Towards Ultimate Motion Estimation: Combining Highest Accuracy with Real-Time Performance

Andrés Bruhn and Joachim Weickert
Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science
Saarland University, 66041 Saarbrücken, Germany
{bruhn,weickert}@mia.uni-saarland.de

Abstract

Although variational methods are among the most accurate techniques for estimating the optical flow, they have not yet entered the field of real-time vision. Main reason is the great popularity of standard numerical schemes that are easy to implement, however, at the expense of being too slow for real-time performance. In our paper we address this problem in two ways: (i) We present an improved version of the highly accurate technique of Brox et al. [9]. Thereby we show that a separate robustification of the constancy assumptions is very useful, in particular if the 1-norm is used as penalizer. As a result, a method is obtained that yields the lowest angular errors in the literature. (ii) We develop an efficient numerical scheme for the proposed approach that allows real-time performance for sequences of size 160 × 120. To this end, we combine two hierarchical strategies: A coarse-to-fine warping strategy as implementation of a fixed point iteration for a non-convex optimisation problem and a nonlinear full multigrid method – a so called full approximation scheme (FAS) – for solving the highly nonlinear equation systems at each warping level. In the experimental section the advantage of the proposed approach becomes obvious: Outperforming standard numerical schemes by two orders of magnitude frame rates of six high quality flow fields per second are obtained on a 3.06 GHz Pentium4 PC.

1 Introduction

Allowing a mathematically sound integration of different concepts into a single minimisation framework, variational methods belong to the best performing and best understood methods for computing the optical flow. In particular, their ability to estimate optical flow at those locations where image information is not available makes them very attractive. They can be designed in such a way that they preserve motion boundaries [9, 22, 23, 25, 29], treat large displacements correctly [1, 2, 4, 9, 22], are robust with respect to illumination changes [9] or perform favorably in the presence of noise and occlusions [4, 5, 9]. But only if several of these concepts are combined within a single energy functional, methods are obtained that allow for a highly accurate and dense estimation of the results and thus show the true potential of variational techniques [5, 9, 22].

However, there is a also a prize to pay when using variational methods: The regularizer, that yields the desired filling-in-effect at locations without image information, also leads to a coupling of neighboring pixels in the optimization problem. This in turn requires a joint optimization of all pixels by solving one or more large systems of equations. Often standard iterative numerical schemes such as the Gauß-Seidel method are applied for this purpose: While they are easy to implement, they require thousands of iterations to reach the desired numerical accuracy. Unfortunately, people inferred from this fact that variational methods are too slow for real-time performance.

From a numerical viewpoint, however, such basic iterative solvers are not the end of the road: Very promising, in particular in the context of variational methods, are so called bidirectional multigrid schemes [7, 8, 20, 27, 30]. They overcome limitations of iterative solvers by creating a sophisticated hierarchy of equation system with excellent error reduction properties. During the last years such methods became more and more popular and have been used for various tasks such as photometric stereo [21], variational deconvolution [13] or image restoration [17]. Also in the field of optical flow estimation such methods have been applied more frequently [6, 19, 16, 18, 26]. Recently, even first real-time implementations have been presented, that, however, were limited to basic variational methods only [10, 11].

In our paper we combine both the excellent quality of recent variational methods and the speed of numerical multigrid strategies. To this end, we present an improved version of the approach of Brox *et al.* [9] and develop a highly efficient numerical scheme for its optimization. While the model allows the computation of the lowest angular errors in the literature, the implementation allows the estimation of more than six of these high quality flow field per second on standard PCs if sequences of size 160 × 120 are used.

Our paper is organized as follows. In Section 2 we give a short review on the method of Brox *et al.* and improve it by a separate robustification of the data terms. A suitable optimization strategy for the obtained energy functional is presented in Section 3. Section 4 describes discretization aspects, the choice of a basic iterative solver and the development of a highly efficient multigrid method. Qualitative experiments and performance benchmarks are presented in Section 5 while a summary concludes this paper.

2 The Variation Model

In this section we derive our variational model for the real-time approach. Since its formulation is essentially based on the method of Brox *et al.* [9] let us start be giving a short review on this technique.

