Motion
(Part 2)

Computer Vision I

CSE 252A

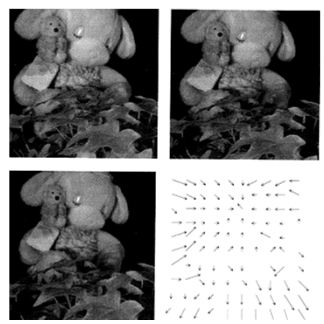
Lecture 16

Announcements

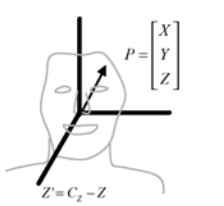
- Read Trucco and Verri
  - Course reserves
- Homework 3 is due Dec 4, 11:59 PM
- Homework 4 is coming
  - Due during finals week
- Please complete evaluations
  - Course
  - TA

Small Motion

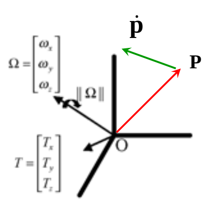
The Motion Field



Rigid Motion: General Case



Position and orientation of a rigid body
Rotation Matrix & Translation vector



Rigid Motion:
Velocity Vector: T
Angular Velocity Vector: ω (or Ω)

$$\dot{\mathbf{p}} = \mathbf{T} + \omega \times \mathbf{p}$$

General Motion

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z}\begin{bmatrix} x \\ y \end{bmatrix}$$

Let (x,y,z) be functions of time (x(t), y(t), z(t)):

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{f}{z}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} - \frac{f\dot{z}}{z^2}\begin{bmatrix} x \\ y \end{bmatrix}$$

$$= \frac{f}{z}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} - \frac{\dot{z}}{z}\begin{bmatrix} u \\ v \end{bmatrix}$$

Substitute $\dot{\mathbf{p}} = \mathbf{T} + \omega \times \mathbf{p}$ where $\mathbf{p}=(x,y,z)^T$

Motion Field Equation

$$\dot{u} = \frac{T_z u - T_x f}{Z} - \omega_y f + \omega_z v + \frac{\omega_x uv}{f} - \frac{\omega_y u^2}{f}$$

$$\dot{v} = \frac{T_z v - T_y f}{Z} + \omega_x f - \omega_z u - \frac{\omega_y uv}{f} - \frac{\omega_x v^2}{f}$$

- **T**: Components of 3-D linear motion
- ω: Angular velocity vector
- (u,v): Image point coordinates
- Z: depth
- f: focal length

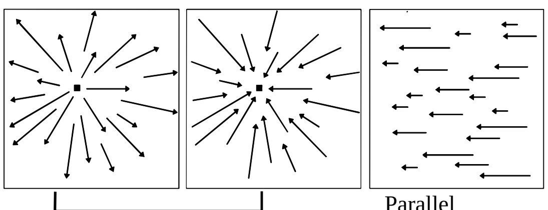
Pure Translation

$\omega = 0$

$$\dot{u} = \frac{T_z u - T_x f}{Z} - \omega_y f + \omega_z v + \frac{\omega_x uv}{f} - \frac{\omega_y u^2}{f}$$

$$\dot{v} = \frac{T_z v - T_y f}{Z} + \omega_x f - \omega_z u - \frac{\omega_y uv}{f} - \frac{\omega_x v^2}{f}$$

Pure Translation



Radial about FOE

Parallel (FOE point at infinity) $T_Z = 0$ Motion parallel to image plane

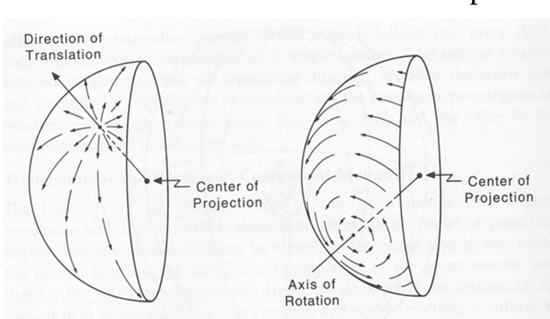
Pure Rotation: **T**=0

$$\dot{u} = \frac{T_z u - T_x f}{Z} - \omega_y f + \omega_z v + \frac{\omega_x uv}{f} - \frac{\omega_y u^2}{f}$$

$$\dot{v} = \frac{T_z v - T_y f}{Z} + \omega_x f - \omega_z u - \frac{\omega_y uv}{f} - \frac{\omega_x v^2}{f}$$

