Lecture 19:

Image Sequence Analysis IV: Numerical Methods

Contents

Basic Idea

- Optic flow methods based on the minimisation of energy functionals can be solved in the same manner as the variational image processing methods from Lecture 11.
- Two basic strategies:
  - **Elliptic Strategy:**
    Solve the linear or nonlinear systems of equations arising from the discretised Euler–Lagrange equations.
  - **Parabolic Strategy:**
    Use methods from diffusion filtering to solve a diffusion–reaction system that results from applying gradient descent.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Numerical Methods for the Parabolic Problem

Model Problem

Let us consider e.g. the 2-D isotropic flow-driven energy functional

$$E(u,v) = \int_\Omega \left((f_x u + f_y v + f_z)^2 + \alpha\, \Psi(|\nabla u|^2 + |\nabla v|^2)\right)\, dx\, dy$$

with some increasing regulariser $\Psi(s^2)$ that is convex in s (cf. Lecture 16).

Its minimiser (u,v) is given as the steady-state $(t \to \infty)$ of

$$\begin{aligned} u_t &= \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right)\, \nabla u\right) - \tfrac{1}{\alpha}\, f_x\, (f_x u + f_y v + f_z), \\ v_t &= \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right)\, \nabla v\right) - \tfrac{1}{\alpha}\, f_y\, (f_x u + f_y v + f_z). \end{aligned}$$

This diffusion–reaction system can be solved with methods from diffusion filtering:

- (modified) explicit schemes
- semi-implicit schemes
- AOS schemes

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Modified Explicit Scheme

An explicit discretisation with an implicitly stabilised data term is given by

$$\begin{aligned} \frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} &= A(\mathbf{u}^k, \mathbf{v}^k)\, \mathbf{u}^k - \tfrac{1}{\alpha}\, \mathbf{f}_x \left(\mathbf{f}_x \mathbf{u}^{k+1} + \mathbf{f}_y \mathbf{v}^k + \mathbf{f}_z\right), \\ \frac{\mathbf{v}^{k+1} - \mathbf{v}^k}{\tau} &= A(\mathbf{u}^k, \mathbf{v}^k)\, \mathbf{v}^k - \tfrac{1}{\alpha}\, \mathbf{f}_y \left(\mathbf{f}_x \mathbf{u}^k + \mathbf{f}_y \mathbf{v}^{k+1} + \mathbf{f}_z\right). \end{aligned}$$

where products of vectors are interpreted pointwise.
Can be solved explicitly for $\mathbf{u}^{k+1}$ and $\mathbf{v}^{k+1}$:

$$\begin{aligned} \mathbf{u}^{k+1} &= \frac{\mathbf{u}^k + \tau A(\mathbf{u}^k, \mathbf{v}^k)\, \mathbf{u}^k - \frac{\tau}{\alpha}\, \mathbf{f}_x\, (\mathbf{f}_y \mathbf{v}^k + \mathbf{f}_z)}{1 + \frac{\tau}{\alpha}\, \mathbf{f}_x^2} \\ \mathbf{v}^{k+1} &= \frac{\mathbf{v}^k + \tau A(\mathbf{u}^k, \mathbf{v}^k)\, \mathbf{v}^k - \frac{\tau}{\alpha}\, \mathbf{f}_y\, (\mathbf{f}_x \mathbf{u}^k + \mathbf{f}_z)}{1 + \frac{\tau}{\alpha}\, \mathbf{f}_y^2} \end{aligned}$$

Same stability restriction as for explicit diffusion scheme ($\tau \le \frac{1}{4}$ if $\Psi'(s^2) \le 1$).
Slow, but popular due to its simplicity.

