FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
INSTITUT FÜR INFORMATIK (MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG)

Lehrstuhl für Informatik 10 (Systemsimulation)

Multigrid solution of the optical flow system using a combined diffusion and curvature based regularizer

H. Köstler, K. Ruhnau, and R. Wienands

Lehrstuhlbericht 07-07

Multigrid solution of the optical flow system using a combined diffusion and curvature based regularizer

H. Köstler
Lehrstuhl für Systemsimulation
Friedrich-Alexander University of Erlangen-Nuremberg
91058 Erlangen, Germany
koestler@informatik.uni-erlangen.de

K. Ruhnau and R. Wienands
Mathematical Institute
University of Cologne
50931 Cologne, Germany
wienands@math.uni-koeln.de

May 18, 2007

Abstract

Optical flow techniques are used to compute an approximate motion field in an image sequence. We apply a variational approach for the optical flow introducing a combined diffusion and curvature based regularizer that requires the solution of a fourth order system of partial differential equations with jumping coefficients. A geometric multigrid solver for that problem is presented which is composed of collective Gauss-Seidel relaxation and standard geometric transfer operators. The adequacy of using pointwise smoothers is demonstrated with the help of local Fourier analysis. Galerkin based coarse grid operators are applied for an efficient treatment of jumping coefficients. Finally, some results on convergence rates, timings and visual quality of the approximated motion field for synthetic and real world images are shown.

1 INTRODUCTION

Optical flow is commonly defined to be the motion of brightness patterns in a sequence of images. It was introduced by Horn and Schunck [22], who proposed a differential method to compute the optical flow from pairs of images using a brightness constancy assumption and an additional smoothness constraint on the magnitude of the gradient of the velocity field in order to regularize the problem, what we call *diffusion based regularization*. Since then optical flow has been studied intensively and many extensions to that simple variational approach, e.g., considering different regularizing terms, were investigated [21, 29, 15, 34, 19, 35, 36, 5, 6] and applied to various applications ranging from robotics to video compression and Particle Image Velocimetry (PIV), where optical flow provides approximate motion of fluid flows. Especially for PIV, it is necessary to incorporate physically more meaningful regularizers to be able to impose, e.g., an incompressibility condition of the velocity field. Suter [31] introduced therefore a smoothness constraint on the divergence and curl of the velocity field that was used intensively in the following [17, 9, 26, 8]. A well-known regularizer in image registration that is related to optical flow [13, 28] and a special case of a second order div-curl based regularizer [17] is the *curvature based regularizer*. The purpose of the curvature based regularizer is to let affine motion unpenalized while higher order motions are still used to enforce smoothness. Another advantage of a higher order regularizer is that for some applications additional information from features or landmarks is given for the optical flow computation [39]. Here, the higher order regularizer is required to avoid singularities in the solution [12, 14].

We present a variational optical flow approach for a *combined diffusion and curvature based regularizer* in Section 2. We will show that this regularizer beats both the simple diffusion and the curvature based regularizer in terms of accuracy.

Besides accuracy of the approximate motion field obtained by optical flow, an important goal is to achieve real time or close to real time performance in many applications, which makes an efficient numerical solution of the underlying system of partial differential equations (PDE) mandatory. First attempts to use multilevel techniques to speed up optical flow computations are due to Glazer [16] and Terzopoulos [32]. After that, several multigrid based solvers were proposed for different optical flow regularizers (see, e.g., [11, 2, 25, 24, 7, 6]). We develop a geometric multigrid method in Section 3 in order to solve the fourth order system of partial differential equations derived from our variational optical flow approach efficiently. Especially the existence and efficiency of point smoothing methods is investigated in some detail.

In Section 4 optical flow results using the combined diffusion and curvature regularizer both for synthetic and real world images are found and compared to the classical regularizer of Horn and Schunck. We end this paper with an outlook for future developments, e.g., the extension to isotropic or anisotropic versions of the combined regularizer to deal with discontinuities in the velocity field.

2 OPTICAL FLOW MODEL AND DISCRETIZATION

2.1 Continuous system

The variational approach to compute the motion field as proposed by Horn and Schunck [22] uses two assumptions. The first is that a moving object in the image does not change its gray values, what means that for example changes of illumination are neglected. For an image sequence $I : \Omega \times \mathbb{T} \to \mathbb{R}, \Omega \subset \mathbb{R}^2$ describing the gray value intensities for each point $\boldsymbol{x} = (x, y)$ in the regular image domain Ω at time $t \in \mathbb{T} = [0, t_{max}], t_{max} \in \mathbb{N}$, this so-called brightness constancy assumption reads

$$\frac{dI}{dt} = 0 \ . \tag{1}$$

This yields the following identity for the movement of a gray value at (x, y, t)

$$I(\boldsymbol{x}, t) = I(x + dx, y + dy, t + dt) \ . \tag{2}$$

Taylor expansion of $I(x+dx, y+dy, t+dt)$ around (x, y, t) neglecting higher order terms and using (2) gives

$$I_x u + I_y v + I_t \approx 0$$

with the partial image derivatives $\frac{\partial I}{\partial x} = I_x$, $\frac{\partial I}{\partial y} = I_y$, $\frac{\partial I}{\partial t} = I_t$ and the optical flow velocity vector $\boldsymbol{u} = (u, v)^T, u := \frac{dx}{dt}, v := \frac{dy}{dt}$. This brightness constancy assumption is used throughout this paper, but by itself results in an ill-posed, under-determined problem. Therefore additional regularization is required. Horn and Schunck proposed as second assumption a smoothness constraint or a *diffusion based regularizer*

$$S_1(\boldsymbol{u}) = \|\nabla u\|^2 + \|\nabla v\|^2$$

and combined both in an energy functional

$$E_1(\boldsymbol{u}) := \int_\Omega (I_x u + I_y v + I_t)^2 + \alpha S_1(\boldsymbol{u}) dx \tag{3}$$

that is to be minimized. $\alpha \in \mathbb{R}^+$ represents a weighting parameter. The *curvature based regularizer* penalizes second derivatives instead and can be written as

$$S_2(\boldsymbol{u}) = (\Delta u)^2 + (\Delta v)^2 \ .$$

As already mentioned, it is a special case of the div-curl based regularizer [17]

$$S_{2'}(\boldsymbol{u}) = \alpha_1 \|\nabla \mathrm{div} \boldsymbol{u}\|^2 + \alpha_2 \|\nabla \mathrm{curl} \boldsymbol{u}\|^2 \ ,$$

where $\alpha_1 = \alpha_2 = 1$. We propose a combination of the regularizers $S_1(\boldsymbol{u})$ and $S_2(\boldsymbol{u})$ resulting in the *combined diffusion and curvature based regularizer*

$$S_3(\boldsymbol{u}) = \beta S_1(\boldsymbol{u}) + (1-\beta) S_2(\boldsymbol{u}) \ ,$$

where $\beta \in [0,1]$.

