17 Regularization and Modeling

17.1 Introduction

17.1.1 Unifying Local Analysis and Global Knowledge

The model-based segmentation technique discussed in Section 16.5 is a first step toward integrating global information into the process of object recognition. It is an inflexible technique, however, as it requires an exact parameterization of the objects to be detected. For real objects, it is often not possible to establish such an explicit type of model.

In this chapter, we discuss very general approaches to link local with global information that does not require an explicit model of the object. Instead it uses flexible constraints to include information of global type. The basic idea is to balance two counteracting requirements. On the one side, the model should reproduce the given image data as close as possible. This requirement is known as the *similarity constraint*. On the other side, the modeled data should meet some general global constraints that can be inferred from the general knowledge about the observed scene. In the simplest case this could be a *smoothness constraint*.

Generally, it is not possible to obtain an exact solution. Because all real-world image data incorporate a certain uncertainty, an exact fit of the data makes no sense. We rather expect a certain deviation of the computed model data from the image data that can be compared with the expected standard deviation of the noise contained in the data.

Thus we end up with a *global optimization* problem. Both kinds of constraint must be combined in an appropriate way to find a solution that has a minimum error with a given error norm.

This general approach can be applied to a wide range of image analysis problems including such diverse tasks as

- *restoration* of images degraded by the image formation process (Chapter 7),
- computation of *depth maps* from *stereo images* or any other imaging sensor based on triangulation techniques (Chapter 8.2),
- computation of *depth maps* from *shape from shading* or *photometric stereo* (Chapter 8.5),
- *reconstruction* of images from 3-D imaging techniques such as *tomography* (Section 8.6) that deliver no direct images,

- computation of motion or *displacement vector fields* from image sequences (Chapter 14),
- partition of images into regions (*segmentation*, Chapter 16), and
- computation of object boundaries (*active contours* or *snakes*).

Most of the features to be computed are scalar fields, but some of them, such as the motion field or surface normals, are vector fields. Therefore it is useful to extend the image modeling method to vector quantities.

Before we start, it is useful to consider the purpose and limits of modeling (Section 17.1.2). After detailing the general approach of variational image modeling in Section 17.2, we will discuss in Section 17.2.5 the important question discontinuities can be adequately incorporated into global smoothness constraints. The variational approach results in partial differential equations that are equivalent to transport equations including diffusion and reaction. Thus the discussion of diffusion models in Section 17.3 casts another interesting view on the problem of image modeling.

In the second part of this chapter, we turn to the discrete part of image modeling and see that it can be understood as a *discrete inverse problem* (Section 17.4). Electrical networks serve as an illustrative example (Section 17.6.2). In Section 17.5 we finally show with the example of *inverse filtering* how inverse problems can be solved efficiently.

17.1.2 Purpose and Limits of Models

The term *model* reflects the fact that any natural phenomenon can only be described to a certain degree of accuracy and correctness. It is one of the most powerful principles throughout all natural sciences to seek the simplest and most general description that still describes the observations with minimum deviations. A handful of basic laws of physics describe an enormously wide range of phenomena in a quantitative way.

Along the same lines, models are a useful and valid approach for image processing tasks. However, models must be used with caution. Even if the data seem to be in perfect agreement with the model assumptions, there is no guarantee that the model assumptions are correct.

Figure 17.1 shows an illustrative example. The model assumptions include a flat black object lying on a white background that is illuminated homogeneously (Fig. 17.1a). The object can be identified clearly by low gray values in the image, and the discontinuities between the high and low values mark the edges of the object.

If the black object has a non-negligible thickness, however, and the scene is illuminated by an oblique parallel light beam (Fig. 17.1c), we receive exactly the same type of profile as for Fig. 17.1a. Thus, we do not detect any deviation from the model assumption. Still only the right

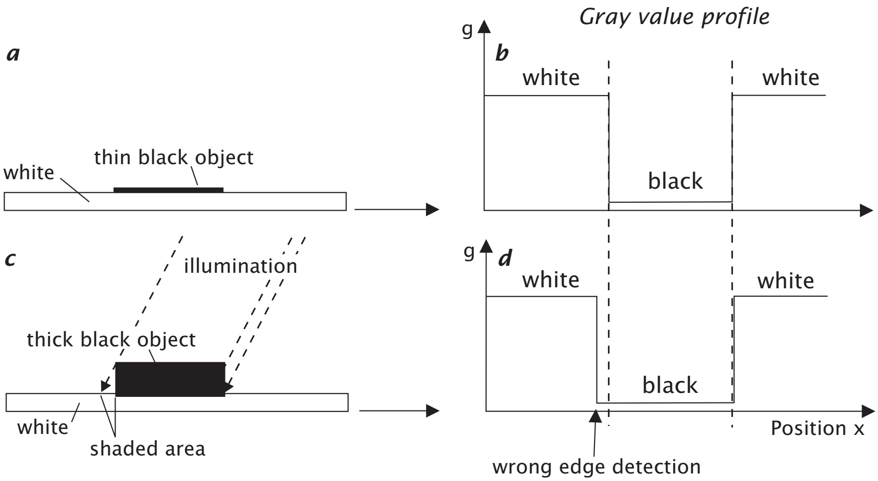

Figure 17.1: *Demonstration of a systematic error which cannot be inferred from the perceived image.* ***a*** *,* **c** *sketch of the object and illumination conditions;* ***b*** *and* ***d*** *resulting gray value profiles for* ***a*** *and* **c** *, respectively.*

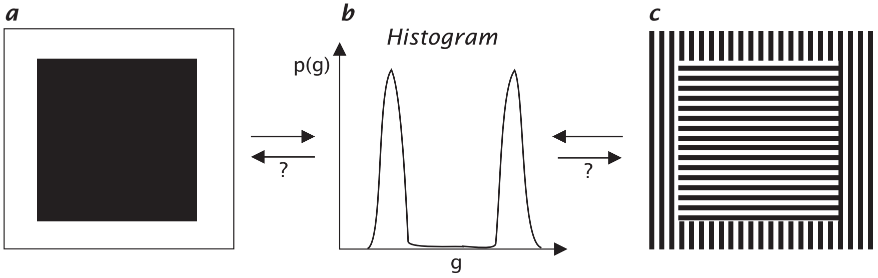

Figure 17.2: *Demonstration of a systematic deviation from a model assumption (object is black, background white) that cannot be inferred from the image histogram.*

edge is detected correctly. The left edge is shifted to the left because of the shadowed region resulting in an image too large for the object.

Figure 17.2 shows another case. A black flat object fills half of the image on a white background. The histogram (the distribution of the gray values) clearly shows a bimodal shape with two peaks of equal height. This tells us that basically only two gray values occur in the image, the lower being identified as the black object and the higher as the white background, each filling half of the image.

This does not mean, however, that any bimodal histogram stems from an image where a black object fills half of the image against a white background. Many other interpretations are possible. For instance, also a white object could be encountered on a black background. The same bimodal histogram is also gained from an image in which both the object and the background are striped black and white. In the latter case, a segmentation procedure that allocates all pixels below a certain threshold to the object and the others to the background would not extract the desired object but the black stripes. This simple procedure only works if the model assumption is met that the objects and the background are of uniform brightness.

The two examples discussed above clearly demonstrate that even in simple cases we can run into situations where the model assumptions appear to be met — as judged by the image or quantities derived from the image such as histograms — but actually are not. While it is quite easy to see the failure of the model assumption in these simple cases, this may be more difficult if not impossible in more complex cases.

17.2 *Continuous Modeling I: Variational Approach*

As discussed in the introduction (Section 17.1.1), a mathematically well-founded approach to image modeling requires the setup of a model function and an error functional that measures the residual deviations of the measured data from the computed model data.

For image segmentation, a suitable modeling function could be a piecewise flat target function $f(\boldsymbol{x})$. Regions with constant values correspond to segmented objects and discontinuities to object boundaries. The free parameters of this model function would be gray values in the different regions and the boundaries between the regions. The boundaries between the objects and the gray values of the regions should be varied in such a way that the deviation between the model function $f(\boldsymbol{x})$ and the image data $g(\boldsymbol{x})$ are minimal.

The global constraints of this segmentation example are rather rigid. Smoothness constraints are more general. They tend to minimize the spatial variations of the feature. This concept is much more general than using a kind of fixed model saying that the feature should be constant as in the above segmentation example or vary only linearly.

Such global constraints can be handled in a general way using *variation calculus*. Before we turn to the application of variation calculus in image modeling, it is helpful to start with a simpler example from physics.

17.2.1 *Temporal Variational Problems: A Simple Example*

Variation calculus has found widespread application throughout the natural sciences. It is especially well known in physics. All basic concepts of theoretical physics can be formulated as extremal principles. Probably the best known is *Hamilton's principle* which leads to the Lagrange equation in *theoretical mechanics* [60].

As a simple example, we discuss the motion of a mass point. Without external forces, the mass point will move with constant speed. The higher the mass, the more force is required to change its speed. Thus the mass tends to smoothen the velocity when the particle is moving through a spatially and temporally varying potential field $V(x,t)$ that applies the force $F = V_x(x,t)$ to the particle.

Hamilton's principle says that the motion follows a path for which the following integral is extreme:

$$\int_{t_1}^{t_2} \frac{1}{2} m x_t^2 - V(x,t) \mathrm{d}t. \tag{17.1}$$

The temporal derivative of x is denoted in Eq. (17.1) by x_t. The function in the integral is known as the *Lagrange function* $L(x, x_t, t)$. The Lagrange function depends on the position x and the time t via the potential $V(x,t)$ and on the temporal derivative of the position, i. e., the velocity, via the kinetic energy $m x_t^2/2$ of the mass point.

The above integral equation is solved by the *Euler-Lagrange equation*

$$\frac{\partial L}{\partial x} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial x_t} = 0 \quad \text{or short} \quad L_x - \frac{\mathrm{d}}{\mathrm{d}t} L_{x_t} = 0. \tag{17.2}$$

By this equation, the integral equation (Eq. (17.1)) can be converted into a differential equation for a given Lagrange function.

As an illustrative example we compute the motion of a mass point in a harmonic potential $V(x) = \epsilon x^2/2$. The Lagrange function of this system is

$$L(x, x_t, t) = T - V = \frac{1}{2} m (x_t)^2 - \frac{1}{2} \epsilon x^2. \tag{17.3}$$

The derivatives of the Lagrange function are

$$\frac{\partial L}{\partial x} = -\epsilon x, \quad \frac{\partial L}{\partial x_t} = m x_t, \quad \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial x_t} = m x_{tt}. \tag{17.4}$$

From the Euler equation (Eq. (17.2)) we obtain the simple second-order differential equation

$$m x_{tt} + \epsilon x = 0. \tag{17.5}$$

This second-order differential equation describes a harmonic oscillation of the mass point in the potential with a circular frequency $\omega = \sqrt{\epsilon/m}$.

17.2.2 *Spatial and Spatiotemporal Variation Problems*

In image processing it is required to formulate the variation problem for spatially and temporally varying variables. The path of the mass point $x(t)$, a scalar function, has to be replaced by a spatial function or spatiotemporal $f(\boldsymbol{x})$, i. e., by a scalar vector function of a vector variable. For image sequences, one of the components of $\boldsymbol{x}$ is the time t.

Consequently, the Lagrange function now depends on the vector variable $\boldsymbol{x}$. Furthermore, it will not only be a function of $f(\boldsymbol{x})$ and $\boldsymbol{x}$ explicitly. There will be additional terms depending on the spatial (and possibly temporal) partial derivatives of f. They are required as soon as we demand that f at a point should be dependent on f in the neighborhood. In conclusion, the general formulation of the *error functional* $\varepsilon(f)$ as a variation integral for g reads

$$\varepsilon(f) = \int_{\Omega} L\left(f, f_{x_w}, \boldsymbol{x}\right) \mathrm{d}x^W \to \text{minimum}. \tag{17.6}$$

The area integral is calculated over a certain image domain $\Omega \in \mathbb{R}^W$. Equation (17.6) already contains the knowledge that the extreme is a minimum. This results from the fact that f should show a minimum deviation from the given functions at certain points with additional constraints.

The corresponding Euler-Lagrange equation is:

$$L_f - \sum_{w=1}^{W} \partial_{x_w} L_{f_{x_w}} = 0. \qquad (17.7)$$

The variational approach can also be extended to vectorial features such as the velocity in image sequences. Then, the Lagrange function depends on the vectorial feature $\boldsymbol{f} = [f_1, f_2, \ldots, f_P]^T$, the partial derivatives of each component f_p of the feature in all directions $(f_p)_{x_w}$, and explicitly on the coordinate $\boldsymbol{x}$:

$$\boxed{\varepsilon(\boldsymbol{f}) = \int_{\Omega} L\left(\boldsymbol{f}, (f_p)_{x_w}, \boldsymbol{x}\right) \mathrm{d}x^W \to \text{minimum.}} \qquad (17.8)$$

From this equation, we obtain an Euler–Lagrange equation for each component f_i of the vectorial feature:

$$\boxed{L_{f_p} - \sum_{p=w}^{W} \partial_{x_w} L_{(f_p)_{x_w}} = 0.} \qquad (17.9)$$

17.2.3 *Similarity Constraints*

The similarity term is used to make the modeled feature similar to the measured feature. For a simple segmentation problem, in which the objects can be distinguished by their gray value, the measured feature is the gray value itself and the similarity term S is given by

$$L(f, \boldsymbol{x}) = S(f, \boldsymbol{x}) = \|f(\boldsymbol{x}) - g(\boldsymbol{x})\|_n. \qquad (17.10)$$

This simply means that the deviation between the modeled feature and the image measured with the L_n norm should be minimal. The most commonly used norm is the L_2 norm, leading to the well known *least squares* (LS) approach.

