Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

Preprint Nr. 136

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

Andrés Bruhn, Joachim Weickert,
Timo Kohlberger and Christoph Schnörr

Saarbrücken 2005

Fachrichtung 6.1 – Mathematik
Universität des Saarlandes

Preprint No. 136
submitted: May 30, 2005

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

Andrés Bruhn
Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science
Saarland University, Building 27.1
66041 Saarbrücken, Germany
bruhn@mia.uni-saarland.de

Joachim Weickert
Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science
Saarland University, Building 27.1
66041 Saarbrücken, Germany
weickert@mia.uni-saarland.de

Timo Kohlberger
Computer Vision, Graphics and Pattern Recognition Group
Dept. of Mathematics and Computer Science
University of Mannheim
68131 Mannheim, Germany
kohlberger@uni-mannheim.de

Christoph Schnörr
Computer Vision, Graphics and Pattern Recognition Group
Dept. of Mathematics and Computer Science
University of Mannheim
68131 Mannheim, Germany
schnoerr@uni-mannheim.de

Edited by
FR 6.1 – Mathematik
Universität des Saarlandes
Postfach 15 11 50
66041 Saarbrücken
Germany

Fax: + 49 681 302 4443
e-Mail: preprint@math.uni-sb.de
WWW: http://www.math.uni-sb.de/

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, allow to deal with large displacements and perform well under noise or 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 systems of equations 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. In particular, discontinuity-preserving regularisation strategies are thereby in the focus of our work. (ii) We show by the examples of classical as well as more advanced variational techniques that real-time performance is possible – even for very complex optic flow models with high accuracy. Experiments show frame rates up to 63 dense flow fields per second for real-world image sequences of size 160 × 120 on a standard PC. Compared to classical iterative methods this constitutes a speedup of two to four orders of magnitude.

AMS 2000 Subject Classification: 68T45, 65N55, 49K20, 65K10, 35J60, 65N04
Key Words: computer vision, optic flow, differential techniques, variational methods, multigrid methods, partial differential equations.

Contents

1 Introduction

In computer vision the estimation of motion information from image sequences is one of the key problems. 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, and a smoothness term that regularises the often non-unique solution of the data term by an additional smoothness assumption.

Although recent developments [9, 12, 34] have shown that variational methods are among the best techniques for computing the optic flow in terms of error measures [4], they are often considered to be too slow for real-time applications. In particular the computational costs for solving the resulting linear and nonlinear system of equations by using standard iterative methods are regarded as too high. In [10] we have already demonstrated for variational methods with homogeneous regularisation that *bidirectional multigrid strategies* [7, 8, 26, 42, 48] do allow for real-time performance. 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 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 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 results. 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.

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 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 discretiation 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 extends work previously published at the 5th Conference on Scale-Space and PDE Methods in Computer Vision [11]. Substantial differences include, among other things, the considerations of all frequently used types of regularsation strategies (homogeneous, image- and flow-driven as well as isotropic and anisotropic), the extension to two more advanced variational optic flow techniques (Bruhn *et al.* [12] and Papenberg *et al.* [37]) and a much more extensive experimental evaluation.

Related Work. In the literature on variational optic flow methods, coarse-to-fine strategies are quite common to speed up the computation (see e.g. Anandan [3], Luettgen et al. [33]). They are based on a successive refinement of the problem whereby coarse grid solutions serve as initial guesses on finer grids. However, from a numerical viewpoint such *unidirectional* schemes are not the end of the road. They are clearly outperformed by *bidirectional* multigrid methods that revisit coarser levels in order to obtain useful correction steps. While there is some literature on these highly efficient schemes for variational optic flow techniques with homogeneous and image-driven regularisation (Glazer [25], Terzopoulos [40], Zini et a. [50], El Kalmoun and Rüde [18], Enkelmann [20], Ghosal and Vaněk [24]), only the work of Borzi et al. [6] is known to the authors where an optic flow problem was solved by means of a nonlinear bidirectional multigrid scheme (FAS). Also for other tasks in image processing and computer vision, multigrid methods have been used successfully. In the context of photometric stereo and image biniarisation Kimmel and Yavneh [29] developed an algebraic multigrid method, while Chan et al. [13] researched geometric multigrid schemes for variational deconvolution with TV regularisation. For TV denoising Vogel [43] proposed the use of a linear multigrid method within a nonlinear fixed-point iteration, while, very recently, Frohn-Schnauf et al. [23] investigated a nonlinear multigrid scheme (FAS) for the same task.

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 [37]. 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

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

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

$$f_x u + f_y v + f_t = 0. \tag{2}$$

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 notation and to allow for a better understanding of the proposed discretisation coarse grid approximation approach (DCA) [48] in Section 5, let us introduce the concept of *motion tensors* [21]. To this end, we reformulate equation 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 data term that based on a squared formulation of this equation as a quadratic form given by

$$E_D(u, v) = (f_x u + f_y v + f_t)^2 = \Big((u, v, 1)\ \nabla_3 f \nabla_3 f^\top\ (u, v, 1)^\top\Big) = \Big((u, v, 1) J\ (u, v, 1)^\top\Big) \tag{3}$$

where the motion tensor $J := \nabla_3 f \nabla_3 f^\top$ is a 3×3 matrix which is positive semidefinite by construction. One should note that such a reformulation by means of a quadratic form and a positive semidefinite motion tensor is possible for all constancy assumptions presented in [37]. However, in the case of the grey value constancy assumption, the obtained quadratic form is very special, since the associated motion tensor coincides with the well-known structure tensor [22].

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 [46], 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 [27], *image-driven* regularisation that assumes piecewise smoothness and respects discontinuities in the image [1, 35] and *flow-driven* regularisation that assumes piecewise smoothness and respects discontinuities in the flow field; see e.g. [16, 39, 46]. 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 presented in detail.

