Lecture 7:
Optic Flow V
High Accuracy Methods and Advanced Numerics

Contents

M I A
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31 32
33 34
35 36
37 38
39

The Method of Brox et al.

How Can We Design a Method that is Robust and Accurate?

- *Idea:* Combine most of the previously introduced concepts
- *Example:* The method of Brox et al. minimises the energy funcional
  (Brox/Bruhn/Papenberg/Weickert 2004)

$$E(\mathbf{w}) = \int_{\Omega\times T} \underbrace{\Psi_{\mathrm{D}}\Big(\big(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w})\big)^2+\gamma\,\big|\nabla f(\mathbf{x})-\nabla f(\mathbf{x}+\mathbf{w})\big|^2\Big)}_{\text{data term}}\,d\mathbf{x}$$
$$+\,\alpha\int_{\Omega\times T} \underbrace{\Psi_{\mathrm{S}}\Big(|\nabla_3 u|^2+|\nabla_3 v|^2\Big)}_{\text{smoothness term}}\,d\mathbf{x}$$

where $\mathbf{x}=(x,y,t)$ and $\Psi_{\mathrm{D}}(s^2)=\Psi_{\mathrm{S}}(s^2)=\sqrt{s^2+\epsilon^2}$ with $\epsilon=10^{-3}$.

- robust data term
- grey value and gradient constancy
- no linearisation
- spatiotemporal smoothness term
- flow-driven isotropic smoothness

The Method of Brox et al. (2)

Advantages and Shortcomings of the Method of Brox et al.

- Advantages
  - robust under varying illumination (gradient constancy)
  - can handle large displacements (no linearisation)
  - robust under noise and outliers (robust data term)
  - preserves motion discontinuities (flow-driven isotropic regulariser)
  - considers more than two frames (spatiotemporal smoothness term)

- Drawbacks
  - difficult to minimise (nonconvex energy functional $\longrightarrow$ requires warping)
  - computationally expensive (series of non-linear equation systems)

How does the minimisation work for such a complex energy functional?

The Method of Brox et al. (3)

Minimisation of a Nonconvex and Nonquadratic Functional

- The Euler-Lagrange equations of the method of Brox et al. are given by

$$
\begin{aligned}
0 \;=\;& \Psi_{\mathrm{D}}'\Big(\big(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w})\big)^2+\gamma\,\big|\nabla f(\mathbf{x})-\nabla f(\mathbf{x}+\mathbf{w})\big|^2\Big)\\
&\times\Big(f_x(\mathbf{x}+\mathbf{w})\,\big(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w})\big)+\gamma\,f_{xx}(\mathbf{x}+\mathbf{w})\,\big(f_x(\mathbf{x})-f_x(\mathbf{x}+\mathbf{w})\big)+\gamma\,f_{yx}(\mathbf{x}+\mathbf{w})\,\big(f_y(\mathbf{x})-f_y(\mathbf{x}+\mathbf{w})\big)\Big)\\
&+\alpha\,\mathrm{div}\Big(\Psi_{\mathrm{S}}'\big(|\nabla_3 u|^2+|\nabla_3 v|^2\big)\,\nabla_3 u\Big)\\
0 \;=\;& \Psi_{\mathrm{D}}'\Big(\big(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w})\big)^2+\gamma\,\big|\nabla f(\mathbf{x})-\nabla f(\mathbf{x}+\mathbf{w})\big|^2\Big)\\
&\times\Big(f_y(\mathbf{x}+\mathbf{w})\,\big(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w})\big)+\gamma\,f_{xy}(\mathbf{x}+\mathbf{w})\,\big(f_x(\mathbf{x})-f_x(\mathbf{x}+\mathbf{w})\big)+\gamma\,f_{yy}(\mathbf{x}+\mathbf{w})\,\big(f_y(\mathbf{x})-f_y(\mathbf{x}+\mathbf{w})\big)\Big)\\
&+\alpha\,\mathrm{div}\Big(\Psi_{\mathrm{S}}'\big(|\nabla_3 u|^2+|\nabla_3 v|^2\big)\,\nabla_3 v\Big)
\end{aligned}
$$

  with (reflecting) Neumann boundary conditions $\mathbf{n}^\top\nabla_3\, u=0$ and $\mathbf{n}^\top\nabla_3\, v=0$.

- *New:* **Nonlinear** equations with **non-unique** solution. In order to solve these equations we have to combine known concepts for both cases:
  - nonconvex optimisation via **warping** (series of convex problems)
  - nonlinear optimisation via **lagged nonlinearity** (series of linear problems)

The Method of Brox et al. (4)

Minimisation of a Nonconvex and Nonquadratic Functional

- *Step 1:* We introduce a fixed point iteration that is semi-implicit in the data term and fully implicit in the smoothness term (cf. Lecture 5). Thus we obtain

$$\begin{aligned}
0 &= \Psi_{\mathrm{D}}'\Big((f(\mathbf{x})-f(\mathbf{x}+\mathbf{w}^{k+1}))^2+\gamma\,|\nabla f(\mathbf{x})-\nabla f(\mathbf{x}+\mathbf{w}^{k+1})|^2\Big)\\
&\quad\times\Big(f_x(\mathbf{x}+\mathbf{w}^k)\,(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w}^{k+1}))+\gamma\,f_{xx}(\mathbf{x}+\mathbf{w}^k)\,(f_x(\mathbf{x})-f_x(\mathbf{x}+\mathbf{w}^{k+1}))+\gamma\,f_{yx}(\mathbf{x}+\mathbf{w}^k)\,(f_y(\mathbf{x})-f_y(\mathbf{x}+\mathbf{w}^{k+1}))\Big)\\
&\quad+\alpha\,\mathrm{div}\,\Big(\Psi_{\mathrm{S}}'(|\nabla_3 u^{k+1}|^2+|\nabla_3 v^{k+1}|^2)\,\nabla_3 u^{k+1}\Big)\\
0 &= \Psi_{\mathrm{D}}'\Big((f(\mathbf{x})-f(\mathbf{x}+\mathbf{w}^{k+1}))^2+\gamma\,|\nabla f(\mathbf{x})-\nabla f(\mathbf{x}+\mathbf{w}^{k+1})|^2\Big)\\
&\quad\times\Big(f_y(\mathbf{x}+\mathbf{w}^k)\,(f(\mathbf{x})-f(\mathbf{x}+\mathbf{w}^{k+1}))+\gamma\,f_{xy}(\mathbf{x}+\mathbf{w}^k)\,(f_x(\mathbf{x})-f_x(\mathbf{x}+\mathbf{w}^{k+1}))+\gamma\,f_{yy}(\mathbf{x}+\mathbf{w}^k)\,(f_y(\mathbf{x})-f_y(\mathbf{x}+\mathbf{w}^{k+1}))\Big)\\
&\quad+\alpha\,\mathrm{div}\,\Big(\Psi_{\mathrm{S}}'(|\nabla_3 u^{k+1}|^2+|\nabla_3 v^{k+1}|^2)\,\nabla_3 v^{k+1}\Big)
\end{aligned}$$