For a linear *restoration* problem, the original image $f(\boldsymbol{x})$ is degraded by a convolution operation with the point spread function of the degradation $h(\boldsymbol{x})$ (for further details, see Section 17.5). Thus the measured image $g(\boldsymbol{x})$ is given by

$$g(\boldsymbol{x}) = h(\boldsymbol{x}) * f(\boldsymbol{x}). \qquad (17.11)$$

In order to obtain a minimum deviation between the measured and reconstructed images, the similarity term is

$$S(f, \boldsymbol{x}) = \|h(\boldsymbol{x}) * f(\boldsymbol{x}) - g(\boldsymbol{x})\|_n. \qquad (17.12)$$

As a last example, we discuss the similarity constraint for motion determination. In Section 14.3.2 we discussed that the optical flow should meet the brightness constraint equation (14.9):

$$\boldsymbol{f}(\boldsymbol{x}, t) \nabla g(\boldsymbol{x}, t) + g_t(\boldsymbol{x}, t) = 0 \qquad (17.13)$$

and used an approach that minimized the deviation from the optical flow in a least squares sense (Eq. (14.15)). With the L_n norm, we obtain the following similarity term:

$$S(\boldsymbol{f}, \boldsymbol{x}, t) = \|\boldsymbol{f}\,\nabla g + g_t\|_n. \tag{17.14}$$

This equation simply expresses that the continuity equation for the optical flow (Eq. (14.9)) should be satisfied as well as possible in a least squares sense. Note that the similarity now also depends explicitly on time, because the minimization problem is extended from images to space-time images.

From the following example, we will learn that similarity constraints alone are not of much use with the variational approach. We use the motion determination problem with the L_2-norm (least squares). With Eq. (17.14), the Lagrange function depends only on the optical flow $\boldsymbol{f}$. To compute the Euler-Lagrange equations, we only need to consider the partial derivatives of the similarity term Eq. (17.14) with respect to the components of the optical flow, $\partial L/\partial f_i$:

$$L_{f_i} = 2\left(\boldsymbol{f}\,\nabla g + g_t\right) g_{x_i}. \tag{17.15}$$

Inserting Eq. (17.15) into the Euler-Lagrange equation (Eq. (17.9)) yields

$$\left(\boldsymbol{f}\,\nabla g + g_t\right) g_x = 0, \quad \left(\boldsymbol{f}\,\nabla g + g_t\right) g_y = 0, \tag{17.16}$$

or, written as a vector equation,

$$\left(\boldsymbol{f}\,\nabla g + g_t\right) \nabla g = \boldsymbol{0}. \tag{17.17}$$

These equations tell us that the optical flow cannot be determined when the spatial gradient of ∇g is a zero vector. Otherwise, they yield no more constraints than the continuity of the optical flow. This example nicely demonstrates the limitation of local similarity constraints. They only yield *isolated local* solutions without any constraints for the spatial variation of the optical flow. This results from the fact that the formulation of the problem does not include any terms connecting neighboring points. Thus, real progress requires inclusion of global constraints.

Therefore, it is required to add another term to the Lagrange function that also depends on the derivatives of f:

$$\boxed{L(f, \nabla f, \boldsymbol{x}) = S(f, \boldsymbol{x}) + R(f, \nabla f, \boldsymbol{x}).} \tag{17.18}$$

17.2.4 *Global Smoothness Constraints*

One of the most elementary global regularizers is smoothness. For many problems in image processing it makes sense to demand that a quantity to be modeled changes only slowly in space and time. For a segmentation problem this demand means that an object is defined just by the fact that it is a connected region with constant or only slowly changing features. Likewise, the depth of a surface and the velocity field of a moving object are continuous at least at most points.

Therefore, we now seek a suitable regularizer R to add to the Lagrange function to force spatially smooth solutions. Such a term requires spatial partial derivatives of the modeled feature. The simplest term, containing only first-order derivatives, for a scalar feature f in a 2-D image is

$$R\left(f_x, f_y\right) = \alpha^2\left(f_x^2 + f_y^2\right) = \alpha^2|\nabla f|^2. \tag{17.19}$$

For a vector feature $\boldsymbol{f} = [f_1, f_2]^T$

$$R\left(\nabla f_1, \nabla f_2\right) = \alpha^2\left(|\nabla f_1|^2 + |\nabla f_2|^2\right). \tag{17.20}$$

In this additional term the partial derivatives emerge as a sum of squares. This means that we evaluate the *smoothness* term with the same norm (L_2-norm, sum of least squares) as the similarity term. Moreover, in this formulation all partial derivatives are weighted equally. The factor α^2 indicates the relative weight of the smoothness term compared to the similarity term.

The complete least-squares error functional for motion determination including the similarity and smoothing terms is then given by

$$\boxed{L\left(\boldsymbol{f}, \nabla f_1, \nabla f_2, \boldsymbol{x}\right) = \left(\boldsymbol{f}\,\nabla g + g_t\right)^2 + \alpha^2\left(|\nabla f_1|^2 + |\nabla f_2|^2\right).} \tag{17.21}$$

Inserting this Lagrange function into the Euler-Lagrange equation (17.9) yields the following differential equation:

$$\begin{aligned} \left(\nabla g\, \boldsymbol{f} + g_t\right) g_x \quad & -\alpha^2\left((f_1)_{xx} + (f_1)_{yy}\right) = 0 \\ \left(\nabla g\, \boldsymbol{f} + g_t\right) g_y \quad & -\alpha^2\left((f_2)_{xx} + (f_2)_{yy}\right) = 0, \end{aligned} \tag{17.22}$$

or summarized in a vector equation:

$$\boxed{\underbrace{\left(\nabla g\, \boldsymbol{f} + \frac{\partial g}{\partial t}\right)\nabla g}_{\text{similarity term}} - \underbrace{\alpha^2 \Delta \boldsymbol{f}}_{\text{smoothness term}} = \boldsymbol{0}.} \tag{17.23}$$

It is easy to grasp how the optical flow results from this formula. First, imagine that the intensity is changing strongly in a certain direction. The similarity term then becomes dominant over the smoothness term and the velocity will be calculated according to the local optical flow. In contrast, if the intensity change is small, the smoothness term becomes dominant. The local velocity will be calculated in such a manner that it is as close as possible to the velocity in the neighborhood. In other words, the flow vectors are interpolated from surrounding flow vectors.

This process may be illustrated further by an extreme example. Let us consider an object with a constant intensity moving against a black background. Then the similarity term vanishes completely inside the object, while at the border the velocity perpendicular to the border can be calculated just from this term.

This is an old and well-known problem in physics: the problem of how to calculate the potential function (without sinks and sources, $\Delta \boldsymbol{f} = \boldsymbol{0}$) with given boundary conditions at the edge of the object. This equation is known as the *Laplacian equation*. We can immediately conclude the form of the solution in areas where the similarity term is zero. As the second-order derivatives are zero, the first-order spatial derivatives are constant. This leads to a modeled feature f that changes linearly in space.

17.2.5 *Controlling Smoothness*

Having discussed the basic properties of smoothness constraints we now turn to the question of how we can adequately treat spatial and temporal discontinuities with this approach. In a segmentation problem, the modeled feature will be discontinuous at the edge of the object. The same is true for the optical flow. The smoothness constraint as we have formulated it so far does not allow for discontinuities. We applied a global smoothness constraint and thus obtained a globally smooth field. Thus we need to develop methods that allow us to detect and to model discontinuities adequately.

We will first discuss the principal possibilities for varying the minimal problem within the chosen frame. To do so, we rewrite the integral equation for the minimal problem (Eq. (17.6)) using the knowledge about the meaning of the Lagrange function obtained in the last section:

$$\int_\Omega \Big(\underbrace{S(\boldsymbol{f})}_{\text{similarity term}} + \underbrace{R(\boldsymbol{f}_{x_p})}_{\text{smoothness term}} \Big) \, \mathrm{d}^W x \rightarrow \text{Minimum.} \tag{17.24}$$

In order to incorporate discontinuities, two approaches are possible:

1. *Limitation of integration area.* The integration area is one of the ways that the problem of discontinuities in the feature $\boldsymbol{f}$ may be solved. If the integration area includes discontinuities, incorrect values are obtained. Thus algorithms must be found that look for edges in $\boldsymbol{f}$ and, as a consequence, restrict the integration area to the segmented areas. Obviously, this is a difficult iterative procedure. First, the edges in the image itself do not necessarily coincide with the edges in the feature $\boldsymbol{f}$. Second, before calculating the feature field $\boldsymbol{f}$ only sparse information is available so that a partition is not possible.
2. *Modification of smoothness term.* Modification of the smoothness term is another way to solve the discontinuity problem. At points where a discontinuity is suspected, the smoothness constraint may be weakened or may even vanish. This allows discontinuities. Again this is an iterative algorithm. The smoothness term must include a control function that switches off the smoothness constraint in appropriate circumstances. This property is called *controlled smoothness* [198].

In the following, we discuss two approaches that modify the integration area for motion determination. The modification of the smoothing term is discussed in detail in Section 17.3.

Integration along Closed Zero-Crossing Curves. Hildreth [76] used the Laplace filtered image, and limited any further computations to zero crossings. This approach is motivated by the fact that zero crossings mark the gray value edges (Section 12.3), i. e., the features at which we can compute the velocity component normal to the edge. The big advantage of the approach is that the preselection of promising features considerably decreases any computation required.

By selecting the zero crossings, the smoothness constraint is limited to a certain contour line. This seems useful, as a zero crossing most likely belongs to an object but does not cross object boundaries. However, this is not necessarily

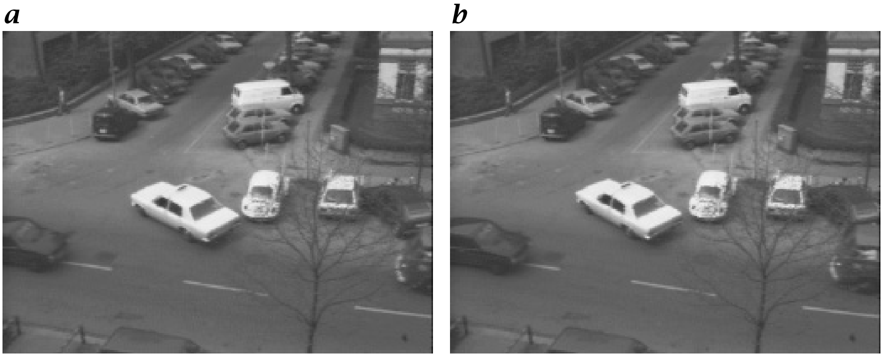

***Figure 17.3:** Two images of a Hamburg taxi. The video images are from the Computer Science Department at Hamburg University and since then have been used as a test sequence for image sequence processing.*

the case. If a zero crossing is contained within an object, the velocity along the contour should show no discontinuities. Selecting a line instead of an area for the smoothness constraint changes the integration region from an area to the line integral along the contour s:

$$\oint \left\{(\bar{\boldsymbol{n}}\boldsymbol{f} - f_{\perp})^2 + \alpha^2 \left[((f_1)_s)^2 + ((f_2)_s)^2\right]\right\} \mathrm{d}\boldsymbol{s} \rightarrow \text{minimum}, \tag{17.25}$$

where $\bar{\boldsymbol{n}}$ is a unit vector normal to the edge and $f_{\perp}$ the velocity normal to the edge.

The derivatives of the velocities are computed in the direction of the edge. The component normal to the edge is given directly by the similarity term, while the velocity term parallel to the edge must be inferred from the smoothness constraint all along the edge. Hildreth [76] computed the solution of the linear equation system resulting from Eq. (17.25) iteratively using the method of conjugate gradients.

Despite its elegance, the edge-oriented method shows significant disadvantages. It is not certain that a zero crossing is contained within an object. Thus we cannot assume that the optical flow field is continuous along the zero crossing. As only edges are used to compute the optical flow field, only one component of the displacement vector can be computed locally. In this way, all features such as either gray value maxima or gray value corners which allow an unambiguous local determination of a displacement vector are disregarded.

Limitation of Integration to Segmented Regions. A region-oriented approach does not omit such points, but still tries to limit the smoothness within objects. Again, *zero crossings* could be used to separate the image into regions or any other basic segmentation technique (Chapter 16). Region-limited smoothness just drops the continuity constraint at the boundaries of the region. The simplest approach to this form of constraint is to limit the integration areas to the different regions and to evaluate them separately.

