(\boldsymbol{x} - \boldsymbol{x}') \exp\left(2\pi i \boldsymbol{k}_0 (\boldsymbol{x} - \boldsymbol{x}')\right) \exp\left(-2\pi i \boldsymbol{k}_0 \boldsymbol{x}\right). \end{aligned} \tag{5.2}$$

Then we can write Eq. (5.1) as a convolution

$$\hat{g}(\boldsymbol{x}, \boldsymbol{k}_0) = (g(\boldsymbol{x}) * w(\boldsymbol{x}) \exp(2\pi \mathrm{i} \boldsymbol{k}_0 \boldsymbol{x})) \exp(-2\pi \mathrm{i} \boldsymbol{k}_0 \boldsymbol{x}) . \qquad (5.3)$$

This means that the local Fourier transform corresponds to a convolution with the complex convolution kernel $w(\boldsymbol{x}) \exp(2\pi \mathrm{i} \boldsymbol{k}_0 \boldsymbol{x})$ except for a phase factor $\exp(-2\pi \mathrm{i} \boldsymbol{k}_0 \boldsymbol{x})$. Using the *shift theorem* (Theorem 2.3, p. 54, ≻ R4), the transfer function of the convolution kernel can be computed to be

$$w(\boldsymbol{x}) \exp(2\pi \mathrm{i} \boldsymbol{k}_0 \boldsymbol{x}) \circ\!\!-\!\!\bullet\ \hat{w}(\boldsymbol{k} - \boldsymbol{k}_0). \qquad (5.4)$$

This means that the convolution kernel $w(\boldsymbol{x}) \exp(2\pi \mathrm{i} \boldsymbol{k}_0 \boldsymbol{x})$ is a *bandpass filter* with a peak wave number of $\boldsymbol{k}_0$. The width of the bandpass is inversely proportional to the width of the window function. In this way, the spatial and wave number resolutions are interrelated to each other. As an example, we take a Gaussian window function

$$\exp\left(-\frac{x^2}{2\sigma_x^2}\right). \qquad (5.5)$$

Its Fourier transform (≻ R4, ≻ R5), is

$$\frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-2\pi^2 k^2 \sigma_x^2\right). \qquad (5.6)$$

Consequently, the product of the standard deviations in the space and wave number domain ($\sigma_k^2 = 1/(4\pi\sigma_x^2)$) is a constant: $\sigma_x^2\sigma_k^2 = 1/(4\pi)$). This fact establishes the classical *uncertainty relation* (Theorem 2.7, p. 57). It states that the product of the standard deviations of any Fourier transform pair is larger than or equal to $1/(4\pi)$. As the Gaussian window function reaches the theoretical minimum it is an optimal choice; a better wave number resolution cannot be achieved with a given spatial resolution.

5.2 Multigrid Representations

5.2.1 Introduction

If we want to process signals in different scales, this can be done in the most efficient way in a *multigrid representation*. The basic idea is simple. While the representation of fine scales requires the full resolution, coarse scales can be represented at lower resolution. This leads to a scale space with smaller and smaller images as the scale parameter increases. In the following two sections we will discuss the *Gaussian pyramid* (Section 5.2.2) and the *Laplacian pyramid* (Section 5.2.3). In this section, we only discuss the basics of multigrid representations. Optimal multigrid smoothing filters are elaborated in Section 11.5 after we got acquainted with smoothing filters.

These pyramids are examples of multigrid data structures that have been introduced into digital image processing in the early 1980s and have led to a tremendous increase in speed of many image processing algorithms in digital image processing since then.

5.2.2 Gaussian Pyramid

If we want to reduce the size of an image, we cannot just *subsample* the image by taking, for example, every second pixel in every second line. If we did so, we would disregard the *sampling theorem* (Section 9.2.3). For example, a structure which is sampled three times per wavelength in the original image would only be sampled one and a half times in the subsampled image and thus appear as an aliased pattern as we will discuss in Section 9.1. Consequently, we must ensure that all structures that are sampled less than four times per wavelength are suppressed by an appropriate smoothing filter to ensure a proper subsampled image. For the generation of the scale space, this means that size reduction must go hand in hand with appropriate smoothing.

Generally, the requirement for the smoothing filter can be formulated as

$$\hat{B}(\tilde{\boldsymbol{k}}) = 0 \quad \forall \tilde{k}_p \geq \frac{1}{r_p}, \tag{5.7}$$

where r_p is the subsampling rate in the direction of the pth coordinate.

The combined smoothing and size reduction can be expressed in a single operator by using the following notation to compute the $q + 1$th level of the Gaussian pyramid from the qth level:

$$\boxed{\boldsymbol{G}^{(0)} = \boldsymbol{G}, \quad \boldsymbol{G}^{(q+1)} = \mathcal{B}_{\downarrow 2}\boldsymbol{G}^{(q)}.} \tag{5.8}$$

The number behind the ↓ in the index denotes the subsampling rate. The 0th level of the pyramid is the original image.

If we repeat the smoothing and subsampling operations iteratively, we obtain a series of images, which is called the *Gaussian pyramid*. From level to level, the resolution decreases by a factor of two; the size of the images decreases correspondingly. Consequently, we can think of the series of images as being arranged in the form of a pyramid as illustrated in Fig. 5.2.