2.1 The Approach of Brox et al.

Let $I(\mathbf{x})$ be a presmoothed image sequence with $\mathbf{x} := (x, y, t)^\top$ where (x, y) denotes the location within a rectangular domain Ω and $t \geq 0$ denotes time. Furthermore let $\mathbf{u} = (u, v, 1)^\top$ be the wanted displacement field between two frames of that sequence at time t and time $t + 1$. Then, the 2-D variant of the algorithm of Brox *et al.* computes the optical flow as minimizer of the energy functional

$$E(\mathbf{u}) = E_D(\mathbf{u}) + \beta\, E_S(\mathbf{u}) \tag{1}$$

where the data term is given by

$$E_D(\mathbf{u}) = \int_\Omega \psi_D\Big(\begin{aligned} &|I(\mathbf{x}+\mathbf{u}) - I(\mathbf{x})|^2 \\ +\alpha\, &|\nabla I(\mathbf{x}+\mathbf{u}) - \nabla I(\mathbf{x})|^2\Big)\, d\mathbf{x} \end{aligned} \tag{2}$$

and the smoothness term reads

$$E_S(\mathbf{u}) = \int_\Omega \psi_S\left(|\nabla u|^2 + |\nabla v|^2\right)\, d\mathbf{x}. \tag{3}$$

Here α and β are non-negative weights. While the first part of the data term models the assumption that the grey value of objects does not change over time, the second one renders the approach more robust against varying illumination. This is achieved by assuming constancy of the spatial image gradient $\nabla I = (I_x, I_y)^\top$. In order to allow for a correct estimation of large displacements, the linearization of both assumptions is postponed to the optimization and the numerical scheme. Moreover, a non-quadratic penalizer ψ is applied to both the data and the smoothness term. While such a robust function in the data term increases the performance with respect to outliers, a non-quadratic penalizer in the smoothness term models the assumption of a piecewise smooth flow field. In both cases a regularized version of the total variation (TV) [24, 15] is used that is given by $\psi(s^2) = \sqrt{s^2 + \epsilon^2}$. The regularization parameter ϵ is thereby set to 10^{-3}.

2.2 Separate Robustification

Instead of applying a single robust function to the whole data term we propose a splitting such that deviations from the grey value and the gradient constancy assumption are penalized separately. Thus, the original energy functional in (1) becomes

$$E(\mathbf{u}) = E_{D_1}(\mathbf{u}) + \alpha\, E_{D_2}(\mathbf{u}) + \beta\, E_S(\mathbf{u}) \tag{4}$$

with the new data terms

$$E_{D_1}(\mathbf{u}) = \int_\Omega \psi_D\left(|I(\mathbf{x}+\mathbf{u}) - I(\mathbf{x})|^2\right)\, d\mathbf{x}, \tag{5}$$

$$E_{D_2}(\mathbf{u}) = \int_\Omega \psi_D\left(|\nabla I(\mathbf{x}+\mathbf{u}) - \nabla I(\mathbf{x})|^2\right)\, d\mathbf{x}. \tag{6}$$

This modification has the following advantage when using the total variation as robust function: Let s_1 and s_2 be two constraints. Then, instead of $\sqrt{s_1^2 + \alpha s_2^2}$ we obtain $|s_1| + \alpha_*|s_2| = \sqrt{s_1^2 + 2\alpha_*|s_1||s_2| + \alpha_*^2\, s_2^2}$. Setting $\alpha_* = \sqrt{\alpha}$ one can easily verify that the proposed variant does only penalize such cases more severely where both assumptions are not fulfilled, i.e. $|s_1| \neq 0$ and $|s_2| \neq 0$. As a consequence, those solutions are favored by the new approach where at least one of the data terms holds.

2.3 The Euler-Lagrange Equations

Let us now derive the Euler-Lagrange equations that are necessary conditions for the minimizer of the proposed energy functional. Following the calculus of variations and using abbreviations of type $I_* := \partial_* I\,(\mathbf{x}+\mathbf{u})$ for spatial derivatives and $I_{*z} := I_*\,(\mathbf{x}+\mathbf{u}) - I_*\,(\mathbf{x})$ for temporal differences, we obtain

$$\begin{aligned} &\Psi'_{D_1}\left(I_z^2\right) \cdot I_z I_x \\ +\quad &\alpha\, \Psi'_{D_2}\left(I_{xz}^2 + I_{yz}^2\right) \cdot \left(I_{xz} I_{xx} + I_{yz} I_{yx}\right) \\ -\quad &\beta\, \text{div}\left(\Psi'_S\left(\|\nabla u\|^2 + \|\nabla v\|^2\right) \nabla u\right) = 0, \end{aligned} \tag{7}$$