As expected, a region-limited smoothness constraint results in an optical flow field with discontinuities at the region's boundaries (Fig. 17.4d) which is in clear

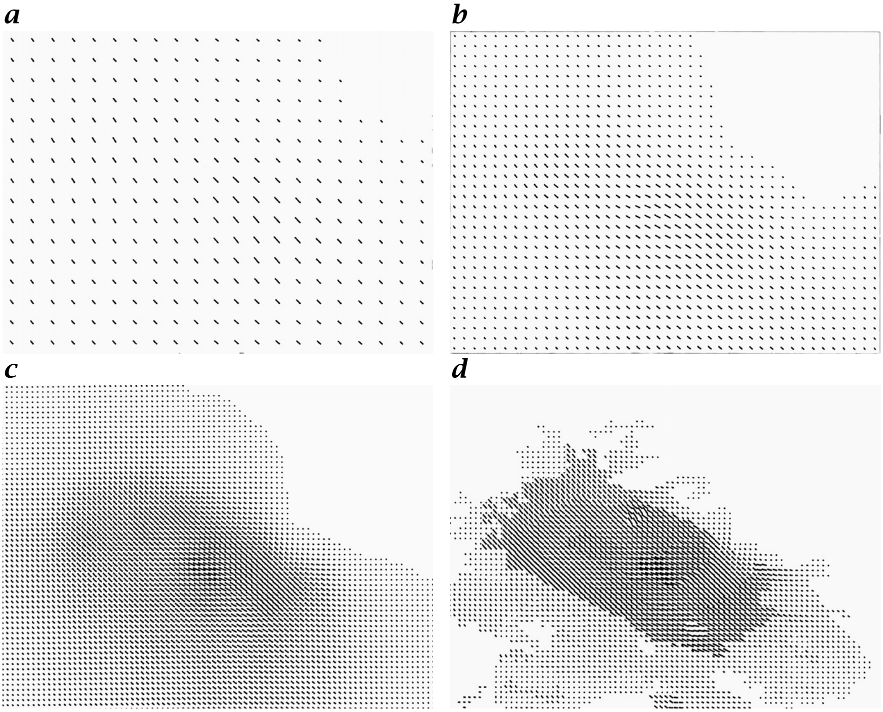

Figure 17.4: *Determination of the DVF in the taxi scene (Fig. 17.3) using the method of the dynamic pyramid:* ***a–c*** *three levels of the optical flow field using a global smoothness constraint;* ***d*** *final result of the optical flow using a region-oriented smoothness constraint; kindly provided by* M. Schmidt *and* J. Dengler, *German Cancer Research Center, Heidelberg.*

contrast to the globally smooth optical flow field in Fig. 17.4c. We immediately recognize the taxi by the boundaries of the optical flow.

We also observe, however, that the car is segmented further into regions with different optical flow field, as shown by the taxi plate on the roof of the car and the back and side windows. The small regions especially show an optical flow field significantly different from that in larger regions. Thus a simple region-limited smoothness constraint does not reflect the fact that there might be separated regions within objects. The optical flow field may well be smooth across these boundaries.

17.3 *Continuous Modeling II: Diffusion*

In this section, we take a new viewpoint of continuous modeling. The least-squares error functional for motion determination Eq. (17.23)

$$\left(\nabla g\, \boldsymbol{f} + \frac{\partial g}{\partial t}\right) \nabla g - \alpha^2 \Delta \boldsymbol{f} = \mathbf{0} \tag{17.26}$$

can be regarded as the stationary solution of a *diffusion-reaction system* with homogeneous diffusion if the constant α^2 is identified with a diffusion coefficient D:

$$\frac{\partial \boldsymbol{f}}{\partial t} = D\Delta \boldsymbol{f} - \left(\nabla g\, \boldsymbol{f} + \frac{\partial g}{\partial t}\right) \nabla g. \tag{17.27}$$

The standard instationary (partial) differential equation for homogeneous diffusion (see Eq. (5.18) in Section 5.3.1) is appended by an additional source term related to the similarity constraint. The source strength is proportional to the deviation from the optical flow constraint. Thus this term tends to shift the values for $\boldsymbol{f}$ to meet the optical flow constraint.

After this introductionary example, we can formulate the relation between a variational error functional and diffusion-reaction systems in a general way. The *Euler-Lagrange equation*

$$\sum_{w=1}^{W} \partial_{x_w} L_{f_{x_w}} - L_f = 0 \tag{17.28}$$

that minimizes the error functional for the scalar spatiotemporal function $f(\boldsymbol{x}), \boldsymbol{x} \in \Omega$

$$\varepsilon(f) = \int_{\Omega} L\left(f, f_{x_w}, \boldsymbol{x}\right) \mathrm{d}x^W \tag{17.29}$$

can be regarded as the steady state of the diffusion-reaction system

$$f_t = \sum_{w=1}^{W} \partial_{x_w} L_{f_{x_w}} - L_f. \tag{17.30}$$

In the following we shall discuss in detail an aspect of modeling, which we have so far only touched in Section 17.2.5, namely the local modification of the smoothness term only in. In the language of a diffusion model this means a locally varying diffusion coefficient in the first-hand term on the right side of Eq. (17.30). From the above discussion we know that to each approach of locally varying diffusion coefficient, a corresponding variational error functional exists that is minimized by the diffusion-reaction system.

In Section 5.3.1 we discussed a homogeneous diffusion process that generated a multiresolution representation of an image, known as the linear *scale space.* If the smoothness constraint is made dependent on local properties of the image content such as the gradient, then the inhomogeneous diffusion process leads to the generation of a nonlinear scale space. With respect to modeling, the interesting point here is that a segmentation can be achieved without a similarity term.

17.3.1 *Inhomogeneous Diffusion*

The simplest approach to a spatially varying smoothing term that takes into account discontinuities is to reduce the diffusion coefficient at the edges. Thus, the diffusion coefficient is made dependent on the strength of the edges as given by the magnitude of the gradient

$$D(f) = D(|\nabla f|^2). \tag{17.31}$$

With a locally varying diffusion constant the diffusion-reaction system becomes

$$f_t = \nabla\left(D(|\nabla f|^2)\nabla f\right) - L_f. \tag{17.32}$$

Note that it is incorrect to write $D(|\nabla f|^2)\,\Delta f$. This can be seen from the derivation of the instationary diffusion equation in Section 5.3.1.

With Eq. (17.32) the regularization term R in the Lagrange function is

$$R = R(|\nabla f|^2), \tag{17.33}$$

where the diffusion coefficient is the derivative of the function R: $D = R'$. This can easily be verified by inserting Eq. (17.33) into Eq. (17.28).

Perona and Malik [151] used the following dependency of the diffusion coefficient on the magnitude of the gradient:

$$D(|\nabla f|) = D_0\frac{\lambda^2}{|\nabla f|^2 + \lambda^2}, \tag{17.34}$$

where λ is an adjustable parameter. For low gradients $|\nabla f| \ll \lambda$, D approaches D_0; for high gradients $|\nabla f| \gg \lambda$, D tends to zero.

As simple and straightforward as this idea appears, it is not without problems. Depending on the functionality of D on ∇f, the diffusion process may become unstable, even resulting in steepening of the edges. A safe way to avoid this problem is to use a regularized gradient obtained from a smoothed version of the image as shown by Weickert [213]. He used

$$D = D_0\left[1 - \exp\left(-\frac{c_m}{(|\nabla(B^R * f)(\boldsymbol{x})|/\lambda)^m}\right)\right]. \tag{17.35}$$

This equation implies that for small magnitudes of the gradient the diffusion coefficient is constant. At a certain threshold of the magnitude of the gradient, the diffusion coefficient quickly decreases towards zero. The higher the exponent m is, the steeper the transition. With the values used by Weickert [213], $m = 4$ and $c_4 = 3.31488$, the diffusion coefficient falls from 1 at $|\nabla f|/\lambda = 1$ to about 0.15 at $|\nabla f|/\lambda = 2$. Note that a regularized gradient has been chosen in Eq. (17.35), because the gradient is not computed from the image $f(\boldsymbol{x})$ directly, but from the image smoothed with a binomial smoothing mask $\mathcal{B}^p$. A properly chosen regularized gradient stabilizes the inhomogeneous smoothing process and avoids instabilities and steepening of the edges.

A simple explicit discretization of inhomogeneous diffusion uses regularized derivative operators as discussed in Section 12.7. In the first step, a gradient image is computed with the vector operator

$$\begin{bmatrix} \mathcal{D}_1 \\ \mathcal{D}_2 \end{bmatrix}. \tag{17.36}$$

In the second step, the gradient image is multiplied pointwise by the control operator $\boldsymbol{S}$ that computes the diffusion coefficient according to Eq. (17.34) or Eq. (17.35):

$$\begin{bmatrix} \mathcal{S}\cdot\mathcal{D}_1 \\ \mathcal{S}\cdot\mathcal{D}_2 \end{bmatrix}. \tag{17.37}$$

The control image $\mathcal{S}$ is one in constant regions and drops towards small values at the edges. In the third step, the gradient operator is applied a second time

$$[\mathcal{D}_1, \mathcal{D}_2] \begin{bmatrix} \mathcal{S} \cdot \mathcal{D}_1 \\ \mathcal{S} \cdot \mathcal{D}_2 \end{bmatrix} = \mathcal{D}_1(\mathcal{S} \cdot \mathcal{D}_1) + \mathcal{D}_2(\mathcal{S} \cdot \mathcal{D}_2). \tag{17.38}$$

Weickert [213] used a more sophisticated implicit solution scheme. However, the scheme is computationally more expensive and less isotropic than the explicit scheme in Eq. (17.38) if gradient operators are used that are optimized for isotropy as discussed in Section 12.7.5.

An even simpler but only approximate implementation of inhomogenous diffusion controls binomial smoothing using the operator

$$\mathcal{I} + \mathcal{S} \cdot (\mathcal{B} - \mathcal{I}). \tag{17.39}$$

The operator $\mathcal{S}$ computes a control image with values between zero and one.

Figure 17.5 shows the application of inhomogeneous diffusion for segmentation of noisy images. The test image contains a triangle and a rectangle. Standard smoothing significantly suppresses the noise but results in a significant blurring of the edges (Fig. 17.5b). Inhomogenous diffusion does not lead to a blurring of the edges and still results in a perfect segmentation of the square and the triangle (Fig. 17.5c). The only disadvantage is that the edges themselves remain noisy because smoothing is suppressed there.

17.3.2 *Anisotropic Diffusion*

As we have seen in the example discussed at the end of the last section, inhomogeneous diffusion has the significant disadvantage that it stops diffusion completely and in all directions at edges, leaving the edges noisy. However, edges are only blurred by diffusion perpendicular to them; diffusion parallel to them is even advantageous as it stabilizes the edges.

An approach that makes diffusion independent of the direction of edges is known as anisotropic diffusion. With this approach, the flux is no longer parallel to the gradient. Therefore, the diffusion can no longer be described by a scalar diffusion coefficient as in Eq. (5.15). Now, a *diffusion tensor* is required:

$$\boldsymbol{j} = -\boldsymbol{D}\nabla f = -\begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}. \tag{17.40}$$

With a diffusion tensor the diffusion-reaction system becomes

$$f_t = \nabla \left(\boldsymbol{D}(\nabla f \nabla f^T)\, \nabla f \right) - L_f \tag{17.41}$$

and the corresponding regularizer in the Lagrange function is

$$R = \operatorname{trace} \boldsymbol{R} \left(\nabla f \nabla f^T \right), \tag{17.42}$$

with $\boldsymbol{D} = \boldsymbol{R}'$. The properties of the diffusion tensor can best be seen if the symmetric tensor is brought into its principal-axis system by a rotation of the coordinate system. Then, Eq. (17.40) reduces to

$$\boldsymbol{j}' = -\begin{bmatrix} D_1' & 0 \\ 0 & D_2' \end{bmatrix} \begin{bmatrix} f_1' \\ f_2' \end{bmatrix} = -\begin{bmatrix} D_1' f_1' \\ D_2' f_2' \end{bmatrix}. \tag{17.43}$$

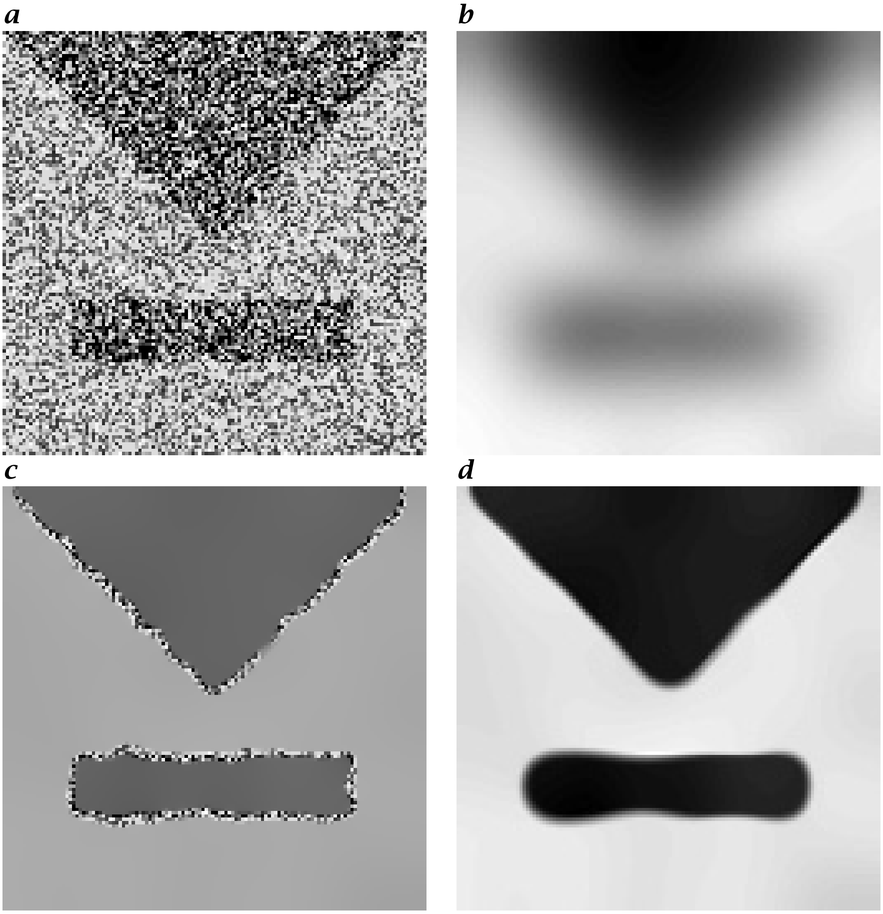

Figure 17.5: ***a*** *Original, smoothed by* ***b*** *linear diffusion,* ***c*** *inhomogenous but isotropic diffusion, and* ***d*** *anisotropic diffusion. From Weickert [213].*

The diffusion in the two directions of the axes is now decoupled. The two coefficients on the diagonal, D'_1 and D'_2, are the *eigenvalues* of the diffusion tensor. By analogy to isotropic diffusion, the general solution for homogeneous anisotropic diffusion can be written as

$$f(\boldsymbol{x},t) = \frac{1}{2\pi\sigma'_1(t)\sigma'_2(t)} \exp\left(-\frac{x'^2}{2\sigma'_1(t)}\right) * \exp\left(-\frac{y'^2}{2\sigma'_2(t)}\right) * f(\boldsymbol{x},0) \quad (17.44)$$

in the spatial domain with $\sigma'_1(t) = \sqrt{2D'_1 t}$ and $\sigma'_2(t) = \sqrt{2D'_2 t}$.

