Lecture 24:
Image Sequence Processing II:
Variational Methods

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

The Method of Horn and Schunck (1981)

Problems with the Local Methods from the Previous Lecture

- nondense flow fields:
  - At areas with vanishing gradients nothing can be said.
    Other areas may still suffer from the aperture problem.
  - Discriminating these situations requires additional parameters.
- rigid, inflexible model assumptions:
  - Local constancy of the optic flow is not satisfied for non-translatory motion (e.g. rotations, divergent motion).

Remedy

- variational model, in which the energy functional contains two assumptions:
  - grey value constancy of corresponding image structures
  - smoothness of the flow field

The Method of Horn and Schunck (2)

Assumption 1: Grey Value Constancy

- Corresponding image structures should have the same grey value.
- This yields the linearised optic flow constraint (OFC, Lecture 23)

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

- Nonuniqueness (aperture problem) requires a second constraint.

The Method of Horn and Schunck (3)

Assumption 2: Smoothness

- The optic flow field should have only small spatial variations:

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

where $\boldsymbol{\nabla} = (\partial_x, \partial_y)^\top$ denotes the *spatial* nabla operator.

Remarks

- The Lucas–Kanade flow constancy constraint was enforced *locally* within B_ρ. The Horn–Schunck smoothness constraint holds *globally* on the image domain Ω.
- The smoothness constraint is less restrictive than a local constancy assumption: It is more adequate for non-translatory motion (e.g. rotation, divergent motion).
- To allow flow discontinuities, we can relax the smoothness assumptions to *piecewise* smoothness. This requires nonquadratic penalisers (cf. Lecture 18).

The Method of Horn and Schunck (4)

Combining the grey value constancy and the smoothness assumption leads to the

Variational Method of Horn and Schunck (1981)

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

$$E(u,v) \;:=\; \frac{1}{2}\int_\Omega \Big(\underbrace{\big(f_x u + f_y v + f_z\big)^2}_{\text{data term}} + \alpha \underbrace{\big(|\nabla u|^2 + |\nabla v|^2\big)}_{\text{smoothness term}} \Big)\, dx\, dy.$$

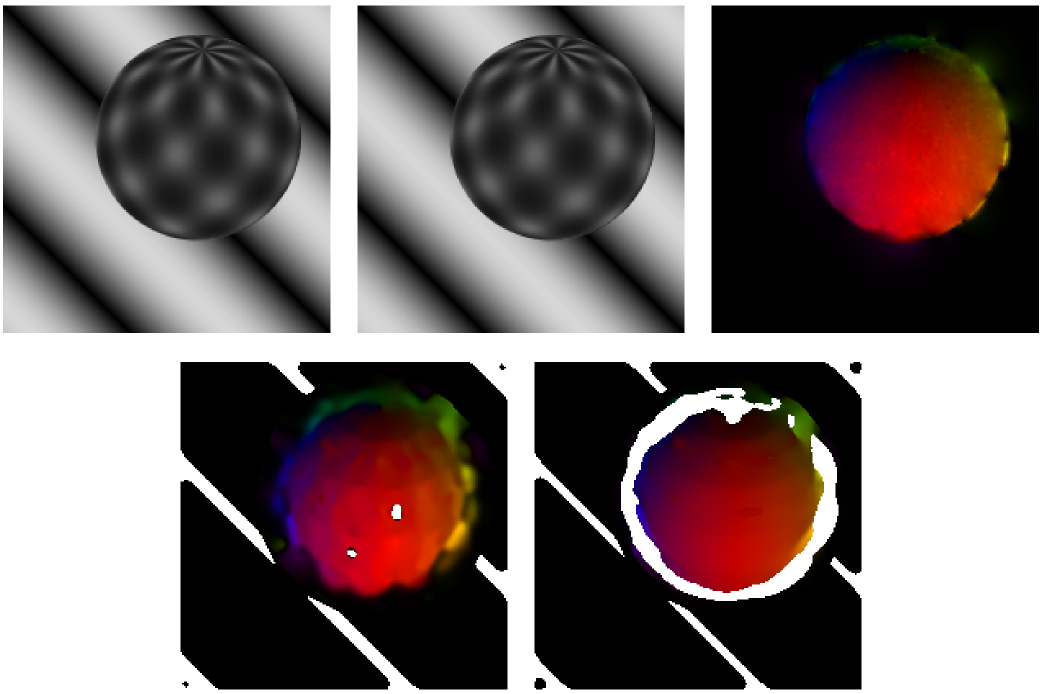
- unique minimiser that depends continuously on the image data (Schnörr 1991)
- similar structure as variational methods for image enhancement (Lecture 18):