(a) Homogeneous Regularisation

Prototype for the class of methods with *homogeneous* regularisation is the classical method of Horn and Schunck [27]. Their method assumes global smoothness by penalising deviations from smoothness in a quadratic way [41]. The corresponding energy functional reads

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

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 down-weighting the smoothness term at locations where the magnitude of the spatial image gradient is large [1]. This form of regularisation that respects discontinuities in the image data is called *image-driven isotropic* regularisation. The associated energy functional is given by

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

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 the Charbonnier [15] function which reads

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

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 [35]. 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

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

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

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

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 where edges in the flow field occur during 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 [28]. The corresponding energy functional reads

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

where $\Psi_{\mathrm{s}}(s^2)$ is a positive increasing function in $\mathbb{R}$ with the aforementioned properties. As prototype we have chosen a method that penalises deviations from the smoothness with the L_1 norm of the flow gradient magnitude. This corresponds to total variation regularisation [38] which we implemented by means of a regularised variant given by

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

Here, ϵ_{S} serves as small regularisation parameter. A related functional that approximates TV regularisation is proposed in [47], while variational approaches for rotationally not invariant versions of TV regularisation have been investigated in [16, 17, 32].

(e) Flow-Driven Anisotropic Regularisation

The fifth and last regularisation strategy are *flow-driven anisotropic* smoothness terms [46]. 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 corresponds to its application to both eigenvalues of the tensor. This proceeding allows an anisotropic penalisation that adapts to the local flow structure. The associated energy functional is given by

$$E_{\mathrm{FA}}(u,v)=\int_\Omega \Big((u,v,1)^\top J\,(u,v,1)+\alpha\ \mathrm{tr}\ \Big(\Psi_{\mathrm{s}}(\,\nabla u\nabla u^\top+\nabla v\nabla v^\top\,)\Big)\Big)\,dxdy, \tag{11}$$

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 the total variation.

2.3 The Euler–Lagrange Equations

Following the calculus of variations [19], 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 [45]. 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

$$0 = J_{11}\,u + J_{12}\,v + J_{13} - \alpha\,\mathcal{L}_{\mathrm{L}}u, \tag{12}$$
$$0 = J_{12}\,u + J_{22}\,v + J_{23} - \alpha\,\mathcal{L}_{\mathrm{L}}v \tag{13}$$

with the *linear* differential operator

$$\mathcal{L}_{\mathrm{L}}z(x,y) = \mathrm{div}\,\left(D(|\nabla f|^2)\;z(x,y)\right) \tag{14}$$

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

(a)	Homogeneous Regularisation	$D(\|\nabla f\|^2) = I$
(b)	Image-Driven Isotropic Regularisation	$D(\|\nabla f\|^2) = w(\|\nabla f\|^2)\;I$
(c)	Image-Driven Anisotropic Regularisation	$D(\|\nabla f\|^2) = D_{\mathrm{NE}}(\|\nabla f\|^2)$

(d)-(e) The Nonlinear Case I

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

$$0 = J_{11}\,u + J_{12}\,v + J_{13} - \frac{\alpha}{2}\,\mathcal{L}_{\mathrm{NL}}(u,v), \tag{15}$$
$$0 = J_{12}\,u + J_{22}\,v + J_{23} - \frac{\alpha}{2}\,\mathcal{L}_{\mathrm{NL}}(v,u) \tag{16}$$

with the *nonlinear* differential operator

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

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

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

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.* [12] and the highly accurate optic flow method of Papenberg *et al.* [37]. 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.* [12] 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 [28] and increases the performance of the approach with respect to noise [5]. As prototype for this class of combined-local-global methods we have chosen a technique with total variation as non-quadratic penaliser in both the smoothness and the data term. The associated energy functional is given by

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

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

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

Apart from robustness under noise there are two further problems in the context of motion estimation: The correct estimation of large displacements and the robustness under varying illumination. In [9, 37] Papenberg *et al.* proposed a sophisticated variational approach that tackles both problems 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 $\nabla I = (I_x, I_y)^\top$. 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

$$E_{\mathrm{PAP}}(u,v) = \int_\Omega \Big(\Psi_{\mathrm{D}}\Big(|f(x+u,y+v,t+1) - f(x,y,t)|^2 + \gamma\, |\nabla f I(x+u,y+v,t+1) - \nabla f(x,y,t)|^2 \Big) + \alpha\, \Psi_{\mathrm{S}}\big(|\nabla u|^2 + |\nabla v|^2 \big) \Big)\, dx dy, \tag{20}$$

where once again the regularised total variation 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 our 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

$$0 = \Psi_{\mathrm{D}}'\Big((u,v,1)^\top J_\rho\, (u,v,1) \Big)\, (J_{\rho 11}\, u + J_{\rho 12}\, v + J_{\rho 13}) - \frac{\alpha}{2}\, \mathcal{L}_{\mathrm{NL}}(u,v), \tag{21}$$

$$0 = \Psi_{\mathrm{D}}'\Big((u,v,1)^\top J_\rho\, (u,v,1) \Big)\, (J_{\rho 12}\, u + J_{\rho 22}\, v + J_{\rho 23}) - \frac{\alpha}{2}\, \mathcal{L}_{\mathrm{NL}}(v,u). \tag{22}$$

Main differences to the flow-driven isotropic case in (d) are the modified motion tensor J_ρ and the additional factor $\Psi_{\mathrm{D}}'\left((u,v,1)^\top J_\rho\, (u,v,1)\right)$ in front of the data term 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

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

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

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 a *nonconvex* optimisation problem. Therefore we follow the idea from [9] and embed the solution of these equations in an incremental computation based on a coarse-to-fine fixed point iteration (warping). To this end, we split up the unknown flow field u^{k+1} and v^{k+1} at iteration $k{+}1$ of this fixed point scheme into the already known part u^k and v^k and the unknown motion increment du^k and dv^k. This allows to finally perform all linearisations of the constancy assumptions – this step has intentionally been postponed from the modelling) – via a Taylor expansions such as

$$
\begin{aligned}
& f(x{+}u^{k+1},y{+}v^{k+1},t{+}1) - f(x,y,t) \\
\approx\;& f(x{+}u^k,y{+}v^k,t{+}1) + f_x(x{+}u^k,y{+}v^k,t{+}1)\,du + f_y(x{+}u^k,y{+}v^k,t{+}1)\,dv - f(x,y,t) \\
=\;& \underbrace{f(x{+}u^k,y{+}v^k,t{+}1) - I(x,y,t)}_{\text{temporal difference !}} + f_x(x{+}u^k,y{+}v^k,t{+}1)\,du + f_y(x{+}u^k,y{+}v^k,t{+}1)\,dv.
\end{aligned}
\tag{25}
$$