This means that anisotropic diffusion is equivalent to cascaded convolution with two 1-D Gaussian convolution kernels that are steered in the directions of the principal axes of the diffusion tensor. If one of the two eigenvalues of the diffusion tensor is significantly larger than the other, diffusion occurs only in the direction of the corresponding eigenvector. Thus the gray values are smoothed only in this direction. The spatial widening is — as for any diffusion process — proportional to the square root of the diffusion constant (Eq. (5.23)).

Using this feature of anisotropic diffusion, it is easy to design a diffusion process that predominantly smoothes only along edges but not perpendicularly to the edges. With the following approach only smoothing across edges is hindered [213]:

$$\begin{aligned} D_1' &= 1 - \exp\left(-\frac{c_m}{(|\nabla(B^r * f)(\boldsymbol{x})|/\lambda)^m}\right) \\ D_2' &= 1. \end{aligned} \qquad (17.45)$$

As shown by Scharr and Weickert [177], an efficient and accurate explicit implementation of anisotropic diffusion is again possible with regularized first-order differential derivative optimized for minimum anisotropy:

$$\begin{aligned} & [\mathcal{D}_1, \mathcal{D}_2] \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{12} & \mathcal{S}_{22} \end{bmatrix} \begin{bmatrix} \mathcal{D}_1 \\ \mathcal{D}_2 \end{bmatrix} = \\ & \mathcal{D}_1(\mathcal{S}_{11} \cdot \mathcal{D}_1 + \mathcal{S}_{12} \cdot \mathcal{D}_2) + \mathcal{D}_2(\mathcal{S}_{12} \cdot \mathcal{D}_1 + \mathcal{S}_{22} \cdot \mathcal{D}_2). \end{aligned} \qquad (17.46)$$

with

$$\begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ \mathcal{S}_{12} & \mathcal{S}_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \mathcal{S}_1' & 0 \\ 0 & \mathcal{S}_2' \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

The $\mathcal{S}_{pq}$, $\mathcal{S}_1'$ und $\mathcal{S}_2'$ are control images with values between zero and one that steer the diffusion into the direction parallel to edges at each point of the image. $\mathcal{S}_1'$ and $\mathcal{S}_2'$ are directly computed from Eq. (17.45), the direction of the edges, the angle θ can be obtained, e. g., from the structure tensor (Section 13.3).

Application of anisotropic diffusion shows that now — in contrast to inhomogeneous diffusion — the edges are also smoothed (Fig. 17.5d). The smoothing along edges has the disadvantage, however, that the corners of the edges are now blurred as with linear diffusion. This did not happen with inhomogeneous diffusion (Fig. 17.5c).

17.4 *Discrete Modeling: Inverse Problems*

In the second part of this chapter, we turn to discrete modeling. Discrete modeling can, of course, be derived, by directly discretizing the partial differential equations resulting from the variational approach. Actually, we have already done this in Section 17.3 by the iterative discrete schemes for inhomogeneous and anisotropic diffusion.

However, by developing discrete modeling independently, we gain further insight. Again, we take another point of view of modeling and now regard it as a *linear discrete inverse problem*. As an introduction, we start with the familiar problem of linear regression and then develop the theory of discrete inverse modeling.

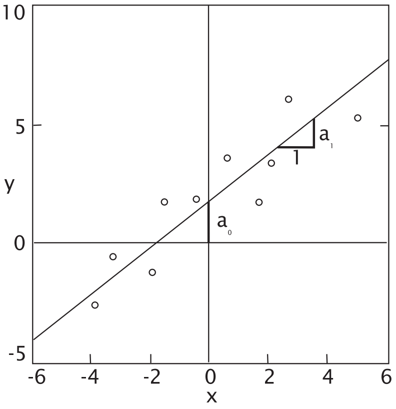

Figure 17.6: Illustration of least-squares linear regression.

17.4.1 A Simple Example: Linear Regression

The fit of a straight line to a set of experimental data points x, y is a simple example of a discrete inverse problem. As illustrated in Fig. 17.6, the quantity y is measured as a function of a parameter x. In this case, our model is a straight line with two parameters, the offset a_0 and the slope a_1: $y = a_0 + a_1 x$. With a set of Q data points $[x_q, y_q]^T$ we end up with the linear equation system

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_Q \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_Q \end{bmatrix} \tag{17.47}$$

which can be abbreviated by

$$\boldsymbol{M}\boldsymbol{p} = \boldsymbol{d}. \tag{17.48}$$

The $Q \times 2$ matrix $\boldsymbol{M}$ is denoted as the *model matrix* or *design matrix*. This matrix reflects both the type of the model (here a linear regression) and the chosen independent measuring points x_q. The *model* or *parameter vector* $\boldsymbol{p}$ contains the parameters of the model to be estimated and the *data vector* $\boldsymbol{d}$ the measured data x_q.

If we have only two data points which do not coincide, $x_1 \neq x_2$, we get an exact solution of the linear equation system. If more than two data points are available, we have more equations than unknowns. We say that the equation system is an *overdetermined inverse problem*. In this case, it is generally no longer possible to obtain an exact solution. We can only compute an estimate of the model parameters $\boldsymbol{p}_{\text{est}}$ in the sense that the deviation of the data $\boldsymbol{d}$ from the data predicted with the model $\boldsymbol{d}_{\text{pre}} = \boldsymbol{M}\boldsymbol{p}_{\text{est}}$ is minimal. This deviation can be expressed by an *error vector* $\boldsymbol{e}$:

$$\boldsymbol{e} = \boldsymbol{d} - \boldsymbol{d}_{\text{pre}} = \boldsymbol{d} - \boldsymbol{M}\boldsymbol{p}_{\text{est}}. \tag{17.49}$$

17.4.2 *Error Norms*

In order to minimize the error vector we need a suitable measure. We may use *norms*, which we discussed when using inner product vector spaces in Section 2.3.1. Generally, the L_n norm of the Q-dimensional vector $\boldsymbol{e}$ is defined as

$$\|\boldsymbol{e}\|_n = \left(\sum_{q=1}^{Q} |e_q|^n\right)^{1/n}. \tag{17.50}$$

A special case is the L_∞ norm

$$\|\boldsymbol{e}\|_\infty = \max_n |e_q|. \tag{17.51}$$

The L_2 norm is more commonly used; it is the root of the sum of the squared deviations of the error vector elements

$$\|\boldsymbol{e}\|_2 = \left(\sum_{q=1}^{Q} (d_q - d_{\text{pre},q})^2\right)^{1/2}. \tag{17.52}$$

Higher norms rate higher deviations with a more significant weighting. The statistics of the data points determines which norm is to be taken. If the measured data points y_q have a *normal density* (Section 3.4.2), the L_2 norm must be used [136].

17.4.3 *Least Squares Solution*

The overdetermined linear inverse problem is solved with a minimum L_2 norm of the error vector by

$$\boldsymbol{p}_{\text{est}} = \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T\boldsymbol{d}. \quad \text{with} \quad \|\boldsymbol{e}\|_2^2 = \|\boldsymbol{d} - \boldsymbol{M}\boldsymbol{p}_{\text{est}}\| \rightarrow \text{minimum}. \tag{17.53}$$

This solution can be made plausible by the following sequence of operations:

$$\begin{array}{lcl|l} \boldsymbol{M}\boldsymbol{p}_{\text{est}} & = & \boldsymbol{d} & \boldsymbol{M}^T \\ \boldsymbol{M}^T\boldsymbol{M}\boldsymbol{p}_{\text{est}} & = & \boldsymbol{M}^T\boldsymbol{d} & \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1} \\ \boldsymbol{p}_{\text{est}} & = & \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T\boldsymbol{d} & \end{array} \tag{17.54}$$

provided that the inverse of $\boldsymbol{M}^T\boldsymbol{M}$ exists.

In the rest of this section we provide a derivation of the solution of the overdetermined discrete linear inverse problem (Eq. (17.48)) that minimizes the L_2 norm of the error vector. Therefore, we compute the solution explicitly by minimizing the L_2 norm of the error vector $\boldsymbol{e}$ (Eq. (17.49)):

$$\|\boldsymbol{e}\|_2^2 = \sum_{q'=1}^{Q}\left(d_{q'} - \sum_{p'=1}^{P} m_{q'p'}p_{p'}\right)\left(d_{q'} - \sum_{p''=1}^{P} m_{q'p''}p_{p''}\right).$$

Factorizing the sum and interchanging the two summations yields

$$\begin{aligned} \|\boldsymbol{e}\|_2^2 &= \underbrace{\sum_{p'=1}^{P}\sum_{p''=1}^{P} p_{p'}p_{p''}\sum_{q=1}^{Q} m_{qp'}m_{qp''}}_{A} \\ &- \underbrace{2\sum_{p'=1}^{P} p_{p'}\sum_{q=1}^{Q} m_{qp'}d_q}_{B} + \sum_{q=1}^{Q} d_q d_q \end{aligned} \tag{17.55}$$

We find a minimum for this expression by computing the partial derivatives with respect to the parameters p_k that are to be optimized. Only the expressions A and B in Eq. (17.55) depend on p_k:

$$\begin{aligned} \frac{\partial A}{\partial p_k} &= \sum_{p'=1}^{P}\sum_{p''=1}^{P}\left(\delta_{k-p''}p_{p'} + \delta_{k-p'}p_{p''}\right)\sum_{q'=1}^{Q} m_{q'p'}m_{q'p''} \\ &= \sum_{p'=1}^{P} p_{p'}\sum_{q'=1}^{Q} m_{q'p'}m_{q'k} + \sum_{p''=1}^{P} p_{p''}\sum_{q'=1}^{Q} m_{q'k}m_{q'p''} \\ &= 2\sum_{p'=1}^{P} p_{p'}\sum_{q'=1}^{Q} m_{q'p'}m_{q'k}, \\ \frac{\partial B}{\partial p_k} &= 2\sum_{q'=1}^{Q} m_{q'k}d_{q'}. \end{aligned}$$

We add both derivatives and set them equal to zero:

$$\frac{\partial\|\boldsymbol{e}\|_2^2}{\partial p_k} = 2\sum_{p'=1}^{P} p_{p'}\sum_{q'=1}^{Q} m_{q'k}m_{q'p'} - 2\sum_{q'=1}^{Q} m_{q'k}d_{q'} = 0.$$

In order to express the sums as matrix-matrix and matrix-vector multiplications, we substitute the matrix $\boldsymbol{M}$ at two places by its transpose $\boldsymbol{M}^T$:

$$\sum_{p'=1}^{P} p_{p'}\sum_{q'=1}^{Q} m^T_{kq'}m_{q'p'} - \sum_{q'=1}^{Q} m^T_{kq'}d_{q'} = 0$$

and finally obtain the matrix equation

$$\underbrace{\underbrace{\underbrace{\boldsymbol{M}^T}_{P\times Q}\underbrace{\boldsymbol{M}}_{Q\times P}}_{P\times P}\underbrace{\boldsymbol{p}_{\text{est}}}_{P}}_{P} = \underbrace{\underbrace{\boldsymbol{M}^T}_{P\times Q}\underbrace{\boldsymbol{d}}_{Q}}_{P}. \tag{17.56}$$

This equation can be solved if the quadratic and symmetric $P \times P$ matrix $\boldsymbol{M}^T\boldsymbol{M}$ is invertible. Then

$$\boldsymbol{p}_{\text{est}} = \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T\boldsymbol{d}. \tag{17.57}$$

The matrix $(\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T$ is known as the *generalized inverse* $\boldsymbol{M}^{-g}$ of $\boldsymbol{M}$.