$$E(u) \;:=\; \frac{1}{2}\int_\Omega \Big((u-f)^2 \,+\, \alpha\, |\nabla u|^2 \Big)\, dx\, dy.$$

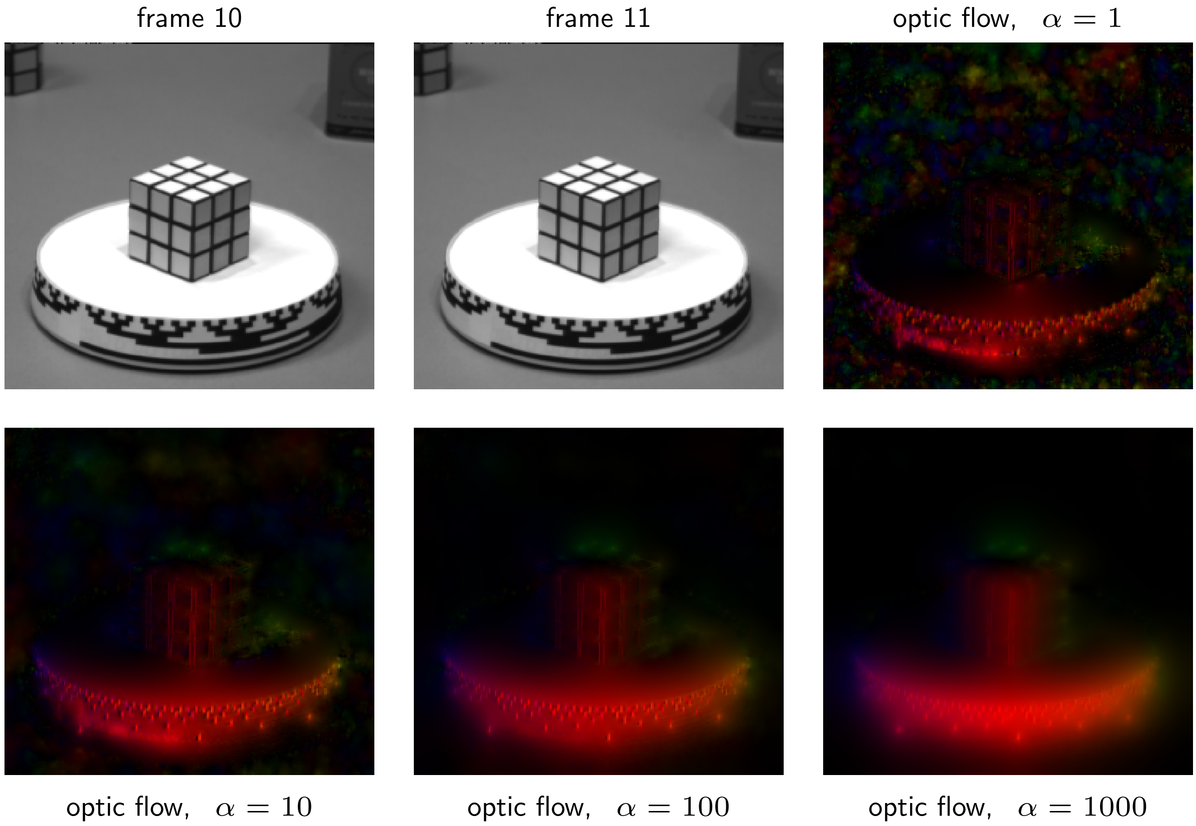
- regularisation parameter $\alpha > 0$ determines smoothness of the flow field:
  - The larger α, the smoother the flow field.
  - The normal flow minimises $E(u,v)$ for $\alpha \to 0$.