The pyramid does not require much storage space. Generally, if we consider the formation of a pyramid from a W-dimensional image with a subsampling factor of two and M pixels in each coordinate direction, the total number of pixels is given by

$$M^W \left(1 + \frac{1}{2^W} + \frac{1}{2^{2W}} + \ldots\right) < M^W \frac{2^W}{2^W - 1}. \tag{5.9}$$

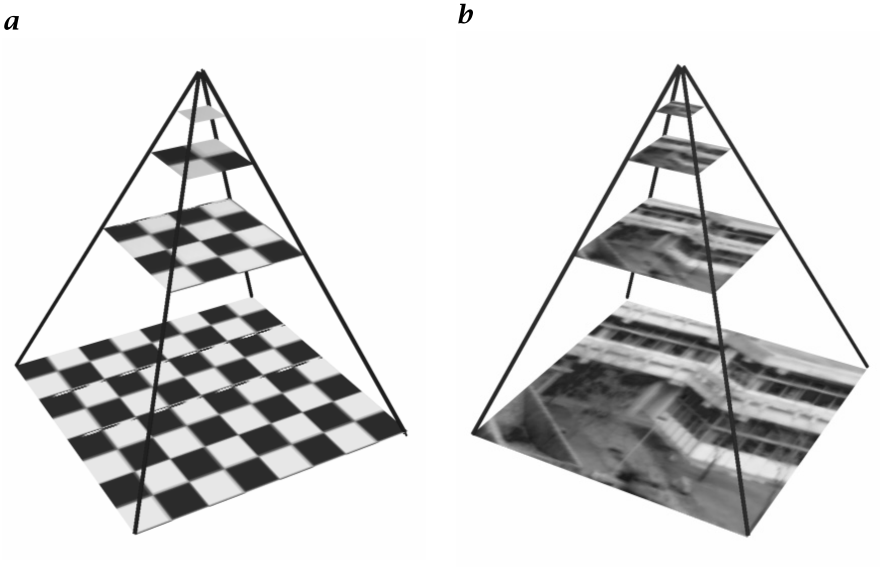

Figure 5.2: *Gaussian pyramid:* ***a*** *schematic representation, the squares of the checkerboard corresponding to pixels;* ***b*** *example.*

For a two-dimensional image, the whole pyramid needs only 1/3 more space than the original image for a three-dimensional image only 1/7 more. Likewise, the computation of the pyramid is equally effective. The *same* smoothing filter is applied to each level of the pyramid. Thus the computation of the *whole* pyramid only needs 4/3 and 8/7 times more operations than for the first level of a two-dimensional and three-dimensional image, respectively.

The pyramid brings large scales into the range of local neighborhood operations with small kernels. Moreover, these operations are performed efficiently. Once the pyramid has been computed, we can perform neighborhood operations on large scales in the upper levels of the pyramid — because of the smaller image sizes — much more efficiently than for finer scales.

The Gaussian pyramid constitutes a series of lowpass-filtered images in which the cut-off wave numbers decrease by a factor of two (an octave) from level to level. Thus only the coarser details remain in the smaller images (Fig. 5.2). Only a few levels of the pyramid are necessary to span all possible wave numbers. For an $N \times N$ image we can compute at most a pyramid with $\mathrm{ld}\, N + 1$ levels. The smallest image consists of a single pixel.

5.2.3 Laplacian Pyramid

From the Gaussian pyramid, another pyramid type can be derived, the *Laplacian pyramid*, which leads to a sequence of bandpass-filtered images. In contrast to the Fourier transform, the Laplacian pyramid only leads to a coarse wave number decomposition without a directional decomposition. All wave numbers, independently of their direction, within the range of about an octave (factor of two) are contained in one level of the pyramid.

Because of the coarse wave number resolution, we can preserve a good spatial resolution. Each level of the pyramid only contains matching scales, which are sampled a few times (two to six) per wavelength. In this way, the Laplacian pyramid is an efficient data structure well adapted to the limits of the product of wave number and spatial resolution set by the *uncertainty relation* (Section 5.1.3 and Theorem 2.7, p. 57,).

In order to achieve this, we subtract two levels of the Gaussian pyramid. This requires an upsampling of the image at the coarser level. This operation is performed by an *expansion operator* $\uparrow_2$. The degree of expansion or upsampling is denoted by the figure after the $\uparrow$ in the index, in a similar notation as for the *reduction operator* Eq. (5.8).

The expansion is significantly more difficult than the size reduction as the missing information must be interpolated. For a size increase of two in all directions, first every second pixel in each row must be interpolated and then every second row. Interpolation is discussed in detail in Section 10.5. With the introduced notation, the generation of the pth level of the Laplacian pyramid can be written as:

$$\boxed{\boldsymbol{L}^{(p)} = \boldsymbol{G}^{(p)} - \uparrow_2 \boldsymbol{G}^{(p+1)}, \quad \boldsymbol{L}^{(P)} = \boldsymbol{G}^{(P)}.} \tag{5.10}$$

The Laplacian pyramid is an effective scheme for a *bandpass decomposition* of an image. The center wave number is halved from level to level. The last image of the Laplacian pyramid, $\boldsymbol{L}^{(P)}$, is a lowpass-filtered image $\boldsymbol{G}^{(P)}$ containing only the coarsest structures.

