International Journal of Computer Vision 70(3), 257–277, 2006

DOI: 10.1007/s11263-006-6616-7

A Multigrid Platform for Real-Time Motion Computation with Discontinuity-Preserving Variational Methods

ANDRÉS BRUHN AND JOACHIM WEICKERT
Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science,
Building E1 1, Saarland University, 66041 Saarbrücken, Germany
bruhn@mia.uni-saarland.de
weickert@mia.uni-saarland.de

TIMO KOHLBERGER AND CHRISTOPH SCHNÖRR
Computer Vision, Graphics, and Pattern Recognition Group, Department of Mathematics and
Computer Science, University of Mannheim, 68131 Mannheim, Germany
kohlberger@uni-mannheim.de
schnoerr@uni-mannheim.de

Received May 31, 2005; Revised December 8, 2005; Accepted December 9, 2005

Abstract. Variational methods are among the most accurate techniques for estimating the optic flow. They yield dense flow fields and can be designed such that they preserve discontinuities, estimate large displacements correctly and perform well under noise and varying illumination. However, such adaptations render the minimisation of the underlying energy functional very expensive in terms of computational costs: Typically one or more large linear or nonlinear equation systems have to be solved in order to obtain the desired solution. Consequently, variational methods are considered to be too slow for real-time performance. In our paper we address this problem in two ways: (i) We present a numerical framework based on bidirectional multigrid methods for accelerating a broad class of variational optic flow methods with different constancy and smoothness assumptions. Thereby, our work focuses particularly on regularisation strategies that preserve discontinuities. (ii) We show by the examples of five classical and two recent variational techniques that real-time performance is possible in all cases—even for very complex optic flow models that offer high accuracy. Experiments show that frame rates up to 63 dense flow fields per second for image sequences of size 160 × 120 can be achieved on a standard PC. Compared to classical iterative methods this constitutes a speedup of two to four orders of magnitude.

Keywords: partial differential equations, variational methods, optic flow, multigrid methods

1. Introduction

The estimation of motion information from image sequences is one of the key problems in computer vision. Typically one is thereby interested in finding the displacement field between two consecutive frames, the so-called *optic flow*. In this context, variational methods play a very important role, since they allow for both a precise and dense estimation of the results. Such techniques are based on the minimisation of a suitable energy functional that consists of two terms: a data term that imposes temporal constancy on certain image features, e.g. on the grey value of objects, and a smoothness term that regularises the often non-unique (local) solution of the data term by an additional smoothness assumption.

Although recent developments (Brox et al., 2004; Bruhn et al., 2005c; Mémin and Pérez, 1998) have shown that variational methods are among the best techniques for computing the optic flow in terms of error measures (Barron et al., 1994), they are often considered to be too slow for real-time applications. In particular, the computational costs for solving the resulting linear and nonlinear systems of equations are regarded as too high.

As a remedy for this problem but also in the context of nonconvex optimisation problems, multigrid methods have become a very popular tool during the last years. In particular so-called *unidirectional* multigrid schemes are frequently applied; see e.g. Anandan (1989); Luettgen et al. (1994). These coarse-to-fine strategies start with a coarse version of the original problem and refine it during the solution process. Thereby, coarse grid solutions serve as initialisations on finer grids. Unidirectional multigrid schemes offer two advantages: One one hand–in the case of *convex* energy functionals—they allow to speed up the computation significantly; see e.g. unidirectional schemes in (Bruhn et al., 2005c). This is not surprising since, coarse grid results are in general good approximations to fine grid solutions. On the other hand–in the case of *nonconvex* energy functionals–they allow to improve the quality of the results significantly (Alvarez et al., 2000; Black and Anandan, 1991; Bruhn et al., 2005c; Mémin and Pérez, 1998; Papenberg et al., 2005). This is a direct consequence of the fact that certain undesired local minima disappear at sufficiently coarse resolutions and thus can be avoided by a coarse-to-fine framework.

In the nonconvex case—in particular in the context of large displacements—unidirectional multigrid schemes are often combined with so-called *warping steps* (Bergen et al., 1992; Black and Anandan, 1991; Mémin and Pérez, 1998; Brox et al., 2004). Thereby the original problem is compensated by the already computed motion from all coarser levels before the resolution is refined. What remains to be solved at each resolution level is the motion increment for the difference problem (Alvarez et al., 2000; Black and Anandan, 1991; Bruhn et al., 2005c; Mémin and Pérez, 1998; Papenberg et al., 2005). Such an incremental computation offers one decisive advantage: It allows to approximate the nonlinear constancy assumptions in the data term by a series of linearised ones. As to be expected the obtained results for large displacements are much better than if the assumptions are only linearised once in the model—such as in Alvarez et al. (1999a), Horn and Schunck (1981), Nagel and Enkelmann (1986) and Weickert and Schnörr (2001a, b).

However, from a numerical viewpoint, these unidirectional multigrid schemes are not the end of the road. Very promising—in particular in the context of variational methods—are so-called *bidirectional* multigrid methods (Brandt, 1977; Briggs et al., 2000; Hackbusch, 1985; Trottenberg et al., 2001; Wesseling, 1992). These techniques that create a sophisticated hierarchy of equation systems with excellent error reduction properties belong to the fastest numerical schemes for solving linear or nonlinear systems of equations. In contrast to unidirectional schemes they Thus, they are able to overcome the typical limitation of basic iterative solvers that is also present in unidirectional multigrid schemes: *the weak attenuation of low error frequencies* (Brandt, 1977; Hackbusch, 1985). Moreover, they can benefit from all advantages of unidirectional multigrid schemes, since they may use the same coarse-to-fine initialisation strategy on top. Then, these bidirectional multigrid schemes are referred to as *full* multigrid methods (Briggs et al., 2000).

In Bruhn et al. (2005a) we have already demonstrated for variational methods with homogeneous regularisation that bidirectional multigrid schemes do allow for real-time performance. In this paper we show that by introducing a suitable notation it is possible to set up a much more general multigrid framework for real-time optic flow computation with variational methods. This allows us to develop such multigrid schemes also for discontinuity-preserving techniques with image- and flow-driven regularisation, both in their isotropic and anisotropic setting. Moreover, it is possible to extend our work to more advanced optic flow methods that are capable of a robust and accurate estimation of the wanted displacement fields. To the best of our knowledge our paper is the first one to report real-time performance for variational optic flow methods of such a quality on standard hardware.

Related Work. Due to the time-consuming adaptation process and the complexity of the resulting approaches bidirectional multigrid schemes are not very often used in the field of computer vision. However, there are a few works that deal with the development of such methods in the context of variational optic flow computation. Unfortunately, these approaches are generally restricted to linear techniques with homogeneous and image-driven regularisation (Glazer, 1984; Terzopoulos, 1986; Zini et al., 1997; El Kalmoun and Rüde, 2003; Enkelmann, 1987; Ghosal and Vaněk, 1996). Only the work of (Borzi et al., 2002) is known to the authors where a nonlinear optic flow problem with flow-driven regularisation was solved by means of a suitable bidirectional multigrid scheme (FAS). Also for other tasks in image processing and computer vision, bidirectional multigrid methods have been used successfully. In the context of photometric stereo and image binarisation (Kimmel and Yavneh, 2003) developed an algebraic multigrid method, while (Chan et al., 1997) researched geometric multigrid schemes for variational deconvolution with total variation (TV) regularisation. For TV denoising (Vogel, 1995) proposed the use of a linear multigrid method within a nonlinear fixed-point iteration, while, very recently, Frohn-Schnauf et al. (2004) investigated a nonlinear multigrid scheme (FAS) for the same task.

Paper Organisation. Our paper is organised as follows. In Section 2 we give a review on five different techniques that serve as prototypes for variational optic flow techniques with and without discontinuity-preserving regularisation. In this context, we also introduce the notation of motion and diffusion tensors that forms the basis of our general multigrid framework. In Section 3 we extend this framework to two more advanced optic flow techniques. Compared to the previously discussed prototypes these approaches offer an improved accuracy and an enhanced robustness. Section 4 is dedicated to discretisation aspects. It shows how to discretise the resulting Euler–Lagrange equations and which kind of linear or nonlinear systems of equations have to be solved. Efficient multigrid schemes for this purpose are developed in Section 5. To this end, different kind of multigrid strategies are discussed. In Section 6 we present an experimental evaluation that includes experiments with different real-world sequences, performance benchmarks for all prototypes and comparisons to results from the literature. Finally, a summary in Section 7 concludes this paper.

Our paper comprises and extends work previously published at two conferences (Bruhn and Weickert, 2005; Bruhn et al., 2005b). Substantial differences include, among other things, the considerations of all frequently used types of regularisation strategies (homogeneous, image- and flow-driven, isotropic and anisotropic), the extension to two more advanced variational optic flow techniques (Bruhn et al., 2005c; Papenberg et al., 2005)—the latter one replaces the method in (Bruhn et al., 2005b) and a much more extensive experimental evaluation.

2. Basic Variational Optic Flow Techniques

2.1. The Data Term

Let us consider some image sequence $f(x, y, t)$, where (x, y) denotes the location within a rectangular image domain Ω, and $t \in [0, T]$ denotes time. In order to retrieve corresponding objects in subsequent frames, one has to assume that certain image features do not to change over time. Such features may include the grey value, higher image derivatives such as the gradient or the Hessian or scalar–valued expression such as the norm of the gradient, the Laplacian or the determinant of the Hessian (Papenberg et al., 2005). Since we focus on basic optic flow techniques, we restrict ourselves at this point to the widely used grey value constancy assumption. It can be formulated as

$$f(x+u, y+v, t+1) - f(x, y, t) = 0, \qquad (1)$$

where t and $t+1$ are two consecutive frames. Performing a Taylor expansion and dropping all higher order terms one obtains its linearised form that is given by

$$f_x u + f_y v + f_t = 0. \qquad (2)$$

Here, the function $(u(x, y, t), v(x, y, t))^\top$ is the wanted displacement field, and subscripts denote partial derivatives.

2.1.1. The Motion Tensor Notation. In order to simplify the notation and to allow for a better understanding of the proposed discretisation coarse grid approximation approach (DCA) (Wesseling, 1992) in Section 5, let us introduce the concept of *motion tensors* (Farnebäck, 2001). To this end, we reformulate Eq. (2) as an inner product between the spatiotemporal flow vector $(u, v, 1)^\top$ and the spatiotemporal image gradient $\nabla_3 f := (f_x, f_y, f_z)^\top$. This allows to rewrite the standard data term that is based on a squared formulation of this equation as a quadratic form given by

$$\begin{aligned} E_D(u, v) &= (f_x u + f_y v + f_t)^2 \\ &= \left((u, v, 1)\ \nabla_3 f \nabla_3 f^\top\ (u, v, 1)^\top\right) \\ &= \left((u, v, 1)\ J\ (u, v, 1)^\top\right) \qquad (3) \end{aligned}$$

where the motion tensor $J := \nabla_3 f \nabla_3 f^\top$ is a 3×3 matrix which is positive semi-definite by construction.

One should note that such a reformulation by means of a quadratic form and a positive semi-definite motion tensor is possible for all constancy assumptions presented in (Papenberg et al., 2005). However, in the case of the grey value constancy assumption, the obtained quadratic form is very special: The associated motion tensor coincides exactly with the well-known structure tensor (Förstner and Gülch, 1987).