The Method of Horn and Schunck (5)

MIA 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



Optic flow computation with the Horn–Schunck method. **Top left:** Frame 10 of a synthetic image sequence. **Top middle:** Frame 11. **Top right:** Dense optic flow result for the Horn–Schunck method, colour-coded. **Bottom left:** Nondense result for Lucas–Kanade from Lecture 23. **Bottom right:** Nondense result for Bigün et al. from Lecture 23. Author: J. Weickert.



Influence of the regularisation parameter α for the rotating cube sequence. Author: J. Weickert.

MIA
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

Advantages of Variational Optic Flow Methods

- transparent concept that makes all model assumptions explicit
- no need to specificy threshold parameters:
  only a single, intuitive regularisation parameter α
- smoothness assumption also suitable for non-translatory motion
- flexible: all model assumptions can be modified without problems
- main advantage: dense flow fields due to *filling-in effect:*
  - At locations, where no reliable flow estimation is possible (small $|\nabla f|$), the smoothness term dominates over the data term.
  - This propagates data from the neighbourhood (like in inpainting; cf. Lecture 18).

Thus, variational models perform particularly well for optic flow computation.

Disadvantages of Variational Methods

- when using the simple linearised OFC:
  - only for small displacements or very smooth flow fields
  - more sensitive w.r.t. noise than local methods that average over neighbourhoods
- Simple algorithms are relatively slow.

These disadvantages can be cured with better models and algorithms.

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

A Simple Algorithm

Step 1: Going to the Euler–Lagrange Equations

Important Result from Calculus of Variations
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

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

with boundary conditions

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

where $\boldsymbol{n}$ is the unit normal vector. This combines results from Lecture 18.

A Simple Algorithm (2)

Application to Our Problem

The integrand

$$F = \frac{1}{2}\left(f_x u + f_y v + f_z\right)^2 + \frac{\alpha}{2}\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right)$$

has the partial derivatives

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

A Simple Algorithm (3)

After multiplication by -1, this yields the Euler–Lagrange equations

$$\begin{aligned}
\alpha\,\Delta u - f_x\left(f_x u + f_y v + f_z\right) &= 0, \\
\alpha\,\Delta v - f_y\left(f_x u + f_y v + f_z\right) &= 0.
\end{aligned}$$

After division by α, the boundary conditions are given by

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

A Simple Algorithm (4)

Step 2: Discretisation

- Note that we have to compute derivatives of the image sequence $f(x, y, z)$. Some spatial or spatiotemporal Gaussian smoothing of f can help (cf. Lecture 11).
- Approximate the image derivatives by finite differences (cf. Lecture 12):
  - forward differences for f_z:
    - two subsequent frames k and $k+1$ sufficient: smallest possible displacement
    - can be seen as first order approximation w.r.t. frame k, or as second order approximation w.r.t. $k+\frac{1}{2}$
  - central differences or Sobel operators for f_x and f_y:
    - when using only frame k: first order approximation in time w.r.t. k
    - when averaging over frames k and $k+1$: second order approximation in $k+\frac{1}{2}$
- standard approximation of the Laplacian with reflecting boundary conditions:

$$\Delta u|_i \approx \frac{1}{h^2} \sum_{j \in \mathcal{N}(i)} (u_j - u_i)$$

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

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

A Simple Algorithm (5)

