Lecture 25:

Image Sequence Processing II: Variational Methods

Contents

The Method of Horn and Schunck

Problems with the Local Methods from Lecture 24:

- nondense flow fields:
  - At locations with low gradients nothing can be said.
    Other locations may suffer from the aperture problem.
  - Discriminating these situations requires additional parameters.
- rigid, less flexible model assumptions:
  Local constancy of the optic flow is not suitable for non-translatory motion.

Remedy:

Variational model, in which the energy functional contains two assumptions:

- grey value constancy of corresponding image structures (OFC)
- smoothness of the flow field

The Method of Horn and Schunck (2)

Assumption 1: Grey Value Constancy

Corresponding image structures have the same grey value:
yields the linearised optic flow constraint (OFC, Lecture 24)

$$f_x u + f_y v + f_z = 0.$$

Nonuniqueness (aperture problem) requires a second constraint.

The Method of Horn and Schunck (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

Assumption 2: Smoothness

The optic flow field has only low spatial variations:

$$\int_\Omega \left(|\nabla u|^2 + |\nabla v|^2 \right) \, dx \, dy \qquad \text{is “small”.}$$

Remarks:

- The constraints from Lecture 24 hold *locally*, while this smoothness constraint holds *globally* on the entire image domain Ω.
- The smoothness constraint is less restrictive than a local constancy assumption: more adequate e.g. for non-translatory motion.
- The smoothness assumptions can be relaxed to *piecewise* smoothness in order to enable flow discontinuities. This requires nonquadratic penalisers (cf. Lecture 16).

The Method of Horn and Schunck (4)

Combining both assumptions lead to the

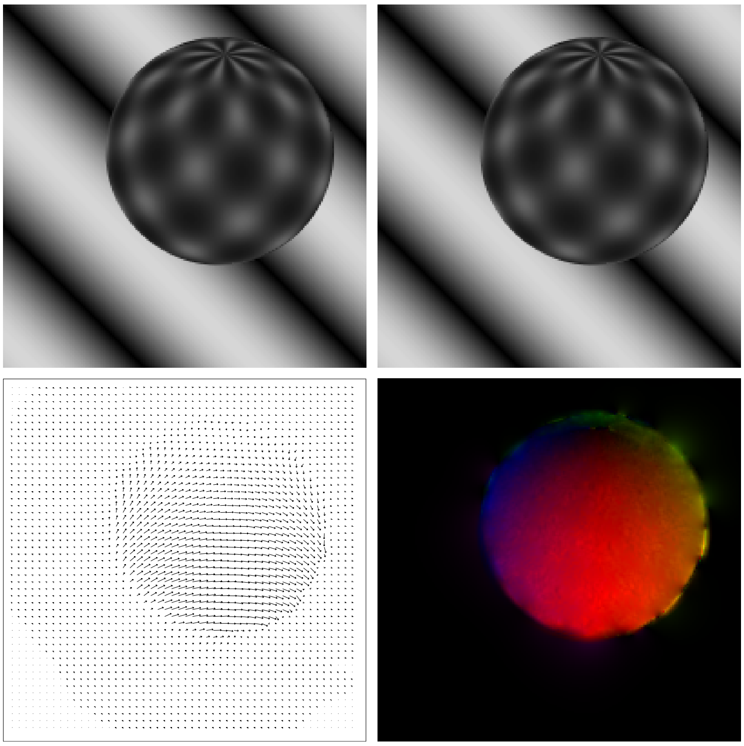
Variational Method of Horn and Schunck (1981):

At some given time z the optic flow field is determined as minimising function $(u(x,y), v(x,y))^\top$ of

$$E(u,v) := \int_\Omega \Big(\underbrace{\big(f_x u + f_y v + f_z\big)^2}_{\text{data term}} + \alpha \underbrace{\big(|\nabla u|^2 + |\nabla v|^2\big)}_{\text{smoothness term}} \Big)\, dx\, dy$$