Numerical Methods for the Parabolic Problem (3)

Semi-Implicit Scheme

A semi-implicit discretisation is given by

$$\begin{aligned}
\frac{\mathbf{u}^{k+1}-\mathbf{u}^{k}}{\tau} &= A(\mathbf{u}^{k},\mathbf{v}^{k})\,\mathbf{u}^{k+1} - \tfrac{1}{\alpha}\,\mathbf{f}_x\left(\mathbf{f}_x\mathbf{u}^{k+1}+\mathbf{f}_y\mathbf{v}^{k}+\mathbf{f}_z\right),\\
\frac{\mathbf{v}^{k+1}-\mathbf{v}^{k}}{\tau} &= A(\mathbf{u}^{k},\mathbf{v}^{k})\,\mathbf{v}^{k+1} - \tfrac{1}{\alpha}\,\mathbf{f}_y\left(\mathbf{f}_x\mathbf{u}^{k}+\mathbf{f}_y\mathbf{v}^{k+1}+\mathbf{f}_z\right).
\end{aligned}$$

Leads to the linear systems

$$\begin{aligned}
\left(I+\tfrac{\tau}{\alpha}\mathbf{f}_x^2-\tau A(\mathbf{u}^{k},\mathbf{v}^{k})\right)\,\mathbf{u}^{k+1} &= \mathbf{u}^{k}-\tfrac{\tau}{\alpha}\,\mathbf{f}_x\left(\mathbf{f}_y\mathbf{v}^{k}+\mathbf{f}_z\right),\\
\left(I+\tfrac{\tau}{\alpha}\mathbf{f}_y^2-\tau A(\mathbf{u}^{k},\mathbf{v}^{k})\right)\,\mathbf{v}^{k+1} &= \mathbf{v}^{k}-\tfrac{\tau}{\alpha}\,\mathbf{f}_y\left(\mathbf{f}_x\mathbf{u}^{k}+\mathbf{f}_z\right).
\end{aligned}$$

Requires iterative solvers such as Jacobi, Gauß-Seidel, SOR, conjugate gradient and multigrid methods; cf. Lecture 11.

Numerical Methods for the Parabolic Problem (4)

AOS Scheme

Formally, the semi-implicit scheme can be rewritten as

$$\begin{aligned}
\mathbf{u}^{k+1} &= \left(I-\frac{\alpha\tau}{\alpha+\tau\mathbf{f}_x^2}\sum_{l=1}^{2}A_l(\mathbf{u}^{k},\mathbf{v}^{k})\right)^{-1}\frac{\alpha\mathbf{u}^{k}-\tau\mathbf{f}_x(\mathbf{f}_y\mathbf{v}^{k}+\mathbf{f}_z)}{\alpha+\tau\mathbf{f}_x^2},\\
\mathbf{v}^{k+1} &= \left(I-\frac{\alpha\tau}{\alpha+\tau\mathbf{f}_y^2}\sum_{l=1}^{2}A_l(\mathbf{u}^{k},\mathbf{v}^{k})\right)^{-1}\frac{\alpha\mathbf{v}^{k}-\tau\mathbf{f}_y(\mathbf{f}_x\mathbf{u}^{k}+\mathbf{f}_z)}{\alpha+\tau\mathbf{f}_y^2}.
\end{aligned}$$

This implies the AOS variant

$$\begin{aligned}
\mathbf{u}^{k+1} &= \frac{1}{2}\sum_{l=1}^{2}\left(I-\frac{2\alpha\tau}{\alpha+\tau\mathbf{f}_x^2}A_l(\mathbf{u}^{k},\mathbf{v}^{k})\right)^{-1}\frac{\alpha\mathbf{u}^{k}-\tau\mathbf{f}_x(\mathbf{f}_y\mathbf{v}^{k}+\mathbf{f}_z)}{\alpha+\tau\mathbf{f}_x^2},\\
\mathbf{v}^{k+1} &= \frac{1}{2}\sum_{l=1}^{2}\left(I-\frac{2\alpha\tau}{\alpha+\tau\mathbf{f}_y^2}A_l(\mathbf{u}^{k},\mathbf{v}^{k})\right)^{-1}\frac{\alpha\mathbf{v}^{k}-\tau\mathbf{f}_y(\mathbf{f}_x\mathbf{u}^{k}+\mathbf{f}_z)}{\alpha+\tau\mathbf{f}_y^2}.
\end{aligned}$$

Time step size τ should be reduced for $k \to \infty$.

