Readings

• Szeliski, R. Ch. 7
• Bergen et al. ECCV 92, pp. 237-252.
• Shi, J. and Tomasi, C. CVPR 94, pp. 593-600.

• Slide Credits: Szeliski and Shah.
• New York Times
• Video from Michael Black
What is Optical Flow?

Optical flow is the relation of the motion field

- the 2D projection of the physical movement of points relative to the observer to 2D displacement of pixel patches on the image plane.

Common assumption:

The appearance of the image patches do not change (brightness constancy)

\[ I(p_i, t) = I(p_i + \vec{v}_i, t + 1) \]

Note: more elaborate tracking models can be adopted if more frames are processed all at once.
Optical flow constraints (grayscale images)

- Let’s look at these constraints more closely
  - brightness constancy: Q: what’s the equation?
    \[ H(x, y) = I(x+u, y+v) \]
  - small motion: (u and v are less than 1 pixel)
    - suppose we take the Taylor series expansion of I:
      \[
      I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \text{higher order terms}
      \]
      \[
      \approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v
      \]
Optical Flow Assumptions: Brightness Constancy

Assumption

Image measurements (e.g. brightness) in a small region remain the same although their location may change.

\[ I(x + u, y + v, t + 1) = I(x, y, t) \]

(assumption)

* Slide from Michael Black, CS143 2003
Optical Flow Assumptions:

Spatial Coherence

Assumption
* Neighboring points in the scene typically belong to the same surface and hence typically have similar motions.
* Since they also project to nearby points in the image, we expect spatial coherence in image flow.
Edge

- large gradients, all the same
- large $\lambda_1$, small $\lambda_2$

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Low texture region

- gradients have small magnitude
- small $\lambda_1$, small $\lambda_2$
High textured region

- gradients are different, large magnitudes
- large $\lambda_1$, large $\lambda_2$

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Estimating Optical Flow

\[ I(x, y, t) \approx I(x + dx, y + dy, t + dt) \]
Estimating Optical Flow

\[ I(x, y, t) \approx I(x + dx, y + dy, t + dt) \]

\[ I(x(t) + u \cdot \Delta t, y(t) + v \cdot \Delta t) - I(x(t), y(t), t) \approx 0 \]
Estimating Optical Flow

\[ I(x, y, t) \approx I(x + dx, y + dy, t + dt) \]

\[ I(x(t) + u \Delta t, y(t) + v \Delta t) - I(x(t), y(t), t) \approx 0 \]

Assuming \( I \) is differentiable function, and expand the first term using Taylor’s series:

\[ \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \]
Estimating Optical Flow

\[ I(x, y, t) \approx I(x + dx, y + dy, t + dt) \]

\[ I(x(t) + u.\Delta t, y(t) + v.\Delta t) - I(x(t), y(t), t) \approx 0 \]

Assuming \( I \) is differentiable function, and expand the first term using Taylor’s series:

\[ \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \]

Compact representation

\[ I_x u + I_y v + I_t = 0 \]

Brightness constancy constraint
Second Assumption: Gradient Constraint

Velocity vector is constant within a small neighborhood (LUCAS AND KANADE)

\[ E(u, v) = \int_{x,y} (I_x u + I_y v + I_t)^2 dxdy \]

\[ \frac{\partial E(u, v)}{\partial u} = \frac{\partial E(u, v)}{\partial v} = 0 \]

\[ 2(I_x u + I_y v + I_t)I_x = 0 \]

\[ 2(I_x u + I_y v + I_t)I_y = 0 \]
Recap: Lucas-Kanade

\[
\begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} = -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]
Recap: Lucas-Kanade

\[
\begin{bmatrix}
\sum I_x^2 & \sum I_x I_y \\
\sum I_x I_y & \sum I_y^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

Structural Tensor representation

\[
\begin{bmatrix}
T_{xx} & T_{xy} \\
T_{xy} & T_{yy}
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
T_{xt} \\
T_{yt}
\end{bmatrix}
\]
Recap: Lucas-Kanade

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Structural Tensor representation

$$\begin{bmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} T_{xt} \\ T_{yt} \end{bmatrix}$$

$$u = \frac{T_{yt} T_{xy} - T_{xt} T_{yy}}{T_{xx} T_{yy} - T^2_{xy}}$$  and  $$v = \frac{T_{xt} T_{xy} - T_{yt} T_{xx}}{T_{xx} T_{yy} - T^2_{xy}}$$
Recap: Pitfalls & Alternatives

- Brightness constancy is not satisfied
  - Correlation based method could be used
Recap: Pitfalls & Alternatives

• Brightness constancy is not satisfied
  – Correlation based method could be used
• A point may not move like its neighbors
  – Regularization based methods
Recap: Pitfalls & Alternatives

- Brightness constancy is not satisfied
  - Correlation based method could be used
- A point may not move like its neighbors
  - Regularization based methods
- The motion may not be small (Taylor does not hold!)
  - Multi-scale estimation could be used
Horn & Schunck

• Global method with smoothness constraint to solve aperture problem
Horn & Schunck

