

# Chapter 3

## Optical Flow Estimation

### 3.1 Introduction

*Optical flow* is the velocity vector field of the projected environmental surfaces when a viewing system moves relative to the environment. Optical flow is a long standing subject of intensive investigation in diverse fields such as psychology, psychophysics, and computer vision [1–9]. In computer vision, of interest to us here, optical flow estimation has been a topic of continued interest and extensively researched. One of the most referenced paper on the subject is *Determining optical flow*, 1981, by B.K.P. Horn and B.G. Schunck [10]. It is also one of the most influential for having served as ground or benchmark for just about every dense flow computation algorithm. The Horn and Schunck variational formulation, which we will describe in detail subsequently (Sect. 3.4), seeks to determine the flow which minimizes a weighted sum of two integrals over the image domain, one to bring the flow to conform to the image spatiotemporal variations and the other to regularize the solution by constraining it to be smooth:

$$\mathcal{E}(u, v) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \lambda \int_{\Omega} (\|\nabla u\|^2 + \|\nabla v\|^2) dx dy, \quad (3.1)$$

where  $I : (x, y, t) \in \Omega \times ]0, T[ \mapsto I(x, y, t) \in \mathbf{R}^+$  is the image sequence of domain  $\Omega$  and duration  $T$ ,  $I_x, I_y, I_t$  its spatiotemporal derivatives,  $\nabla u, \nabla v$  the spatial gradients of the coordinates  $u, v$  of optical flow, and  $\lambda$  is a real constant to balance the contribution of the two terms in the functional. The corresponding Euler-Lagrange equations yield the flow via efficient implementation by Jacobi/Gauss-Seidel iterations.

A paper published the same year as [10] by B. D. Lucas and T. Kanade [11] on image registration and application to stereo-vision, has also been extensively referenced and used for optical flow estimation. The view taken was quite different as the scheme sought to determine the coordinate transformation between two images

which minimized the displaced frame difference (DFD), i.e., the squared difference between one image and the other evaluated after the coordinate transformation (displaced, or warped as it is sometimes called). If the images are  $I_1$  and  $I_2$ , and  $\mathbf{x} \rightarrow \mathbf{f}(\mathbf{x}; \theta)$  is a parametric coordinate transformation with parameter vector  $\theta$ , the scheme minimizes with respect to  $\theta$  the objective function:

$$E(\theta) = \sum_{\mathbf{x} \in D} (I_1(\mathbf{f}(\mathbf{x}; \theta)) - I_2(\mathbf{x}))^2, \quad (3.2)$$

where  $D$  is a discretization of  $\Omega$ . The minimization is carried out iteratively by expanding linearly  $I_1$  at each step about the transformed coordinates of the previous step. The displacement at each point is computed subsequently from the estimated coordinate transformation: therefore, one of the significant conceptual differences between the methods of [10] and [11] is that the scheme in [10] references a vector field, i.e., a velocity vector as a variable at each point of the image domain, whereas the unknown in [11] is a global coordinate transformation between two images. Another difference is that the points at which there is no texture, i.e., where the spatial gradient is zero in the transformed image, do not contribute to determining the coordinate transformation whereas spatial regularization is a central concept in [10] which makes every point contribute to optical flow. From a computational point of view, the method of [11] involves a coordinate transformation and evaluation of the transformed image via spatial interpolation, an operation which does not occur in [10]. Often, the transformation has been applied locally in windows to allow spatial variations of the parameter vector estimate, which would improve its accuracy, but the window size affects the outcome which also suffers from the so-called block effect due to the lack of spatial regularization. However, both schemes [10, 11] have been combined in a continuous variational framework [12].

The Horn and Schunck algorithm solves a large but significantly sparse system of linear equations, which can be done very efficiently by convergent Jacobi or Gauss-Seidel iterations, particularly block-wise iterations [13]. A parallel hardware version has also been implemented [14, 15]. However, the basic neighborhood operations which drive the algorithm blur the optical flow estimate at motion boundaries. This serious problem is caused by the quadratic smoothness regularization term of the objective functional which leads to a Laplacian operator in the Euler-Lagrange equations. The discrete version of the operator reduces to averaging the estimate locally, which has the undesirable effect of blurring the computed flow at motion boundaries. Therefore, studies have subsequently considered using motion boundary preserving spatial regularizations. The problem has been addressed from four different perspectives: image driven smoothing, robust statistics, boundary length penalties, and nonlinear diffusion.

With image driven smoothing, the view is that motion edges coincide or tend to coincide with the image intensity edges, which would then justify that image motion smoothing be mediated by the image gradient [16–20]. However, this may also cause undue smoothing of motion because motion edges do not always occur at intensity edges, although image edges generally occur at motion edges.

Along the vein of robust statistics [21], motion discontinuity preservation is based on the notion of *outliers*. From this viewpoint, basically, the underlying interpretation is that the optical flow values over an image positional array satisfy the spatiotemporal data constraint and are smoothly varying everywhere except at motion boundaries where there are treated as outliers [22, 23]. This led to the use of robust functions such as the  $L^1$  norm, the truncated quadratic [24], and the Lorentzian, in lieu of the quadratic to evaluate the objective functional terms. If  $\rho$  is a robust function, a discrete objective function one can seek to minimize is:

$$E(u, v) = \sum_{\mathbf{x} \in D} \left( (I_x u + I_y v + I_t)^2(\mathbf{x}) + \lambda \sum_{\mathbf{y} \in \mathcal{N}_{\mathbf{x}}} (\rho(u(\mathbf{x}) - u(\mathbf{y})) + \rho(v(\mathbf{x}) - v(\mathbf{y}))) \right), \quad (3.3)$$

where  $\mathcal{N}_{\mathbf{x}}$  is a set of neighbors of  $\mathbf{x}$  (e.g., the 4-neighborhood). Slightly more general expressions can be adopted [22, 23, 25]. From this view of the problem, the effect of the robust function is to reduce the influence of the outliers on the estimation of optical flow and, therefore, provide a better definition of motion boundaries where the outliers are anticipated.

Another view of motion boundary preservation is to reference motion edges in the formulation and introduce a motion boundary length penalty term in the objective functional. This has been done via a line process in Markov random field (MRF) modelling [26]. Such a formulation, where edges are referenced by the MRF line process, has led to applications to optical flow estimation [27–29]. The objective function data term is still as in Eq. (3.3) and the regularization has the form:

$$\lambda \sum_{\mathbf{x} \in D} \sum_{\mathbf{y} \in \mathcal{N}_{\mathbf{x}}} \left( \alpha (1 - l_{\mathbf{x}, \mathbf{y}}) \|W(\mathbf{x}) - W(\mathbf{y})\|^2 + \beta l_{\mathbf{x}, \mathbf{y}} \right), \quad (3.4)$$

where  $W = (u, v)$ ,  $\alpha$  and  $\beta$  are constants, and  $l_{\mathbf{x}, \mathbf{y}} \in \{0, 1\}$  is the binary variable of the MRF line process to represent the motion boundaries. Motion edges can also be referenced in a functional with a boundary length penalty term [30, 31] as in image segmentation [32, 33].

Finally, motion discontinuity preservation can be investigated from the perspective of nonlinear diffusion [34]. The rationale, basically, is that spatial regularization should be selective by allowing isotropic smoothing inside motion regions where optical flow is thought to be smooth, i.e., varies little spatially, and inhibit it across motion boundaries [35–45]. In particular, the  $L_1$  norm regularization, also called total variation regularization and often abbreviated TV, which has been extensively investigated in inverse problems [46], notably image restoration [47], has been used for continuous variational optical flow estimation in several studies [39, 42–45, 48]. To simplify the rather elaborate TV minimization algorithm, and expedite the implementation thereof, the absolute value function in TV regularization,  $g(z) = |z|$ , is often replaced by a function of the sort  $g(z) = \sqrt{z^2 + \varepsilon^2}$ , for some small  $\varepsilon$ .

In the context of nonlinear diffusion optical flow estimation, the study in [37, 38] is singular in that it investigated regularization functions  $g$  in the functional:

$$\mathcal{E}(u, v) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \lambda \int_{\Omega} (g(\|\nabla u\|) + g(\|\nabla v\|)) dx dy \quad (3.5)$$

by analyzing conditions which impose isotropic smoothing within motion regions and smoothing along motion boundaries but not across. The analysis leads to functions of the sort  $g(s) = 2\sqrt{(1+s^2)} - 2$  (Aubert),  $g(z) = \log(1+s^2)$  (Perona-Malik [49, 50]), and others.

The optical flow constraint, on which most formulations of optical flow estimation rely, refers to the image sequence temporal derivative. In practice, image motion is often of large extent, typically causing displacements of several pixels between consecutive views. As a result, the image temporal derivative may not be approximated accurately to bear on motion estimation. In such a case, motion estimation has been addressed efficiently by multiresolution/multigrid processing [22, 51, 52]. Multiresolution and multigrid processing are “multilevel” computations which solve a system of equations on a given discretization grid by solving smaller similar systems on grids at coarser discretization.

Optical flow estimation has also been cast in a framework of simultaneous motion estimation and segmentation [31, 53–61], where the purpose is to divide the image into regions corresponding to distinct motions. Joint estimation and segmentation accounts for motion boundaries since those coincide with motion region boundaries. However, the emphasis in segmentation is not necessarily on accurate motion estimation because motion regions can be distinguished using simple motion models, the piecewise constant or affine models for instance, which do not necessarily describe the fine variations of motion that may be occurring.

Finally, it is worth mentioning that disparity estimation in binocular images resembles optical flow estimation and both problems can be cast in similar variational formulations. As a result, benefits can accrue from their joint estimation in stereoscopic image sequences [62, 63].

The purpose in this chapter forthcoming sections is to provide a digest of optical flow estimation by variational methods. We will not review the very large literature but rather describe a number of methods that would expose the fundamental concepts underlying image motion estimation by variational methods. The important ideas presented include (i) the basic formulation of image motion estimation as the minimization of a functional containing a data term and a regularization term; (ii) the use of optical flow smoothness in the regularization term; (iii) the notion of a motion boundary and definitions of it which would allow its preservation; (iv) the representation of motion by parametric functions; (v) the relationship between motion estimation and motion-based segmentation; and (vi) the concepts of multiresolution and multigrid computation and their role in processing large-magnitude motion. Motion in stereoscopy will also be brought up. This reductive, concept-oriented description of image motion estimation will be enhanced by references to recent studies which

build upon the basic formulations we will discuss and provide important computational details.

We will start the presentation with the optical flow constraint (Sect. 3.2) and immediately follow with the benchmark algorithms of Lucas–Kanade (Sect. 3.3) and of Horn and Schunck (Sect. 3.4). Motion boundary preservation will be treated next with the scheme of Deriche, Aubert, and Kornprobst [37] (Sect. 3.5), followed by image intensity based regularization [20] (Sect. 3.6) and the minimum description length (MDL) [31] (Sect. 3.7) formulations. Section 3.8 will describe parametric motion representation and computation. After a brief mention of variants of the smoothness and regularization terms (Sect. 3.9), the chapter will continue with a discussion of multiresolution/multigrid processing (Sect. 3.10) and a presentation of joint optical flow estimation and segmentation [54] (Sect. 3.11). Motion estimation in stereoscopy will be investigated in Sect. 3.12. The chapter does not provide an experimental evaluation or comparison of the methods but it gives examples of results. Evaluations of methods can be found in some of the cited papers.

## 3.2 The Optical Flow Constraint

Let the image sequence be a  $C^1$  function  $I : (x, y, t) \in \Omega \times ]0, T[ \mapsto I(x, y, t) \in \mathbf{R}^+$ . Let  $\mathbf{P}$  be a point on an environmental surface and  $\mathbf{p}$  its image with coordinates  $x(t), y(t)$  at instant  $t$ . As  $\mathbf{P}$  moves in space, let the spatiotemporal trajectory of  $\mathbf{p}$  have the parametric representation  $t \rightarrow \mathbf{c}(t) = (x(t), y(t), t)$ . Let  $h$  be the function  $t \rightarrow h(t) = I \circ \mathbf{c}(t) = I(x(t), y(t), t)$ , where  $\circ$  indicates function composition. Function  $h$  is the image intensity along the motion trajectory of  $\mathbf{p}$ . If we assume that the intensity recorded from the environmental point  $\mathbf{P}$  does not change as the surface it lies on moves, i.e., if  $h$  is constant, then we have the *optical flow constraint* (OFC) at  $\mathbf{p}$ :

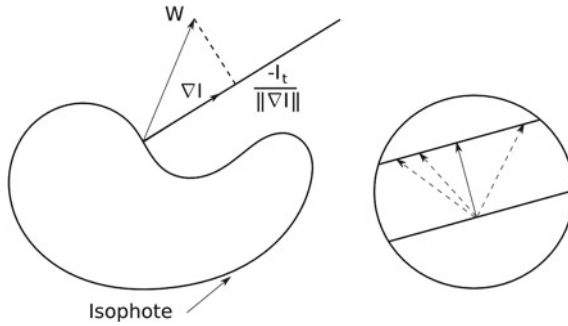
$$\frac{dh}{dt} = \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (3.6)$$

or, using the usual subscript notation for the partial derivatives:

$$I_x u + I_y v + I_t = 0, \quad (3.7)$$

where  $(u, v) = (\frac{dx}{dt}, \frac{dy}{dt})$  is called the *optical velocity* vector. The field over  $\Omega$  of optical velocities is the *optical flow*.

The assumption that the recorded intensity is constant along motion trajectories is valid for translating Lambertian surfaces under constant uniform lighting. It is generally accepted that it is a good approximation for small velocity motions of non specular surfaces occurring over a short period of time. There have been a few attempts at determining constraints other than the invariance of image intensity along motion trajectories [64–67] but, by and large, the Horn and Schunck OFC (or discrete writings of it) has been the basic constraint in optical flow studies.



**Fig. 3.1** *Left* The projection of the optical flow vector  $W$  on the image spatial gradient  $\nabla I$  can be estimated from the image first-order spatiotemporal variations; whenever  $\nabla I \neq 0$  it is equal to  $-\frac{I_t}{\|\nabla I\|}$ . *Right* The aperture problem: the movement of a straight edge seen through an aperture (a circular window in this figure) is ambiguous because only the component of motion in the direction perpendicular to the edge can be determined

If the optical velocity vector is denoted by  $W$ , the OFC is written  $\nabla I \cdot W + I_t = 0$  and its projection  $W^\perp$  in the direction of the image gradient, called the *normal component*, can be written:

$$W^\perp = \frac{\nabla I}{\|\nabla I\|} \cdot W = \frac{-I_t}{\|\nabla I\|}. \quad (3.8)$$

The spatiotemporal derivatives can be estimated from the image sequence data. Hence, the OFC determines the component of optical flow in the direction of the image gradient and only this component. This is a reflection of the *aperture problem*, the ambiguity in interpreting the translational motion of a straight line seen through an aperture, i.e., in the absence of any external visual cues (Fig. 3.1). The aperture problem is responsible for illusory percepts such as rotating spirals which appear to expand or contract and translating sine waves which appear highly non rigid [68]. The aperture problem was apprehended as early as 1911 by P. Stumpf [69] and has been the subject of many studies in psychophysics. In computer vision, it has been investigated in Hildreth's computational theory of visual motion measurement [70].

Local methods have been considered to solve the aperture problem. The simplest treatment assumes that optical flow is constant in the neighborhood of each point but that the image spatiotemporal gradient is not, leading to write one OFC for the same velocity at each point of the neighborhood [71]. Local processing of the aperture problem has also been addressed by the multiple OFC constraints method which assumes that there are  $m \geq 2$  distinct image functions  $I_1, \dots, I_m$  satisfying the assumption of invariance to motion and giving  $m$  independent OFC equations to solve simultaneously. Several sources of these functions have been looked at [6]: (a) *multispectral images*, i.e., signals of different wavelengths as in colour images [72, 73], (b) *operators/filters*, where  $I_1, \dots, I_m$  are obtained by applying  $m$  operators/filters  $O_1, \dots, O_m$  to a single image function  $f$ . Examples include spatial filters

applied to the original image [74] and differential operators [75–78]. Another source of functions is (c) *multiple illumination sources*, each giving a different image [79].

The assumptions supporting the local methods do not hold generally, leading to local systems of equations which are rank deficient or ill-conditioned. This is one important reason why variational methods which regularize the velocity field, such as those we reviewed in the introduction and some of which we will describe next, have been so much more effective.

### 3.3 The Lucas-Kanade Algorithm

The original study [11] addressed a general setting of image registration and developed an iterative algorithm which it applied to determining depth from stereoscopy. When used for optical flow evaluation, it has been applied in windows, typically  $5 \times 5$  [80].

Let  $I_1$  and  $I_2$  be two images with the same domain  $\Omega$  and  $\mathbf{f} = (f_1, f_2)$  a coordinate transformation parametrized by  $\theta = (\theta_1, \dots, \theta_n) \in \mathbb{R}^n$ :

$$\mathbf{f} : (\mathbf{x}, \theta) \in \Omega \times \mathbb{R} \rightarrow \mathbf{f}(\mathbf{x}, \theta) \in \Omega \quad (3.9)$$

Mapping  $\mathbf{f}$  is often called a warp to distinguish it from a transformation that would act on the intensity image rather than on the image domain. The problem is to determine the transformation that minimizes the smallest displaced frame difference, i.e., determine  $\tilde{\theta}$  such that:

$$\tilde{\theta} = \arg \min_{\theta} \sum_{\mathbf{x} \in D} (I_1(\mathbf{f}(\mathbf{x}; \theta)) - I_2(\mathbf{x}))^2 \quad (3.10)$$

The algorithm is developed from a first-order Taylor expansion of  $I_1(\mathbf{f}(\mathbf{x}, \theta))$  with respect to  $\theta$ . In a open neighborhood  $V$  of  $\theta_0 \in \mathbb{R}$  we have, assuming  $I_1(\mathbf{f}(\mathbf{x}, \theta))$  is differentiable in  $\Omega \times V$ ,

$$I_1(\mathbf{f}(\mathbf{x}, \theta)) = I_1(\mathbf{f}(\mathbf{x}, \theta_0)) + \nabla I_1(\mathbf{x}, \theta_0) J_{\mathbf{f}}(\mathbf{x}, \theta_0) \mathbf{h} + o(\|\mathbf{h}\|^2), \quad (3.11)$$

where  $\mathbf{h} = \theta - \theta_0$ ,  $\nabla I_1$  is the spatial gradient of  $I_1$  written as a row vector, and  $J_{\mathbf{f}}$  is the Jacobian of  $\mathbf{f}$  with respect to  $\theta$ :

$$J_{\mathbf{f}} = \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_n} \\ \frac{\partial f_2}{\partial \theta_1} & \cdots & \frac{\partial f_2}{\partial \theta_n} \end{pmatrix} \quad (3.12)$$

Dropping the little  $o$  remainder, the objective function to minimize following this expansion is:

$$E(\mathbf{x}, \theta) = \sum_{\mathbf{x} \in D} (I_1(\mathbf{f}(\mathbf{x}, \theta_0)) + \nabla I_1(\mathbf{x}, \theta_0) J_{\mathbf{f}}(\mathbf{x}, \theta_0) \mathbf{h} - I_2(\mathbf{x}))^2 \quad (3.13)$$

Therefore, the first-order Taylor expansion has done two things: (1) the objective function turns into a linear equation in  $\mathbf{h}$  and, (2) viewing  $\theta_0$  as a current estimate, its minimization turns into iterations which consist at each step of determining an update  $\mathbf{h}$  by solving a linear system of equation by least squares. The scheme involves “warping” the image, i.e., evaluating  $I_1$  at the points of grid  $D$  transformed by  $\mathbf{f}$ . In general, this involves interpolating the image  $I_1$ . The original paper [11] mentions solving for  $\mathbf{h}$  using the least squares solution analytic expression, which involves matrix inversion, but other numerical schemes are more efficient, for instance the singular value decomposition method [81], and others which were investigated in the context of the Lucas-Kanade image registration algorithm [80]. The main weakness of the Lucas-Kanade formulation is its lack of regularization. In the Horn and Schunck formulation, which we review next, regularization of the flow field is fundamental.