- yields the following difference equations for all pixels $(i = 1, ..., N)$:

$$0 = \frac{\alpha}{h^2} \sum_{j \in \mathcal{N}(i)} (u_j - u_i) - f_{xi}\,(f_{xi}\,u_i + f_{yi}\,v_i + f_{zi}),$$

$$0 = \frac{\alpha}{h^2} \sum_{j \in \mathcal{N}(i)} (v_j - v_i) - f_{yi}\,(f_{xi}\,u_i + f_{yi}\,v_i + f_{zi})$$

- large but sparse linear system of equations with $2N$ unknowns $\{u_1, ..., u_N, v_1, ..., v_N\}$
- Example:
  - 131072 unknowns for an image size of 256×256 pixels
  - Loading the entire 131072×131072 matrix in floating point precision in your CPU would require 69 gigabytes !!!
    Don't try this at home or anywhere else !
  - However, we have not more than six nonvanishing matrix entries in each row.
  - Do not use direct algorithms such as Gaussian elimination:
    It has cubic complexity in the number of unknowns.

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

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

Step 3: Solving the Linear System

Reminder: The Jacobi Method (cf. Lecture 17)

- iterative method for solving a linear system $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{d}$
- Let $\boldsymbol{B} = \boldsymbol{D} - \boldsymbol{N}$ with a diagonal matrix $\boldsymbol{D}$ and a remainder $\boldsymbol{N}$.
  Then the problem $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{N}\boldsymbol{x} + \boldsymbol{d}$ is solved iteratively using
$$\boldsymbol{x}^{(k+1)} = \boldsymbol{D}^{-1}(\boldsymbol{N}\boldsymbol{x}^{(k)} + \boldsymbol{d}).$$
- low computational effort per iteration if $\boldsymbol{B}$ is sparse:
  1 matrix–vector product (sparse), 1 vector addition, 1 vector scaling
- only small additional memory requirement: vector $\boldsymbol{x}^{(k)}$
- well-suited for parallel computing
- residue $\boldsymbol{r}^{(k)} := \boldsymbol{B}\boldsymbol{x}^{(k)} - \boldsymbol{d}$ as simple stopping criterion: stop if $|\boldsymbol{r}^{(k)}| \leq \varepsilon\, |\boldsymbol{r}^{(0)}|$.

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 28

Application to Our Problem

- Take the difference equations

$$
\begin{aligned}
0 &= \frac{\alpha}{h^2} \sum_{j \in \mathcal{N}(i)} (u_j - u_i) \;-\; f_{xi}\,(f_{xi}\,u_i + f_{yi}\,v_i + f_{zi}),\\
0 &= \frac{\alpha}{h^2} \sum_{j \in \mathcal{N}(i)} (v_j - v_i) \;-\; f_{yi}\,(f_{xi}\,u_i + f_{yi}\,v_i + f_{zi})
\end{aligned}
$$

and shift the diagonal entries to the left hand side:

$$
\begin{aligned}
\frac{\alpha}{h^2}\,|\mathcal{N}(i)|\,u_i \;+\; f_{xi}^2\,u_i &= \frac{\alpha}{h^2} \sum_{j \in \mathcal{N}(i)} u_j \;-\; f_{xi}\,(f_{yi}\,v_i + f_{zi}),\\
\frac{\alpha}{h^2}\,|\mathcal{N}(i)|\,v_i \;+\; f_{yi}^2\,v_i &= \frac{\alpha}{h^2} \sum_{j \in \mathcal{N}(i)} v_j \;-\; f_{yi}\,(f_{xi}\,u_i + f_{zi})
\end{aligned}
$$

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

A Simple Algorithm (8)

- variables at left hand side are evaluated at iteration level $k+1$, and those on the right hand side at level k
- division by the diagonal coefficient gives a simple iterative scheme:

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

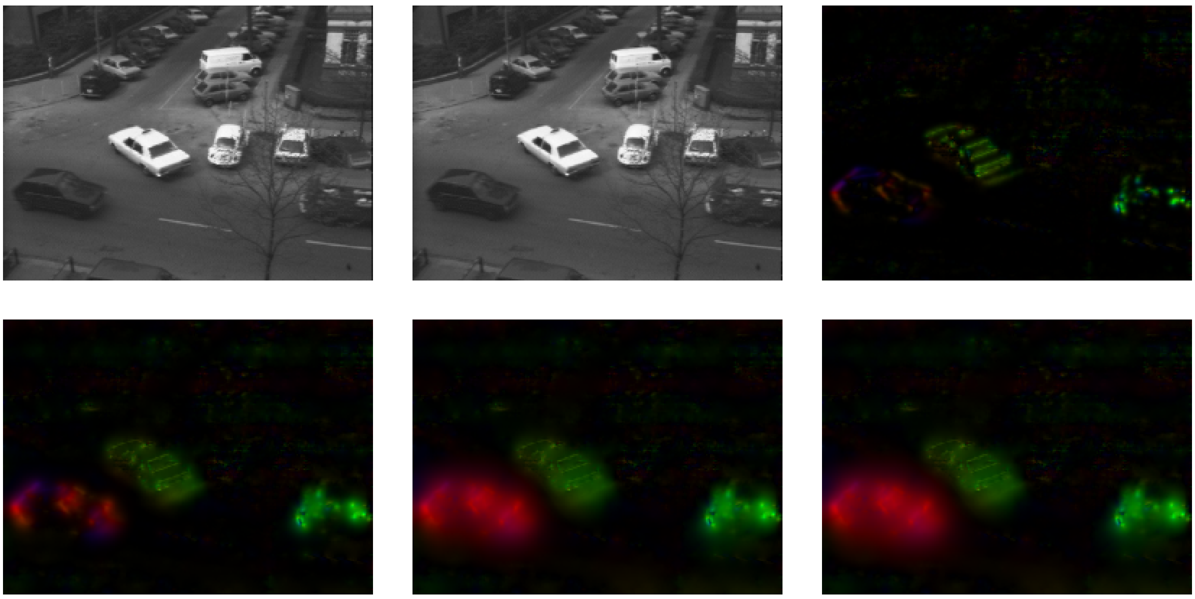
$$v_i^{(k+1)} = \frac{\frac{\alpha}{h^2} \sum\limits_{j \in \mathcal{N}(i)} v_j^{(k)} - f_{yi}\,(f_{xi}\,u_i^{(k)} + f_{zi})}{\frac{\alpha}{h^2}\,|\mathcal{N}(i)| + f_{yi}^2}$$

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

- reliable in practice, but may require quite some iterations

A Simple Algorithm (9)

frame 10 | frame 11 | 10 iterations



100 iterations | 1000 iterations | 10000 iterations

Influence of the number of Jacobi iterations on the optic flow result for the Hamburg taxi sequence. Author: J. Weickert.