$$\begin{aligned} &\Psi'_{D_1}\left(I_z^2\right) \cdot I_z I_y \\ +\quad &\alpha\, \Psi'_{D_2}\left(I_{xz}^2 + I_{yz}^2\right) \cdot \left(I_{xz} I_{xy} + I_{yz} I_{yy}\right) \\ -\quad &\beta\, \text{div}\left(\Psi'_S\left(\|\nabla u\|^2 + \|\nabla v\|^2\right) \nabla v\right) = 0 \end{aligned} \tag{8}$$

with reflecting Neumann boundary conditions.

3 The Optimization

In this section, we discuss a suitable optimization procedure for our energy functional. In accordance with [9] we propose a fixed point iteration of the form

$$\begin{aligned}
&\Psi'_{D_1}\left(\left(I_z^2\right)^{k+1}\right)\cdot I_z^{k+1} I_x^k && (9)\\
+&\ \alpha\,\Psi'_{D_2}\left(\left(I_{xz}^2\right)^{k+1}+\left(I_{yz}^2\right)^{k+1}\right)\cdot\left(I_{xz}^{k+1}I_{xx}^k+I_{yz}^{k+1}I_{yx}^k\right)\\
-&\ \beta\,\mathrm{div}\left(\Psi'_S\left(\|\nabla u^{k+1}\|^2+\|\nabla v^{k+1}\|^2\right)\nabla u^{k+1}\right)=0,\\
&\Psi'_{D_1}\left(\left(I_z^2\right)^{k+1}\right)\cdot I_z^{k+1} I_y^k && (10)\\
+&\ \alpha\,\Psi'_{D_2}\left(\left(I_{xz}^2\right)^{k+1}+\left(I_{yz}^2\right)^{k+1}\right)\cdot\left(I_{xz}^{k+1}I_{xy}^k+I_{yz}^{k+1}I_{yy}^k\right)\\
-&\ \beta\,\mathrm{div}\left(\Psi'_S\left(\|\nabla u^{k+1}\|^2+\|\nabla v^{k+1}\|^2\right)\nabla v^{k+1}\right)=0
\end{aligned}$$

which is semi-implicit in the data and fully implicit in the smoothness term. Compared to an explicit scheme this offers a faster convergence and a better stability.

3.1 Incremental Computation

In a next step we perform those linearization that have been intentionally postponed from the modeling phase to the optimization. To this end, we split the unknown iteration variable $\mathbf{u}^{k+1}$ into the known variable $\mathbf{u}^k$ and an unknown increment $\mathbf{du}^k=\left(du^k,dv^k,0\right)^\top$. This allows us to perform a Taylor expansion and linearize the temporal differences $I_{*z}^{k=1}$ via

$$\begin{aligned}
I_{*z}^{k+1} &= I_*\left(\mathbf{x}+\mathbf{u}^{k+1}\right)-I_*\left(\mathbf{x}\right)\\
&\approx I_*\left(\mathbf{x}+\mathbf{u}^{k}\right)+I_{*x}^k du^k+I_{*y}^k dv^k-I_*\left(\mathbf{x}\right)\\
&= I_{*x}^k du^k+I_{*y}^k dv^k+I_{*z}^k.
\end{aligned}$$

Moreover, we simplify our notation by introducing the vector $\mathbf{I}_\nabla:=(I_x,I_y,I_z)^\top$ and defining the symmetric tensors $S:=\mathbf{I}_\nabla{\mathbf{I}_\nabla}^\top$ and $T:=\mathbf{I}_{\nabla\mathbf{x}}{\mathbf{I}_{\nabla\mathbf{x}}}^\top+\mathbf{I}_{\nabla\mathbf{y}}{\mathbf{I}_{\nabla\mathbf{y}}}^\top$ that are positive semi-definite by construction. Then, the partly linearized fixed point iteration can be rewritten as