- *Step 2a+2b:* We introduce an incremental computation and linearise all expressions in the data term that involve $\mathbf{w}^{k+1}$ (here $f_* \in \{f, f_x, f_y\}$)

$$\begin{aligned}
f_*(\mathbf{x}+\mathbf{w}^{k+1}) &= f_*(x+u^k+du^k, y+v^k+dv^k, t+1)\\
&= f_*(x+u^k, y+v^k, t+1)+f_{*x}(x+u^k, y+v^k, t+1)\,du^k+f_{*y}(x+u^k, y+v^k, t+1)\,dv^k\\
&= f_*(\mathbf{x}+\mathbf{w}^k)+f_{*x}(\mathbf{x}+\mathbf{w}^k)\,du^k+f_{*y}(\mathbf{x}+\mathbf{w}^k)\,dv^k
\end{aligned}$$

The Method of Brox et al. (5)

Minimisation of a Nonconvex and Nonquadratic Functional

- *Step 2c:* Using the **compensated motion tensor** for representing the grey value and the gradient constancy assumptions given by

$$J^k=\underbrace{\nabla_3 f(\mathbf{x}+\mathbf{w}^k)\,\nabla_3 f(\mathbf{x}+\mathbf{w}^k)^\top}_{\text{grey value constancy}}+\gamma\,\underbrace{\Big(\nabla_3 f_x(\mathbf{x}+\mathbf{w}^k)\,\nabla_3 f_x(\mathbf{x}+\mathbf{w}^k)^\top+\nabla_3 f_y(\mathbf{x}+\mathbf{w}^k)\,\nabla_3 f_y(\mathbf{x}+\mathbf{w}^k)^\top\Big)}_{\text{gradient constancy}}$$

and defining $\mathbf{dw^k}=(du^k, dv^k, 1)^\top$ we obtain the following **partly linearised** fixed point iteration step

$$\begin{aligned}
0 &= \Psi_{\mathrm{D}}'\Big(\mathbf{dw}^{\mathbf{k}\top}J^k\,\mathbf{dw^k}\Big)\Big(J_{11}^k\,du^k+J_{12}^k\,dv^k+J_{13}^k\Big)-\alpha\,\mathrm{div}\,\Big(\Psi_{\mathrm{S}}'(|\nabla_3(u^k+du^k)|^2+|\nabla_3(v^k+dv^k)|^2)\nabla_3(u^k+du^k)\Big)\\
0 &= \Psi_{\mathrm{D}}'\Big(\mathbf{dw}^{\mathbf{k}\top}J^k\,\mathbf{dw^k}\Big)\Big(J_{12}^k\,du^k+J_{22}^k\,dv^k+J_{23}^k\Big)-\alpha\,\mathrm{div}\,\Big(\Psi_{\mathrm{S}}'(|\nabla_3(u^k+du^k)|^2+|\nabla_3(v^k+dv^k)|^2)\nabla_3(v^k+dv^k)\Big)
\end{aligned}$$

- *New:* These equations that have to be solved at each fixed point step are still **nonlinear**. However, due to the linearisation they have now a **unique** solution!
- *Step 3*: Finally, we embed this fixed point iteration into a coarse-to-fine strategy to avoid local minima (cf. Lecture 5).

The Method of Brox et al. (6)

Minimisation of a Nonconvex and Nonquadratic Functional

- *Discretisation:* One obtains the following **nonlinear system of equations**

$$
\begin{aligned}
0 &= [\Psi'_{\mathrm{D}}]^k_{i,j,z} \left([J^k_{11}]_{i,j,z}\, du^k_{i,j,z} + [J^k_{12}]_{i,j,z}\, dv^k_{i,j,z} + [J^k_{13}]_{i,j,z} \right) \\
&\quad -\alpha \sum_{l \in x,y,t} \sum_{\mathcal{N}_l(i,j,z)} \frac{[\Psi'_{\mathrm{S}}]^k_{\tilde{i},\tilde{j},\tilde{z}} + [\Psi'_{\mathrm{S}}]^k_{i,j,z}}{2} \left(\frac{u^k_{\tilde{i},\tilde{j},\tilde{z}} + du^k_{\tilde{i},\tilde{j},\tilde{z}} - u^k_{i,j,z} - du^k_{i,j,z}}{h_l^2} \right) \\
0 &= [\Psi'_{\mathrm{D}}]^k_{i,j,z} \left([J^k_{12}]_{i,j,z}\, du^k_{i,j,z} + [J^k_{22}]_{i,j,z}\, dv^k_{i,j,z} + [J^k_{23}]_{i,j,z} \right) \\
&\quad -\alpha \sum_{l \in x,y,t} \sum_{\mathcal{N}_l(i,j,z)} \frac{[\Psi'_{\mathrm{S}}]^k_{\tilde{i},\tilde{j},\tilde{z}} + [\Psi'_{\mathrm{S}}]^k_{i,j,z}}{2} \left(\frac{v^k_{\tilde{i},\tilde{j},\tilde{z}} + dv^k_{\tilde{i},\tilde{j},\tilde{z}} - v^k_{i,j,z} - dv^k_{i,j,z}}{h_l^2} \right)
\end{aligned}
$$

for $i = 1, ..., N$, $j = 1, ..., M$ and $z = 1, ..., Z$. The nonlinear expressions read

$$
[\Psi'_{\mathrm{D}}]^k_{i,j,z} = \Psi'_{\mathrm{D}}(\mathbf{dw^k}^{\top}_{i,j,z}\, J^k_{i,j,z}\, \mathbf{dw^k}_{i,j,z}) = \Psi'_{\mathrm{D}} \left(\begin{pmatrix} du^k_{i,j,z} \\ dv^k_{i,j,z} \\ 1 \end{pmatrix}^{\top} \begin{pmatrix} [J^k_{11}]_{i,j,z} & [J^k_{12}]_{i,j,z} & [J^k_{13}]_{i,j,z} \\ [J^k_{12}]_{i,j,z} & [J^k_{22}]_{i,j,z} & [J^k_{23}]_{i,j,z} \\ [J^k_{13}]_{i,j,z} & [J^k_{23}]_{i,j,z} & [J^k_{33}]_{i,j,z} \end{pmatrix} \begin{pmatrix} du^k_{i,j,z} \\ dv^k_{i,j,z} \\ 1 \end{pmatrix} \right)
$$

$$
[\Psi'_{\mathrm{S}}]^k_{i,j,z} = \Psi'_{\mathrm{S}}(|\nabla_3 u|^2_{i,j,z} + |\nabla_3 v|^2_{i,j,z}) = \Psi'_{\mathrm{S}}([u_x]^2_{i,j,z} + [u_y]^2_{i,j,z} + [u_t]^2_{i,j,z} + [v_x]^2_{i,j,z} + [v_y]^2_{i,j,z} + [v_t]^2_{i,j,z}) \,.
$$

The Method of Brox et al. (7)