2.2. The Smoothness Term

Obviously, in case of a singular motion tensor, the solution of equation 3 is non-unique. Variational methods overcome this so–called aperture problem by additionally assuming (piecewise) smoothness of the result. As classified in (Weickert and Schnörr, 2001a), there are basically five different types of strategies to regularise this often non-unique solution of a data term: *homogeneous* regularisation that assumes overall smoothness and does not adapt to semantically important image or flow structures (Horn and Schunck, 1981) *image-driven* regularisation that assumes piecewise smoothness and respects discontinuities in the image (Alvarez et al., 1999a; Nagel and Enkelmann, 1986) *flow-driven* regularisation that assumes piecewise smoothness and respects discontinuities in the flow field; see e.g. (Cohen, 1993; Schnörr, 1994; Weickert and Schnörr, 2001a). Moreover, when considering image and flow-driven regularisation, one can distinguish between *isotropic* and *anisotropic* smoothness terms. While isotropic regularisers do not impose any smoothness at discontinuities, anisotropic ones permit smoothing along the discontinuity but not across it.

For each of the five strategies we have chosen one prototype based on the motion tensor formulation for the linearised grey value constancy assumption. In the following these approaches are discussed in detail.

(a) Homogeneous Regularisation

Prototype for the class of methods with *homogeneous* regularisation is the classical method of Horn and Schunck (1981). Their method assumes global smoothness by penalising deviations from smoothness in a quadratic way (Tikhonov and Arsenin, 1977). The corresponding energy functional reads

$$E_{\mathrm{HOM}}(u,v) = \int_\Omega \big((u,v,1)J(u,v,1)^\top + \alpha\,\big(\,|\nabla u|^2 + |\nabla v|^2\,\big)\big)\,dxdy, \tag{4}$$

where the regularisation parameter α is a positive number that steers the smoothness of the resulting flow field.

(b) Image-Driven Isotropic Regularisation

Instead of penalising deviations from smoothness in a quadratic way, one may think of downweighting the smoothness term at locations where the magnitude of the spatial image gradient is large (Alvarez et al., 1999a). This form of regularisation that respects discontinuities in the image data is called *image-driven isotropic* regularisation. The associated energy functional is given by

$$E_{\mathrm{II}}(u,v) = \int_\Omega \big((u,v,1)J(u,v,1)^\top + \alpha\ w(|\nabla f|^2)\,(|\nabla u|^2 + |\nabla v|^2\,)\big)\,dxdy, \tag{5}$$

where $w(s^2)$ is a positive decreasing function in $\mathbb{R}$. The method we have chosen to represent this class of regularisation is based on a function proposed by Charbonnier et al. (1994) which reads

$$w(s^2) = \frac{1}{\sqrt{1 + \frac{s^2}{\epsilon_S^2}}} \tag{6}$$

where ϵ_S is a parameter to steer the smoothness.

(c) Image-Driven Anisotropic Regularisation

As prototype for the class of optic flow methods with *image-driven anisotropic* regularisation we consider the technique of Nagel and Enkelmann (1986). Their method accounts for the problem of discontinuities by smoothing only along a projection of the flow gradient, namely its component orthogonal to the local image gradient. As a consequence, flow fields are obtained that avoid smoothing across discontinuities in the image data. The energy functional associated to this anisotropic form of regularisation is given by

$$\begin{aligned} E_{\mathrm{IA}}(u,v) = \int_\Omega \big((u,v,1)J(u,v,1)^\top &+ \alpha(\nabla u^\top D_{\mathrm{NE}}(\nabla f)\nabla u \\ &+ \nabla v^\top D_{\mathrm{NE}}(\nabla f)\nabla v)\big)\,dxdy, \end{aligned} \tag{7}$$

where $\nabla := (\partial_x, \partial_y)^\top$ denotes the spatial gradient and $D_{\mathrm{NE}}(\nabla f)$ is a projection matrix perpendicular to ∇f

that is defined as

$$D_{\mathrm{NE}}(\nabla f) = \frac{1}{|\nabla f|^2 + 2\epsilon_{\mathrm{S}}^2} \begin{pmatrix} f_y^2 + \epsilon_{\mathrm{S}}^2 & -f_x f_y \\ -f_x f_y & f_x^2 + \epsilon_{\mathrm{S}}^2 \end{pmatrix} =: \begin{pmatrix} a & b \\ b & c \end{pmatrix}. \qquad (8)$$

In this context ϵ_{S} serves as regularisation parameter that prevents the matrix $D_{\mathrm{NE}}(\nabla f)$ from getting singular.

(d) Flow-Driven Isotropic Regularisation

In contrast to image-driven regularisation methods, flow-driven techniques reduce smoothing at those locations where edges in the flow field occur during the computation. *Flow-driven isotropic* methods realise this by penalising deviations from smoothness less severely than in the quadratic setting (L_2 norm). As a consequence, large gradient features such as edges are better preserved. Such a form of penalisation can be related to statistically robust error norms (Huber, 1981). The corresponding energy functional reads

$$E_{\mathrm{FI}}(u, v) = \int_\Omega \Big((u, v, 1) J \; (u, v, 1)^\top + \alpha \, \Psi_{\mathrm{S}}\big(\, |\nabla u|^2 + |\nabla v|^2 \, \big) \Big) \, dx dy, \qquad (9)$$

where $\Psi_{\mathrm{s}}(s^2)$ is a positive increasing function in $\mathbb{R}$ with the property to increase less severely than a quadratic function. As prototype we have chosen a method that penalises deviations from the smoothness with the L_1 norm. This corresponds to total variation (TV) regularisation (Rudin et al., 1992) which we implemented by means of a regularised variant given by

$$\Psi_{\mathrm{S}}(s^2) = \sqrt{s^2 + \epsilon_{\mathrm{S}}^2}. \qquad (10)$$

Here, ϵ_{S} serves as small regularisation parameter. A similar functional that approximates TV regularisation is proposed in Weickert and Schnörr (2001b), while variational approaches for rotationally not invariant versions of TV regularisation have been investigated in Cohen (1993), Deriche et al. (1995) and Kumar et al. (1996).

(e) Flow-Driven Anisotropic Regularisation

The fifth and last regularisation strategy are *flow-driven anisotropic* smoothness terms (Weickert and Schnörr, 2001a). In contrast to the isotropic case where the non-quadratic function Ψ_{s} penalises the magnitude of the flow vector, it is now applied to the local flow tensor $\nabla u \nabla u^\top + \nabla v \nabla v^\top$ which additionally contains directional information. Thereby the application of the function to the tensor can be realised by means of an eigenvalue decomposition. This allows to access the eigenvalues directly and thus yields an penalisation that adapts to the local flow structure. The associated energy functional is given by

$$E_{\mathrm{FA}}(u, v) = \int_\Omega \Big((u, v, 1) J (u, v, 1)^\top + \alpha \; \mathrm{tr}\big(\Psi_{\mathrm{S}}(\, \nabla u \nabla u^\top + \nabla v \nabla v^\top) \big) \Big) \, dx dy, \qquad (11)$$

where tr is the trace of the local flow tensor. As for the isotropic case we have chosen a method as prototype that is based on a regularised variant of the total variation.

2.3. *The Euler–Lagrange Equations*

Following the calculus of variations (Elsgolc, 1961), the minimisation of the previously discussed energy functionals comes down to solving their Euler–Lagrange equations. As for the motion tensor in the data term, also a very compact and general formulation for the smoothness term is possible: the diffusion tensor notation (Weickert, 1998). Let us now explain this notation by the example of the Euler-Lagrange equations (a)–(e).

2.3.1. The Diffusion Tensor Notation

(a)–(c) The Linear Case

In the first three cases (a), (b) and (c) the Euler–Lagrange equations have the coupled form

$$0 = J_{11} \, u + J_{12} \, v + J_{13} \; - \; \alpha \, \mathcal{L}_{\mathrm{L}} u, \qquad (12)$$

$$0 = J_{12} \, u + J_{22} \, v + J_{23} \; - \; \alpha \, \mathcal{L}_{\mathrm{L}} v \qquad (13)$$

with the *linear* differential operator

$$\mathcal{L}_{\mathrm{L}} z(x, y) = \mathrm{div} \; (D(\nabla f) \; \nabla z(x, y)) \qquad (14)$$

and homogeneous Neumann boundary conditions. The 2 × 2 matrix D within the divergence expression is thereby called *diffusion tensor* and is given by

(a) Homogeneous Regularisation

$D(\nabla f) = I$

(b) Image-Driven Isotropic Regularisation

$D(\nabla f) = w(|\nabla f|^2)I$

(c) Image-Driven Anisotropic Regularisation

$D(\nabla f) = D_{\mathrm{NE}}(\nabla f)$

(d)–(e) The Nonlinear Case I

In the cases (d) and (e) the associated Euler-Lagrange equations have a structure that is very similar to the one for (a), (b) and (c). They are given by the coupled form

$$0 = J_{11}\, u + J_{12}\, v + J_{13} \,-\, \alpha\, \mathcal{L}_{\mathrm{NL}}(u, v), \qquad (15)$$

$$0 = J_{12}\, u + J_{22}\, v + J_{23} \,-\, \alpha\, \mathcal{L}_{\mathrm{NL}}(v, u) \qquad (16)$$

with the *nonlinear* differential operator

$$\mathcal{L}_{\mathrm{NL}}(z(x,y), \tilde{z}(x,y)) = \operatorname{div}\, (D(\nabla z(x,y), \nabla \tilde{z}(x,y))\, \nabla z(x,y)) \qquad (17)$$

and homogeneous Neumann boundary conditions. Here, $\mathcal{L}_{\mathrm{NL}}$ is a nonlinear differential operator, because it depends nonlinearly on its arguments z and $\tilde{z}$ (which are in fact u and v). This can be directly seen from the corresponding diffusion tensors that are given by

(d) Flow-Driven Isotropic Regularisation

$D(\nabla z, \nabla \tilde{z}) = \Psi'_{\mathrm{S}}(|\nabla z|^2 + |\nabla \tilde{z}|^2)\ I$

(e) Flow-Driven Anisotropic Regularisation

$D(\nabla z, \nabla \tilde{z}) = \Psi'_{\mathrm{S}}(\nabla z \nabla z^\top + \nabla \tilde{z} \nabla \tilde{z}^\top)$

where the derivative of the regularised total variation is obviously nonlinear since it reads

$$\Psi'_{\mathrm{S}}(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon_{\mathrm{S}}^2}}. \qquad (18)$$

As we will see later, this nonlinearity of the differential operator $\mathcal{L}_{\mathrm{NL}}$ has serious impact on the resulting discrete system of equations and on the derived multigrid strategy.

3. More Advanced Variational Optic Flow Techniques

After having introduced our prototypes for the five different types of regularisation strategies, let us now discuss two advanced prototypes for more advanced optic flow techniques: The noise robust combined-local-global (CLG) approach of (Bruhn et al., 2005c) and the highly accurate optic flow method of Papenberg et al. (2005). In the following both techniques are explained in detail.

(f) Noise Robustness—The Method of Bruhn et al.

In motion estimation the sensitivity of approaches with respect to noise is a very important aspect for the design of algorithms. In this context (Bruhn et al., 2005c) presented a variational optic flow approach that tackles this problem in two ways: (i) It combines the robustness of local methods with the full density of global approaches. This is achieved by embedding a local least square fit into the motion tensor formulation of the data term. As a result the original tensor J is integrated over a neighbourhood of fixed size, which is realised by a channelwise convolution of J with a Gaussian kernel K_ρ of standard deviation ρ. Thus, a modified motion tensor $J_\rho := K_\rho * J$ is obtained that renders the method more robust against noise. (ii) Apart from this substitution, a non-quadratic function Ψ_{D} is applied to the whole data term. As for the flow-driven isotropic regularisation, such a proceeding is related to statistically robust error norms (Huber, 1981) and increases the performance of the approach with respect to noise (Black et al., 1996). As prototype for this class of combined-local-global methods we have chosen a technique with regularised L_1 norm as non-quadratic penaliser in both the data and the smoothness term. The associated energy functional is given by

$$E_{\mathrm{CLG}}(u, v) = \int_\Omega \Big(\Psi_{\mathrm{D}}\left((u, v, 1) J_\rho\ (u, v, 1)^\top\right) + \alpha\ \Psi_{\mathrm{S}}\big(\ |\nabla u|^2 + |\nabla v|^2\ \big) \Big)\, dx dy, \qquad (19)$$

where ϵ_{D} and ϵ_{S} serve as small regularisation parameters for the L_1 norm in the data and in the smoothness term, respectively.