As shown in [9] the incorporation of this partly linearised fixed point iteration into a coarse-to-fine multiresolution strategy leads to the well-known warping technique. Thereby the resolution ratio between two consecutive resolution levels is reduced by ν^2, where ν is the downsampling factor for each dimension. At resolution level k the following system of coupled PDEs has to be solved:

$$
\begin{aligned}
0 \;=\;& \Psi_{\mathrm{D}}'\left((u^k{+}du^k,v^k{+}dv^k,1)^\top S^k\,(u^k{+}du^k,v^k{+}dv^k,1)\right)\left(S^k_{11}\,du^k + S^k_{12}\,dv^k + S^k_{13}\right) \\
& - \frac{\alpha}{2}\,\mathcal{L}_{\mathrm{NL}}(u^k{+}du^k,v^k{+}dv^k),
\end{aligned}
\tag{26}
$$

$$
\begin{aligned}
0 \;=\;& \Psi_{\mathrm{D}}'\left((u^k{+}du^k,v^k{+}dv^k,1)^\top S^k\,(u^k{+}du^k,v^k{+}dv^k,1)\right)\left(S^k_{12}\,du^k + S^k_{22}\,dv^k + S^k_{23}\right) \\
& - \frac{\alpha}{2}\,\mathcal{L}_{\mathrm{NL}}(v^k{+}dv^k,u^k{+}du^k).
\end{aligned}
\tag{27}
$$

Here, once again the motion tensor notation has been used. The tensor S is thereby given as $S = \tilde{J}{+}\gamma\,\tilde{G}$ 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 $\tilde{G} = \tilde{\nabla} f_y \tilde{\nabla} f_x^\top + \tilde{\nabla} f_y \tilde{\nabla} f_y^\top$. 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 (cf. Eq. 25). 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).

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 $i = 1,...,N_x$ and $j = 1,...,N_y$. 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

$$D =: \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{28}$$

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

$$\partial_x \left(a(x,y)\, \partial_x z(x,y) \right) \approx D_x^{-,\mathbf{h}} \left(M_x^{+,\mathbf{h}}(a_{i,j})\, D_x^{+,\mathbf{h}}(z_{i,j}) \right), \tag{29}$$

$$\partial_x \left(b(x,y)\, \partial_y z(x,y) \right) \approx D_x^{\mathbf{h}} \left(b_{i,j}\; D_y^{\mathbf{h}}\; (z_{i,j}) \right), \tag{30}$$

$$\partial_y \left(b(x,y)\, \partial_x z(x,y) \right) \approx D_y^{\mathbf{h}} \left(b_{i,j}\; D_x^{\mathbf{h}}\; (z_{i,j}) \right), \tag{31}$$

$$\partial_y \left(c(x,y)\, \partial_y z(x,y) \right) \approx D_y^{-,\mathbf{h}} \left(M_y^{+,\mathbf{h}}(c_{i,j})\, D_y^{+,\mathbf{h}}(z_{i,j}) \right). \tag{32}$$

Details on the required averaging and differential operators within the approximations are given in Table 1.

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

$$0 = J^{\mathbf{h}}_{11,i,j}\, u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{12,i,j}\, v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{13,i,j} - \alpha\, L^{\mathbf{h}}_{\mathrm{L\,i,j}}\, u^{\mathbf{h}}_{i,j}, \tag{33}$$

$$0 = J^{\mathbf{h}}_{12,i,j}\, u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{22,i,j}\, v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{23,i,j} - \alpha\, L^{\mathbf{h}}_{\mathrm{L\,i,j}}\, v^{\mathbf{h}}_{i,j}, \tag{34}$$

for $i = 1, .., N_x$ and $j = 1, .., N_y$, where $L^{\mathbf{h}}_{\mathrm{NE\,i,j}}$ 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).

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	$\left\|D^{2,\mathbf{h}}(z_{i,j})\right\| := \sqrt{D_x^{2,\mathbf{h}}(z_{i,j}) + D_y^{2,\mathbf{h}}(z_{i,j})}$	(43)

(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

$$0 = J^{\mathbf{h}}_{11,i,j}\, u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{12,i,j}\, v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{13,i,j} - \frac{\alpha}{2}\, L^{\mathbf{h}}_{\mathrm{NL\ i,j}}(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j})\, u^{\mathbf{h}}_{i,j}, \tag{44}$$

$$0 = J^{\mathbf{h}}_{12,i,j}\, u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{22,i,j}\, v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{23,i,j} - \frac{\alpha}{2}\, L^{\mathbf{h}}_{\mathrm{NL\ i,j}}(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j})\, v^{\mathbf{h}}_{i,j}, \tag{45}$$

for $i = 1, .., N_x$ and $j = 1, .., N_y$. 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

$$\begin{aligned} 0 = {} & \Psi'_{\mathrm{D}}\left((u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)^\top J^{\mathbf{h}}_{\rho}\,(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)\right)\left(J^{\mathbf{h}}_{11,i,j}\, u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{12,i,j}\, v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{13,i,j}\right) \\ & - \frac{\alpha}{2}\, L^{\mathbf{h}}_{\mathrm{NL\ i,j}}(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j})\, u^{\mathbf{h}}_{i,j}, \end{aligned} \tag{46}$$

$$\begin{aligned} 0 = {} & \Psi'_{\mathrm{D}}\left((u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)^\top J^{\mathbf{h}}_{\rho}\,(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)\right)\left(J^{\mathbf{h}}_{12,i,j}\, u^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{22,i,j}\, v^{\mathbf{h}}_{i,j} + J^{\mathbf{h}}_{23,i,j}\right) \\ & - \frac{\alpha}{2}\, L^{\mathbf{h}}_{\mathrm{NL\ i,j}}(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j})\, v^{\mathbf{h}}_{i,j}, \end{aligned} \tag{47}$$

for $i = 1, .., N_x$ and $j = 1, .., N_y$. Here, 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 $\Psi'_{\mathrm{D}}\left((u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)^\top J^{\mathbf{h}}_\rho\,(u^{\mathbf{h}}_{i,j}, v^{\mathbf{h}}_{i,j}, 1)\right)$ 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 a hierarchy of *nonlinear* equation system. For warping level k the discretised Euler–Lagrange equations are given by