Numerical Methods for the Elliptic Problem

Basic Strategy

- We consider

$$E(u,v) = \int_\Omega \left((f_x u + f_y v + f_z)^2 + \alpha\, S(\nabla f, \nabla u, \nabla v)\right)\, dx\, dy$$

with two different prototypes of regularisers.

- A discretisation of the Euler-Lagrange equations leads to (linear or nonlinear) systems of equations.

- We will discuss the most advanced and most efficient solvers for these problems: multigrid methods.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Prototypes of Regularisers

- **Method of Nagel and Enkelmann (1986)**
  - image-driven anisotropic regulariser

$$S(\nabla f, \nabla u, \nabla v) = \nabla u^\top D(\nabla f) \nabla u + \nabla v^\top D(\nabla f) \nabla v$$

  - projection matrix $D(\nabla f)$ orthogonal to ∇f
  - smoothness imposed everywhere except across edges in the image data

- **TV-Based Regularisation Method (Cohen 1993, Schnörr 1994)**
  - flow-driven isotropic regulariser

$$S(\nabla f, \nabla u, \nabla v) = \sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}$$

  - TV instead of Tikhonov penalisation
  - allows discontinuity-preserving smoothing in the flow field

Numerical Methods for the Elliptic Problem (3)

Euler-Lagrange Equations

- **The Method of Nagel and Enkelmann**

$$\begin{aligned} 0 &= f_x^2\, u + f_x f_y\, v + f_x f_z \;-\; \alpha\, \mathcal{L}_{\mathrm{NE}}\, u \\ 0 &= f_x f_y\, u + f_y^2\, v + f_y f_z \;-\; \alpha\, \mathcal{L}_{\mathrm{NE}}\, v \end{aligned}$$

image-driven regularisation $\longrightarrow$ linear differential operator

$$\mathcal{L}_{\mathrm{NE}}\, w = \mathrm{div}\, \left(D(\nabla f)\, \nabla w \right)$$

- **The TV-Based Regularisation Method**

$$\begin{aligned} 0 &= f_x^2\, u + f_x f_y\, v + f_x f_z \;-\; (\alpha/2)\, \mathcal{L}_{\mathrm{TV}}(u,v) \\ 0 &= f_x f_y\, u + f_y^2\, v + f_y f_z \;-\; (\alpha/2)\, \mathcal{L}_{\mathrm{TV}}(v,u) \end{aligned}$$

flow-driven regularisation $\longrightarrow$ nonlinear differential operator

$$\mathcal{L}_{\mathrm{TV}}(w,\tilde{w}) = \mathrm{div}\, \left(D(\nabla w, \nabla \tilde{w})\, \nabla w \right), \qquad D(\nabla w, \nabla \tilde{w}) = \frac{1}{\sqrt{|\nabla w|^2 + |\nabla \tilde{w}|^2 + \epsilon^2}}\, I$$

Numerical Methods for the Elliptic Problem (4)

MIA 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Discretisation

- finite difference discretisation of the 2-D Euler-Lagrange equations using central difference approximations

- **linear** or **nonlinear** system of equations discretised on grid of **cell size** h:

$$A^h \boldsymbol{\eta}^h = \boldsymbol{\rho}^h \quad , \qquad\qquad A^h(\boldsymbol{\eta}^h) = \boldsymbol{\rho}^h$$

where $\boldsymbol{\eta} = (u_1, ..., u_N, v_1, ..., v_N)^\top$ contains the unknown flow values.