Solving the Nonlinear System

- *Idea:* Solve nonlinear system as series of linear systems (cf. Lecture 5)
- *Example:* Fixed point iteration where the nonlinear expressions of data and smoothness term are evaluated at the **old time step**. This yields the system

$$
\begin{aligned}
0 &= [\Psi'_{\mathrm{D}}]^{k,l}_{i,j,z} \left([J^k_{11}]_{i,j,z}\, du^{k,l+1}_{i,j,z} + [J^k_{12}]_{i,j,z}\, dv^{k,l+1}_{i,j,z} + [J^k_{13}]_{i,j,z} \right) \\
&\quad -\alpha \sum_{l \in x,y,t} \sum_{\mathcal{N}_l(i,j,z)} \frac{[\Psi'_{\mathrm{S}}]^{k,l}_{\tilde{i},\tilde{j},\tilde{z}} + [\Psi'_{\mathrm{S}}]^{k,l}_{i,j,z}}{2} \left(\frac{u^k_{\tilde{i},\tilde{j},\tilde{z}} + du^{k,l+1}_{\tilde{i},\tilde{j},\tilde{z}} - u^k_{i,j,z} - du^{k,l+1}_{i,j,z}}{h_l^2} \right) \\
0 &= [\Psi'_{\mathrm{D}}]^{k,l}_{i,j,z} \left([J^k_{12}]_{i,j,z}\, du^{k,l+1}_{i,j,z} + [J^k_{22}]_{i,j,z}\, dv^{k,l+1}_{i,j,z} + [J^k_{23}]_{i,j,z} \right) \\
&\quad -\alpha \sum_{l \in x,y,t} \sum_{\mathcal{N}_l(i,j,z)} \frac{[\Psi'_{\mathrm{S}}]^{k,l}_{\tilde{i},\tilde{j},\tilde{z}} + [\Psi'_{\mathrm{S}}]^{k,l}_{i,j,z}}{2} \left(\frac{v^k_{\tilde{i},\tilde{j},\tilde{z}} + dv^{k,l+1}_{\tilde{i},\tilde{j},\tilde{z}} - v^k_{i,j,z} - dv^{k,l+1}_{i,j,z}}{h_l^2} \right)
\end{aligned}
$$

for $i = 1, ..., N$, $j = 1, ..., M$ and $z = 1, ..., Z$ at each fixed point iteration.

- *Properties:* Same as in case of robust data or flow-driven smoothness terms only (positive definiteness, convergence even for non-exact solutions)

The Method of Brox et al. (8)

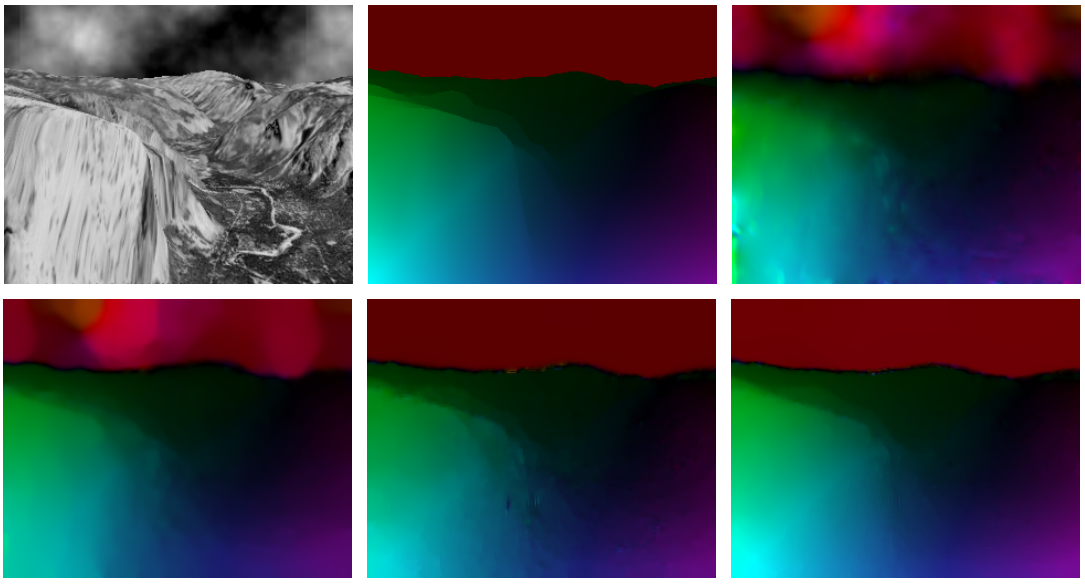
Summary of the Optimisation

- Two nested fixed point iterations
  - *Outer Iteration:* **Coarse-to-Fine Warping**
    approximates nonlinear nonconvex problem as series of nonlinear convex problems (via fixed point iteration and linearisation in the numerics)
  - *Inner Iteration:* **Lagged Nonlinearity Method**
    solves nonlinear convex problem as series of linear convex problems (by keeping the nonlinear expression in the data and smoothness term fixed)
- Additional iteration loop if linear systems are solved iteratively
  - *Solver Iteration:* **e.g. Gauß-Seidel Method**
    solves linear system of equations by iteratively approximating the inverse

Does this effort pay off in terms of accuracy?

The Method of Brox et al. (9)

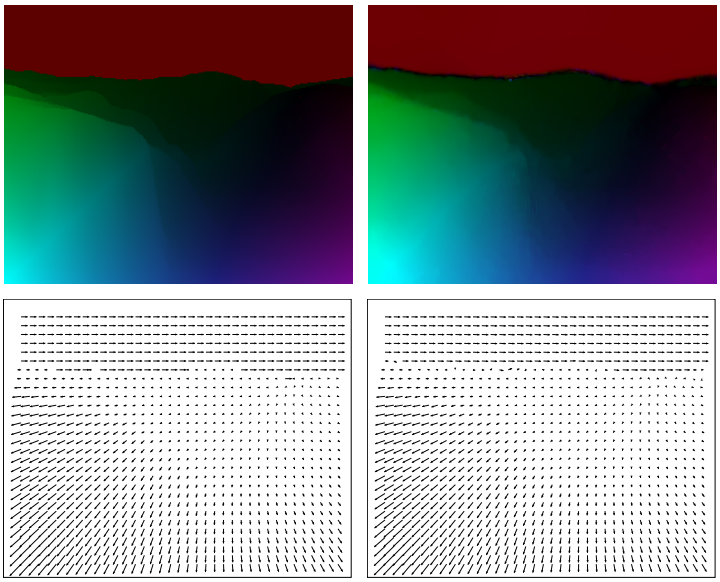
Results for the Brox et al. Method



Results for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Frame 8. **(b) Upper Center:** Ground truth. **(c) Upper Right:** Horn and Schunck **(d) Lower Left:** Robust data and flow-driven smoothness term **(e) Lower Center:** Brox *et al.* 2-D **(f) Lower Right:** Brox *et al.* 3-D

The Method of Brox et al. (10)

Ground Truth vs. Brox et al.



Ground truth vs. Brox *et al.* for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Ground truth (colour plot). **(b) Upper Right:** Brox *et al.* 3-D (colour plot). **(c) Lower Left:** Ground truth (vector plot). **(d) Lower Right:** Brox *et al.* 3-D (vector plot).