The corresponding energy functional $E_3(\boldsymbol{u})$ is obtained by simply replacing S_1 by S_3 in (3). The resulting Euler-Lagrange equations constituting a necessary condition for a minimum of $E_3(\boldsymbol{u})$ are

$$\begin{aligned} \alpha\left((1-\beta)(-\Delta)^2 u + \beta(-\Delta)u\right) + I_x(I_x u + I_y v + I_t) &= 0 && \text{(4a)} \\ \alpha\left((1-\beta)(-\Delta)^2 v + \beta(-\Delta)v\right) + I_y(I_x u + I_y v + I_t) &= 0 && \text{(4b)} \end{aligned}$$

with natural Neumann boundary conditions on $\boldsymbol{u}$ and $-\Delta\boldsymbol{u}$ in case of $\beta \neq 1$. For $\beta = 0$ we obtain a fourth order system from (4), whereas for $\beta = 1$ the original Horn and Schunck second order system results.

The diffusion based regularizer only allows small changes of near vectors and produces very smooth motion fields, but it also smoothes edges out. The curvature based regularizer lets affine motion unpenalized since they are in its kernel, and uses higher order motions to enforce smoothness. By the combination of these two types of regularizers we make a first step towards the physically motivated regularizer $S_{2'}(\boldsymbol{u})$. We will show that the combined regularizer can be computed efficiently and that it is more accurate than each of the two regularizers separately.

2.2 Discrete system

The biharmonic operator Δ^2 which appears in (4) is known to lead to poor multigrid performance. Therefore it is a common approach to split up the biharmonic operator into a system of two Poisson-type equations [33]. Employing this idea, (4) can be transformed into the following system using additional unknown functions w^1 and w^2:

$$\begin{aligned} 0 &= -\Delta u - w^1 && \text{(5a)} \\ 0 &= -\Delta v - w^2 && \text{(5b)} \\ 0 &= \alpha\left(-(1-\beta)\Delta w^1 + \beta w^1\right) + I_x\left(I_x u + I_y v + I_t\right) && \text{(5c)} \\ 0 &= \alpha\left(-(1-\beta)\Delta w^2 + \beta w^2\right) + I_y\left(I_x u + I_y v + I_t\right) \ . && \text{(5d)} \end{aligned}$$

(5a)-(5d) is discretized by finite differences using the standard five-point central discretization Δ_h of the Laplacian (compare with, e.g., [33]). Then the discrete system under consideration for a fixed time t reads

$$\boldsymbol{L}_h(\boldsymbol{x}) \begin{pmatrix} u_h(\boldsymbol{x}) \\ v_h(\boldsymbol{x}) \\ w_h^1(\boldsymbol{x}) \\ w_h^2(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -I_x(\boldsymbol{x},t) I_t(\boldsymbol{x},t) \\ -I_y(\boldsymbol{x},t) I_t(\boldsymbol{x},t) \end{pmatrix} \qquad \text{with} \qquad \text{(6)}$$

$$\boldsymbol{L}_h(\boldsymbol{x}) = \begin{pmatrix} -\Delta_h & 0 & -1 & 0 \\ 0 & -\Delta_h & 0 & -1 \\ I_x^2(\boldsymbol{x},t) & I_x(\boldsymbol{x},t) I_y(\boldsymbol{x},t) & \alpha\left(-(1-\beta)\Delta_h + \beta\right) & 0 \\ I_x(\boldsymbol{x},t) I_y(\boldsymbol{x},t) & I_y^2(\boldsymbol{x},t) & 0 & \alpha\left(-(1-\beta)\Delta_h + \beta\right) \end{pmatrix} ,$$

$\boldsymbol{x} \in \Omega_h$ and discrete functions u_h, v_h, w_h^1, w_h^2. Here, Ω_h denotes the discrete image domain, i.e., each $\boldsymbol{x} \in \Omega_h$ refers to a pixel. The mesh size h is usually set to 1 for optical flow applications. The corresponding homogeneous Neumann boundary conditions for the four unknown functions are discretized by central differences as well.

The determinant of the discrete system is given by

$$\begin{aligned} \det\left(\boldsymbol{L}_h\right) &= \alpha^2(\beta-1)^2\left(-\Delta_h\right)^4 + 2\alpha^2(\beta-\beta^2)\left(-\Delta_h\right)^3 && \text{(7)} \\ &+ \left(\alpha^2\beta^2 + \alpha(1-\beta)(I_x^2+I_y^2)\right)\left(-\Delta_h\right)^2 + \alpha\beta(I_x^2+I_y^2)\left(-\Delta_h\right) \end{aligned}$$

omitting the dependency on space and time for convenience. For the special cases $\beta = 0$ and $\beta = 1$ we obtain

$$\det\left(\boldsymbol{L}_h\right) = \alpha^2\Delta_h^4 + \alpha\left(I_x^2+I_y^2\right)\Delta_h^2 \qquad \text{and} \qquad \det\left(\boldsymbol{L}_h\right) = \alpha^2\Delta_h^2 - \alpha\left(I_x^2+I_y^2\right)\Delta_h \ ,$$

respectively. The principle part of $\det(\boldsymbol{L}_h)$ is Δ_h^m with $m = 4$ for $\beta \in [0, 1)$ and $m = 2$ for $\beta = 1$ due to $\alpha > 0$. Hence four boundary conditions for $\beta \neq 1$ are required and two boundary conditions for $\beta = 1$ (see, e.g., [3, 33]). This requirement is met by our choice of boundary conditions since we use natural Neumann boundary conditions on u, v and additionally on $-\Delta u = w^1, -\Delta v = w^2$, if $\beta \neq 1$, resulting from the minimization of the energy functional, see above.

3 MULTIGRID SOLVER

In nowadays applications a real-time solution of the optical flow system becomes more and more important. Hence an appropriate multigrid solver is an obvious choice for the numerical solution of the resulting linear system, since multigrid methods are known to be among the fastest solvers for discretized elliptic partial differential equations.

Multigrid methods (see, e.g., [3, 18, 4, 33, 37]) are mainly motivated by two basic principles.

1. Smoothing principle: Many iterative methods have a strong error smoothing effect if they are applied to discrete elliptic problems.

2. Coarse grid correction principle: A smooth error term can be well represented on a coarser grid where its approximation is substantially less expensive.

These two principles suggest the following structure of a two-grid cycle: Perform ν_1 steps of an iterative relaxation method S_h on the fine grid (pre-smoothing), compute the defect of the current fine grid approximation, restrict the defect to the coarse grid, solve the coarse grid defect equation, interpolate the obtained error correction to the fine grid, add the interpolated correction to the current fine grid approximation (coarse grid correction), perform ν_2 steps of an iterative relaxation method on the fine grid (post-smoothing). Instead of an exact solution of the coarse grid equation, it can be solved by a recursive application of the two-grid iteration, yielding a multigrid method. We assume standard coarsening here, i.e., the sequence of coarse grids is obtained by repeatedly doubling the mesh size in each space direction, i.e. $h \to 2h$.

The crucial point for any multigrid method is to identify the "correct" multigrid components (i.e., relaxation method, restriction, interpolation, ...) yielding an efficient interplay between relaxation and coarse grid correction. A useful tool for a proper selection is local Fourier analysis.

