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

Lehrstuhl für Informatik 10 (Systemsimulation)

A variational multigrid for computing the optical flow

El Mostafa Kalmoun and Ulrich Rüde

Lehrstuhlbericht 03-4

Abstract

Computing the optical flow for a sequence of images is currently a standard low-level problem in machine vision. A classical way to solve this problem is the Horn-Schunck algorithm. It corresponds to a coupled Gauss-Seidel relaxation for solving a system of two PDEs. The convergence of the algorithm is in general poor. A multigrid strategy can be expected to provide a significant acceleration. We investigate in this paper a V-cycle multigrid implementation based on the Galerkin approach. Experiments on synthetic and natural images show that the method provides a significant performance improvement.

Key words. Optical flow, Horn-Schunck algorithm, multigrid, Galerkin discretization.

1 Introduction

Optical flow is the motion of brightness patterns in time-varying images. It is our attempt to estimate the 2D-projection of the 3D-real world motion. Applications range from video compression to medical imaging. Differential approaches, which estimate velocity vectors from spatial and temporal intensity derivatives are the most common visited techniques in the literature. They are based on the assumption that the intensity variations are small and are only due to a movement in the image plane.

This constant brightness assumption leads to an ill-posed problem that can only be solved by imposing additional assumptions. A standard technique, which is due to to Horn and Schunck [1], is to require the flow field to be smooth by means of a standard regularization approach. This results in a system of elliptic PDEs of reaction diffusion type. The second order terms are induced by the regularization and become straightforward Laplace operators. The zero order terms are linear (but variable) and are computed from the derivatives of the image data. Since the image data (and even more so its derivatives) are usually nonsmooth, this poses some nontrivial problems. The PDEs are usually discretized by standard finite differences.

The Horn-Schuck algorithm is the standard way to discretize and solve the PDEs. In multigrid terminology, it is simply a coupled pointwise Gauss-Seidel relaxation. As expected, it has acceptable convergence only if the zero-order terms dominate but the performance is poor, when the diffusion character dominates. In this case, a multigrid strategy can be expected to provide a significant acceleration by allowing a quick propagation of information from non-zero flow field regions into homogeneous or untextured image areas. The Horn-Schunck algorithm can still be used as a smoother for such a multigrid method. The most difficult aspect is to deal with the strongly discontinuous coefficients in an efficient way.

We report in this note results from our implementation of a variational multigrid scheme for the Horn–Schunck algorithm in the following order. First, we recall in the next section the problem and state the optical flow constraint equation. In Section 3, we explain the need of additional constraints in order to make the problem well-posed. Thus, we briefly review the regularization technique used to solve the optical flow problem. Since we are interested in the quadratic regularization used by Horn and Schunck, we describe their method in Section 4. Then, we propose in Section 5 a bidirectional multigrid scheme for this algorithm based on the Galerkin approach. Finally, we present and discuss results of some experiments on synthetic and real images.

2 The optical flow constraint equation

Assume we are given a sequence of images with an intensity value $I(x, y, t)$ of the image point (x, y) at time t. In order to state the problem, the intensity values $(I(x, y, t))_{x,y,t}$ are supposed to be described by a differentiable function $I : \Omega \times [0, T] \to \mathbf{R}$, where $\Omega \subset \mathbf{R}^2$ is the image spatial domain and T is a strictly positive scalar. Let us denote also by I_x, I_y and I_t the partial derivatives of I in the direction of x, y and t, respectively. Most optical flow approaches are based on the assumption that Image objects keep the same intensity value under motion for at least a short period of time. In term of equations, this can be stated as follows: $\forall x, y \in \Omega, \forall t \in [0, T]$,

$$I(x, y, t) = I(x + dx, y + dy, t + dt).$$

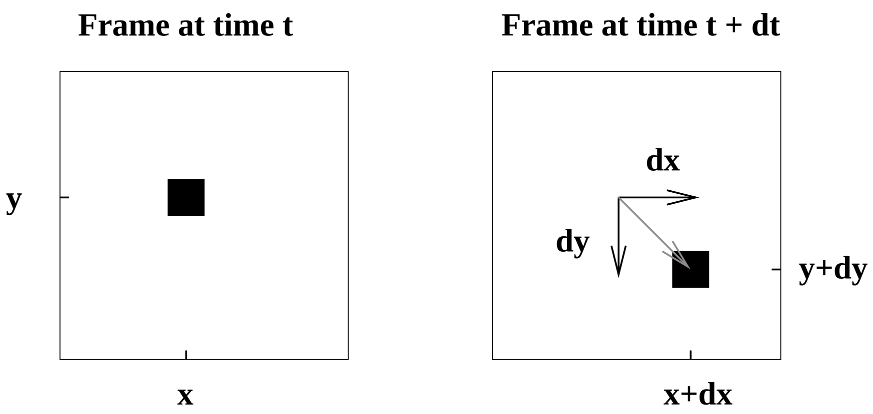

Figure 1: The optical flow at the pixel (x, y) is the 2D-velocity vector $(u, v) = (\frac{dx}{dt}, \frac{dy}{dt})$.