### 3.4 The Horn and Schunck Algorithm

We recall the Horn and Schunck optical flow estimation functional [10] for an image sequence  $I : (x, y, t) \in \Omega \times ]0, T[ \mapsto I(x, y, t) \in \mathbf{R}^+$ :

$$\mathcal{E}(u, v) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \lambda \int_{\Omega} (\|\nabla u\|^2 + \|\nabla v\|^2) dx dy,$$

where  $I_x, I_y, I_t$  are the image spatiotemporal derivatives,  $\nabla u, \nabla v$  are the spatial gradients of the optical flow coordinates  $u, v$ , and  $\lambda$  is a constant factor to weigh the contribution of the two terms in the objective functional. The corresponding Euler-Lagrange equations are:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) - \lambda \nabla^2 u &= 0 \\ I_y(I_x u + I_y v + I_t) - \lambda \nabla^2 v &= 0, \end{aligned} \quad (3.14)$$

with Neumann boundary conditions

$$\frac{\partial u}{\partial \mathbf{n}} = 0, \quad \frac{\partial v}{\partial \mathbf{n}} = 0, \quad (3.15)$$

where  $\frac{\partial}{\partial \mathbf{n}}$  designates differentiation in the direction of the normal  $\mathbf{n}$  to the boundary of the image domain  $\Omega$ , and  $\nabla^2$  denotes the Laplacian operator.



### 3.4.1 Discretization

Let  $D$  be a unit-spacing grid over  $\Omega$  with the grid points indexed left-to-right and top-down by the integers  $\{1, 2, \dots, N\}$ . For all grid point indices  $i \in \{1, 2, \dots, N\}$ , a discrete approximation of the Euler-Lagrange Equations Eq. (3.14) is :

$$\begin{aligned} I_{xi}^2 u_i + I_{xi} I_{yi} v_i + I_{xi} I_{ti} - \lambda \sum_{j \in \mathcal{N}_i} (u_j - u_i) &= 0 \\ I_{yi} I_{xi} u_i + I_{yi}^2 v_i + I_{yi} I_{ti} - \lambda \sum_{j \in \mathcal{N}_i} (v_j - v_i) &= 0, \end{aligned} \quad (3.16)$$

where  $\lambda$  has absorbed the averaging constant of the Laplacian approximation;  $(u_i, v_i) = (u, v)_i$  is the optical flow vector at grid point  $i$ ;  $I_{xi}, I_{yi}, I_{ti}$  are the spatiotemporal derivatives  $I_x, I_y, I_t$  evaluated at  $i$ ; and  $\mathcal{N}_i$  is the set of indices of the neighbors of  $i$  for some neighborhood system. For the 4-neighborhood, for instance,  $card(\mathcal{N}_i) < 4$  for pixels on the boundary of  $D$  and  $card(\mathcal{N}_i) = 4$  for interior pixels. By accounting for the cardinality of  $\mathcal{N}_i$ , the approximation of the Laplacian in Eq. (3.16) is consistent with the Neumann boundary conditions Eq. (3.15) because it is equivalent to considering neighbors  $j$  of  $i$  outside the image domain but giving these the same flow vector as  $i$ . This is sometime called mirroring.

Re-arranging terms in Eq. (3.16), we have the following linear system of equations, for  $i \in \{1, \dots, N\}$ :

$$(S) \begin{cases} (I_{xi}^2 + \lambda c_i) u_i + I_{xi} I_{yi} v_i - \lambda \sum_{j \in \mathcal{N}_i} u_j = -I_{xi} I_{ti} \\ I_{xi} I_{yi} u_i + (I_{yi}^2 + \lambda c_i) v_i - \lambda \sum_{j \in \mathcal{N}_i} v_j = -I_{yi} I_{ti}, \end{cases}$$

where  $c_i = card(\mathcal{N}_i)$ . Let  $\mathbf{z} = (z_1, \dots, z_{2N})^t \in \mathbf{R}^{2N}$  be the vector defined by

$$z_{2i-1} = u_i, \quad z_{2i} = v_i, \quad i \in \{1, \dots, N\}. \quad (3.17)$$

Also, let  $\mathbf{b} = (b_1, \dots, b_{2N})^t \in \mathbf{R}^{2N}$  be defined by

$$b_{2i-1} = -I_{xi} I_{ti}, \quad b_{2i} = -I_{yi} I_{ti}, \quad i \in \{1, \dots, N\}. \quad (3.18)$$

In matrix form, linear system (S) is:

$$\mathbf{A} \mathbf{z} = \mathbf{b}, \quad (3.19)$$

where  $\mathbf{A}$  is the  $2N \times 2N$  matrix the elements of which are, for  $i \in \{1, \dots, N\}$ :

$$\begin{aligned} \mathbf{A}_{2i-1, 2i-1} &= I_{xi}^2 + \lambda c_i, & \mathbf{A}_{2i, 2i} &= I_{yi}^2 + \lambda c_i, \\ \mathbf{A}_{2i-1, 2i} &= I_{xi} I_{yi}, & \mathbf{A}_{2i, 2i-1} &= I_{xi} I_{yi}, \\ \mathbf{A}_{2i-1, 2j-1} &= \mathbf{A}_{2i, 2j} = -\lambda, & j &\in \mathcal{N}_i, \end{aligned} \quad (3.20)$$

all other elements being equal to zero. System Eq. (3.19) is a large scale sparse system of linear equations. Such systems are best solved by iterative algorithms such as the Jacobi and Gauss-Seidel iterations [82, 83] which we will give next. We will assume that  $\mathbf{A}$  is non-singular.

### 3.4.2 Gauss-Seidel and Jacobi Iterations

One can show that matrix  $\mathbf{A}$  is positive definite [13]. This implies that the point-wise and block-wise Gauss-Seidel iterations to solve system Eq. (3.19) will converge. This is a standard result in numerical linear algebra [82, 83]. For the  $2 \times 2$  block division of matrix  $\mathbf{A}$ , the Gauss-Seidel iterations are [13], for all  $i \in \{1, \dots, N\}$ :

$$u_i^{k+1} = \frac{I_{yi}^2 + \lambda c_i}{c_i(I_{xi}^2 + I_{yi}^2) + \lambda c_i^2} \left( \sum_{j \in \mathcal{N}_i; j < i} u_j^{k+1} + \sum_{j \in \mathcal{N}_i; j > i} u_j^k \right) - \frac{I_{xi} I_{yi}}{c_i(I_{xi}^2 + I_{yi}^2) + \alpha c_i^2} \left( \sum_{j \in \mathcal{N}_i; j < i} v_j^{k+1} + \sum_{j \in \mathcal{N}_i; j > i} v_j^k \right) - \frac{I_{xi} I_{ti}}{I_{xi}^2 + I_{yi}^2 + \lambda c_i} \quad (3.21)$$

$$v_i^{k+1} = \frac{-I_{xi} I_{yi}}{c_i(I_{xi}^2 + I_{yi}^2) + \lambda c_i^2} \left( \sum_{j \in \mathcal{N}_i; j < i} u_j^{k+1} + \sum_{j \in \mathcal{N}_i; j > i} u_j^k \right) + \frac{I_{xi}^2 + \lambda c_i}{c_i(I_{xi}^2 + I_{yi}^2) + \lambda c_i^2} \left( \sum_{j \in \mathcal{N}_i; j < i} v_j^{k+1} + \sum_{j \in \mathcal{N}_i; j > i} v_j^k \right) - \frac{I_{yi} I_{ti}}{I_{xi}^2 + I_{yi}^2 + \lambda c_i}$$

Horn and Schunck [10] solve system Eq. (3.19) with the  $2 \times 2$  block-wise Jacobi method. The iterations are:

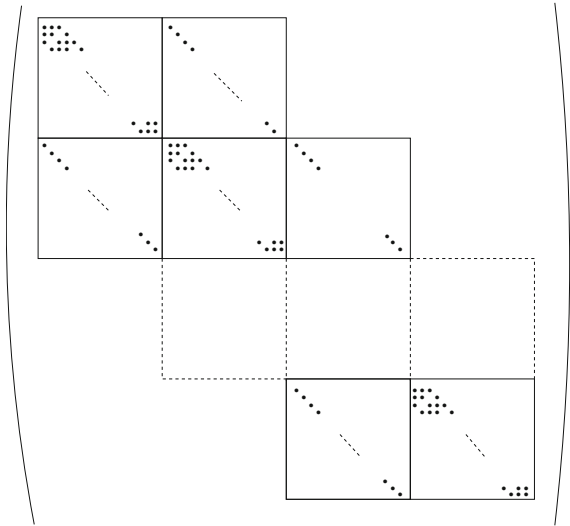
$$u_i^{k+1} = \frac{I_{yi}^2 + \lambda c_i}{c_i(I_{xi}^2 + I_{yi}^2) + \lambda c_i^2} \sum_{j \in \mathcal{N}_i} u_j^k - \frac{I_{xi} I_{yi}}{c_i(I_{xi}^2 + I_{yi}^2) + \lambda c_i^2} \sum_{j \in \mathcal{N}_i} v_j^k - \frac{I_{xi} I_{ti}}{I_{xi}^2 + I_{yi}^2 + \lambda c_i} \quad (3.22)$$

$$v_i^{k+1} = \frac{-I_{xi} I_{yi}}{c_i(I_{xi}^2 + I_{yi}^2) + \alpha c_i^2} \sum_{j \in \mathcal{N}_i} u_j^k + \frac{I_{xi}^2 + \lambda c_i}{c_i(I_{xi}^2 + I_{yi}^2) + \lambda c_i^2} \sum_{j \in \mathcal{N}_i} v_j^k - \frac{I_{yi} I_{ti}}{I_{xi}^2 + I_{yi}^2 + \lambda c_i}$$

The fact that matrix  $\mathbf{A}$  is symmetric positive definite is not sufficient to imply that the Jacobi iterations converge. However, it can be shown directly that they do. This has been done using a vector norm in  $\mathbb{R}^{2N}$  adapted to the special structure of the linear system (S) [13].

The differences between the Gauss-Seidel and the Jacobi methods are well known: the Jacobi method does the update for all points of the image domain grid and then uses the updated values at the next iteration, whereas the Gauss-Seidel method uses

**Fig. 3.2** Data matrix  $\mathbf{A}$  is block tridiagonal. The dots represent possibly nonzero elements. For an  $n \times n$  discrete image, the blocks are  $2n \times 2n$ . The block tridiagonal form comes from the fact that points with index  $Kn$ ,  $1 \leq K \leq n$ , do not have a right-side neighbor, and those with index  $Kn + 1$ ,  $0 \leq K \leq n - 1$ , do not have a left-side neighbor



the updated values as soon as they are available and, as a result, can be more efficient than the Jacobi method in sequential computations. However, in contrast with the Gauss-Seidel iterations, Jacobi's can be performed in parallel for all pixels, which can result in a very fast hardware implementation [14, 15]. As to memory storage, the Jacobi method requires at each iteration the  $2N$  values of the previous iteration in memory store. With the Gauss-Seidel iterations Eq. (3.22), only a few of these values are stored.

There is a remarkable block division which makes matrix  $\mathbf{A}$  block tridiagonal (Fig. 3.2). Combined with the property that  $\mathbf{A}$  is symmetric positive definite, this characteristic affords efficient resolution of the corresponding linear system [82]. For an  $n \times n$  discrete image, the blocks are  $2n \times 2n$ . The block tridiagonal form is due to the fact that points with index  $Kn$ ,  $1 \leq K \leq n$ , do not have a neighbor on the right, and those with index  $Kn + 1$ ,  $0 \leq K \leq n - 1$ , do not have a neighbor on the left. The block-wise iterations for a block tridiagonal symmetric positive definite matrix, i.e., the iterations corresponding to the tridiagonal block decomposition (Fig. 3.2), converge for both the Jacobi and the Gauss-Seidel implementations [82]. The spectral radius of the Gauss-Seidel matrix is equal to the square of the spectral radius of the Jacobi matrix, which signifies that the Gauss-Seidel implementation is in this case much faster than the Jacobi. The readers interested in the details may refer to [13].

### 3.4.3 Evaluation of Derivatives

Horn and Schunck have used approximations of the image spatial and temporal derivatives as averages of forward first differences. From two consecutive  $n \times n$  images  $I^1$  and  $I^2$  the formulas are:

$$\begin{aligned}
I_{xi} &= \frac{1}{4}(I_{i+1}^1 - I_i^1 + I_{i-n+1}^1 - I_{i-n}^1 + I_{i+1}^2 - I_i^2 + I_{i-n+1}^2 - I_{i-n}^2) \\
I_{yi} &= \frac{1}{4}(I_{i-n}^1 - I_i^1 + I_{i-n+1}^1 - I_{i+1}^1 + I_{i-n}^2 - I_i^2 + I_{i-n+1}^2 - I_{i+1}^2) \quad (3.23) \\
I_{ii} &= \frac{1}{4}(I_i^2 - I_i^1 + I_{i+1}^2 - I_{i+1}^1 + I_{i-n}^2 - I_{i-n}^1 + I_{i-n+1}^2 - I_{i-n+1}^1),
\end{aligned}$$

for  $i = 1, \dots, n^2$ . Alternatively, the spatial derivatives can be estimated using central differences. Using central differences to compute the temporal derivatives would not be consistent with the in-between consecutive frames velocities to be estimated because it would require using the frames preceding and following the current, rather than consecutive frames.

Points in the formulas which fall outside the image domain are often given the index of the image wrapped around on its boundary to form a (digital) torus or, alternatively, boundary points are simply given the spatiotemporal derivative values of an immediate interior neighbor.

### 3.4.4 Ad hoc Variations to Preserve Motion Boundaries

As alluded to in the introduction, the single serious drawback of the Horn and Schunck method is that the quadratic (Tikhonov) regularization it uses ignores motion boundaries which it smooths out as a result. This technically translates into the occurrence of the isotropic Laplacian operator in the Euler-Lagrange equations. The original study of Horn and Schunck approximates the discrete Laplacian  $\Delta^2 w$  as:

$$\Delta^2 w \propto \bar{w} - w, \quad (3.24)$$

where  $w$  stands for either  $u$  or  $v$  and  $\bar{w}$  is a weighted neighborhood average of  $w$  according to the weights in Fig. 3.3.

This Laplacian approximation is used explicitly in their discretization of the Euler-Lagrange equations to arrive at the following form of the iterations to compute optical flow, where  $\lambda$  has absorbed the coefficient of proportionality:

$$\begin{aligned}
u_i^{k+1} &= \bar{u}_i^k - I_{xi} \frac{I_{xi} \bar{u}_i^k + I_{yi} \bar{v}_i^k + I_t}{\lambda + I_{xi}^2 + I_{yi}^2} \\
v_i^{k+1} &= \bar{v}_i^k - I_{yi} \frac{I_{xi} \bar{u}_i^k + I_{yi} \bar{v}_i^k + I_t}{\lambda + I_{xi}^2 + I_{yi}^2}
\end{aligned} \quad (3.25)$$

Boundary conditions aside, the average  $\bar{w}$  is computed according to the set of fixed weights in Fig. 3.3. This suggests that one can be more general and approximate the operator by spatially variant filters, rather than a fixed weighted average, with the purpose of preserving motion boundaries, i.e., dampening blurring at motion

discontinuities. In such a case, iterations Eq.(3.25) are executed with:

$$\begin{aligned} \bar{u}_i^k &= g\left(\{u_j^k\} : j \in \mathcal{N}_i\right) \\ \bar{v}_i^k &= g\left(\{v_j^k\} : j \in \mathcal{N}_i\right), \end{aligned} \tag{3.26}$$

with filters  $g$  such as those suggested in [84]. These can be dependent of the image or on the flow itself:

**Image-based adaptive average:** Under the assumption that the image of environmental objects is smooth except at the projection of their occluding boundaries, flow edges and image edges will coincide, justifying an intensity-based filter of the form:

$$g\left(\{w_j^k\} : j \in \mathcal{N}_i\right) = \sum_{j \in \mathcal{N}_i} \alpha_j w_j, \tag{3.27}$$

where coefficients  $\alpha_j$  are commensurate with the image contrast between  $i$  and  $j$ , for instance by using:

$$\alpha_j = \frac{\frac{1}{1+|I_j-I_i|}}{\sum_{j \in \mathcal{N}_i} \frac{1}{1+|I_j-I_i|}} \tag{3.28}$$

In general, of course, flow discontinuities are only a subset of intensity edges so that smoothing of the flow field according to Eq.(3.28) will follow the image intensity structure rather than the structure of the motion field and, as a result, can cause undesirable artefacts.

**Optical flow-based adaptive average:** The coefficients of a flow-based version of the image-based filter would be:

**Fig. 3.3** The discrete Laplacian  $\Delta^2 w$  can be written as  $\Delta^2 w \propto \bar{w} - w$ , where  $w$  stands for either  $u$  or  $v$  and  $\bar{w}$  is a weighted neighborhood average of  $w$  using the weights above as suggested in the original investigation of optical flow estimation by the Horn and Schunck method

$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$
$\frac{1}{6}$	-1	$\frac{1}{6}$
$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$

$$\alpha_j = \frac{\left(\frac{1}{1+|w_j-w_i|}\right)^\beta}{\sum_{j \in \mathcal{N}_i} \left(\frac{1}{1+|w_j-w_i|}\right)^\beta} \quad (3.29)$$

where  $w$  stands for either of the optical flow coordinates and  $\beta > 1$ . The purpose of exponent  $\beta$  is to discern better the coefficients values when the range of the flow coordinates is small. This filter is expected to dampen smoothing across motions discontinuities while stressing it along.

**Median filtering:** Here, filter  $g$  at pixel  $i$  would be the median of the current flow estimates in the neighborhood  $\mathcal{N}_i$  of  $i$ . At a flow discontinuity, median filtering is more likely to yield a value representative of the values on a single side of the discontinuity. A reasonable alternative consists of averaging the values of the flow velocity in  $\mathcal{N}_i$  which are above or below the median, whichever are more homogeneous. In the event of a flow edge at  $i$ , these values would most likely come from pixels on a single side of the edge.

**Modulating the weight coefficient  $\lambda$ :** The ad hoc variations above use digital approximations of the Laplacian which adjust to the local structure of the image or of the flow field at any stage of its estimation by the Horn and Schunck iterations, in the hope that this structure is actually indicative of the actual flow discontinuities. Along this vein of thought, one can also look at varying the weighing coefficient  $\lambda$  during the iterations depending on the structure of the image or the current flow field [85]. Since smoothing increases with  $\lambda$ , the rationale is that the value of this coefficient should be low at suspected motion boundaries and high elsewhere. For instance the study in [85] uses thresholds on  $\|\nabla I\|^2$  and  $\|\nabla u\|^2 + \|\nabla v\|^2$  to decide whether to smooth sufficiently, according to some threshold  $\lambda_h$ , when neither of these gradient norms is high or, instead, inhibit smoothing using a small coefficient  $\lambda_s$ .