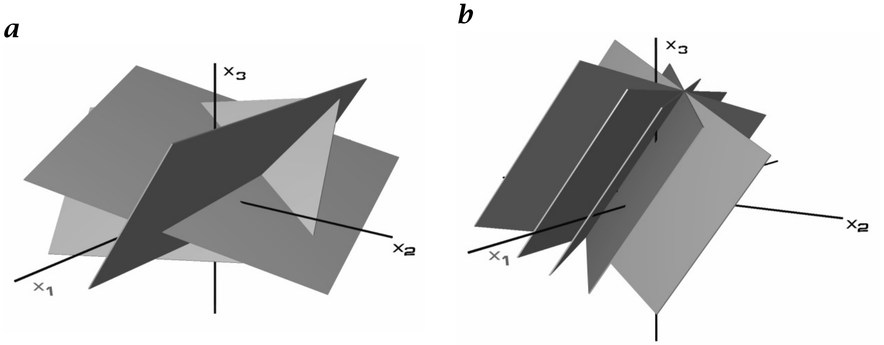

Figure 17.7: *Geometric illustration of the solution of a linear equation system with three unknowns using the Hough transform:* ***a*** *exact soluble equation system;* ***b*** *overdetermined equation system with a non-unique solution.*

17.4.4 Geometric Illustration

Before we study methods for solving huge linear equation systems, it is helpful to illustrate linear equation systems geometrically. The P model parameters $\boldsymbol{p}$ span a P"=dimensional vector space. This space can be regarded as the space of all possible solutions of an inverse problem with P model parameters. Now, we ask ourselves what it means to have one data point d_q. According to Eq. (17.48), one data point results in one linear equation involving all model parameters $\boldsymbol{p}$

$$\sum_{k=p'}^{P} m_{qp'} p_{p'} = d_q \quad \text{or} \quad \boldsymbol{m}_q \boldsymbol{p} = d_q. \tag{17.58}$$

This equation can be regarded as the scalar product of a row q of the model matrix $\boldsymbol{m}_q$ with the model vector $\boldsymbol{p}$. In the model space, this equation constitutes a $P - 1$-dimensional *hyperplane* of all vectors $\boldsymbol{p}$ which has a normal vector $\boldsymbol{m}_q$ and a distance d_q from the origin of the *model space*. Thus, the linear equation establishes a one-to-one correspondence between a data point in the *data space* and a $(P-1)$-dimensional hyperplane in the model space. This mapping of data points into the model space is called the *Hough transform*, which we introduced in Section 16.5.2. Each data point reduces the space of possible solutions to a $(P - 1)$-dimensional hyperplane in the model space.

Figure 17.7a illustrates the solution of a linear equation system with three unknowns. With three equations, three planes meet at a single point, provided that the corresponding 3×3 model matrix is invertible. Even in an overdetermined case, the solution needs not necessarily be unique. Figure 17.7b shows a case of five planes intersecting at a line. Then, the solution is not unique, but only restricted to a line. If this line is oriented along one of the axes, the corresponding model parameter may take any value; the two other model parameters, however, are fixed.

In case of an arbitrarily oriented line, things are more complex. Then, the parameter combinations normal to the line are fixed, but the parameter combination represented by a vector in the direction of the line is not. Using the *singular* *value decomposition* [61, 158], we can solve singular linear equation systems and separate the solvable from the unsolvable parameter combinations.

An overdetermined linear equation system that has no unique solution is not just a mathematical curiosity. It is rather a common problem in image processing. We have encountered it already, for example in motion determination with the *aperture problem* (Section 14.3.2).

17.4.5 Error of Model Parameters

An overdetermined linear equation system that has been solved by minimizing the L_2 norm allows an analysis of the errors. We can study not only the deviations between model and data but also the errors of the etsimated model parameter vector $\boldsymbol{p}_{\text{est}}$.

The mean deviation between the measured and predicted data points is directly related to the norm of the error vector. The *variance* is

$$\sigma^2 = \frac{1}{Q-P}\|\boldsymbol{e}\|^2 = \frac{1}{Q-P}\|\boldsymbol{d} - \boldsymbol{M}\boldsymbol{p}_{\text{est}}\|_2^2. \qquad (17.59)$$

In order not to introduce a bias in the estimate of the variance, we divide the norm by the *degree of freedom* $Q - P$ and not by Q.

According to Eq. (17.57), the estimated parameter vector $\boldsymbol{p}_{\text{est}}$ is a linear combination of the data vector $\boldsymbol{d}$. Therefore we can apply the *error propagation* law (Eq. (3.27)) derived in Section 3.3.3. The *covariance matrix* (for a definition see Eq. (3.19)) of the estimated parameter vector $\boldsymbol{p}_{\text{est}}$ using $(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T$ is given by

$$\operatorname{cov}(\boldsymbol{p}_{\text{est}}) = \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T \operatorname{cov}(\boldsymbol{d})\,\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}. \qquad (17.60)$$

If the individual elements in the data vector $\boldsymbol{d}$ are uncorrelated and have the same variance σ^2, i.e., $\operatorname{cov}(\boldsymbol{d}) = \sigma^2\boldsymbol{I}$, Eq. (17.60) reduces to

$$\operatorname{cov}(\boldsymbol{p}_{\text{est}}) = \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\sigma^2. \qquad (17.61)$$

In this case, $(\boldsymbol{M}^T\boldsymbol{M})^{-1}$ is — except for the factor σ^2 — directly the covariance matrix of the model parameters. This means that the diagonal elements contain the variances of the model parameters.

17.4.6 Regularization

So far, the error functional (Eq. (17.52)) only contains a similarity constraint but no regularization or smoothing constraint. For many discrete inverse problems — such as the linear regression discussed in Section 17.4.1 — a regularization of the parameters makes no sense. If the parameters to be estimated are, however, the elements of a time series or the pixels of an image, a smoothness constraint makes sense. A suitable smoothness parameter could then be the norm of the time series or image convolved by a derivative filter:

$$\|\boldsymbol{r}\|_2 = \|\boldsymbol{h} * \boldsymbol{p}\|_2^2. \qquad (17.62)$$

In the language of matrix algebra, convolution can be expressed by a vector matrix multiplication:

$$\|\boldsymbol{r}\|_2 = \|\boldsymbol{H}\boldsymbol{p}\|_2^2 . \tag{17.63}$$

Because of the convolution operation, the matrix $\boldsymbol{H}$ has a special form. Only the coefficients around the diagonal are nonzero and all values in diagonal direction are the same.

As an example, we discuss the same smoothness criterion that we used also in the variational approach (Section 17.2.4), the first derivative. It can be approximated, for instance, by convolution with a forward difference filter that results into the matrix

$$\boldsymbol{H} = \begin{bmatrix} -1 & 1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 1 & 0 & \ldots & 0 \\ 0 & 0 & -1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \end{bmatrix} . \tag{17.64}$$

Minimizing the combined error functional using the L_2 norm:

$$\boxed{\|\boldsymbol{e}\|_2^2 = \underbrace{\|\boldsymbol{d} - \boldsymbol{M}\boldsymbol{p}\|_2^2}_{\text{similarity}} + \alpha^2 \underbrace{\|\boldsymbol{H}\boldsymbol{p}\|_2^2}_{\text{smoothness}}} \tag{17.65}$$

results in the following least-squares solution [136]:

$$\boxed{\boldsymbol{p}_{\text{est}} = \left(\boldsymbol{M}^T\boldsymbol{M} + \alpha^2\boldsymbol{H}^T\boldsymbol{H}\right)^{-1} \boldsymbol{M}^T\boldsymbol{d}.} \tag{17.66}$$

The structure of the solution is similar to the least-squares solution in Eq. (17.53). The smoothness term just causes the additional term $\alpha^2\boldsymbol{H}^T\boldsymbol{H}$.

In the next section, we learn how to map an image to a vector, so that we can apply discrete inverse problems also to images.

17.4.7 *Algebraic Tomographic Reconstruction*

In this section we discuss an example of a discrete inverse problem that includes image data: reconstruction from projections (Section 8.6). In order to apply the discrete inverse theory as discussed so far, the image data must be mapped onto a vector, the *image vector*. This mapping is easily performed by renumbering the pixels of the image matrix row by row (Fig. 17.8). In this way, an $M \times N$ image matrix is transformed into a column vector with the dimension $P = M \times N$:

$$\boldsymbol{p} = \left[m_1, m_2, \ldots, m_p, \ldots, m_P\right]^T . \tag{17.67}$$

Now we take a single projection beam that crosses the image matrix (Fig. 17.8). Then we can attribute a weighting factor to each pixel of the image vector that represents the contribution of the pixel to the projection beam. We can combine these factors in a Q-dimensional vector $\boldsymbol{g}_q$:

$$\boldsymbol{g}_q = \left[g_{q,1}, g_{q,2}, \ldots, g_{q,p}, \ldots, g_{Q,P}\right]^T . \tag{17.68}$$

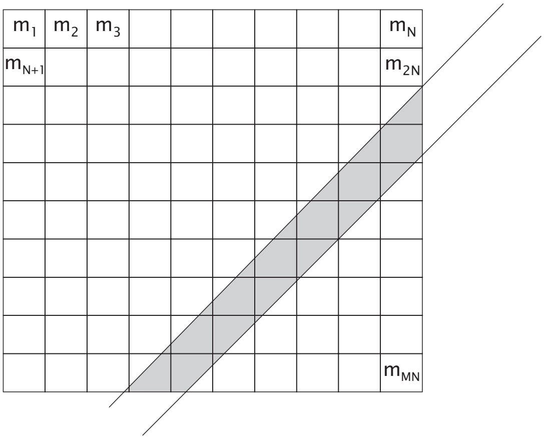

Figure 17.8: *Illustration of algebraic reconstruction from projections: a projection beam d_k crosses the image matrix. All the pixels met by the beam contribute to the projection.*

The total emission or absorption along the qth projection beam d_q can then be expressed as the scalar product of the two vectors $\boldsymbol{g}_q$ and $\boldsymbol{p}$:

$$d_q = \sum_{p=1}^{P} g_{q,p} m_p = \boldsymbol{g}_q \boldsymbol{p}. \qquad (17.69)$$

If Q projection beams cross the image matrix, we obtain a linear equation system of Q equations and P unknowns:

$$\underbrace{\boldsymbol{d}}_{Q} = \underbrace{\boldsymbol{M}}_{Q\times P} \underbrace{\boldsymbol{p}}_{P}. \qquad (17.70)$$

The *data vector* $\boldsymbol{d}$ contains the measured projections and the *parameter vector* $\boldsymbol{p}$ contains the pixel values of the image matrix that are to be reconstructed. The *design matrix* $\boldsymbol{M}$ gives the relationship between these two vectors by describing how in a specific set up the projection beams cross the image matrix. With appropriate weighting factors, we can take into direct account the limited detector resolution and the size of the radiation source.

Algebraic tomographic reconstruction is a general and flexible method. In contrast to the filtered backprojection technique (Section 8.6.3) it is not limited to parallel projection. The beams can cross the image matrix in any manner and can even be curved. In addition, we obtain an estimate of the errors of the reconstruction.

However, algebraic reconstruction involves solving huge linear equation systems. At this point, it is helpful to illustrate the enormous size of these equation systems. In a typical problem, the model vector includes all pixels of an image. Even with moderate resolution, e. g., 256 × 256 pixels, the inverse of a 65536 × 65536 matrix would have to be computed. This matrix contains about $4 \cdot 10^9$ points and does not fit into the memory of any but the most powerful computers. Thus alternative solution techniques are required.

17.4.8 Further Examples of Inverse Problems

Problems of this kind are very common in the analysis of experimental data in natural sciences. Experimentalists look at a discrete inverse problem in the following way. They perform an experiment from which they gain a set of measuring results, and they combine them in a Q-dimensional data vector $\boldsymbol{d}$. These data are compared with a model of the observed process. The parameters of this model are given by a P-dimensional model vector $\boldsymbol{p}$. Now we assume that the relationship between the model and the data vector can be described as linear. It can then be expressed by a model matrix $\boldsymbol{M}$ and we obtain Eq. (17.70).

For image processing, inverse problems are also common. They do not only include the complete list of problems discussed in the introduction of this chapter (Section 17.1.1) but also optimization of filters. In this book, least-squares optimized filters for interpolation (Section 10.6.2) and edge detection (Sections 12.6 and 12.7.5) are discussed.

17.5 *Inverse Filtering*

Now we study a class of inverse problems that is common in signal processing and show the way to fast iterative solutions of huge inverse problems.

17.5.1 *Image Restoration*

No image formation system is perfect because of inherent physical limitations. Therefore, images are not identical to their original. As scientific applications always push the limits, there is a need to correct for limitations in the sharpness of images. Humans also make errors in operating imaging systems. Images blurred by a misadjustment in the focus, smeared by the motion of objects or the camera or a mechanically unstable optical system, or degraded by faulty or misused optical systems are more common than we may think. A famous recent example was the flaw in the optics of the Hubble space telescope where an error in the test procedures for the main mirror resulted in a significant residual aberration of the telescope. The correction of known and unknown image degradation is called *restoration*.

The question arises whether, and if so, to what extent, the effects of degradation can be reversed. It is obvious, of course, that information that is no longer present at all in the degraded image cannot be retrieved. To make this point clear, let us assume the extreme case that only the mean gray value of an image is retained. Then, it will not be possible by any means to reconstruct its content. However, images contain a lot of redundant information. Thus, we can hope that a distortion only partially removes the information of interest even if we can no longer “see” it directly.

In Sections 7.6 and 9.2.1, we saw that generally any optical system including digitization can be regarded as a *linear shift-invariant system* and, thus, described to a good approximation by a *point spread function* and a *transfer function*.

The first task is to determine and describe the image degradation as accurately as possible. This can be done by analyzing the image formation system either theoretically or experimentally by using some suitable test images. If this is not possible, the degraded image remains the only source of information.

17.5.2 *Survey of Image Distortions*

Given the enormous variety of ways to form images (Chapter 7), there are many reasons for image degradation. Imperfections of the optical system, known as lens aberrations, limit the sharpness of images. However, even with a perfect optical system, the sharpness is limited by diffraction of electromagnetic waves at the aperture stop of the lens. While these types of degradation are an inherent property of a given optical system, blurring by *defocusing* is a common misadjustment that limits the sharpness in images. Further reasons for blurring in images are unwanted motions and vibrations of the camera system during the exposure time. Especially systems with a narrow field of view (telelenses) are very sensitive to this kind of image degradation. Blurring can also occur when objects move more than a pixel at the image plane during the exposure time.