Using Taylor expansion and dropping the nonlinear terms, we get the optical flow constraint equation (OFCE):

$$I_x u + I_y v + I_t = 0.$$

Here $(u, v) = (\frac{dx}{dt}, \frac{dy}{dt})$ is called the optical flow vector. To explain this equation geometrically, we write it in the following equivalent forms:

$$\vec{\nabla} I.\vec{w} = -I_t \quad \Leftrightarrow \quad \vec{D} I.(\vec{w}, 1) = 0,$$

where $\vec{\nabla} I = (I_x, I_y)$, $\vec{D} I = (I_x, I_y, I_t)$ and $\vec{w} = (u, v)$. This means the solution set of the (OFCE) define a line in the $\vec{w}$-space which is perpendicular to the intensity spatial gradient $\vec{\nabla} I$, see Figure 2. Hence the problem is ill-posed since we can only calculate the normal component of the velocity $\vec{w}$ but not the tangent component (one equation and two unknowns). In optical flow terminology, this is commonly referred to as the aperture problem. To solve the problem, gradient-based methods adopt the classical way of forming a well-posed problem via the addition of global constraints. This is explained in the next section.

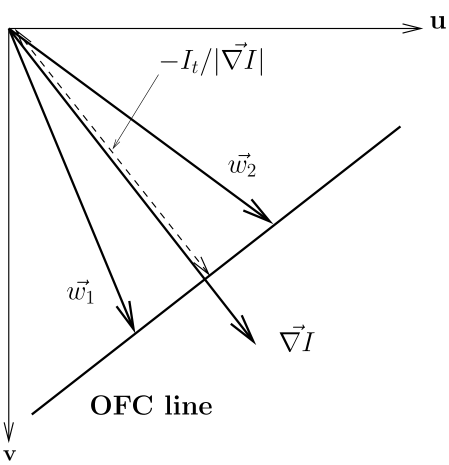

Figure 2: The aperture problem: the solutions of the OFCE define a line in the (u, v)-space (the image plan). The vectors $\vec{w}_1$ and $\vec{w}_2$ are possible solutions.

3 Regularization techniques

Regularization in the sense of Tikhonov [2] means to incorporate a priori information to the problem. In general, smoothness assumptions on the solution are imposed so that to guide an optimization procedure to a desired solution. During the 60s' and 70's, the mathematical theory was developed by Tikhonov's school [3]. The techniques have been then imported to computer vision for different low-level problems besides optical flow as shape-from-shading, surface reconstruction from stereo or motion analysis; see [4] for more details. In the case of our problem, the (OFCE) is replaced by the following minimization problem:

$$\min_{(u,v)} \int_x \int_y E(u,v)\,\mathrm{d}x\,\mathrm{d}y \tag{1}$$

where $E = E_d + \alpha E_r$. Here $E_d(u,v) = (I_x u + I_y v + I_t)^2$ is the data term and E_r is a regularization term. The parameter α is a positive scalar to trade-off the goodness of E_d with E_r. In general, it would be required to interactively adjust α to find the best value. The method for solving (1) will depend of course on the choice of E_r. Some previously used terms are the quadratic smoother by Horn and Schunck [1], oriented smoother by Nagel [5, 6] and anisotropic smoother by Weickert [7], among others; see [7] for a review on regularizers used in the optical flow problem. In this paper, we are only interested in the quadratic regularization method and for that we review the Horn–Schunck algorithm in the next section.