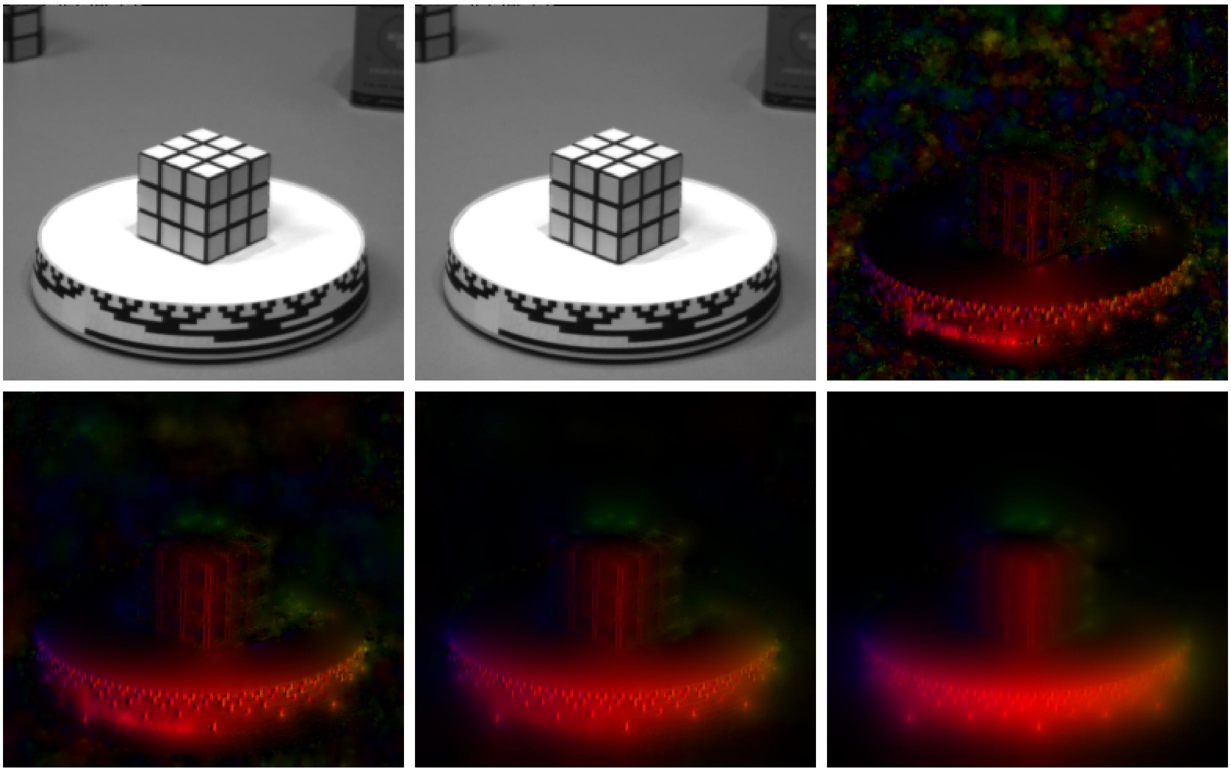
- unique solution that depends continuously on the image data (Schnörr 1991)
- method closely resembles variational methods for image enhancement (Lecture 16)
- regularisation parameter $\alpha > 0$ determines the desired smoothness of the flow field:
  - $\alpha \to 0$ yields the normal flow.
  - The larger α, the smoother the flow field.

The Method of Horn and Schunck (5)



Optic flow computation using the Horn–Schunck method. **(a) Top left:** Frame 10 of a synthetic image sequence. **(b) Top right:** Frame 11. **(c) Bottom left:** Optic flow, vector plot. **(d) Bottom right:** Optic flow, colour-coded. Author: J. Weickert (2000).

The Method of Horn and Schunck (6)



Influence of the regularisation parameter. **(a) Top left:** Frame 10 of the rotating cube sequence. **(b) Top middle:** Frame 11. **(c) Top right:** Optic flow, $\alpha = 1$. **(d) Bottom left:** $\alpha = 10$. **(e) Bottom middle:** $\alpha = 100$. **(f) Bottom right:** $\alpha = 1000$. Author: J. Weickert (2000).

The Method of Horn and Schunck (7)

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

Advantages of Variational Methods

- transparent concept without hidden model assumptions
- also suitable for non-translatory motion
- flexible: all model assumptions can be modified without problems
- main advantage: dense flow fields due to automatic *filling-in*:
  At locations, where no reliable flow estimation is possible (small image gradient), the relative weight of the smoothness term increases, and data from the vicinity are propagated.
- no additional threshold parameters necessary

Disadvantages of Variational Methods

- when using the linearised OFC:
  only for small displacements, fairly sensitive w.r.t. noise
- Simple algorithms are relatively slow.