Although ad hoc approximations of key variables in Eq. (3.25), such as the Laplacian or the weight coefficient, can produce sharper motion boundaries at practically no additional computational expense, there have been no extensive experimental verification which would allow a definite conclusion about their effectiveness compared to other boundary preserving formulations such as the ones we will describe next. These are formal methods which aim at preserving motion discontinuities by referencing motion edges via boundary length or by using a boundary preserving regularization function in the objective functional. We will describe both an image-based and a flow-based boundary preserving regularization function.

### 3.5 Deriche–Aubert–Kornprobst Method

The Laplacian operator which appears in the Euler-Lagrange equations associated with Eq. (3.14) causes smoothing, and blurring thereof, across motion boundaries. To circumvent the problem, the study in [37] proposed to investigate regularization functions  $g$  in the following generalization of the Horn and Schunck functional:

$$E(u, v) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \lambda \int_{\Omega} (g(\|\nabla u\|) + g(\|\nabla v\|)) dx dy, \quad (3.30)$$

such that motion boundaries are preserved. With  $g(z) = z^2$ , Eq. (3.30) reduces to the Horn and Schunck functional Eq. (3.14). The purpose of the analysis in [37, 38] was to determine  $g$  from conditions that would ensure isotropic smoothing of motion where it varies smoothly and allow smoothing along motion boundaries while inhibiting or dampening it across. The analysis is summarized in the following.

The Euler-Lagrange equations corresponding to Eq. (3.30) are:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) - \frac{\lambda}{2} \operatorname{div} \left( g'(\|\nabla u\|) \frac{\nabla u}{\|\nabla u\|} \right) &= 0 \\ I_y(I_x u + I_y v + I_t) - \frac{\lambda}{2} \operatorname{div} \left( g'(\|\nabla v\|) \frac{\nabla v}{\|\nabla v\|} \right) &= 0, \end{aligned} \quad (3.31)$$

where  $\operatorname{div}$  is the divergence operator and  $g'$  is the first derivative of  $g$ . The corresponding Neumann boundary conditions are:

$$\begin{aligned} \frac{g'(\|\nabla u\|)}{\|\nabla u\|} \frac{\partial u}{\partial \mathbf{n}} &= 0 \\ \frac{g'(\|\nabla v\|)}{\|\nabla v\|} \frac{\partial v}{\partial \mathbf{n}} &= 0, \end{aligned} \quad (3.32)$$

where  $\mathbf{n}$  is the unit normal vector to the boundary  $\partial\Omega$  of the image domain  $\Omega$ , and  $\partial/\partial\mathbf{n}$  is the derivative operator in the direction of  $\mathbf{n}$ .

For  $w \in \{u, v\}$ , i.e., where  $w$  stands for either of the optical flow components, consider at each point a local orthonormal direct coordinate system  $(\eta, \xi)$  defined by unit vectors  $\frac{\nabla w}{\|\nabla w\|}$  and its (counter clockwise) orthogonal unit vector  $\left(\frac{\nabla w}{\|\nabla w\|}\right)^\perp$ . In this reference system, the divergence terms in Eq. (3.31) are written:

$$\operatorname{div} \left( \frac{g'(\|\nabla w\|)}{\|\nabla w\|} \nabla w \right) = \frac{g'(\|\nabla w\|)}{\|\nabla w\|} w_{\xi\xi} + g''(\|\nabla w\|) w_{\eta\eta} \quad (3.33)$$

In a region where  $w$  is homogeneous, i.e., where  $\|\nabla w\|$  is small, we want  $g$  to allow smoothing in both orthogonal directions  $\eta$  and  $\xi$ , and in the same manner (isotropy). Considering Eqs. (3.31) and (3.33), the conditions to impose are:

$$\begin{aligned} \lim_{s \rightarrow 0} g''(s) &= g''(0) > 0 \\ \lim_{s \rightarrow 0} \frac{g'(s)}{s} &= g''(0) \end{aligned} \quad (3.34)$$

At the limit when  $\|\nabla w\| \rightarrow 0$ , we have:

$$\operatorname{div} \left( \frac{g'(\|\nabla w\|)}{\|\nabla w\|} \nabla w \right) = g''(0)(w_{\eta\eta} + w_{\xi\xi}) = g''(0)\nabla^2 w$$

Therefore, the Euler-Lagrange equations in this case, when  $\|\nabla w\| \rightarrow 0$ , would be:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) &= \frac{\lambda}{2} g''(0) \nabla^2 u \\ I_y(I_x u + I_y v + I_t) &= \frac{\lambda}{2} g''(0) \nabla^2 v, \end{aligned} \quad (3.35)$$

with Neumann boundary conditions

$$\frac{\partial u}{\partial \mathbf{n}} = 0, \quad \frac{\partial v}{\partial \mathbf{n}} = 0. \quad (3.36)$$

These equations are those of the Horn and Schunck formulation, which is what we want.

When  $\nabla w$  is large, as it would be at motion boundaries, we want to smooth  $w$  along  $\xi$  but inhibit smoothing in the orthogonal direction, i.e., along  $\eta$ . The conditions to set are:

$$\begin{aligned} \lim_{s \rightarrow \infty} g''(s) &= 0 \\ \lim_{s \rightarrow \infty} \frac{g'(s)}{s} &= \beta > 0, \end{aligned} \quad (3.37)$$

and the divergence term at the limit when  $\|\nabla w\| \rightarrow \infty$  would be:

$$\operatorname{div} \left( \frac{g'(\|\nabla w\|)}{\|\nabla w\|} \nabla w \right) = \beta w_{\xi\xi}. \quad (3.38)$$

However, both conditions in Eq. (3.37) cannot be satisfied simultaneously [37, 38]. Instead, the following weaker conditions can be imposed:

$$\begin{aligned} \lim_{s \rightarrow \infty} g''(s) &= \lim_{s \rightarrow \infty} \frac{g'(s)}{s} = 0 \\ \lim_{s \rightarrow \infty} \frac{g''(s)}{g'(s)} &= 0. \end{aligned} \quad (3.39)$$

Accordingly, diffusion is inhibited in both directions at the limit, when  $\|\nabla w\| \rightarrow \infty$ , but is otherwise dampened more in direction  $\eta$  than  $\xi$ , i.e, smoothing will be dampened more across motion boundaries than along. There are several functions satisfying conditions Eqs. (3.34) and (3.39),  $g(s) = 2\sqrt{1+s^2} - 2$  (Aubert), for instance, and the ones shown in Table 3.1.

**Table 3.1** Boundary preserving functions for the estimation of optical flow

	$g(s)$
Aubert	$2\sqrt{1+s^2} - 2$
Geman and Reynolds	$\frac{s}{1+s^2}$
Perona-Malik	$\log(1+s^2)$
Green	$2 \log(\cosh(s))$



A discretization of the Euler-Lagrange equations gives a large scale sparse system of nonlinear equations. Instead of solving directly this system, the study in [37, 38] proposed a more efficient implementation using the half-quadratic minimization algorithm applied to the following functional, the change from the original functional being justified by a duality theorem [38]:

$$E(u, v, b_1, b_2) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \lambda \int_{\Omega} \left( b_1 \|\nabla u\|^2 + b_2 \|\nabla v\|^2 + \psi(b_1) + \psi(b_2) \right) dx dy \quad (3.40)$$

Two new functions,  $b_1(x, y)$  and  $b_2(x, y)$ , called auxiliary variables, appear in this functional. Also appearing is a function  $\psi$ , convex and decreasing, related implicitly to  $g$  and such that, for every fixed  $s$ , the value of  $b$  which minimizes  $bs^2 + \psi(b)$  is given by

$$b = \frac{g'(s)}{2s} \quad (3.41)$$

This result is the basis of the half-quadratic greedy minimization algorithm which, after initialization, repeats two consecutive steps until convergence. Each iteration performs a minimization with respect to  $u, v$  with  $b_1, b_2$  assumed constant followed by a minimization with respect to  $b_1, b_2$  with  $u, v$  assumed constant.

Minimization with respect to  $u, v$ , with  $b_1, b_2$  considered constant, consists of minimizing the following functional:

$$\int_{\Omega} (I_x u + I_y v + I_t)^2 + \lambda \left( b_1 \|\nabla u\|^2 + b_2 \|\nabla v\|^2 \right) dx dy \quad (3.42)$$

The corresponding Euler-Lagrange equations are:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) &= \lambda \operatorname{div}(b_1 \nabla u) \\ I_y(I_x u + I_y v + I_t) &= \lambda \operatorname{div}(b_2 \nabla v), \end{aligned} \quad (3.43)$$

with Neumann boundary conditions  $\partial u / \partial \mathbf{n} = \partial v / \partial \mathbf{n} = 0$ . Discretization of the equations yields a large scale sparse linear system of equations which can be solved efficiently with the Gauss-Seidel or the Jacobi method. The divergence terms in Eq. (3.43) can be discretized as in [49].

The minimization with respect to  $b_1, b_2$ , with  $u, v$  considered constant, consists of minimizing the functional:

$$\int_{\Omega} \left( b_1 \|\nabla u\|^2 + b_2 \|\nabla v\|^2 + \psi(b_1) + \psi(b_2) \right) dx dy \quad (3.44)$$

The unique solution is given analytically following Eq. (3.41):

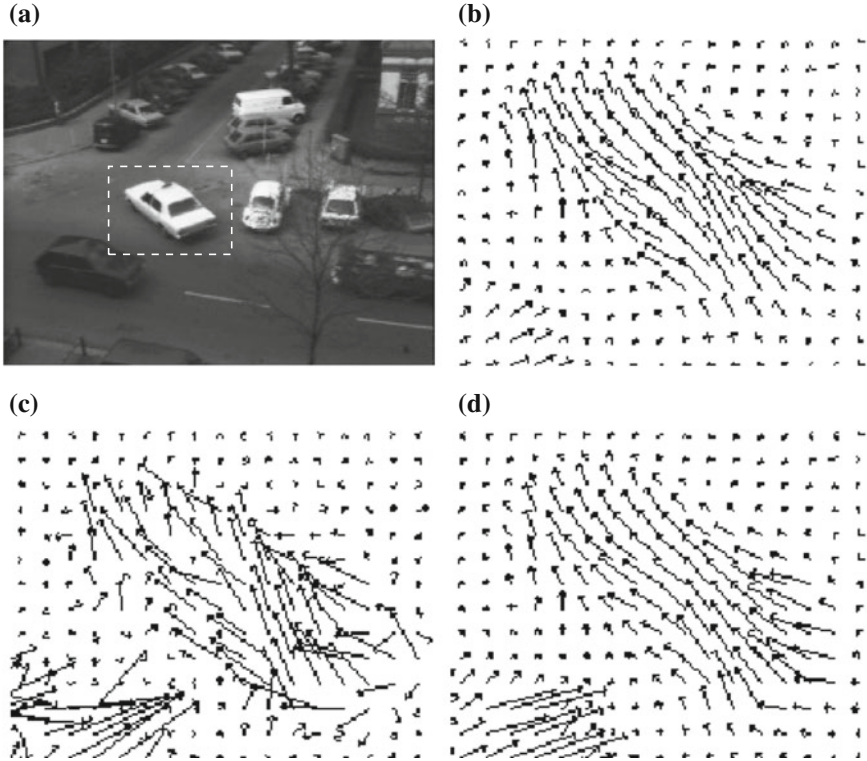
$$\begin{aligned}
 b_1 &= \frac{g'(\|\nabla u\|)}{2\|\nabla u\|} \\
 b_2 &= \frac{g'(\|\nabla v\|)}{2\|\nabla v\|}
 \end{aligned}
 \tag{3.45}$$

The half-quadratic algorithm to minimize Eq. (3.40) can be summarized as follows:

1. Initialize  $b_1, b_2$
2. Repeat until convergence
  - a. Minimize with respect to  $u, v$  using Jacobi (or Gauss-Seidel) iterations to solve the linear system of equations corresponding to the discretized Eq. (3.43).
  - b. Minimize with respect to  $b_1, b_2$  using Eq. (3.45) [ $b_1 = \frac{g'(\|\nabla u\|)}{2\|\nabla u\|}$ ,  $b_2 = \frac{g'(\|\nabla v\|)}{2\|\nabla v\|}$ ]

**Example:** This example (courtesy of R.Deriche) uses the Hamburg Taxi sequence of a street intersection scene (from Karlsruhe University, Germany, Institut für Algorithmen und Kognitive Systeme, [http://i21www.ira.uka.de/image\\_sequences/](http://i21www.ira.uka.de/image_sequences/)): Fig. 3.4a shows one of the two consecutive images used. The other figures contain a graphical display of the flow field in the rectangular zoom window drawn in (a) (which includes the white car in the center of the intersection and a small portion of the dark-coloured car next to it): Methods of (b) Horn and Schunck, (c) Lucas-Kanade, and (d) Deriche-Aubert-Kornprobst. Visual inspection reveals a motion smoothing spread in (b) and a lack of spatial regularization in (c). In (d) the smooth motion field is well confined to the moving cars as a result of discontinuity preserving smoothness regularization.

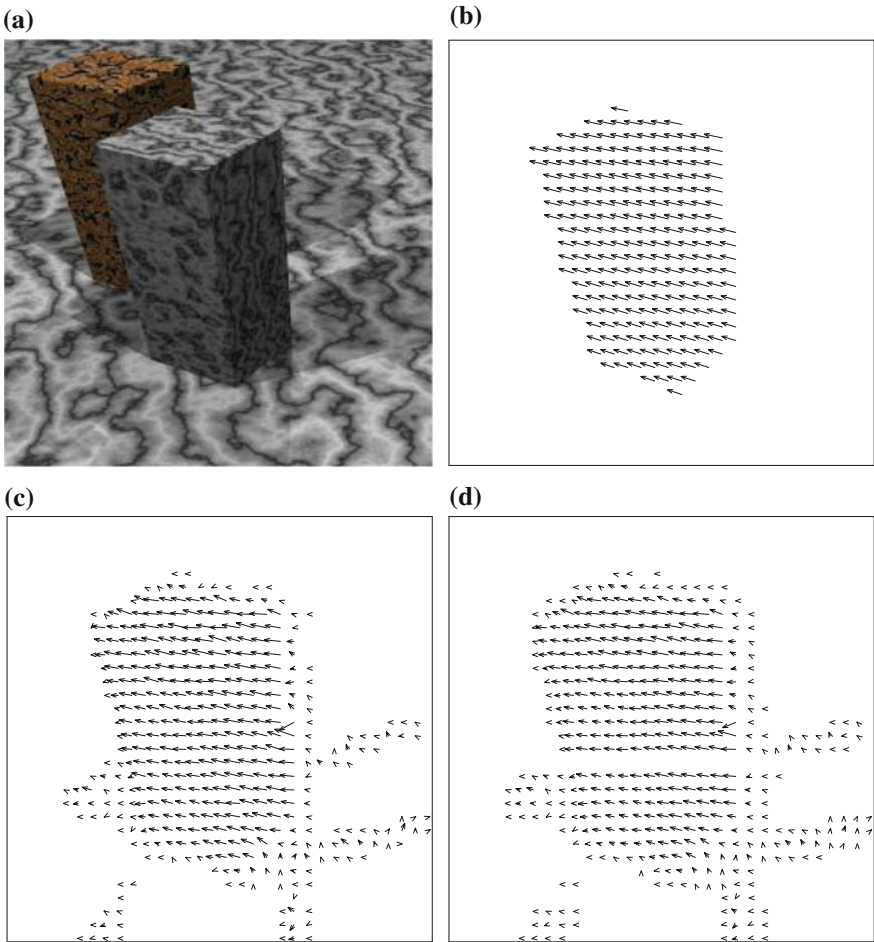
**Example:** This other example uses the synthetic sequence depicted in Fig. 3.5a (*Marbled blocks* sequence from Karlsruhe University, Germany, Institut für Algorithmen und Kognitive Systeme). The camera and the block on the left do not move. The block on the right moves away to the left. The images had noise added. The texture variation is weak at the top edges of the blocks. Depth, and image motion thereof, varies sharply at the blocks boundaries not in contact with the floor. The blocks cast shadows which display apparent motion. The ground truth optical flow vectors are displayed in Fig. 3.5b. Vectors computed with the method of Horn and Schunck and of Deriche, Aubert, Kornprobst are displayed in Fig. 3.5c and d, respectively. The average errors per pixel in magnitude (pixels) and direction (degrees) are (0.142, 5.095) and (0.130, 4.456) for the Horn and Schunck and the Aubert, Deriche, Kornprobst methods, respectively [61]. The better performance of the latter scheme is likely due to the better handling of motion discontinuities as a visual inspection tends to corroborate.



**Fig. 3.4** Optical flow estimation on the Hamburg Taxi sequence (courtesy of Deriche, Aubert, and Kornprobst): **a** One of the two consecutive images used. A graphical display of the flow field in the rectangular window shown in **a**, by the methods of **b** Horn and Schunck, **c** Lucas-Kanade, and **d** Deriche, Aubert, and Kornprobst. This last method produces a smooth field confined to the moving objects (the white car and part of the dark car) as a result of discontinuity preserving smoothness regularization

### 3.6 Image-Guided Regularization

Consider from a probabilistic viewpoint the problem of estimating optical flow from two consecutive images  $I_1$  and  $I_2$ . This consists of maximizing the posterior probability  $P(W|I_1, I_2)$  over the space of all possible optical flow fields  $W$ . This probability is proportional to the product  $P(I_2|W, I_1)P(W|I_1)$ . The first term,  $P(I_2|W, I_1)$ , is a term of conformity of  $W$  to the data because it is the likelihood that connects  $I_2$  to  $I_1$  via  $W$ . The second term,  $P(W|I_1)$ , is a prior on  $W$  which exhibits a partial dependence on data through the conditioning on  $I_1$ . This dependence is often ignored and the conditioning on  $I_1$  is removed, resulting in a prior independent of any observation. This is equivalent to imposing statistical independence of  $W$  and  $I_1$ . However, the dependence is genuine because motion edges often occur at image intensity edges [27]. Therefore, its inclusion in a prior, or a regularization term in



**Fig. 3.5** Optical flow estimation on the Marbled block sequence: **a** the first of the two images used, **b** ground truth optical flow, and optical flow by the method of **c** Horn and Schunck, **d** Deriche, Aubert, Kornprobst. This last method produces a smooth field confined to the moving objects as a result of discontinuity preserving smoothness regularization

energy based formulations, affords the opportunity to smooth the motion field without blurring its boundaries by allowing smoothing along the isophote, i.e., in the direction perpendicular to the image spatial gradient, and inhibiting or dampening it across. This can be done via an appropriate gradient-dependent linear transformation  $\mathbf{A}(\nabla I)$  of the motion field in the prior/regularization term. Here following are two possible formulations [16, 19, 20].