- Independent of T_x T_y T_z
- Independent of Z
- Only function of (u,v), f and ω

Pure Rotation: Motion Field on Sphere



Motion Field Equation: Estimate Depth

$$\dot{u} = \frac{T_z u - T_x f}{Z} - \omega_y f + \omega_z v + \frac{\omega_x uv}{f} - \frac{\omega_y u^2}{f}$$

$$\dot{v} = \frac{T_z v - T_y f}{Z} + \omega_x f - \omega_z u - \frac{\omega_y uv}{f} - \frac{\omega_x v^2}{f}$$

If **T**, ω, and f are known or measured, then for each image point (u,v), one can solve for the depth Z given measured motion (du/dt, dv/dt) at (u,v).

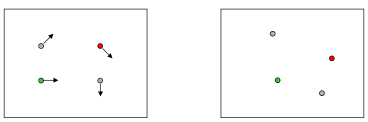
$$Z = \frac{T_z u - T_x f}{\dot{u} + \omega_y f - \omega_z v - \frac{\omega_x uv}{f} + \frac{\omega_y u^2}{f}}$$

Motion

Estimating the motion field from images

1. Feature-based (Sect. 8.4.2 of Trucco & Verri)
   1. Detect Features (corners) in an image
   2. Search for the same features nearby (Feature tracking).
2. Differential techniques (Sect. 8.4.1)

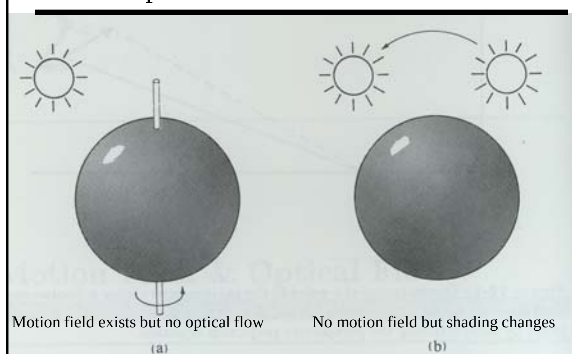
Problem Definition: Optical Flow



$H(x, y)$ $\quad$ $I(x, y)$

- How to estimate pixel motion from image H to image I?
  - Find pixel correspondences
    - Given a pixel in H, look for nearby pixels of the same color in I
- Key assumptions
  - **color constancy:** a point in H looks "the same" in image I
    - For grayscale images, this is **brightness constancy**
  - **small motion**: points do not move very far

Optical Flow ≠ Motion Field



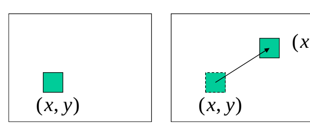
Motion field exists but no optical flow $\quad$ No motion field but shading changes

Definition of optical flow

OPTICAL FLOW = apparent motion of brightness patterns

Ideally, the optical flow is the motion field, i.e., the projection of the three-dimensional velocity vectors on the image

Optical Flow Constraint Equation



Optical Flow: Velocities (u, v)

Displacement:

$$(\delta x, \delta y) = (u\ \delta t, v\ \delta t)$$

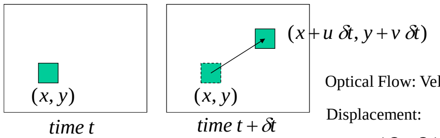
1. Assume brightness of patch remains same in both images:

$$I(x + u\ \delta t, y + v\ \delta t, t + \delta t) = I(x, y, t)$$

2. Assume small motion: (Taylor expansion of LHS up to first order)

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

Optical Flow Constraint Equation



Optical Flow: Velocities (u,v)

Displacement:

$$(\delta x, \delta y) = (u\,\delta t, v\,\delta t)$$

3. Subtracting I(x,y,t) from both sides and dividing by δt

$$\frac{\delta x}{\delta t}\frac{\partial I}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0$$