$$\begin{aligned}
&\Psi'^k_{D_1}\cdot\left(S_{11}^k du^k+S_{12}^k dv^k+S_{13}^k\right) && (11)\\
+&\ \alpha\Psi'^k_{D_2}\cdot\left(T_{11}^k du^k+T_{12}^k dv^k+T_{13}^k\right)\\
-&\ \beta\ \mathrm{div}\left(\Psi'^k_S\nabla\left(u^k+dv^k\right)\right)=0,\\
&\Psi'^k_{D_1}\cdot\left(S_{12}^k du^k+S_{22}^k dv^k+S_{23}^k\right) && (12)\\
&\ \alpha\Psi'^k_{D_2}\cdot\left(T_{12}^k du^k+T_{22}^k dv^k+S_{23}^k\right)\\
-&\ \beta\ \mathrm{div}\left(\Psi'^k_S\nabla\left(u^k+dv^k\right)\right)=0
\end{aligned}$$

with the nonlinear abbreviations

$$\begin{aligned}
\Psi'^k_{D_1} &:= \Psi'\left(\left(du^k,dv^k,1\right)^\top S^k\left(du^k,dv^k,1\right)\right),\\
\Psi'^k_{D_2} &:= \Psi'\left(\left(du^k,dv^k,1\right)^\top T^k\left(du^k,dv^k,1\right)\right),\\
\Psi'^k_{S} &:= \Psi'\left(\|\nabla\left(u^k+du^k\right)\|^2+\|\nabla\left(v^k+dv^k\right)\|^2\right).
\end{aligned}$$

3.2 Coarse-to-Fine Warping

Since the underlying energy functional is non-convex, this fixed point iteration should be embedded in a multiresolution framework in order to avoid local minima. Such a proceeding yields the well-known warping technique: Starting from a coarse version, the original problem is successively refined while being compensated by the already computed motion at the same time. The motion increment that remains to be solved at each level is the solution of the nonlinear equation system (11)-(12) with respect to du^k and dv^k. Note that this problem, however, is convex due to the usage of a convex penalizer such as regularized TV in both data and smoothness term.

4 The Numerical Scheme

In our numerical section, we derive a highly efficient nonlinear solver for the aforementioned nonlinear system of equations (11)-(12). This is done in three steps: First, we propose a suitable discretization. Then, we select a nonhierachical nonlinear method that serves as basic solver for our multigrid approach. And finally, we develop such a multigrid approach - a full approximation scheme (FAS)[7].

4.1 Discretization

Let us now discuss how the nonlinear equation system for the motion increment at each resolution level k can be discretized appropriately. To this end, we consider the unknown functions $du^k(\mathbf{x})$ and $dv^k(\mathbf{x})$ on a rectangular pixel grid with grid size $h_x^k\times h_y^k$. Furthermore, we introduce a consecutive numbering of all pixels from $i=1,...,N^k$, where N^k is the total number of pixels at level k. Then, the approximation to du^k at some pixel i is given by du_i^k.

While spatial derivatives of the original and the warped image data are computed with the fourth order approximation $(1,-8,0,8,-1)/(12h_l^k)$, $l\in\{x,y\}$, temporal derivatives are calculated with a simple two point stencil. Expressions of type $I\left(\mathbf{x}+\mathbf{u}^k\right)$ that are required for the computation of the tensors S and T are obtained by a backward registration approach based on linear interpolation. The resulting tensor components (n,m) at pixel i are denoted by S_{nmi} and T_{nmi}. Finally, we define by $\mathcal{N}_l(i)$ the set of neighbors of pixel i in direction of dimension l. Then, a finite difference approximation of the nonlinear equation system for the motion increment at level k is given by

$$\begin{aligned}
0 \;=\;& \Psi'^k_{D_1i}\left(S_{11i}^k\,du_i^k+S_{12i}^k\,dv_i^k+S_{13i}^k\right) && (13)\\
+&\ \alpha\Psi'^k_{D_2i}\left(T_{11i}^k\,du_i^k+T_{12i}^k\,dv_i^k+T_{13i}^k\right)\\
-&\ \beta\sum_{l=x,y}\sum_{j\in\mathcal{N}_l(i)}\frac{\Psi'^k_{Si}+\Psi'^k_{Sj}}{2}\,\frac{u_j^k+du_j^k-u_i^k-du_i^k}{(h_l^k)^2},
\end{aligned}$$

$$
\begin{aligned}
0 = \; & {\Psi'}^{k}_{D_1 i} \left(S^k_{12i}\, du^k_i + S^k_{22i}\, dv^k_i + S^k_{23i} \right) \\
+ \; & \alpha {\Psi'}^{k}_{D_2 i} \left(T^k_{12i}\, du^k_i + T^k_{22i}\, dv^k_i + T^k_{23i} \right) \\
- \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}_l(i)} \frac{{\Psi'}^{k}_{Si} + {\Psi'}^{k}_{Sj}}{2} \, \frac{v^k_j + dv^k_j - v^k_i - dv^k_i}{(h^k_l)^2}
\end{aligned} \tag{14}
$$

for $i = 1, ..., N^k$. One should note that all nonlinearities within this sparse system of equations are hidden in the expressions ${\Psi'}^{k}_{D_1}$, ${\Psi'}^{k}_{D_2}$ and ${\Psi'}^{k}_{S}$ which depend on du^k_i and dv^k_i.