4 Horn–Schunck algorithm

Horn and Schunck [1] assumed as an additional constraint that the optical flow is varying smoothly in the sense that neighboring object points have almost the same velocity. This corresponds to a standard choice of E_r to be the following isotropic regularizer

$$E_r(u,v) = \frac{1}{2}\left(|\nabla u|^2 + |\nabla v|^2\right).$$

With the help of calculus of variations, we get from (1) a system of two elliptic PDEs:

$$\begin{aligned} \alpha\Delta u - I_x(I_x u + I_y v + I_t) &= 0 \\ \alpha\Delta v - I_y(I_x u + I_y v + I_t) &= 0. \end{aligned} \tag{2}$$

This coupled system is symetric in the two components of the velocity u and v. In order to capture the coupling effect between the two equations, one would preferably solve them simultaneously. Horn and Schunck suggested thus to use the following block Gauss-Seidel relaxation

$$\begin{aligned} u^{k+1} &= \bar{u}^k - I_x\,\frac{I_x\bar{u}^k + I_y\bar{v}^k + I_t}{\alpha + I_x^2 + I_y^2} \\ v^{k+1} &= \bar{v}^k - I_y\,\frac{I_x\bar{u}^k + I_y\bar{v}^k + I_t}{\alpha + I_x^2 + I_y^2}. \end{aligned}$$

Here $\bar{u}$ (resp. $\bar{v}$) denotes an average of the neighboring points to u (resp. v).

5 Multigrid scheme

Multigrid methods are known to be among the most efficient solution methods for elliptic PDEs. For a comprehensive overview on multigrid methods, we refer to [8] and [9]. The core idea of multigrid is to use a sequence of coarse grids as a means to accelerate the solution process (by relaxation such as Gauss-Seidel) on the finest grid. This leads to recursive algorithms like the so-called V- or W-cycle, wich traverse between fine and coarse grids in the mesh hierarchy. Since ultimately only a small number of relaxation steps on each level must be performed, multigrid provides an asymptotically optimal method whose complexity is only $O(N)$, where N is the number of mesh points.

The first attempts to use multilevel techniques for low-level problems in computer vision, and particularly for the optical flow are due to Glazer [10] and Terzopoulos [11]. They both reported improvement in performance but only by testing simple synthetic images. Enkelmann [12] adopted a coarse-to-fine strategy and used an image pyramid to subsample the images into different resolutions. The spatial intensity gradient information need not to be transferred to the current coarse level as in [10] since they could be calculated from the corresponding level of the image pyramid. Both methods (direct restriction of the image gradient and building an image pyramid) when used in a standard multigrid lead to similar results: acceptable convergence rate for *smooth* images and slow convergence or even divergence for highly-textured images, especially if we consider more than two levels in the multigrid hierarchy. Clearly, there is a loss of information when we represent the problem on coarse levels. Battiti *et al* [13] stated that a usual multigrid cycle is not appropriate due to a possible information conflict between different scales, and they preferred rather a one-way (coarse-to-fine) strategy. However, one-way multigrid methods - also known as cascadic multigrid, do not reach the optimal efficiency of a classical (bidirectional) multigrid method with a good coarse grid approximation of the original problem. A close look on system (2) shows that the finest grid operator has some nice properties that could be exploited for coarse grid approximation. Let us develop this point.

First, we write system (2) as follows

$$Lw = F \tag{3}$$

where

$$w = \begin{pmatrix} u \\ v \end{pmatrix} \quad , \quad F = \begin{pmatrix} -I_x I_t \\ -I_y I_t \end{pmatrix},$$

$$L = L_d + L_r \quad , \quad L_r = \begin{pmatrix} -\alpha\Delta & 0 \\ 0 & -\alpha\Delta \end{pmatrix},$$

and L_d is a 2x2 block diagonal matrix with entries

$$\begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

It is clear that the term L_r is symmetric positive definite and one can show that L_d is symmetric positive semi-definite. Hence the sum L is also symmetric positive definite. This suggests to put the linear system (3) into its equivalent variational minimization form

$$w = arg \min_{\Omega} a(z).$$

Here a is the quadratic form given by $a(z) = \frac{1}{2}(Lz, z) - (F, z)$.
Let's denote by h and H the current resolution and the next coarser resolution, respectively. Let also $I_H^h : \Omega^H \mapsto \Omega^h$ be a full rank linear mapping. An optimal coarse grid correction $I_H^h w_H$ of the current approximation w_h is characterized by

$$((I_H^h)^T L_h I_H^h) w_H = (I_H^h)^T (F - L_h w_h).$$

The Galerkin approach consists then to choose the coarse grid operator as follows

$$L_H = I_h^H L_h I_H^h \quad and \quad I_h^H = (I_H^h)^T.$$

For our system, this yields

$$\begin{aligned} L_H &= \begin{pmatrix} R & 0 \\ 0 & R \end{pmatrix} \begin{pmatrix} L_h^1 & L_h^2 \\ L_h^2 & L_h^3 \end{pmatrix} \begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix} \\ &= \begin{pmatrix} R L_h^1 P & R L_h^2 P \\ R L_h^2 P & R L_h^3 P \end{pmatrix} \end{aligned}$$

where R and P are any 2D-grid transfer operators satisfying $P = R^T$. In our implementation, R is the full weighting and P is the bilinear interpolation, see [9]. The components of our V-cycle are now as follows