4. Assume small interval, this becomes:

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

Solving for flow

Optical flow constraint equation :

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

- We can measure $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}$
- We want to solve for $\frac{dx}{dt}, \frac{dy}{dt}$
- One equation, two unknowns

Aperture Problem and Normal Flow

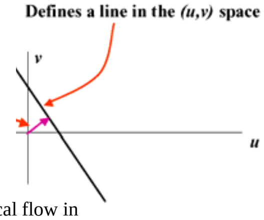
Measurements

$I_x = \frac{\partial I}{\partial x}$,
$I_y = \frac{\partial I}{\partial y}$,
$I_t = \frac{\partial I}{\partial t}$

The gradient constraint:

$$I_x u + I_y v + I_t = 0$$

Defines a line in the *(u,v)* space



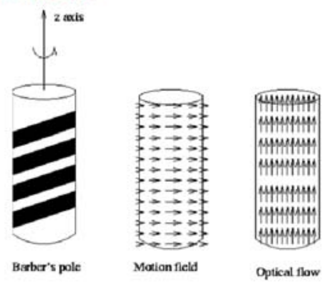
Flow vector

$u = \frac{dx}{dt}$,
$v = \frac{dy}{dt}$

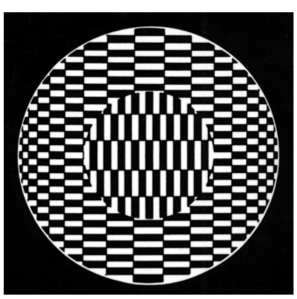
The component of the optical flow in the direction of the image gradient.

Optical Flow Constraint

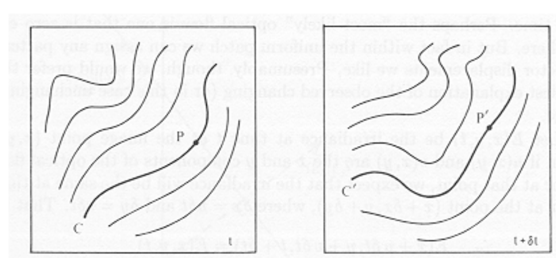
Barber pole illusion



Apparently an aperture problem

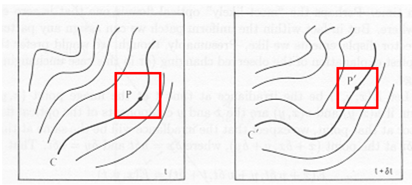


What is the correspondence of P & P'



Contour plots of image intensity in two images

Two ways to get flow



1. Think globally, and regularize over image

2. Look over window and assume constant motion in the window

Horn & Schunck algorithm

Additional smoothness constraint :

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))\,dxdy,$$

besides optical flow constraint equation term

$$e_c = \iint (I_x u + I_y v + I_t)^2\,dxdy,$$

minimize $e_s + \lambda e_c$

Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

$$E(u,v) = \sum_{x,y \in \Omega} \left(I_x(x,y)u + I_y(x,y)v + I_t\right)^2$$

$$\frac{dE(u,v)}{du} = \sum 2I_x\left(I_x u + I_y v + I_t\right) = 0$$

$$\frac{dE(u,v)}{dv} = \sum 2I_y\left(I_x u + I_y v + I_t\right) = 0$$

Solve with:

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

On the LHS: sum of the 2x2 outer product tensor of the gradient vector

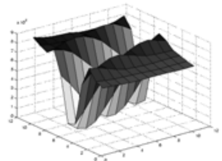
$$\left(\sum \nabla I \nabla I^T\right)\vec{U} = -\sum \nabla I I_t$$

Lucas-Kanade: Singularities and the Aperture Problem

Let $M = \sum (\nabla I)(\nabla I)^T$ and $b = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$

- Algorithm: At each pixel compute U by solving $MU = b$
- M is singular if all gradient vectors point in the same direction
  - -- e.g., along an edge
  - -- of course, trivially singular if the summation is over a single pixel
  - -- i.e., only *normal flow* is available (aperture problem)
- Corners and textured areas are OK