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

A Simple Algorithm

Step 1: Going to the Euler–Lagrange Equations

Important Result from Variational Calculus:
Minimiser of the energy functional

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

satisfies the Euler–Lagrange equations

$$\begin{aligned} \partial_x F_{u_x} + \partial_y F_{u_y} - F_u &= 0, \\ \partial_x F_{v_x} + \partial_y F_{v_y} - F_v &= 0 \end{aligned}$$

with boundary conditions

$$\mathbf{n}^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = 0, \qquad \mathbf{n}^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} = 0$$

where $\mathbf{n}$ is the unit normal vector. This generalises results from Lecture 16.

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

Application to Our Problem

The integrand

$$F = (f_x u + f_y v + f_z)^2 + \alpha\,(u_x^2 + u_y^2 + v_x^2 + v_y^2)$$

gives the partial derivatives

$$\begin{aligned} F_u &= 2f_x(f_x u + f_y v + f_z), \\ F_v &= 2f_y(f_x u + f_y v + f_z), \\ F_{u_x} &= 2\alpha u_x, \\ F_{u_y} &= 2\alpha u_y, \\ F_{v_x} &= 2\alpha v_x, \\ F_{v_y} &= 2\alpha v_y. \end{aligned}$$

A Simple Algorithm (3)

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

This yields the Euler–Lagrange equations

$$\begin{aligned} \Delta u \;-\; \tfrac{1}{\alpha}\, f_x \,(f_x u + f_y v + f_z) \;&=\; 0, \\ \Delta v \;-\; \tfrac{1}{\alpha}\, f_y \,(f_x u + f_y v + f_z) \;&=\; 0 \end{aligned}$$

with the boundary conditions

$$\begin{aligned} \mathbf{n}^\top \nabla u \;&=\; 0, \\ \mathbf{n}^\top \nabla v \;&=\; 0. \end{aligned}$$

(All equations have been divided by 2α.)

A Simple Algorithm (4)

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

Step 2: Discretisation

- Two subsequent frames are sufficient.
- Gaussian presmoothing of the images can be useful (Lecture 11).
- Approximate image derivatives f_x, f_y, f_z by simple difference operators (central differences or Sobel operators, cf. Lecture 18).
- Approximation of the Laplacian (for pixel size 1):

$$\Delta u|_i \;\approx\; \sum_{j \in \mathcal{N}(i)} (u_j - u_i)$$

$\mathcal{N}(i)$: 4 neighbours of pixel i (boundary pixels have 3 neighbours, corner pixels 2)

A Simple Algorithm (5)

- yields the difference equations

$$0 = \sum_{j\in\mathcal{N}(i)} (u_j - u_i) - \tfrac{1}{\alpha} f_{xi} (f_{xi}u_i + f_{yi}v_i + f_{zi}),$$
$$0 = \sum_{j\in\mathcal{N}(i)} (v_j - v_i) - \tfrac{1}{\alpha} f_{yi} (f_{xi}u_i + f_{yi}v_i + f_{zi}).$$

for all pixels $(i = 1, ..., N)$.

- sparse but large linear system of equations
- Example:
  - 131072 unknows for an image size of 256×256 pixels
  - Loading the entire 131072×131072 matrix in foating point precision in your CPU would require 69 gigabytes !!! Don't try this at home or anywhere else !
  - Do not use direct algorithms such as Gaussian elimination (which has cubic complexity in the number of unknowns) !

A Simple Algorithm (6)

Step 3: Solving the Linear System

Reminder: The Jacobi Method (cf. Lecture 15)

- iterative method for solving a linear system $A\mathbf{x} = \mathbf{b}$.
- Let $A = D - N$ with diagonal D and rest N.
  Then the problem $D\mathbf{x} = N\mathbf{x} + \mathbf{b}$ is solved by the fixed point iteration

$$\mathbf{x}^{(k+1)} = D^{-1}(N\mathbf{x}^{(k)} + \mathbf{b})$$

- low effort per iteration, if the matrix is sparse:
  1 matrix-vector product, 1 vector addition, 1 vector scaling
- low additional memory requirements: vector $\mathbf{x}^{(k)}$
- well-suited for parallel computing
- residue $\mathbf{r}^{(k)} := A\mathbf{x}^{(k)} - \mathbf{b}$ yields stopping criterion: stop, if $\frac{|\mathbf{r}^{(k)}|}{|\mathbf{r}^{(0)}|} < \varepsilon$.