3.1 Basic elements of loacal Fourier analysis

Local Fourier analysis [3, 33, 38] is mainly valid for operators with constant or smoothly varying coefficients. It is based on the simplification that boundary conditions are neglected and all occurring operators are extended to an infinite grid

$$G_h := \left\{ \boldsymbol{x} = (x, y)^T = h(n_x, n_y)^T \text{ with } (n_x, n_y) \in \mathbb{Z}^2 \right\}.$$

On an infinite grid the discrete solution, its current approximation and the corresponding error or residual can be represented by linear combinations of certain exponential functions—the Fourier components—which form a unitary basis of the space of bounded infinite grid functions, the Fourier space. Regarding our optical flow system composed of four discrete equations (5a)-(5d), a proper unitary basis of vector-valued Fourier components is given by

$$\boldsymbol{\varphi}_h(\boldsymbol{\theta}, \boldsymbol{x}) := \exp(\mathrm{i}\,\boldsymbol{\theta}\boldsymbol{x}/h) \cdot \mathbf{I} \quad \text{with} \quad \mathbf{I} = (1, 1, 1, 1)^T,\ \boldsymbol{\theta} \in \Theta := (-\pi, \pi]^2,\ \boldsymbol{x} \in G_h$$

and complex unit $\mathrm{i} = \sqrt{-1}$ yielding the Fourier space

$$F(G_h) := \operatorname{span}\left\{ \boldsymbol{\varphi}_h(\boldsymbol{\theta}, \boldsymbol{x}) \ :\ \boldsymbol{\theta} \in \Theta \right\}.$$

Then, the main idea of local Fourier analysis is to analyze different multigrid components or even complete two-grid cycles by evaluating their effect on the Fourier components. Especially the analysis of the smoothing method is based on a distinction between "high" and "low" Fourier frequencies governed by the coarsening strategy under consideration. If standard coarsening is selected, each "low-frequency"

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{00} \in \Theta_{\text{low}} := (-\pi/2, \pi/2]^2$$

is coupled with three "high-frequencies"

$$\boldsymbol{\theta}^{11} := \boldsymbol{\theta}^{00} - (\operatorname{sign}(\theta_1), \operatorname{sign}(\theta_2))\,\pi, \quad \boldsymbol{\theta}^{10} := \boldsymbol{\theta}^{00} - (\operatorname{sign}(\theta_1), 0)\,\pi,$$
$$\boldsymbol{\theta}^{01} := \boldsymbol{\theta}^{00} - (0, \operatorname{sign}(\theta_2))\,\pi \qquad \left(\boldsymbol{\theta}^{11}, \boldsymbol{\theta}^{10}, \boldsymbol{\theta}^{01} \in \Theta_{\text{high}} := \Theta \setminus \Theta_{\text{low}}\right)$$

in the transition from G_h to G_{2h}. That is, the related three high-frequency components are not visible on the coarse grid G_{2h} as they coincide with the coupled low-frequency component:

$$\boldsymbol{\varphi}_h\left(\boldsymbol{\theta}^{00}, \boldsymbol{x}\right) = \boldsymbol{\varphi}_h\left(\boldsymbol{\theta}^{11}, \boldsymbol{x}\right) = \boldsymbol{\varphi}_h\left(\boldsymbol{\theta}^{10}, \boldsymbol{x}\right) = \boldsymbol{\varphi}_h\left(\boldsymbol{\theta}^{01}, \boldsymbol{x}\right) \qquad \text{for} \quad \boldsymbol{x} \in G_{2h}.$$

This is of course due to the 2π-periodicity of the exponential function.

3.2 Measure of h-ellipticity

A well-chosen relaxation method obviously has to take care of the high-frequency error components since they cannot be reduced on coarser grids by the coarse grid correction. The measure of h-ellipticity is often used to decide whether or not this can be accomplished by a point relaxation method [3, 33, 38]. A sufficient amount of h-ellipticity indicates that point-wise error smoothing procedures can be constructed for the discrete operator $\boldsymbol{L}_h$ under consideration. The measure of h-ellipticity for our system of equations is defined by

$$E_h(\boldsymbol{L}_h) := \frac{\min\left\{\left|\det\left(\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta})\right)\right| \;:\; \boldsymbol{\theta} \in \Theta_{\text{high}}\right\}}{\max\left\{\left|\det\left(\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta})\right)\right| \;:\; \boldsymbol{\theta} \in \Theta\right\}}$$

where the complex (4×4)-matrix

$$\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}) = \begin{pmatrix} -\widetilde{\Delta}_h(\boldsymbol{\theta}) & 0 & -1 & 0 \\ 0 & -\widetilde{\Delta}_h(\boldsymbol{\theta}) & 0 & -1 \\ I_x^2 & I_x I_y & \alpha\left(-(1-\beta)\widetilde{\Delta}_h(\boldsymbol{\theta}) + \beta\right) & 0 \\ I_x I_y & I_y^2 & 0 & \alpha\left(-(1-\beta)\widetilde{\Delta}_h(\boldsymbol{\theta}) + \beta\right) \end{pmatrix}$$

is the Fourier symbol of $\boldsymbol{L}_h$, i.e.,

$$\boldsymbol{L}_h \boldsymbol{\varphi}_h(\boldsymbol{\theta}, \boldsymbol{x}) = \widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}) \boldsymbol{\varphi}_h(\boldsymbol{\theta}, \boldsymbol{x}).$$

The Fourier symbol of the Laplacian reads (see, e.g., [3, 33, 38])

$$-\widetilde{\Delta}_h(\boldsymbol{\theta}) = \frac{4}{h^2}\left(\sin^2(\theta_1/2) + \sin^2(\theta_2/2)\right) \quad \text{with} \quad \boldsymbol{\theta} \in \Theta.$$

Now, $\det(\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}))$ is simply given by (7) where $-\Delta_h$ has to be replaced by $-\widetilde{\Delta}_h(\boldsymbol{\theta})$. For the derivation of $E_h(\boldsymbol{L}_h)$ it is important to note that $-\widetilde{\Delta}_h(\boldsymbol{\theta}) \geq 0$. Moreover, for the four coefficients

$$c_1 := \alpha\beta(I_x^2 + I_y^2), \quad c_2 := \alpha^2\beta^2 + \alpha(1-\beta)(I_x^2 + I_y^2), \quad c_3 := 2\alpha^2(\beta - \beta^2), \quad c_4 := \alpha^2(\beta - 1)^2$$

occurring in $\det(\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}))$, we have $c_1, c_2, c_3, c_4 \geq 0$ for $\alpha > 0$, $\beta \in [0, 1]$. Since

$$f(x) = c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4$$

is monotonically increasing for $x, c_1, c_2, c_3, c_4 \geq 0$, the minimal ($\boldsymbol{\theta} \in \Theta_{\text{high}}$) and maximal ($\boldsymbol{\theta} \in \Theta$) values of $-\widetilde{\Delta}(\boldsymbol{\theta})$ and $|\det(\widetilde{\boldsymbol{L}}_h(\boldsymbol{\theta}))|$ coincide. In particular we have

$$\min_{\boldsymbol{\theta} \in \Theta_{\text{high}}}\left(-\widetilde{\Delta}_h(\boldsymbol{\theta})\right) = -\widetilde{\Delta}_h(-\pi/2, 0) = \frac{2}{h^2}, \qquad \max_{\boldsymbol{\theta} \in \Theta}\left(-\widetilde{\Delta}_h(\boldsymbol{\theta})\right) = -\widetilde{\Delta}_h(\pi, \pi) = \frac{8}{h^2}.$$