$$\begin{aligned}
0 &= \Psi'_{\mathrm{D}}\left((u^{\mathbf{k},\mathbf{h}}_{i,j} + du^{\mathbf{k},\mathbf{h}}_{i,j}, v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}, 1)^\top S^{\mathbf{k},\mathbf{h}}_{i,j}\,(u^{\mathbf{k},\mathbf{h}}_{i,j} + du^{\mathbf{k},\mathbf{h}}_{i,j}, v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}, 1)\right) \\
&\quad \left(S^{\mathbf{k},\mathbf{h}}_{11,i,j}\,du^{\mathbf{k},\mathbf{h}}_{i,j} + S^{\mathbf{k},\mathbf{h}}_{12,i,j}\,dv^{\mathbf{k},\mathbf{h}}_{i,j} + S^{\mathbf{k},\mathbf{h}}_{13,i,j}\right) \\
&\quad - \frac{\alpha}{2}\,L^{\mathbf{k},\mathbf{h}}_{\mathrm{NL\,i,j}}(u^{\mathbf{k},\mathbf{h}}_{i,j} + u^{\mathbf{k},\mathbf{h}}_{i,j}, v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j})\,\left(u^{\mathbf{k},\mathbf{h}}_{i,j} + du^{\mathbf{k},\mathbf{h}}_{i,j}\right), \qquad (48) \\
0 &= \Psi'_{\mathrm{D}}\left((u^{\mathbf{k},\mathbf{h}}_{i,j} + du^{\mathbf{k},\mathbf{h}}_{i,j}, v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}, 1)^\top S^{\mathbf{k},\mathbf{h}}_{i,j}\,(u^{\mathbf{k},\mathbf{h}}_{i,j} + du^{\mathbf{k},\mathbf{h}}_{i,j}, v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}, 1)\right) \\
&\quad \left(S^{\mathbf{k},\mathbf{h}}_{12,i,j}\,du^{\mathbf{k},\mathbf{h}}_{i,j} + S^{\mathbf{k},\mathbf{h}}_{22,i,j}\,dv^{\mathbf{k},\mathbf{h}}_{i,j} + S^{\mathbf{k},\mathbf{h}}_{23,i,j}\right) \\
&\quad - \frac{\alpha}{2}\,L^{\mathbf{k},\mathbf{h}}_{\mathrm{NL\,i,j}}(v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}, u^{\mathbf{k},\mathbf{h}}_{i,j} + u^{\mathbf{k},\mathbf{h}}_{i,j})\,\left(v^{\mathbf{k},\mathbf{h}}_{i,j} + dv^{\mathbf{k},\mathbf{h}}_{i,j}\right), \qquad (49)
\end{aligned}$$

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.

5 Multigrid

5.1 Basic Concept

In general, the preceeding 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 [36, 49]. 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 undampened. In order to overcome this problem multigrid methods [7, 8, 26, 42, 48] 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

$$A^{\mathbf{h}}x^{\mathbf{h}} = f^{\mathbf{h}}. \qquad (50)$$

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

$$e^{\mathbf{h}} = x^{\mathbf{h}} - \tilde{x}^{\mathbf{h}}. \tag{51}$$

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

$$A^{\mathbf{h}}e^{\mathbf{h}} = A^{\mathbf{h}}(x^{\mathbf{h}} - \tilde{x}^{\mathbf{h}}) = A^{\mathbf{h}}x^{\mathbf{h}} - A^{\mathbf{h}}\tilde{x}^{\mathbf{h}} = f^{\mathbf{h}} - A^{\mathbf{h}}\tilde{x}^{\mathbf{h}} = r^{\mathbf{h}}, \tag{52}$$

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 $\mathbf{H} = (H_x, H_y)^\top$:

$$A^{\mathbf{h}}e^{\mathbf{h}} = r^{\mathbf{h}} \quad \rightarrow \quad A^{\mathbf{H}}e^{\mathbf{H}} = r^{\mathbf{H}}. \tag{53}$$

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

$$\tilde{x}^{\mathbf{h}}_{\text{new}} = \tilde{x}^{\mathbf{h}} + e^{\mathbf{h}}. \tag{54}$$

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

$$A^{\mathbf{h}}(x^{\mathbf{h}}) = f^{\mathbf{h}} \tag{55}$$

where $A^{\mathbf{h}}(x^{\mathbf{h}})$ is a nonlinear operator. The FAS strategy [7] 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

$$A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}} + e^{\mathbf{h}}) - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = f^{\mathbf{h}} - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = r^{\mathbf{h}}. \tag{56}$$

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

$$A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}} + e^{\mathbf{h}}) = r^{\mathbf{h}} + A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) \quad \rightarrow \quad A^{\mathbf{H}}(\boxed{\tilde{x}^{\mathbf{H}}} + e^{\mathbf{H}}) = r^{\mathbf{H}} + \boxed{A^{\mathbf{H}}(\tilde{x}^{\mathbf{H}})}. \tag{57}$$

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. Refining the original problem step by step (unidirectional coarse-to-fine approach) and solving the resulting linear or nonlinear equation system at each level by using some bidirectional V– or W–cycles, the multigrid strategy with the best performance is obtained: *full multigrid* [8]. An overview of all three types of multigrid methods is given in Figure 1.

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) [10] 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) [48]. 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 direction were considered: Lines in x–and y–direction as well as the direction of different unknowns at each point, namely (u, v) itself. For the nonlinear variants of the Gauß–Seidel solver we used the strategy of frozen coefficients [23]. In the literature this technique is also known as lagged diffusivity method [14] or Quasi-Newton scheme [44]. Direct nonlinear Gauß–Seidel Newton methods [8] 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 times.
One can also see 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 should note the increasing number of pre- and

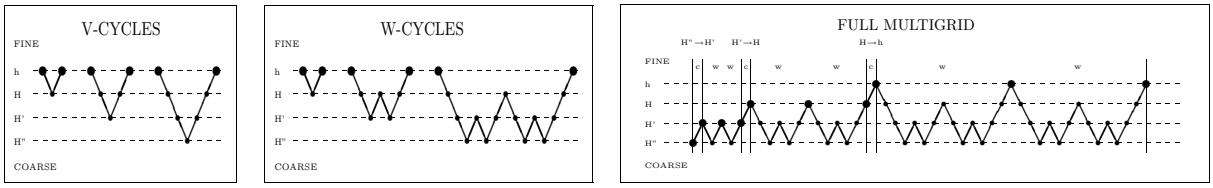

Figure 1: *Left:* Example for V-ycles with two, three and four levels. *Center:* Ditto for W-cycles. *Right:* Full multigrid implementation with 2 W-cycles per resolution level. Refinement steps are marked with 'c'. Each W-cycle is marked with a 'w'.