(g) Large Displacements, Varying Illumination—The Method of Papenberg et al.

Apart from noise there are two further problems that have to be addressed in the context of motion estimation: the estimation of large displacements and varying illumination. In (Brox et al., 2004; Papenberg et al., 2005) proposed a sophisticated variational approach that tackles both problems at the same time and allows for a very accurate estimation of the results. In their approach the standard grey value constancy assumption is supplemented by an additional term: The constancy of the spatial image gradient $\nabla f = (f_x, f_y)^\top$. This assumption allows to deal with global illumination changes of additive type. Moreover, in order to overcome the limitation of linearised constancy assumptions—they only hold for small displacements—their linearisation is postponed to the numerical scheme. The corresponding energy functional to this method reads

$$\begin{aligned} &E_{\text{PAP}}(u, v)\\ &\quad = \int_\Omega \big(\Psi_{\text{D}}(\big|f(x+u, y+v, t+1) - f(x, y, t)\big|^2\\ &\qquad + \gamma\, |\nabla f(x+u, y+v, t+1) - \nabla f(x, y, t)|^2)\\ &\qquad + \alpha\, \Psi_{\text{S}}(\, |\nabla u|^2 + |\nabla v|^2\,)\big)\, dx dy, \end{aligned} \tag{20}$$

where once again the regularised variant of the L_1 norm is applied to both the data and the smoothness term. Here, the scalar γ serves as weight between the gradient and the grey value constancy assumption.

3.1. The Euler–Lagrange Equations

Let us now derive the Euler–Lagrange equations for the prototypes of the more advanced optic flow methods.

(f) The Nonlinear Case II—The Method of Bruhn et al.

In the case of the CLG method the Euler–Lagrange equations are very similar to those of the flow-driven isotropic regularisation in (d). The have the same *nonlinear* differential operator (and diffusion tensor) and are given by the coupled form

$$\begin{aligned} 0 = {}& \Psi'_{\text{D}}\left((u, v, 1) J_\rho\, (u, v, 1)^\top\right)\\ & \times (J_{\rho 11}\, u + J_{\rho 12}\, v + J_{\rho 13}) - \alpha\, \mathcal{L}_{\text{NL}}(u, v), \end{aligned} \tag{21}$$

$$\begin{aligned} 0 = {}& \Psi'_{\text{D}}\left((u, v, 1) J_\rho (u, v, 1)^\top\right)\\ & \times \left(J_{\rho 12}\, u + J_{\rho 22}\, v + J_{\rho 23}\right) - \alpha\, \mathcal{L}_{\text{NL}}(v, u). \end{aligned} \tag{22}$$

Main differences to the flow-driven isotropic case in (d) are the modified motion tensor J_ρ and the additional factor $\Psi'_{\text{D}}\left((u, v, 1) J_\rho (u, v, 1)^\top\right)$ in front of the data term that results from its non-quadratic penalisation in the energy functional (via Ψ_{D}).

(g) The Nonlinear Case III (Warping)—The Method of Papenberg et al.

The Euler Lagrange equations for the method of Papenberg *et al.* are also based on flow-driven isotropic regularisation, so the *nonlinear* differential operator (and diffusion tensor) is once more the same than in the case (d). However, as one can see from the following equations

$$\begin{aligned} 0 = {}& \Psi'_{\text{D}}\big(|f(x+u, y+v, t+1) - f(x, y, t)|^2\\ & + \gamma\, |\nabla f(x+u, y+v, t+1) - \nabla f(x, y, t)|^2\big)\\ & \times \Big((f(x+u, y+v, t+1) - f(x, y, t))\\ & \times f_x(x+u, y+v, t+1)\\ & + \gamma\, (f_x(x+u, y+v, t+1) - f_x(x, y, t))\\ & \times f_{xx}(x+u, y+v, t+1)\\ & + \gamma\, \big(f_y(x+u, y+v, t+1) - f_y(x, y, t)\big)\\ & \times f_{yx}(x+u, y+v, t+1)\Big)\\ & - \alpha \mathcal{L}_{\text{NL}}(u, v), \end{aligned} \tag{23}$$

$$\begin{aligned} 0 = {}& \Psi'_{\text{D}}\big(|f(x+u, y+v, t+1) - f(x, y, t)|^2\\ & + \gamma\, |\nabla f(x+u, y+v, t+1) - \nabla f(x, y, t)|^2\big)\\ & \times \Big((f\ \ (x+u, y+v, t+1) - f\ (x, y, t))\\ & \quad \times f_x(x+u, y+v, t+1)\\ & + \gamma\, (f_x(x+u, y+v, t+1) - f_x(x, y, t))\\ & \quad \times f_{xy}(x+u, y+v, t+1)\\ & + \gamma\, \big(f_y(x+u, y+v, t+1) - f_y(x, y, t)\big)\\ & \quad \times f_{yy}(x+u, y+v, t+1)\Big)\\ & - \alpha\, \mathcal{L}_{\text{NL}}(v, u) \end{aligned} \tag{24}$$

the part for the data term is rather complex without linearisations. Moreover, in contrast to the previous cases (a)–(e) where globally convergent algorithms can be used to find the unique solution of the Euler-Lagrange equations, this time the solution process comes down to solving an equation system that is related to a *non-convex* optimisation problem. Therefore we follow the idea of Brox et al. (2004) and embed the solution of these equations in an incremental computation based on a coarse-to-fine fixed point iteration. This is done in three steps:

- Firstly, a fixed point iteration is applied to the Euler-Lagrange equations in (23)–(24). In order to allow for a faster convergence and a better stability than explicit schemes, we use an approach that is semi implicit in the data and fully implicit in the smoothness term. This yields an iteration instruction that requires to solve a coupled system of nonlinear equations in each iteration step. For the iteration step $k+1$ this system is given by

$$\begin{aligned}
0 = \Psi'_{\mathrm{D}} \Big(& \big|f(x+u^{k+1}, y+v^{k+1}, t+1) - f(x,y,t)\big|^2 \\
& + \gamma \, \big|\nabla f(x+u^{k+1}, y+v^{k+1}, t+1) - \nabla f(x,y,t)\big|^2 \Big) \\
& \times \Big((f(x+u^{k+1}, y+v^{k+1}, t+1) - f(x,y,t)\Big) \\
& \quad \times f_x(x+u^k, y+v^k, t+1) \\
& + \gamma \, \big(f_x(x+u^{k+1}, y+v^{k+1}, t+1) \; - f_x(x,y,t)\big) \\
& \quad \times f_{xx}(x+u^k, y+v^k, t+1) \\
& + \gamma \, \big(f_y(x+u^{k+1}, y+v^{k+1}, t+1) - f_y(x,y,t)\big) \\
& \quad \times f_{yx}(x+u^k, y+v^k, t+1)\Big) \\
& - \alpha \, \mathcal{L}_{\mathrm{NL}}(u^{k+1}, v^{k+1}),
\end{aligned} \tag{25}$$

$$\begin{aligned}
0 = \Psi'_{\mathrm{D}} \Big(& \big|f(x+u^{k+1}, y+v^{k+1}, t+1) - f(x,y,t)\big|^2 \\
& + \gamma \big|\nabla f(x+u^{k+1}, y+v^{k+1}, t+1) - \nabla f(x,y,t)\big|^2 \Big) \\
& \times \Big(\, (f(x+u^{k+1}, y+v^{k+1}, t+1) - f(x,y,t) \, \Big) \\
& \quad \times f_y(x+u^k, y+v^k, t+1) \\
& + \gamma \, \big(f_x(x+u^{k+1}, y+v^{k+1}, t+1) - f_x(x,y,t)\big) \\
& \quad \times f_{xy}(x+u^k, y+v^k, t+1) \\
& + \gamma \, \big(f_y(x+u^{k+1}, y+v^{k+1}, t+1) - f_y(x,y,t)\big) \\
& \quad \times f_{yy}(x+u^k, y+v^k, t+1)\Big) \\
& - \alpha \, \mathcal{L}_{\mathrm{NL}}(v^{k+1}, u^{k+1}).
\end{aligned} \tag{26}$$

- Secondly, we split up the unknown flow field u^{k+1} and v^{k+1} at this iteration step into the already known part u^k and v^k and the unknown motion increment du^k and dv^k. This allows us to linearise all constancy assumptions via a first order Taylor expansion of type

$$\begin{aligned}
& f_*(x+u^{k+1}, y+v^{k+1}, t+1) - f_*(x,y,t) \\
\approx \; & f_*(x+u^k, y+v^k, t+1) \\
& + f_{*x}(x+u^k, y+v^k, t+1) \, du \\
& + f_{*y}(x+u^k, y+v^k, t+1) \, dv \\
& - f_*(x,y,t) \\
= \; & \underbrace{f_*(x+u^k, y+v^k, t+1) - f_*(x,y,t)}_{\text{temporal difference !}} \\
& + f_{*x}(x+u^k, y+v^k, t+1) \, du \\
& + f_{*y}(x+u^k, y+v^k, t+1) \, dv,
\end{aligned} \tag{27}$$

where f_* denotes either f, f_x or f_y. Hereby one should note that these linearisations have been intentionally postponed from the modelling phase in order to allow for an correct handling of large displacements. Thus, we obtain the partially linearised fixed point iteration step

$$\begin{aligned}
0 = \Psi'_{\mathrm{D}} & \big((du^k, dv^k, 1) S^k (du^k, dv^k, 1)^\top\big) \\
& \times \big(S^k_{11} \, du^k + S^k_{12} \, dv^k + S^k_{13}\big) \\
& - \alpha \, \mathcal{L}_{\mathrm{NL}}(u^k + du^k, v^k + dv^k),
\end{aligned} \tag{28}$$

$$\begin{aligned}
0 = \Psi'_{\mathrm{D}} & \big((du^k, dv^k, 1) S^k (du^k, dv^k, 1)^\top\big) \\
& \times \big(S^k_{12} \, du^k + S^k_{22} \, dv^k + S^k_{23}\big) \\
& - \alpha \, \mathcal{L}_{\mathrm{NL}}(v^k + dv^k, u^k + du^k)
\end{aligned} \tag{29}$$

that is still nonlinear due to the expressions of type Ψ'_{D} and Ψ'_{S}. However, the strict usage of a convex functions in both the smoothness and the data term is rewarded by a unique solution of this step.

Here, once again the motion tensor notation has been used. The tensor S is thereby given by $S = \tilde{J} + \gamma \, \tilde{G}$ which is the weighted sum of the motion tensor for the grey value constancy assumption $\tilde{J} = \tilde{\nabla} f \tilde{\nabla} f^\top$ and the motion tensor for the gradient constancy assumption $\tilde{G} = \tilde{\nabla} f_x \tilde{\nabla} f_x^\top + \tilde{\nabla} f_y \tilde{\nabla} f_y^\top$. In contrast to the cases (a)–(f) where these assumptions have already been linearised in the model, not the original ∇ operator has to be considered, but a variant $\tilde{\nabla}$, where the third component is not a temporal derivative but a temporal difference; see (27). This shows that also in the case of variational techniques with originally nonlinearised constancy assumptions, we can keep to the simple and compact notation with motion tensors and differential operators (based on diffusion tensors).