As a consequence, the measure of h-ellipticity for the discrete system (5a)-(5d) turns out to be

$$E_h(\boldsymbol{L}_h) = \frac{8\alpha(\beta-1)^2 + 8\alpha(\beta-\beta^2)h^2 + 2(\alpha\beta^2 + (1-\beta)(I_x^2 + I_y^2))h^4 + \beta(I_x^2 + I_y^2)h^6}{2048\alpha(\beta-1)^2 + 512\alpha(\beta-\beta^2)h^2 + 32(\alpha\beta^2 + (1-\beta)(I_x^2 + I_y^2))h^4 + 4\beta(I_x^2 + I_y^2)h^6}.$$

For the special cases $\beta = 0, 1$ this gives

$$E_h(\boldsymbol{L}_h) = \frac{4\alpha + (I_x^2 + I_y^2)h^4}{1024\alpha + 16(I_x^2 + I_y^2)h^4} \qquad \text{and} \qquad E_h(\boldsymbol{L}_h) = \frac{2\alpha + h^2}{32\alpha + 4h^2},$$

respectively. Note that $E_h(\boldsymbol{L}_h) > 0$ for all possible choices of $\alpha, h > 0, I_x^2 + I_y^2 \geq 0, \beta \in [0, 1]$. This is a strong and very satisfactory robustness result for such a complicated system involving several parameters. Even in the limit of small mesh size $h \to 0$ the measure of h-ellipticity is bounded away from zero since we have

$$\lim_{h \to 0} E_h(\boldsymbol{L}_h) = \begin{cases} 1/16 & \text{for } \beta = 1 \\ 1/256 & \text{for } \beta \neq 1. \end{cases}$$

3.3 Smoothing method

Due to the above derivations it can be expected that the optical flow system under consideration is appropriate to point smoothing. The straight-forward generalization of a scalar smoothing method to a system of PDEs like (6) is a collective relaxation method. This relaxation method sweeps over all grid points $\boldsymbol{x} \in \Omega_h$ in a certain order, for example, in a lexicographic or a red-black manner. At each grid point the four difference equations are solved simultaneously, i.e., the corresponding variables $u_h(\boldsymbol{x})$, $v_h(\boldsymbol{x})$, $w_h^1(\boldsymbol{x})$ and $w_h^2(\boldsymbol{x})$ are updated simultaneously. This means that a (4×4)-system has to be solved at each grid point.

First of all we have to note that the large sparse matrix which corresponds to the discrete system (6) is neither symmetric nor diagonally dominant. Furthermore, it is not an M-Matrix due to positive off-diagonal entries. As a consequence, most of the classical convergence criteria for standard iterative methods like Jacobi or Gauss-Seidel relaxation do not apply and it has to be expected that these methods might diverge for certain parameter choices. In our numerical tests for collective lexicographic or red-black Gauss-Seidel relaxation (abbreviated by GS-LEX and GS-RB, respectively) we always observed an overall convergence, although for certain combinations of α, β, I_x, I_y there were single relaxation steps with an increasing residual. An example of such a convergence history is shown in Fig. 1 for collective Jacobi, GS-LEX and GS-RB relaxation.

However, if a relaxation method is applied within a multigrid algorithm then we are mainly interested in its smoothing properties. That is, the relaxation is aimed at a sufficient reduction of the high-frequency components of the error between the exact solution and the current approximation, see above. A quantitative measure of its efficiency represents the smoothing factor μ_{loc} obtained by local Fourier analysis. μ_{loc} is defined as the worst asymptotic error reduction by one relaxation step of all high-frequency error components. For more details on local Fourier analysis we refer to the literature [3, 33, 38]. Dealing with operators based on variable coefficients prevents a direct application of local Fourier analysis. However, the analysis can be applied to the locally frozen operator at a fixed grid point $\boldsymbol{\xi}$. Replacing the variable $\boldsymbol{x}$ by a constant $\boldsymbol{\xi}$, one obtains an operator $\boldsymbol{L}_h(\boldsymbol{\xi})$ with constant frozen coefficients. In case of smoothly varying coefficients the smoothing factor for $\boldsymbol{L}_h(\boldsymbol{x})$ can be bounded by the maximum over the smoothing factors for the locally frozen operator, i.e.,

$$\mu_{\text{loc}}\left(\boldsymbol{L}_h(\boldsymbol{x})\right) = \max_{\boldsymbol{\xi} \in \Omega_h} \mu_{\text{loc}}\left(\boldsymbol{L}_h(\boldsymbol{\xi})\right). \tag{8}$$

As a popular test case we consider frame 8 of the Yosemite sequence shown in Fig. 4. Table 1 presents the corresponding smoothing factors calculated via (8) for GS-LEX and GS-RB with varying β. α is fixed at 1500 which turned out to be a proper choice w.r.t. the average angular error (9) in many situations, see below. Obviously there is hardly any influence of the parameter β on the resulting smoothing factor. We always observe nearly the same smoothing factors as they are well-known for the Poisson equation (i.e. $\mu = 0.5$ for GS-LEX and $\mu = 0.25$ for GS-RB). Systematic tests show that the same statement is also valid for the parameter α. As a consequence we can expect to obtain the typical multigrid efficiency as long as the coarse grid correction works properly, compare with Section 3.4. The situation is considerably more complicated if we apply decoupled relaxations (compare with [33]) which will be discussed elsewhere.

Note that I_x and I_y are not varying smoothly over the image domain Ω_h for this test case. Instead we have moderate jumps in the coefficients. As a consequence, the smoothing factors from

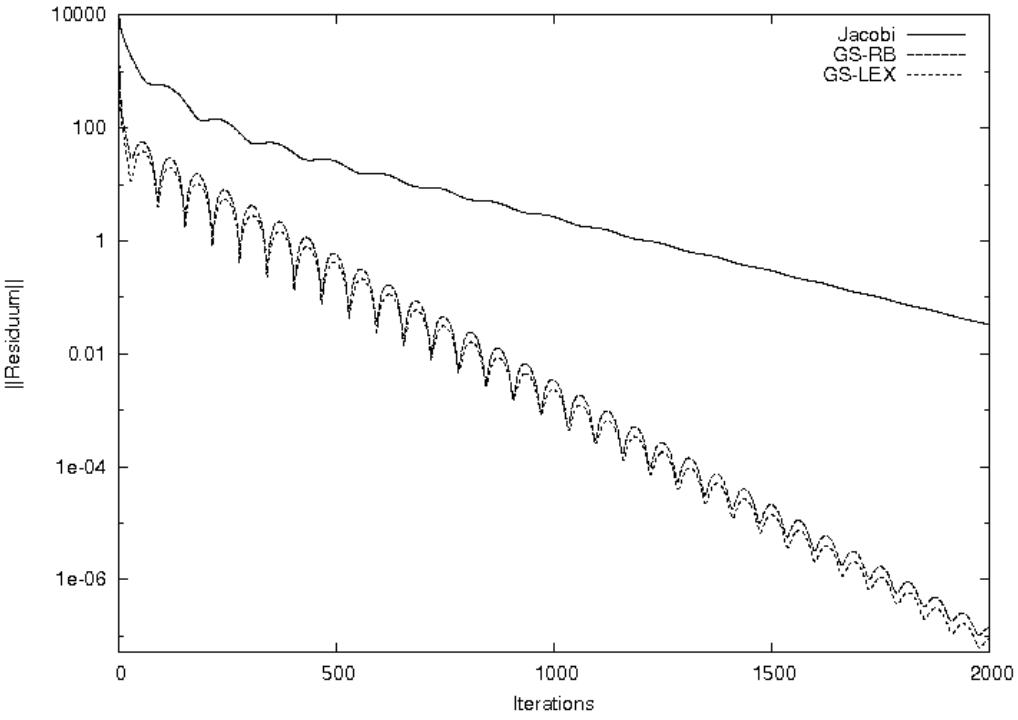


Figure 1: Residual improvement of relaxations.