Table 2: Implemented multigrid schemes the different variational models. 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.

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

postsmoothing relaxation iterations. This is due to 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 [10] they were realised by constant interpolation and simple averaging. Coarser versions of the linear and nonlinear operators were created by a discretisation coarse grid approximation (DCA) [48]. To this end, the entries of the motion tensor were restricted *channelwise*:

$$J_{nm}^{\mathbf{h}} \longrightarrow J_{nm}^{\mathbf{H}} \qquad n,m = 1,..,3. \tag{58}$$

One should note that our choice for the restriction operator (simple averaging) maintains their positive semidefiniteness. In the linear cases, also the entries of the coarser diffusion tensors were created in this manner.

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 a modified explicit scheme [47]. 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 multigrid methods 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.

Table 3: Performance benchmark for all five 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.

(a) Linear : Homogeneous regularisation (Horn and Schunck)
$\sigma = 1.0$, $\alpha = 1000$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.25$)	4425	3.509	0.285	1
Gauß-Seidel (CPR)	2193	1.152	0.868	3
Full Multigrid	**1**	**0.016**	**62.790**	**220**

(b) Linear : Image-driven isotropic regularisation (Charbonnier)
$\sigma = 1.0$, $\alpha = 1000$, $\epsilon_S = 1.0$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.25$)	8894	12.048	0.083	1
Gauß-Seidel (CPR)	2856	2.793	0.358	4
Full Multigrid	**1**	**0.048**	**20.850**	**251**

(c) Linear : Image-driven anisotropic regularisation (Nagel-Enkelmann)
$\sigma = 1.0$, $\alpha = 1000$, $\epsilon_S = 10^{-2}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.1666$)	36558	47.053	0.021	1
Gauß-Seidel (ALR)	607	3.608	0.277	13
Full Multigrid	**1**	**0.171**	**5.882**	**275**

(d) Nonlinear : Flow-driven isotropic regularisation (TV)
$\sigma = 1.0$, $\alpha = 10$, $\epsilon_S = 10^{-2}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.0025$)	10631	30.492	0.033	1
Gauß-Seidel (CPR)	2679	6.911	0.145	4
FAS - Full Multigrid	**1**	**0.082**	**12.172**	**372**

(e) Nonlinear : Flow-driven anisotropic regularisation (TV)
$\sigma = 1.0$, $\alpha = 10$, $\epsilon_S = 10^{-2}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.0025$)	9208	58.824	0.017	1
Gauß-Seidel (ALR)	591	12.508	0.080	5
FAS - Full Multigrid	**1**	**0.491**	**2.038**	**120**

Table 4: Performance benchmark for the more advanced optic flow methods: The nonlinear variant of the CLG method by Bruhn *et al.* [12] and the method of Papenberg *et al.* [37] (see also Brox *et al.* [9]). 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.

(f) Bruhn *et al.* : Robust data term with local integration + flow-driven isotropic regularisation (TV)
$\sigma = 0.0$, $\rho = 15$, $\alpha = 15$, $\epsilon_{\mathrm{D}} = 10^{-1}$, $\epsilon_{\mathrm{S}} = 10^{-3}$

Solver	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.00025$)	81947	244.798	0.004	1
Gauß-Seidel (CPR)	3720	9.524	0.105	26
FAS - Full Multigrid	1	0.087	11.473	2809

(g) Papenberg *et al.* : Robust nonlinearised data term with additional gradient constancy + flow-driven isotropic regularisation (TV)
$\sigma = 1.0$, $\alpha = 121$, $\gamma = 230$, $\nu = 0.65$, $\epsilon_{\mathrm{D}} = 10^{-1}$, $\epsilon_{\mathrm{S}} = 10^{-3}$

Solver for each **warping** level	Iterations	Time $[s]$	FPS $[s^{-1}]$	Speedup
Mod. Explicit Scheme ($\tau = 0.00025$)	79112	445.831	0.002	1
Gauß-Seidel (CPR)	6549	34.483	0.029	13
FAS - Full Multigrid	1	0.396	2.527	1127

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. 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*. 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 other standard 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 [10] 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 Figure 2 and Figure 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 gives slightly better results