- Thirdly, we embed this partly linearised fixed point iteration into a coarse-to-fine framework by connecting the variable k not only to the iteration index but also to a resolution level. While in the continuous setting this can only be realised by a scale-space focusing framework (Alvarez et al., 1999b), the discrete world offers us also multiresolution techniques for this purpose. Thereby, from one level to the next, each dimension is resampled with a downsampling factor ν. The actual resolution ratio between two levels is then given by ν^2. As shown in Brox et al. (2004) the incorporation of the partly linearised fixed point iteration into a coarse-to-fine strategy yields the well-known warping technique (Bergen et al., 1992; Black Anandan, 1991; Mémin and Pérez, 1998; Brox et al., 2004). In fact, before the equations (28)–(29) can be solved at a certain resolution level, all expressions of f that depend on u^k or v^k have to be recalculated. This in turn requires to compensate the second image given by $f(.,.t{+}1)$ for the already computed flow field which is nothing else than a warping step. In our case, such a warping step is realised by using a backward registration approach that is based on bilinear interpolation.

Let us once again point out the goals of the three previous steps: While the coarse-to-fine strategy can easily be identified with an unidirectional multigrid scheme that helps to *avoid local minima* during the optimisation, the incremental computation via a fixed point iteration is nothing else than an approximation of the nonlinearised constancy assumptions in the data term by means of *a series of linearised* ones. Although the coarse-to-fine technique works well in most of the cases, one should note that there exists no convergence proof for it. This results from the fact that motion increments on finer grids depend strongly on the computed increments from coarser ones. As a consequence, errors resulting from the bilinear interpolation or the handling of out-of-bound displacements may propagate in such a way, that they spoil the overall results completely.

4. Discretisation

4.1. General Discretisation Aspects

Let us now discuss a suitable discretisation for the Euler-Lagrange equations (a)–(g). To this end we consider the unknown functions $u(x, y, t)$ and $v(x, y, t)$ on a rectangular pixel grid with cell size $\mathbf{h} = (h_x, h_y)^\top$, and we denote by $u^{\mathbf{h}}_{i,j}$ the approximation to u at some pixel i, j with $i = 1,\ldots,N_x$ and $j = 1,\ldots,N_y$. Spatial derivatives of the image data are approximated using a fourth-order approximation with the stencil $(1, -8, 0, 8, -1)/(12h)$, while temporal derivatives are computed with a simple two-point stencil. If we denote the entries of the different diffusion tensors by

$$D =: \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{30}$$

we can discretise the divergence expressions in the differential operators L_{L} and L_{NL} by means of the following finite difference approximations:

$$\partial_x\,(\,a(x,y)\,\partial_x z(x,y)) \approx D_x^{-,\mathbf{h}}\left(M_x^{+,\mathbf{h}}(a_{i,j})\;D_x^{+,\mathbf{h}}(z_{i,j})\right), \tag{31}$$

$$\partial_x\left(\,b(x,y)\,\partial_y z(x,y)\right) \approx D_x^{\mathbf{h}}\quad\left(\qquad b_{i,j}\;\;D_y^{\mathbf{h}}\,(z_{i,j})\right), \tag{32}$$

$$\partial_y\,(\,b(x,y)\,\partial_x z(x,y)) \approx D_y^{\mathbf{h}}\quad\left(\qquad b_{i,j}\;\;D_x^{\mathbf{h}}\,(z_{i,j})\right), \tag{33}$$

$$\partial_y\left(\,c(x,y)\,\partial_y z(x,y)\right) \approx D_y^{-,\mathbf{h}}\left(M_y^{+,\mathbf{h}}(c_{i,j})\;D_y^{+,\mathbf{h}}(z_{i,j})\right). \tag{34}$$

Details on the required averaging and differential operators within the approximations are given in Table 1. One should note, that these discretisations can also be derived from discrete versions of the original energy functionals.

4.2. The Discrete Euler–Lagrange Equations

As we have seen before there are basically four types of Euler–Lagrange equations. Their discretisation is now discussed in detail.

(a)–(c) The Linear Case

We are now in the position to write down the discrete Euler-Lagrange equations for the linear case. They are given by

$$0 = J^{\mathbf{h}}_{11,i,j}\;u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{12,i,j}\;v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{13,i,j} - \alpha\;L^{\mathbf{h}}_{\mathrm{L\;i,j}}\;u^{\mathbf{h}}_{i,j}, \tag{44}$$

$$0 = J^{\mathbf{h}}_{12,i,j}\;u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{22,i,j}\;v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{23,i,j} - \alpha\;L^{\mathbf{h}}_{\mathrm{L\;i,j}}\;v^{\mathbf{h}}_{i,j}, \tag{45}$$

Table 1. Discretisations of averaging and differential operators.

One-sided averaging	$M_x^{\pm,\mathbf{h}}(z_{i,j})$	$:= \frac{z_{i\pm1,j}+z_{i,j}}{2}$	(35)
	$M_y^{\pm,\mathbf{h}}(z_{i,j})$	$:= \frac{z_{i,j\pm1}+z_{i,j}}{2}$	(36)
One-sided differences	$D_x^{\pm,\mathbf{h}}(z_{i,j})$	$:= \pm\frac{z_{i\pm1,j}-z_{i,j}}{h_x}$	(37)
	$D_y^{\pm,\mathbf{h}}(z_{i,j})$	$:= \pm\frac{z_{i,j\pm1}-z_{i,j}}{h_y}$	(38)
Central differences	$D_x^{\mathbf{h}}(z_{i,j})$	$:= \frac{z_{i+1,j}-z_{i-1,j}}{2h_x}$	(39)
	$D_y^{\mathbf{h}}(z_{i,j})$	$:= \frac{z_{i,j+1}-z_{i,j-1}}{2h_y}$	(40)
Squared differences	$D_x^{2,\mathbf{h}}(z_{i,j})$	$:= \frac{1}{2}\left(D_x^{+,\mathbf{h}}(z_{i,j})\right)^2 + \frac{1}{2}\left(D_x^{-,\mathbf{h}}(z_{i,j})\right)^2$	(41)
	$D_y^{2,\mathbf{h}}(z_{i,j})$	$:= \frac{1}{2}\left(D_y^{+,\mathbf{h}}(z_{i,j})\right)^2 + \frac{1}{2}\left(D_y^{-,\mathbf{h}}(z_{i,j})\right)^2$	(42)
Gradient magnitude	$\lvert D^{2,\mathbf{h}}(z_{i,j})\rvert$	$:= \sqrt{D_x^{2,\mathbf{h}}(z_{i,j}) + D_y^{2,\mathbf{h}}(z_{i,j})}$	(43)

for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$, where $L^{\mathbf{h}}_{\mathrm{L\ i,j}}$ denotes the discrete version of the corresponding linear operator $\mathcal{L}_{\mathrm{L}}$ at some pixel i, j. These $2N_xN_y$ equations constitute a *linear* system for the unknowns $u^{\mathbf{h}}_{i,j}$ and $v^{\mathbf{h}}_{i,j}$. One should note that there are two different types of coupling in the equations. The pointwise coupling between $u^{\mathbf{h}}_{i,j}$ and $v^{\mathbf{h}}_{i,j}$ in the data term and the neighbourhood coupling via the operator $L^{\mathbf{h}}_{\mathrm{L\ i,j}}$ in the smoothness term (within both equations separately).

(d)–(e) The Nonlinear Case I

Analogously, we discretise the Euler Lagrange equations for the nonlinear case I. The obtained *nonlinear* system of equations then reads

$$\begin{aligned} 0 = {} & J^{\mathbf{h}}_{11,i,j}\ u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{12,i,j}\ v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{13,i,j} && (46)\\ & -\alpha\ L^{\mathbf{h}}_{\mathrm{NL\ i,j}}\left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}\right)\ u^{\mathbf{h}}_{i,j}, && (47) \end{aligned}$$

$$\begin{aligned} 0 = {} & J^{\mathbf{h}}_{12,i,j}\ u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{22,i,j}\ v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{23,i,j} && (48)\\ & -\alpha\ L^{\mathbf{h}}_{\mathrm{NL\ i,j}}\left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}\right)\ v^{\mathbf{h}}_{i,j}, && (49) \end{aligned}$$

for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. Here, the finite difference approximation of $\mathcal{L}_{\mathrm{NL}}(u, v)$ and $\mathcal{L}_{\mathrm{NL}}(v, u)$ results in the product of a common nonlinear operator $L^{\mathbf{h}}_{\mathrm{NL\ i,j}}(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j})$ and the pixel $u^{\mathbf{h}}_{i,j}$ and $v^{\mathbf{h}}_{i,j}$, respectively. Evidently, this constitutes a third way of coupling.

(f) The Nonlinear Case II—The Method of Bruhn et al.

As in the previous case the discretisation of the Euler-Lagrange equations for the nonlinear case II yields a *nonlinear* system of equations. It is given by

$$\begin{aligned} 0 = {} & \Psi'_{\mathrm{D}}\left(\left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1\right) J^{\mathbf{h}}_{\rho}\ \left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1\right)^{\top}\right) && (50)\\ & \times\left(J^{\mathbf{h}}_{11,i,j}\ u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{12,i,j}\ v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{13,i,j}\right) \\ & -\alpha\ L^{\mathbf{h}}_{\mathrm{NL\ i,j}}\left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}\right)\ u^{\mathbf{h}}_{i,j}, && (51) \end{aligned}$$

$$\begin{aligned} 0 = {} & \Psi'_{\mathrm{D}}\left(\left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1\right) J^{\mathbf{h}}_{\rho}\ \left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1\right)^{\top}\right) && (52)\\ & \times\left(J^{\mathbf{h}}_{12,i,j}\ u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{22,i,j}\ v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{23,i,j}\right) \\ & -\alpha\ L^{\mathbf{h}}_{\mathrm{NL\ i,j}}\left(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}\right)\ v^{\mathbf{h}}_{i,j}, && (53) \end{aligned}$$

for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. One should note that the linear point coupling in the data term that appears in the cases (a)–(f) may become strongly nonlinear, since it it is now reweighted by the factor $\Psi'_{\mathrm{D}}\left((u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1) J^{\mathbf{h}}_{\rho}\ (u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)^{\top}\right)$ that depends nonlinearly on both $u^{\mathbf{h}}_{i,j}$ and $v^{\mathbf{h}}_{i,j}$.

(g) The Nonlinear Case III (Warping)—The Method of Papenberg et al.

Due to the hierarchical optimisation in the nonlinear case III one obtains not a single nonlinear equation system but a *hierarchy of nonlinear* equations system. For the warping level k the corresponding equation system is thereby given by

$$\begin{aligned} 0 = {} & \Psi'_{\mathrm{D}}\ \left(\left(du^{\mathbf{k,h}}_{i,j}, dv^{\mathbf{k,h}}_{i,j}, 1\right) S^{\mathbf{k,h}}_{i,j}\left(du^{\mathbf{k,h}}_{i,j}, dv^{\mathbf{k,h}}_{i,j}, 1\right)^{\top}\right) \\ & \times\left(S^{\mathbf{k,h}}_{11,i,j}\ du^{\mathbf{k,h}}_{i,j} + S^{\mathbf{k,h}}_{12,i,j}\ dv^{\mathbf{k,h}}_{i,j} + S^{\mathbf{k,h}}_{13,i,j}\right) \\ & -\alpha\ L^{\mathbf{k,h}}_{\mathrm{NL\ i,j}}\left(u^{\mathbf{k,h}}_{i,j} + du^{\mathbf{k,h}}_{i,j}, v^{\mathbf{k,h}}_{i,j} + dv^{\mathbf{k,h}}_{i,j}\right)\left(u^{\mathbf{k,h}}_{i,j} + du^{\mathbf{k,h}}_{i,j}\right), \end{aligned} \qquad (54)$$

$$\begin{aligned}0 = \Psi'_{\mathrm{D}} &\left(\left(du^{\mathbf{k},\mathbf{h}}_{i,j}, dv^{\mathbf{k},\mathbf{h}}_{i,j}, 1\right) S^{\mathbf{k},\mathbf{h}}_{i,j} \left(du^{\mathbf{k},\mathbf{h}}_{i,j}, dv^{\mathbf{k},\mathbf{h}}_{i,j}, 1\right)^{\top}\right)\\ &\times\left(S^{\mathbf{k},\mathbf{h}}_{12,i,j}\, du^{\mathbf{k},\mathbf{h}}_{i,j} + S^{\mathbf{k},\mathbf{h}}_{22,i,j}\, dv^{\mathbf{k},\mathbf{h}}_{i,j} + S^{\mathbf{k},\mathbf{h}}_{23,i,j}\right)\\ &-\alpha\, L^{\mathbf{k},\mathbf{h}}_{\mathrm{NL\ i,j}}\left(u^{\mathbf{k},\mathbf{h}}_{i,j} + du^{\mathbf{k},\mathbf{h}}_{i,j}, v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}\right)\left(v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}\right),\end{aligned} \tag{55}$$

where $i = 1, .., N_x$ and $j = 1, .., N_y$. However, in contrast to all other cases, the $2N_xN_y$ unknowns are this time given by the the variables $du^{\mathbf{k},\mathbf{h}}_{i,j}$ and $dv^{\mathbf{k},\mathbf{h}}_{i,j}$ for the motion increment. Moreover, one should note that due to the use of the warping strategy the properties of the resulting nonlinear equation systems may be significantly different from level to level.