Table 1: Smoothing factors for GS-LEX and GS-RB, $\alpha = 1500$.

β	0	0.4	1
GS-LEX	0.49973	0.49980	0.49970
GS-RB	0.25003	0.25009	0.25000

Table 1 are not justified rigorously. However from practical experience, they can be considered as heuristic but reliable estimates for the actual smoothing properties especially since we only have moderate jumps. To back up the theoretical results from smoothing analysis we also tested the smoothing effect of the collective relaxations numerically. The smoothing effect of GS-LEX can be clearly seen from Fig. 2. Here, the initial (random) error on a 33×33 grid (a scaled down version of frame 8 from the Yosemite sequence) and the error after five collective GS-LEX steps of the first component u of the optical flow velocity vector is shown.

Summarizing, there is sufficient evidence that collective damped Jacobi, GS-LEX and GS-RB relaxation are reasonable *smoothing* methods even though they might diverge for single relaxation steps as *stand alone solvers*.

3.4 Coarse grid correction

Next to the collective GS relaxation standard multigrid components are applied. To handle the jumping coefficients in I_x and I_y we use Galerkin coarse grid operators. Since there are only moderate jumps it is not necessary to consider operator-dependent transfers but we can stay with straight-forward geometric transfers like full-weighting and bilinear interpolation. Throughout our numerical experiments V(2,2)-cycles are employed (i.e., $\nu_1 = 2$ pre- and $\nu_2 = 2$ post-relaxations). For details concerning these multigrid components we refer to the well-known literature again.

Since we are interested in a real-time solution it is necessary to use the full multigrid (FMG) technique (see, e.g., [3, 33]). Here, the initial approximation on the fine grid is obtained by the computation and interpolation of approximations on coarser grids. A properly adjusted FMG algorithm yields an asymptotically optimal method, i.e. the number of arithmetic operations is proportional to the number of grid points and at the same time the error of the resulting fine grid solution is approximately equal to the discretization error.

4 EXPERIMENTAL RESULTS

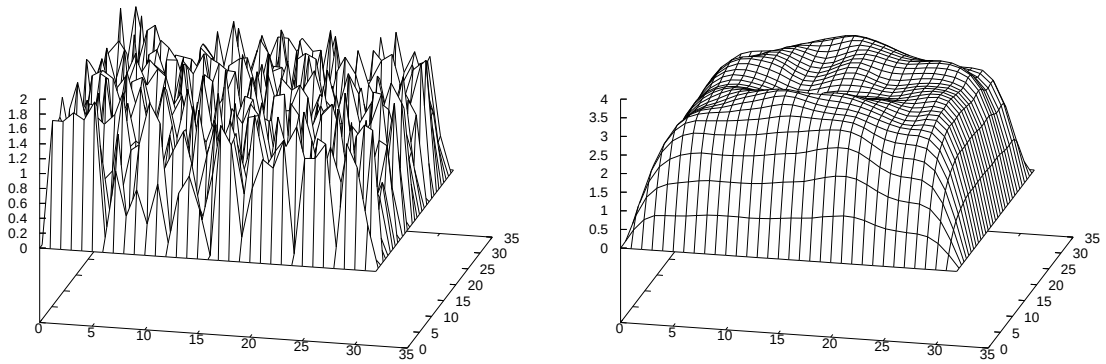
Next, the numerical performance of the multigrid solver described above is investigated and the quality of our optical flow model is demonstrated.

In general, it is very hard to quantify the quality of the optical flow velocity field. For synthetic image sequences, often a ground truth motion field (see [27] for details) is used to measure the quality of a computed optical flow field by the Average Angular Error (AAE). It is calculated via (cf. [6])

$$AAE(\boldsymbol{u}_c, \boldsymbol{u}_e) = \frac{1}{|\Omega|} \int_\Omega \arccos\left(\frac{\boldsymbol{u}_c^T \boldsymbol{u}_e}{|\boldsymbol{u}_c||\boldsymbol{u}_e|} \right) dx \ , \tag{9}$$

where $\boldsymbol{u}_c = (u_c, v_c, 1)$ is the ground truth and $\boldsymbol{u}_e = (u_e, v_e, 1)$ the estimated optical flow vector.

Most real world image sequences do not offer a ground truth motion field, therefore in this case the quality of the optical flow is often measured visually by plotting the vector field and comparing



Initial Error | Error after five GS-LEX steps

Figure 2: Error smoothing of GS iteration for a scaled down version of frame 8 from the Yosemite sequence.

it to the expected result. For example one can check, if the vector field is smooth inside objects, and if edges from different movements are preserved, e.g., objects moving over a static background.

All experiments for different combinations of α and β (see below) were performed using a single FMG-V(2,2) cycle with collective GS-RB as smoother. The same visual and AAE results can be also obtained by five V(2,2) cycles. Input images are smoothed by a discrete Gaussian filter mask (standard deviation $\sigma = 1.2$) in order to ensure a robust computation of the image derivatives by finite difference approximations.

For constant coefficients I_x and I_y one obtains the typical multigrid convergence factors similar as for the Poisson equation which can be nicely predicted by local Fourier analysis. For jumping coefficients a slight deterioration of the convergence rate can be observed. Table 2 lists some representative results. Different values of α that are useful for the application do not have a substantial impact on the convergence rates. The best convergence rates are achieved when the combination of α and β is optimal with respect to the quality of the solution which is an interesting observation by itself. Fig. 3 shows an AAE (9) plot over β for $\alpha = 1500$. There the best quality with respect to AAE is obtained for $\beta \approx 0.4$. On the other hand the best convergence rates for $\alpha = 1500$ are also obtained for $\beta \approx 0.4$.

To give an impression of the performance of our optical flow algorithm we list in Table 3 runtimes for a FMG-V(2,2) cycle for different image sizes. The time measurement are done on an AMD Opteron 248 Cluster node with 2.2 GHz, 64 KB L1 cache, 1 MB L2 cache, and 4 GByte DDR-333 RAM. Of course, by a hardware-specific performance optimization of the multigrid solver on current architectures these times can be improved for real applications [10, 23].

In the following we use two sequences, one synthetic and one real world [1], to evaluate our optical flow model.

