Optic Flow and Basics Towards Horn-Schunck[1]

Lecture 7

See Section 4.1 and Beginning of 4.2 in
Reinhard Klette: Concise Computer Vision
Springer-Verlag, London, 2014

[1] See last slide for copyright information.

Frames in a Video Sequence

Sequence of images or video *frames*

Time difference δt between two subsequent frames

Example: $\delta t = 1/30$ s means

30 Hz (read: "hertz") or 30 fps (read: "frames per second")

$I(.,.,t)$ is the frame at time t with values $I(x,y,t)$

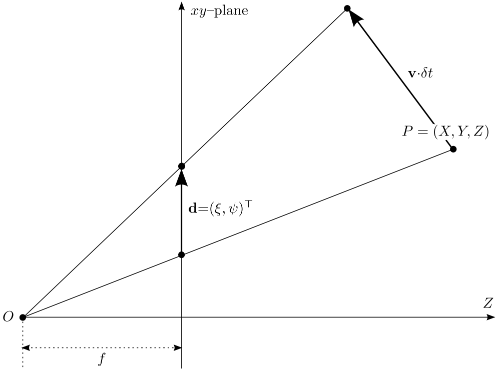
Agenda

Projected Motion

$P = (X, Y, Z)$ projected at $t \cdot \delta t$ into $p = (x, y)$ in $I(.,.,t)$

Camera: *focal length* f, *projection centre* O, looks along *optic axis*

Ideal model defines *central projection* into xy image plane



Projection of motion $\mathbf{v} \cdot \delta t$ into displacement $\mathbf{d}$ in the image plane

2D Motion

Assumptions: motion of P between $t \cdot \delta t$ and $(t+1) \cdot \delta t$

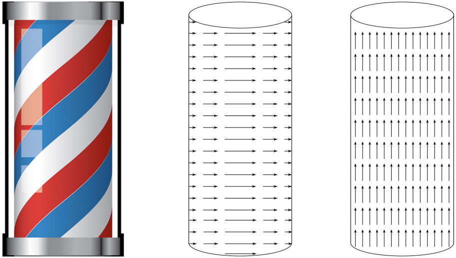
1. linear
2. with constant speed

Local displacement:
Projection $\mathbf{d} = (\xi, \psi)^\top$ of this 3D motion

Visible displacement:
The *optic flow* $\mathbf{u} = [u, v]^\top$ from $p = (x, y)$ to $p = (x + u, y + v)$

Often: optic flow not identical to local displacement

2D Motion $\neq$ Optic Flow



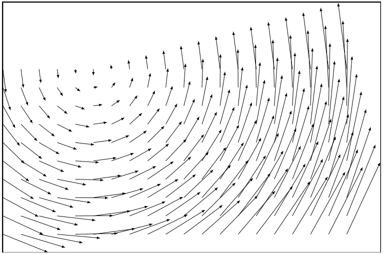
Rotating barber's pole: sketch of 2D motion (without scaling vectors) and sketch of optic flow, an optical illusion

Lambertian sphere: rotation but not visible

Moving light source: "textured" static object and a moving light source (e.g., the sun) generate optic flow

Vector Fields

Rotating rectangle: around a fixpoint, parallel to the image plane:



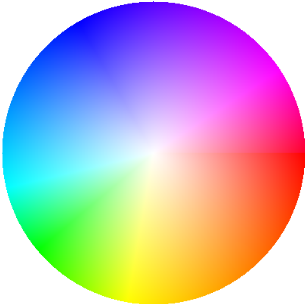
Motion maps: vectors start at time t and end at time $t+1$

To be visualized by using a color key

Dense if vectors at (nearly) all pixel locations; otherwise *sparse*

Difficult to infer the shape of a polyhedron from a motion map

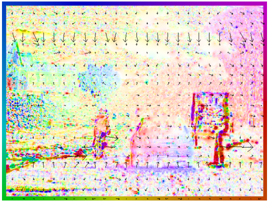
Color Key



Colors represent direction of vector (start at the center of the disk)

Saturation represents magnitude of the vector, with White for “no motion”

Example 1 for Horn-Schunck-Algorithm

Subsequent frames taken at 25 fps

Color-coded motion field calculated with basic Horn-Schunck algorithm

Sparse (magnified) vectors are redundant information

Example 2 for Horn-Schunck-Algorithm

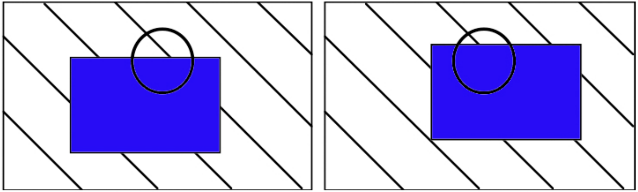
Two frames of video sequence

Color-coded motion field calculated with basic Horn-Schunck algorithm

Agenda

Aperture Defined by Available Window

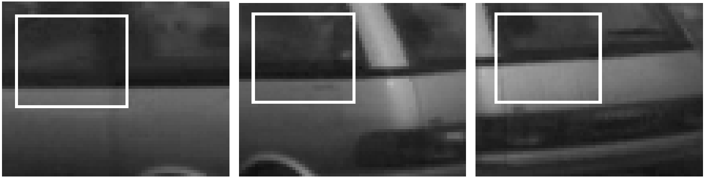
Sitting in a waiting train and assuming to move because the train on the next track started to move