- **data term** $\longrightarrow$ only contribution to block main diagonals
smoothness term $\longrightarrow$ also contribution to block off-diagonals

$$A^h = \left(\begin{array}{ccccc|ccccc} d &&&&& d &&&& \\ & d &&&&& d &&& \\ && d &&&&& d && \\ &&& d &&&&& d & \\ &&&& d &&&&& d \\ \hline d &&&&& d &&&& \\ & d &&&&& d &&& \\ && d &&&&& d && \\ &&& d &&&&& d & \\ &&&& d &&&&& d \end{array} \right) + \left(\begin{array}{ccccc|ccccc} s & s && s &&&&&& \\ s & s & s && s &&&&& \\ & s & s & s &&&&&& \\ s && s & s & s &&&&& \\ & s && s & s &&&&& \\ \hline &&&&& s & s && s & \\ &&&&& s & s & s && s \\ &&&&&& s & s & s & \\ &&&&& s && s & s & s \\ &&&&&& s && s & s \end{array} \right)$$

Numerical Methods for the Elliptic Problem (5)

The Discretised Nagel-Enkelmann Approach

- linear differential operator $\longrightarrow$ large linear system of equations

$$
\begin{aligned}
0 &= (\mathbf{f}_x^2)_i^h\,\mathbf{u}_i^h + (\mathbf{f}_x\mathbf{f}_y)_i^h\,\mathbf{v}_i^h + (\mathbf{f}_x\mathbf{f}_z)_i^h - \alpha\,(L_{\text{NE}}^{\mathbf{h}}\,\mathbf{u}^h)_i \qquad (i=1,...,N)\\
0 &= (\mathbf{f}_x\mathbf{f}_y)_i^h\,\mathbf{u}_i^h + (\mathbf{f}_y^2)_i^h\,\mathbf{v}_i^h + (\mathbf{f}_y\mathbf{f}_z)_i^h - \alpha\,(L_{\text{NE}}^{\mathbf{h}}\,\mathbf{v}^h)_i \qquad (i=1,...,N)
\end{aligned}
$$

- **Coupling**
  - anisotropic neighbourhood coupling within $\mathbf{u}$ and $\mathbf{v}$ via L_{NE}
  - point coupling of $\mathbf{u}$ and $\mathbf{v}$ via data term
- **Specific Basic Solver**
  - linear Gauß-Seidel solver with alternating line relaxation (ALR):
    - $*$ updates whole lines of unknowns simultaneously by solving tridiagonal systems
    - $*$ afterwards coupled point relaxation:
      simultaneous update of $\mathbf{u}$ and $\mathbf{v}$ (2×2 systems)

Numerical Methods for the Elliptic Problem (6)

The Discretised TV Approach

- nonlinear differential operator $\longrightarrow$ large nonlinear system of equations

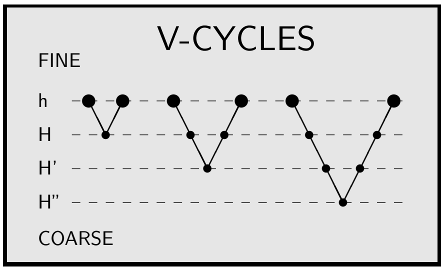
$$
\begin{aligned}
0 &= (\mathbf{f}_x^2)_i^h\,\mathbf{u}_i^h + (\mathbf{f}_x\mathbf{f}_y)_i^h\,\mathbf{v}_i^h + (\mathbf{f}_x\mathbf{f}_z)_i^h - (\alpha/2)\,(L_{\text{TV}}^{\mathbf{h}}\,(\mathbf{u}^h,\mathbf{v}^h))_i \qquad (i=1,...,N)\\
0 &= (\mathbf{f}_x\mathbf{f}_y)_i^h\,\mathbf{u}_i^h + (\mathbf{f}_y^2)_i^h\,\mathbf{v}_i^h + (\mathbf{f}_y\mathbf{f}_z)_i^h - (\alpha/2)\,(L_{\text{TV}}^{\mathbf{h}}\,(\mathbf{v}^h,\mathbf{u}^h))_i \qquad (i=1,...,N)
\end{aligned}
$$