### 3.6.1 The Oriented-Smoothness Constraint

The following functional was investigated in [16, 86]:

$$\mathcal{E}(u, v) = \int_{\Omega} (I_1(x - u, y - v) - I_2(x, y))^2 dx dy \quad (3.46)$$

$$+ \lambda \int_{\Omega} \left( \nabla u^T \mathbf{A}(\nabla I_1) \nabla u + \nabla v^T \mathbf{A}(\nabla I_1) \nabla v \right) dx dy. \quad (3.47)$$

Matrix  $\mathbf{A}$  is defined as a function of the image partial derivatives by:

$$\mathbf{A}(\nabla I_1) = \frac{1}{\|\nabla I_1\|^2 + 2\mu^2} \left[ \begin{pmatrix} I_{1y} \\ -I_{1x} \end{pmatrix} (I_{1y} - I_{1x}) + \mu^2 \mathbf{I} \right], \quad (3.48)$$

where  $\mathbf{I}$  is the identity matrix and  $\mu$  a constant. The functional was later modified [87] to remove the peculiarity that motion is applied to  $I_2$  in the data term but to  $I_1$  in the regularization term.

An analysis in [20] determined an image-guided regularization matrix  $\mathbf{A}$  by imposing on it conditions which would cause smoothing along intensity edges but dampening it across. The analysis is as follows.

### 3.6.2 Selective Image Diffusion

Consider the following objective functional:

$$\mathcal{E}(W) = \int_{\Omega} (I_x u + I_y v + I_t)^2 dx dy + \lambda \int_{\Omega} (\|\mathbf{A} \nabla u\|^2 + \|\mathbf{A} \nabla v\|^2) dx dy, \quad (3.49)$$

where  $\mathbf{A} = \mathbf{A}(\nabla I)$  is a  $2 \times 2$  matrix which depends on the image structure via the image spatial gradient. Matrix  $\mathbf{A}$  must be chosen so as to allow smoothing at each point in the direction of the perpendicular to the image gradient, i.e., along the isophote, and dampen it in the direction of the gradient, i.e., perpendicular to the isophote. This can be done by imposing the following conditions on the eigenvalues  $\alpha_1, \alpha_2$  of  $\mathbf{A}$  [20]:

1. For  $\|\nabla I\| \neq 0$ ,  $\mathbf{x}_1 = \frac{\nabla I}{\|\nabla I\|}$ ,  $\mathbf{x}_2 = \left( \frac{\nabla I}{\|\nabla I\|} \right)^\perp$  are the unit eigenvectors corresponding to  $\alpha_1, \alpha_2$ ,
2.  $\alpha_2 = 1$
3.  $\alpha_1$  is a monotonically decreasing continuous function of  $\|\nabla I\|$  such that:  
 $\lim_{\|\nabla I\| \rightarrow 0} \alpha_1 = 1$  and  $\lim_{\|\nabla I\| \rightarrow \infty} \alpha_1 = 0$ .

Intuitively, the purpose of these conditions is as follows: the first condition says that the two orthogonal directions which should be considered for smoothing are those of the isophote and the image gradient; the second condition is to allow full smoothing along the isophote; the third condition stipulates that smoothing along the gradient direction is to be allowed only to a degree that decreases with the intensity edge strength, varying from full to no strength.

The Euler-Lagrange equations corresponding to Eq. (3.49) are:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) - \lambda \operatorname{div}(\mathbf{B} \nabla u) &= 0 \\ I_y(I_x u + I_y v + I_t) - \lambda \operatorname{div}(\mathbf{B} \nabla v) &= 0, \end{aligned} \quad (3.50)$$

where  $\mathbf{B} = \mathbf{A}^t \mathbf{A} + \mathbf{A} \mathbf{A}^t$ , with Neumann boundary conditions:

$$\begin{aligned} \mathbf{B} \nabla u \cdot \mathbf{n} &= 0 \\ \mathbf{B} \nabla v \cdot \mathbf{n} &= 0 \end{aligned} \quad (3.51)$$

Let  $\mathbf{P}$  be the  $2 \times 2$  orthogonal matrix  $\mathbf{P} = (\mathbf{x}_1, \mathbf{x}_2)$ , i.e., whose columns are  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , and let

$$\Lambda = \begin{pmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{pmatrix} \quad (3.52)$$

Using the first condition, we have, by definition,  $\mathbf{A} \mathbf{P} = \mathbf{P} \Lambda$ . Therefore,  $\mathbf{A} = \mathbf{P} \Lambda \mathbf{P}^{-1} = \mathbf{P} \Lambda \mathbf{P}^t$ , since  $\mathbf{P}$  is orthogonal. This gives, using the second condition ( $\alpha_2 = 1$ ),

$$\mathbf{A}(\nabla I) = \frac{1}{\|\nabla I\|^2} \begin{pmatrix} \alpha_1 I_x^2 + I_y^2 & (\alpha_1 - 1) I_x I_y \\ (\alpha_1 - 1) I_x I_y & I_x^2 + \alpha_1 I_y^2 \end{pmatrix} \quad (3.53)$$

Using the following  $\alpha_1$ , which satisfies the third condition,

$$\alpha_1 = \frac{1}{\left(1 + \frac{\|\nabla I\|^2}{\mu^2}\right)^{\frac{1}{2}}}, \quad (3.54)$$

where  $\mu$  is a parameter to modulate the strength of smoothing, we have:

$$\mathbf{B} = \frac{1}{\mu^2 \|\nabla I\|^2} \begin{pmatrix} \mu^2 + I_y^2 & -I_x I_y \\ -I_x I_y & \mu^2 + I_x^2 \end{pmatrix} \quad (3.55)$$

Assuming  $\|\nabla I\|$  is bounded on  $\Omega$ , this matrix is positive definite, which means that Eq. (3.50) are diffusion equations. To see intuitively that they realize the desired diffusion, note that where  $\|\nabla I\|$  is small,  $\alpha_1$  is close to 1 and, therefore,  $\mathbf{A}(\nabla I)$  is close to the identity, causing the regularization term in Eq. (3.49) to be close to the  $L^2$  norm and Eq. (3.50) to behave isotropically. When, instead,  $\|\nabla I\|$  is large,  $\mathbf{A}(\nabla I)$  approximates a projection onto the direction perpendicular to  $\nabla I$  and only the projection of  $\nabla u$  and  $\nabla v$  along that direction will contribute to the regularization.

Another way to see that we have the desired diffusion is by looking at the behaviour of Eq. (3.50) locally at each point, in a small neighborhood where  $\nabla I$  is constant and nonzero. In this neighborhood, consider the local coordinate system  $(\eta, \xi)$  according to the reference system defined by  $\frac{\nabla I}{\|\nabla I\|}, \left(\frac{\nabla I}{\|\nabla I\|}\right)^\perp$ . In this orthonormal reference system, we have  $\nabla u = (u_\eta, u_\xi)$  and  $\nabla v = (v_\eta, v_\xi)$ , and

$$\mathbf{B} = \begin{pmatrix} \alpha_1^2 & 0 \\ 0 & \alpha_2^2 \end{pmatrix}. \quad (3.56)$$

which gives the following local form of the divergence term, using  $\alpha_2 = 1$ :

$$\operatorname{div}(\mathbf{B}\nabla u) = \alpha_1^2 u_{\eta\eta} + u_{\xi\xi} \quad (3.57)$$

and, therefore, the local form of the Euler-Lagrange equations:

$$\begin{aligned} I_x(I_x u + I_y v + I_t) - \lambda(\alpha_1^2 u_{\eta\eta} + u_{\xi\xi}) &= 0 \\ I_y(I_x u + I_y v + I_t) - \lambda(\alpha_1^2 v_{\eta\eta} + v_{\xi\xi}) &= 0 \end{aligned} \quad (3.58)$$

It is clear from these equations that diffusion will occur along axis  $\xi$ , i.e. along the intensity edge and that it will be dampened along axis  $\eta$ , i.e., along the direction of the gradient. Since  $\alpha_1$  is a decreasing function of  $\|\nabla I\|$ , the degree of dampening will be commensurate with the edge strength. Parameter  $\mu$  in Eq. (3.54), although not essential, can be used to control how fast with respect to edge strength dampening occurs across edges.

The minimization of Eq. (3.49) can be done by the corresponding Euler-Lagrange descent equations [20], namely,

$$\begin{aligned} \frac{\partial u}{\partial \tau} &= -I_x(I_x u + I_y v + I_t) + \lambda \operatorname{div}(\mathbf{B}\nabla u) \\ \frac{\partial v}{\partial \tau} &= -I_y(I_x u + I_y v + I_t) + \lambda \operatorname{div}(\mathbf{B}\nabla v) \end{aligned} \quad (3.59)$$

One can also discretize the Euler-Lagrange equations Eq. (3.50). This would give a large scale sparse system of linear equations which can be solved efficiently by Gauss-Seidel or Jacobi iterations.

## 3.7 Minimum Description Length

A way to preserve motion discontinuities is to bring in the length of the discontinuity set in the regularization [30]. A boundary length term commonly appears in image segmentation functionals, first in the Mumford and Shah functional [32]. It is essential in the Leclerc's minimum description length (MDL) formulation [33] which we focus on in this section and transpose to optical flow estimation. The Leclerc's method can

be viewed as a discrete implementation of the Mumford-Shah functional [88]. It minimizes an objective function which assigns a code length to an image partition described according to a predefined “description language.”

Let  $I_0$  be an observed image with discrete domain  $D$  and  $I$  an approximation corresponding to a partition  $\mathcal{R} = \{R_k\}$  of the image domain  $D$  into regions where the image is modelled by a parametric model with parameters  $\{\theta_k\}$ . The Leclerc MDL criterion [33] to estimate the image underlying  $I_0$  is:

$$E(\mathcal{R}, \{\theta_k\}) = \frac{a}{2} \sum_k l_k - \sum_k \sum_{i \in R_k} \log_2 P(I_i | \theta_k) + \sum_k b_k, \quad (3.60)$$

where  $l_k$  is the length of the boundary of  $R_k$  in terms of the number of pixels it threads through,  $a$  is the bit cost of coding one edge element, and  $b_k$  is the bit cost of coding the (discrete) parameter vector of region  $R_k$ . The first term is the code length for the boundaries and the second for the image given the region parameters. The last term is the code length to describe the region models via their parameters; assuming equal code length for all regions, the term can be dropped from the criterion. For a piecewise constant description of  $I$  and quantized Gaussian noise, the criterion can be re-written as [33]:

$$E(I) = \frac{a}{2} \sum_{i \in D} \sum_{j \in \mathcal{N}_i} (1 - \delta(I_i - I_j)) + b \sum_{i \in D} \frac{(I_i - I_{0i})^2}{\sigma^2}, \quad (3.61)$$

where  $a \approx 2$  and  $b = \frac{1}{2 \log 2}$ ;  $\mathcal{N}_i$  is some fixed neighborhood of pixel  $i$ ; and

$$\delta(z) = \begin{cases} 1 & \text{for } z = 0 \\ 0 & \text{for } z \neq 0 \end{cases} \quad (3.62)$$

Energy Eq. (3.61) can be solved by a *continuation* scheme indexed by the standard deviation of a Gaussian substituted for  $\delta$ : Starting from an initial large value, the standard deviation is gradually lowered and, at each step, a solution to the corresponding problem is computed using as initial approximation the solution to the previous problem.

A continuum version of the Leclerc’s MDL criterion is [89], assuming the code length to describe the parametric models is common to all regions:

$$\mathcal{E}(\Gamma, \{\theta_k\}) = \sum_k \left( \frac{a}{2} \int_{\partial R_k} ds - \log P(\{I(\mathbf{x}) : \mathbf{x} \in R_k\} | \theta_k) \right), \quad (3.63)$$

where  $\{R_k\}$  is a partition of the image domain  $\Omega$ ,  $\Gamma = \{\partial R_k\}$  its boundaries, and  $\{\theta_k\}$  the regions parameters. The code length to describe the parametric models was assumed common to all regions and has been dropped. A transposition to optical flow can be written:



$$\mathcal{E}(\Gamma, \{\theta_k\}) = \sum_k \left( \frac{a}{2} \int_{\partial R_k} ds - \log P(\{r(\mathbf{x})\} : \mathbf{x} \in R_k | \theta_k) \right), \quad (3.64)$$

where  $r(\mathbf{x}) = (I_x u + I_y v + I_t)(\mathbf{x})$ . If we assume independent identical probability models for  $r$  everywhere on  $\Omega$ , then Eq. (3.64) can be re-written:

$$\mathcal{E}(\Gamma, \{\theta_k\}) = \frac{a}{2} \sum_k \int_{\partial R_k} ds - \int_{\Omega} \log P(r(\mathbf{x})) d\mathbf{x}. \quad (3.65)$$

A discretization of the length term is:

$$\frac{a}{2} \sum_{i \in D} \sum_{j \in \mathcal{N}_i} (1 - \delta(u_i - u_j) \delta(v_i - v_j)), \quad (3.66)$$

where  $a \approx 2$ . For  $r$  normally distributed with variance  $\sigma^2$ , a discretization of the data term can be written:

$$c + b \sum_{i \in D} \frac{(I_{xi} u_i + I_{yi} v_i + I_{ti})^2}{\sigma^2}, \quad (3.67)$$

where  $b = \frac{1}{2 \log 2}$  and  $c$  is a constant to ignore [33]. The minimum description length estimate of optical flow is the motion field  $\tilde{W}$  over  $D$  which corresponds to a minimum of the total code length of description:

$$E(W) = b \sum_{i \in D} \frac{(I_{xi} u_i + I_{yi} v_i + I_{ti})^2}{\sigma^2} + \frac{a}{2} \sum_{i \in D} \sum_{j \in \mathcal{N}_i} (1 - \delta(u_i - u_j) \delta(v_i - v_j)). \quad (3.68)$$

### Numerical Implementation

The objective function Eq. (3.68) is not differentiable due to the presence of the delta function, as in the Leclerc objective function for intensity images. This suggests to embed the minimization of Eq. (3.68) in a family of minimizations indexed by the parameter of a differentiable approximation of the  $\delta$  function, and use *continuation* [33, 81] to carry out the estimation. Continuation can be based on the following substitution:

$$\delta(u_i - u_j) \delta(v_i - v_j) \leftarrow e_{ij}(W, s) = e^{-\frac{(u_i - u_j)^2 + (v_i - v_j)^2}{(s\sigma)^2}} \quad (3.69)$$

Using  $s\sigma$  in Eq. (3.69), rather than  $s$ , simplifies subsequent expressions without causing a loss of generality. The actual parameter of continuation remains  $s$ . With

substitution Eq. (3.69), the objective function to minimize is re-written:

$$E(W, s) = b \sum_{i \in D} \frac{(I_{xi}u_i + I_{yi}v_i + I_{ti})^2}{\sigma^2} + \frac{a}{2} \sum_{i \in D} \sum_{j \in \mathcal{N}_i} (1 - e_{ij}(W, s)) \quad (3.70)$$

Let  $s_1, s_2, \dots$  be a decreasing sequence of  $s$  values tending to zero. Continuation solves the following sequence of problems indexed by these values of  $s$ :

$$\text{Minimize } E(W, s_l) \quad (3.71)$$

For each value  $s_l$  of  $s$ , the necessary conditions for a minimum of  $E$  give two constraints at each  $i \in D$ :

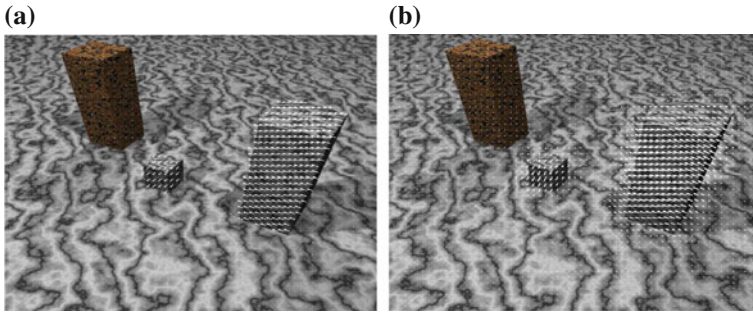
$$\begin{aligned} I_{xi}(I_{xi}u_i + I_{yi}v_i + I_{ti}) + a_l \sum_{j \in \mathcal{N}_i} (u_i - u_j) e_{ij}(W, s_l) &= 0 \\ I_{yi}(I_{xi}u_i + I_{yi}v_i + I_{ti}) + a_l \sum_{j \in \mathcal{N}_i} (v_i - v_j) e_{ij}(W, s_l) &= 0, \end{aligned} \quad (3.72)$$

where  $a_l = a \log 2/s_l^2$ . This yields a large scale sparse system of equations most of which are linear, and that can be solved using the following Jacobi-type iterative scheme where each iteration applies a Jacobi update to a linear system of equations obtained by evaluating the exponential term with the values of motion computed at the preceding iteration:

$$\begin{aligned} u_i^{k+1} &= \frac{-I_{xi}I_{ti} - I_{xi}I_{yi}v_i^k + a_l \sum_{j \in \mathcal{N}_i} e_{ij}^k(W, s_l)u_j^k}{I_{xi}^2 + a_l \sum_{j \in \mathcal{N}_i} e_{ij}^k(W, s_l)} \\ v_i^{k+1} &= \frac{-I_{yi}I_{ti} - I_{xi}I_{yi}u_i^k + a_l \sum_{j \in \mathcal{N}_i} e_{ij}^k(W, s_l)v_j^k}{I_{yi}^2 + a_l \sum_{j \in \mathcal{N}_i} e_{ij}^k(W, s_l)} \end{aligned} \quad (3.73)$$

The solution of each problem in Eq. (3.71) serves as the initial solution for the next problem. As  $s$  approaches zero, the problem approaches the original Eq. (3.68) because  $e_{ij}(s)$  tends to  $\delta$ . When  $s$  tends to  $\infty$ , the second term in Eq. (3.70) approaches 0. This suggest that the first problem be stated with  $s$  large, using, for instance, the normal component vector of optical flow as initial approximation. The iterations are continued up to a small  $s_l$ . As a rule of thumb, about 100 iterations of continuation and 5 of Eq. (3.73) sufficed in experiments.

**Example:** The MDL estimation scheme is illustrated using the Marbled blocks synthetic test sequence (*Marmor-2* sequence from the KOGS/ IAKS laboratory database, University of Karlsruhe, Germany). The rightmost block moves away to the left and the small center block forward to the left. The camera and the leftmost block are static. The images have been noised. The texture variation is weak at the top edges of the blocks. Depth varies sharply at the blocks boundaries not in contact with the floor. The blocks cast some shadows. The scene and the actual motion field are shown in Fig. 3.6a, and the MDL motion estimate in Fig. 3.6b. In spite of its embedded motion



**Fig. 3.6** **a** Ground truth image motion superimposed on the first of the two Marbled blocks images used in the experiment and, **b** the MDL motion estimate. In spite of its embedded motion boundary preservation, the scheme has let some smoothing, although mild, across the blocks occluding contours where there are motion discontinuities

boundary preservation, the scheme has let some smoothing, although mild, across the blocks occluding contours where depth, and motion thereof, vary sharply and significantly. The average error on the motion magnitude, over the whole image, is 0.13 pixel and the average direction error on the two moving blocks is  $4.7^\circ$ . The standard deviations are 0.2 for the magnitude, 5.8 for the direction for the small block and 3.5 for the large block. These statistics are comparable to those of the Horn and Shunck and the Deriche, Aubert, and Kornprobst schemes.

### 3.8 Parametric Estimation

Parametric motion estimation in a support region  $R \subset \Omega$  consists of representing the motion field in  $R$  by a parametric model and using the spatiotemporal data to determine the parameters which provide the best fit. One of the main motivations for parametric motion estimation is economy of description because motion in the support region  $R$  can be compactly described by the set of model parameters. Parametric estimation also forgoes the need for regularization in  $R$  because it implies smoothness of motion. We will focus on linear parametric models. They are analytically convenient to use and, when chosen properly, can be powerful so as to represent fine details of motion.