- Vertex-centered grid and standard coarsening
- Coupled lexicographic point Gauss-Seidel smoother
- Full weighting and bilinear interpolation
- Galerkin coarse grid approximation (GCA approach)

6 Implementation issues

We discretize the Laplacien Δ by the standard 5 point stencil, which means that the averages $\bar{u}$ and $\bar{v}$ are computed from the stencil

$$\frac{1}{4}\begin{pmatrix} & 1 & \\ 1 & & 1 \\ & 1 & \end{pmatrix}.$$

As suggested by Horn and Schunck, we calculate spatial and temporal image derivatives halfway between pixels in the x, y, t directions. Precisely, we use the following masks

$$M_x = \tfrac{1}{4}\begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad M_y = \tfrac{1}{4}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \quad M_t = \tfrac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

and then the partial derivatives are calculated by

$$I_x = M_x * (I_1 + I_2) \quad I_y = M_y * (I_1 + I_2) \quad I_t = M_t * (I_2 - I_1).$$

The code is written in C language and the full scheme is described in Figure 3.

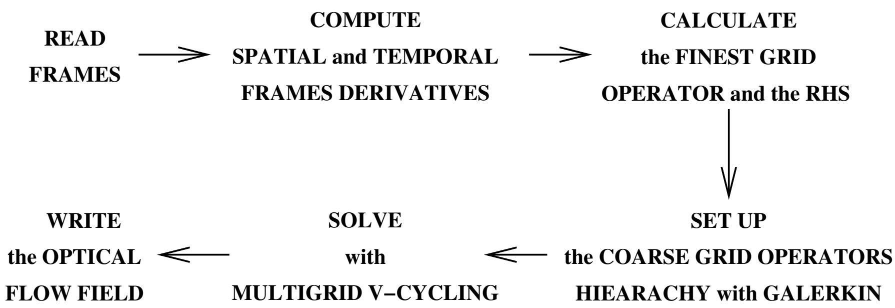

Figure 3: The general scheme used for computing the optical flow

7 Experimental results

7.1 A simple example

For comparison purposes, we have first implemented a multigrid V-cycle in a similar way to [10]. This means the use of the same discretization of the Laplace operator on all levels and restriction of the intensity gradient by injection. We refer to that as the DCA approach (Discretization coarse grid approximation).
We consider then a simple intensity function $I(x, y, t) = x + y + t$. One of the images from this sequence is shown in Figure 4. The corresponding system of PDE's now collapses simply to

$$\alpha\Delta u = u + v + 1 = 0$$
$$\alpha\Delta v = u + v + 1 = 0$$

For $\alpha = 1$ and an initial guess with $u_0 \neq v_0$, as shown in Table 1, multigrid is far superior to the unaccelerated Horn-Schnck algorithm. Note also the performance of DCA and GCA is comparable as could be expected for smooth coeffiecients.

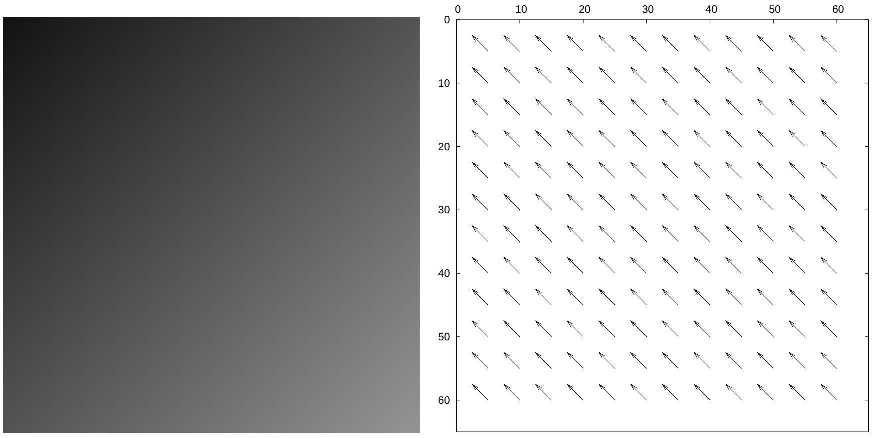

Figure 4: Left: First frame from the images sequence with intensity $I(x, y, t) = x + y + t$ and size 65x65. Right: Corresponding optical flow field computed with Horn-Schucnk method.