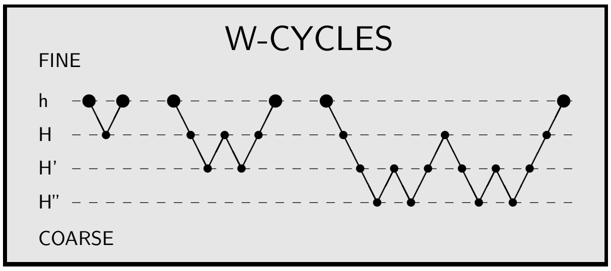
- **Coupling**
  - isotropic neighbourhood coupling within $\mathbf{u}$ and $\mathbf{v}$ via $L_{\text{TV}}(\mathbf{u},\mathbf{v})$
  - point coupling of $\mathbf{u}$ and $\mathbf{v}$ via joint diffusivity in $L_{\text{TV}}(\mathbf{u},\mathbf{v})$
  - point coupling of $\mathbf{u}$ and $\mathbf{v}$ via data term
- **Specific Basic Solver**
  - nonlinear Gauß-Seidel solver with coupled point relaxation (CPR):
    updates $\mathbf{u}$ and $\mathbf{v}$ at each point simultaneously by solving 2×2 systems

Multigrid Methods

Correcting Multigrid

- hierarchical application of a two–grid cycle with a basic solver (e.g. Gauß–Seidel)
- either **one** or **two** recursive calls per level (→ V–cycle, W–cycle)
- example of V– and W–cycles for two, three and four levels:



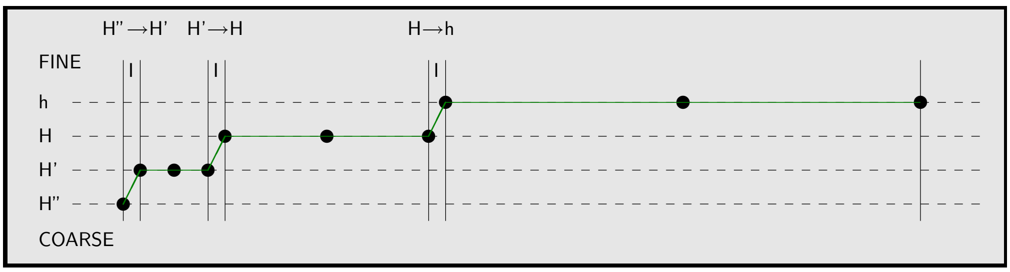


How can we further improve efficiency ?

- start with better initialisation
- embed V– and W–cycles in hierarchical initialisation strategy

Full Multigrid

- **Hierarchical Initialisation: Coarse-to-Fine Approach**
  - start with coarse version of original problem
  - refine problem step by step
  - use coarse solution as initial guess on next finer grid
- **At Each Level: Correcting Multigrid Solver**
  - coarse grid corrections → error explicitly computable (linear case)
    → via full approximation scheme (nonlinear case)
    (FAS, Brandt 1977)
- **Example**



Full Multigrid

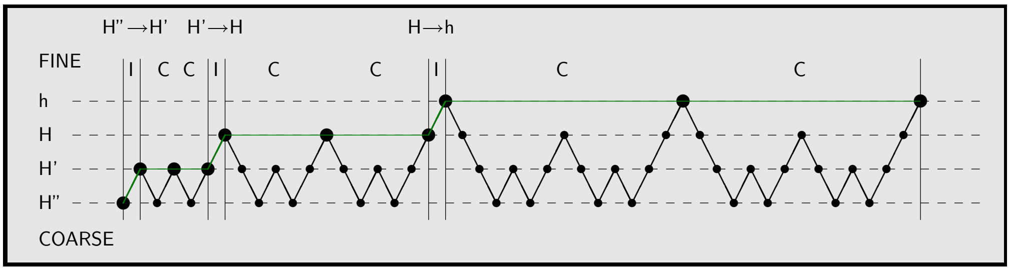
Hierarchical Initialisation: Coarse-to-Fine Approach

- start with coarse version of original problem
- refine problem step by step
- use coarse solution as initial guess on next finer grid

At Each Level: Correcting Multigrid Solver

- coarse grid corrections → error explicitly computable (linear case)
  → via full approximation scheme (nonlinear case)
  (FAS, Brandt 1977)