The Laplacian pyramid has the significant advantage that the original image can be reconstructed quickly from the sequence of images in the Laplacian pyramid by recursively expanding the images and summing them up. The recursion is the inverse of the recursion in Eq. (5.10). In a Laplacian pyramid with $p+1$ levels, the level p (counting starts with zero!) is the coarsest level of the Gaussian pyramid. Then the level $p-1$ of the Gaussian pyramid can be reconstructed by

$$\boxed{\boldsymbol{G}^{(P)} = \boldsymbol{L}^{(P)}, \quad \boldsymbol{G}^{(p-1)} = \boldsymbol{L}^{(p-1)} + \uparrow_2 \boldsymbol{G}^{p}} \tag{5.11}$$

Note that this is just an inversion of the construction scheme for the Laplacian pyramid. This means that even if the interpolation algorithms

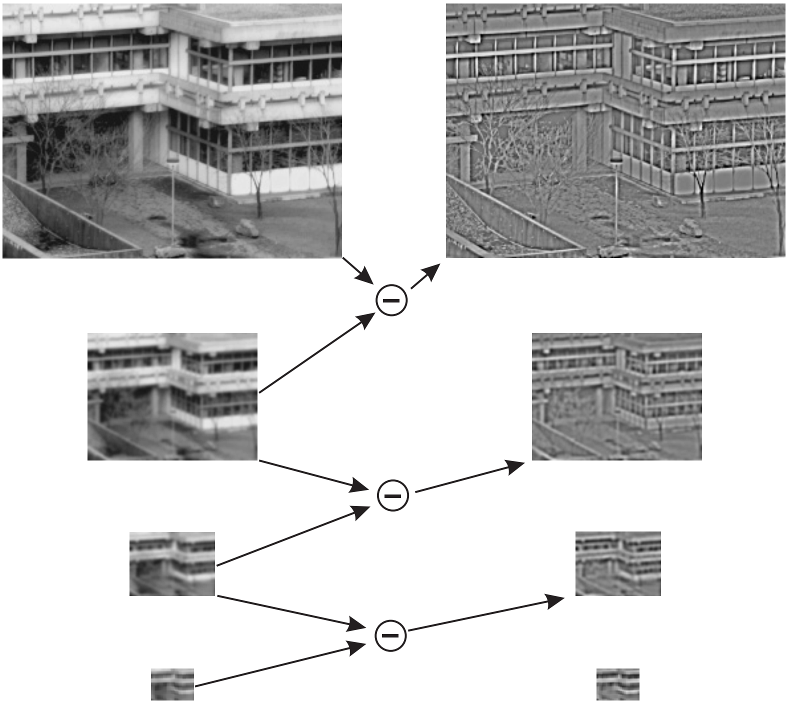

Figure 5.3: Construction of the Laplacian pyramid (right column) from the Gaussian pyramid (left column) by subtracting two consecutive planes of the Gaussian pyramid.

required to expand the image contain errors, they affect only the Laplacian pyramid and not the reconstruction of the Gaussian pyramid from the Laplacian pyramid, as the same algorithm is used. The recursion in Eq. (5.11) is repeated with lower levels until level 0, i. e., the original image, is reached again. As illustrated in Fig. 5.3, finer and finer details become visible during the reconstruction process. Because of the progressive reconstruction of details, the Laplacian pyramid has been used as a compact scheme for image compression. Nowadays, more efficient schemes are available on the basis of wavelet transforms, but they operate on principles very similar to those of the Laplacian pyramid.

5.2.4 *Directio-Pyramidal Decomposition*

In multidimensional signals a directional decomposition is as important as a scale decomposition. Directional decompositions require suitable directional filters. Ideally, all directional components should add up to the complete image. A combined decomposition of an image into a pyramid and on each pyramid level into directional components is known as a *directiopyramidal decomposi-*

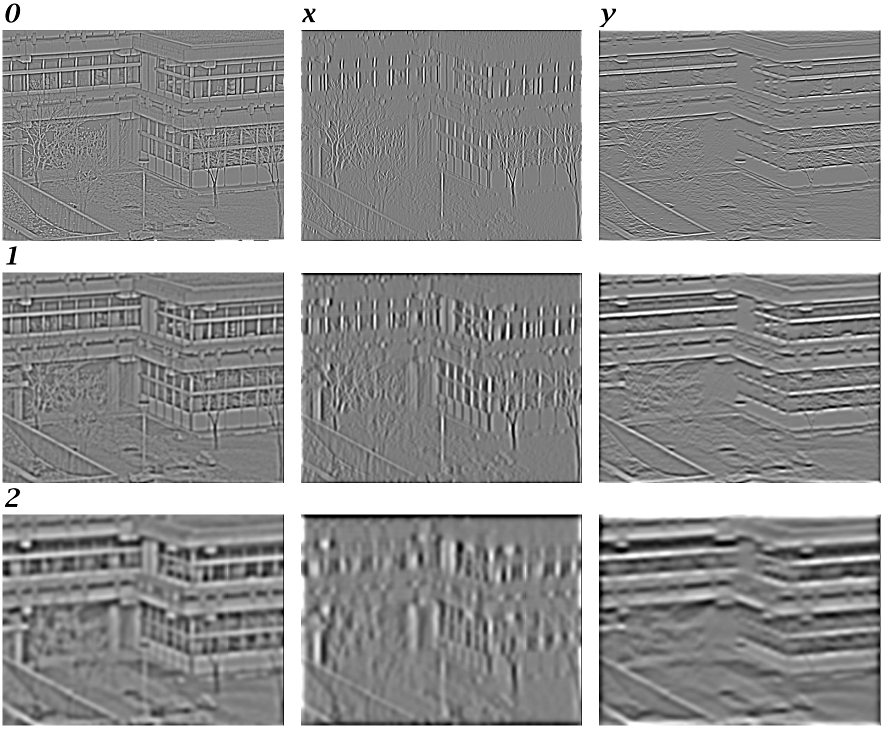

***Figure 5.4:** First three planes of a directiopyramidal decomposition of Fig. 5.6a: the rows shown are planes 0, 1, and 2, the columns **L**, $\boldsymbol{L}_x$, $\boldsymbol{L}_y$ according to Eqs. (5.13) and (5.14).*

tion [86]. Generally, such a decomposition is a difficult filter design problem. Therefore, we illustrate a directiopyramidal decomposition here only with a simple and efficient decomposition scheme with two directional components.