5. Multigrid

5.1. Basic Concept

In general, the preceding linear and nonlinear systems of equations are solved by using non-hierarchical iterative schemes; e.g. variants of the Jacobi or the Gauß-Seidel method (Ortega and Rheinboldt, 2000; Young, 1971). However, such techniques are not well–suited for equation systems that are only coupled via a small local neighbourhood: It may take thousands of iterations to transport local information between unknowns that are not coupled directly. A Fourier analysis of the error confirms this observation: While high frequency components (small wavelength, local impact) are attenuated efficiently, lower frequency components (large wavelength, global impact) remain almost un-dampened. In order to overcome this problem multigrid methods (Brandt, 1977; Briggs et al., 2000; Hackbusch, 1985; Trottenberg et al., 2001; Wesseling 1992) are based on a sophisticated strategy. They make use of correction steps that compute the error (not a coarser version of the fine grid solution) on a coarser grid. Thus, lower frequency components of the error reappear as higher ones and allow for an efficient attenuation with standard iterative methods. In the following we explain this strategy in detail for both the linear and the nonlinear case by the example of a basic bidirectional two-grid cycle.

5.2. The Linear Two-Grid Cycle

For the sake of clarity, let us reformulate the linear equation systems of the methods (a)–(c) as

$$A^{\mathbf{h}}x^{\mathbf{h}} = f^{\mathbf{h}}. \tag{56}$$

Here $x^{\mathbf{h}}$ denotes the concatenated vector $((u^{\mathbf{h}})^{\top}, (v^{\mathbf{h}})^{\top})^{\top}$, $A^{\mathbf{h}}$ is a symmetric positive definite matrix and $f^{\mathbf{h}}$ stands for the right hand side.

I) Multigrid methods starts by performing several iterations with a basic iterative solver. This is the so-called presmoothing relaxation step, where high frequency components of the error are removed. If we denote the result after these iterations by $\tilde{x}^{\mathbf{h}}$, the error is given by

$$e^{\mathbf{h}} = x^{\mathbf{h}} - \tilde{x}^{\mathbf{h}}. \tag{57}$$

II) Evidently, one is interested in finding $e^{\mathbf{h}}$ in order to correct the approximated solution $\tilde{x}^{\mathbf{h}}$. Although $e^{\mathbf{h}}$ cannot be computed directly, the linearity of $A^{\mathbf{h}}$ allows its computation via

$$\begin{aligned}A^{\mathbf{h}}e^{\mathbf{h}} &= A^{\mathbf{h}}(x^{\mathbf{h}} - \tilde{x}^{\mathbf{h}}) = A^{\mathbf{h}}x^{\mathbf{h}} - A^{\mathbf{h}}\tilde{x}^{\mathbf{h}}\\ &= f^{\mathbf{h}} - A^{\mathbf{h}}\tilde{x}^{\mathbf{h}} = r^{\mathbf{h}},\end{aligned} \tag{58}$$

where $r^{\mathbf{h}}$ is called residual. Since high frequencies of the error have already been removed, we can speed up the computation by solving this equation system at a coarser resolution with grid cell size $\mathbf{H} = (H_x, H_y)^{\top}$:

$$A^{\mathbf{h}}e^{\mathbf{h}} = r^{\mathbf{h}} \quad\rightarrow\quad A^{\mathbf{H}}e^{\mathbf{H}} = r^{\mathbf{H}}. \tag{59}$$

One should note that at this point, a transfer of the original equation system to a coarser grid makes no sense: Unlike the error, the solution very probably contains (desired) high frequency components. A restriction of these components would severely deteriorate the approximative solution (aliasing).

III) After we have solved the residual equation system on the coarse grid with a method of our choice, we transfer the solution back to the fine grid and correct our approximation by the computed error

$$\tilde{x}^{\mathbf{h}}_{\mathrm{new}} = \tilde{x}^{\mathbf{h}} + \tilde{e}^{\mathbf{h}}. \tag{60}$$

One should note that the interpolated coarse grid solution is denoted by $\tilde{e}^{\mathbf{h}}$, since it is only an approximation to the desired correction $e^{\mathbf{h}}$.

IV) In general, the transfer of the computed correction from a coarse grid by means of interpolation introduces some new high frequency components. To this end, a so-called postsmoothing relaxation step is performed, where once again some iteration of the basic iterative solver are applied.

5.3. *The Nonlinear (FAS) Two-Grid Cycle*

Also in this case, let us start with a reformulation of the nonlinear equation system resulting from the methods (d)–(g) as

$$A^{\mathbf{h}}(x^{\mathbf{h}}) = f^{\mathbf{h}} \tag{61}$$

where $A^{\mathbf{h}}(x^{\mathbf{h}})$ is a nonlinear operator. The FAS strategy (Brandt, 1977) works as follows:

I) We perform a presmoothing relaxation step with a nonlinear basic solver.

II) However, since $A^{\mathbf{h}}(x^{\mathbf{h}})$ is a nonlinear operator, the way of deriving a suitable coarse grid correction is significantly different from the linear case. The (implicit) relation between the error and the residual is given by

$$A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}} + e^{\mathbf{h}}) - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = f^{\mathbf{h}} - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = r^{\mathbf{h}}. \tag{62}$$

In order to compute the desired correction we transfer the following nonlinear equation system to the coarse grid

$$A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}} + e^{\mathbf{h}}) = r^{\mathbf{h}} + A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) \tag{63}$$

$$\rightarrow \; A^{\mathbf{H}}(\boxed{\tilde{x}^{\mathbf{H}}} + e^{\mathbf{H}}) = r^{\mathbf{H}} + \boxed{A^{\mathbf{H}}(\tilde{x}^{\mathbf{H}})}. \tag{64}$$

Here, frames visualise the additional terms compared to the linear case.

III) After we have solved the nonlinear residual equation system on the coarse grid, we subtract $\tilde{x}^{\mathbf{H}}$ from the solution in order to obtain $e^{\mathbf{H}}$. Its transfer to the fine grid then allows to perform the correction step.

IV) We perform a postsmoothing relaxation step with a nonlinear basic solver.

5.4. *Advanced Multigrid Strategies*

In order to increase the computational efficiency, the presented two-grid cycles are generally applied in a hierarchical way. While *V–cycles* make one recursive call of a two-grid cycle per level, faster converging *W–cycles* perform two. Nevertheless, multiple of such advanced cycles are required to reach the desired accuracy. As already discussed in the introduction it makes sense to combine these bidirectional schemes with the hierarchical initialisation properties of unidirectional multigrid methods. Refining the original problem step by step (unidirectional approach) and solving the resulting linear or nonlinear equation system at each level by using some V– or W–cycles (bidirectional approach), the multigrid strategy with the best performance is obtained: *full multigrid* (Briggs et al., 2000). An overview of all three types of multigrid strategies is given in Fig. 1. One should note than in the nonlinear case III (warping) such a full multigrid scheme

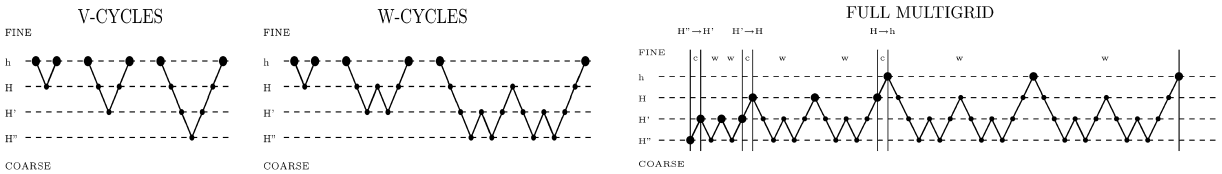

Figure 1. *Left:* Example for V-cycles with two, three and four levels. *Center:* Ditto for W-cycles. *Right:* Full multigrid implementation with 2 W-cycles per resolution level. Refinement steps are marked with 'c'. Each W-cycle is marked with a 'w'. Adapted from Bruhn et al. (2005a) and Bruhn et al. (2003).

Table 2. Implemented multigrid schemes the different variational models.

Case	Model	MG Solver	Cyc	Basic Solver	Pre/Post
L	(a) Homogeneous	FMG-W	1	GS-CPR	1–1
	(b) Image-Driven Isotropic	FMG-W	2	GS-CPR	2–2
	(c) Image-Driven Anisotropic	FMG-W	4	GS-ALR	1–1
NL I	(d) Flow-Driven Isotropic	FAS-FMG-W	2	GS-CPR	2–2
	(e) Flow-Driven Anisotropic	FAS-FMG-W	4	GS-ALR	1–1
NL II	(f) Bruhn et al.	FAS-FMG-W	2	GS-CPR	2–2
NL III	(g) Papenberg et al.	WARP-FAS-FMG-W	2	GS-CPR	6–6

MG = multigrid. FMG = full multigrid. WARP = warping. Cyc = multigrid cycles per level. GS = Gauß–Seidel. CPR = coupled point relaxation. Pre/Post = pre- and postsmoothing relaxation iterations. L = linear. NL = nonlinear.

has to be applied at *each* resolution level. This means we combine two strategies in this case: On one hand, we approximate the nonlinearised constancy assumptions in the data term by a series of linearised constancy assumptions; see Subsection 3.1(f). This scheme is embedded in a unidirectional multigrid framework to avoid local minima. On the other hand we use a full multigrid method to solve the resulting nonlinear equation system at *each* level of this unidirectional framework.

5.5. Implementation Details

Let us now discuss some implementation details. As one can see from Table 2 we have developed full multigrid schemes for all linear and nonlinear cases. Thereby we used two different types of basic solvers: While in the cases of homogeneous and isotropic regularisation, a Gauß–Seidel solver with coupled point relaxation (CPR) (Bruhn et al., 2005a) was sufficient, the anisotropy of the neighbourhood coupling in the remaining methods required the use of a Gauß–Seidel solver with alternating line relaxation (ALR) (Wesseling, 1992). Instead of updating the two unknowns u and v at each pixel at the same time (CPR), the ALR method computes whole lines of unknowns simultaneously. Thereby three directions were considered: Lines in $x-$ and y–direction as well as the direction of the different unknowns at each pixel, namely (u, v) itself. For the nonlinear variants of the Gauß–Seidel solver we used the strategy of frozen coefficients (Frohn-Schnauff et al., 2004). In the literature this technique is also known as lagged diffusivity method (Chan and Mulet, 1999) or Quasi-Newton scheme (Vogel, 2002). Direct nonlinear Gauß–Seidel Newton methods (Briggs et al., 2000) have not been considered as basic solver. Experiments using this kind of methods have shown a similar performance in terms of error reduction, however, at the the expense of significantly increased computational costs. One can also see from Table 2 that an increasing anisotropy of the diffusion tensor (homogeneous → isotropic → anisotropic) required more multigrid cycles at each level of the full multigrid implementation. In the case of the method of Papenberg et al. one can also observe an increased number of pre- and postsmoothing relaxations. This can be explained by the combination of the warping technique and the strongly nonlinear flow-driven regulariser (TV).