Example



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Implementation Details

The Method of Nagel and Enkelmann

- full Multigrid strategy with 4 W(1,1)-cycles each level
- Gauß-Seidel solver with alternating line relaxation (ALR)
- restriction by averaging, prolongation by integrated linear interpolation
- discretisation coarse grid approximation (DCA)

The TV-Based Regularisation Method

- full Multigrid strategy with 2 FAS W(2,2)-cycles each level
- Gauß-Seidel solver with coupled point relaxation (CPR)
- restriction by averaging, prolongation by integrated linear interpolation
- discretisation coarse grid approximation (DCA)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Numerical Methods for the Elliptic Problem (11)

Results

- **Testbed**
  - C implementation on standard desktop PC (3.06 GHz Pentium4)
  - stopping criterion : relative error $e_{\mathbf{rel}} := \|\tilde{\boldsymbol{\eta}} - \boldsymbol{\eta}\|_2 / \|\boldsymbol{\eta}\|_2$ of 10^{-2}
  - run times refer to the computation of one flow field of size 160×120

- **Image-driven anisotropic regularisation (Nagel-Enkelmann)**

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

- **Flow-driven isotropic regularisation (TV)**

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

Numerical Methods for the Elliptic Problem (12)

Extensions

- **High Accuracy Optic Flow Technique of Brox et al. 2004 (Lecture 18)**
  - combined grey value and gradient constancy
  - constancy assumptions without linearisation
  - robust data term $\Psi_D(s^2) = \sqrt{s^2 + \epsilon^2}$
  - isotropic flow-driven regularisation $\Psi_S(s^2) = \sqrt{s^2 + \epsilon^2}$
  - warping

- **Associated Energy Functional**

$$E(\mathbf{u}) = \int_\Omega \Psi_D\big(|f(\mathbf{x}+\mathbf{u}) - f(\mathbf{x})|^2 + \gamma|\nabla f(\mathbf{x}+\mathbf{u}) - \nabla f(\mathbf{x})|^2\big)\, d\mathbf{x} + \alpha \int_\Omega \Psi_S\big(|\nabla u_1|^2 + |\nabla u_2|^2\big)\, d\mathbf{x}$$

- multigrid strategy at each level

- similar performance gains

Realtime Demo Optical Flow

- **Realtime Computation with Webcam (160×120)**

Start

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Summary

- two main approaches for the numerical solution of variational optic flow methods: parabolic and elliptic strategies
- Parabolic strategies solve diffusion–reaction systems arising from gradient descent. They may apply methods from diffusion filtering (explicit, semi-implicit, AOS).
- Elliptic approaches discretise the Euler–Lagrange equations. They lead to large (linear or nonlinear) systems of equations.
- Often solvers for elliptic approaches are numerically more efficient.
- Suitable multigrid methods for the elliptic PDEs allow speedups of two to three orders of magnitude compared to standard solvers.
- real-time performance for small image sizes (120×160 pixels)
- High quality and real-time performance are not contradictive.
- opens door to a number of application areas (robotics, driver assistance systems, video coding etc.)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

References

- A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, C. Schnörr: Variational optic flow computation in real-time. *IEEE Transactions on Image Processing*, Vol. 14, 608–615, 2005.
  (`http://www.mia.uni-saarland.de/Publications/bruhn_pp89.pdf`).
  (multigrid for the linear CLG method)

- A. Bruhn, J. Weickert: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. *Proc. Tenth IEEE International Conference on Computer Vision* (Beijing, China, Oct. 2005), IEEE Computer Society Press, Vol. 1, 749-755.
  (`http://www.mia.uni-saarland.de/bruhn/ieee/ieee_bruhn_iccv05.html`).
  (multigrid for the nonlinear method of Brox et al. 2004)

- A. Bruhn, J. Weickert, T. Kohlberger, C. Schnörr: A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, Vol. 70, No. 3, 257–277, Dec. 2006.
  (`http://www.mia.uni-saarland.de/Publications/bruhn_pp136.pdf`).
  (multigrid for nonlinear methods in general)