Parametric estimation of optical flow over a support region  $R$  can be set up as follows [60]. Let  $\theta_j : (x, y) \in \Omega \rightarrow \theta_j(x, y) \in \mathbb{R}$ ,  $j = 1, \dots, M$  be basis functions and  $L$  their span:  $L = \text{span}\{\theta_1, \dots, \theta_M\}$ . Each of the coordinate functions  $u, v$  of optical flow  $W$  is considered an element of  $L$ :

$$W = \alpha^T \theta \tag{3.74}$$

where  $\theta$  is the vector of basis functions:

$$\theta = (\theta_1 \theta_2 \cdots \theta_M)^T \quad (3.75)$$

and  $\alpha$  is the matrix of *parameters*, i.e., of the coefficients of the expansion of motion in the basis of  $L$ :

$$\alpha = \begin{pmatrix} \alpha_{11} & \alpha_{21} & \cdots & \alpha_{M1} \\ \alpha_{12} & \alpha_{22} & \cdots & \alpha_{M2} \end{pmatrix}^T \quad (3.76)$$

The first row of  $\alpha^T$  has the parameters of  $u$  and the second row those of  $v$ . The parameters in  $R$  are computed by minimizing the following functional which uses the optical flow constraint in which Eq.(3.74) is substituted:

$$\mathcal{E}(\alpha) = \int_R (\nabla I \cdot \alpha^T \theta + I_t)^2 dx dy. \quad (3.77)$$

The corresponding least squares equations to determine the parameters are:

$$\mathbf{B}\beta + \mathbf{d} = \mathbf{0} \quad (3.78)$$

where:

- Vector  $\beta$  is the  $2M \times 1$  vector constructed by vertical concatenation of the parameters  $\alpha_1$  and  $\alpha_2$  corresponding to optical flow components  $u$  and  $v$ :

$$\begin{aligned} \beta[m] &= \alpha_{m1} \\ \beta[M + m] &= \alpha_{m2}, \end{aligned} \quad (3.79)$$

for  $m = 1, \dots, M$ .

- Matrix  $\mathbf{B}$  is the following  $2M \times 2M$  matrix formed by the vertical and horizontal concatenation of  $4 M \times M$  sub-matrices  $\mathbf{B}_{rc}$ :

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}, \quad (3.80)$$

where the elements of the sub-matrices are defined by:

$$B_{rc}[m, n] = \int_R I_r I_c \theta_m \theta_n dx dy, \quad (3.81)$$

for  $m = 1, \dots, M, n = 1, \dots, M$  and  $I_l$  being the spatial derivative of  $I$  in the horizontal ( $l = 1$ ) and vertical ( $l = 2$ ) directions.

- Vector  $\mathbf{d}$  is the  $2M \times 1$  vector with the following elements, for  $m = 1, \dots, M$ :

$$\begin{aligned} d[m] &= \int_R I_r I_1 \theta_m dx dy \\ d[M + m] &= \int_R I_r I_2 \theta_m dx dy \end{aligned} \quad (3.82)$$

Region  $R$  is the support for the formulas above and the question arises as to which region to use to compute the motion field in  $\Omega$ . Several possibilities can be envisaged. One can use  $R = \Omega$ . In this case, the problem would be to choose the basis functions and their number. Large images in which several complex motions occur, independent human motions for instance, are likely to require a large number of parameters, which might invalidate the argument of representation economy and also reduce the effectiveness of the scheme. Another possibility is to formulate the problem as joint parametric motion estimation and segmentation. Segmentation would be a partition  $\mathcal{R} = \{R_i\}_1^N$  of  $\Omega$  into  $N$  regions differing by their motion as described by a parametric model, i.e., regions each with its own set of parameters. This problem will be investigated in (Sect. 3.11).

Another way to do parametric motion estimation, which does not resort to least squares fit over  $\Omega$  or use joint estimation and segmentation, has been investigated in [44]. The scheme, called over-parametrization, uses a set of parameters at each point  $(x, y) \in \Omega$ , i.e.,  $\alpha = \alpha(x, y)$  and, showing the dependence of the parameters on position:

$$W(x, y) = \alpha(x, y)^T \theta(x, y) \quad (3.83)$$

A linearized optical flow constraint version of the data term in [44] can be written:

$$\int_{\Omega} g \left( \nabla I(x, y) \cdot \alpha^T(x, y) \theta(x, y) + I_t(x, y) \right) dx dy, \quad (3.84)$$

where  $g(z) = \sqrt{z^2 + \varepsilon^2}$ , for some small  $\varepsilon$ , which induces an approximate  $L^1$  metric. The idea of over-parametrization was also used in image segmentation by Leclerc's MDL scheme [33] which looked at an image as a position-dependent parametric function of position. The constant and polynomial models were explicitly treated. Leclerc used the length of the motion boundary set to regularize the parameter field. This set is evaluated in the MDL cost by explicitly defining an edge to be a point between two regions of differing parametric motion descriptions. In [44], the regularization acts directly on the parameters and has the form:

$$\int_{\Omega} g \left( \sum_{i=1}^2 \sum_{j=1}^M \|\alpha_{ij}\|^2 \right). \quad (3.85)$$

Alternatively, it may be appropriate to use a boundary-preserving function of the type we discussed earlier. As with the boundary length term of Leclerc MDL formulation, this regularization implies that regions formed by functional minimization are characterized by one set of motion parameters and regions differ from each other by this set.

Let  $\delta$  be the optical flow parametric representation data term:

$$\delta = \nabla I \cdot \alpha^T \theta + I_t. \quad (3.86)$$

The Euler-Lagrange equations corresponding to the minimization of the over-parametrization functional:

$$\int_{\Omega} g(\delta^2) dx dy + \lambda \int_{\Omega} g\left(\sum_{i=1}^2 \sum_{j=1}^M \|\alpha_{ij}\|^2\right) dx dy \quad (3.87)$$

are given by, for  $j = 1, \dots, M$ :

$$\begin{aligned} g'(\delta^2) \delta I_x \alpha_{1j} + \lambda \operatorname{div} \left( g' \left( \sum_{i=1}^2 \sum_{j=1}^M \|\alpha_{ij}\|^2 \right) \nabla \alpha_{1j} \right) &= 0 \\ g'(\delta^2) \delta I_y \alpha_{2j} + \lambda \operatorname{div} \left( g' \left( \sum_{i=1}^2 \sum_{j=1}^M \|\alpha_{ij}\|^2 \right) \nabla \alpha_{2j} \right) &= 0 \end{aligned} \quad (3.88)$$

The formulation can be generalized to use the displaced frame difference in the data term rather than its Horn and Schunck linearized form [44, 90]. The equations are nonlinear. An efficient numerical implementation, within multiresolution processing (Sect. 3.10), is described in [44], with a validation experiment using the Yosemite test image sequences.

### 3.9 Variations on the Data and Smoothness Terms

To preserve motion boundaries some studies have used the  $L^1$ -norm for the optical flow smoothness term of the objective functional [91, 44, 92], in lieu of the quadratic regularization term of Horn and Schunck [10]. However, there has been no analysis or experimentation to support a comparison of the  $L^1$  norm and discontinuity preserving functions of the type in Table 3.1, the Aubert function for instance. The  $L^1$  norm has also been considered for the data term, to evaluate the displaced frame difference, or its continuous total temporal derivative expression. However, there is some evidence from an investigation of temporal noise in image sequences [93] that the  $L^2$  norm may be more appropriate.

Data functions other than the displaced frame difference, or its total temporal derivative continuous expression, have been suggested and some have been investigated experimentally [42], for instance those which arise from the invariance along motion trajectories of the image gradient or of its norm, the norm of the Laplacian, and the norm or trace of the Hessian. Some of these variations have exhibited very accurate results on the Yosemite test sequences.

### 3.10 Multiresolution and Multigrid Processing

Multiresolution and multigrid processing are “multilevel” computations which solve a system of equations on a given discretization grid by solving similar systems on grids at coarser discretizations. Although conceptually similar from this general point of view, multiresolution and multigrid processing differ in the order they visit the coarser grids and in the type of variables they compute at each of these grids.

### 3.10.1 Multiresolution Processing

The optical flow constraint, which enters the formulations we have discussed, uses the image temporal derivative, i.e., the image rate of change along the temporal axis. In practice, of course, we have to estimate the motion field from a discrete-time image sequence and if velocities are large, typically to cause displacements of over a pixel between consecutive views, the image temporal derivative may not be approximated sufficiently accurately to bear on velocity estimation, even when the image spatial resolution is high. In such a case, motion estimation can benefit from *multiresolution* processing [51, 22, 94, 28, 95]. In this coarse-to-fine strategy, estimation is served by a pyramidal image representation [96] in which an image is processed by filtering-and-subsampling into a pyramid of images of successively lower resolution. The original image is at the base of the pyramid. As motion extent decreases with resolution, the goal is to start processing at a pyramid level where this extent is within range of estimation. The estimate at this level is then projected on the level immediately below to warp the image at this level. The warped image is processed for an increment of motion, also assumed within range of estimation, and the scheme is repeated down to the original image at the base of the pyramid. Several variants of this basic coarse-to-fine scheme have been investigated but these have the same driving concepts, as just described, and differ mainly in the way the various steps are accomplished. Black's thesis [22] contains an introductory review of the subject. An actual use of multiresolution processing within a thorough motion estimation framework is given in [51, 28, 95].

The following algorithmic steps show the basic concepts involved in multiresolution estimation of optical flow. First, a pyramid of images is constructed from each of the two original images  $I_1$  and  $I_2$  used in the estimation, by repeated low-pass filtering, with a Gaussian, for instance, and subsampling at a rate of 2:

$$I_j^{l-1} \left( \frac{\mathbf{x}}{2} \right) = h * I_j^l(\mathbf{x}) \quad j = 1, 2, \quad (3.89)$$

where  $\mathbf{x} = (x, y)$ ,  $l$  designates the resolution level, corresponding to image size  $2^l \times 2^l$  (we assume that the length and width of the original image are powers of 2), the coarsest level being  $l = 0$ ,  $h$  is the filter and  $*$  designates convolution. Subsampling brings down the optical flow magnitude by a factor of two, the purpose being to have a valid discrete representation of the optical velocity components  $u$  and  $v$ . The intended purpose of low pass filtering is to bring the wavelength of the image spatial frequency components below the motion magnitude so as to have a valid discrete evaluation of the image temporal derivative. Both operations, filtering and subsampling, concur to make continuous formulations applicable at a pyramid level high enough, i.e., at an image resolution low enough. Optical flow is estimated at this coarsest level and estimation is continued down successive pyramid levels, i.e., up successively higher image resolution, using three basic operations at each level: (1) *prolongation* of the optical flow from the immediately preceding (higher, coarser) level, (2) transformation, at this level, of the first of the two images by this projected

flow, called image *warping*, and (3) estimation, at this level, of an *incremental flow* using the warped image and the second image:

At coarsest level  $l = 0$  initialize flow field  $W^0$

From  $l = 1$  to  $L$

1. Prolong the flow field:  $W^l = p(W^{l-1})$
2. Displace (warp) the first image by the flow field:  $I_1^l(\mathbf{x}) \leftarrow I_1^l(\mathbf{x} + W)$
3. Estimate the flow increment  $\delta W^l$  from  $I_1^l$  and  $I_2^l$
4. Update the flow field:  $W^l \leftarrow W^l + \delta W^l$

Prolongation is a coarse-to-fine interpolation operator which assigns to each fine-level point a value interpolated from the values at neighboring coarse-level points, i.e., it fills in the empty grid positions at level  $l$  by interpolating neighboring flow values at level  $l - 1$  multiplied by 2. The prolongation is generally called *projection* in the optical flow literature although the appellation is discordant with the common mathematical usage of the term. As well, image displacement (warping) at any level is done by interpolating the non-grid (displaced) values of the image.

### 3.10.2 Multigrid Computation

*Multigrid* procedures have been used in optical flow estimation [52] generally to complement mutiresolution processing at each level of the image pyramid [97, 95, 67, 98]. Multigrid schemes have had a great impact in numerical analysis where they were developed, particularly to solve iteratively large scale linear systems of equations in boundary value problems for partial differential equations [99, 100]. The main reason for using the multigrid computation is to refine via coarser grids a fine grid approximate solution cheaper, faster, and more accurately than using only the fine grid.

The multigrid method is better explained with (large) systems of linear equations although it is also applicable to nonlinear equations. Let  $\mathbf{A}^h \mathbf{z}^h = \mathbf{b}^h$  be a fine-grid system of linear equations,  $h$  designating the domain grid spacing. Let  $\tilde{\mathbf{z}}^h$  be an approximate solution and  $\mathbf{r}^h = \mathbf{b}^h - \mathbf{A}^h \tilde{\mathbf{z}}^h$ , called the *residual*. The *error*  $\mathbf{e}^h = \mathbf{z}^h - \tilde{\mathbf{z}}^h$  then satisfies:

$$\mathbf{A}^h \mathbf{e}^h = \mathbf{r}^h \quad (3.90)$$

This equation can be transferred to a coarser grid with spacing  $H$ , double the spacing for instance,  $H = 2h$ , as:

$$\mathbf{A}^H \mathbf{e}^H = \mathbf{R}_h^H \mathbf{r}^h, \quad (3.91)$$

where  $\mathbf{A}^H$  is a coarse-grid approximation of the fine-grid  $\mathbf{A}^h$  and  $\mathbf{R}_h^H$  is a *restriction* operator from the fine to the coarse grid, which assigns to each point of the coarse grid some weighed average of its argument evaluated at the neighboring fine-grid



points. An approximate solution  $\tilde{\mathbf{e}}^H$  of Eq.(3.91) is then computed to *correct* the fine-grid approximation  $\tilde{\mathbf{z}}^h$ :

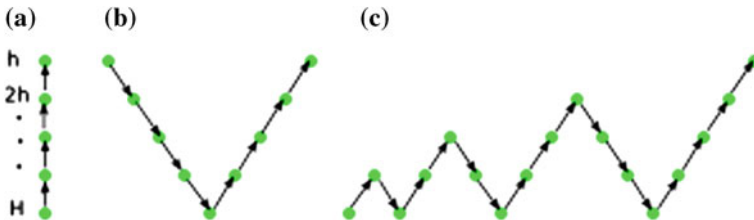
$$\tilde{\mathbf{z}}^h \leftarrow \tilde{\mathbf{z}}^h + \mathbf{P}_H^h \tilde{\mathbf{e}}^H, \tag{3.92}$$

where  $\mathbf{P}_H^h$  is a coarse-to-fine interpolation operator, also called prolongation, which assigns to each fine-grid point a value interpolated from the values at neighboring coarse-grid points. This basic two-level process is summarized by the following steps [100]:

1. Compute approximate solution  $\tilde{\mathbf{z}}^h$  by iterating a few times on  $\mathbf{A}^h \mathbf{z}^h = \mathbf{b}^h$
2. Compute fine grid residual  $\mathbf{r}^h = \mathbf{b}^h - \mathbf{A}^h \tilde{\mathbf{z}}^h$
3. Restrict  $\mathbf{r}^h$  to coarse grid residual  $\mathbf{r}^H$  by  $\mathbf{r}^H = \mathbf{R}_h^H \mathbf{r}^h$
4. Solve  $\mathbf{A}^H \mathbf{e}^H = \mathbf{r}^H$  for error  $\tilde{\mathbf{e}}^H$
5. Prolong  $\tilde{\mathbf{e}}^H$  to fine grid error  $\tilde{\mathbf{e}}^h$  by  $\tilde{\mathbf{e}}^h = \mathbf{P}_H^h \tilde{\mathbf{e}}^H$
6. Correct  $\tilde{\mathbf{z}}^h$  by  $\tilde{\mathbf{z}}^h \leftarrow \tilde{\mathbf{z}}^h + \tilde{\mathbf{e}}^h$
7. Iterate a few times on  $\mathbf{A}^h \mathbf{z}^h = \mathbf{b}^h$  from  $\tilde{\mathbf{z}}^h$

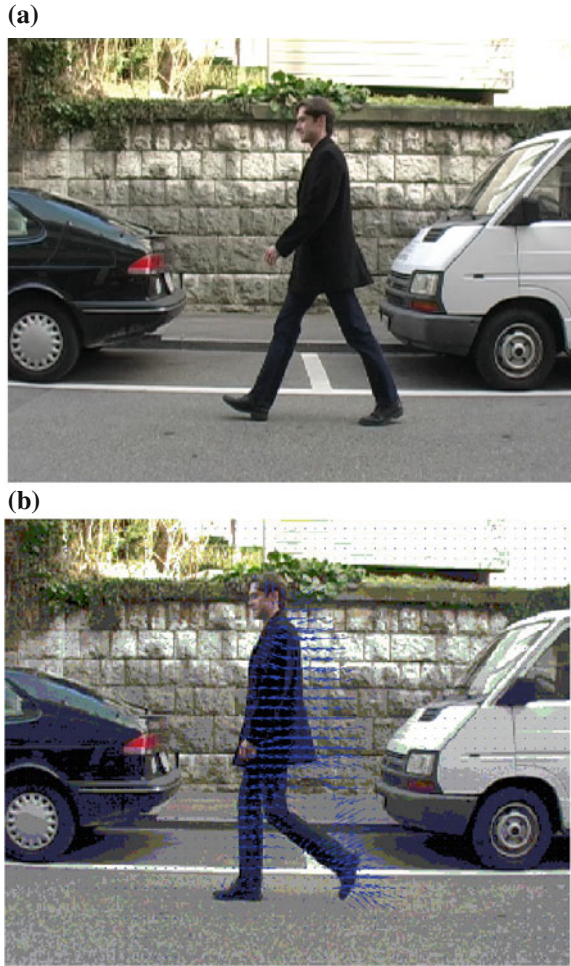
For common problems, there are standard restriction and prolongation operators, and the coarse-grid version  $\mathbf{A}^H$  can be computed from these as  $\mathbf{A}^H = \mathbf{R}_h^H \mathbf{A}^h \mathbf{P}_H^h$  [100]. The two-level algorithm above can be extended to a pyramid of more levels by using a hierarchy of coarse grids, for instance with spacings  $h, 2h, 4h, \dots, H$ , and calling the two-level algorithm *recursively* at each level except the coarsest, i.e., step 4 of the two-level algorithm is now a recursive call to it except at the coarsest grid where the error is computed to trigger an upward string of error prolongations and corresponding solution corrections. This “deep” V-path is illustrated in Fig. 3.7b.

The multigrid method is essentially different from the multiresolution in that it is an *error estimation* scheme which successively refines an initial approximate solution on the original high-resolution discretization grid using errors calculated on successively coarser grids. Multiresolution computations, instead, refine an initial approximation solved on the coarsest grid by working successively through higher resolutions up to the original fine grid. From this perspective, a multiresolution



**Fig. 3.7** Multiresolution and multigrid paths: **a** Multiresolution processing proceeds from low resolution to high; **b** V-cycle multigrid computations start at the original finest resolution grid to move through successively coarser grids to the coarsest and then up through successively finer grids until the finest; **c** Full multigrid cycle links several V-cycles of different size and the same depth starting at the coarsest grid

**Fig. 3.8** **a** The first of the two images used to compute optical flow; **b** the second image and the flow estimated by embedding in multiresolution processing. The flow occurs, predictably, in both the region uncovered by motion in the first image and the region covered in the second image. Multiresolution computations have been able to capture well the overall movement of the person



scheme adopts a coarse-to-fine strategy, i.e., after creating the image pyramid by low-pass filtering and sub-sampling, it works strictly down the image pyramid, i.e., from lowest resolution to highest (Fig. 3.7a), whereas multigrid processing moves both ways in this pyramid, first down to successively coarser resolutions and then back up to successively finer resolutions up to the original to apply corrections computed from an error solved at the coarsest resolution. Several of these V-shaped paths of different sizes but of the same depth can be linked into a string that starts at the coarsest grid to give the *full multigrid cycle*. This is illustrated in Fig. 3.7c. Nonlinear equations are generally resolved with such a cycle of computations.