The smoothing is performed by separable smoothing filters, one filter that smoothes only in the x direction ($\mathcal{B}_x$) and one that smoothes only in the y direction ($\mathcal{B}_y$): then the next higher level of the Gaussian pyramid is given as in Eq. (5.8) by

$$\boldsymbol{G}^{(q+1)} = \downarrow_2 \mathcal{B}_x\mathcal{B}_y\boldsymbol{G}^{(q)}. \tag{5.12}$$

The Laplacian pyramid is

$$\boldsymbol{L}^{(q)} = \boldsymbol{G}^{(q)} - \uparrow_2 \boldsymbol{G}^{(q+1)}. \tag{5.13}$$

Then, the two directional components are given by

$$\begin{aligned} \boldsymbol{L}_x^{(q)} &= 1/2(\boldsymbol{G}^{(q)} - \uparrow_2 \boldsymbol{G}^{(q+1)} - (\mathcal{B}_x - \mathcal{B}_y)\boldsymbol{G}^{(q)}), \\ \boldsymbol{L}_y^{(q)} &= 1/2(\boldsymbol{G}^{(q)} - \uparrow_2 \boldsymbol{G}^{(q+1)} + (\mathcal{B}_x - \mathcal{B}_y)\boldsymbol{G}^{(q)}). \end{aligned} \tag{5.14}$$

From Eq. (5.14) it is evident that the two directional components $\boldsymbol{L}_x$ and $\boldsymbol{L}_y$ add up to the isotropic Laplacian pyramid: $\boldsymbol{L} = \boldsymbol{L}_x + \boldsymbol{L}_y$. Example images with the first three levels of a directional decomposition are shown in Fig. 5.4.

5.3 *Scale Spaces*

The Gaussian and Laplacian pyramid are effective but rather inflexible multigrid data structure. From level to level the scale parameter changes by a fixed factor of two. A finer scale selection is not possible. In this section we discuss a more general scheme, the *scale space* that allows a continuous scale parameter.

As we have seen with the example of the windowed Fourier transform in Section 5.1.3, the introduction of a characteristic *scale* adds a new coordinate to the representation of image data. Besides the spatial resolution, we have a new parameter that characterizes the current resolution level of the image data. The scale parameter is denoted by ξ. A data structure that consists of a sequence of images with different resolutions is known as a *scale space*; we write $g(\boldsymbol{x}, \xi)$ to indicate the scale space of the image $g(\boldsymbol{x})$.

Next, in Section 5.3.1, we discuss a physical process, diffusion, that is suitable for generating a scale space. Then we discuss the general properties of a scale space in Section 5.3.2.

5.3.1 *Scale Generation by Diffusion*

The generation of a scale space requires a process that can blur images to a controllable degree. Diffusion is a transport process that tends to level out concentration differences [27]. In physics, diffusion processes govern the transport of heat, matter, and momentum leading to an ever increasing equalization of spatial concentration differences. If we identify the time with the scale parameter ξ, the diffusion process establishes a scale space.

To apply a diffusion process to a multidimensional signal with W dimensions, we regard the gray value g as the concentration of a chemical species. The elementary law of diffusion states that the flux density $\boldsymbol{j}$ is directed against the concentration gradient ∇g and proportional to it:

$$\boldsymbol{j} = -D\nabla g \tag{5.15}$$

where the constant D is known as the *diffusion coefficient*. Using the continuity equation

$$\frac{\partial g}{\partial t} + \nabla \boldsymbol{j} = 0 \tag{5.16}$$

the diffusion equation is

$$\boxed{\frac{\partial g}{\partial t} = \nabla(D\nabla g).} \tag{5.17}$$

For the case of a homogeneous diffusion process (D does not depend on the position), the equation reduces to

$$\frac{\partial g}{\partial t} = D\Delta g \tag{5.18}$$

where

$$\Delta = \sum_{w=1}^{W} \frac{\partial^2}{\partial x_w^2} \tag{5.19}$$

is the *Laplacian operator*. It is easy to show that the general solution to this equation is equivalent to a convolution with a smoothing mask. To this end, we perform a spatial Fourier transform which results in

$$\frac{\partial \hat{g}(\boldsymbol{k})}{\partial t} = -4\pi^2 D|\boldsymbol{k}|^2 \hat{g}(\boldsymbol{k}) \tag{5.20}$$

by using Theorem 2.5, p. 55 and reduces the equation to a linear first-order differential equation with the general solution

$$\hat{g}(\boldsymbol{k}, t) = \exp(-4\pi^2 D|\boldsymbol{k}|^2 t)\hat{g}(\boldsymbol{k}, 0), \tag{5.21}$$

where $\hat{g}(\boldsymbol{k}, 0)$ is the Fourier transformed image at time zero.

Multiplication of the image in the Fourier space with the Gaussian function in Eq. (5.21) is equivalent to a convolution with the same function but of reciprocal width (Theorem 2.4, p. 54, $\succ$ R4 und $\succ$ R6). Thus,

$$g(\boldsymbol{x}, t) = \frac{1}{[2\pi\sigma^2(t)]^{W/2}} \exp\left(-\frac{|\boldsymbol{x}|^2}{2\sigma^2(t)}\right) * g(\boldsymbol{x}, 0) \tag{5.22}$$

with

$$\sigma(t) = \sqrt{2Dt}. \tag{5.23}$$

Equation (5.23) shows that the degree of smoothing expressed by the standard deviation σ increases only with the square root of the time. Therefore we set the scale parameter ξ equal to the square of the standard deviation:

$$\xi = 2Dt. \tag{5.24}$$

It is important to note that this formulation of the scale space is valid for images of any dimension. It could also be extended to image sequences. The scale parameter is not identical to the time although we used a physical diffusion process that proceeds with time to derive it. If we compute a scale space representation of an image sequence, it is useful to scale the time coordinate with a characteristic velocity u_0 so that it has the same dimension as the spatial coordinates:

$$t' = u_0 t. \tag{5.25}$$

We add this coordinate to the spatial coordinates and get a new coordinate vector

$$\boldsymbol{x} = [x_1, x_2, u_0 t]^T \quad \text{or} \quad \boldsymbol{x} = [x_1, x_2, x_3, u_0 t]^T. \tag{5.26}$$