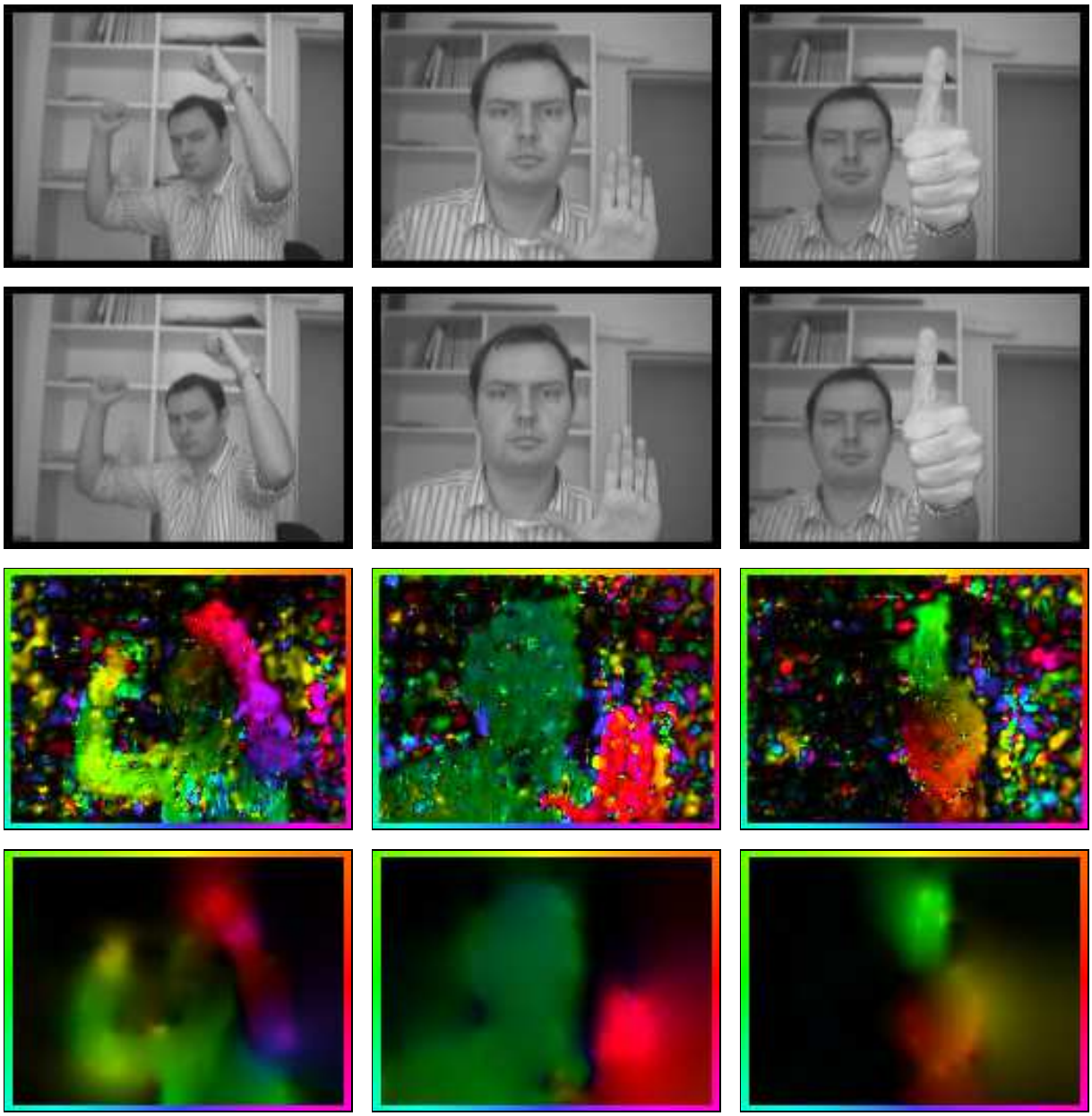

Figure 2: *Left to right:* dancing sequence, waving sequence, rotating thumb sequence. *Top to bottom:* first frame, second frame, no regularisation (normal flow), homogeneous regularisation (Horn and Schunck). *Colour code:* The direction of a flow vector is represented by colour as shown on flow field boundaries (e.g. green corresponds to a motion to the left). The magnitude of a flow vector is encoded by its brightness.

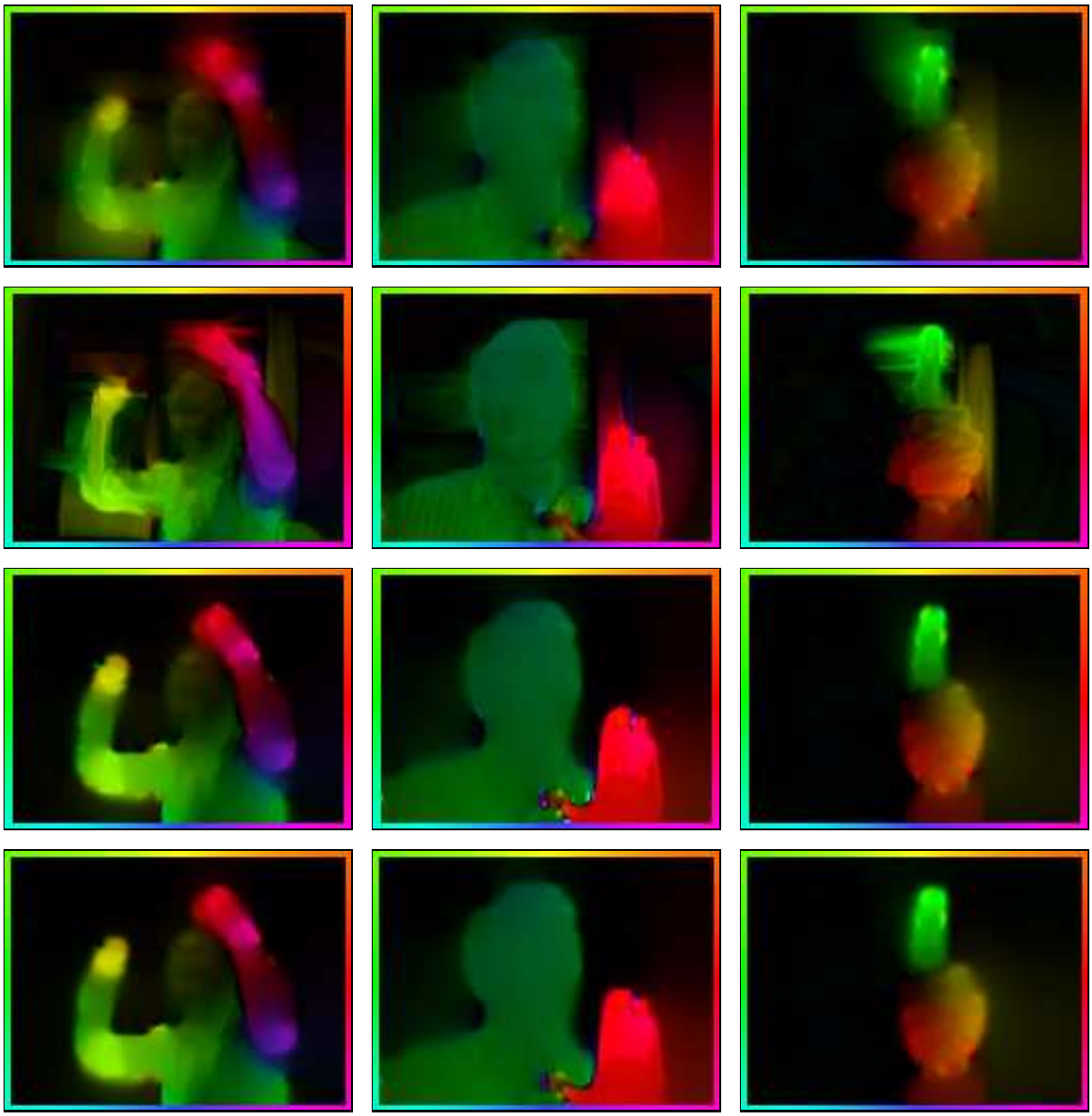

Figure 3: *Left to right:* dancing sequence, waving sequence, rotating thumb sequence. *Top to bottom:* data-driven isotropic regularisation (Charbonnier), data-driven anisotropic regularisation (Nagel-Enkelmann), flow-driven isotropic regularisation (TV), flow-driven anisotropic regularisation (TV). *Colour code:* The direction of a flow vector is represented by colour as shown on flow field boundaries (e.g. green corresponds to a motion to the left). The magnitude of a flow vector is encoded by its brightness.