4.2 The Point Coupled Gauß–Seidel Method

Since the pixels du^k_i and dv^k_i are point coupled explicitly in the data term and implicitly in both the smoothness and the data term – as argument of the nonlinear expressions ${\Psi'}^{k}$ – a solver is desirable that reflects this property. Therefore we propose the usage of a Gauß-Seidel solver with coupled point relaxation (CPR) and frozen coefficients [17]. This amounts to setting the nonlinear expressions fixed and performing one Gauß-Seidel relaxation step where du^k_i and dv^k_i are computed simultaneously. It requires the solution of a linear 2×2 equation system for each pixel i given by

$$
\begin{pmatrix} du^{k,n+1} \\ dv^{k,n+1} \end{pmatrix} = \begin{pmatrix} M^{k,n}_{11} & M^{k,n}_{12} \\ M^{k,n}_{12} & M^{k,n}_{22} \end{pmatrix}^{-1} \begin{pmatrix} ru^{k,n} \\ rv^{k,n} \end{pmatrix} \tag{15}
$$

with matrix entries

$$
\begin{aligned}
M^{k,n}_{11} = \; & {\Psi'}^{k,n}_{D_1 i} S^{k,n}_{11i} + \alpha {\Psi'}^{k,n}_{D_2 i} T^{k,n}_{11i} \\
+ \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}_l(i)} \frac{{\Psi'}^{k,n}_{Si} + {\Psi'}^{k,n}_{Sj}}{2 (h^k_l)^2}
\end{aligned} \tag{16}
$$

$$
\begin{aligned}
M^{k,n}_{22} = \; & {\Psi'}^{k,n}_{D_1 i} S^{k,n}_{22i} + \alpha {\Psi'}^{k,n}_{D_2 i} T^{k,n}_{22i} \\
+ \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}_l(i)} \frac{{\Psi'}^{k,n}_{Si} + {\Psi'}^{k,n}_{Sj}}{2 (h^k_l)^2}
\end{aligned} \tag{17}
$$

$$
M^{k,n}_{12} = {\Psi'}^{k,n}_{D_1 i} S^{k,n}_{12i} + \alpha {\Psi'}^{k,n}_{D_2 i} T^{k,n}_{12i} \tag{18}
$$

and right hand side

$$
\begin{aligned}
ru^{k,n} = \; & -{\Psi'}^{k,n}_{D_1 i} S^{k,n}_{13i} - \alpha {\Psi'}^{k,n}_{D_2 i} T^{k,n}_{13i} \\
+ \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}^-_l(i)} \frac{{\Psi'}^{k,n}_{Si} + {\Psi'}^{k,n}_{Sj}}{2} \, \frac{u^k_j + du^{k,n+1}_j - u^k_i}{(h^k_l)^2} \\
+ \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}^+_l(i)} \frac{{\Psi'}^{k,n}_{Si} + {\Psi'}^{k,n}_{Sj}}{2} \, \frac{u^k_j + du^{k,n}_j - u^k_i}{(h^k_l)^2}
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
rv^{k,n} = \; & -{\Psi'}^{k,n}_{D_1 i} S^{k,n}_{23i} - \alpha {\Psi'}^{k,n}_{D_2 i} T^{k,n}_{23i} \\
+ \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}^-_l(i)} \frac{{\Psi'}^{k,n}_{Si} + {\Psi'}^{k,n}_{Sj}}{2} \, \frac{v^k_j + dv^{k,n+1}_j - v^k_i}{(h^k_l)^2} \\
+ \; & \beta \sum_{l=x,y} \sum_{j \in \mathcal{N}^+_l(i)} \frac{{\Psi'}^{k,n}_{Si} + {\Psi'}^{k,n}_{Sj}}{2} \, \frac{v^k_j + dv^{k,n}_j - v^k_i}{(h^k_l)^2}
\end{aligned} \tag{20}
$$

where n is the iteration index of the point coupled Gauß-Seidel solver. Moreover, $\mathcal{N}^-_l(i) := \{j \in \mathcal{N}_l(i) \mid j < i\}$ denotes the set of already processed pixels, while $\mathcal{N}^+_l(i) := \{j \in \mathcal{N}_l(i) \mid j > i\}$ stands for the set of pixels that have yet to be processed.