A Simple Algorithm (7)

Application to Our Problem

- Take the difference equations

$$0 = \sum_{j\in\mathcal{N}(i)} (u_j - u_i) - \frac{1}{\alpha} f_{xi} \, (f_{xi} u_i + f_{yi} v_i + f_{zi}),$$

$$0 = \sum_{j\in\mathcal{N}(i)} (v_j - v_i) - \frac{1}{\alpha} f_{yi} \, (f_{xi} u_i + f_{yi} v_i + f_{zi})$$

and shift the diagonal entries to the left hand side:

$$|\mathcal{N}(i)| \, u_i + \frac{1}{\alpha} f_{xi}^2 \, u_i = \sum_{j\in\mathcal{N}(i)} u_j - \frac{1}{\alpha} f_{xi} \, (f_{yi} \, v_i + f_{zi}),$$

$$|\mathcal{N}(i)| \, v_i + \frac{1}{\alpha} f_{yi}^2 \, v_i = \sum_{j\in\mathcal{N}(i)} v_j - \frac{1}{\alpha} f_{yi} \, (f_{xi} \, u_i + f_{zi})$$

where $|\mathcal{N}(i)|$ is the number of neighbours of pixel i.

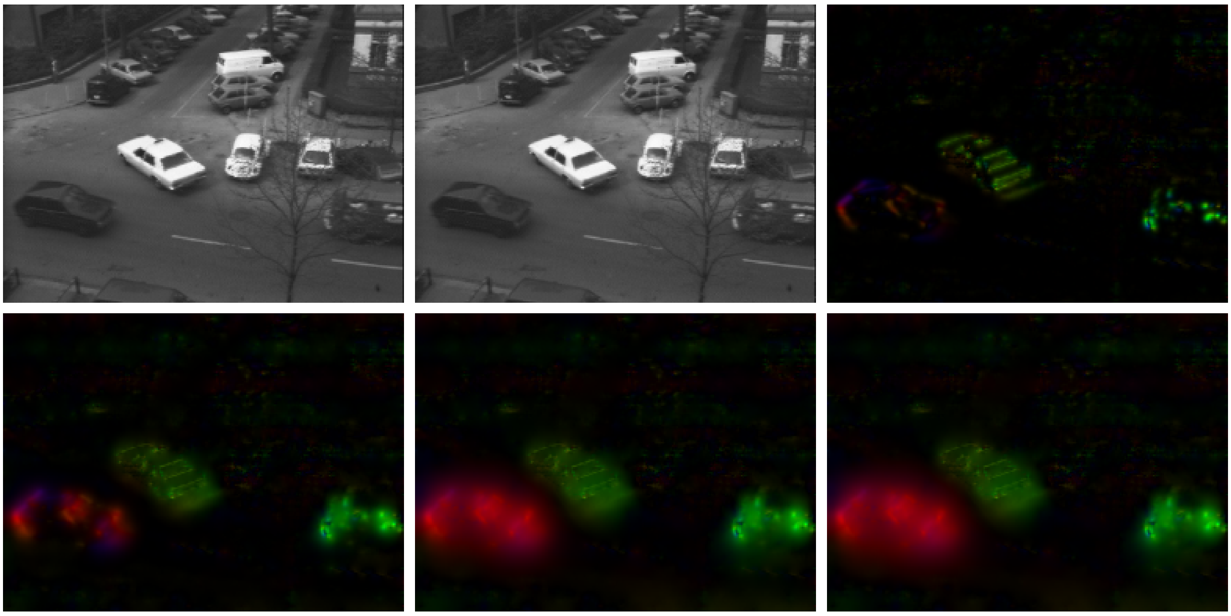
A Simple Algorithm (8)

- division by the diagonal coefficient gives a simple iterative scheme:

$$u_i^{(k+1)} = \frac{\sum_{j\in\mathcal{N}(i)} u_j^{(k)} - \frac{1}{\alpha} f_{xi} \, (f_{yi} \, v_i^{(k)} + f_{zi})}{|\mathcal{N}(i)| + \frac{1}{\alpha} f_{xi}^2},$$

$$v_i^{(k+1)} = \frac{\sum_{j\in\mathcal{N}(i)} v_j^{(k)} - \frac{1}{\alpha} f_{yi} \, (f_{xi} \, u_i^{(k)} + f_{zi})}{|\mathcal{N}(i)| + \frac{1}{\alpha} f_{yi}^2}$$

with $k = 0, 1, 2, ...$ and an arbitrary initialisation (e.g. null vector).



Influence of the number of Jacobi iterations. **(a) Top left:** Frame 10 of the Hamburg taxi sequence. **(b) Top middle:** Frame 11. **(c) Top right:** Optic flow after 10 iterations. **(d) Bottom left:** 100 iterations. **(e) Bottom middle:** 1000 iterations. **(f) Bottom right:** 10000 iterations. Author: J. Weickert (2000).