Defocusing and *lens aberrations* are discussed together in this section as they are directly related to the optical system. The effect of blurring or aberration is expressed by the *point spread function* $h(\boldsymbol{x})$ and the *optical transfer function* (*OTF*), $\hat{h}(\boldsymbol{k})$; see Section 7.6. Thus, the relation between object $g(\boldsymbol{x})$ and image $g'(\boldsymbol{x})$ is in the spatial and Fourier domain

$$g'(\boldsymbol{x}) = (h * g)(\boldsymbol{x}) \quad \circ\!\!-\!\!\!-\!\!\bullet \quad \hat{g}'(\boldsymbol{k}) = \hat{h}(\boldsymbol{k})\hat{g}(\boldsymbol{k}). \tag{17.71}$$

Lens aberrations are generally more difficult to handle. Most aberrations increase strongly with distance from the optical axis and are, thus, not shift invariant and cannot be described with a position-independent PSF. However, the aberrations change only slowly and continuously with the position in the image. As long as the resulting blurring is limited to an area in which we can consider the aberration to be constant, we can still treat them with the theory of linear shift-invariant systems. The only difference is that the PSF and OTF vary gradually with position.

If defocusing is the dominant blurring effect, the PSF has the shape of the aperture stop. As most aperture stops can be approximated by a circle, the function is a disk. The Fourier transform of a disk with radius r is a Bessel function of the form (≻ R5):

$$\frac{1}{\pi r^2}\Pi\left(\frac{|\boldsymbol{x}|}{2r}\right) \circ\!\!-\!\!\!-\!\!\bullet \frac{J_1(2\pi|\boldsymbol{k}|r)}{\pi|\boldsymbol{k}|r}. \tag{17.72}$$

This Bessel function, as shown in Fig. 17.9a, has a series of zeroes and, thus, completely eliminates certain wave numbers. This effect can be observed in Fig. 17.9b, which shows a defocused image of the ring test pattern.

While blurring by defocusing and lens aberrations tend to be isotropic, blurring effects by motion are one-dimensional, as shown in Fig. 17.10b. In the simplest case, motion is constant during the exposure. Then, the PSF of motion blur is a one-dimensional box function. Without loss of generality, we first assume that the direction of motion is along the x axis. Then (≻ R4, ≻ R5),

$$h_{Bl}(x) = \frac{1}{u\Delta t}\Pi\left(\frac{x}{u\Delta t}\right) \circ\!\!-\!\!\!-\!\!\bullet \hat{h}_{Bl}(k) = \operatorname{sinc}(ku\Delta t), \tag{17.73}$$

where u is the magnitude of the velocity and Δt the exposure time. The blur length is $\Delta x = u\Delta t$.

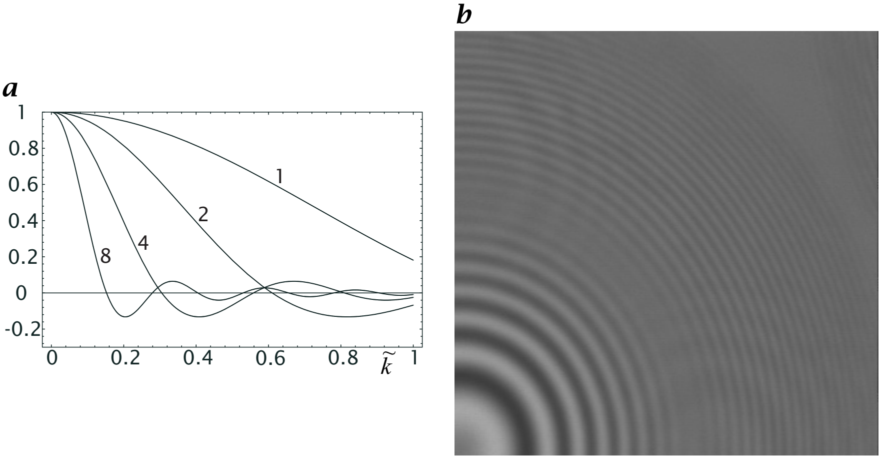

Figure 17.9: *__a__ Transfer functions for disk-shaped blurring. The parameters for the different curves are the radius of the blur disk; __b__ defocused image of the ring test pattern.*

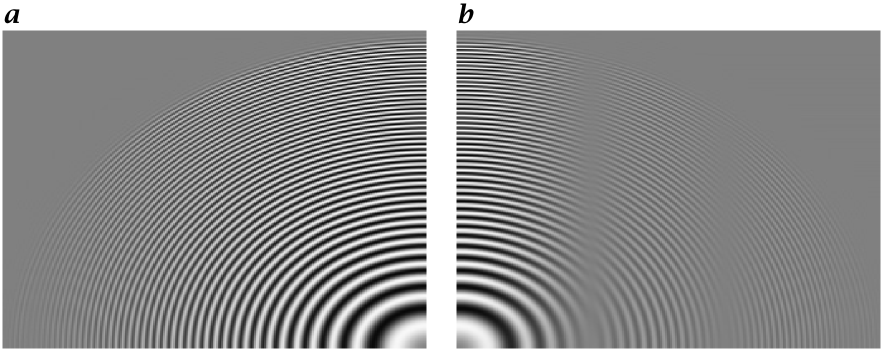

Figure 17.10: *Simulation of blurring by motion using the ring test pattern: __a__ small and __b__ large velocity blurring in horizontal direction.*

If the velocity $\boldsymbol{u}$ is oriented in another direction, Eq. (17.73) can be generalized to

$$h_{Bl}(\boldsymbol{x}) = \frac{1}{|\boldsymbol{u}|\Delta t}\Pi\left(\frac{\boldsymbol{x}\bar{\boldsymbol{u}}}{|\boldsymbol{u}|\Delta t}\right)\delta(\boldsymbol{u}\boldsymbol{x}) \circ\!\!-\!\!\!-\!\!\bullet\ \hat{h}_{Bl}(\boldsymbol{k}) = \operatorname{sinc}(\boldsymbol{k}\boldsymbol{u}\Delta t), \qquad (17.74)$$

where $\bar{\boldsymbol{u}} = \boldsymbol{u}/|\boldsymbol{u}|$ is a unit vector in the direction of the motion blur.

17.5.3 Deconvolution

Common to defocusing, motion blur, and 3-D imaging by such techniques as focus series or confocal microscopy (Section 8.2.4) is that the object function $g(\boldsymbol{x})$ is convolved by a point spread function. Therefore, the principal procedure for reconstructing or restoring the object function is the same. Essentially, it is a *deconvolution* or an *inverse filtering* as the effect of the convolution by the PSF is to be inverted. Given the simple relations in Eq. (17.71), inverse filtering is in principle an easy procedure. The effect of the convolution operator $\mathcal{H}$ is reversed by the application of the inverse operator $\mathcal{H}^{-1}$. In the Fourier space we can write:

$$\hat{\boldsymbol{G}}_R = \frac{\hat{\boldsymbol{G}}'}{\hat{\boldsymbol{H}}'} = \hat{\boldsymbol{H}}^{-1} \cdot \hat{\boldsymbol{G}}'. \tag{17.75}$$

The reconstructed image $\boldsymbol{G}_R$ is then given by applying the inverse Fourier transform:

$$\boldsymbol{G}_R = \mathcal{F}^{-1}\hat{\boldsymbol{H}}^{-1} \cdot \mathcal{F}\hat{\boldsymbol{G}}'. \tag{17.76}$$

The reconstruction procedure is as follows. The Fourier transformed image, $\mathcal{F}\boldsymbol{G}'$, is multiplied by the inverse of the OTF, $\hat{\boldsymbol{H}}^{-1}$, and then transformed back to the spatial domain. The inverse filtering can also be performed in the spatial domain by convolution with a mask that is given by the inverse Fourier transform of the inverse OTF:

$$\boldsymbol{G}_R = (\mathcal{F}^{-1}\hat{\boldsymbol{H}}^{-1}) * \boldsymbol{G}'. \tag{17.77}$$

At first glance, inverse filtering appears straightforward. In most cases, however, it is useless or even impossible to apply Eqs. (17.76) and (17.77). The reason for the failure is related to the fact that the OTF is often zero in wide ranges. The OTFs for motion blur (Eq. (17.74)) and defocusing (Eq. (17.72)) have extended zero range. In these areas, the inverse OTF becomes infinite.

Not only the zeroes of the OTF cause problems; already all the ranges in which the OTF becomes small do so. This effect is related to the influence of noise. For a quantitative analysis, we assume the following simple image formation model:

$$\boldsymbol{G}' = \boldsymbol{H} * \boldsymbol{G} + \boldsymbol{N} \quad \circ\!\!-\!\!\bullet \quad \hat{\boldsymbol{G}}' = \hat{\boldsymbol{H}} \cdot \hat{\boldsymbol{G}} + \hat{\boldsymbol{N}} \tag{17.78}$$

Equation (17.78) states that the noise is added to the image *after* the image is degraded. With this model, according to Eq. (17.75), inverse filtering yields

$$\hat{\boldsymbol{G}}_R = \hat{\boldsymbol{H}}^{-1} \cdot \hat{\boldsymbol{G}}' = \hat{\boldsymbol{G}} + \hat{\boldsymbol{H}}^{-1} \cdot \hat{\boldsymbol{N}} \tag{17.79}$$

provided that $\hat{\boldsymbol{H}} \neq 0$. This equation states that the restored image is the restored original image $\hat{\boldsymbol{G}}$ plus the noise amplified by $\hat{\boldsymbol{H}}^{-1}$.

If $\hat{\boldsymbol{H}}$ tends to zero, $\hat{\boldsymbol{H}}^{-1}$ becomes infinite, and so does the noise level. Equations (17.78) and (17.79) also state that the signal to noise ratio is not improved at all but remains the same because the noise and the useful image content in the image are multiplied by the same factor.

From this basic fact we can conclude that inverse filtering does not improve the image quality at all. More generally, it is clear that no linear technique will do so. All we can do with linear techniques is to amplify the structures attenuated by the degradation up to the point where the noise level still does not reach a critical level.

As an example, we discuss the 3-D reconstruction from microscopic *focus series*. A focus series is an image stack of microscopic images in which we scan the focused depth. Because of the limited *depth of field* (Section 7.4.3), only objects in a thin plane are imaged sharply. Therefore, we obtain a 3-D image. However, it is distorted by the point spread function of optical imaging. Certain structures are completely filtered out and blurred objects are superimposed over sharply imaged objects. We can now use inverse filtering to try to limit these distortions.

It is obvious that an exact knowledge of the PSF is essential for a good reconstruction. In Section 7.6.1, we computed the 3-D *PSF* of optical imaging neglecting lens errors and resolution limitation due to diffraction. However, high magnification microscopy images are diffraction-limited.

The diffraction-limited 3-D PSF was computed by Erhardt et al. [40]. The resolution limit basically changes the double cone of the 3-D PSF (Fig. 7.13) only close to the focal plane. At the focal plane, a point is no longer imaged to a point but to a diffraction disk. As a result, the *OTF* drops off to higher wave numbers in the $k_x k_y$ plane. To a first approximation, we can regard the diffraction-limited resolution as an additional lowpass filter by which the OTF is multiplied for geometrical imaging and by which the PSF is convolved.

The simplest approach to obtain an optimal reconstruction is to limit application of the inverse OTF to the wave number components that are not damped below a critical threshold. This threshold depends on the noise in the images. In this way, the true inverse OTF is replaced by an *effective inverse OTF* which approaches zero again in the wave number regions that cannot be reconstructed.

The result of such a reconstruction procedure is shown in Fig. 17.11. A 64 × 64 × 64 focus series has been taken of the nucleus of a cancerous rat liver cell. The resolution in all directions is 0.22 μm. The images clearly verify the theoretical considerations. The reconstruction considerably improves the resolution in the xy image plane, while the resolution in the z direction — as expected — is clearly worse. Structures that change in the z direction are completely eliminated in the focus series by convolution with the PSF of optical images and therefore can not be reconstructed.

17.5.4 Iterative Inverse Filtering

Iterative techniques form an interesting variant of inverse filtering as they give control over the degree of reconstruction to be applied. Let $\mathcal{H}$ be the blurring operator. We introduce the new operator $\mathcal{H}' = \mathcal{I} - \mathcal{H}$. Then the inverse operator

$$\mathcal{H}^{-1} = \frac{\mathcal{I}}{\mathcal{I} - \mathcal{H}'} \tag{17.80}$$

can be approximated by the Taylor expansion

$$\mathcal{H}^{-1} = \mathcal{I} + \mathcal{H}' + \mathcal{H}'^2 + \mathcal{H}'^3 + \ldots, \tag{17.81}$$

or, written explicitly for the OTF in the continuous Fourier domain,

$$\hat{h}^{-1}(\boldsymbol{k}) = 1 + \hat{h}' + \hat{h}'^2 + \hat{h}'^3 + \ldots. \tag{17.82}$$

In order to understand how the iteration works, we consider periodic structures. First, we take one that is only slightly attenuated. This means that $\hat{h}$ is only slightly less than one. Thus, $\hat{h}'$ is small and the iteration converges rapidly.