4.1 Yosemite sequence

The Yosemite sequence with clouds, created by Lynn Quam [20], is a rather complex test case (see Fig. 4). It consists of 15 frames of size 316×252 and depicts a flight through the Yosemite national park. In this sequence translational (clouds) and divergent motion (flight) is present. Additionally we have varying illumination in the region of the clouds, thus our constant brightness assumption is not fulfilled there.

All tests were obtained with frames 8 and 9 of the Yosemite sequence. First we consider in Fig. 3 the AAE for $\alpha = 500, 1500, 5000$ and varying β. $\alpha = 500$ was chosen because it was tested to give the optimal value – w.r.t. a minimal AAE – for the second order system. The combined regularizer produces the best result. It is able to outperform both the diffusion based and also the curvature based regularizer. Since the AAE is measured over the whole image domain, also small improvements of the AAE can lead to a substantial improvement of the local visual quality of the resulting optical flow field.

Figure 4 shows image details of the resulting velocity fields for the Yosemite sequence, where we choose $\alpha = 1500$ to do a visual comparison of different values of β. The right half of this detail includes the high mountain from the middle of the images. The mountains are moving from right to left, while the clouds region is moving (pure horizontally) from left to right. For $\beta = 1$ one can see the usual behavior of the original Horn and Schunck regularizer, which tries to produce a smooth

Table 2: Convergence rates for the computation of the optical flow from frame 8 and 9 of the Yosemite sequence with $\alpha = 1500$.

	GS-LEX			GS-RB		
Cycle	$\beta = 0$	$\beta = 0.4$	$\beta = 1$	$\beta = 0$	$\beta = 0.4$	$\beta = 1$
1	0.053	0.051	0.048	0.091	0.090	0.074
2	0.054	0.042	0.045	0.070	0.055	0.044
3	0.096	0.065	0.148	0.115	0.069	0.127
4	0.124	0.086	0.196	0.156	0.093	0.181
5	0.131	0.093	0.232	0.172	0.110	0.233

Table 3: Runtimes of the optical flow FMG-V(2,2) multigrid solver for different image sizes.

size	runtime (in ms)
256×192	305
256×256	420
316×252	560
640×480	1900

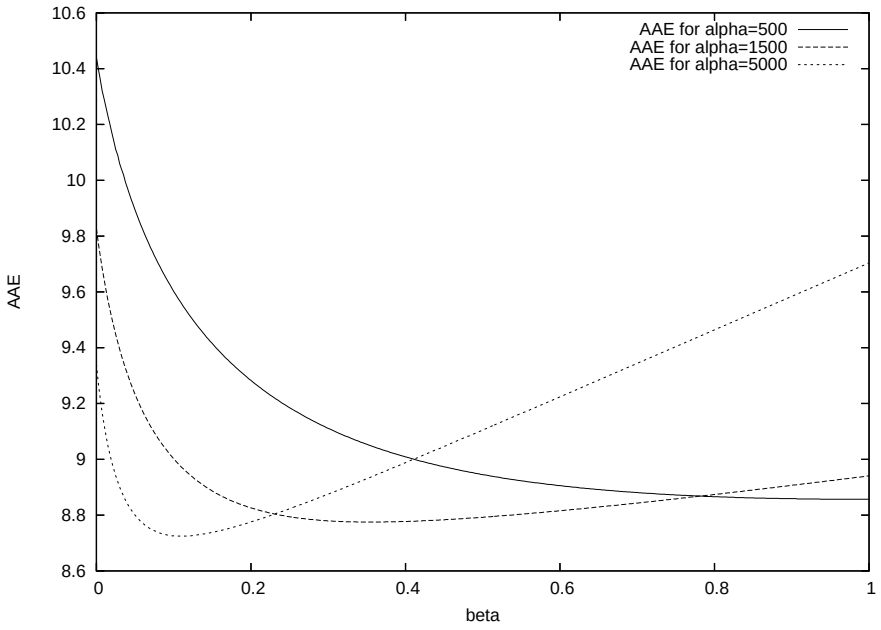


Figure 3: AAE plot of the calculated optical flow between pictures 8 and 9 from the Yosemite sequence for $\alpha = 500$, $\alpha = 1500$ and $\alpha = 5000$.

solution even over the mountain crest. The fourth order system performs better in this regard, as the region of influence is notably smaller for example at the right crossover. The combined regularizer with $\beta = 0.4$ exhibits a mixture of both effects and leads to a smaller AAE over the whole image. One can also observe that all methods fail to calculate the pure horizontal flow in the clouds region. That is due to the fact, that the brightness varies here and thus the constant brightness assumption of the data term does not hold.

4.2 Hamburg Taxi sequence

As a real image sequence we take the famous Hamburg Taxi sequence of size 256×190 shown in Fig. 5. For the multigrid solver it was scaled up to 256×192. In this street scene, originally provided by the University of Hamburg and available from `http://i21www.ira.uka.de/image_sequences`, we find four moving objects: the taxi turning the corner, a car in the lower left driving from left to right, a van in the lower right entering the picture and driving right to left, and a pedestrian in the upper left.

The two image details from the optical flow are taken between the first and the second car. In the lower left corner the movement of the dark car and in the top right corner the diagonal movement of the bright car is visualized. The left vector field plot shows the optical flow for the diffusion based regularizer and optimized α. As one can see, the top right area includes the diagonal vectors for the bright car, but the movement of the dark car suffers from the second order smoothness. Also the static background area between both cars is covered with a smooth solution that tries to couple the movements of both cars. The vector field from our fourth order system is visualized on