In the same way, we extend the wave number vector by a scaled frequency:

$$\boldsymbol{k} = [k_1, k_2, \nu/u_0]^T \quad \text{or} \quad \boldsymbol{k} = [k_1, k_2, k_3, \nu/u_0]^T. \tag{5.27}$$

With Eqs. (5.26) and (5.27) all equations derived above, e.g., Eqs. (5.21) and (5.22), can also be applied to scale spaces of image sequences. For discrete spaces, of course, no such scaling is required. It is automatically fixed by the spatial and temporal sampling intervals: $u_0 = \Delta x/\Delta t$.

As an illustration, Fig. 5.5 shows the scale space of some characteristic one-dimensional signals: noisy edges and lines, a periodic pattern, a random signal, and a row of an image. These examples nicely demonstrate a general property

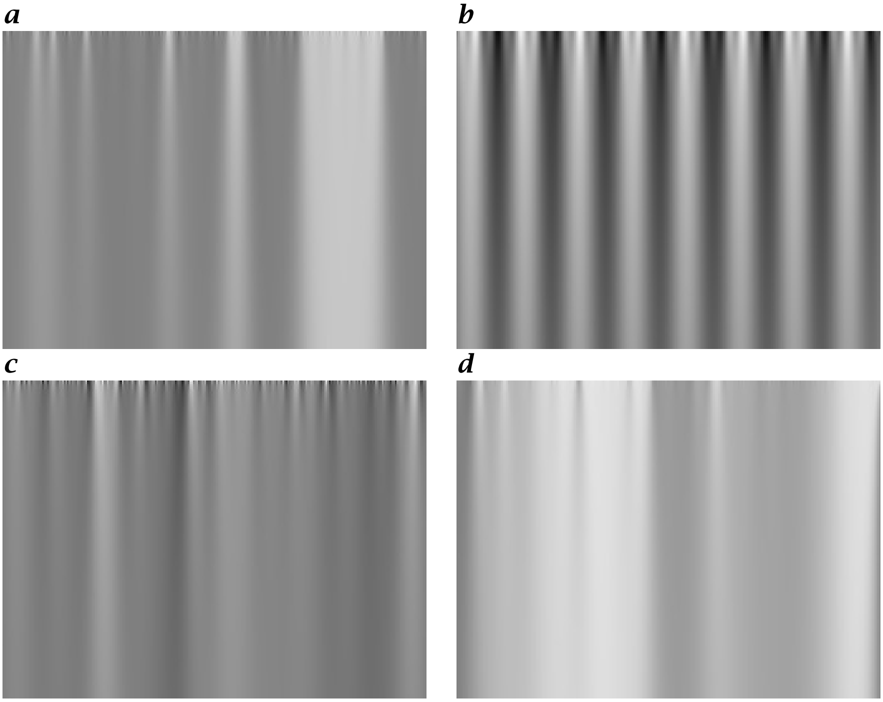

Figure 5.5: Scale space of some one-dimensional signals: ***a*** *edges and lines;* ***b*** *a periodic pattern;* ***c*** *a random signal;* ***d*** *row 10 from the image shown in Fig. 11.6a. The vertical coordinate is the scale parameter ξ.*

of scale spaces. With increasing scale parameter ξ, the signals become increasingly blurred, more and more details are lost. This feature can be most easily seen by the transfer function of the scale space representation in Eq. (5.21). The transfer function is always positive and monotonically decreasing with the increasing scale parameter ξ for all wave numbers. This means that no structure is amplified. All structures are attenuated with increasing ξ, and smaller structures always faster than coarser structures. In the limit of $\xi \to \infty$ the scale space converges to a constant image with the mean gray value. A certain feature exists only over a certain scale range. In Fig. 5.5a we can observe that edges and lines disappear and two objects merge into one.

For two-dimensional images, a continuous representation of the scale space would give a three-dimensional data structure. Therefore Fig. 5.6 shows individual images for different scale parameters ξ as indicated.

5.3.2 *General Properties of a Scale Space*

In this section, we discuss some general properties of scale spaces. More specifically, we want to know what kind of conditions must be met by a filter kernel generating a scale space. We will discuss two basic requirements. First, no new

