Lecture 4:
Optic Flow II
Global Differential Methods, Horn and Schunck

Contents

Variational Optic Flow Computation

What is a Functional?

- *Known:* A function maps an **input value** to an output value, e.g.

$$f(x) = x^2 .$$

- *New:* A functional maps an **input function** to an output value, e.g.

$$E(f(x)) = \frac{1}{|\Omega|} \int_\Omega f(x)\, dx\, dy .$$

- Functionals
  - are often based on integration strategies (to obtain a scalar value)
  - can be used to rate the quality of a function w.r.t. certain assumptions
  - form the basis of variational optic flow methods

Principle of Variational Optic Flow Methods

- Variational optic flow methods compute the displacement field as **minimiser** of a suitable energy functional:

$$E(u,v) = \int_\Omega \underbrace{D(u,v)}_{\textbf{data term}} + \alpha \underbrace{S(u,v)}_{\textbf{smoothness term}} \, dx\,dy \; .$$

  - **data term** $D(u,v)$ penalises deviations from constancy assumptions
  - **smoothness term** $S(u,v)$ penalises dev. from smoothness of the solution
  - regularisation parameter $\alpha > 0$ determines the degree of smoothness

- Remarks w.r.t. to the minimiser
  - The minimising functions u and v are those functions that fit best to all model assumptions, i.e. yield the smallest value for the energy functional.
  - Since in general the assumptions are partly contradictive, the result can be seen as a **compromise** between all assumptions.

The Method of Horn and Schunck

The Method of Horn and Schunck

- *Idea:* Assume overall smoothness of the resulting flow field
- The method of Horn and Schunck computes the optic flow as minimiser of *(Horn/Schunck 1981)*

$$E(\mathbf{w}) = \int_\Omega \underbrace{\mathbf{w}^\top J\, \mathbf{w}}_{\textbf{data term}} + \alpha \underbrace{\left(|\nabla u|^2 + |\nabla v|^2\right)}_{\textbf{smoothness term}} \, dx\,dy \; .$$

  - **data term** penalises deviations from the brightness constancy assumption

$$(f_x u + f_y v + f_t)^2 = \mathbf{w}^\top J\, \mathbf{w} = 0$$

    where the motion tensor J can be regarded as the structure tensor for $\rho = 0$

  - **smoothness term** penalises deviations from smoothness of the flow field, i.e. from variations of the functions given by their first derivatives

The Method of Horn and Schunck (2)

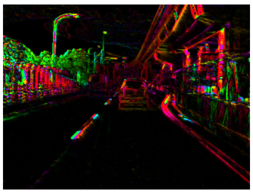
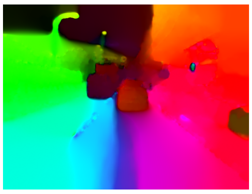
What Happens At Locations Where No Information is Available?

- *Data Term:* If no information is available $|\nabla f| \approx 0$ and thus u and v have no influence on the contribution of the data term

$$(f_x u + f_y v + f_t)^2 \approx f_t^2 \; .$$

- *Smoothness Term:* Consequently, u and v will be adapted completely to the local solution(s) of the neighbourhood (to fulfill at least the smoothness term).

- This propagation of information via the neighbourhood is called **filling-in-effect**. It is not limited to a neighbourhood of fixed size (in contrast to local methods).

edge information   filling-in

Minimisation (1)

Minimisation

How to Find the Minimiser of an Energy Functional?

- *Idea:* Similar strategy as for ordinary functions $\longrightarrow$ derive necessary conditions
- These necessary conditions are called **Euler-Lagrange equations**. They state that the first variation of the energy functional must vanish ($\approx$ first derivative).
- For a typical optic flow energy functional of type

$$E(u,v) = \int_\Omega F(x,y,u,v,u_x,u_y,v_x,v_y)\,dx\,dy$$

the Euler-Lagrange equations are given by the following coupled system of PDEs

$$0 \stackrel{!}{=} F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} \; ,$$
$$0 \stackrel{!}{=} F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y}$$

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

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

How Do These Equations Look Like for the Method of Horn and Schunck?