The Method of Brox et al. (11)

Comparison to the Literature

- Qualitative Evaluation for the Yosemite Sequence with Clouds

Technique	AAE
Horn/Schunck, orig.	31.69°
Anandan	13.36°
Singh, step 2	10.44°
Nagel	10.22°
Horn/Schunck, mod.	9.78°
Uras *et al.*	8.94°
Homogeneous	**7.12°**
Liu *et al.*	6.85°
Image-driven isotropic	**6.44°**
Flow-driven anisotropic	**6.42°**
Flow-driven isotropic	**6.32°**
Image-driven anisotropic	**6.28°**
Bruhn *et al.* (2-D, SD)	**5.74°**
Alvarez *et al.*	5.53°
Bruhn *et al.* (3-D, SD)	**5.18°**

Technique	AAE
Farnebäck	4.84°
Mémin/Pérez	4.69°
Bruhn *et al.* (3-D, LD)	**4.17°**
Wu *et al.*	3.54°
Brox *et al.* (2-D, SD)	**3.50°**
Brox *et al.* (3-D, SD)	**2.78°**
Teng *et al.*	2.70°
Bruhn/Weickert (2-D, LD)	**2.42°**
Bab Hadiashar/Suter	2.05°
Amiaz/Kiryati	2.04°
Brox *et al.* (3-D, LD)	**1.78°**
Amiaz/Kiryati	1.73°
Bruhn/Weickert (3-D, LD)	**1.72°**
Amiaz *et al.*	1.42°
Brox/Bruhn/Weickert	1.22°

The Method of Brox et al. (12)

Results for Real-World Sequences

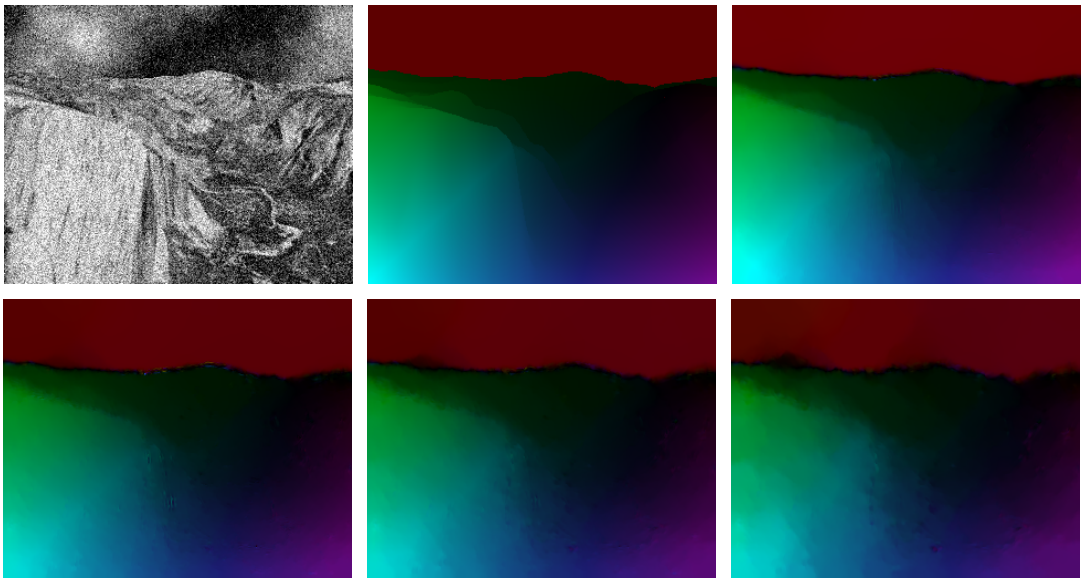
- Traffic Sequence Analysis
  (*Brox/Bruhn/Papenberg/Weickert 2004, Papenberg/Bruhn/Brox/Didas/Weickert 2006*)

Karl Wilhelm Straße (H.-H. Nagel)

Ettlinger Tor (H.-H. Nagel)

The Method of Brox et al. (13)

Results for the Brox et al. Method under Noise $\sigma_{\mathrm{n}} = 10, \ldots, 40$



Results for the Yosemite Sequence with clouds (L. Quam) with Gaussian noise of different standard deviation. **(a) Upper Left:** Frame 8. **(b) Upper Center:** Ground truth. **(c) Upper Right:** Brox *et al.* 3-D ($\sigma_{\mathrm{n}} = 0$). **(d) Lower Left:** Brox *et al.* 3-D ($\sigma_{\mathrm{n}} = 10$). **(e) Lower Center:** Brox *et al.* 3-D ($\sigma_{\mathrm{n}} = 20$). **(f) Lower Right:** Brox *et al.* 3-D ($\sigma_{\mathrm{n}} = 40$).

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

Results for the Brox et al. Method under Noise

- Qualitative Evaluation for the Yosemite Sequence with Clouds

Technique		AAE
Horn and Schnuck	$(\sigma_n = 40)$	16.80°
Horn and Schnuck	$(\sigma_n = 0)$	7.12°
Brox *et al.* 3-D	**$(\sigma_n = 40)$**	**4.49°**
Brox *et al.* 3-D	**$(\sigma_n = 30)$**	**3.87°**
Brox *et al.* 3-D	**$(\sigma_n = 20)$**	**3.21°**
Brox *et al.* 3-D	**$(\sigma_n = 10)$**	**2.49°**
Brox *et al.* 3-D	**$(\sigma_n = 0)$**	**1.78°**

- Even under Gaussian noise of $\sigma_n = 40$ accurate results can be obtained. This is mainly due to use of a robust data term and a spatiotemporal smoothness term.

Successive Overrelaxation

- *Known:* So far we have used the Jacobi method or its improved variant the Gauß-Seidel method to solve the linear system of equation $A\mathbf{x} = \mathbf{b}$.
- *Jacobi Method:* The iteration step for the **Jacobi method** is given by

$$\mathbf{x}^{k+1} = D^{-1}(\mathbf{b} + (L+U)\,\mathbf{x}^k) \quad \Leftrightarrow \quad x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ij}x_j^k\right) .$$

- *Gauß-Seidel Method:* The iteration step for the **Gauß-Seidel method** reads

$$\mathbf{x}^{k+1} = (D-L)^{-1}(\mathbf{b} + U\,\mathbf{x}^k) \quad \Leftrightarrow \quad x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{k+1} - \sum_{j>i} a_{ij}x_j^k\right) .$$

Here, D is the diagonal part, L is the lower triangular part and U ist the upper triangular part of the system matrix A.

Successive Overrelaxation (2)

How Can We Accelerate the Gauß-Seidel Method?

- *Idea:* **Pointwise extrapolate** the result of the Gauß-Seidel method
- *Successive Overrelaxation Method (SOR):* If we denote by $\bar{x}_i^{k+1}$ the result of the Gauß-Seidel method for the iteration step $k+1$, the **SOR technique** is given by

$$x_i^{k+1} = (1-\omega)\, x_i^k + \omega\, \bar{x}_i^{k+1} = (1-\omega)\, x_i^k + \omega \underbrace{\frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k \right)}_{\text{Gauß-Seidel result}}$$

with **overrelaxation parameter** $\omega \in [0, 2)$. In matrix notation this reads

$$\mathbf{x}^{k+1} = (D + \omega L)^{-1}(\omega \mathbf{b} - (\omega U + (1-\omega) D)\, \mathbf{x}^k)) \neq (1-\omega)\, \mathbf{x}^k + \omega \bar{\mathbf{x}}^k$$

- *Properties:* For positive definite system matrices the SOR method
  - converges if the overrelaxation parameter ω is chosen in the interval $[0, 2)$
  - is 1-2 orders of magnitude more efficient than the Gauß-Seidel method
  - comes down to the Gauß-Seidel method for $\omega = 1$

Successive Overrelaxation (3)

How Does the SOR Method looks like for the Horn and Schunck Method?