	Convergence rate	
	DCA	GCA
V(1,0)	0.370	0.356
V(1,1)	0.183	0.137
V(2,1)	0.116	0.070
V(3,3)	0.056	0.024
Horn-Schunck	0.998	

Table 1: Convergence rates for the Horn-Schunck algorithm and DCA and GCA nultigrid implemenatations with different numbers of presmoothing and postsomoothing iterations.

7.2 Synthetic and real images

To compare the performance of our multigrid implementation to the Horn-Schunck algorithm, we have tested two synthetic and one natural image sequences with different sizes.
The first synthetic image is a rotating sphere with a size equal to 129x129. The second is the "marbled block" image sequence with size 512x512. It was recorded by Otte and is available at `http://i21www.ira.uka.de/image_sequences`. It shows four columns and one moving marbled block. The last sequence is the Hamburg taxi sequence with size 256x190. The sequence contains a movement of four objects: a pedestrian near to the upper left corner and the three cars. Samples from these sequences and the estimated optical flows are shown in Figures 5-7. We point out that all tested images were presmoothed by a Gaussian filter before processing.
Results showing the convergence rates and CPU times for reaching similar solutions by Horn-Schunck algorithm and our variational multigrid by taking two consecutive frames from the three sequences are given in Table 7.2. Note also that the standard multigrid implementation (DCA approach) has failed to converge for all the three sequences when considering more than three levels.

8 Discussion and conclusion

We have proposed a variational multigrid for computing the optical flow using the Horn-Schunck model. Experiments conducted on synthetic and real images have demonstrated the robustness of the method as it speeds up the convergence considerably. As expected from a multigrid scheme, the number of iterations required to achieve a specified accuracy is independent of the fine-grid resolution. For the tested images, we need only a few V-cycle iterations (5 or 6) to compute a good-accuracy solution regardless of the size of the images. The convergence rate of the scheme

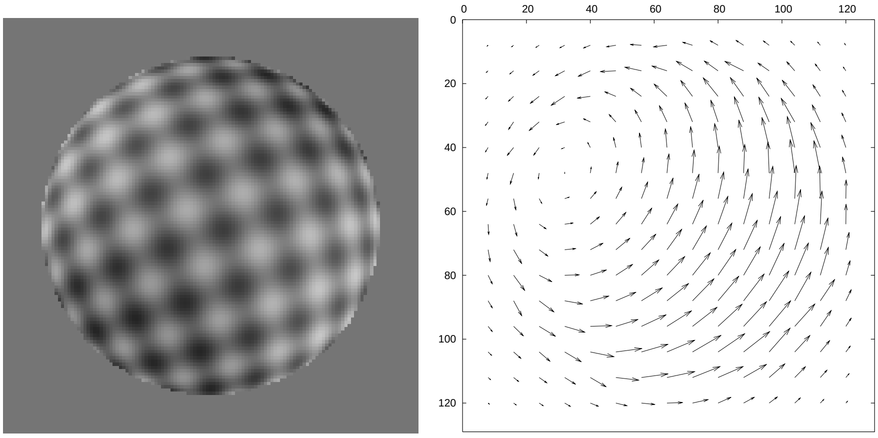

Figure 5: Left: A frame from the sphere sequence with size 65x65. Right: Estimated optical flow field.

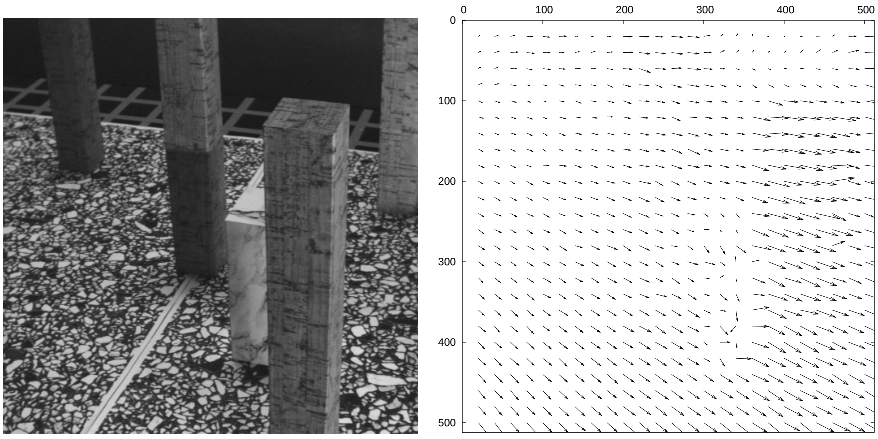

Figure 6: Left: A frame from the marbled block sequence with size 512x512. Right: Estimated optical flow field.