- For the Method of Horn and Schunck $F(x,y,u,v,u_x,u_y,v_x,v_y)$ is given by

$$\begin{aligned} F &= \mathbf{w}^\top J\,\mathbf{w} + \alpha\,(|\nabla u|^2 + |\nabla v|^2) \\ &= J_{11}u^2 + J_{22}v^2 + J_{33} + 2J_{12}uv + 2J_{13}u + 2J_{23}v + \alpha\,(u_x^2 + u_y^2 + v_x^2 + v_y^2)\,. \end{aligned}$$

- The required partial derivatives can then be computed as

$$\begin{aligned} F_u &= 2J_{11}u + 2J_{12}v + 2J_{13}\,, \quad & F_{u_x} &= \alpha\, 2u_x\,, \quad & F_{u_y} &= \alpha\, 2u_y\,, \\ F_v &= 2J_{12}u + 2J_{22}v + 2J_{23}\,, \quad & F_{v_x} &= \alpha\, 2v_x\,, \quad & F_{v_y} &= \alpha\, 2v_y\,. \end{aligned}$$

- As necessary condition for a minimiser this yields the Euler–Lagrange equations

$$\begin{aligned} 0 &= F_u - \tfrac{\partial}{\partial x}F_{u_x} - \tfrac{\partial}{\partial y}F_{u_y} &= \cancel{2}\Big(J_{11}u + J_{12}v + J_{13} - \alpha\,\overbrace{(u_{xx} + u_{yy})}^{\Delta u}\Big) \\ 0 &= F_v - \tfrac{\partial}{\partial x}F_{v_x} - \tfrac{\partial}{\partial y}F_{v_y} &= \cancel{2}\Big(J_{12}u + J_{22}v + J_{23} - \alpha\,\underbrace{(v_{xx} + v_{yy})}_{\Delta v}\Big) \end{aligned}$$

with Neumann boundary conditions $\mathbf{n}^\top \nabla u = 0$ and $\mathbf{n}^\top \nabla v = 0$.

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

Existence and Uniqueness of the Minimiser

- Strictly convex energy functionals such as the one of Horn and Schunck have
  - at most one minimum
  - if this minimum exists, the solution is **unique** (global minimum)
- What is strict convexity?
  - an energy functional is said to be strictly convex, if the following inequality holds for all $\alpha \in [0, ..., 1]$:

$$E(\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) < \alpha E(\mathbf{u}_1) + (1-\alpha)E(\mathbf{u}_2)\,.$$

- How can we prove strict convexity?
  - by showing that the definition holds
  - by showing that the Hessian of $E(\mathbf{u})$ is positive definite, i.e. that the gradient $\nabla E(\mathbf{u})$ is strictly monotone increasing

Existence and Uniqueness of the Minimiser

- *Sketch of Proof:* Strict Convexity
  - data and smoothness term are strictly convex in (u, v) due to the linearised BCCE and the use of quadratic penaliser functions in both terms
  - linear combinations of two strictly convex terms remain strictly convex
- Further properties of strictly convex variational optic flow methods
  (Schnörr 1994, Weickert/Schnörr 2001)
  - existence of a solution
  - solution depends continuously on the input data

Existence + Uniqueness + Continuous Dependency
=
Well-posedness (in the sense of Hadamard)

Discretisation

How Can We Solve The Euler-Lagrange-Equations Numerically?

- *Idea:* Discretise the Euler-Lagrange equations of the Horn and Schunck method

$$
\begin{aligned}
0 &= J_{11}u + J_{12}v + J_{13} - \alpha \Delta u \\
0 &= J_{12}u + J_{22}v + J_{23} - \alpha \Delta v
\end{aligned}
$$

  on a rectangular grid with spacing h_x in x-direction and spacing h_y in y-direction.

- This requires the following three steps:
  - discretisation of u and v
  - discretisation of the motion tensor entries $J_{11}, ..., J_{33}$
  - discretisation of $\Delta u = u_{xx} + u_{yy}$ and $\Delta v = v_{xx} + v_{yy}$

Discretisation (2)

Discretisation of the Flow Field

- *First Step:* We have to discretise the flow functions u and v
  - based on the grid spacings h_x and h_y we set

$$u_{i,j} = u(ih_x, jh_y) \quad \text{and} \quad v_{i,j} = v(ih_x, jh_y)$$

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

- *Second Step:* We have to discretise the motion tensor entries $J_{11}, ..., J_{33}$
  - these entries are given by

$$[J_{11}]_{i,j} = [f_x]^2_{i,j}\ , \qquad [J_{22}]_{i,j} = [f_y]^2_{i,j}\ , \qquad [J_{33}]_{i,j} = [f_t]^2_{i,j}\ ,$$
$$[J_{12}]_{i,j} = [f_x]_{i,j}\,[f_y]_{i,j}\ , \quad [J_{13}]_{i,j} = [f_x]_{i,j}\,[f_t]_{i,j}\ , \quad [J_{23}]_{i,j} = [f_y]_{i,j}\,[f_t]_{i,j}\ ,$$

    which in turn require the discretisation of f_x, f_y and f_t and therefore f.