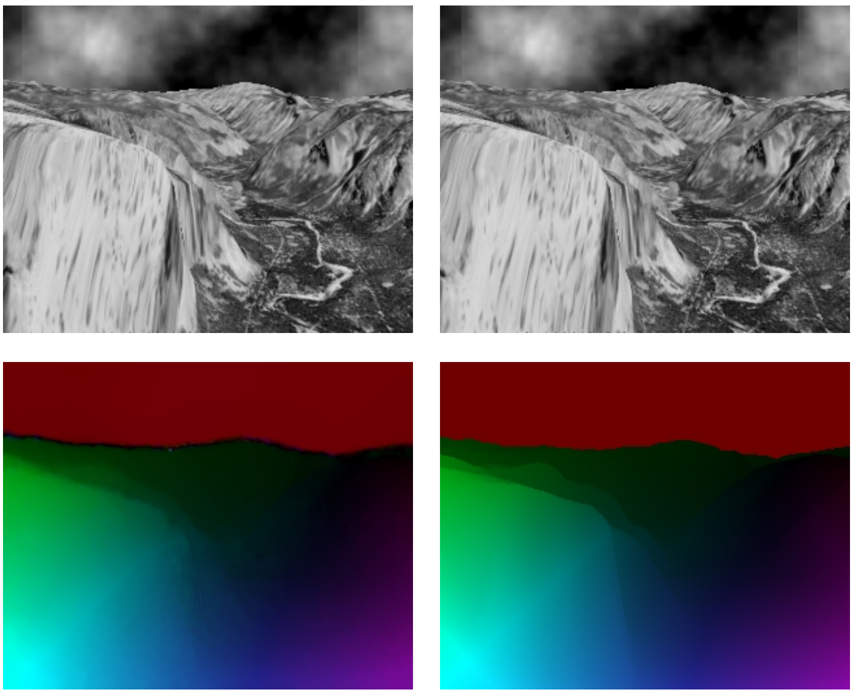
Extensions and Generalisations

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 (cf. Lecture 18)
  - spatiotemporal smoothness terms instead of pure spatial regularisers: 3-D problems instead of 2-D

Better Algorithms

- faster iterative solvers: Gauß–Seidel, SOR, PCG, multigrid
- for large displacements: coarse-to-fine warping in a pyramidal setting



Top left: Frame 10 of the synthetic Yosemite sequence. **Top right:** Frame 11. **Bottom left:** Computed optic flow field in colour representation. The method incorporates nonquadratic data and smoothness terms, grey value and gradient constancy, spatiotemporal regularisation and coarse-to-fine warping. **Bottom right:** Ground truth optic flow field. Authors: Brox et al.

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

Summary

- Variational methods for computing the optic flow are *global* methods.
- create dense flow fields by *filling-in*
- model assumptions of the variational Horn and Schunck approach: grey value constancy, smoothness of the flow field
- 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.

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

References

- 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.
  (`http://www.mia.uni-saarland.de/Publications/brox-eccv04-of.pdf`)
  (prototype of a highly accurate optic flow method)

Assignment C12 – Classroom Work

Problem 1 (Optic Flow Constraint in 1-D)

Consider a 1-D spatiotemporal signal $f(x,t)$, where x specifies the location and t denotes time. Derive the corresponding linearised optic flow constraint.
Is there an aperture problem or a situation where no flow can be computed?

Problem 2 (Optic Flow Constraint in 3-D)

Consider a 3-D spatiotemporal signal $f(x,y,z,t)$ where (x,y,z) specifies the location and t denotes time.

(a) Derive the corresponding linearised optic flow constraint and embed it into a Bigün-like approach.

(b) Assume that you are given an grey value image sequence where a cube floats around in a 3-D space.
Give a detailed classification of all situations based on the number of non-zero eigenvalues where the optic flow can be fully computed, partially computed, or not computed at all.
Connect these situations to the features of the cube.

Assignment H12 – Homework

Problem 1 (Lucas and Kanade Method for Colour Sequences) (1+1+1+1 points)

Let $\boldsymbol{f}(x,y,z) = (f_R(x,y,z), f_G(x,y,z), f_B(x,y,z))^\top$ be an image sequence in the RGB colour space and let $(u,v)^\top$ be the corresponding optic flow.

(a) Derive the linearised optic flow constraints for this problem.

(b) What can you say about the unique solution of this system? Distinguish three cases.

(c) Embed the optic flow constraints into an approach that corresponds to the Lucas and Kanade method. State the energy that is minimised.

(d) Determine the system of equations that a minimiser of E necessarily satisfies.

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

Assignment H12 (2)

Problem 2 (Variational Optic Flow) (1+2+1+2+2 points)

Let $f(x, y, z)$ be a sufficiently often continuously differentiable 2-D image sequence, where z denotes time. In order to compute the optic flow from such a sequence, many algorithms assume that the grey value of objects remains constant over time. However, in case of (approximately) additive illumination changes it may make sense to formulate constancy assumptions on image features that are based on derivatives. Therefore, let us assume for the moment that not the grey value, but the gradient remains constant over time.