• Global method with smoothness constraint to solve aperture problem
• Minimize a global energy function
Horn & Schunck

• Global method with smoothness constraint to solve aperture problem

• Minimize a global energy function

\[ E(u, v) = \int_{x,y} [(I_x u + I_y v + I_t)^2 + \alpha^2 (|\nabla u|^2 + |\nabla v|^2)] dx dy \]

\[ E(u, v) = \int_{x,y} (I_x u + I_y v + I_t)^2 dx dy \]
Horn & Schunck

- Global method with smoothness constraint to solve aperture problem
- Minimize a global energy function

\[ E(u, v) = \int_{x,y} \left[ (I_x u + I_y v + I_t)^2 + \alpha^2 (\mid \nabla u \mid^2 + \mid \nabla v \mid^2) \right] dx dy \]

- Take partial derivatives w.r.t. u and v:
Horn & Schunck

- Global method with smoothness constraint to solve aperture problem
- Minimize a global energy function

\[ E(u, v) = \int_{x,y} [(I_x u + I_y v + I_t)^2 + \alpha^2 (|\nabla u|^2 + |\nabla v|^2)] dxdy \]

- Take partial derivatives w.r.t. \( u \) and \( v \):

\[
(I_x u + I_y v + I_t) I_x - \alpha^2 \nabla u = 0 \\
(I_x u + I_y v + I_t) I_y - \alpha^2 \nabla v = 0
\]
Horn & Schunck

- Iterative scheme

\[
 u^{k+1} = \bar{u}^k - \frac{I_x (I_x \bar{u}^k + I_y \bar{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}
\]

\[
 v^{k+1} = \bar{v}^k - \frac{I_y (I_x \bar{u}^k + I_y \bar{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}
\]

- Yields high density flow
- Fill in missing information in the homogenous regions
- More sensitive to noise than local methods
Optical Flow Matlab/C++/Python Code

- [https://github.com/Itseez/opencv_attic/blob/a6078cc8477ff055427b67048a95547b3efe92a5/opencv/samples/python2/lk_track.py](https://github.com/Itseez/opencv_attic/blob/a6078cc8477ff055427b67048a95547b3efe92a5/opencv/samples/python2/lk_track.py)
Applications: Target Tracking
Applications: Action Recognition

Recognition actions at a distance, Efros et al.
Applications: Motion Modeling

Flipping between image 1 and 2. Estimated flow field (hue indicates orientation and saturation indicates magnitude) Second image is warped!

*Slide credit to C.Liu*
Applications: Motion Segmentation
Global Motion Models (Parametric)

All pixels are considered to summarize global motion!

• **2D Models**
  – Affine
  – Quadratic
  – Planar projective (homography)

• **3D Models**
  – Inst. Camera motion models
  – Homography+epipole
  – Plane+parallalx
Motion Models

- Translation: 2 unknowns
- Affine: 6 unknowns
- Perspective: 8 unknowns
- 3D rotation: 3 unknowns
Example: Affine Motion

\[ u(x, y) = a_1 + a_2 x + a_3 y \]

\[ u(x, y) = a_4 + a_5 x + a_6 y \]
Example: Affine Motion

\[ u(x, y) = a_1 + a_2x + a_3y \]
\[ u(x, y) = a_4 + a_5x + a_6y \]

\[ I_x u + I_y v + I_t \approx 0 \]
Example: Affine Motion

\[ u(x, y) = a_1 + a_2 x + a_3 y \]

\[ u(x, y) = a_4 + a_5 x + a_6 y \]

\[ I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t \approx 0 \]
Example: Affine Motion

\[ u(x, y) = a_1 + a_2 x + a_3 y \]
\[ u(x, y) = a_4 + a_5 x + a_6 y \]

\[ I_x (a_1 + a_2 x + a_3 y) + I_y (a_4 + a_5 x + a_6 y) + I_t \approx 0 \]

Each pixel provides 1 linear constraint in **6 global unknowns** (a1,..,a6)
Example: Affine Motion

\[ u(x, y) = a_1 + a_2x + a_3y \]
\[ u(x, y) = a_4 + a_5x + a_6y \]

\[ I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y) + I_t \approx 0 \]

Each pixel provides 1 linear constraint in 6 global unknowns \((a_1, \ldots, a_6)\)

Over all pixels, minimize the least square to find unknowns!

\[ Err_{a_1, \ldots, a_6} = \sum [I_x(a_1 + a_2x + a_3y) + I_y(a_4 + a_5x + a_6y)]^2 \]
Other 2D Motion Models

- Quadratic
  \[ u = q_1 + q_2 x + q_3 y + q_7 x^2 + q_8 x y \]
  \[ u = q_4 + q_5 x + q_6 y + q_7 x y + q_8 y^2 \]

- Projective
  \[ u = \frac{h_1 + h_2 x + h_3 y}{h_7 + h_8 x + h_9 y} - x \]
  \[ v = \frac{h_4 + h_5 x + h_6 y}{h_7 + h_8 x + h_9 y} - y \]
FlowCap: 2D Human Pose from Optical FFlow
Questions?