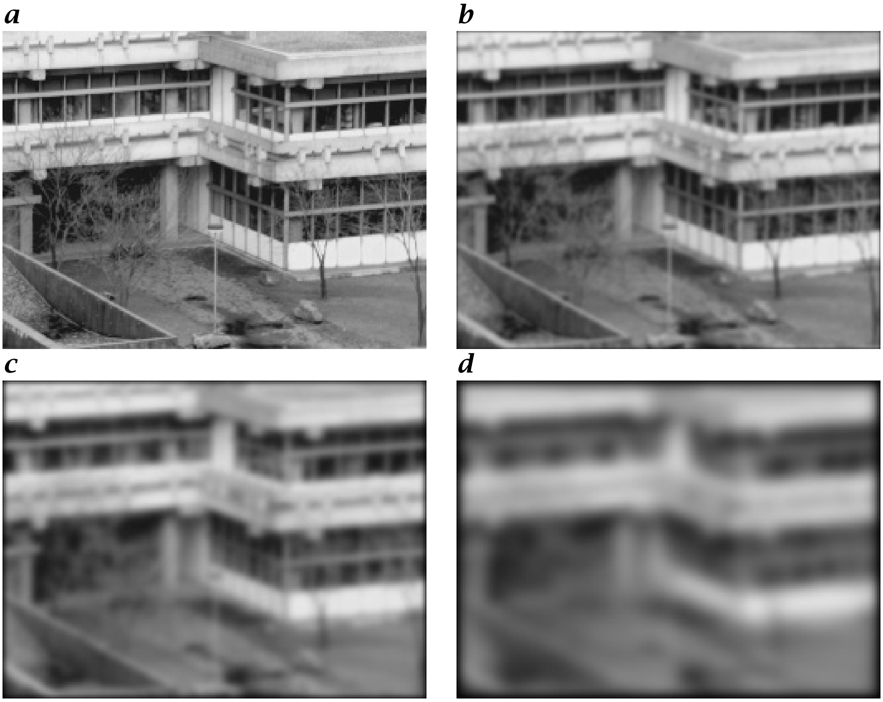

***Figure 5.6:** Scale space of a two-dimensional image: **a** original image; **b**, **c**, and **d** at scale parameters σ 1, 2, and 4, respectively.*

details must be added with increasing scale parameter. From the perspective of information theory, we may say that the information content in the signal should continuously decrease with the scale parameter.

The second property is related to the general principle of *scale invariance*. This basically means that we can start smoothing the signal at any scale parameter in the scale space and still obtain the same scale space. Here, we will give only some basic ideas about these elementary properties and no proofs. For a detailed treatment of the scale space theory we refer to the recent monograph on linear scale space theory by Lindeberg [125].

The linear homogenous and isotropic diffusion process has according to Eq. (5.22) the convolution kernel

$$\boldsymbol{B}(\boldsymbol{x},\xi) = \frac{1}{2\pi\xi}\exp\left(-\frac{|\boldsymbol{x}|^2}{2\xi}\right) \tag{5.28}$$

and the transfer function Eq. (5.21)

$$\hat{\boldsymbol{B}}(\boldsymbol{k},\xi) = \exp(-4\pi^2|\boldsymbol{k}|^2\xi/2). \tag{5.29}$$

In these equations, we have replaced the explicit dependence on time by the scale parameter ξ using Eq. (5.24). In a representation-independent way, we denote the scale space generating operator as

$$\mathcal{B}(\xi). \tag{5.30}$$

The information-decreasing property of the scale space with ξ can be formulated mathematically in different ways. We express it here with the *minimum-maximum principle* which states that local extrema must not be enhanced. This means that the gray value at a local maximum or minimum must not increase or decrease, respectively. For a diffusion process this is an intuitive property. For example, in a heat transfer problem, a hot spot must not become hotter or a cool spot cooler. The Gaussian kernel Eq. (5.28) meets the minimum-maximum principle.

The second important property of the scale space is related to the *scale invariance* principle. We want to start the generating process at any scale parameter and still get the same scale space. More quantitatively, we can formulate this property as

$$\mathcal{B}(\xi_2)\mathcal{B}(\xi_1) = \mathcal{B}(\xi_1 + \xi_2). \tag{5.31}$$

This means that the smoothing of the scale space at the scale ξ_1 by an operator with the scale ξ_2 is equivalent to the application of the scale space operator with the scale $\xi_1 + \xi_2$ to the original image. Alternatively, we can state that the representation at the coarser level ξ_2 can be computed from the representation at the finer level ξ_1 by applying

$$\mathcal{B}(\xi_2) = \mathcal{B}(\xi_2 - \xi_1)\mathcal{B}(\xi_1) \quad \text{with} \quad \xi_2 > \xi_1. \tag{5.32}$$

From Eqs. (5.28) and (5.29) we can easily verify that Eqs. (5.31) and (5.32) are true. In mathematics the properties Eqs. (5.31) and (5.32) are referred to as the *semi-group property*.

Conversely, we can ask what scale space generating kernels exist that meet both the minimum-maximum principle and the semi-group property. The answer to this question may be surprising. The Gaussian kernel is the *only* convolution kernel that meets both these criteria and is in addition isotropic and homogeneous [125]. This feature puts the Gaussian convolution kernel and — as we will see later — its discrete counterpart the binomial kernel into a unique position for image processing. It will be elaborated in more detail in Section 11.4.

It is always instructive to discuss a counterexample. The most straightforward smoothing kernel for a W-dimensional image — known as the *moving average* — is the box filter

$$\boldsymbol{R}(\boldsymbol{x}, \xi) = \frac{1}{\xi^W} \prod_{w=1}^{W} \Pi\left(\frac{x_w}{\xi}\right) \tag{5.33}$$

with the transfer function

$$\hat{\boldsymbol{R}}(\boldsymbol{k}, \xi) = \prod_{w=1}^{W} \frac{\sin(k_w \xi/2)}{k_w \xi/2}. \tag{5.34}$$

This kernel meets neither the minimum-maximum principle nor the semi-group property. Figure 5.7 compares scale spaces of a periodic signal with varying wave number generated with a Gaussian and a box kernel. In Fig. 5.7b it becomes evident that the box kernel does not meet the minimum-maximum principle as structures decrease until they are completely removed but then appear again.