Discretisation (3)

Discretisation of Image Derivatives

- *Second Step:* We have to discretise the motion tensor entries $J_{11}, ..., J_{33}$
  - analogously to u and v, we obtain for $f(x, y, t)$ and $f(x, y, t+1)$

$$f_{i,j,t} = f(ih_x, jh_y, t)$$

    for $i = 1, ..., N$, $j = 1, ..., M$ and $t = 1, 2$.

  - then, we can discretise f_x, f_y and f_t for instance via

$$[f_x]_{i,j} = \frac{1}{2}\left(\frac{f_{i+1,j,t+1} - f_{i-1,j,t+1}}{2h_x} + \frac{f_{i+1,j,t} - f_{i-1,j,t}}{2h_x}\right) \quad \text{(avg. central diff.)}$$
$$[f_y]_{i,j} = \frac{1}{2}\left(\frac{f_{i,j+1,t+1} - f_{i,j-1,t+1}}{2h_y} + \frac{f_{i,j+1,t} - f_{i,j-1,t}}{2h_y}\right) \quad \text{(avg. central diff.)}$$
$$[f_t]_{i,j} = \frac{f_{i,j,t+1} - f_{i,j,t}}{h_t} \quad \text{(forward diff.)}$$

    where the distance between both frames h_t is in general assumed to be 1.

Discretisation of Flow Derivatives

- *Third Step:* We have to discretise $\Delta u = u_{xx} + u_{yy}$ and $\Delta v = v_{xx} + v_{yy}$
  - to this end, one may use the following discretisation based on nested central differences with half the grid sizes $\frac{1}{2}h_x$ and $\frac{1}{2}h_y$, respectively

$$\begin{aligned}
\Delta u &= (u_x)_x + (u_y)_y \\
&\approx \frac{(u_x)_{i+\frac{1}{2},j} - (u_x)_{i-\frac{1}{2},j}}{2(\frac{1}{2}h_x)} + \frac{(u_y)_{i,j+\frac{1}{2}} - (u_y)_{i,j-\frac{1}{2}}}{2(\frac{1}{2}h_y)} \\
&\approx \frac{\frac{u_{i+1,j}-u_{i,j}}{2(\frac{1}{2}h_x)} - \frac{u_{i,j}-u_{i-1,j}}{2(\frac{1}{2}h_x)}}{2(\frac{1}{2}h_x)} + \frac{\frac{u_{i,j+1}-u_{i,j}}{2(\frac{1}{2}h_y)} - \frac{u_{i,j}-u_{i,j-1}}{2(\frac{1}{2}h_y)}}{2(\frac{1}{2}h_y)} \\
&= \frac{u_{i+1,j} - u_{i,j}}{h_x^2} - \frac{u_{i,j} - u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - u_{i,j}}{h_y^2} - \frac{u_{i,j} - u_{i,j-1}}{h_y^2} .
\end{aligned}$$

  - differences across the boundary vanish (Neumann boundary conditions)
  - the second expression, i.e. Δv, is approximated in the same way

Discrete Euler-Lagrange Equations

- The discrete Euler-Lagrange equations for the method of Horn and Schunck can finally be written as

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

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

  - here, $\mathcal{N}_l(i,j)$ denotes the set of neighbours of pixel i,j in direction of axis l (assuming four direct neighbours, i.e. two in each direction)
  - these equations constitute a **linear system of equations** w.r.t. the $2N \times M$ unknowns $u_{i,j}$ and $v_{i,j}$ for $i = 1, ..., N$ and $j = 1, ..., M$