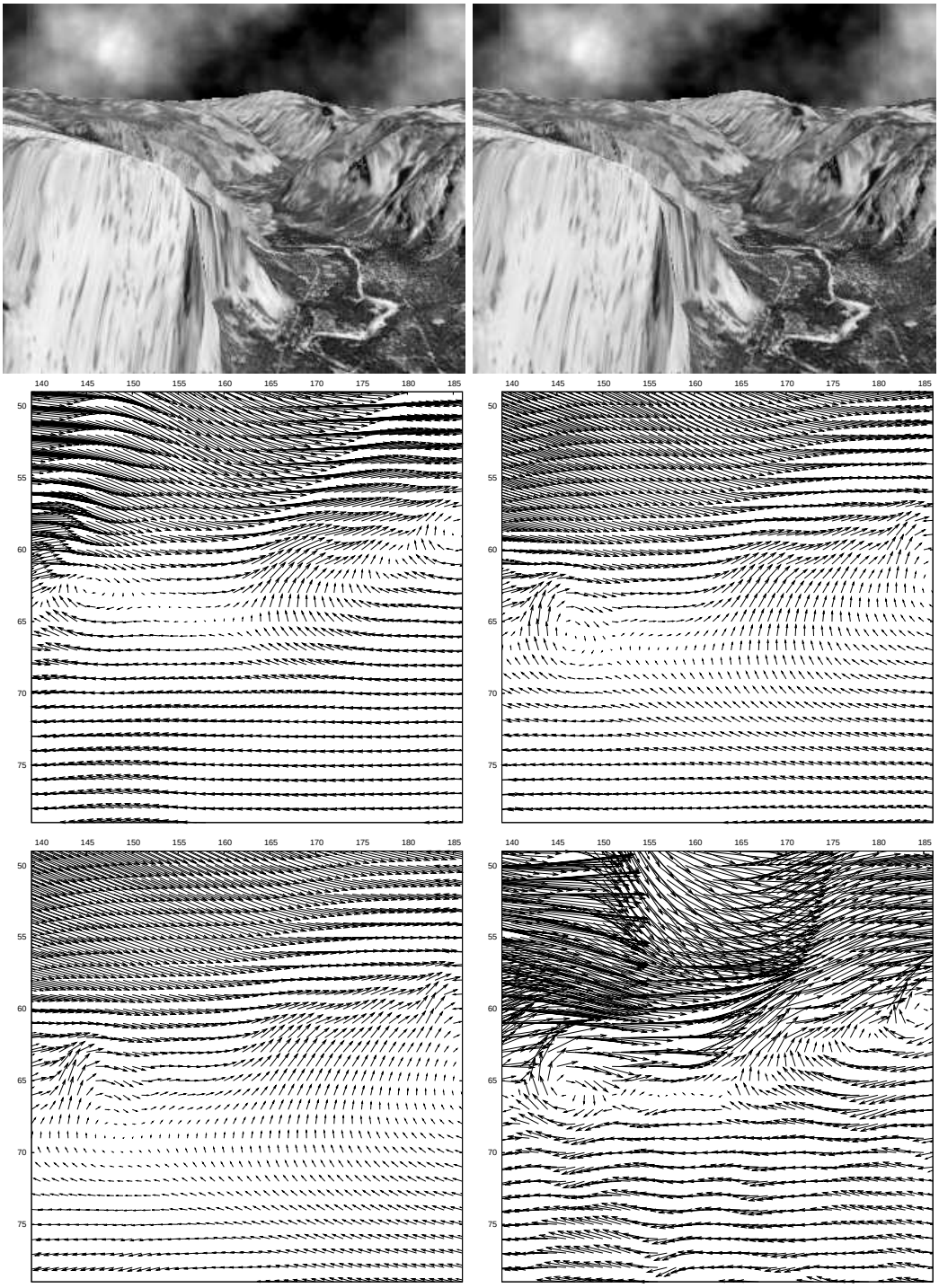


Figure 4: First line: Frames 8 and 9 from Yosemite sequence. Second line: A detail from the optical flow located left from the highest mountain in the middle of the image. It was calculated with $\alpha = 1500$ and (from left to right) $\beta = 0$ and $\beta = 0.4$. Third line: The same detail for $\beta = 1$ and on the right for $\beta = 0$ using S_4.

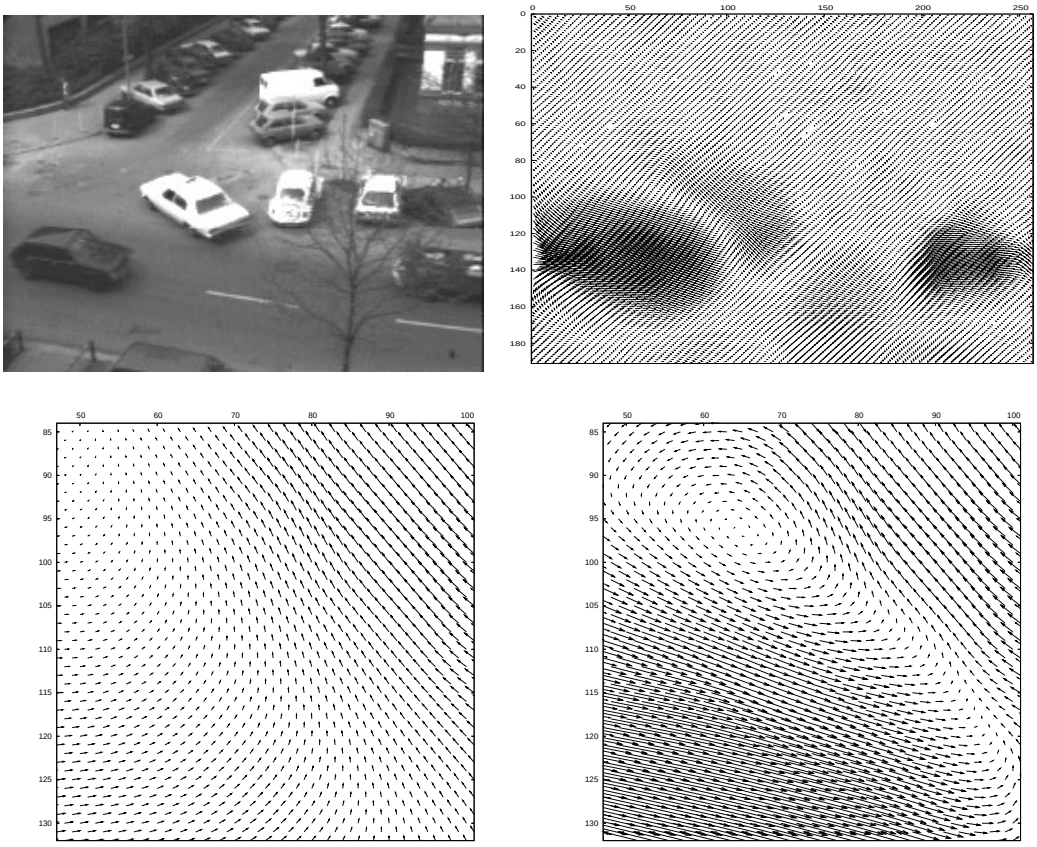


Figure 5: First line: Image one from the Hamburg Taxi sequence and the full vector field. Second line: Two vector fields from a detail of the optical flow using $\alpha = 1500$ and the second order regularizer ($\beta = 1$) on the left and our fourth order system ($\beta = 0$) on the right.

the right, again with optimized α. One can clearly see the two areas from the movement of both cars. Also the background between both cars is preserved by a relatively small flow. One can even recognize the curves from the dark car, as the flow field quickly changes outside the car object. These observations demonstrate that the fourth order system does not penalize affine motion that occurs in this sequence. The combined regularizer mixes up the two optical flow fields obtained from $\beta = 1$ and $\beta = 0$ and it depends on the application, which value of β is best.

5 CONCLUSIONS AND OUTLOOK

We presented and evaluated a combined diffusion and curvature based regularizer for optical flow computations. The arising fourth order system of PDEs was solved efficiently by a geometric multigrid solver. Here, it shows that the best results are obtained, when the weighting between regularizer and brightness constancy assumption is chosen such that the multigrid solver shows an optimal convergence rate. This is an interesting observation and it has to be investigated, if this can be used to choose the weighting parameter automatically.

To improve the static weighting of the regularizer, which produces an equally smooth solution throughout the picture, one could introduce a diffusivity function $g : \mathbb{R} \to \mathbb{R}^+$ given by, for instance, the regularized total variation diffusivity [30]

$$g(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon^2}} \ , \tag{10}$$

where $\epsilon = 10^{-3}$ is used to avoid unbounded diffusivities. This leads to an *isotropic regularizer*

$$S_4(\boldsymbol{u}) = g(|\nabla I|^2) S_3(\boldsymbol{u}) \ ,$$

what can also be thought of as using a space-dependent α parameter. Fig. 4 contains the results from the same image detail of the Yosemite sequence. What one can see here is a very sharp edge from the mountain crest and a smooth solution directly below.