**Example:** It is remarkable that multiresolution/multigrid processing works at all when displacements between views are significantly large as in the following exam-

ple. The displacements in the two images used here, of a person walking, are quite large. The first image is shown in Fig. 3.8a and the second in Fig. 3.8b which also displays the optical flow estimated by embedding in multiresolution processing. What should be pointed out in this example is that the flow seems to capture well the overall movement of the person in spite of the large displacements. Predictably, motion is found and estimated in both the region unveiled by motion in the first image and the region covered by motion in the second image. This kind of result can serve motion detection as we will see in Chap. 4.

### 3.11 Joint Estimation and Segmentation

Segmentation, or partitioning, of the flow field with concurrent estimation within each segmentation region with no particular concern about motion boundaries is an alternative to estimation with boundary-preserving regularization because segmentation will place boundaries between regions of significantly differing motion, therefore at significant flow edges. The usefulness of joint optical flow estimation and segmentation by variational methods was first investigated in [25, 101]. Active contours as motion boundary variables were used [54, 60, 102, 103], and embedding three-dimensional rigid body interpretation in estimation was considered in [58, 61]. When motion-based image partitioning is the main purpose of concurrent flow field estimation and segmentation, a coarse model of image motion such as piecewise constant or affine can be sufficient, particularly when this motion is due to viewing system movement and rigid environmental objects. However, given a flow-based segmentation obtained with a coarse motion model, one can apply a more accurate optical flow algorithm a posteriori in each segmentation region separately, the Horn and Schunck algorithm, for instance, or least squares in linear space parametrization [44, 60] or, yet, by over-parametrization for added accuracy and motion edge definition [44] (Sect. 3.8).

The following shows how active contours can be used to formulate joint motion estimation and segmentation. The formulation has been investigated in [54] for an arbitrarily fixed number of regions using the piecewise constant model of motion, i.e., optical flow in each segmentation region is considered constant. It has been extended to higher order linear models of motion, the affine for instance, and to the spatiotemporal domain in [103]. The expansion of motion in a general linear space of functions was studied in [60]. To bring out the main concepts involved in concurrent optical flow estimation and segmentation it is sufficient to use the piecewise constant model of motion and the case of a segmentation into two regions. We will use the method of Cremers [54] for this purpose.

#### Two-region partitioning.

Consider the case of segmenting the flow field into two regions and let  $\mathcal{R} = \{R_i\}_1^2$  be any two-region partition of the image sequence domain  $\Omega$ . Let  $\gamma$  be a regular closed plane curve parametrized by arc length,  $\gamma : [0, l] \leftarrow \mathbb{R}^2$ , where  $l$  is the curve length, such that  $\gamma$  and all its derivatives agree at the endpoints 0 and  $l$ . We will

further request that  $\gamma$  be a simple curve, i.e., that it has no other self-intersections but at  $0, l$ , i.e.,  $s_1, s_2 \in ]0, l[, s_1 \neq s_2 \implies \gamma(t_1) \neq \gamma(t_2)$ . Let  $R_\gamma$  be the interior of  $\gamma$ . The regions  $R_1$  and  $R_2$  of the two-region partition  $\mathcal{R}$  of  $\Omega$  will be represented by  $R_\gamma$  and  $R_\gamma^c$ , respectively. i.e.,  $R_1 = R_\gamma$  and  $R_2 = R_\gamma^c$ .

Under the piecewise constant model of optical flow, i.e., where the flow is assumed constant, equal to some velocity vector  $(a_i, b_i)$  in each region  $R_i$  of  $\mathcal{R}$ , the worth, or quality, of  $\mathcal{R}$  as an optical flow based segmentation of the image sequence  $I$  at some time of observation can be represented by the following functional:

$$\mathcal{E}(\mathcal{R}, \{a_i, b_i\}_1^2) = \mathcal{E}(\gamma, \{a_i, b_i\}_1^2) = \sum_{i=1}^2 \int_{R_i} e_i(x, y) dx dy + \lambda \int_\gamma ds, \quad (3.93)$$

where, for  $i = 1, 2$ ,  $e_i$  is a function which evaluates how well the piecewise constant representation of optical flow fits the observed data, namely the spatiotemporal variations of the image sequence within  $R_i$ . For instance, we can use the squared piecewise constant parametric expression of the lefthand side of the Horn and Schunck equation, a special case of the more general representation in Eqs. 3.74–3.77. An alternative is to use the squared cosine of the angle between the image spatiotemporal gradient and the spatiotemporal velocity vector  $(u, v, 1)$ , i.e., the square of the dot product of the unit image spatiotemporal gradient and unit spatiotemporal velocity vector, which is just what the lefthand side of the Horn and Schunck equation expresses would we normalize the two vectors by their respective length. Precisely, if the constant velocity vector of region  $R_i$  is  $\mathbf{w}_i = (a_i, b_i, 1)^T$ ,  $i = 1, 2$ , then:

$$e_i = \frac{(\mathbf{w}_i^T \nabla_3 I)^2}{\|\mathbf{w}_i\|^2 \|\nabla_3 I\|^2}, \quad (3.94)$$

where  $\nabla_3$  designates the spatiotemporal gradient,  $\nabla_3 I = (I_x, I_y, I_t)^T = (\nabla I, I_t)^T$ . Of course,  $\mathbf{w}_i \neq 0$  because the third component of the vectors is 1. To avoid zero denominators, a small quantity may be added to the image spatiotemporal gradient norm:  $\|\nabla_3 I\| + \varepsilon \leftarrow \|\nabla_3 I\|$ .

The main difference between the data function of Eq. 3.94 and the one used commonly in optical flow estimation, namely the squared lefthand side of the Horn and Schunck gradient equation:

$$e_i = (I_x u + I_y v + I_t)^2, \quad (3.95)$$

is the normalization of the spatiotemporal image gradient and motion vectors occurring in Eq. (3.94). The normalization to a unit vector of the image spatiotemporal gradient gives equal strength to the contribution in the objective functional from every point of  $\Omega$  where this vector is not zero. This is not the case with Eq. (3.95) which gives more weight, therefore more importance, to high contrast points. It is unclear whether high image contrast should or should not be given priority in determining optical flow. However, Eq. 3.94 has the merit, as we will see, of leading directly to

the expression of a small-matrix eigenvalue problem when minimizing the objective functional with respect to the optical flow model parameters  $a_i, b_i, i = 1, 2$ .

The integral in the second term of the objective functional Eq. (3.93) is the length of  $\gamma$  and has the effect of shortening it, therefore smoothing it. We know from Chap. 2 that this smoothing manifests as curvature in the Euler-Lagrange equation corresponding to this term in the minimization of Eq. (3.93) with respect to  $\gamma$ .

**Minimization with respect to the motion parameters.**

Let  $\mathbf{S}$  be the  $3 \times 3$  matrix defined by:

$$\mathbf{S} = \frac{\nabla_3 I (\nabla_3 I)^T}{\|\nabla_3 I\|^2}. \quad (3.96)$$

This matrix is, of course, a function of image position:  $\mathbf{S} = \mathbf{S}(x, y)$ . The data function for each region  $R_i, i = 1, 2$  can then be rewritten as:

$$e_i = \frac{\mathbf{w}_i^T \mathbf{S} \mathbf{w}_i}{\|\mathbf{w}_i\|^2}, \quad (3.97)$$

With this notation, differentiation with respect to  $\{a_i, b_i\}_1^2$  of Eq. (3.93) under the integral sign gives for each region  $R_i$  the solution  $\tilde{\mathbf{w}}_i$  defined by:

$$\tilde{\mathbf{w}}_i = \arg \min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}{\mathbf{w}^t \mathbf{w}}, \quad (3.98)$$

where  $\mathbf{M}_i$  is the  $3 \times 3$  data matrix given by:

$$\mathbf{M}_i = \int_{R_i} \mathbf{S}(x, y) dx dy, \quad (3.99)$$

obtained by integrating each element of  $\mathbf{S}$  over  $R_i$ . Because  $\mathbf{M}_i$  is a symmetric matrix, its smallest eigenvalue  $\mu_i$  is characterized by [104]:

$$\mu_i = \min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{M}_i \mathbf{w}}{\mathbf{w}^t \mathbf{w}}. \quad (3.100)$$

Therefore, the solution  $\tilde{\mathbf{w}}_i$  is the eigenvector corresponding to this smallest eigenvalue and which has the third component equal to 1.

**Minimization with respect to  $\gamma$ : curve evolution equation.**

With the motion parameters fixed, i.e., assuming they are independent of  $\gamma$  (or  $R_\gamma$ ), the functional derivative of the integral on  $R_\gamma$  of the objective functional data term is (see Chap. 2 for basic formulas):

$$\frac{\partial}{\partial \gamma} \int_{R_\gamma} e_1(x, y) dx dy = e_1 \mathbf{n}, \quad (3.101)$$

where  $\mathbf{n}$  is the outward unit normal function of  $\gamma$ . Similarly for the integral over  $R_\gamma^c$ :

$$\frac{\partial}{\partial \gamma} \int_{R_\gamma^c} e_2(x, y) dx dy = -e_2 \mathbf{n}. \quad (3.102)$$

The minus sign on the right-hand side of Eq.(3.102) is due to the fact that the boundary of  $R_\gamma^c$  is  $-\mathbf{n}$ . The functional derivative of the length integral of Eq.(3.93) is (see Chap. 2):

$$\frac{\partial}{\partial \gamma} \int_\gamma ds = \kappa \mathbf{n}, \quad (3.103)$$

where  $\kappa$  is the curvature function of  $\gamma$ . Accounting for all its terms, the functional derivative of the objective functional Eq.(3.93) is:

$$\frac{\partial \mathcal{E}}{\partial \gamma} = (e_1 - e_2 + \lambda \kappa) \mathbf{n}, \quad (3.104)$$

Let  $\gamma$  be embedded in a one-parameter family of curves  $\gamma(s, \tau)$  indexed by algorithmic time  $\tau$ . The evolution equation to minimize the objective functional with respect to  $\gamma$  is (see Chap 2):

$$\frac{\partial \gamma}{\partial \tau} = -\frac{\partial \mathcal{E}}{\partial \gamma} = -(e_1 - e_2 + \lambda \kappa) \mathbf{n}, \quad (3.105)$$

Recall that the evolving curve is called an *active curve*, or an active contour.

### Level set representation and evolution equation.

We recall from Chap. 2 some basic facts about level sets: an implementation of Eq.(3.105) which would explicitly discretize  $\gamma$  as a set of particles and move these, would be tantamount to numerical breakdown because changes in the curve topology would not be resolvable in general. Fans, where the particles separate widely to create large gaps, and shocks, where the particles come so close together as to collide or cross paths, would also be major hurdles. The level set method [105] avoids these serious problems by representing  $\gamma$  implicitly as a level set, the zero level set, for instance, of a function  $\phi$  defined on the plane. The level set function  $\phi$  is then evolved, rather than evolving  $\gamma$ , in a manner that is consistent with the evolution of  $\gamma$ , enabling the recovery of the curve at any time as its zero level set. With this representation,  $\gamma$  remains a well defined curve in the face of topology changes, fans, and shocks. Refer to Chap. 2 for a review.

Let the evolving curve  $\gamma(s, \tau)$  be represented at all times  $\tau$  by the zero level-set of function  $\phi : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ , taken by convention to be positive inside  $R_\gamma$  and negative outside. The evolution equation of  $\phi$  is given by:

$$\frac{\partial \phi}{\partial \tau} = (e_1 - e_2 + \lambda \kappa) \|\nabla \phi\| \quad (3.106)$$

In terms of the level set function, curvature  $\kappa$  of  $\gamma$  is expressed as:

$$\kappa = \operatorname{div} \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right) \quad (3.107)$$

when the normal unit vector  $\mathbf{n}$  is oriented outward:

$$\mathbf{n} = -\frac{\nabla \phi}{\|\nabla \phi\|} \quad (3.108)$$

### General linear models of motion.

The method [54] is easily extended to general linear models of optical flow by writing the spatiotemporal motion vector  $\mathbf{w} = (u, v, 1)^T$  in terms of the motion parameters via a transformation matrix  $\mathbf{T} = \mathbf{T}(x, y)$  independent of the image. The temporal dimension can also be included in the writing. For a model of  $n$  parameters  $a_1, \dots, a_n$ ,

$$\mathbf{w} = \mathbf{T}\alpha, \quad (3.109)$$

where  $\alpha$  is the vector of parameters augmented by a last element equal to 1,  $\alpha = (a_1, \dots, a_n, 1)^T$ . The first half of the parameters correspond to the component  $u$  of optical flow and the other half to the  $v$  component. Matrix  $\mathbf{T}$  is of size  $3 \times (n + 1)$ . For instance, for the affine model, we have:

$$\mathbf{T} = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.110)$$

With this model of motion, the data function of region  $R_i$  in the objective functional becomes:

$$e_i = \frac{(\alpha_i^T \mathbf{T} \nabla_3 I)^2}{\|\alpha_i\|^2 \|\mathbf{T} \nabla_3 I\|^2}, \quad (3.111)$$

From here on, the problem statement remains the same as with the piecewise constant model of motion.

A formulation using the standard data function Eq. (3.95), rather than Eq. (3.111), and an expansion of each component of optical flow in the span of a general basis of functions as in Eqs. 3.74–3.76 of Sect. 3.8, leads to similar computations, namely an algorithm which iterates two steps, least-squares estimation of the motion parameters, which can be done efficiently by the singular value decomposition (SVD) method, and active curve evolution with a velocity of the same expression as Eq. (3.105). More specifically, and using the notation and definitions in Eqs. 3.74–3.76 of Sect. 3.8, the formulation would seek to minimize:

$$\mathcal{E}(\gamma, \alpha_1, \alpha_2) = \sum_{i=1}^2 \int_{\mathbf{R}_i} \left( \nabla I \cdot \alpha_i^T \theta + I_t \right)^2 dx dy + \lambda \int_{\gamma} ds, \quad (3.112)$$

where  $\alpha_i$ ,  $i = 1, 2$  are the coefficient vectors for  $R_i$ ,  $i = 1, 2$ , with  $R_1 = R_\gamma$ ,  $R_2 = R_\gamma^c$ , and  $\theta$  is the vector of basis functions. The minimization is done by iterations of two steps, one to compute the parameters by least squares (Eq. 3.78), via SVD for instance, in each region separately, and the other to evolve the curve with:

$$\frac{d\gamma}{d\tau} = - \left( \left( \nabla I \cdot \alpha_1^T \theta + I_t \right)^2 - \left( \nabla I \cdot \alpha_2^T \theta + I_t \right)^2 + \lambda \kappa \right) \mathbf{n}. \quad (3.113)$$

The dependence of the parameters on the segmentation does not produce extra terms [106] in the evolution equation and the minimization corresponds to gradient descent rather than simply greedy descent.

An important question arises in parametric motion estimation as to which model complexity to use, i.e., how many basis functions to use in the representation of the components of optical flow. In what concerns estimation on a given support, the higher the model order the higher the accuracy. However, when the emphasis is on segmentation, then estimation accuracy is of secondary concern as long as it is sufficient to serve the segmentation, i.e., the model order to use is the least complex that permits a distinction between the regions of segmentation. For flow fields caused by moving rigid environmental objects, or by camera motion, a low-order model such as piecewise constant or affine will probably be satisfactory. However, other flow fields, such as those due to articulated objects or elastic environmental motion, may require higher order models. Ideally, one should use the smallest order that allows discriminating between the different relevant motions of the flow field because models of higher order might represent flow variations so fine as to produce a segmentation that is an artifact of the model rather than coherent motion. This is the problem of over-fitting mentioned in [103] which observed in practice cases of reduced curve evolution stability with increased model complexity. At any rate, accurate region-confined flow estimation can always follow a reasonably correct segmentation obtained with a lower order model.

### Multiregion segmentation.

A segmentation into more than two regions, called *multiregion segmentation*, or *multiphase segmentation*, uses two or more active curves. In essence, the objective functional data term for  $N$  regions  $\{R_i\}$  is:

$$\mathcal{D} = \sum_{i=1}^N \int_{R_i} e_i(x, y) dx dy \quad (3.114)$$

If one has several active curves and uses the interior of each to define a region, one must make sure that at algorithm completion the regions so defined form a partition, i.e., that they cover the image domain and do not intersect. Therefore, one cannot simply generalize a two-region algorithm by using more curves and assigning a region to the interior of each.

Multiregion segmentation has been addressed in several different ways. The methods have been described in detail in their original papers and have also been



reviewed in [31]. We will merely give of them here a brief account for a quick introduction to the literature on the subject. Matlab code of several algorithms is freely available on the web at [mathworks.de/matlabcentral/fileexchange/29447-multiphase-level-set-image-segmentation](http://mathworks.de/matlabcentral/fileexchange/29447-multiphase-level-set-image-segmentation) and elsewhere.

The earliest investigations of multiregion segmentation [89, 107] addressed the problem in two quite different ways. In a region competition framework [89], the curves  $\{\gamma_i\}$  were taken together as a set formed by their union, started as a partition, and moved as a set, i.e., the curve evolution equations resulting from region competition were applied to  $\Gamma = \cup \gamma_i$  considered a single curve. This representation does not extend to the level set method and, as a result,  $\Gamma$  is tracked by discretization particles, predisposing the scheme to numerical ills which do not occur with the level set method. Along a different vein in [107], several active curves mediate multiregion segmentation, each with its own speed function, but also with a contribution from a term in the objective functional dedicated to bias the segmentation toward a partition. However, a partition is not guaranteed at algorithm convergence because this term is weighed against the others in the functional and, therefore, the weight value conditions the outcome. The scheme also appears in the investigations of image segmentation of [108, 109].

Using several active contours and stating segmentation as spatially regularized clustering, the investigations in [60, 110, 111] were able to obtain coupled functionals, one for each curve. The resulting movement of the curves ends in a partition when the curves are started so as to define a partition [110]. However, the scheme can be quite slow because it sweeps through the image several times at each of many iterations and can sometimes produce artifacts such as elongated portions along region borders.