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

Structure of the Linear System

- This linear system of equations $A\mathbf{x} = \mathbf{b}$ has the following block structure

$$\underbrace{\left(\left(\begin{array}{ccccc|ccccc} J_{11} & & & & & J_{12} & & & & \\ & J_{11} & & & & & J_{12} & & & \\ & & J_{11} & & & & & J_{12} & & \\ & & & J_{11} & & & & & J_{12} & \\ & & & & J_{11} & & & & & J_{12} \\ \hline J_{12} & & & & & J_{22} & & & & \\ & J_{12} & & & & & J_{22} & & & \\ & & J_{12} & & & & & J_{22} & & \\ & & & J_{12} & & & & & J_{22} & \\ & & & & J_{12} & & & & & J_{22} \end{array}\right) + \left(\begin{array}{ccccc|ccccc} 2 & -1 & & -1 & & & & & & \\ -1 & 3 & -1 & & -1 & & & & & \\ & -1 & 2 & -1 & & & & & & \\ -1 & & -1 & 3 & -1 & & & & & \\ & -1 & & -1 & 2 & & & & & \\ \hline & & & & & 2 & -1 & & -1 & \\ & & & & & -1 & 3 & -1 & & -1 \\ & & & & & & -1 & 2 & -1 & \\ & & & & & -1 & & -1 & 3 & -1 \\ & & & & & & -1 & & -1 & 2 \end{array}\right)\right)}_{A} \underbrace{\left(\begin{array}{c} u \\ u \\ u \\ u \\ u \\ \hline v \\ v \\ v \\ v \\ v \end{array}\right)}_{\mathbf{x}} = \underbrace{\left(\begin{array}{c} -J_{13} \\ -J_{13} \\ -J_{13} \\ -J_{13} \\ -J_{13} \\ \hline -J_{23} \\ -J_{23} \\ -J_{23} \\ -J_{23} \\ -J_{23} \end{array}\right)}_{\mathbf{b}}$$

  - **smoothness term** only contributes to block main diagonals
  - **data term** also contributes to block off-diagonals
- For non-constant input images the matrix A is positive definite
- For an image with 1M pixels, the matrix A has $4 \cdot 10^{12}$ entries. Assuming 32-bit float precision this requires 16 Terabyte memory ($\rightarrow$ store only non-zero entries). Direct Gauss-Elimination with complexity $O(n^3)$ is not practicable.

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

Iterative Solvers

How Can We Solve The Linear System of Equation $A\mathbf{x} = \mathbf{b}$?

- *Idea:* Find a cheap but accurate approximation of A via the decomposition *(e.g. Saad 1996)*

$$A = A_1 + A_2$$

- Introduce fixed point iteration of type

$$A_1\, \mathbf{x}^{k+1} = \mathbf{b} - A_2\, \mathbf{x}^{k}$$

$$\Leftrightarrow \mathbf{x}^{k+1} = A_1^{-1}(\mathbf{b} - A_2\, \mathbf{x}^{k})$$

- In each iteration a linear system of equations with matrix A_1 has to be solved
  - A_1 should be a reasonable approximation of A
  - A_1^{-1} should be cheap to compute, i.e. the system should be simple to solve

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

Iterative Solvers (2)

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

Frequent Approach

- Use matrix decomposition of type

$$A = D - L - U\,.$$

  - D is the diagonal part of A
  - L is the strictly lower triangular part of A
  - U is the strictly upper triangular part of A

- For the method of Horn and Schunck this yields

$$A = \left(\begin{array}{ccccc|ccccc} J_{11} & & & & & J_{12} & & & & \\ & J_{11} & & & & & J_{12} & & & \\ & & J_{11} & & & & & J_{12} & & \\ & & & J_{11} & & & & & J_{12} & \\ & & & & J_{11} & & & & & J_{12} \\ \hline J_{12} & & & & & J_{22} & & & & \\ & J_{12} & & & & & J_{22} & & & \\ & & J_{12} & & & & & J_{22} & & \\ & & & J_{12} & & & & & J_{22} & \\ & & & & J_{12} & & & & & J_{22} \end{array}\right) + \left(\begin{array}{ccccc|ccccc} 2 & -1 & & -1 & & & & & & \\ -1 & 3 & -1 & & -1 & & & & & \\ & -1 & 2 & -1 & & & & & & \\ -1 & & -1 & 3 & -1 & & & & & \\ & -1 & & -1 & 2 & & & & & \\ \hline & & & & & 2 & -1 & & -1 & \\ & & & & & -1 & 3 & -1 & & -1 \\ & & & & & & -1 & 2 & -1 & \\ & & & & & -1 & & -1 & 3 & -1 \\ & & & & & & -1 & & -1 & 2 \end{array}\right)$$

Iterative Solvers (3)

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

The Jacobi Method

- Set $A_1 = D$, since diagonal matrices are simple to invert. $A_2 = -L - U$.
- Yields the fixed point iteration