In order to allow for a complete multigrid hierarchy we considered the use of non-dyadic intergrid transfer operators in all approaches. As proposed in (Bruhn et al., 2005a) they were realised by constant interpolation and simple averaging. Coarser versions of the linear and nonlinear operators were created by a coarse grid approximation approach (DCA) (Wesseling, 1992). To this end, we restricted all linear expression such as the motions tensors and the linear diffusion tensors and adapted the grid size for the nonlinear ones. At this point one should note that the positive semi-definiteness of both tensors is essential to ensure convergence of the basic solvers. Therefore, all tensor entries were restricted *channelwise*:

$$J^{\mathbf{h}}_{nm} \longrightarrow J^{\mathbf{H}}_{nm} \qquad n, m = 1\,,\,..\,,3. \tag{65}$$

In combination with the use of a *linear* restriction operator such as simple averaging this ensures that the the resulting coarse grid tensors remain positive semidefinite.

6. Experiments

Let us now evaluate the different multigrid implementations. To this end, all computations are carried out on a standard desktop PC with a 3.06 GHz Intel Pentium 4 CPU executing C / C++ code.

In our first experiment we compare the efficiency of different numerical schemes for the five prototypes of regularisation strategies that have been discussed in Section 2. Apart from the developed full multigrid schemes we also implemented stand-alone versions of their basic iterative solvers, namely the Gauß-Seidel methods with alternating line relaxation (ALR) and the Gauß-Seidel method with coupled point relaxation (CPR). Moreover, we considered unidirectional multigrid variants of the basics solvers—so-called *cascadic* multigrid methods (Bornemann and Deuflhard, 1996)—as well as modified explicit schemes (Weickert and Schnörr, 2001). Compared to ordinary explicit schemes (e.g. gradient descent methods) such modified schemes allow for larger time step sizes τ. For our evaluation we used a 160 × 120 real-world sequence in which a person dances in front of the camera. Before we applied the different numerical schemes we preprocessed the sequence by convolution with a Gaussian kernel of standard deviation $\sigma = 1$. The iterations were stopped when the relative error $e_{rel} := \|x - \tilde{x}_n\|_2 / \|x\|_2$ dropped below 10^{-2}, where x denotes the correct solution and $\tilde{x}_n$ stands for the computed result after n iterations/ cycles. In all cases a zero flow field was used as initialisation.

Table 3 shows the excellent performance of the proposed numerical schemes for all five types of regularisers. In the linear cases (a), (b) and (c) the modified explicit schemes and the basic iterative solvers are outperformed by *two to three* and *one to two orders of magnitude*, respectively. Apart from the linear case (a) where the result is rather smooth, also the unidirectional variants are clearly outperformed by *one order of magnitude*. This is reflected in high frame rates of up to 63 dense flow fields per second. In the nonlinear cases (d) and (e), our comparison shows a very similar tendency. Here, speedup factors are in the range of *two to three orders of magnitude* for the modified explicit schemes and the basic solvers and again *one order of magnitude* compared to the unidirectional multigrid schemes. Frame rates of twelve and two dense flow fields per second clearly demonstrate that also in this case real-time performance is well within our computational reach. One should note that for all five methods only a single full multigrid cycle was sufficient, while in particular the non-hierarchical iterative methods required thousands of iterations.

In our second experiment we juxtapose the estimation quality of the proposed multigrid implementations for the different regularisation strategies. In particular, the comparison of the four discontinuity-preserving real-time approaches (b)–(f) to the multigrid implementation in (a) with homogeneous regularisation (Bruhn et al., 2005a) is thereby of interest. To this end, we have computed flow fields for three different real-world sequences: for the previously used *Dancing Sequence* (complex motion), the *Waving Sequence* (translations and discontinuities) and the *Rotating Thumb Sequence* (rotation). The depicted colour plots in Figs. 2 and 3 make the qualitative progress in the field of real-time variational optic flow computation explicit: One can easily see, that image- and flow-driven methods yield results that are much more accurate, since the underlying regularisation strategies allow for a preservation of motion boundaries and discontinuities. Moreover, one can observe that the anisotropic techniques give slightly better results than the isotropic ones and that the nonlinear methods are able to overcome the problem of oversegmentation that lies in the nature of image-driven techniques in the presence of textured scenes.

In our third experiment we investigate the efficiency of our multigrid implementations for the more advanced variational optic flow methods discussed in Section 3: The noise robust CLG approach of Bruhn et al. (2005c) and the highly accurate optic flow technique of Papenberg et al. (2005). As test sequence in this experiment served a downsampled variant (160 × 120) of the *Rheinhafen* sequence by Nagel which is available at http://i21www.ira.uka.de/ image_sequences. As before, a relative error of $e_{rel} := 10^{-2}$ was used as stopping criterion. One should note that in the case of the method of Papenberg et al. this relative error does not only refer to a single nonlinear system of equations. Here, the coarse-to-fine optimisation by means of the warping strategy requires the solution of whole hierarchy of equation systems. This constitutes a significant difference all methods that have been previously discussed in this paper including the recent technique of Bruhn et al. In particular, one should note that warping errors on coarser levels influence the results on finer levels such that errors can propagate in this case.

Table 3. Performance benchmark for all types of regularisers on a standard desktop computer with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field from the 160 × 120 dancing sequence. FPS = frames per second.

Solver	Iterations	Time [s]	FPS [s^{-1}]	Speedup
(a) Linear: Homogeneous regularisation (Horn and Schunck) $\sigma = 1.0, \alpha = 1000$				
Mod. Explicit Scheme ($\tau = 0.25$)	4425	3.509	0.285	1
Gauß-Seidel (CPR)	2193	1.152	0.868	3
Cascadic Gauß-Seidel (CPR)	16	0.018	56.189	197
Full Multigrid	**1**	**0.016**	**62.790**	**220**
(b) Linear: Image-driven isotropic regularisation (Charbonnier) $\sigma = 1.0, \alpha = 1000, \epsilon_{\mathrm{S}} = 1.0$				
Mod. Explicit Scheme ($\tau = 0.25$)	8896	12.048	0.083	1
Gauß-Seidel (CPR)	2856	2.793	0.358	4
Cascadic Gauß-Seidel (CPR)	215	0.085	3.508	42
Full Multigrid	**1**	**0.048**	**20.850**	**251**
(c) Linear: Image-driven anisotropic regularisation (Nagel-Enkelmann) $\sigma = 1.0, \alpha = 1000, \epsilon_{\mathrm{S}} = 10^{-2}$				
Mod. Explicit Scheme ($\tau = 0.1666$)	36433	47.087	0.021	1
Gauß-Seidel (ALR)	607	3.608	0.277	13
Cascadic Gauß-Seidel (ALR)	473	3.218	0.311	15
Full Multigrid	**1**	**0.171**	**5.882**	**275**
(d) Nonlinear: Flow-driven isotropic regularisation (TV) $\sigma = 1.0, \alpha = 10, \epsilon_{\mathrm{S}} = 10^{-2}$				
Mod. Explicit Scheme ($\tau = 0.0025$)	10633	30.492	0.033	1
Gauß-Seidel (CPR)	2679	6.911	0.145	4
Cascadic Gauß-Seidel (CPR)	371	0.853	1.173	36
FAS - Full Multigrid	**1**	**0.082**	**12.172**	**372**
(e) Nonlinear: Flow-driven anisotropic regularisation (TV) $\sigma = 1.0, \alpha = 10, \epsilon_{\mathrm{S}} = 10^{-2}$				
Mod. Explicit Scheme ($\tau = 0.0025$)	9224	58.837	0.017	1
Gauß-Seidel (ALR)	591	12.508	0.080	5
Cascadic Gauß-Seidel (ALR)	138	3.816	0.262	15
FAS - Full Multigrid	**1**	**0.491**	**2.038**	**120**

Let us now take a look at the obtained results in Table 4. As one can see, the speedups for the more advanced optic flow methods are even better than for the basic techniques with different types of regularisation. With *three to four orders of magnitude* the modified explicit scheme (that needs almost one hundred thousand iterations) is outperformed more than significantly. The same applies to the basic basic iterative solvers, although the speed up is here *two orders of magnitude*. Even in the case of unidirectional multigrid methods one can observe a significant speed up. It is in the range of *one order of magnitude*. The corresponding frame rates show clearly that in the case of such highly advanced optic flow methods, real-time performance is still possible.

In Fig. 4 the computed flow fields are depicted. Evidently they look fairly realistic: The motion of the van in the foreground as well as the motion of all other vehicles in the background is computed with good precision. Moreover, object boundaries within the flow field are rather sharp and allow for a simple separation of the different motions layers via thresholding. This segmentation-like behaviour, that is desired in many optic flow applications, is a direct consequence of using TV as discontinuity-preserving regulariser.

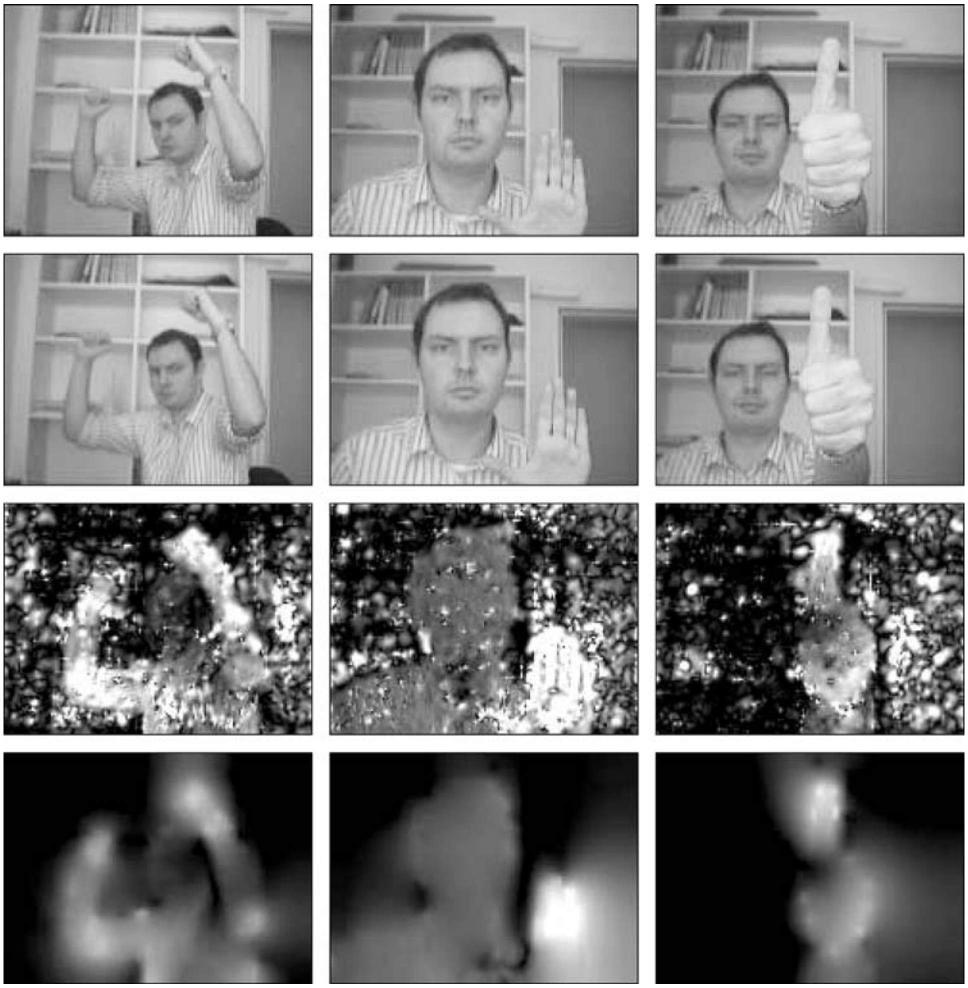

Figure 2. *Left to right:* dancing sequence, waving sequence, rotating thumb sequence. *Top to bottom:* first frame, second frame, no regularisation (normal flow), homogeneous regularisation (Horn and Schunck). *Brightness code:* The magnitude of a flow vector is encoded by its brightness. Brighter pixels stand for larger displacements. Color versions of the flow fields are available at http://www.mia.uni-saarland.de/bruhn/ijcv05/flowfields/.