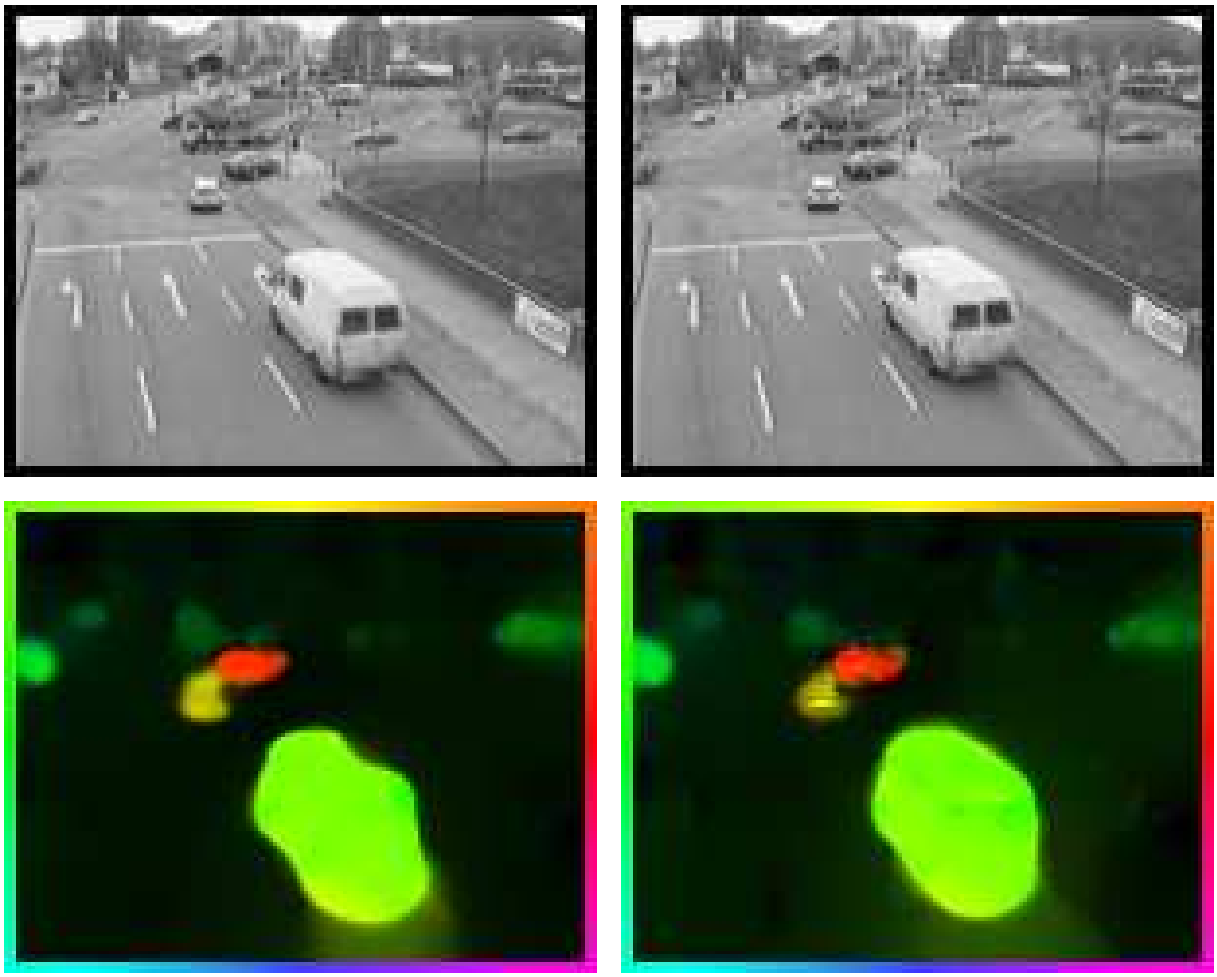

Figure 4: *(a) Top Left:* Frame 1130 of the *Rheinhafen* sequence by Nagel (resized to 160 × 120). *(b) Top Right:* Frame 1131. *(c) Bottom Left:* Computed flow field by the multigrid implementation of the 2-D method of Bruhn *et al.*. Computing time: 87 milliseconds. *(d) Bottom Right:* Computed flow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