Extensions and Generalisations

◆ **Better Models:**

- Data Term:
  - other / multiple constraints (e.g. for severe illumination changes)
  - refraining from Taylor linearisation (for large displacements)
  - robust, nonquadratic terms allowing for outliers

- Smoothness Term:
  - discontinuity preserving nonquadratic regularisers
  - spatiotemporal smoothness terms instead of pure spatial regularisers: 3-D problems instead of 2-D

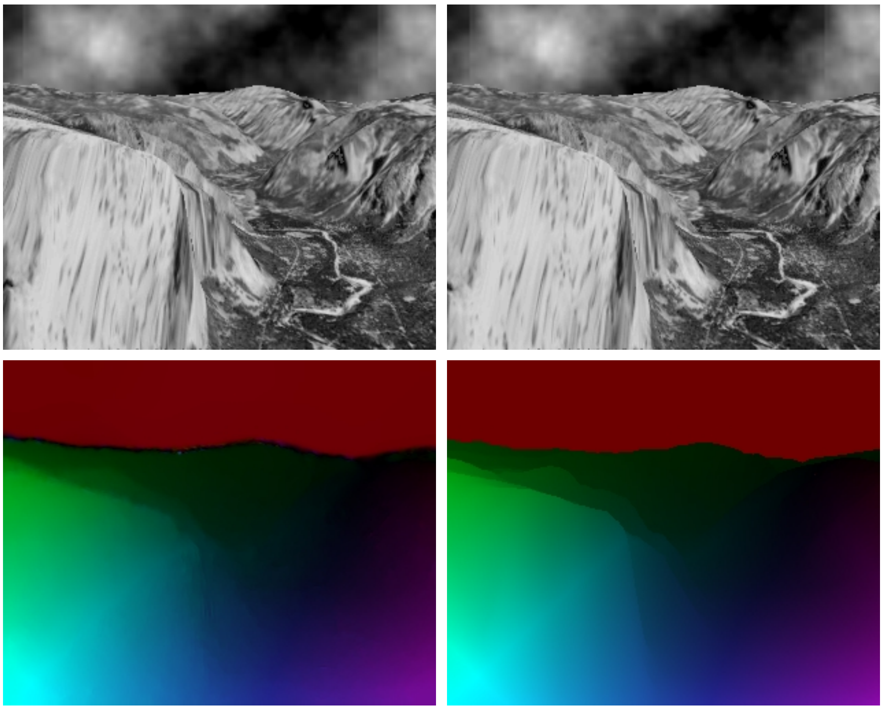
◆ **Better Algorithms:**

- faster iterative solvers: Gauß–Seidel, SOR, PCG, ADI, multigrid

- for large displacements: coarse-to-fine warping in a pyramidal setting

Extensions and Generalisations (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



(a) Top left: Frame 10 of the synthetic Yosemite sequence. **(b) Top right:** Frame 11. **(c) Bottom left:** Optic flow field in colour representation. The method that incorporates nonquadratic data and smoothness terms, grey value and gradient constancy, spatiotemporal regularisation and warping. **(d) Bottom right:** Ground truth optic flow field. Authors: Brox et al. (2004).

Summary (1)

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

Summary

- Variational methods for computing the optic flow are *global* methods.
- model assumptions: grey value constancy, smoothness of the flow field
- create dense flow fields by *filling-in*
- mathematically well-founded
- Minimising the energy functional leads to coupled differential equations.
- Discretisation creates a large, sparse linear system of equations
- can be solved iteratively, e.g. using the Jacobi method
- Variational methods can be extended and generalised in numerous ways, both with respect to models and to algorithms.

Literature

- B. K. P. Horn: *Robot Vision*. MIT Press, Cambridge, 1986.
  (a classical computer vision textbook by one of the pioneers of optic flow computation)
- B. Horn, B. Schunck: Determining optical flow. *Artificial Intelligence*, Vol. 17, pp. 185–203, 1981.
  (original paper by Horn and Schunck)
- C. Schnörr: Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *International Journal of Computer Vision*, Vol. 6, No. 1, pp. 25–38, April 1991.
  (well-posedness analysis of the Horn–Schunck method)
- 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, Part IV. Springer LNCS 3024, 25–36, 2004.
  `www.mia.uni-saarland.de/Publications/brox-eccv04-of.pdf`
  (one of the currently most accurate optic flow methods)