A program only "sees" both circular windows at time t (*left*) and time $t+1$ (*right*); it concludes an upward shift and misses the shift diagonally towards the upper right corner

Camera Aperture

Visible motion defined by the aperture of the camera



Images taken at times t, $t+1$, and $t+2$

Inner rectangles: we conclude an upward translation with minor rotation

Three images: indicate a motion of this car to the left

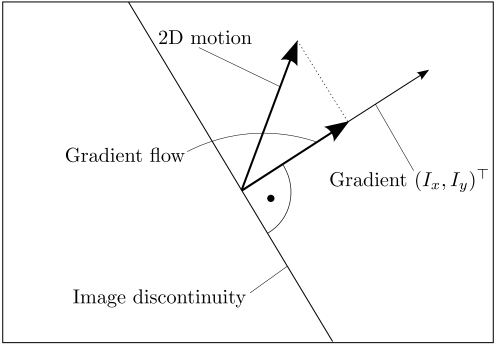
Ground truth: car is actually driving around a roundabout

Gradient Flow

Due to aperture problem: local optic flow detects *gradient flow*

2D gradient $\nabla_{x,y} I = (I_x(x,y,t), I_y(x,y,t))^\top$

I_x and I_y are partial derivatives of $I(.,.,t)$ w.r.t. x and y



True 2D motion **d**: diagonally up

Identified motion: projection of **d** onto gradient vector

Agenda

Origin of the Algorithm

The algorithm was published in

B.K.P. Horn and B.G. Schunck. Determining optic flow. Artificial Intelligence, vol 17, pp. 185–203, 1981

as a pioneering work for estimating optic flow.

We discuss this algorithm in detail

Agenda

Taylor expansion

Difference quotient

$$\frac{\phi(x) - \phi(x_0)}{x - x_0} = \frac{\phi(x_0 + \delta x) - \phi(x_0)}{\delta x}$$

of function ϕ converges into differential quotient

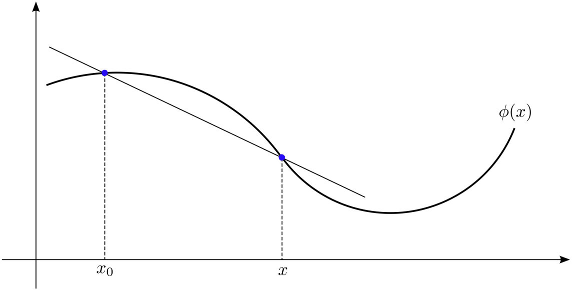
$$\frac{\mathrm{d}\phi\left(x_0\right)}{\mathrm{d}x}$$

for $\delta x \to 0$. **First-order Taylor expansion**:

$$\phi\left(x_0 + \delta x\right) = \phi\left(x_0\right) + \delta x \cdot \frac{\mathrm{d}\phi\left(x_0\right)}{\mathrm{d}x} + e$$

where *error* e equals zero if ϕ is linear in $[x_0, x_0 + \delta x]$

Taylor expansion for 1D Case



General 1D Taylor expansion:

$$\phi\left(x_0 + \delta x\right) = \sum_{i=0,1,2,\ldots} \frac{1}{i!} \cdot \delta x^i \cdot \frac{\mathrm{d}^i \phi\left(x_0\right)}{\mathrm{d}^i x}$$

with $0! = 1$, $i! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot i$ for $i \geq 1$, and d^i is the ith derivative

3D Taylor Expansion

In our case of the frame sequence:

$$
\begin{aligned}
& I(x+\delta x, y+\delta y, t+\delta t) \\
& \qquad = I\,(x,y,t) + \delta x \cdot \tfrac{\partial I}{\partial x}\,(x,y,t) + \delta y \cdot \tfrac{\partial I}{\partial y}\,(x,y,t) \\
& \qquad\qquad + \delta t \cdot \tfrac{\partial I}{\partial t}\,(x,y,t) + e
\end{aligned}
$$

Assumption 1.
Let $e = 0$, i.e. function $I(.,.,.)$ behaves like linear for *small* values of δx, δy, and δt

Simplifications

Assumption 2.
δx and δy model the motion of one pixel between t and $t+1$

Assumption 3.
Intensity constancy assumption (ICA)

$$I\left(x+\delta x, y+\delta y, t+\delta t\right)=I\left(x, y, t\right)$$

Results into

$$0=\delta x \cdot \frac{\partial I}{\partial x}\left(x, y, t\right)+\delta y \cdot \frac{\partial I}{\partial y}\left(x, y, t\right)+\delta t \cdot \frac{\partial I}{\partial t}\left(x, y, t\right)$$