In our final experiment we evaluate the accuracy of our real-time implementations for the two previously discussed optic flow methods. To this end, we have considered the famous *Yosemite* test sequence *with clouds*. This sequence that was created by Lynn Quam is very popular due to the fact that it combines translative and divergent motion under varying illumination. In Table 5 the computed average angular errors (Barron et al., 1994) for both approaches are presented where they are compared to the best results from the literature. The raw numbers show that the developed multigrid schemes do not have any negative effect on the quality of the results. However, there are small differences in terms of the average angular error: While in the case of the method of Bruhn et al. we obtained a slightly lower average angular error by using the total variation (TV) instead of the Charbonnier function as flow-driven isotropic regulariser, the relatively small coarsening factor of $\nu = 0.65$ limited the average angular error to 2.52° in the case of the method of Papenberg et al. A larger coarsening factor ν close to 1 would of course allow to obtain the original average angular error, however, at the expense of much higher computational costs (since the number of nonlinear system of equations would increase significantly).

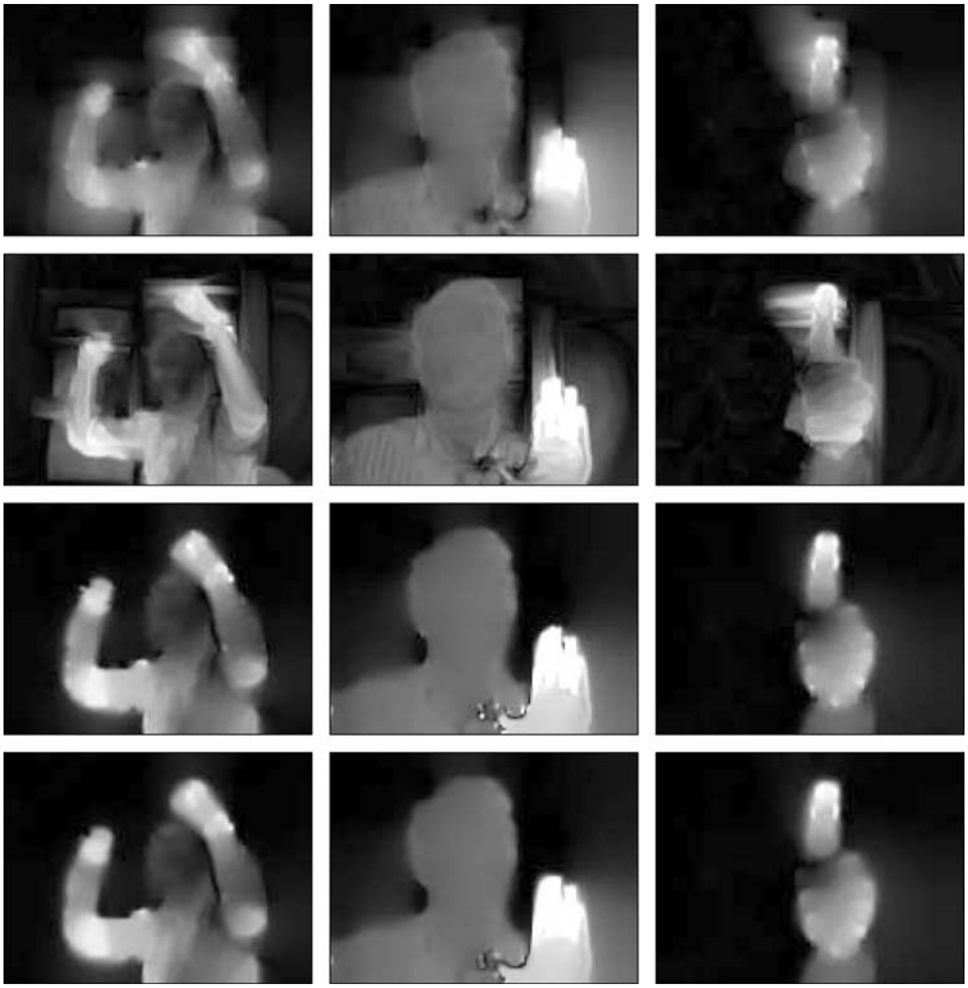

Figure 3. *Left to right:* dancing sequence, waving sequence, rotating thumb sequence. *Top to bottom:* data-driven isotropic regularisation (Charbonnier), data-driven anisotropic regularisation (Nagel-Enkelmann), flow-driven isotropic regularisation (TV), flow-driven anisotropic regularisation (TV). *Brightness code:* The magnitude of a flow vector is encoded by its brightness. Brighter pixels stand for larger displacements. Color versions of the flow fields are available at http://www.mia.uni-saarland.de/bruhn/ijcv05/flowfields/.

The flow fields computed by both approaches are shown in Fig. 5. Apart from these flow fields also the ground truth solution is depicted that allows to access the quality of the real-time capable methods. As one can see, already the method of Bruhn et al. gives relatively good results. The method of Papenberg et al., however, does match the ground truth almost perfectly.

7. Summary and Conclusions

In this paper we presented a unifying multigrid approach to variational optic flow computation in real-time. This was accomplished by introducing the systematic notation of motion and diffusion tensors and deriving highly efficient bidirectional multigrid approaches to solve the resulting linear and nonlinear systems of equations. We showed that by carefully designing such multigrid methods a variety of discontinuity-preserving optic flow techniques can be implemented in real-time. Moreover, we extended our approach to two recent and more advanced variational optic flow methods. Experiments demonstrated that compared to classical iterative solvers speedups of two to four orders magnitude can be achieved. Even

Table 4. Performance benchmark for the more advanced optic flow methods: The nonlinear variant of the CLG method by Bruhn et al. (2005) and the method of Papenberg et al. (2005) (see also (Brox et al. 2004)). Benchmark was performed on a standard desktop computer with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field from the downsampled *Rheinhafen* sequence (size 160 × 120). FPS = frames per second.

Solver	Iterations	Time [s]	FPS [s^{-1}]	Speedup
(f) Bruhn et al.: Robust data term with local integration + flow-driven isotropic regularisation (TV) $\sigma = 0.0, \rho = 1.0, \alpha = 15, \epsilon_D = 10^{-1}, \epsilon_S = 10^{-3}$				
Mod. Explicit Scheme ($\tau = 0.00025$)	81064	246.812	0.004	1
Gauß-Seidel (CPR)	3720	9.524	0.105	26
Cascadic Gauß-Seidel (CPR)	138	0.409	2.448	603
FAS - Full Multigrid	**1**	**0.087**	**11.473**	**2836**
(g) Papenberg et al.: Robust nonlinearised data term with additional gradient constancy + flow-driven isotropic regularisation (TV) $\sigma = 1.0, \alpha = 121, \gamma = 230, \nu = 0.65, \epsilon_D = 10^{-1}, \epsilon_S = 10^{-3}$				
Mod. Explicit Scheme ($\tau = 0.00025$)	79112	445.831	0.002	1
Gauß-Seidel (CPR)	6549	34.483	0.029	13
Cascadic Gauß-Seidel (CPR)	427	2.670	0.375	167
FAS - Full Multigrid	**1**	**0.396**	**2.527**	**1127**

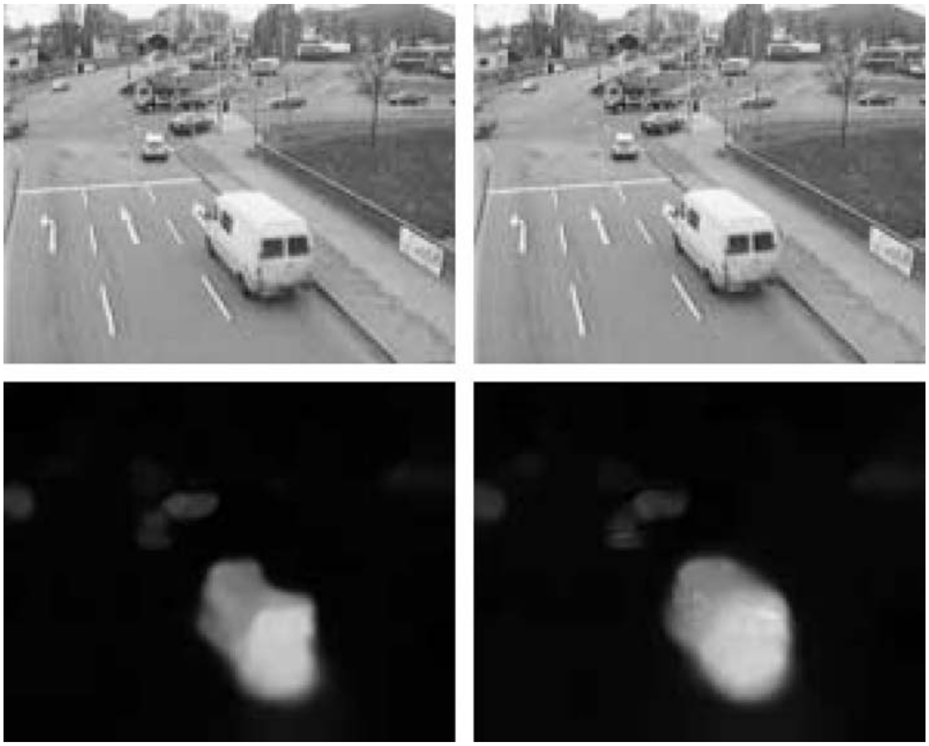

Figure 4. *(a) Top Left:* Frame 1130 of the *Rheinhafen* sequence by Nagel (resized to 160 × 120). *(b) Top Right:* Frame 1131. *(c) Bottom Left:* Computed flow field by the multigrid implementation of the 2-D method of Bruhn et al. Computing time: 87 milliseconds. *(d) Bottom Right:* Computed flow field by the multigrid implementation of the 2-D method of Papenberg et al. Computing time: 396 milliseconds.