The other extreme is when the periodic structure has nearly vanished. Then, $\hat{h}'$ is close to one. Consequently, the amplitude of the periodic structure increases

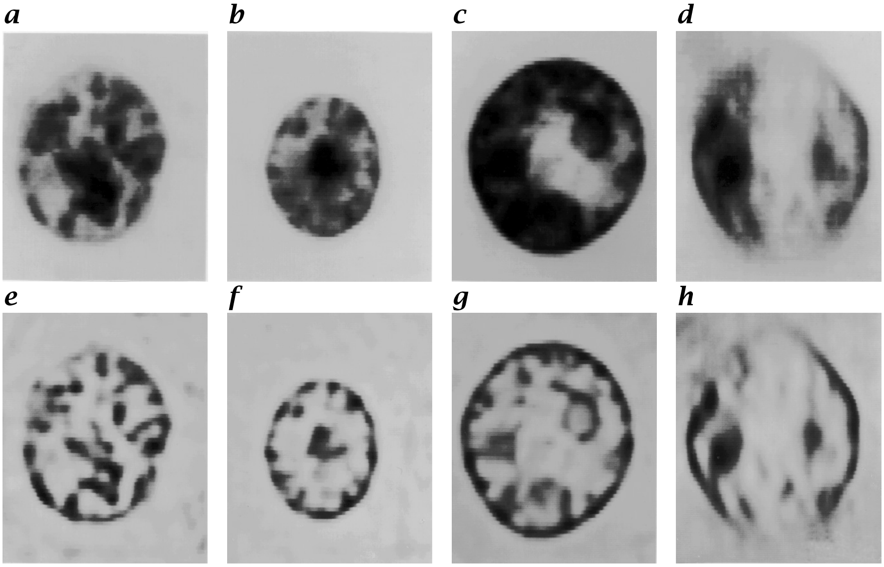

Figure 17.11: *3-D reconstruction of a focus series of a cell nucleus taken with conventional microscopy. Upper row:* ***a–c*** *selected original images;* ***d*** *xz cross section perpendicular to the image plane. Lower row:* ***e–h*** *reconstructions of the images* ***a–d*** *; courtesy of Dr. Schmitt and Prof. Dr. Komitowski, German Cancer Research Center, Heidelberg.*

by the same amount with each iteration step (linear convergence). This procedure has the significant advantage that we can stop the iteration as soon as the noise patterns become noticeable.

A direct application of the iteration makes not much sense because the increasing exponents of the convolution masks become larger and thus the computational effort increases from step to step. A more efficient scheme known as *Van Cittert iteration* utilizes Horner's scheme for polynomial computation:

$$\boldsymbol{G}_0 = \boldsymbol{G}', \quad \boldsymbol{G}_{k+1} = \boldsymbol{G}' + (\boldsymbol{I} - \boldsymbol{H}) * \boldsymbol{G}_k. \tag{17.83}$$

In Fourier space, it is easy to examine the convergence of this iteration. From Eq. (17.83)

$$\hat{g}_k(\boldsymbol{k}) = \hat{g}'(\boldsymbol{k}) \sum_{i=0}^{k} (1 - \hat{h}(\boldsymbol{k}))^i. \tag{17.84}$$

This equation constitutes a geometric series with the start value $a_0 = \hat{g}'$ and the factor $q = 1 - \hat{h}$. The series converges only if $|q| = |1 - \hat{h}| < 1$. Then the sum is given by

$$\hat{g}_k(\boldsymbol{k}) = a_0 \frac{1 - q^k}{1 - q} = \hat{g}'(\boldsymbol{k}) \frac{1 - |1 - \hat{h}(\boldsymbol{k})|^k}{\hat{h}(\boldsymbol{k})} \tag{17.85}$$

and converges to the correct value $\hat{g}'/\hat{h}$. Unfortunately, this condition for convergence is not met for all transfer functions that have negative values. Therefore the Van Cittert iteration cannot be applied to motion blurring and to defocusing.

A slight modification of the iteration process, however, makes it possible to use it also for degradations with partially negative transfer functions. The simple trick is to apply the transfer function twice. The transfer function $\hat{h}^2$ of the cascaded filter $\boldsymbol{H} * \boldsymbol{H}$ is positive.

The modified iteration scheme is

$$\boldsymbol{G}_0 = \boldsymbol{H} * \boldsymbol{G}', \quad \boldsymbol{G}_{k+1} = \boldsymbol{H} * \boldsymbol{G}' + (\boldsymbol{I} - \boldsymbol{H} * \boldsymbol{H}) * \boldsymbol{G}_k. \tag{17.86}$$

With $a_0 = \hat{h}\hat{g}'$ and $q = 1 - \hat{h}^2$ the iteration again converges to the correct value

$$\lim_{k\to\infty} \hat{g}_k(\boldsymbol{k}) = \lim_{k\to\infty} \hat{h}\hat{g}' \frac{1 - |1 - \hat{h}^2|^k}{\hat{h}^2} = \frac{\hat{g}'}{\hat{h}}, \quad \text{if} \quad |1 - \hat{h}^2| < 1 \tag{17.87}$$

17.6 *Further Equivalent Approaches*

This final section shows further equivalent approaches to modeling, which shed light to modeling from different point of views. As another continuous approach elasticity models are discussed in Section 17.6.1 and as an interesting discrete approach electric network models (Section 17.6.2).

17.6.1 *Elasticity Models*

At this point of our discussion, it is useful to discuss an analogous physical problem that gives further insight how similarity and smoothing constraints balance each other. With a physical model these two terms correspond to two types of forces.

Again, we will use the example of optical flow determination. We regard the image as painted onto an *elastic membrane*. Motion will shift the membrane from image to image. Especially nonuniform motion causes a slight expansion or contraction of the membrane. The similarity term acts as an external force that tries to pull the membrane towards the corresponding *displacement vector* (DV). The inner elastic forces distribute these deformations continuously over the whole membrane, producing a smooth *displacement vector field* (DVF).

Let us first consider the external forces in more detail. It does not make much sense to set the deformations at those points where we can determine the DV to the estimated displacement without any flexibility. Instead we allow deviations from the expected displacements which may be larger, the more uncertain the determination of the DV is. Physically, this is similar to a pair of springs whose spring constant is proportional to the certainty with which the displacement can be calculated. The zero point of the spring system is set to the computed displacement vector. As the membrane is two-dimensional, two pairs of springs are required. The direction of the spring system is aligned according to the *local orientation* (Section 13.3). At an edge, only the displacement normal to the edge can be computed (*aperture problem*, Section 14.2.2). In this case, only one spring pair is required; a displacement parallel to the edge does not result in a restoring force.

The external spring forces are balanced by the inner elastic forces trying to even out the different deformations. Let us look again at the Euler-Lagrange equation of the optical flow (Eq. (17.23)) from this point of view. We can now understand this equation in the following way:

$$\underbrace{(\nabla g\, \boldsymbol{f} + g_t)\, \nabla g}_{\text{external force}} - \underbrace{\alpha^2 \Delta \boldsymbol{f}}_{\text{internal force}} = \boldsymbol{0}, \tag{17.88}$$

where α^2 is an *elasticity constant*. In the expression for the internal forces only second derivatives appear, because a constant gradient of the optical flow does not result in net inner forces.

The elasticity features of the membrane are expressed in a single constant. Further insight into the inner structure of the membrane is given by the Lagrange function (Eq. (17.19)):

$$L\left(\boldsymbol{f}, \boldsymbol{f}_{x_p}, \boldsymbol{x}\right) = \underbrace{\alpha^2 \left(|\nabla \boldsymbol{f}_1|^2 + |\nabla \boldsymbol{f}_2|^2\right)}_{T\text{, deformation energy}} + \underbrace{(\nabla g\, \boldsymbol{f} + g_t)^2}_{-V\text{, potential}}. \tag{17.89}$$

The Lagrange function is composed of the *potential* of the external force as it results from the continuity of the optical flow and an energy term related to the inner forces. This term is thus called *deformation energy*. This energy appears in place of the kinetic energy in the classical example of the Lagrange function for a mass point, as the minimum is not sought in time but in space.

The deformation energy may be split up into several terms which are closely related to the different modes of deformation:

$$T\left(\boldsymbol{f}_{x_p}\right) = \frac{1}{2}\left[\underbrace{\left((f_1)_x + (f_2)_y\right)^2}_{\text{dilation}} + \underbrace{\left((f_1)_x - (f_2)_y\right)^2 + \left((f_1)_y + (f_2)_x\right)^2}_{\text{shear}} + \underbrace{\left((f_1)_y - (f_2)_x\right)^2}_{\text{rotation}}\right]. \tag{17.90}$$

Clearly, the elasticity features of the membrane match the kinematics of motion optimally. Each possible deformation that may occur because of the different modes of 2-D motion on the image plane is equally weighted.

Physically, such a membrane makes no sense. The differential equation for a real physical membrane is different [45]:

$$\boldsymbol{f} - (\lambda + \mu)\nabla(\nabla \boldsymbol{u}) - \mu \Delta \boldsymbol{u} = \boldsymbol{0}. \tag{17.91}$$

The elasticity of a physical membrane is described by the two constants λ and μ. $\lambda = -\mu$ is not possible; as a result, the additional term $\nabla(\nabla \boldsymbol{u})$ (in comparison to the model membrane for the DVF) never vanishes. If there is no cross contraction, λ can only be zero.

With the membrane model, only the elongation is continuous, but not the first-order derivative. Discontinuities occur exactly at the points where external forces are applied to the membrane. This results directly from Eq. (17.23). A locally applied external force corresponds to a δ distribution in the similarity

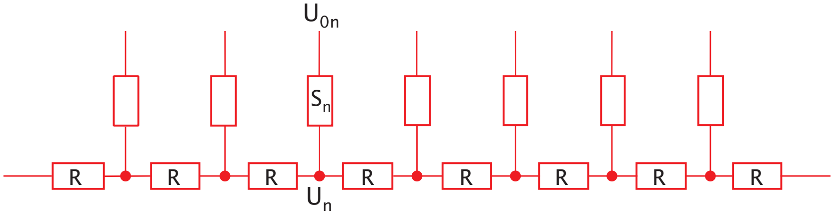

Figure 17.12: *Simple 1-D network for a 1-D smooth DVF; after Harris [70].*

term. Integrating Eq. (17.23), we obtain a discontinuity in the first-order derivatives.

These considerations call into question the smoothness constraints considered so far. We know that the motion of planar surface elements does not result in such discontinuities. Smoothness of the first-order derivatives can be forced if we include second-order derivatives in the smoothness term (Eq. (17.23)) or the deformation energy (Eq. (17.89)). Physically, such a model is similar to a thin *elastic plate* that cannot be folded like a membrane.

17.6.2 *Network Models*

In this section we discuss another method emerging from electrical engineering, the *network model*. It has the advantage of being a discrete model which directly corresponds to discrete imagery. This section follows the work of Harris [69, 70]. The study of network models has become popular since network structures can be implemented directly on such massive parallel computer systems as the Connection Machine at Massachusetts Institute of Technology (MIT) [70] or in analog VLSI circuits [135].

One-Dimensional Networks. First, we consider the simple 1-D case. The displacement corresponds to an electric tension. Continuity is forced by interconnecting neighboring pixels with electrical resistors. In this way, we build up a linear resistor chain as shown in Fig. 17.12. We can force the displacement at a pixel to a certain value by applying a potential at the corresponding pixel. If only one voltage source exists in the resistor chain, the whole network is put to the same constant voltage. If another potential is applied to a second node of the network and all interconnecting resistors are equal, we obtain a linear voltage change between the two points. In summary, the network of resistors forces continuity in the voltage, while application of a voltage at a certain node forces similarity.

There are different types of boundary conditions. On the one hand, we can apply a certain voltage to the edge of the resistor chain and thus force a certain value of the displacement vector at the edge of the image. On the other hand, we can make no connection. This is equivalent to setting the first-order spatial derivative to zero at the edge. The voltage at the edge is then equal to the voltage at the next connection to a voltage source.

In the elasticity models (Section 17.6.1) we did not set the displacements to the value resulting from the similarity constraint directly, but allowed for some flex-

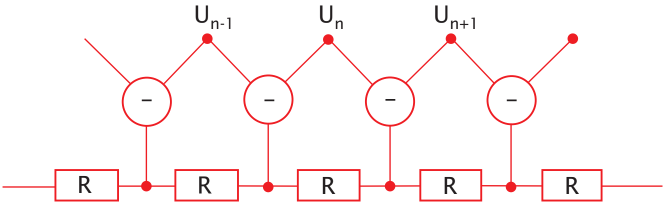

Figure 17.13: *Discrete network model for a 1-D scalar feature with smooth first-order derivatives; after Harris [70].*

ibility by applying the displacement via a spring. In a similar manner we apply the voltage, U_{0n}, to the node n not directly but via the resistor S_n (Fig. 17.12). We set the resistance proportional to the uncertainty of the displacement vector.

The difference equation for the network model is given by the rule that the sum of all currents must cancel each other at every node of the network. Using the definitions given in Fig. 17.12, we obtain for the node n of the network

$$\frac{U_n - U_{0n}}{S_n} + \frac{U_n - U_{n-1}}{R} + \frac{U_n - U_{n+1}}{R} = 0. \tag{17.92}$$

The two fractions on the right side constitute the second-order discrete differentiation operator $\mathcal{D}_x^2$ (see Section 12.5.2). Thus Eq. (17.92) results in

$$\frac{1}{S}(U - U_0) - \frac{1}{R}\frac{\partial^2 U}{\partial x^2} = 0. \tag{17.93}$$

This equation is the 1-D form of Eq. (17.23). For a better comparison, we rewrite this equation for the 1-D case:

$$(\partial_x g)^2 \left(f + \frac{\partial_t g}{\partial_x g} \right) - \alpha^2 \frac{\partial^2 f}{\partial x^2} = 0. \tag{17.94}$$

Now we can quantify the analogy between the displacement vectors and the network model. The application of the potential U_0 corresponds to the computation of the local velocity by $-(\partial_t g)/(\partial_x g)$. The similarity and smoothness terms are weighted with the reciprocal resistance (conductance) $1/S$ and $1/R$ instead of with the squared gradient $(\partial_x g)^2$ and α^2.