Optic Flow Equation:

$$0=\frac{\delta x}{\delta t} \cdot \frac{\partial I}{\partial x}\left(x, y, t\right)+\frac{\delta y}{\delta t} \cdot \frac{\partial I}{\partial y}\left(x, y, t\right)+\frac{\partial I}{\partial t}\left(x, y, t\right)$$

Horn-Schunck Constraint

Take changes in x- and y-coordinate during δt as optic flow $\mathbf{u}(x, y, t) = (u(x, y, t), v(x, y, t))^\top$

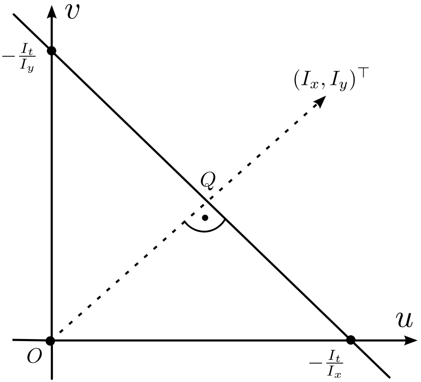
Horn-Schunck Constraint or **Optic Flow Equation**:

$$0 = u\,(x, y, t) \cdot \frac{\partial I}{\partial x}\,(x, y, t) + v\,(x, y, t) \cdot \frac{\partial I}{\partial y}\,(x, y, t) + \frac{\partial I}{\partial t}\,(x, y, t)$$

Short form:

$$0 = uI_x + vI_y + I_t$$

The *uv* Velocity Space



Straight line

$$-I_t = u \cdot I_x + v \cdot I_y = \mathbf{u} \cdot \nabla_{x,y} I$$

in *uv* velocity space, with optic flow vector $\mathbf{u} = [u, v]^\top$

Inner or Dot Vector Product

Vectors $\mathbf{a} = (a_1, a_2, \ldots, a_n)^\top$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)^\top$

Dot Product or **Inner Product**:

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$$

$$\mathbf{a} \cdot \mathbf{b} = ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2 \cdot \cos \alpha$$

$||\mathbf{a}||_2 = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$

α: angle between both vectors, $0 \leq \alpha < \pi$

What We Have So Far

I_x, I_y and I_t are estimated in given frames

(u, v) for pixel (x, y, t) is a point on the given straight line

But: where on this straight line?

Agenda

Labeling Model and First Labeling Constraint

Labeling Model:

Labeling function f assigns *label* (u, v) to all $p \in \Omega$ in $I(.,.,t)$

Possible set of vectors $(u, v) \in \mathbb{R}^2$ defines the set of labels

First Constraint for Labeling f:

data error or *data energy*

$$E_{data}(f) = \sum_{\Omega} [\ u \cdot I_x + v \cdot I_y + I_t\ ]^2$$

needs to be minimized, with $u \cdot I_x + v \cdot I_y + I_t = 0$ in the ideal case

Second Labeling Constraint

One option: motion constancy within pixel neighborhoods at time t

Smoothness Constraint, defined for

smoothness error or *smoothness energy*

$$E_{smooth}(f) = \sum_{\Omega} u_x^2 + u_y^2 + v_x^2 + v_y^2$$

where u_x is the 1^{st} order derivative of u with respect to x, and so forth.

The Optimization Problem

Calculate labeling function f which minimizes

$$E_{total}(f) = E_{data}(f) + \ \lambda \cdot E_{smooth}(f)$$

where $\lambda > 0$ is a weight

Example: $\lambda = 0.1$

Total Variation (TV):
Search for an optimum f in the set of al possible labelings

Error terms apply L_2-penalties, thus a TV-L_2 optimization problem

Least-Square Error Optimization

Least-square error (LSE) *optimization* follows a standard scheme:

1. Define an error or energy function.
2. Calculate derivatives of this function with respect to all the unknown parameters.
3. Set derivatives equal to zero and solve this equational system with respect to the unknowns. The result defines a minimum of the error function.

Task: LSE for Error Function Defined by Both Constraints

We assume that all pixels have all their four 4-adjacent pixels also in the image.

$$E_{data}(f) = \sum_{\Omega} [\ u \cdot I_x + v \cdot I_y + I_t\ ]^2$$

$$\begin{aligned} E_{smooth}(f) = \sum_{\Omega} \quad (\quad & u_{x+1,y} - u_{xy})^2 + (u_{x,y+1} - u_{xy})^2 \\ & + (v_{x+1,y} - v_{xy})^2 + (v_{x,y+1} - v_{xy})^2 \end{aligned}$$

Now we have all prepared for solving the LSE problem

Copyright Information

This slide show was prepared by Reinhard Klette
with kind permission from Springer Science+Business Media B.V.

The slide show can be used freely for presentations.
However, *all the material* is copyrighted.

In case of citation: just cite the book, that's fine.