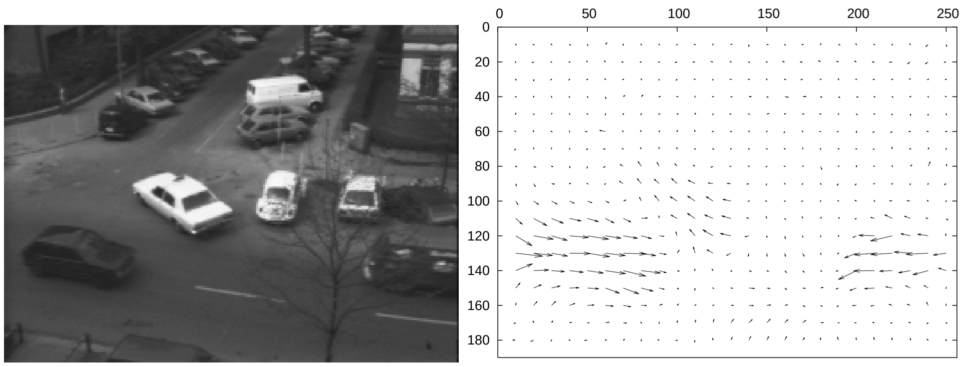

Figure 7: Left: A frame from the Hamburg taxi sequence with size 256x190. Right: Estimated optical flow field.

$\alpha = 5$ and $u_0 = v_0 = 0$ on P4 2.4GHz						
	Sphere		Marble		Taxi	
	ρ	CPU	ρ	CPU	ρ	CPU
Horn-Schunck	0.98	5.16	0.98	125	0.98	27.3
VMG V(2,1)	0.15	1.29	0.43	28.1	0.45	5.35

Table 2: The convergence rates and CPU times in seconds for reaching equivalent results between two consecutive frames from the three sequences by Horn-Schunck algorithm and our variational multigrid.

is thus independent of the resolution; nevertheless it depends on the input images, i.e. on the coefficients of the zero-order term of our system. For highly-textured images, the method provides good, but probably not yet optimal convergence rates. A possible improvement would be to use matrix dependent transfer grid operators. On the other hand, the Galerkin approach already leads to high memory costs. For our implementation, we have to store three 9-points stencils for each image point at different resolutions. To make the algorithm competitive, we need then to look for a better representation of the coarse grid operators. Future work is to consider also other convex regularizers which do not blur motion dicontinuities. We will address also possible applications in medical imaging.

References

[1] Horn B.K.P, Schunck B.G. Determining optical flow, *Artificial Intelligence* 1981, **17**:185-203.

[2] Tikhonov A.N. Solution of incorrectly formulated problems and the regularization method, *Soviet Math. Dokl.* 1963, **4**:1035-1038.

[3] Tikhonov A.N, Arsenin V.Y. *Solution of ill-posed problems*, Wiley:New York, 1977.

[4] Poggio T, Torre V, Koch C. Computational vision and regularization theory,*Nature* 1985, **317**:314-319.

[5] Nagel H.-H. Displacement vectors derived from second-order intensity variations in image sequences, *Computer Vision, Graphics, and Image Processing* 1983, **21**:85-117.

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

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

[8] Briggs W.L, Hensen V.E, McCormick S.F. *A multigrid tutorial* . SIAM, 2000.

[9] Trottenberg U, Oosterelee C, Schüller A. *Multigrid.* Academic Press, 2001.

[10] Glazer F. Multilevel relaxation in low-level computer vision, in *Multi-resolution image processing and Analysis (A. Rosenfeld, Ed.) Springer-Verlag* 1984, 312-330.

[11] Terzopoulos D. Image analysis using multigrid methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1986, **8**:129-139.

[12] Enkelmann W. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences, *Computer Vision, Graphics, and Image Processing* 1988, **43**:150-177.

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