- *Equation System:* For the method of Horn and Schunck the associated linear system of equations is given by (cf. Lecture 4)

$$\begin{aligned} 0 &= [J_{11}]_{i,j}\, u_{i,j} + [J_{12}]_{i,j}\, v_{i,j} + [J_{13}]_{i,j} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u_{\tilde{i},\tilde{j}} - u_{i,j}}{h_l^2} \\ 0 &= [J_{12}]_{i,j}\, u_{i,j} + [J_{22}]_{i,j}\, v_{i,j} + [J_{23}]_{i,j} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v_{\tilde{i},\tilde{j}} - v_{i,j}}{h_l^2} \end{aligned}$$

for $i = 1, ..., N$ and $j = 1, ..., M$.

- *Example:* The corresponding SOR iteration step then reads

$$u_{i,j}^{k+1} = (1-\omega)\, u_{i,j}^k + \omega \frac{\left(-[J_{13}]_{i,j} - \left([J_{12}]_{i,j}\, v_{i,j}^k - \alpha \sum_{l \in x,y} \sum_{\mathcal{N}_l^-(i,j)} \frac{u_{\tilde{i},\tilde{j}}^{k+1}}{h_l^2} - \alpha \sum_{l \in x,y} \sum_{\mathcal{N}_l^+(i,j)} \frac{u_{\tilde{i},\tilde{j}}^k}{h_l^2} \right) \right)}{[J_{11}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}},$$

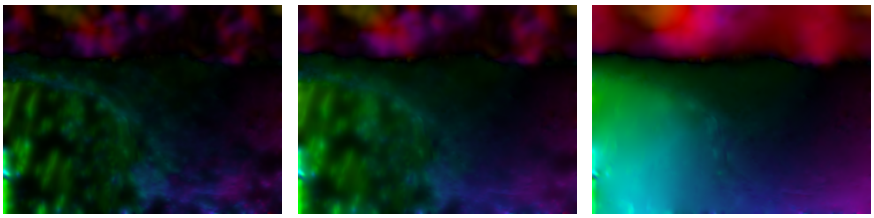
$$v_{i,j}^{k+1} = (1-\omega)\, v_{i,j}^k + \omega \frac{\left(-[J_{23}]_{i,j} - \left([J_{12}]_{i,j}\, u_{i,j}^{k+1} - \alpha \sum_{l \in x,y} \sum_{\mathcal{N}_l^-(i,j)} \frac{v_{\tilde{i},\tilde{j}}^{k+1}}{h_l^2} - \alpha \sum_{l \in x,y} \sum_{\mathcal{N}_l^+(i,j)} \frac{v_{\tilde{i},\tilde{j}}^k}{h_l^2} \right) \right)}{[J_{22}]_{i,j} + \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{1}{h_l^2}}.$$

Convergence Comparisons for Different Iterative Methods

- Qualitative Comparison for the Horn and Schunck Method after 100 Iterations

Solver	AAE
Jacobi method	29.58°
Gauß-Seidel method	22.63°
Successive Overrelaxation ($\omega = 1.96$)	7.18°

- Visual Comparison for the Horn and Schunck Method after 100 Iterations

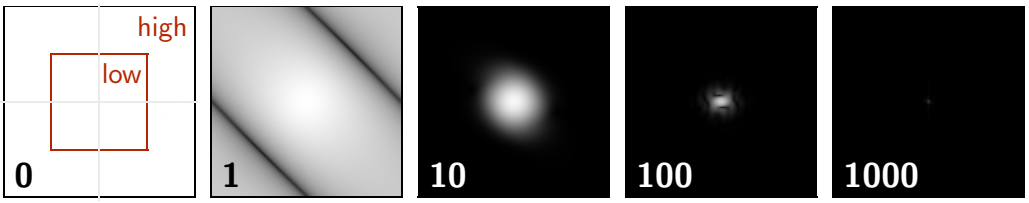


Results for the Yosemite Sequence with clouds (L. Quam) using 100 solver iterations. **(a) Left:** Jacobi method. **(b) Center:** Gauß-Seidel method. **(c) Right:** Successive Overrelaxation method ($\omega = 1.96$).

Basic Linear Multigrid

How Can We Solve Linear Systems of Equations Even More Efficiently?

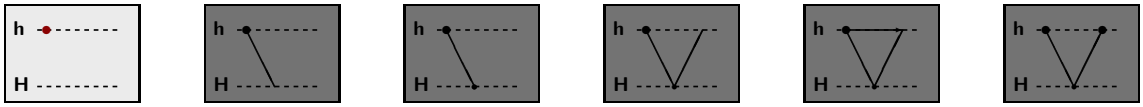
- *Observation:* Slow convergence of iterative solvers (Jacobi, Gauß Seidel, SOR) already after a few iterations. What is the reason of this behaviour?
  - logarithmic error spectrum reveals slow decrease of lower frequency parts (→ only efficient damping of **higher error frequency parts**)



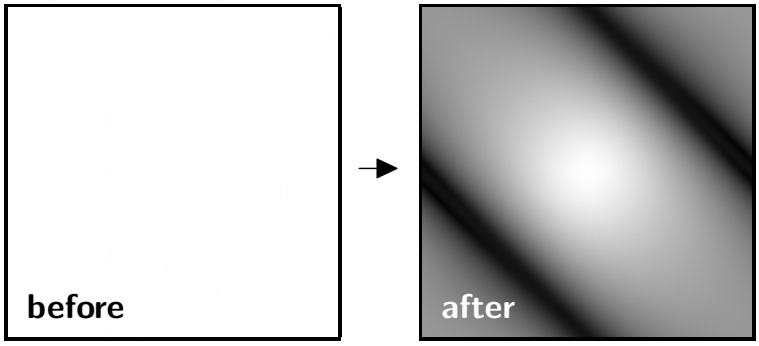
- *Sophisticated Idea*: Transfer and compute **error(!)** on coarser grids *(Brand 1977, Hackbusch 1985)*
  - low frequencies reappear as higher frequencies (→ also efficient damping of **lower error frequency parts**)

Basic Linear Multigrid (2)

Basic Linear Multigrid – The Two–Grid Cycle

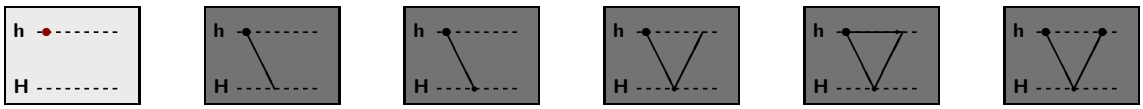


- *Step 1:* Presmoothing Relaxation
  - smoothing of higher error frequencies
    $\rightarrow$ application of n_1 solver iterations to $A^h\mathbf{x}^h = \mathbf{b}^h$
  - logarithmic error spectrum shows decrease of higher frequency parts



Basic Linear Multigrid (3)

Basic Linear Multigrid – The Two–Grid Cycle



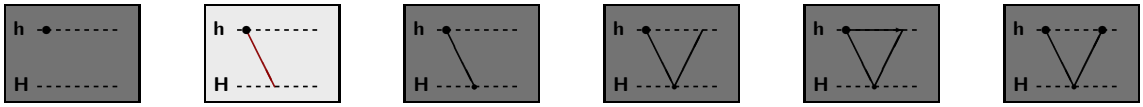
- *Question:* How shall we proceed?
  - error $\mathbf{e}^h = \mathbf{x}^h - \tilde{\mathbf{x}}^h$ **cannot** be computed directly
  - residual $\mathbf{r}^h = \mathbf{f}^h - A^h\tilde{\mathbf{x}}^h$ **can** be computed directly
  - linearity of matrix A^h yields the **residual equation**

$$\begin{aligned} A^h\mathbf{e}^h &= A^h(\mathbf{x}^h - \tilde{\mathbf{x}}^h) \\ &= A^h\mathbf{x}^h - A^h\tilde{\mathbf{x}}^h \\ &= \mathbf{f}^h - A^h\tilde{\mathbf{x}}^h \qquad = \mathbf{r}^h \,. \end{aligned}$$