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

- For the method of Horn and Schunck the Jacobi iteration step for the pixel i,j finally reads

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

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

Iterative Solvers (4)

The Gauß-Seidel Method

- Set $A_1 = D - L$, since this triangular matrix is a better approximation to A than the diagonal D alone. Triangular matrices are still simple to invert. $A_2 = -U$.
- Yields the fixed point iteration

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

- For the method of Horn and Schunck the Gauß-Seidel iteration step for the pixel i, j finally reads

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

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

Iterative Solvers (5)

Remarks to the Gauß-Seidel Method

- Notation
  - $\mathcal{N}_l^-(i,j)$ denotes the set of neighbours of pixel i, j in direction of axis l that have already been computed
  - $\mathcal{N}_l^+(i,j)$ denotes the set of neighbours of pixel i, j in direction of axis l that have still to be computed
- Advantages
  - positive definiteness of the matrix A sufficient for convergence
  - about twice as fast as the Jacobi technique
    (due to the immediate propagation of new information)
  - does not require to store values from the previous iteration k
    (less memory consumption, easier to implement)
- Drawbacks
  - performance depends on the order in which the unknowns are traversed
    (symmetric variants exist that partly account for that problem)

Advantages and Shortcomings

Advantages and Shortcomings of Variational Optic Flow Methods

- Advantages
  - transparent modelling (no hidden model assumptions)
  - rotational invariance (can be arbitrarily well approximated)
  - unique minimiser and well-posedness (for strictly convex functionals)
  - straightforward minimisation (by simple algorithms such as e.g. Gauß-Seidel)
  - dense flow fields (due to filling-in effect)
  - sub-pixel precision (by continuous modelling)
- Drawbacks
  - require to solve large and sparse (non-)linear systems of equations
  - difficult to implement compared to local differential 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 22 23 24 25 26 27

Specific drawbacks of the Method of Horn and Schunck

- Missing robustness under varying illumination, outliers and noise
  - relies only on the grey value constancy assumption
  - uses quadratic data term, no explicit neighbourhood integration
- No handling of large displacements
  - employs linearised constancy assumption → BCCE
- No preservation of discontinuities
  - uses quadratic smoothness terms that assume overall smoothness

No Need to Worry → All These Problems Will Be Addressed

Lecture 5: Advanced constancy assumptions and large displacements
Lecture 6: Robust data terms and discontinuity preserving smoothness terms
Lecture 7: Efficient numerical solvers

Results

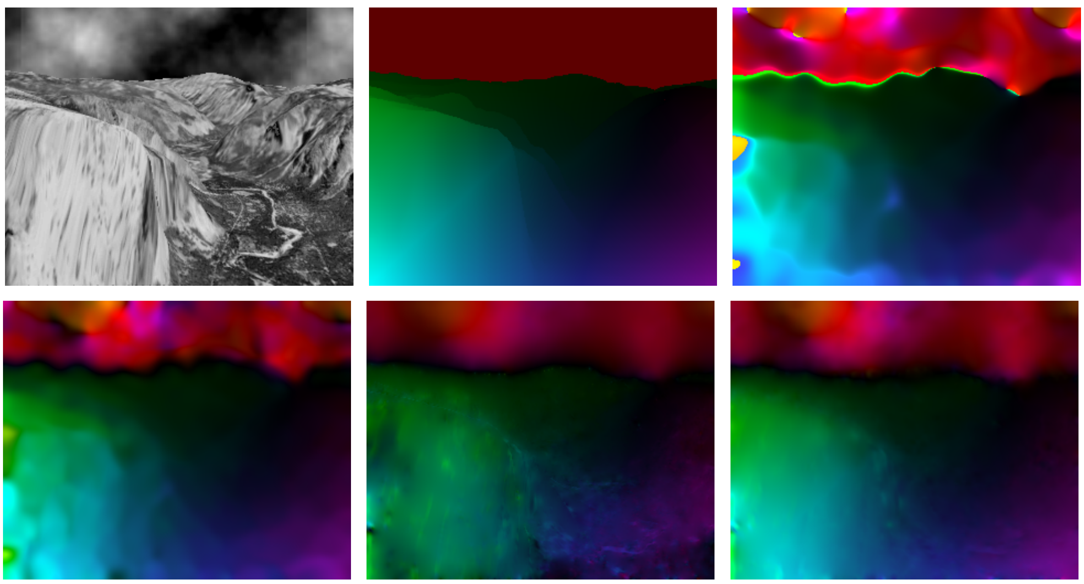
Comparison in Terms of the Average Angular Error (AAE)