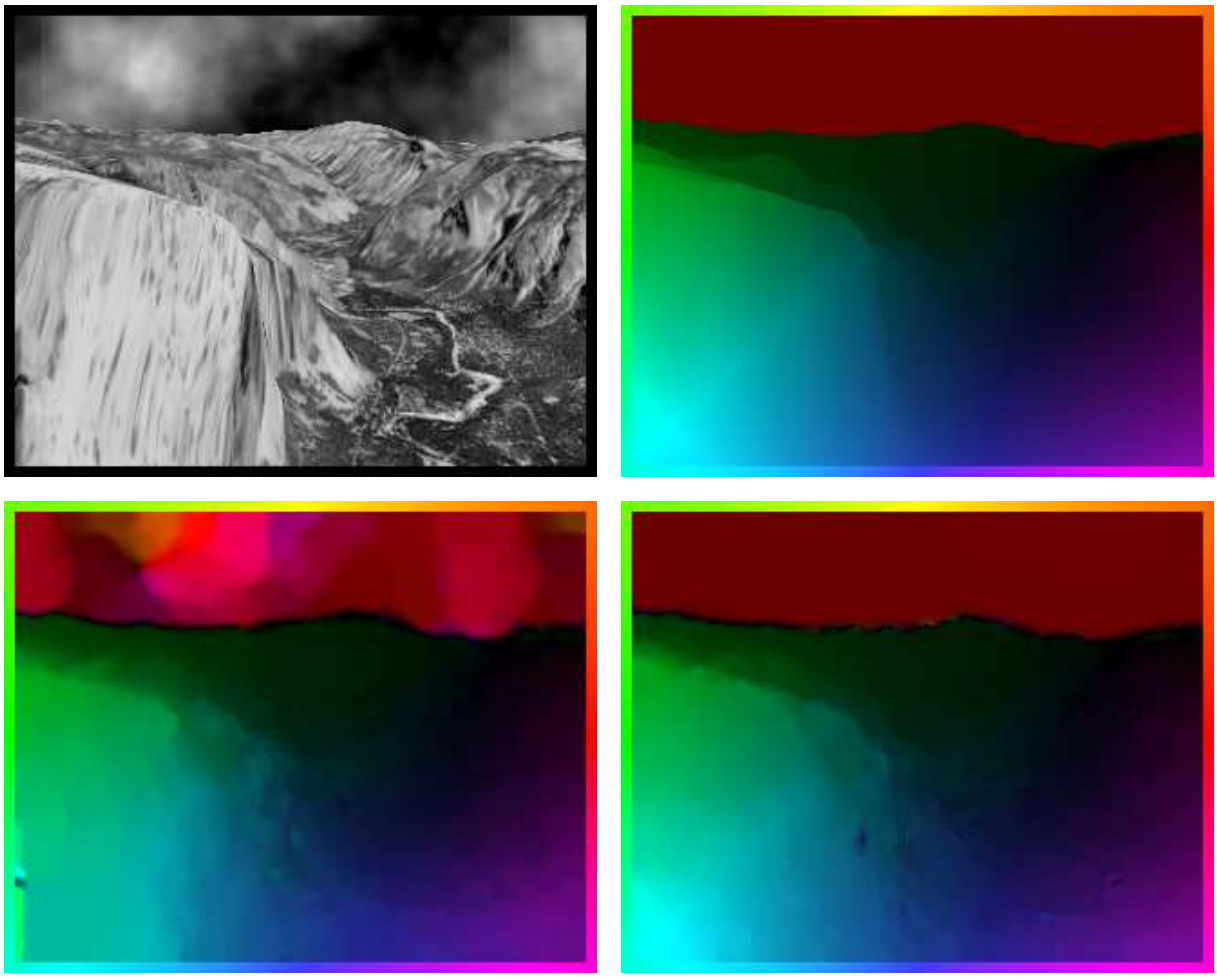

Figure 5: *(a) Top Left:* Frame 8 of the *Yosemite* sequence with clouds by Lynn Quam (size 312×256). *(b) Top Right:* Ground truth flow field. *(c) Bottom Left:* Computed flow field by our CLG 2-D multigrid implementation. Average angular error: 5.77°. Computing time: 384 milliseconds. *(d) Bottom Right:* 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 [4]	13.36°	15.64°
Nagel [4]	10.22°	16.51°
Horn–Schunck, mod. [4]	9.78°	16.19°
Uras *et al.* [4]	8.94°	15.61°
Bruhn *et al.* nonlinear (2D) [12]	6.03°	8.61°
Bruhn *et al.* nonlinear – Multigrid (2D)	**5.77°**	**8.47°**
Alvarez *et al.* [2]	5.53°	7.40°
Mémin–Pérez [34]	4.69°	6.89°
Papenberg *et al.* – Multigrid (2D)	**2.51°**	**6.58°**
Papenberg *et al.* (2D) [37]	2.44°	6.90°
Papenberg *et al.* (3D) [37]	1.78°	7.00°

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.* [12] and the highly accurate optic flow technique of Papenberg *et. al* [37]. 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 relate only 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 to all previously discussed methods in this paper including the technique of Bruhn *et al.*. In particular, this means that warping errors on coarser levels influence the result on finer levels such that in this case errors can propagate.

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. And even compared to the basic iterative solvers an excellent speedup of *two orders of magnitude* is obtained. The corresponding frame rates show clearly that in the case of such highly advanced optic flow methods, real-time performance is still possible.

In Figure 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 tresholding. This segmentation-like behaviour, that is desired in many optic flow applications, is only a direct

consequence of using TV as discontinuity-preserving regulariser.
In our final experiment we evaluate the accuracy of our real-time implementations for the two previously discussed more advanced 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 [4] 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 maintain the quality of their original methods and are capable of giving very accurate 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, in the case of the method of Papenberg *al.* the relatively small coarsening factor of $\nu = 0.65$ limited the average angular error to $2.52°$. A larger coarsening factor ν close to 1 would allow of coarse 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).
The flow fields computed by both approaches are illustrated in Figure 5. Apart from these flow fields also the ground truth solution is shown 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 more advanced variational optic flow methods. Experiments demonstrated that compared to classical iterative solvers speedups of two to four orders magnitude can be achieved. This clearly 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 parallelisation strategies is ongoing work [30, 31]. Their usage allows the processing of even higher resolution video sequences in real-time.

Acknowledgements

Our optic flow research is partly funded by the *Deutsche Forschungsgemeinschaft (DFG)* under the project WE 2602/3-1. This is gratefully acknowledged.

References

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

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

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

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

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

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

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

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

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

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

[11] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. Discontinuity-preserving computation of variational optic flow in real-time. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, Lecture Notes in Computer Science. Springer, Berlin, 2005.

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

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

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

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

[16] I. Cohen. Nonlinear variational method for optical flow computation. In *Proc. Eighth Scandinavian Conference on Image Analysis*, volume 1, pages 523–530, Tromsø, Norway, May 1993.

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

[18] M. El Kalmoun and U. Rüde. 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, editors, *Vision, Modelling and Visualization*, pages 577–584. IOS Press, 2003.

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

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

[21] G. Farnebäck. 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*, volume 1, pages 171–177, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[22] W. Förstner and E. Gülch. 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*, pages 281–305, Interlaken, Switzerland, June 1987.

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

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

[25] F. Glazer. Multilevel relaxation in low-level computer vision. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 312–330. Springer, Berlin, 1984.

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

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

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

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

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

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

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

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

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

[35] H.-H. Nagel and W. Enkelmann. 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, 1986.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[50] G. Zini, A. Sarti, and C. Lamberti. 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, Mar. 1997.