Instead of using a linear solver with frozen coefficients one may also think of solving the nonlinear 2×2 system by means of a point coupled Newton-Gauß-Seidel method [8]. However, experiments have shown that such a proceeding is much more expensive in terms of computational costs and thus not efficient enough for our purpose.

4.3 An Efficient Multigrid Algorithm

Common iterative solvers such as the presented point coupled Gauß-Seidel solver have one decisive drawback: Since only locally neighboring pixels are coupled in the relaxation scheme, it may take several thousand iterations to spread information over large distances. As a consequence, only high frequencies (local wavelength) of the error are reduced, while low frequencies (global wavelength) remain almost undamped. This leads to a convergence rate that is typically very fast at the beginning, but slows down significantly already after a few iterations.

In order to overcome this problem bidirectional multigrid methods [7, 8, 20, 27, 30] make use of coarser levels where they obtain useful correction steps. How this works exactly is now described in detail by the example of a nonlinear 2-grid cycle which forms the basic entity of our implementation.

4.3.1 The Basic Nonlinear 2-Grid Cylce (FAS)

Let us start by reformulating the nonlinear system (13)–(14) as

$$
A^{\mathbf{h}}(x^{\mathbf{h}}) = f^{\mathbf{h}}, \tag{21}
$$

where $\mathbf{h} = (h^k_x, h^k_y)^\top$ is the grid size, $A^{\mathbf{h}}(x^{\mathbf{h}})$ is a nonlinear operator, $x^{\mathbf{h}}$ denotes the concatenated vector $((du^{\mathbf{h}})^\top, (dv^{\mathbf{h}})^\top)^\top$ and $f^{\mathbf{h}} := ((f^{\mathbf{h}}_1)^\top, (f^{\mathbf{h}}_2)^\top)^\top$ stands for the right hand side. Then, the FAS [7] strategy works as follows:

I. We perform a presmoothing relaxation step. By applying our basic solver (point coupled Gauß-Seidel) several times we reduce all high frequency components in the error.

II. Since this only gives us an approximation $\tilde{x}^{\mathbf{h}}$ of the solution $x^{\mathbf{h}}$, it would be helpful to know the error $e^{\mathbf{h}} = x^{\mathbf{h}} - \tilde{x}^{\mathbf{h}}$ in order to correct our approximation. Unfortunately, only the residual $f^{\mathbf{h}} - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = r^{\mathbf{h}}$ can be computed directly. However, the following (implicit) relation holds:

$$A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}} + e^{\mathbf{h}}) - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = f^{\mathbf{h}} - A^{\mathbf{h}}(\tilde{x}^{\mathbf{h}}) = r^{\mathbf{h}}. \qquad (22)$$

The basic idea of multigrid methods is now to transfer this residual equation system to a coarser grid (restriction), where it becomes

$$A^{\mathbf{H}}(\underbrace{\tilde{x}^{\mathbf{H}} + e^{\mathbf{H}}}_{x^{\mathbf{H}}}) = \underbrace{r^{\mathbf{H}} + A^{\mathbf{H}}(\tilde{x}^{\mathbf{H}})}_{f^{\mathbf{H}}}. \qquad (23)$$

with coarse grid size $H_x^k \times H_y^k$, $x^{\mathbf{H}} := ((x_1^{\mathbf{H}})^\top, (x_2^{\mathbf{H}})^\top)^\top$ and $f^{\mathbf{H}} := ((f_1^{\mathbf{H}})^\top, (f_2^{\mathbf{H}})^\top)^\top$. Such a proceeding has three advantages: Since high frequencies of the error have already been removed during the presmoothing relaxation step a transfer of the residual equation system is less problematic. Moreover, high frequencies reappear as higher ones on coarser grid, so that they can efficiently be attenuated by our basic solver. And finally, solving a system of equations on a coarser grid reduces the computational costs.