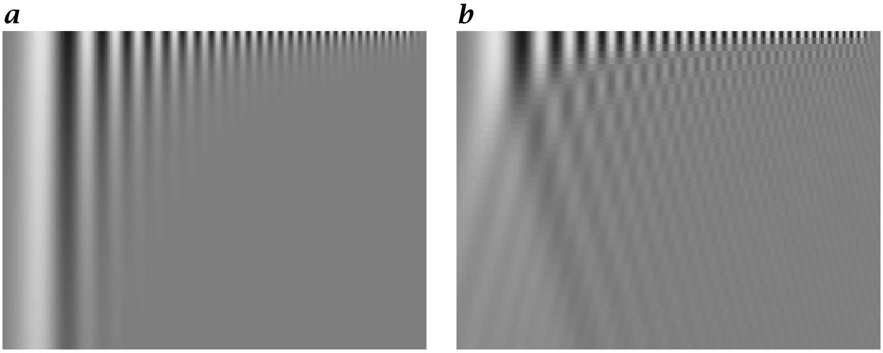

Figure 5.7: *Scale space of a 1-D signal with varying wave number computed with a **a** Gaussian and **b** box kernel. The scale parameter runs from top to bottom.*

5.3.3 *Quadratic and Exponential Scale Spaces*

Despite the mathematical beauty of scale space generation with a Gaussian convolution kernel, this approach has one significant disadvantage. The standard deviation of the smoothing increases only with the square root of the time, see Eq. (5.23). Therefore the scale parameter ξ is only proportional to the square of the standard deviation. This results in a nonlinear scale coordinate. While smoothing goes fast for fine scales, it becomes increasingly slower for larger scales.

There is a simple cure for this problem. We need a diffusion process where the diffusion constant increases with time. We first discuss a diffusion coefficient that increases linearly with time. This approach results in the differential equation

$$\frac{\partial g}{\partial t} = D_0 t \Delta g. \qquad (5.35)$$

A spatial Fourier transform results in

$$\frac{\partial \hat{g}(\boldsymbol{k})}{\partial t} = -4\pi^2 D_0 t |\boldsymbol{k}|^2 \hat{g}(\boldsymbol{k}). \qquad (5.36)$$

This equation has the general solution

$$\hat{g}(\boldsymbol{k}, t) = \exp(-2\pi^2 D_0 t^2 |\boldsymbol{k}|^2) \hat{g}(\boldsymbol{k}, 0) \qquad (5.37)$$

which is equivalent to a convolution in the spatial domain. Thus,

$$g(\boldsymbol{x}, t) = \frac{1}{2\pi D_0 t^2} \exp\left(-\frac{|\boldsymbol{x}|^2}{2 D_0 t^2}\right) * g(\boldsymbol{x}, 0). \qquad (5.38)$$

From these equations we can write the convolution kernel and transfer function in the same form as in Eqs. (5.28) and (5.29) with the only exception that the scale parameter

$$\xi_q = D_0 t^2. \qquad (5.39)$$

Now the standard deviation for the smoothing is proportional to time for a diffusion process that increases linearly in time. As the scale parameter ξ is proportional to the time squared, we denote this scale space as the *quadratic scale space*. This modified scale space still meets the minimum-maximum principle and the semi-group property.

For even more accelerated smoothing, we can construct an *exponential scale space*, i. e., a scale space where the logarithm of the scale parameter increases linearly with time. We use a diffusion coefficient that increases exponentially in time

$$\frac{\partial g}{\partial t} = D_0 \exp(t/\tau)\Delta g. \tag{5.40}$$

Again, we obtain a convolution kernel and a transfer function as in Eqs. (5.28) and (5.29), now with the scale parameter

$$\xi_l = 2D_0\tau \exp(t/\tau). \tag{5.41}$$

5.3.4 *Differential Scale Spaces*

The interest in a *differential scale space* stems from the fact that we want to select optimum scales for processing of features in images. In a differential scale space, the change of the image with scale is emphasized. We use the transfer function of the scale space kernel Eq. (5.29) which is also valid for quadratic and logarithmic scale spaces. The general solution for the scale space can be written in the Fourier space as

$$\hat{g}(\boldsymbol{k}, \xi) = \exp(-2\pi^2|\boldsymbol{k}|^2\xi)\hat{g}(\boldsymbol{k}, 0). \tag{5.42}$$

Differentiating this signal with respect to the scale parameter ξ yields

$$\frac{\partial \hat{g}(\boldsymbol{k}, \xi)}{\partial \xi} = -2\pi^2|\boldsymbol{k}|^2 \exp(-2\pi^2|\boldsymbol{k}|^2\xi)\hat{g}(\boldsymbol{k}, 0) = -2\pi^2|\boldsymbol{k}|^2\hat{g}(\boldsymbol{k}, \xi). \tag{5.43}$$

The multiplication with $-|\boldsymbol{k}|^2$ is equivalent to a second-order spatial derivative (≻ R4), the *Laplacian operator*. Thus we can write in the spatial domain

$$\frac{\partial g(\boldsymbol{x}, \xi)}{\partial \xi} = \frac{1}{2}\Delta g(\boldsymbol{x}, \xi). \tag{5.44}$$

Equations (5.43) and (5.44) constitute a basic property of the differential scale space. The differential scale space is equivalent to a second-order spatial derivation with the Laplacian operator and thus leads to an isotropic *bandpass decomposition* of the image. The transfer function at the scale ξ is

$$-2\pi^2|\boldsymbol{k}|^2 \exp(-2\pi^2|\boldsymbol{k}|^2\xi). \tag{5.45}$$

For small wave numbers, the transfer function is proportional to $-|\boldsymbol{k}|^2$. It reaches a maximum at

$$k^2_{\max} = \frac{2}{\xi} \tag{5.46}$$

and then decays exponentially.