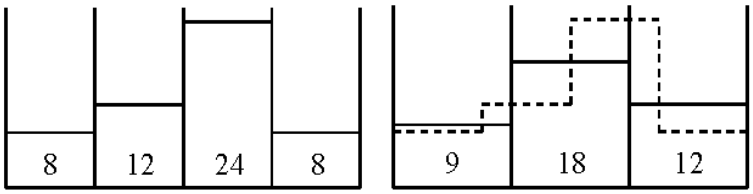
  - solving this **linear** system of equations $A^h\mathbf{e}^h = \mathbf{r}^h$ allows the desired correction of the approximate solution $\tilde{\mathbf{x}}^h$ by its error $\mathbf{e}^h$

Basic Linear Multigrid (4)

Basic Linear Multigrid – The Two–Grid Cycle



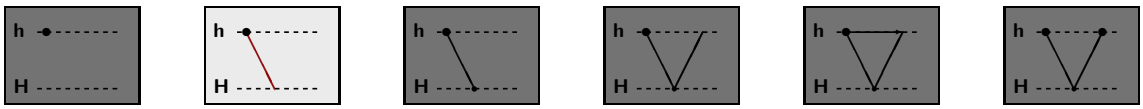
- *Step 2:* Restriction
  - transfer residual equation $A^h\mathbf{e}^h = \mathbf{r}^h$ to coarser grid $\rightarrow A^H\mathbf{x}^H = \mathbf{f}^H$
  - decision 1: choice of **coarse cell/grid size** H
    e.g. halving the pixel number yields doubling of the cell/grid size
  - decision 2: choice of **restriction operator** $R^{h\rightarrow H}$
    e.g. area-based averaging over $h \times h$ pixels



  - coarse grid RHS is then obtained by restriction : $\mathbf{f}^H = R^{h\rightarrow H}\mathbf{r}^h$

Basic Linear Multigrid (5)

Basic Linear Multigrid – The Two–Grid Cycle



- *Step 2:* Restriction (continued)
  - transfer residual equation $A^h\mathbf{e}^h = \mathbf{r}^h$ to coarser grid $\rightarrow A^H\mathbf{x}^H = \mathbf{f}^H$
  - decision 3: choice of **coarse grid matrix** A^H of operator A^h
    e.g. by Discretisation Coarse Grid Approximation (DCA)

- *Discretisation Coarse Grid Approximation (DCA)*
  - rediscretisation of Euler–Lagrange equations (restriction of motion tensors)

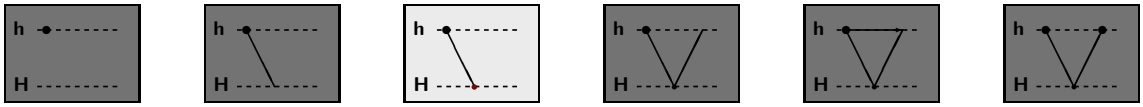
$$[J_{nm}]^H = R^{h\rightarrow H}[J_{nm}]^h \quad \text{for} \quad n,m \in \{1,2,3\}$$

  - substitution of fine grid size h by coarse grid size H (smoothness term)

$$\sum_{l\in x,y} \sum_{(\tilde{i},\tilde{j})\in\mathcal{N}_l(i,j)} \frac{u^h_{\tilde{i},\tilde{j}} - u^h_{i,j}}{h_l^2} \rightarrow \sum_{l\in x,y} \sum_{(\tilde{i},\tilde{j})\in\mathcal{N}_l(i,j)} \frac{u^H{}_{\tilde{i},\tilde{j}} - u^H{}_{i,j}}{H_l^2}$$

Basic Linear Multigrid (6)

Basic Linear Multigrid – The Two–Grid Cycle



- *Step 3:* Coarse Grid Computation
  - solve the restricted linear system of equations $A^H \mathbf{x}^H = \mathbf{f}^H$ given by

$$0 = [J_{11}]^H{}_{i,j}\, u^H{}_{i,j} + [J_{12}]^H{}_{i,j}\, v^H{}_{i,j} + [f_1]^H_{i,j} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{u^H{}_{\tilde{i},\tilde{j}} - u^H{}_{i,j}}{H_l^2}$$

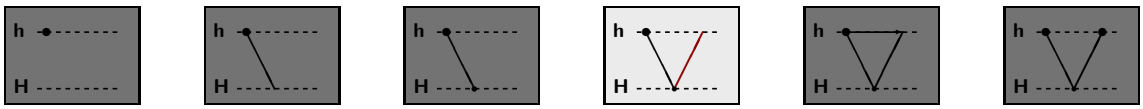
$$0 = [J_{12}]^H{}_{i,j}\, u^H{}_{i,j} + [J_{22}]^H{}_{i,j}\, v^H{}_{i,j} + [f_2]^H_{i,j} - \alpha \sum_{l \in x,y} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_l(i,j)} \frac{v^H{}_{\tilde{i},\tilde{j}} - v^H{}_{i,j}}{H_l^2}$$

  for $i = 1, ..., N^H$ and $j = 1, ..., M^H$ on the coarse grid.

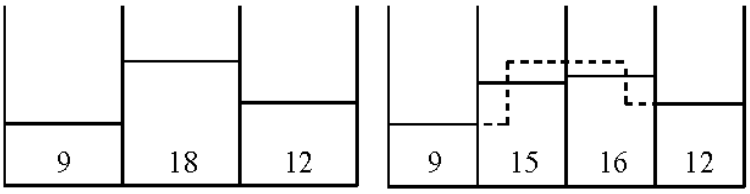
  - if pixel number is small, direct computation via Gaussian elimination
  - else iterative computation, e.g. by using the SOR method

Basic Linear Multigrid (7)

Basic Linear Multigrid – The Two–Grid Cycle



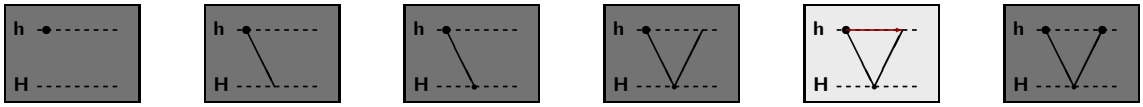
- *Step 4:* Prolongation
  - decision 4: choice of **prolongation operator** $P^{H \to h}$
    e.g. area-based interpolation over $h \times h$ pixels



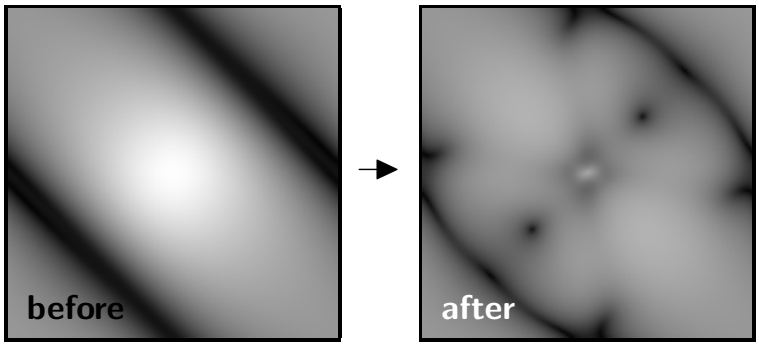
  - transfer result from coarse grid to fine grid : $\mathbf{e}^h = P^{H \to h} \mathbf{x}^H$