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

IV. Finally we perform a postsmoothing relaxation step with our basic iterative solver. Thus high error frequencies are removed that have been introduced by the interpolation of the coarse grid result.

4.3.2 Advanced Multigrid Strategies

Instead of solving the coarse grid equation system directly one may think of using a third, even coarser grid that provides a correction step for the second one. Such a hierarchical application of the basic 2-grid cycle is called V–cycle. Visiting each coarse grid twice per level yields the so called W–cycle, which offers better convergence rates at the expense of slightly increased computational costs. In Figure 1 examples of such V– and W–cycles are shown.

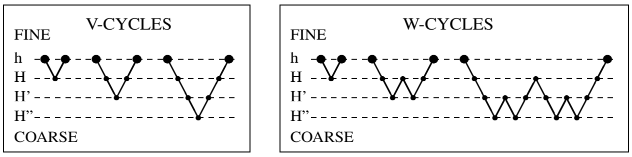

Figure 1. Grid transfers during V– and W–Cycles for two, three and four levels.

4.3.3 Implementation Details

In our implementation we have used one W-cycle as solver at each warping level. Thereby, the number of pre- and postsmoothing iterations was set to 5 each. The transfer between the different grids was realized by non-dyadic versions of averaging and constant interpolation as proposed in [10] while for the construction of the nonlinear coarse grid operators a discretization coarse grid approximation (DCA) [8] approach was used. In this context, one should note that it is important to restrict the tensors S and T in such a way that their positive semi-definiteness is preserved. Then, the point coupled Gauß-Seidel method can still be applied (the data term provides uniqueness of the solution at each level).

5 Results

Let us start by evaluating the qualitative performance of the proposed approach. To this end, we used the *Yosemite* sequence with clouds by Lynn Quam and computed the average angular error [3] for our model with $\alpha = 16.5$, $\beta = 160$, $\epsilon_D = 10^{-1}$ and $\epsilon_S = 10^{-3}$. This synthetic benchmark is one of the most popular benchmarks, since it combines divergent and translational motion, has a large motion discontinuity and non-constant illumination in the sky region. In Table 1 our results are compared to the best results from the literature. As one can see, our approach performs favourably: With an average angular error of $1.72°$ and $2.42°$, it outperforms all other techniques including the original method of Brox *et al.* [9]. In this context one

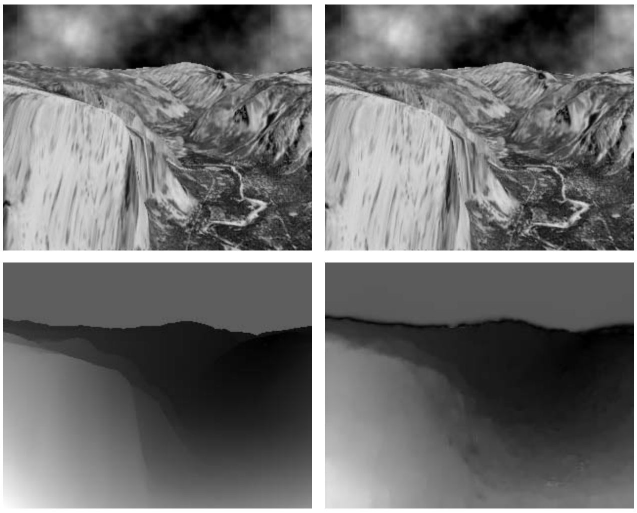

Figure 2. Top Row: Frame 8 and 9 of the Yosemite sequence with clouds (312 x 256). Bottom Row: Ground truth and our result. Average angular error: 2.42 degrees. Computing time: 1089 milliseconds.

Table 1. Comparison to results from the literature with 100 % density. AAE = average angular error. STD = standard deviation. 2D/3D = spatial/spatio-temporal smoothness.

Yosemite with clouds

Technique	AAE	STD
Horn–Schunck, mod. [3]	9.78°	16.19°
Uras *et al.* [3]	8.94°	15.61°
Alvarez *et al.* [1]	5.53°	7.40°
Mémin–Pérez [22]	4.69°	6.89°
Bruhn *et al.* [12]	4.17°	7.72°
Brox *et al.* (2D)	2.46°	7.31°
Our method (2D)	**2.42°**	**6.70°**
Brox *et al.* (3D)	1.94°	6.02°
Our method (3D)	**1.72°**	**6.88°**

should note that the 3-D version of our approach is not real-time capable due to its blockwise computation of the flow fields. However, the corresponding result is still listed in order to show the full potential of the improved model.