- Qualitative Evaluation for the Yosemite Sequence with Clouds

Technique	AAE
Normalised Cross Correlation (NCC)	21.84°
Block Matching + Subpixel (SSD)	21.46°
Horn and Schunck (2-D)	**13.29°**
Bigün et al. + Presmoothing (2-D)	10.60°
Lucas/Kanade + Presmoothing (2-D)	8.79°
Horn and Schunck + Presmoothing (2-D) *(as presented in this lecture)*	**7.61°**
Horn and Schunck + Presmoothing (2-D) *(with improved derivative approximations)*	**7.12°**

MI 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

Results for the Horn and Schunck Method



Results for the Yosemite Sequence with clouds (L. Quam). **(a) Upper Left:** Frame 8. **(b) Upper Center:** Ground truth. **(c) Upper Right:** Bigün et al. 2-D ($\sigma = 1.6$, $\rho = 8.4$). **(d) Lower Left:** Lucas/Kanade 2-D ($\sigma = 1.4$, $\rho = 6.3$). **(d) Lower Center:** Horn and Schunck 2-D ($\sigma = 0.0$, $\alpha = 1650$). **(d) Lower Right:** Horn and Schunck 2-D ($\sigma = 1.4$, $\alpha = 470$).

Summary

- Variational methods compute the optic flow as minimiser of an energy functional
- They make use of global smoothness assumptions on the solution to overcome the aperture problem (filling-in-effect by the smoothness term → dense results)
- They are minimised by solving their (discretised) Euler-Lagrange equations
- They offer many advantages such as
  - transparent modelling
  - dense flow fields
  - well-posedness
  - sub-pixel precision
- The method of Horn and Schunck is the simplest variational approach
- There are many adaptations/modifications of this basic method possible that improve the performance even further (see forthcoming lecture)

Literature

- A. Bruhn, J. Weickert, C. Schnörr:
  Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods.
  In *International Journal of Computer Vision*, Vol. 61, No. 3, pp. 211–231, 2005.
  (detailed description of implementation issues for the method of Horn and Schunck and variants)
- B. Horn, B. Schunck:
  Determining optical flow. In *Artificial Intelligence*, Vol. 17, pp. 185–203, 1981.
  (*the original method of Horn and Schunck*)
- Y. Saad:
  Iterative Methods for Sparse Linear Systems.
  Thomson Learning, 1996.
  (book on iterative solvers for linear systems)
- C. Schnörr:
  Unique reconstruction of piecewise smooth images by minimizing strictly convex non-quadratic functionals.
  In *Journal of Mathematical Imaging and Vision*, Vol. 4, pp. 189–198, 1994.
  (*well-posedness of strictly convex variational optic flow methods*)
- J. Weickert, C. Schnörr:
  A theoretical framework for convex regularizers in PDE-based computation of image motion.
  In *International Journal of Computer Vision*, Vol. 45, No. 3, pp. 245–264, 2001.
  (more theoretical aspects of strictly convex variational methods)

Assignment 3

Programming Exercise (Horn and Schunck)

You can download the file copcv09_ex03.tgz from the web page

`http://www.mia.uni-saarland.de/Teaching/copcv09.shtml`

To unpack these data, use `tar xzvf copcv09_ex03.tgz`.

1. Supplement the routines `compute_motion_tensor()` and `horn_schunck_jacobi()` in the C programme `horn_schunck.c` with the missing code such that it becomes an implementation of the variational method of Horn and Schunck. In order to compile your programme please use the contained makefile. The compiled programme is then executed by

   `./frontend <input_image1.pgm> <input_image2.pgm> <zoom_ratio> [ground_truth.F]`

   where the integer parameter `zoom_ratio` is in general set to 1. The use of a ground truth file `ground_truth.F` is optional and triggers the computation of the average angular error (AAE).

2. Use the provided image pairs `ett1.pgm` and `ett2.pgm` as well as `yos1.pgm` and `yos2.pgm` and evaluate the performance of the method of Horn and Schunck for different values of the smoothness weight α. Is there a relation between the value for α and the number of required Jacobi iterations? Investigate also the influence of the presmoothing parameter σ on the quality of the results. In the case of the Yosemite sequence you can make use of the ground truth `yos_truth.F` to optimise your results with respect to the average angular error. Which is the smallest error you can obtain?