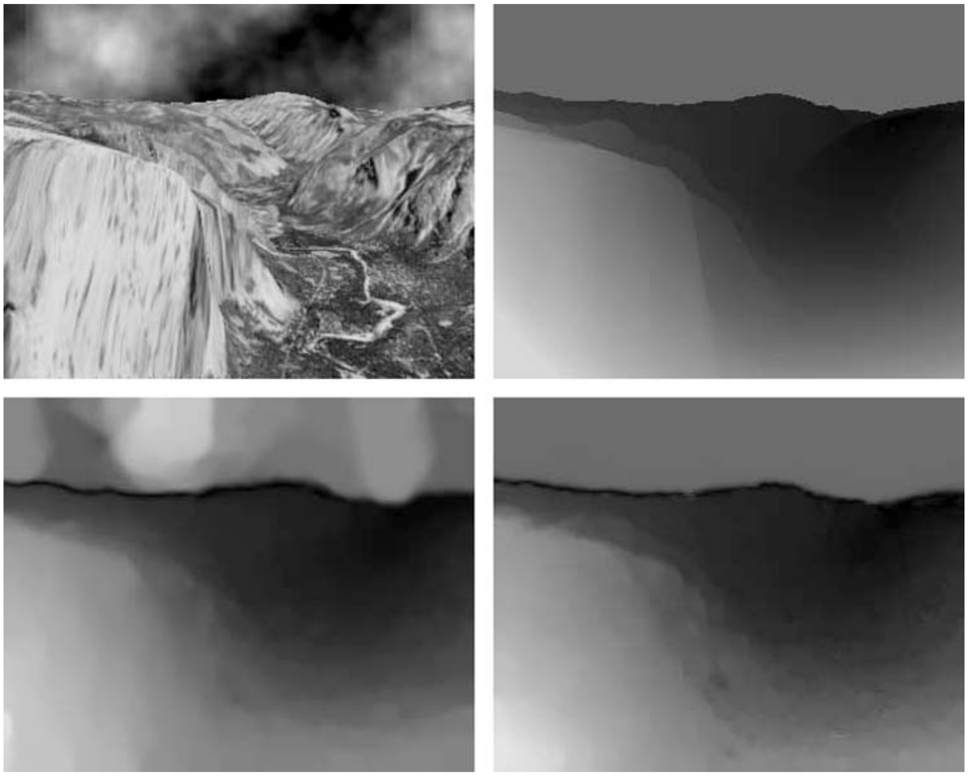

Figure 5. *(a) Top Left:* Frame 8 of the *Yosemite* sequence with clouds by Lynn Quam (size 312 × 256). *(b) Top Right:* Ground truth flow field. *(c) Bottom Left:* Computed flow field by our CLG 2-D multigrid implementation without warping. Average angular error: 5.77°. Computing time: 384 milliseconds. *(d) Bottom Right:* Computed flow field by our Papenberg et al. 2-D multigrid implementation. Average angular error: 2.51°. Computing time: 1814 milliseconds.

Table 5. Qualitative comparison between results from the literature with 100 % density and the results for our multigrid implementations. AAE = average angular error. STD = standard deviation. 2D = spatial smoothness assumption. 3D = spatio-temporal smoothness assumption.

Yosemite with clouds

Technique	AAE	STD
Anandan (Baron et al., 1994)	13.36°	15.64°
Nagel (Baron et al., 1994)	10.22°	16.51°
Horn–Schunck, mod. (Baron et al., 1994)	9.78°	16.19°
Uras et al. (Barron et al., 1994)	8.94°	15.61°
Bruhn et al. nonlinear (2D) (Bruhn et al., 2005c)	6.03°	8.61°
Bruhn et al. nonlinear – Multigrid (2D)	**5.77°**	**8.47°**
Alvarez et al. (2000)	5.53°	7.40°
Mémin–Pérez (1998)	4.69°	6.89°
Papenberg et al. – Multigrid (2D)	**2.52°**	**6.58°**
Papenberg et al. (2005) (2D)	2.44°	6.90°
Papenberg et al. (2005) (3D)	1.78°	7.00°
Bruhn and Weickert (2005) (3D)	1.72°	6.88°

unidirectional multigrid methods that are a frequent tool in the motion estimation community are clearly outperformed by one order of magnitude. This shows that high quality optic flow computation and real-time performance are not opposing worlds. They can be combined if recent optic flow methods are implemented by means of highly efficient numerical schemes.

The investigation of suitable paralleliaation strategies is ongoing work (Kohlberger et al. 2004, 2005). Their usage would allow to process even high-resolution video sequences in real-time.

Acknowledgments

Our optic flow research is partly funded by the *Deutsche Forschungsgemeinschaft (DFG)* under the project WE 2602/3-1. This is gratefully acknowledged. Moreover, we thank the reviewers for their useful remarks.

References

Alvarez, L., Esclarín J., Lefébure M., and Sánchez J. 1999a. A PDE model for computing the optical flow. In *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pp. 1349–1356, Las Palmas de Gran Canaria, Spain, Sept.

Alvarez, L., Weickert, J., and Sánchez J. 1999b. A scale-space approach to nonlocal optical flow calculations. In M. Nielsen, P. Johansen, O.F. Olsen, and J. Weickert, (eds.), *Scale-Space Theories in Computer Vision*, vol. 1682 of *Lecture Notes in Computer Science*, pp. 235–246. Springer, Berlin.

Alvarez, L., Weickert, J., and Sánchez J. 2000. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56.

Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310.

Barron, J.L., Fleet, D.J., and Beauchemin, S.S. 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

Bergen, J.R., Anandan, P., Hanna, K.J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In F. Hodnett, (ed.), *Proc. Sixth European Conference on Mathematics in Industry*, pp. 237–252. Teubner, Stuttgart.

Black, M.J. and Anandan, P. 1996. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Black, M.J. and Anandan, P. 1991. Robust dynamic motion estimation over time. In *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 292–302, Maui, HI, IEEE Computer Society Press.

Bornemann, F. and Deuflhard, P. 1996. The cascadic multigrid method for elliptic problems. *Numerische Mathematik*, 75:135–152.

Borzi, A., Ito, K., and Kunisch, K. 2002. Optimal control formulation for determining optical flow. *SIAM Journal on Scientific Computing*, 24(3):818–847.

Brandt, A. 1977. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390.

Briggs, W.L., Henson, V.E., and McCormick S.F. 2000. *A Multigrid Tutorial*. 2 edition, SIAM, Philadelphia.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. 2004. High accuracy optic flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision – ECCV 2004*, vol. 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, Berlin.

Bruhn, A. and Weickert, J. 2005. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proc. Tenth International Conference on Computer Vision*, vol. 1, pages 749–755, Beijing, China, IEEE Computer Society Press.

Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., and Schnörr C. 2003. Real-time optic flow computation with variational methods. In N. Petkov and M.A. Westberg, editors, *Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pp. 222–229. Springer, Berlin.

Bruhn, A., Weickert, J., and Schnörr C. 2005c. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.

Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., and Schnörr C. 2005a. Variational optical flow computation in real-time. *IEEE Transactions on Image Processing*, 14(5):608–615.

Bruhn, A., Weickert, J., Kohlberger, T., and Schnörr C. 2005b. Discontinuity-preserving computation of variational optic flow in real-time. In R. Kimmel, N. Sochen, and J. Weickert, (eds.), *Scale-Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pp. 279–290. Springer, Berlin.

Chan, R.H., Chan, T.F., and Wan, W.L. 1997. Multigrid for differential-convolution problems arising from image processing. In G. Golub, S.H. Lui, F. Luk, and R. Plemmons, (eds.), *Proc. Workshop on Scientific Computing*, pp. 58–72, Hong Kong.

Chan, T.F. and Mulet, P. 1999. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367.

Charbonnier, P., Blanc-Féraud L., Aubert, G., and Barlaud, M. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. 1994 IEEE International Conference on Image Processing*, vol. 2, pp. 168–172, Austin, TX, IEEE Computer Society Press.

Cohen, I. 1993. Nonlinear variational method for optical flow computation. In *Proc. Eighth Scandinavian Conference on Image Analysis*, vol. 1, pp. 523–530, Tromsø, Norway.

Deriche, R., Kornprobst, P., and Aubert, G. 1995. Optical-flow estimation while preserving its discontinuities: a variational approach. In *Proc. Second Asian Conference on Computer Vision*, vol. 2, pp. 290–295, Singapore.

El Kalmoun M. and Rüde U. 2003. A variational multigrid for computing the optical flow. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, (eds.), *Vision, Modelling and Visualization*, pp. 577–584. IOS Press.

Elsgolc, L.E. 1961. *Calculus of Variations*. Pergamon, Oxford.

Enkelmann, W. 1987. Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics and Image Processing*, 43:150–177.

Farnebäck G. 2001. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *Proc. Eighth International Conference on Computer Vision*, vol. 1, pp. 171–177, Vancouver, Canada, IEEE Computer Society Press.

Frohn-Schnauf C., Henn, S., and Witsch, K. 2004. Nonlinear multigrid methods for total variation denoising. *Computating and Visualization in Sciene*, 7(3-4):199–206.

Förstner W. and Gülch E. 1987. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pp. 281–305, Interlaken, Switzerland.

Ghosal, S. and Vaněk P. Č. 1996. Scalable algorithm for discontinuous optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):181–194.

Glazer, F. 1984. Multilevel relaxation in low-level computer vision. In A. Rosenfeld, (ed.), *Multiresolution Image Processing and Analysis*, pp. 312–330. Springer, Berlin.

Hackbusch, W. 1985. *Multigrid Methods and Applications*. Springer, New York.

Horn, B. and Schunck, B. 1981. Determining optical flow. *Artificial Intelligence*, 17:185–203.

Huber, P.J. 1981. *Robust Statistics*. Wiley, New York.

Kimmel, R. and Yavneh, I. 2003. An algebraic multigrid approach for image analysis. *SIAM Journal on Scientific Computing*, 24(4):1218–1231.

Kohlberger, T., Schnörr C., Bruhn, A., and Weickert, J. 2004. Parallel variational motion estimation by domain decomposition and cluster computing. In T. Pajdla and J. Matas, (eds.), *Computer Vision – ECCV 2004*, volume 3024 of *Lecture Notes in Computer Science*, pages 205–216. Springer, Berlin.

Kohlberger, T., Schnörr C., Bruhn, A., and Weickert, J. 2005. Domain decomposition for variational optical flow computation. *IEEE Transactions on Image Processing*, to appear.

Kumar, A., Tannenbaum, A.R., and Balas, G.J. 1996. Optic flow: a curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):598–610.

Luettgen, M.R., Karl, W.C., and Willsky, A.S. 1994. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Transactions on Image Processing*, 3(1):41–64.

Mémin E. and Pérez P. 1998a. A multigrid approach for hierarchical motion estimation. In *Proc. 6th International Conference on Computer Vision*, pp. 933–938, Bombay, India.

Mémin E. and Pérez P. 1998b. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719.

Nagel H.-H. and Enkelmann, W. 1986. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593.

Ortega, J.M. and Rheinboldt, W.C. 2000. *Iterative Solution of Nonlinear Equations in Several Variables*, vol. 30 of *Classics in Applied Mathematics*. SIAM, Philadelphia.

Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. 2005. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, to appear.

Rudin, L.I., Osher, S., and Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268.

Schnörr C. 1994. Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Proc. Twelfth International Conference on Pattern Recognition*, vol. A, pp. 661–663, Jerusalem, Israel, IEEE Computer Society Press.

Terzopoulos, D. 1986. Image analysis using multigrid relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):129–139.

Tikhonov, A.N. and Arsenin, V.Y. 1977. *Solutions of Ill–Posed Problems*. Wiley, Washington, DC.

Trottenberg, U., Oosterlee, C., and Schüller A. 2001. *Multigrid*. Academic Press, San Diego.

Vogel, C.R. 1995. A multigrid method for total variation-based image denoising. *Computation and Control IV*, 20:323–331.

Vogel, C.R. 2002. *Computational Methods for Inverse Problems*. SIAM, Philadelphia.

Weickert, J. 1998. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart.

Weickert, J. and Schnörr C. 2001a. A theoretical framework for convex regularizers in PDE-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264.

Weickert, J. and Schnörr C. 2001b. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255.

Wesseling, P. 1992. *An Introduction to Multigrid Methods*. Wiley, Chichester.

Young, D.M. 1971. *Iterative Solution of Large Linear Systems*. Academic Press, New York.

Zini, G., Sarti, A., and Lamberti, C. 1997. Application of continuum theory and multi-grid methods to motion evaluation from 3D echocardiography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 44(2):297–308.

El Kalmoun M. and Rüde U. 2003. A variational multigrid for computing the optical flow. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, (eds.), *Vision, Modelling and Visualization*, pp. 577–584. IOS Press.