5.3.5 Discrete Scale Spaces

The construction of a *discrete scale space* requires a discretization of the diffusion equation. We start with a discretization of the one-dimensional diffusion equation

$$\frac{\partial g(x,\xi)}{\partial \xi} = D\frac{\partial^2 g(x,\xi)}{\partial x^2}. \tag{5.47}$$

The derivatives are replaced by discrete differences in the following way:

$$\begin{aligned}\frac{\partial g(x,\xi)}{\partial \xi} &= \frac{g(x,\xi+\Delta\xi)-g(x,\xi)}{\Delta\xi}\\ \frac{\partial^2 g(x,\xi)}{\partial x^2} &= \frac{g(x+\Delta x,\xi)-2g(x,\xi)+g(x-\Delta x,\xi)}{\Delta x^2}.\end{aligned} \tag{5.48}$$

This leads to the following iterative scheme for computing a discrete scale space with $\epsilon = D\Delta\xi/\Delta x^2$:

$$g(x,\xi+\Delta\xi) = \epsilon g(x+\Delta x,\xi) + (1-2\epsilon)g(x,\xi) + \epsilon g(x-\Delta x,\xi) \tag{5.49}$$

or written with discrete coordinates ($\xi \to i$, $x \to n$)

$${}^{i+1}g_n = \epsilon\, {}^{i}g_{n+1} + (1-2\epsilon)\, {}^{i}g_n + \epsilon\, {}^{i}g_{n-1}. \tag{5.50}$$

Lindeberg [125] shows that this iteration results in a discrete scale space that meets the minimum-maximum principle and the semi-group property if and only if

$$\epsilon \leq 1/4. \tag{5.51}$$

The limiting case of $\epsilon = 1/4$ leads to the especially simple iteration

$${}^{i+1}g_n = 1/4\, {}^{i}g_{n+1} + 1/2\, {}^{i}g_n + 1/4\, {}^{i}g_{n-1}. \tag{5.52}$$

Each step of the scale space computation is given by a spatial smoothing of the signal with the mask $\boldsymbol{B}^2 = [1\ 2\ 1]/4$. We can also formulate the general scale space generating operator in Eq. (5.49) using the convolution operator $\mathcal{B}$. Written in the operator notation introduced in Section 4.1.4, the operator for one iteration step to generate the discrete scale space is

$$(1-4\epsilon)\mathcal{I} + 4\epsilon\mathcal{B}^2 \quad \text{with} \quad \epsilon \leq 1/4, \tag{5.53}$$

where $\mathcal{I}$ denotes the identity operator.

This expression is significant, as it can be extended directly to higher dimensions by replacing $\mathcal{B}^2$ with a correspondingly higher-dimensional smoothing operator. The convolution mask $\boldsymbol{B}^2$ is the simplest mask in the class of smoothing binomial filters. These filters will be discussed in detail in Section 11.4.

5.4 *Exercises*

5.1: Pyramids

Interactive demonstration of Gaussian und Laplacian pyramids (dip6ex05.01).

5.2: **Smoothing filters for Gaussian pyramids

The first papers about pyramids from Burt and Adelson [19] and Burt [18] used smoothing filters with 5 coefficients, e. g., the filters

$$[1\ 4\ 6\ 4\ 1]/16, \quad [1\ 2\ 3\ 2\ 1]/9.$$

These filters were first applied in horizontal direction and then in vertical direction.

1. Do these filters meet the condition expressed by Eq. (5.7) that the transfer function should be zero for $\tilde{k}_1 > 1/2$ or $\tilde{k}_2 > 1/2$?
2. Is it possible at all that a filter with finite point spread function can meet this condition *exactly*?

5.3: **Construction of the Laplacian pyramid

The Laplacian pyramid could also be constructed according to the following scheme as an alternative to Eq. (5.10):

$$\boldsymbol{L}^{(p)} = \boldsymbol{G}^{(p)} - \boldsymbol{B}\boldsymbol{G}^{(p)}, \quad \boldsymbol{G}^{(p+1)} = \downarrow_2 \boldsymbol{B}\boldsymbol{G}^{(p)}, \quad \boldsymbol{L}^{(P)} = \boldsymbol{G}^{(P)}.$$

The smoothed pth level of the Gaussian pyramid is simply subtracted from itself without applying a downsampling. A downsampling is only applied to compute the $(p+1)$th level of the Gaussian pyramid.

1. Determine the equation that is aquivalent to Eq. (5.11) in order to reconstruct the Gaussian pyramid from the Laplacian pyramid.
2. Do you see any advantage or disadvantage with this scheme as compared to the scheme described by Eqs. (5.10) and (5.11)?

5.4: ***Pyramid with finer scale resolution

One problem of conventional pyramids is that the size decreasing in every direction by a fixed factor of two. Some applications call for a finer scale resolution.

How could you generate a pyramid where the size in both directions decreases not by a factor of two but by a factor of $\sqrt{2}$?

(Hint: You need to find a scheme that selects only every second pixel from a 2-D image.)

5.5: Scale space

Interactive demonstration of various scale spaces and their properties (dip6ex05.02).

5.6: ****Discrete scale space with box filters**

A discrete scale space should be constructed using box filters (running average) with increasing filter length. The filter length determines the scale parameter $\xi = 2R + 1$. Answer the following questions:

1. Is the minimum-maximum principle met?
2. Is this scale space scale invariant, i. e., does it meet the semi-group property
$$\mathcal{R}(\xi_1)\mathcal{R}(\xi_2) = \mathcal{R}(\xi_1 + \xi_2)?$$

5.5 *Further Readings*

Multiresolutional image processing developed in the early 1980ies. An excellent overview of this early work is given by Rosenfeld [171]. Linear scale spaces are described in detail by the monograph of Lindeberg [125], nonlinear scale spaces including inhomogeneous and anisotropic diffusion by Weickert [214]. Readers interested in the recent development of scale space theory are referred to the proceedings of the international conferences on “Scale-Space”: 1997 [197], 1999 [145], 2001 [106], 2003 [65], and 2005 [107].

Part II

Image Formation and Preprocessing