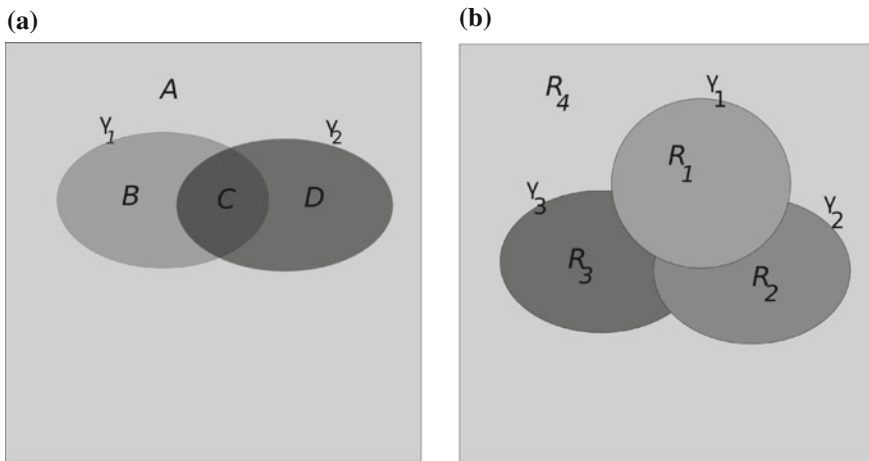
A definite means of ensuring a partition in multiregion segmentation is simply to define a general mapping between the regions  $\{R_i\}$  of the segmentation formulation and partition-defining regions drawn from the various sets which regions  $\{R_{\gamma_i}\}$  form when they intersect [112, 113]. Two such mappings are shown in Fig. 3.9. Both methods guarantee a partition by construction but the computational load can quickly become excessive when the number of regions increases.

A first order-order analysis of the region data functions in the two-region case brings out the interpretation of curve evolution as point membership operations. This directs to enforcing a simple partition constraint in the multiregion case directly in the functional minimization process and which states that a point relinquished by a region is claimed by another without transition through intermediate regions, thereby maintaining implicitly a partition of the image domain at all times when segmentation is started with a partition [114, 115]. This can lead in general to very efficient execution.

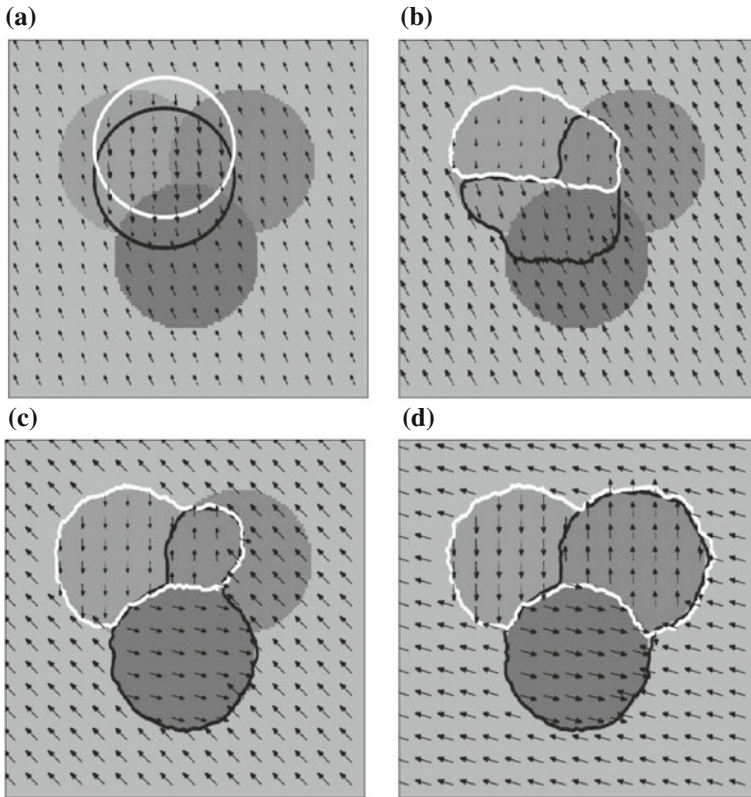
Multiregion segmentation raises the question as to what the number of regions is. In general, this is just fixed to equal the number one expects to occur, but there are many cases where this is not applicable. With curve evolution methods, there have been some efforts at determining the number of regions automatically, either as part of curve evolution optimization [116] or by an external process [89, 117, 111]. However,

experimentation regarding determining the number of regions automatically remains by and large insufficient, even though the question is quite important.

**Example:** This example (courtesy of D. Cremers) illustrates joint parametric estimation and segmentation of optical flow by the general active curve framework in [54] which we have just described. The scene of this sequence contains three circular objects moving against a mobile background. The purpose, therefore, is to segment the image into four regions on the basis of the direction of motion to be simultaneously estimated. The true image movements in the scene are: down (top left object), up (top right object), right (lower object), and left (background). The multiple region representation in terms of active contours is that of Chan and Vese [112]; therefore, two curves are needed (refer to Fig. 3.9a). The initial position of these curves is shown in Fig. 3.10a which also displays the evolving motion field. Intermediate motion fields and positions of the curves are shown in Figs. 3.10b and c. The curves and the motion field at convergence are in Fig. 3.10d. The curves define regions which correspond accurately to the objects and background, and the motion field fits the ground truth.



**Fig. 3.9** **a** Partition construction in [112]: A single curve defines two regions. Two intersecting simple closed curves give four disjoint subsets  $A$ ,  $B$ ,  $C$ ,  $D$ . These can be combined to have partitions of up to four regions. For four regions,  $R_1 = B = R_{\gamma_1} \cap R_{\gamma_2}^c$ ;  $R_2 = D = R_{\gamma_2} \cap R_{\gamma_1}^c$ ;  $R_3 = C = R_{\gamma_1} \cap R_{\gamma_2}$ ;  $R_4 = A = (R_1 \cup R_2)^c$ . In general,  $N$  regions necessitate  $\lceil \log N \rceil$  curves; **b** The mapping of [113]: three curves map to four partition regions:  $R_1 = R_{\gamma_1}$ ;  $R_2 = R_{\gamma_2} \cap R_{\gamma_1}^c$ ;  $R_3 = R_{\gamma_3} \cap R_{\gamma_2}^c \cap R_{\gamma_1}^c$ ;  $R_4 = (\cup_{i=1}^3 R_i)^c$ . In general, the mapping requires  $N-1$  curves for  $N$  regions



**Fig. 3.10** Joint segmentation and parametric estimation of optical flow by the Cremers method (Courtesy of Daniel Cremers): The true motions in the scene are: vertically down for the top left object, vertically up for the top right object, horizontally to the right for the lower object, and horizontally to the left for the background. Two curves are used to represent four regions according to the Chan and Vese mapping (Fig. 3.9a). The initial position of the curves is shown in **a**, intermediate positions and the evolving motion field are displayed in **b** and **c**. The final segmentation and motion field are shown in **d**. Both the segmentation and the motion field fit the ground truth

### 3.12 Joint Optical Flow and Disparity Estimation

In stereoscopy, the disparity field and optical flow are related by the *stereokinematic constraint* [62, 63]. Therefore, their joint estimation, via this constraint, can be advantageous [118–124]. Joint estimation involves computing two motion fields and two disparity fields using the stereokinematic constraint [62, 63] which relates three of these fields to the fourth.

Let the image sequence be a real positive function over a domain  $\Omega \times ]0, S[ \times ]0, T[$ , where  $]0, T[$  is an interval of time, and  $]0, S[$  an interval of  $\mathbb{R}$ :

$$I : \Omega \times ]0, S[ \times ]0, T[ \mapsto \mathbb{R}$$

$$\mathbf{x}, s, t \mapsto I(\mathbf{x}, s, t)$$

Variable  $s$  can be thought of as the parameter of the trajectory of a sequence of image planes in these planes coordinate domain. For a fixed value of  $s$  we have a temporal image sequence of images and for two distinct fixed values we obtain a stereoscopic image sequence. Therefore, this generalizes the definition of a stereoscopic image sequence. Let  $(\mathbf{x}, s, t) \in \Omega \times ]0, S[ \times ]0, T[$  and  $\mathbf{x} + \mathbf{d}(\mathbf{x}, s + ds, t + dt)$  its correspondent at  $(s + ds, t + dt)$ , where  $\mathbf{d}$  designates a displacement. The assumption that  $I$  does not change at corresponding points,

$$I(\mathbf{x} + \mathbf{d}(\mathbf{x}, s + ds, t + dt), s + ds, t + dt) = I(\mathbf{x}, s, t)$$

gives the following motion and disparity constraints:

$$\begin{aligned} \nabla I \cdot W + I_t &= 0 \\ \nabla I \cdot D + I_s &= 0, \end{aligned} \tag{3.115}$$

where  $W$  is the optical velocity vector,  $D$  the disparity vector,  $\nabla I$  the spatial gradient of  $I$ ,  $I_t$  and  $I_s$  the partial derivatives of  $I$  with respect to  $t$  and  $s$ . Because

$$W = \frac{\partial \mathbf{d}}{\partial t}, \quad D = \frac{\partial \mathbf{d}}{\partial s}, \tag{3.116}$$

we also have the integrability constraint:

$$\frac{\partial W}{\partial s} = \frac{\partial D}{\partial t} \tag{3.117}$$

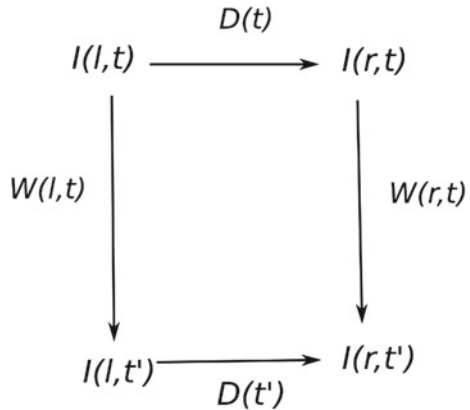
The integrability constraint is the continuous-disparity form of the stereokinematic constraint of [62] which was written for optical flow in discrete-disparity stereoscopic image sequences.

A fully discrete version of the integrability constraint can be written as follows. Let  $I^{l,t}, I^{r,t}$  be the left and right images at time  $t$  and  $I^{l,t'}, I^{r,t'}$  the left and right images at the next time  $t'$ . Let  $W^{l,t} = (u^{l,t}, v^{l,t})$  and  $W^{r,t} = (u^{r,t}, v^{r,t})$  designate left and right optical motion vectors at time  $t$ , and  $D^t = (\delta_1^t, \delta_2^t)$ ,  $D^{t'} = (\delta_1^{t'}, \delta_2^{t'})$  the disparity vectors at  $t$  and  $t'$ . A discrete representation of the integrability constraint can then be written:

$$W^{r,t} - W^{l,t} = D^{t'} - D^t \tag{3.118}$$

This is the *quadrilateral expression* of the stereokinematic constraint (Fig. 3.11). It is the expression generally used in practice [118–122].

**Fig. 3.11** Quadrilateral representing the stereokinematic constraint Eq. 3.118: Knowing three sides, we can deduce the fourth.



The two motion fields and the two disparity fields are related via the stereokinematic constraint. There is no other relation between any three of them but through the fourth. Therefore, joint estimation of the four fields which would treat the left and right data of stereoscopy even-handedly, can proceed along the following two veins:

(1) The four fields are estimated concurrently, for instance by minimizing an objective functional containing data and smoothness terms for each field, and a term to account for the stereokinematic constraint to bind the fields together. A slightly more efficient version of this, computationally, would estimate concurrently three of the fields bound by the stereokinematic constraint which uses the fourth field computed beforehand independently.

(2) Three of the fields are estimated independently, for instance by a variational formulation such as we have seen, and the fourth field is deduced directly using the stereokinematic constraint.

Estimation along the first vein entails solving a very large system of equations, nonlinear equations when using discontinuity preserving formulations. For instance, with a  $400 \times 400$  image, the number of scalar variables to determine in four fields is  $2^8 \times 10^4$  at each instant of observation. Along the second vein, one would solve independently three much smaller problems to estimate three fields, and follow with a direct application of the stereokinematic constraint to compute the fourth field. The process, therefore, is much more efficient along this vein. Also, and as we shall see, prolonging the estimation through time can be done at each instant of time by computing independently only the two flow fields, followed by an execution of the stereokinematic constraint using the previously computed disparity field.

According to the second paradigm, whereby three fields are computed independently and the fourth deduced, we can use constraints Eq. (3.115) to estimate separately the left and right motion fields and the disparity field at time  $t$  before computing the disparity field at time  $t'$  using the integrability/stereokinematic constraint Eq. (3.118). The left and right motion fields at  $t$  can be estimated for instance as in Section 3.5 by solving:

**Table 3.2** The two lines in the top box show the ground truth (constant) disparity for the background (B) and each of the two objects ( $O_1$  and  $O_2$ ) between the left and right images of the second stereoscopic image (Fig. 3.12a,b)

Actual	B	$O_1$	$O_2$
$x$	-1.0	1.0	1.0
$y$	0.0	1.0	1.0
<b>Joint estimation</b>	B	$O_1$	$O_2$
$x$	-0.8	0.96	0.92
$y$	0.02	0.81	0.87
<b>Deriche et al.</b>	B	$O_1$	$O_2$
$x$	-0.81	0.87	0.90
$y$	0.02	0.78	0.87

The middle box displays the average disparities, for the background and each object, computed by joint estimation optical flow and disparity as described in the text. The bottom box gives the average disparities computed by the Deriche-Aubert-Kornprobst method

$$W^{(l,r),t} = \arg \min_W \int_{\Omega} \left( (\nabla I^{(l,r),t} \cdot W + I_t^{(l,r),t})^2 + \lambda(g(\|\nabla u\|) + g(\|\nabla v\|)) \right) d\mathbf{x} \quad (3.119)$$

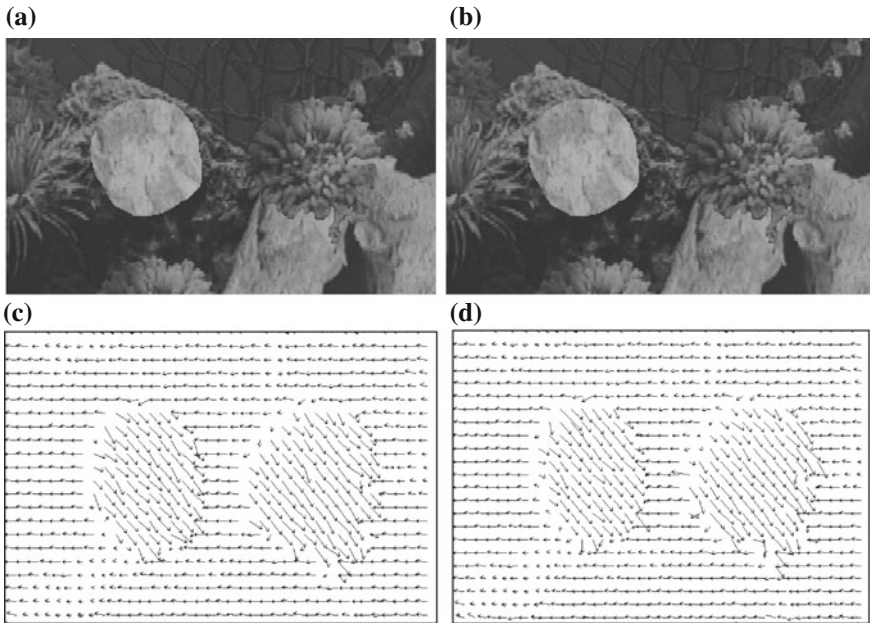
The disparity field can be computed by variational methods in a similar fashion, with or without the epipolar constraint [87, 125–128].

When the left and right motion fields and the disparity field are estimated at time  $t$ , the disparity field at time  $t'$  is deduced using the integrability/stereokinematic constraint, i.e.,

$$D^{t'} = W^{r,t} - W^{l,t} + D^t \quad (3.120)$$

We can make two observations: (1) Initially, the disparity field at time  $t$  is computed independently. Subsequently, the current disparity field is the disparity field computed at the previous instant, i.e., at each instant of time, except at the start, only the two motion fields are computed by Eq. (3.119), followed by an application of Eq. 3.120 and, (2) the formulation assumes that the disparity and motion are both of small extent. In the presence of either motion or disparity of large extent, estimation must resort to some form of multiresolution/multigrid computations (Sect. 3.10).

**Example:** The second of the two stereoscopic pairs of images used (constructed from the *Aqua* sequence) in this verification example (from [123]) is displayed in Fig. 3.12a, b. The scene consists of a circular object on the left (sea shell like) and a circular object on its right (sponge like), against a background (in an aquarium). Both objects are cutouts from real images. The background and the objects are given disparities in the first stereoscopic pair of images and are made to move such that disparities in the second stereoscopic pair are  $(-1, 0)$  for the background, and  $(1, 1)$  for the two objects (Table 3.2 upper box). The results are shown graphically in Fig. 3.12 for a qualitative appraisal, and quantitatively in Table 3.2 (lower two boxes).



**Fig. 3.12** Joint estimation of small extent optical flow and disparity: **a, b** the second of the two pairs of stereoscopic images used; **c** a graphical display of the disparities computed by joint estimation using the integrability/stereokinematic constraint; **d** A graphical display of the disparities computed with the Deriche-Aubert-Kornprobst method

### 3.13 State-of-the-Art

This chapter has presented the fundamental concepts underlying optical flow and its estimation, namely (i) the optical flow constraint which relates optical velocity to the image spatiotemporal gradient, (ii) the variational principle and the basic roles that conformity to data and regularization play in problem formulations, (iii) the necessity and mechanisms to preserve the sharpness of motion boundaries, (iv) multiresolution/multigrid processing to deal with long-range motion, (v) the combination of motion segmentation and motion estimation as joint processes, and (vi) the concurrent estimation of the optical flow and disparity fields. These concepts are self-contained and, as such, they were described separately to allow their full meaning to be exposed unconcealed by other considerations. Algorithms which account for each concept have been described, such as the Horn and Schunck method, the Deriche-Aubert-Kornprobst's and the Cremers'. The purpose of the presentation was to focus on explaining the idea underlying each abstraction and on means of effecting it, and no attempt was made to describe algorithms that would embody together several concepts. Such algorithms have been the concern of a number of studies investigating various mechanisms for accurate estimation. The domain is now mature enough to allow a thorough treatment of the problem leading to fast, effective, and accurate



algorithms with results that can be used for a variety of useful purposes, including motion detection and three-dimensional structure and motion recovery. This is the case, for instance, with the investigations in [12, 41, 42] which describe detailed optical flow computations that have produced impressive results. Faster computations using the conjugate gradient method to solve a large linear system of equations, rather than Gauss-Seidel or similar, have been implemented in Matlab/C++ and made available by [129] (<http://people.csail.mit.edu/celiu/OpticalFlow/>). The availability of good algorithms and implementations is complemented by useful collections of test image sequences and motion data, notably the Middlebury database (<http://vision.middlebury.edu/flow/>). Finally, the successful computational formulations and mechanisms used in optical flow estimation have found good use in joint disparity and optical flow estimation [124].