Basic Linear Multigrid (8)

Basic Linear Multigrid – The Two–Grid Cycle

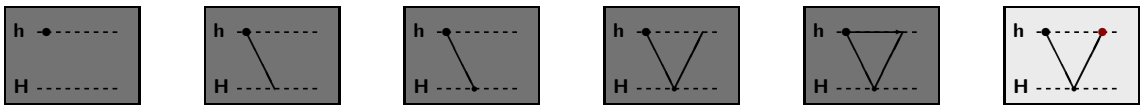


- *Step 5:* Correction From Coarse Grid
  - correction of approximation from presmoothing relaxation : $\tilde{x}^h_{\mathbf{new}} = \tilde{x}^h + e^h$
  - logarithmic error spectrum shows decrease of lower frequency parts
  - however, prolongation of error introduces new high frequency parts

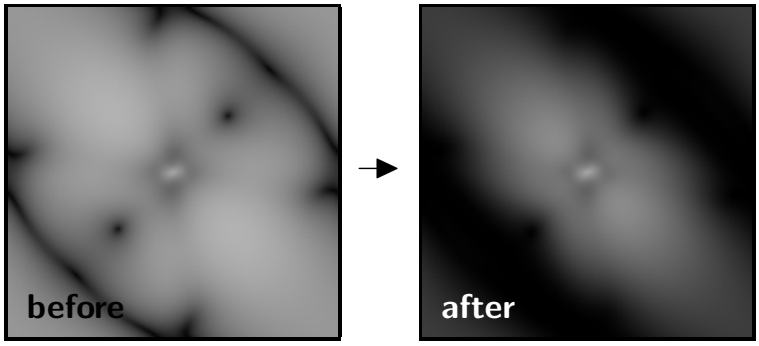


Basic Linear Multigrid (9)

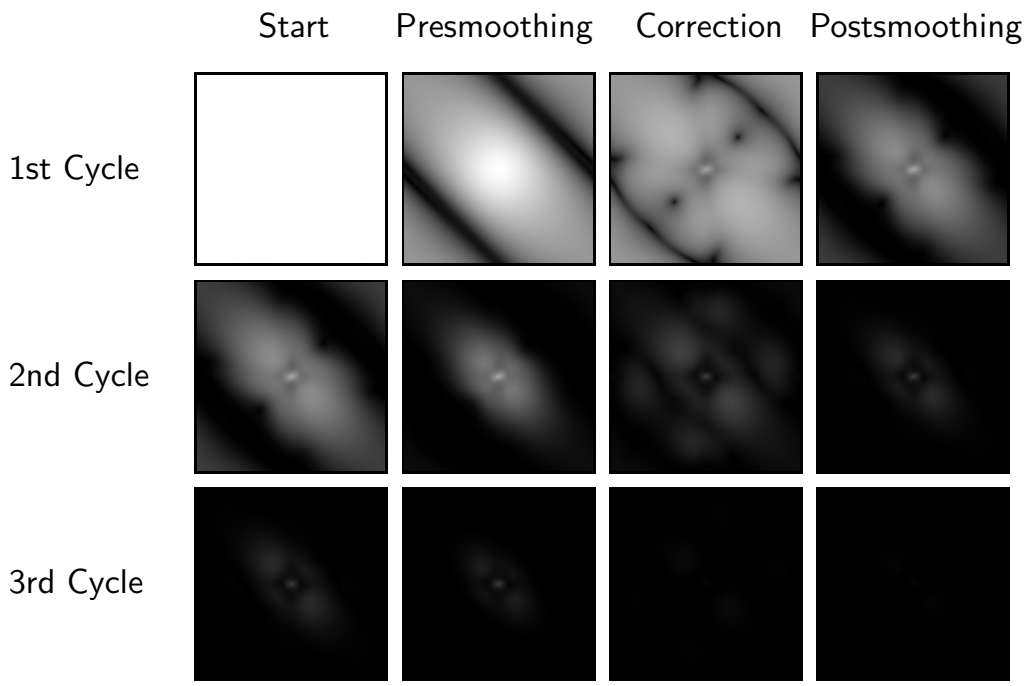
Basic Linear Multigrid – The Two–Grid Cycle



- *Step 6:* Postsmoothing Relaxation
  - smoothing of higher error frequencies introduced by interpolation
    $\rightarrow$ Application of n_2 solver iterations to $A^h x^h = f^h$
  - logarithmic error spectrum shows decrease of higher frequency parts



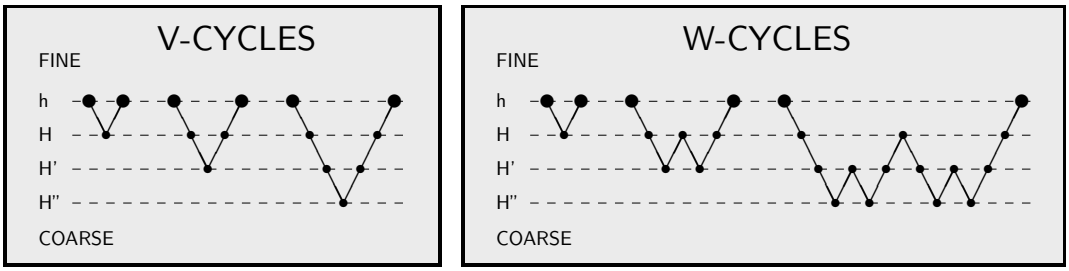
Efficient Error Reduction Through Three Cycles



Advanced Multigrid Strategies

How Can We Improve the Convergence of Multigrid Methods Even Further?

- *Idea 1:* Hierarchical application of the two–grid correction cycle
- *Example:* **one** or **two** recursive calls per level (→ **V–cycle**, **W–cycle**)

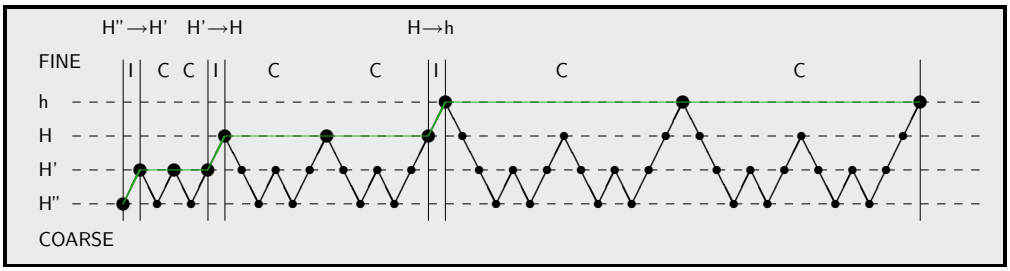


- *Idea 2:* Additionally start with better initialisation
- *Example:* embed V–/W–cycles in hierarchical initialisation (→ **Full Multigrid**)

Advanced Multigrid Strategies (2)

The Full Multigrid Strategy

- *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 via V-/W-cycles



Advanced Multigrid Strategies (3)