Next steps are the extension of the regularizer to the physically motivated div-curl based regularizer, or nonlinear regularizers, where α and β depend on the velocity field.

Furthermore we want to apply the curvature based regularizer to motion blur computed by a combined optical flow and ray tracer motion field [39], to overcome the problem of the diffusion based regularizer that introduces singularities in the Euler-Lagrange equations, since some motion vectors are fixed within the optical flow model.

References

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

[2] R. Battiti, E. Amaldi, and C. Koch, *Computing optical flow across multiple scales: an adaptive coarse-to-fine strategy*, International Journal of Computer Vision, 6 (1991), pp. 133–145.

[3] A. Brandt, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, GMD-Studie Nr. 85, Sankt Augustin, West Germany, (1984).

[4] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, 2nd ed., 2000.

[5] T. Brox and J. Weickert, *Nonlinear matrix diffusion for optic flow estimation*, in Pattern Recognition, L. van Gool, ed., vol. 2449 of Lecture Notes in Computer Science, 2002, pp. 446–453.

[6] A. Bruhn, *Variational Optic Flow Computation: Accurate Modeling and Efficient Numerics*, PhD thesis, Department of Mathematics, Saarland University, Saarbrücken, Germany, 2006.

[7] I. Christadler, H. Köstler, and U. Rüde, *Robust and efficient multigrid techniques for the optical flow problem using different regularizers*, in Proceedings of 18th Symposium Simulationstechnique ASIM 2005, F. Hülsemann, M. Kowarschik, and U. Rüde, eds., vol. 15 of Frontiers in Simulation, SCS Publishing House, 2005, pp. 341–346. also exists as Tech. Rep. 05-6.

[8] T. Corpetti, D. Heitz, G. Arroyo, E. Mémin, and A. Santa-Cruz, *Fluid experimental flow estimation based on an optical-flow scheme*, Experiments in Fluids, 40 (2006), pp. 80–97.

[9] T. Corpetti, E. Mémin, and P. Pérez, *Dense estimation of fluid flows*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (2002), pp. 365–380.

[10] C. Douglas, J. Hu, M. Kowarschik, U. Rüde, and C. Weiss, *Cache Optimization for Structured and Unstructured Grid Multigrid*, Electronic Transactions on Numerical Analysis (ETNA), 10 (2000), pp. 21–40.

[11] W. Enkelmann, *Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences*, Computer Vision, Graphics, and Image Processing, (1988), pp. 150–177.

[12] B. Fischer and J. Modersitzki, *Combining landmark and intensity driven registrations*, PAMM, 3 (2003), pp. 32–35.

[13] ——, *Curvature based image registration*, Journal of Mathematical Imaging and Vision, 18 (2003), pp. 81 – 85.

[14] I. Galic, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H. Seidel, *Towards PDE-based image compression*, in Proceedings of variational, geometric, and level set methods in computer vision, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, New York, 2005, pp. 37–48.

[15] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, *Recovering motion fields: An evaluation of eight optical flow algorithms*, in British Machine Vision Conference, 1998.

[16] F. Glazer, *Multilevel relaxation in low-level computer vision*, Multi-resolution image processing and Analysis, (1984), pp. 312–330.

[17] S. Gupta and J. Prince, *Stochastic models for div-curl optical flow methods*, IEEE Signal Processing Letters, 3 (1996), pp. 32–34.

[18] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, Heidelberg, New York, 1985.

[19] H. Haussecker and D. Fleet, *Computing optical flow with physical models of brightness variation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (2001), pp. 661–673.

[20] D. Heeger, *Model for the extraction of image flow*, Journal of the Optical Society of America A: Optics, Image Science, and Vision, 4 (1987), pp. 1455–1471.

[21] B. Horn, *Robot vision*, MIT Press,Cambridge, MA , USA, 1986.

[22] B. Horn and B. Schunck, *Determining optical flow*, Artificial Intelligence, (1981), pp. 185–203.

[23] F. Hülsemann, M. Kowarschik, M. Mohr, and U. Rüde, *Parallel geometric multigrid*, in Numerical Solution of Partial Differential Equations on Parallel Computers, A. Bruaset and A. Tveito, eds., vol. 51 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, Heidelberg, New York, 2005, ch. 5, pp. 165–208.

[24] E. Kalmoun, H. Köstler and U. Rüde, *3D optical flow computation using a parallel variational multigrid scheme with application to cardiac C-arm CT motion*, Image and Vision Computing, (2006). In press.

[25] E. Kalmoun and U. Rüde, *A variational multigrid for computing the optical flow*, Vision, Modeling and Visualization, (2003), pp. 577–584.

[26] T. Kohlberger, E. Mémin, and C. Schnörr, *Variational dense motion estimation using the helmholtz decomposition*, in 4th Int. Conf. on Scale Space Methods in Computer Vision, L. Griffin and M. Lillholm, eds., vol. 2695 of Lecture Notes in Computer Science, Isle of Skye, UK, 2003, pp. 432–448.

[27] B. McCane, K. Novins, D. Crannitch, and B. Galvin, *On benchmarking optical flow*, Computer Vision and Image Understanding, 84 (2001), pp. 126–143.

[28] J. Modersitzki, *Numerical methods for image registration*, Oxford University Press, Oxford, 2004.

[29] H.-H. Nagel and W. Enkelmann, *An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences*, IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-8 (1986), pp. 565–593.

[30] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, in Proceedings of the eleventh annual international conference of the Center for Nonlinear Studies on Experimental mathematics : Computational issues in nonlinear science, Amsterdam, The Netherlands, 1992, Elsevier North-Holland, Inc., pp. 259–268.

[31] D. Suter, *Motion estimation and vector splines*, in Proceedings of the Conference on Computer Vision and Pattern Recognition, Los Alamos, USA, 1994, pp. 939–948.

[32] D. Terzopoulos, *Image analysis using multigrid methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1986), pp. 129–139.

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

[34] A. Verri and T. Poggio, *Motion field and optical flow: Qualitative properties*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 490–498.

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

[36] J. Weickert and C. Schnörr, *Variational Optic Flow Computation with a Spatio-Temporal Smoothness Constraint*, Journal of Mathematical Imaging and Vision, 14 (2001), pp. 245–255.

[37] P. Wesseling, *Multigrid Methods*, Edwards, Philadelphia, PA, USA, 2004.

[38] R. Wienands and W. Joppich, *Practical Fourier Analysis for Multigrid Methods*, vol. 5 of Numerical Insights, Chapmann and Hall/CRC Press, Boca Raton, Florida, USA, 2005.

[39] Y. Zheng, H. Köstler, N. Thürey, and U. Rüde, *Enhanced Motion Blur Calculation with Optical Flow*, in Proceedings of Vision, Modeling and Visualization, RWTH Aachen, Germany, 2006, Aka GmbH, IOS Press, pp. 253–260.