## References

1. J.J. Gibson, *The Perception of the Visual World* (Houghton Mifflin, Boston, 1950)
2. K. Nakayama, Biological image motion processing: a survey. *Vision. Res.* **25**, 625–660 (1985)
3. H.-H. Nagel, On the estimation of optical flow: relations between different approaches and some new results. *Artif. Intell.* **33**(3), 299–324 (1987)
4. H.-H. Nagel, Image sequence evaluation: 30 years and still going strong, in *International Conference on Pattern Recognition*, 2000, pp. 1149–1158
5. J. Barron, D. Fleet, S. Beauchemin, Performance of optical flow techniques. *Int. J. Comput. Vision* **12**(1), 43–77 (1994)
6. A. Mitiche, *Computational Analysis of Visual Motion* (Plenum Press, New York, 1994)
7. A. Mitiche, P. Bouthemy, Computation and analysis of image motion: A synopsis of current problems and methods. *Int. J. Comput. Vision* **19**(1), 29–55 (1996)
8. C. Stiller, J. Konrad, Estimating motion in image sequences: A tutorial on modeling and computation of 2D motion. *IEEE Signal Process. Mag.* **16**(4), 70–91 (1999)
9. G. Aubert, P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations* (Springer, New York, 2006)
10. B. Horn, B. Schunck, Determining optical flow. *Artif. Intell.* **17**, 185–203 (1981)
11. B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in *IJCAI*, 1981, pp. 674–679
12. A. Bruhn, J. Weickert, C. Schnörr, Lucas/kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vision* **61**(3), 211–231 (2005)
13. A. Mitiche, A. Mansouri, On convergence of the Horn and Schunck optical flow estimation method. *IEEE Trans. Image Process.* **13**(6), 848–852 (2004)
14. C. Koch, J. Luo, C. Mead, J. Hutchinson, Computing motion using resistive networks, in *NIPS*, 1987, pp. 422–431
15. J. Hutchinson, C. Koch, J. Luo, C. Mead, Computing motion using analog and binary resistive networks. *IEEE Comput.* **21**(3), 52–63 (1988)
16. H. Nagel, W. Enkelmann, An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(5), 565–593 (1986)
17. H.-H. Nagel, On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(1), 13–30 (1989)
18. H.-H. Nagel, Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow, in *European Conference on Computer Vision*, 1990, pp. 139–148



19. M. Snyder, On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(11), 1105–1114 (November 1991)
20. A. Mansouri, A. Mitiche, J. Konrad, Selective image diffusion: application to disparity estimation, in *International Conference on Image Processing*, 1998, pp. 284–288
21. F. Hampel, E. Ronchetti, P. Rousseeuw, W. Stahel, *Robust Statistics: The Approach Based on Influence Functions* (Wiley-Interscience, New York, 1986)
22. M. Black, “Robust incremental optical flow”, in Ph.D. Thesis, Yale University, Research Report YALEU-DCS-RR-923, 1992
23. M. J. Black, P. Anandan, A framework for the robust estimation of optical flow, in *International Conference on Computer Vision*, 1993, pp. 231–236
24. A. Blake, A. Zisserman, *Visual Reconstruction* (MIT Press, Cambridge, 1987)
25. E. Memin, P. Perez, Joint estimation-segmentation of optic flow, in *European Conference on Computer Vision*, 1998, vol. II, pp. 563–578
26. S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(6), 721–741 (1984)
27. J. Konrad, E. Dubois, Bayesian estimation of motion vector fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(9), 910–927 (1992)
28. F. Heitz, P. Boutheymy, Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(12), 1217–1232 (1993)
29. J. Zhang, G.G. Hanauer, The application of mean field theory to image motion estimation. *IEEE Trans. Image Process.* **4**(1), 19–33 (1995)
30. P. Nesi, Variational approach to optical flow estimation managing discontinuities. *Image Vision Comput.* **11**(7), 419–439 (1993)
31. A. Mitiche, I. Ben Ayed, *Variational and Level Set Methods in Image Segmentation* (Springer, New York, 2010)
32. D. Mumford, J. Shah, Boundary detection by using functionals. *Comput. Vis. Image Underst.* **90**, 19–43 (1989)
33. Y.G. Leclerc, Constructing simple stable descriptions for image partitioning. *Int. J. Comput. Vision* **3**(1), 73–102 (1989)
34. J. Weickert, A review of nonlinear diffusion filtering, in *Scale-Space*, 1997, pp. 3–28
35. L. Blanc-Feraud, M. Barlaud, T. Gaidon, Motion estimation involving discontinuities in a multiresolution scheme. *Opt. Eng.* **32**, 1475–1482 (1993)
36. M. Proesmans, L.J.V. Gool, E.J. Pauwels, A. Oosterlinck, Determination of optical flow and its discontinuities using non-linear diffusion, in *European Conference on Computer Vision*, 1994, pp. 295–304
37. R. Deriche, P. Kornprobst, G. Aubert, Optical-flow estimation while preserving its discontinuities: A variational approach, in *Asian Conference on Computer Vision*, 1995, pp. 71–80
38. G. Aubert, R. Deriche, P. Kornprobst, Computing optical flow via variational techniques. *SIAM J. Appl. Math.* **60**(1), 156–182 (1999)
39. A. Kumar, A. Tannenbaum, G.J. Balas, Optical flow: a curve evolution approach. *IEEE Trans. Image Process.* **5**(4), 598–610 (1996)
40. J. Weickert, C. Schnörr, Variational optic flow computation with a spatio-temporal smoothness constraint. *J. Math. Imaging Vision* **14**(3), 245–255 (2001)
41. T. Brox, A. Bruhn, N. Papenber, J. Weickert, High accuracy optical flow estimation based on a theory for warping, 2004. <http://citeseer.ist.psu.edu/brox04high.html>.
42. N. Papenber, A. Bruhn, T. Brox, S. Didas, J. Weickert, Highly accurate optic flow computation with theoretically justified warping. *Int. J. Comput. Vision* **67**(2), 141–158 (2006)
43. C. Zach, T. Pock, H. Bischof, A duality based approach for realtime tv-l1 optical flow, in *Annual Symposium of the German Association Pattern Recognition*, 2007, pp. 214–223
44. T. Nir, A.M. Bruckstein, R. Kimmel, Over-parameterized variational optical flow. *Int. J. Comput. Vision* **76**(2), 205–216 (2008)
45. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, H. Bischof, Anisotropic huber-l1 optical flow, in *BMVC*, 2009

46. C. Vogel, *Computational Methods for Inverse Problems* (SIAM, Philadelphia, 2002)
47. L. I. Rudin, S. Osher, Total variation based image restoration with free local constraints, in *ICIP*, vol. 1, 1994, pp. 31–35
48. I. Cohen, Nonlinear variational method for optical flow computation, in *SCIA93*, 1993, pp. 523–530
49. P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1981)
50. G. Bellettini, On the convergence of discrete schemes for the perona-malik equation. *Proc. Appl. Math. Mech.* **7**(1), 1023401–1023402 (2007)
51. W. Enkelmann, Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics, and Image Processing* **43**(2), 150–177 (August 1988)
52. D. Terzopoulos, Efficient multiresolution algorithms for computing lightness, shape-from-shading, and optical flow, in *AAAI conference*, 1984, pp. 314–317
53. D. Cremers, C. Schnorr, Motion competition: Variational integration of motion segmentation and shape regularization, in *DAGM Symposium on, Pattern Recognition*, 2002, pp. 472–480
54. D. Cremers, A multiphase level set framework for motion segmentation, in *Scale Space Theories in Computer Vision*, ed. by L. Griffin, M. Lillholm (Springer, Isle of Skye, 2003), pp. 599–614
55. A. Mansouri, J. Konrad, Multiple motion segmentation with level sets. *IEEE Trans. Image Process.* **12**(2), 201–220 (Feb. 2003)
56. D. Cremers, S. Soatto, Motion competition: A variational approach to piecewise parametric motion segmentation. *Int. J. Comput. Vision* **62**(3), 249–265 (2005)
57. T. Brox, A. Bruhn, J. Weickert, Variational motion segmentation with level sets, in *European Conference on Computer Vision*, vol. 1, 2006, pp. 471–483
58. H. Sekkati, A. Mitiche, Joint optical flow estimation, segmentation, and 3D interpretation with level sets. *Comput. Vis. Image Underst.* **103**(2), 89–100 (2006)
59. H. Sekkati, A. Mitiche, Concurrent 3D motion segmentation and 3D interpretation of temporal sequences of monocular images. *IEEE Trans. Image Process.* **15**(3), 641–653 (2006)
60. C. Vazquez, A. Mitiche, R. Laganiere, Joint segmentation and parametric estimation of image motion by curve evolution and level sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(5), 782–793 (2006)
61. A. Mitiche, H. Sekkati, Optical flow 3D segmentation and interpretation: A variational method with active curve evolution and level sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(11), 1818–1829 (Nov. 2006)
62. A. Mitiche, On combining stereopsis and kineopsis for space perception, in *IEEE Conference on Artificial Intelligence Applications*, 1984, pp. 156–160
63. A. Mitiche, A computational approach to the fusion of stereopsis and kineopsis, in *Motion Understanding: Robot and Human Vision*, ed. by W.N. Martin, J.K. Aggarwal (Kluwer Academic Publishers, Boston, 1988), pp. 81–99
64. S. Negahdaripour, C. Yu, A generalized brightness change model for computing optical flow, in *ICCV*, 1993, pp. 2–11
65. M. Mattavelli, A. Nicoulin, Motion estimation relaxing the constancy brightness constraint, in *ICIP*, vol. 2, 1994, pp. 770–774
66. R.P. Wildes, M.J. Amabile, A.-M. Lanzillotto, T.-S. Leu, Physically based fluid flow recovery from image sequences, in *CVPR*, 1997, pp. 969–975
67. T. Corpetti, É. Mémin, P. Pérez, Dense estimation of fluid flows. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(3), 365–380 (2002)
68. K. Nakayama, S. Shimojo, Intermediate and higher order aspects of motion processing, in *Neural Mechanisms of Visual Perception*, ed. by D.M.-K. Lam, C.D. Gilbert (Portfolio Publishing Company, The Woodlands, Texas, 1989), pp. 281–296
69. D. Todorovic, A gem from the past: Pleikart Stumpf's (1911) anticipation of the aperture problem, reichardt detectors, and perceived motion loss at equiluminance. *Perception* **25**(10), 1235–1242 (1996)

70. E. Hildreth, *The Measurement of Visual Motion* (MIT Press, Cambridge, 1983)
71. J. Kearney, W. Thompson, D. Boley, Optical flow estimation: an error analysis of gradient-based methods with local optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **9**(2), 229–244 (1987)
72. K. Wahn, L.S. Davis, P. Thrift, Motion estimation based on multiple local constraints and nonlinear smoothing. *Pattern Recogn.* **16**(6), 563–570 (1983)
73. V. Markandey, B. Flinchbaugh, Multispectral constraints for optical flow computation, in *International Conference on Computer Vision*, 1990, pp. 38–41
74. A. Mitiche, Y.F. Wang, J.K. Aggarwal, Experiments in computing optical flow with the gradient-based, multiconstraint method. *Pattern Recogn.* **20**(2), 173–179 (1987)
75. O. Treliak, L. Pastor, Velocity estimation from image sequences with second order differential operators, in *International Conference on Pattern Recognition and Image Processing*, 1984, pp. 16–19
76. A. Verri, F. Girosi, V. Torre, Differential techniques for optical flow. *J. Opt. Soc. Am. A* **7**, 912–922 (May 1990)
77. M. Campani, A. Verri, “Computing optical flow from an overconstrained system of linear algebraic equations, in *International Conference on Computer Vision*, 1990, pp. 22–26
78. M. Tistarelli, Computation of coherent optical flow by using multiple constraints, in *International Conference on Computer Vision*, 1995, pp. 263–268
79. R. Woodham, Multiple light source optical flow, in *International Conference on Computer Vision*, 1990, pp. 42–46
80. S. Baker, I. Matthews, Lucas-kanade 20 years on: a unifying framework. *Int. J. Comput. Vision* **56**(3), 221–255 (2004)
81. G. Dahlquist, A. Bjork, *Numerical Methods* (Prentice Hall, Englewood Cliffs, 1974)
82. P. Ciarlet, *Introduction à l’analyse numérique matricielle et à l’optimisation*, 5th edn. (Masson, Paris, 1994)
83. J. Stoer, P. Burlisch, *Introduction to Numerical Methods*, 2nd edn. (Springer, New York, 1993)
84. R. Feghali, A. Mitiche, Fast computation of a boundary preserving estimate of optical flow. *SME Vision Q.* **17**(3), 1–4 (2001)
85. L. Yuan, J. Li, B. Zhu, Y. Qian, A discontinuity-preserving optical flow algorithm, in *IEEE International Symposium on Systems and Control in Aerospace and Aeronautics*, 2006, pp. 450–455
86. W. Enkelmann, K. Kories, H.-H. Nagel, G. Zimmermann, An experimental investigation of estimation approaches for optical flow fields, in *Motion Understanding: Robot and Human Vision*, ed. by W.N. Martin, J.K. Aggarwal, (Chapter 6), (Kluwer Academic Publications, Boston, 1988), pp. 189–226
87. L. Álvarez, J. Weickert, J. Sánchez, Reliable estimation of dense optical flow fields with large displacements. *Int. J. Comput. Vision* **39**(1), 41–56 (2000)
88. S. Solimini, J.M. Morel, *Variational Methods in Image Segmentation* (Springer, New York, 2003)
89. S. Zhu, A. Yuille, Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **118**(9), 884–900 (1996)
90. T. Brox, B. Rosenhahn, D. Cremers, H.-P. Seidel, High accuracy optical flow serves 3- D pose tracking: Exploiting contour and flow based constraints, in *European Conference on Computer Vision*, 2006, pp. 98–111
91. C. Zach, T. Pock, H. Bischof, A duality based approach for realtime tv-l1 optical flow, in *DAGM*, 2007, pp. 214–223
92. A. Wedel, T. Pock, C. Zach, H. Bischof, D. Cremers, An improved algorithm for tv-l1 optical flow, in *Statistical and Geometrical Approaches to Visual Motion Analysis*, ed. by D. Cremers, B. Rosenhahn, A. Yuille, F. Schmidt. ser. Lecture Notes in Computer Science, (Springer, Heidelberg, 2009), pp. 23–45
93. A. Foi, M. Trimeche, V. Katkovnik, K. Egiazarian, Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. Image Process.* **17**(10), 1737–1754 (2008)

94. J. Bergen, P. Anandan, K. Hanna, R. Hingorani, Hierarchical model-based motion estimation, in *European Conference on Computer Vision*, 1992, pp. 237–252
95. É. Mémin, P. Pérez, Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. Image Process.* **7**(5), 703–719 (1998)
96. P. Burt, E. Adelson, The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* **31**(4), 532–540 (April 1983)
97. F. Heitz, P. Perez, P. Boutheymy, Multiscale minimization of global energy functions in some visual recovery problems. *CVGIP: Image Underst.* **59**(1), 125–134 (1994)
98. C. Cassisa, V. Prinnet, L. Shao, S. Simoens, C.-L. Liu, Optical flow robust estimation in a hybrid multi-resolution mrf framework, in *IEEE Acoustics, Speech, and, Signal Processing*, 2008, pp. 793–796
99. W. Hackbusch, U. Trottenberg (eds.), *Multigrid Methods. Lecture Notes in Mathematics*, vol. 960, (Springer, New York, 1982)
100. W.L. Briggs, *A Multigrid Tutorial* (SIAM, Philadelphia, 1987)
101. M. Chang, A. Tekalp, M. Sezan, Simultaneous motion estimation and segmentation. *IEEE Trans. Image Process.* **6**(9), 1326–1333 (1997)
102. D. Cremers, A. Yuille, A generative model based approach to motion segmentation, in *German Conference on Pattern Recognition (DAGM)*, (Magdeburg, Sept 2003), pp. 313–320
103. D. Cremers, S. Soatto, Variational space-time motion segmentation, in *International Conference on Computer Vision*, vol 2 (Nice, France, 2003), pp. 886–892
104. R.A. Horn, C.R. Johnson, *Matrix Analysis* (Cambridge University Press, Cambridge, 1985)
105. J.A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge University Press, Cambridge, 1999)
106. G. Aubert, M. Barlaud, O. Faugeras, S. Jehan-Besson, Image segmentation using active contours: calculus of variations or shape gradients? *SIAM J. Appl. Math.* **63**(6), 2128–2154 (2003)
107. H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion. *J. Comput. Phys.* **127**(1), 179–195 (1996)
108. N. Paragios, R. Deriche, Coupled geodesic active regions for image segmentation: A level set approach, in *European Conference on Computer vision*, (Dublin, Ireland, June 2000), pp. 224–240
109. C. Samson, L. Blanc-Feraud, G. Aubert, J. Zerubia, A level set model for image classification. *Int. J. Comput. Vision* **40**(3), 187–197 (2000)
110. C. Vazquez, A. Mitiche, I. Ben Ayed, Image segmentation as regularized clustering: A fully global curve evolution method, in *International Conference on Image Processing*, 2004, pp. 3467–3470
111. T. Brox, J. Weickert, Level set segmentation with multiple regions. *IEEE Trans. Image Process.* **15**(10), 3213–3218 (2006)
112. L. Vese, T. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vision* **50**(3), 271–293 (2002)
113. A. Mansouri, A. Mitiche, C. Vazquez, Multiregion competition: a level set extension of region competition to multiple region partitioning. *Comput. Vis. Image Underst.* **101**(3), 137–150 (2006)
114. I. Ben Ayed, A. Mitiche, Z. Belhadj, Polarimetric image segmentation via maximum likelihood approximation and efficient multiphase level sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(9), 1493–1500 (2006)
115. I. Ben Ayed, A. Mitiche, A partition constrained minimization scheme for efficient multiphase level set image segmentation, in *International Conference on Image Processing*, 2006, pp. 1641–1644
116. I. Ben Ayed, A. Mitiche, A region merging prior for variational level set image segmentation. *IEEE Trans. Image Process.* **17**(12), 2301–2313 (2008)
117. T. Kadir, M. Brady, Unsupervised non-parametric region segmentation using level sets, in *International Conference on Computer Vision*, 2003, pp. 1267–1274

118. A. Tamtaoui, C. Labit, Constrained disparity and motion estimators for 3DTV image sequence coding. *Signal Process.: Image Commun.* **4**(1), 45–54 (1991)
119. J. Liu, R. Skerjanc, Stereo and motion correspondence in a sequence of stereo images. *Signal Process.: Image Commun.* **5**(4), 305–318 (October 1993)
120. Y. Altunbasak, A. Tekalp, G. Bozdagi, Simultaneous motion-disparity estimation and segmentation from stereo, in *IEEE International Conference on Image Processing*, vol. III, 1994, pp. 73–77
121. R. Laganière, Analyse stéréocinétique d'une séquence d'images: Estimation des champs de mouvement et de disparité, Ph.D. dissertation, Institut national de la recherche scientifique, INRS-EMT, 1995
122. I. Patras, N. Alvertos, G. Tziritas, Joint disparity and motion field estimation in stereoscopic image sequences, in *IAPR International Conference on Pattern Recognition*, vol. I, 1996, pp. 359–363
123. H. Weiler, A. Mitiche, A. Mansouri, Boundary preserving joint estimation of optical flow and disparity in a sequence of stereoscopic images, in *International Conference on Visualization, Imaging, and Image Processing*, 2003, pp. 102–106
124. A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, D. Cremers, Stereoscopic scene flow computation for 3D motion understanding. *Int. J. Comput. Vision* **95**(1), 29–51 (2011)
125. L. Robert, R. Deriche, Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities, in *European Conference on Computer Vision*, 1996, pp. I:439–451
126. O. Faugeras, R. Keriven, Variational principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Trans. Image Process.* **7**(3), 336–344 (1998)
127. H. Zimmer, A. Bruhn, L. Valgaerts, M. Breuß, J. Weickert, B. Rosenhahn, H.-P. Seidel, PDE-based anisotropic disparity-driven stereo vision, in *Vision Modeling and Visualization*, 2008, pp. 263–272
128. C. Wohler, *3D Computer Vision: Efficient Methods and Applications* (Springer, Berlin, 2009)
129. C. Liu, Beyond pixels: Exploring new representations and applications for motion analysis, in *Ph.D. Thesis*, MIT, May 2009