Generalized Networks. Now we turn to the question of how to integrate the continuity of first-order derivatives into the network model. Harris [69] used an active subtraction module which computes the difference of two signals. All three connections of the element serve as both inputs and outputs. At two arbitrary inputs we apply a voltage and obtain the corresponding output voltage at the third connection.

Such a module requires active electronic components [69]. Figure 17.13 shows how this subtraction module is integrated into the network. It computes the difference voltage between two neighboring nodes. These differences — and not the voltages themselves — are put into the resistor network.

In this way we obtain a network that keeps the first derivative continuous. We can generalize this approach to obtain networks that keep higher-order derivatives continuous by adding several layers with subtraction modules (Fig. 17.14).

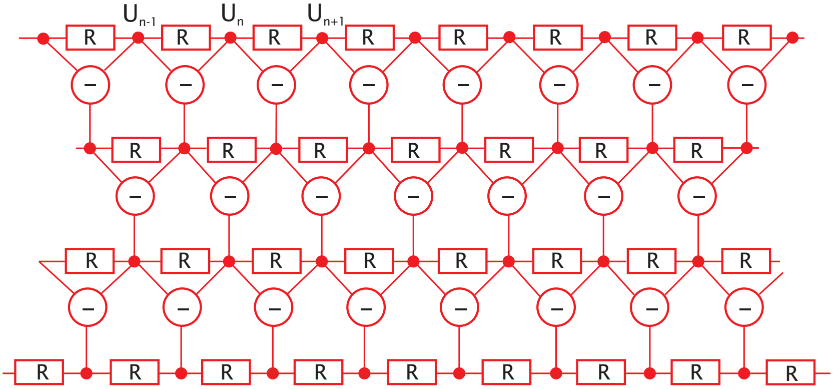

Figure 17.14: *Generalized network for a 1-D DVF that keeps higher-order derivatives smooth; after Harris [70].*

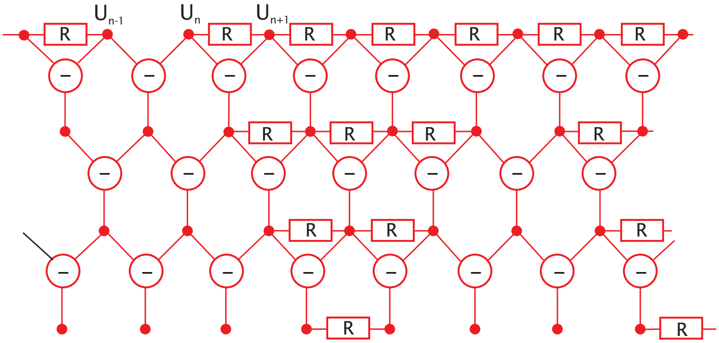

Figure 17.15: *Generalized 1-D network with a discontinuity in the DVF and its first spatial derivative as indicated.*

Discontinuities in Networks. Displacement vector fields show discontinuities at the edges of moving objects. Discontinuities can easily be implemented in the network model. In the simple network with zero-order continuity (Fig. 17.12), we just remove the connecting resistor between two neighboring nodes to produce a potential jump between these two nodes. In order to control the smoothness (Section 17.2.5), we can also think of a nonlinear network model with voltage-dependent resistors. We might suspect discontinuities at steep gradients in the velocity field. If the resistance increases with the tension, we have a mechanism to produce implied discontinuities. These brief considerations illustrate the flexibility and suggestiveness of network models.

Integration of discontinuities is more complex in a generalized network. Here we may place discontinuities at each level of the network, i. e., we may make either the DVF or any of its derivatives discontinuous by removing a resistor at the corresponding level. We need to remove all resistors of deeper-lying nodes that are connected to the point of discontinuity (Fig. 17.15). Otherwise, the

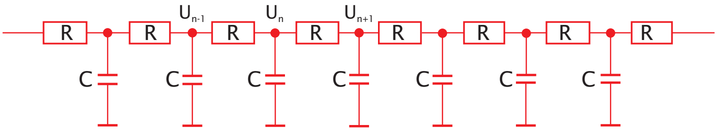

***Figure 17.16:** 1-D network with capacitors to simulate the convergence of iterative solutions.*

higher-order derivatives stay continuous and cause the lower-order derivatives to become continuous.

Two-Dimensional Networks. The network model can also be used for higher-dimensional problems. For a 2-D network model with zero-order continuity, we build up a 2-D mesh of resistors. The setup of generalized 2-D network models with higher-order continuity constraints is more complex. In each level we must consider the continuity of several partial derivatives. There are two first-order spatial derivatives, a horizontal and a vertical one. For each of them, we need to build up a separate layer with subtraction modules as shown in Fig. 17.13, in order to observe the smoothness constraint. Further details can be found in Harris [70].

Multigrid Networks. One of the most important practical issues is finding the rate of convergence of iterative methods for solving large equation systems in order to model them with networks. The question arises of whether it is also possible to integrate this important aspect into the network model. Iteration introduces a time dependency into the system, which can be modeled by adding capacitors to the network (Fig. 17.16). The capacitors do not change at all the static properties of the network.

When we start the iteration, we know the displacement vectors only at some isolated points. Therefore we want to know how many iterations it takes to carry this information to distant points where we do not have any displacement information. To answer this question, we derive the difference equation for the resistor-capacitor chain as shown in Fig. 17.16. It is given by the rule that the sum of all currents flowing into one node must be zero. In addition, we need to know that the current flowing into a capacitor is proportional to its capacitance C and the temporal derivative of the voltage $\partial U/\partial t$:

$$\frac{U_{n-1}-U_n}{R}+\frac{U_{n+1}-U_n}{R}-C\frac{\partial U_n}{\partial t}=0 \tag{17.95}$$

or

$$\frac{\partial U_n}{\partial t}=\frac{(\Delta x)^2}{RC}\frac{\partial^2 U_n}{\partial x^2}. \tag{17.96}$$

In the second equation, we have introduced Δx as the spatial distance between neighboring nodes in order to formulate a spatial derivative. Also, $RC = \tau$, the time constant of an individual resistor-capacitor circuit. Equation (17.96) is the discrete 1-D formulation of one of the most important equations in natural sciences, the *transport* or *diffusion* equation, which we discussed in detail in Sections 5.3.1 and 17.3. Without explicitly solving Eq. (17.96), we can answer the question as to the time constant needed to smooth the displacement vector field over a certain space scale. Let us assume a spatially varying potential with a wavelength λ decreasing exponentially with a time constant τ_λ that depends on the wavelength λ (compare Section 5.3.1):

$$U(x) = U_0(x)\exp(-t/\tau)\exp(\mathrm{i}kx). \tag{17.97}$$

Introducing this equation into Eq. (17.96), we obtain

$$\tau_\lambda = \frac{\tau}{(\Delta x\, k)^2} = \frac{\tau}{4\pi^2(\Delta x)^2}\lambda^2. \tag{17.98}$$

With this result, we can answer the question as to the convergence time of the iteration. The convergence time goes with the square of the wavelength of the structure. Consequently, it takes four times longer to get gray values at double the distance into equilibrium. Let us arbitrarily assume that we need one iteration step to bring neighboring nodes into equilibrium. We then need 100 iteration steps to equilibrate nodes that are 10 pixels distant. If the potential is only known at isolated points, this approach converges too slowly to be useful.

Multigrid data structures, which we discussed in Chapter 5, are an efficient tool to accelerate the convergence of the iteration. At the coarser levels of the pyramid, distant points come much closer together. In a pyramid with only six levels, the distances shrink by a factor of 32. Thus we can compute the large-scale structures of the DVF with a convergence rate that is about 1000 times faster than on the original image. We do not obtain any small-scale variations, but can use the coarse solution as the starting point for the iteration at the next finer level.

In this way, we can refine the solution from level to level and end up with a full-resolution solution at the lowest level of the pyramid. The computations at all the higher levels of the pyramid do not add a significant overhead, as the number of pixels at all levels of the pyramid is only one third more than at the lowest level. The computation of the DVF of the taxi scene (Fig. 17.3) with this method is shown in Fig. 17.4.

17.7 *Exercises*

17.1: Inhomogeneous and anisotropic diffusion

Interactive demonstration of smoothing using inhomogeneous and anisotropic diffusion (dip6ex17.01)

17.2: Regularized motion analysis

Interactive demonstration of several techniques for regularized motion analysis (dip6ex17.02)

17.3: Iterative inverse filtering

Interactive demonstration of iterative inverse filtering; generation of test images with motion blur and defocusing (dip6ex17.03).

17.4: ****Plane regression**

Study the regression of an image function by a plane with the least squares technique discussed in Section 17.4.1:

$$d(x, y) = a_0 + a_1 x + a_2 y$$

Questions:

1. Determine the overdetermined equation system ($\boldsymbol{G m} = \boldsymbol{d}$).
2. Under which conditions does the overdetermined equation system result in a unique least-squares solution? Discuss the properties of the matrix $\boldsymbol{G}^T\boldsymbol{G}$, which needs to be inverted. (Hint: it is easy to answer this question if you diagonalize the symmetric matrix (principal coordinate system).)
3. Under which conditions are the parameters of the plane fit

   $$\boldsymbol{m} = [a_0, a_1, a_2]^T$$

   statistically uncorrelated? (Hint: you need the covariance matrix of $\boldsymbol{m}$, which is given by $\mathrm{cov}(\boldsymbol{m}) = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\sigma^2$ for statistically uncorrelated data d with an equal variance σ^2.)
4. Solve the equation system explicitly for the case of $3 \times 3 = 9$ data points on a square grid with the distance Δx that is centered at the origin. How does the accuracy of the regression parameters $\boldsymbol{m}$ depend on the distance Δx?
5. Can you express the estimate of the three regression parameters $\boldsymbol{m} = [a_0, a_1, a_2]^T$ as convolution operations? If yes, compute the corresponding convolution masks.

17.5: ***Inverse filtering**

Consider the following point spread functions for a 1-D blurring:

1. $\boldsymbol{H} = [1/3, 1/3, 1/3]$ (box mask)
2. $\boldsymbol{H} = [1/4, 1/2, 1/4]$ (binomial mask)
3. $\boldsymbol{H} = [1/8, 3/4, 1/8]$

Answer the following questions

- Is it possible to remove the blurring by inverse filtering?
- If yes, determine the transfer function of the inverse filter.
- If yes, determine the convolution mask of the inverse filter (Tip: series expansion).

17.6: ****Iterative inverse filtering**

We assume that the image $\boldsymbol{G}'$ is blurred by a convolution with the mask $\boldsymbol{H}$ and denote the series of iteratively restored images with $\boldsymbol{G}_k$. Three well-known iteration schemes are
Van Cittert iteration:

$$\boldsymbol{G}_0 = \boldsymbol{G}', \quad \boldsymbol{G}_{k+1} = \boldsymbol{G}' + (\boldsymbol{I} - \boldsymbol{H}) * \boldsymbol{G}_k$$

Stabilized VanCittert iteration:

$$\boldsymbol{G}_0 = \boldsymbol{H} * \boldsymbol{G}', \quad \boldsymbol{G}_{k+1} = \boldsymbol{H} * \boldsymbol{G}' + (\boldsymbol{I} - \boldsymbol{H} * \boldsymbol{H}) * \boldsymbol{G}_k$$

Regularized iteration:

$$\boldsymbol{G}_0 = \boldsymbol{H} * \boldsymbol{G}', \quad \boldsymbol{G}_{k+1} = \boldsymbol{H} * \boldsymbol{G}' + (\boldsymbol{B} - \boldsymbol{H} * \boldsymbol{H}) * \boldsymbol{G}_k$$

($\boldsymbol{I}$ means the identity operator and $\boldsymbol{B}$ a smoothing mask.)
Use the following degradation masks

1. $\boldsymbol{H} = [1/3, 1/3, 1/3]$ (box mask)
2. $\boldsymbol{H} = [1/8, 3/4, 1/8]$

to answer the following questions:

- Does the iteration converge?
- If yes, against which limit?

(Hint: The questions cans be answered easily in Fourier space!)

17.8 *Further Readings*

This subject of this chapter relies heavily on matrix algebra. Golub and van Loan [63] give an excellent survey on matrix computations. Variational methods (Section 17.2) are expounded by Jähne et al. [96, Vol. 2, Chapter 16] and Schnörr and Weickert [181]. The usage of the membran model (Section 17.6.1) was first reported by Broit [15], who applied it in computer tomography. Later it was used and extended by Dengler [35] for image sequence processing. Nowadays, elasticity models are a widely used tool in quite different areas of image processing such as modeling and tracking of edges [104], reconstruction of 3-D objects [202] and reconstruction of surfaces [201]. Anisotropic diffusion (Section 17.3) and nonlinear scale spaces are an ongoing research topic. An excellent account is given by Weickert [215] and Jähne et al. [96, Vol. 2, Chapter 15]. Optimal filters for fast anisotropic diffusion are discussed by Scharr and Weickert [179] and Scharr and Uttenweiler [178].