Comparison of Numerical Solvers (Image Size 160 × 120)

- *Testbed for Various Prototypes*
  - Standard desktop PC with 3.06 GHz Pentium4 CPU
  - C/C++ implementation
  - stopping criterion : relative error $e_{\mathbf{rel}} := ||\tilde{x} - x||_2/||x||_2$ of 10^{-2}
- *Linear Case:* Homogeneous Smoothness Term (Horn and Schunck) (*Bruhn/Weickert/Kohlberger/Schnörr 2005*)

Solver	Iterations	Time [s]	FPS [s^{-1}]	Speedup
Mod. Explicit Scheme	4425	3.509	0.285	1
Gauß-Seidel (CPR)	2193	1.152	0.868	3
SOR	82	0.052	19.233	67
Full Multigrid	**1**	**0.016**	**62.790**	**220**

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

Advanced Multigrid Strategies (4)

Comparison of Numerical Solvers (Image Size 160 × 120)

- *Linear Case:* Image-Driven Anisotropic Smoothness Term
  (*Bruhn/Weickert/Kohlberger/Schnörr 2005*)

Solver	Iterations	Time [s]	FPS [s^{-1}]	Speedup
Mod. Explicit Scheme	36433	47.087	0.021	1
Gauß-Seidel (ALR)	607	3.608	0.277	13
SOR	202	0.212	4.417	224
Full Multigrid	**1**	**0.171**	**5.882**	**275**

- *Nonlinear Case:* Flow-Driven Isotropic Smoothness Term
  (*Bruhn/Weickert/Kohlberger/Schnörr 2005*)

Solver	Iterations	Time [s]	FPS [s^{-1}]	Speedup
Mod. Explicit Scheme	10633	30.492	0.033	1
Gauß-Seidel (CPR)	2679	6.911	0.145	4
SOR	17/5	0.174	5.748	174
Full Multigrid	**1**	**0.082**	**12.172**	**372**

Advanced Multigrid Strategies (5)

Multigrid Speedups (Image Size 160 × 120)

- *Overview For Different Model Prototypes*
  (*Bruhn/Weickert/Kohlberger/Schnörr 2006*)

  - **two to three orders of magnitude** for different smoothness terms

Type	Solver	FPS	Speedup
Homogeneous	**Full Multigrid**	**62.7**	**220**
Image-Driven Isotropic	Full Multigrid	20.8	251
Image-Driven Anisotropic	**Full Multigrid**	**5.8**	**275**
Flow-Driven Isotropic	**FAS Full Multigrid**	**12.1**	**372**
Flow-Driven Anisotropic	FAS Full Multigrid	2.0	120

  - **three to four orders of magnitude** for high accuracy methods

Type	Solver	FPS	Speedup
Bruhn *et al.* 2-D, SD	FAS Full Multigrid	11.5	**2836**
Papenberg *et al.* 3-D, SD	FAS Full Multigrid	9.9	**10588**
Bruhn/Weickert 2-D, LD	Warp FAS Full Multigrid	2.9	**5454**

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

Real-Time Live Demo

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

Start
Stop

Flow fields are computed with a 1.7 GHz PentiumM CPU

Summary

- High accuracy methods combine many successful concepts
  - illumination-invariant constancy assumptions without linearisation
  - robust penalisation in the data term
  - spatiotemporal discontinuity-preserving smoothness term
- There exists a variety of efficient solvers
  - SOR as extrapolation variant of Gauß-Seidel (simple to implement)
  - Multigrid is even faster but requires problem specific adaptations (offers in the optimal case linear complexity → real-time capable)

Literature

- W. L. Briggs, V. E. Henson, S. F. McCormick.
  A Multigrid Tutorial.
  SIAM, Philadelphia, 2000.
  (introduction to multigrid methods)
- T. Brox, A. Bruhn, N. Papenberg, J. Weickert:
  High accuracy optical flow estimation based on a theory for warping.
  In T. Pajdla, J. Matas (Eds.): *Computer Vision – ECCV 2004*,
  Lecture Notes in Computer Science, Springer, Vol. 3024, pp. 25–36, 2004.
  (high accuracy method)
- A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, C. Schnörr:
  Variational optical flow computation in real-time.
  In *IEEE Transactions on Image Processing*, Vol. 14, No. 5, pp. 608–615, 2005.
  (linear multigrid for the CLG method)
- A. Bruhn, J. Weickert, T. Kohlberger, C. Schnörr:
  A multigrid platform for real-time motion computation with discontinuity-preserving variational methods.
  In *International Journal of Computer Vision*, Vol. 70, No. 3, pp. 257–277, 2006.
  (linear and nonlinear multigrid for a variety of prototypes)
- A. Bruhn, J. Weickert:
  Towards ultimate motion estimation: Combining highest accuracy with real-time performance.
  In *Proc. 2005 IEEE International Conference on Computer Vision*, Vol. 1, pp. 749–755,
  IEEE Computer Society Press, 2005.
  (multigrid for a high accuracy method)

Assignment 6

Theoretical Exercise 1 (Warping and SOR)

As shown in the lecture, the Horn and Schunck method with original nonlinear brightness constancy assumption requires to solve the following coupled system of PDEs at each warping level:

$$\begin{aligned} 0 &= J_{11}^k\, du^k + J_{12}^k\, dv^k + J_{13}^k - \alpha\Delta u^k - \alpha\Delta du^k \,, \\ 0 &= J_{12}^k\, du^k + J_{22}^k\, dv^k + J_{23}^k - \alpha\Delta v^k - \alpha\Delta dv^k \,. \end{aligned}$$

Here, J_{mn}^k with $n, m \in \{1, 2, 3\}$ are the motion compensated motion tensor entries at level k.

- Derive the corresponding linear system of equations after discretisation. In this context, it makes sense to visualise the structure of the resulting linear system. Which are the differences to the system without warping (with linearised constancy assumption)?
- After you have set up the linear equation system for each level, you must solve it. To this end, derive the corresponding SOR iteration step. This step will form the basis of your warping implementation in the next programming assignment (Assignment 7).

Assignment 6

Theoretical Exercise 2 (Photometric Invariants and Robustification)

Let $(R, G, B)^\top(x, y, t)$ denote a colour image sequence, where R, G and B are the red, green and blue channel respectively.

- Classify the following expressions with respect to their degree of photometric invariance w.r.t. changes of the overall intensity, shadow and shading, as well as highlights and specularities:

$$\begin{aligned} p_1 &= R - 3B + G & \qquad p_4 &= 2\,\tfrac{BG+RG}{B^2-R^2} - \tfrac{B+R}{B-R} \\ p_2 &= R^2 + B^2 - 2BR & \qquad p_5 &= (\ln B)_x + (\ln R)_x \\ p_3 &= \tfrac{R-B}{R+G} & \qquad p_6 &= \ln B_x - \ln R_x \end{aligned}$$

- Let us cast all these expression into a multichannel image with six channels $(p_1, \ldots, p_6)^\top(x, y, t)$. Derive an energy functional that reflects the assumption that all these expressions are constant over time and that allows to handle large displacements. Moreover, include robustness w.r.t. outliers in the data term. Complete the energy functional with a smoothness term of your choice.
- Compare the variants of the energy functional from (b) with joint and separate robustification. Derive in both cases the Euler-Lagrange equations. Taking a look at these equations, can you interpret the difference?