(a) Prove that the gradient is invariant under additive illumination changes.

(b) Formulate the corresponding constancy assumption(s) without linearisation as well as the linearised optic flow constraint(s) that can be derived from it/them.

(c) What can you say regarding the presence of the aperture problem?

(d) Embed the corresponding squared data term into a Horn-and-Schunck-like approach with suitable quadratic smoothness term.

(e) Compute the associated Euler-Lagrange equations. You do not have to specify the boundary conditions.

Assignment H12 (3)

Please download the required files from the webpage

`http://www.mia.uni-saarland.de/Teaching/ipcv19.shtml`

into your own directory. You can unpack them with the command `tar xvzf Ex12.tar.gz`.

Problem 3 (Optic Flow Method of Lucas and Kanade) (2+2+2 points)

The program `lucaskanade.c` should be extended to the method by Lucas and Kanade. Starting the program with two frames of a sequence will yield two output images: the colour coded optic flow and a flow classification. This classification distinguishes three cases:

- one has no information (black),
- one can only calculate the normal flow (aperture problem, grey),
- one can calculate the full flow (white).

(a) Supplement the code in the subroutine `create_eq_systems` which is supposed to calculate the entries of the linear systems of equations solved in the Lucas-Kanade approach.

(b) The goal of the subroutine `lucas_kanade` is to reuse the entries calculated before and to solve the linear systems of equations. It should distinguish the three cases given above. The normal flow or the full flow should be calculated if possible, otherwise u and v should be set to zero. Supplement the missing code. You can use Cramer's rule to invert the linear system of equations, if this is possible.

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

Assignment H12 (4)

(c) Compile the program using the command

```
gcc -O2 -o lukaskanade lukaskanade.c -lm
```

and test the program with the image pairs `pig{1,2}.pgm`, and `sphere{1,2}.pgm`. What is the influence of the integration scale ρ? You can use a threshold $\varepsilon = 0.1$ for testing.

Problem 4 (Variational Optic Flow Estimation) (4+2 points)

(a) The program `hornschunck.c` should be extended to a simple Horn-Schunck method which means that the image sequence is not presmoothed by Gaussian convolution.
Please supplement the subroutine `flow` with the missing code. Make sure that the image boundaries are treated correctly. You can compile your program using the command

```
gcc -O2 -o hornschunck hornschunck.c -lm
```

(b) The *filling-in effect* that is characteristic for variational methods can be studied with the Danish pig images `pig1.pgm` and `pig2.pgm`. To this end, investigate the result for different numbers of iterations. What is a good value for the regularisation parameter α in this case?

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

Assignment H12 (5)

Submission

The theoretical Problems 1 and 2 should be submitted in handwritten form before the lecture.
For the practical Problems 3 and 4, please submit your files as follows: Rename the main directory `Ex12` to `Ex12_<your_name>` and use the command

```
tar czvf Ex12_<your_name>.tar.gz Ex12_<your_name>
```

to pack the data. The directory that you pack and submit should contain the following files:

- the source code of Problem 3 and 4.
- the created images of Problem 3(c) for two different values for ρ (i.e. eight images).
- two images created in Problem 4(b), that demonstrate the behaviour of the result for different numbers of iterations based on a reasonable choice for α.
- a text file README that
  - states the used parameters
  - answers the question of Problem 3(c) and 4(b),
  - contains information on all people working together for this assignment.

Please make sure that only your final version of the programs and images are included.
Submit the file via e-mail to your tutor via the address:

```
ipcv-xx@mia.uni-saarland.de
```

where `xx` is either `t1`, `t2`, `t3`, `t4`, `t5`, `w1`, `w2`, `w3` or `w4` depending on your tutorial group.

Deadline for submission: Friday, July 5, 10 am (before the lecture)

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