In order to get a visual impression of the quality of the estimation we have compared the magnitude of the computed flow field and the ground truth in Figure 2. Evidently, the computed result matches the ground truth very well. The discontinuity at the horizon is preserved and the sky region is estimated correctly, as well. This confirms our observations of a small angular error.

After we have evaluated the quality of our method, let us investigate the efficiency of the proposed numerical scheme. To this end we have run our implementation (with the same parameters as before) on a standard desktop PC with 3.06 GHz Pentium4 CPU and compared it to its basic solver as well an optimized variant of a Quasi-Newton scheme [9, 14, 28]. As test sequence served a downsampled variant of the *Rheinhafen* sequence by Nagel of size 160 × 120. This and other challenging traffic sequence are available at `http://i21www.ira.uka.de/image_sequences`

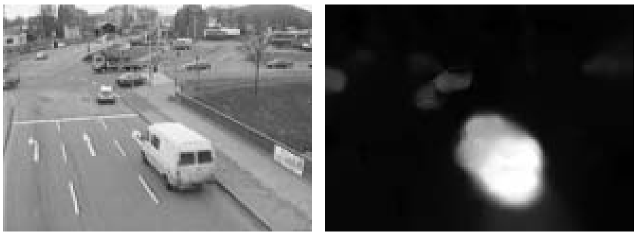

Figure 3. Frame 1130 of the Rheinhafen sequence (resized to 160 x 120) and our result. Computing time: 150 milliseconds.

Table 2. Performance benchmark on a standard desktop PC with 3.06 GHz Pentium 4 CPU. Run times refer to the computation of a single flow field of size 160 x 120.

Solver	Iter.	FPS $[s^{-1}]$	Speedup
Gauß–Seidel (CPR)	4320	0.037	1
Quasi-Newton + SOR	19/5	2.375	64
FAS-W(5,5)	**1**	**6.651**	**180**

All solvers were stopped at exactly that relative error $e_{rel} := \|e\|_2/\|x\|_2$ that was achieved by our multigrid implementation. In this context one should note that this error refers to the final result which is the outcome of sequence of nonlinear equation systems: Thus, errors on coarser warping levels influence the result on a finer warping level such that errors propagate.

The obtained speedups are presented in Table 2. As on can see, the proposed approach outperforms the other methods by far. With a speedup factor of 180 it is even more than two orders of magnitude faster than its basic solver. Moreover, it achieves real-time performance of an unseen quality with more than 6 dense flow fields per second.

The corresponding flow field is shown in Figure 3. One can see that the result looks very realistic. While the van in the foreground moves faster, the cars in the background move slower. Moreover, the boundaries of the objects are rather sharp. One should keep in mind that the computation of such a high quality flow field took only 150 milliseconds.

6 Summary and Conclusions

In this paper we have shown that state-of-the-art accuracy and real-time performance in optical flow computation are not contradictive. Our contributions are twofold: Firstly, we have demonstrated that a modification of the approach of Brox *et al.* [9] which uses a separate instead of a joint robustification yields a model that computes the currently best angular errors in the literature. Secondly, and more importantly, we have focused on deriving a highly efficient multigrid algorithm for this method. It is based on a coarse-to-fine warping strategy combined with a full approximation scheme (FAS) as numerical solver for the resulting nonlinear systems of equations. Benchmarks have shown that this multigrid algorithm is more than two orders of magnitude more efficient than a basic iterative solver of Gauß-Seidel type. As a consequnce, the computation of six dense high quality flow fields per second became possible on standard PCs for sequences of size 160 × 120.

We hope that our work contributes to render variational optical flow techniques more attractive for time-critical applications such as robot navigation or driver assistance systems. Then, these methods can show their true potential in a new and challenging environment.

References

[1] 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.

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

[3] 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.

[4] 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.

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

[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*, volume 3459 of *Lecture Notes in Computer Science*, pages 279–290. 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 *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] 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.

[16] 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.

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

[18] 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.

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

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

[21] R. Kimmel and I. Yavneh. An algebraic multigrid approach for image analysis. *SIAM Journal on Applied Mathematics*, 24(4):1218–1231, 2003.

[22] 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.

[23] 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.

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

[25] 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.

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

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

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

[29] 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.

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