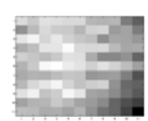
Edge

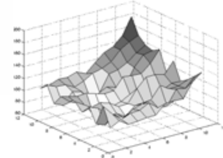


$\sum \nabla I (\nabla I)^T$

- large gradients, all the same
- large λ_1, small λ_2

Low texture region

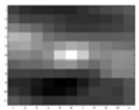


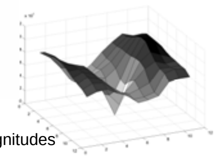


$\sum \nabla I (\nabla I)^T$

- gradients have small magnitude
- small λ_1, small λ_2

High textured region





$\sum \nabla I (\nabla I)^T$

– gradients are different, large magnitudes
– large λ_1, large λ_2

Some variants

- Coarse to fine (image pyramids)
- Local/global motion models
- Robust estimation

Iterative Refinement

- Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation
- Warp one image toward the other using the estimated flow field
  (easier said than done)
- Refine estimate by repeating the process

Revisiting the small motion assumption

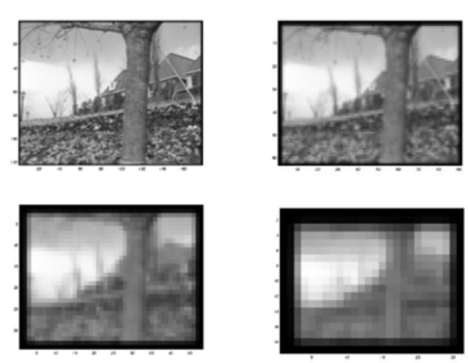
- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2nd order terms dominate)
  - How might we solve this problem?

Limits of the (local) gradient method

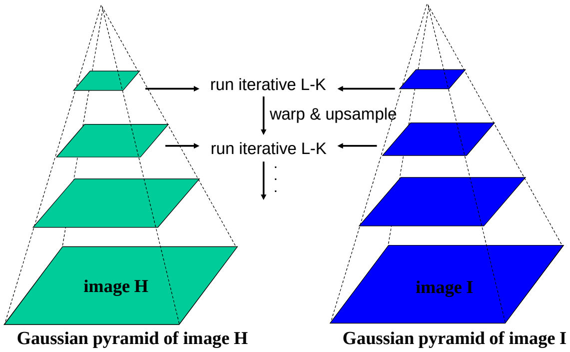
1. Fails when intensity structure within window is poor
2. Fails when the displacement is large (typical operating range is motion of 1 pixel per iteration!)
   – *Linearization of brightness is suitable only for small displacements*

Also, brightness is not strictly constant in images
   – *actually less problematic than it appears, since we can pre-filter images to make them look similar*

Pyramid / "Coarse-to-fine"

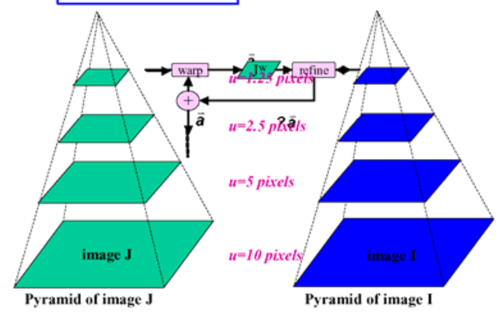


Coarse-to-fine optical flow estimation



Coarse-to-Fine Estimation

$I_x \cdot u + I_y \cdot v + I_t \approx 0$ **==> small *u* and *v* ...**



Multi-resolution Lucas Kanade Algorithm

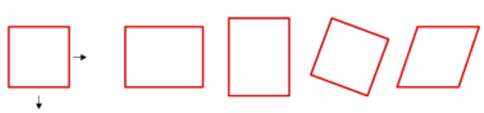
- Compute 'simple' LK at highest level
- At level i
  - Take flow u_{i-1}, v_{i-1} from level i-1
  - bilinear interpolate it to create u_i^*, v_i^* matrices of twice resolution for level i
  - multiply u_i^*, v_i^* by 2
  - compute f_t from a block displaced by $u_i^*(x,y)$, $v_i^*(x,y)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections u_i' v_i', *i.e.* $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$.

Parametric (Global) Motion Models

2D Models:
(Translation)
Affine
Quadratic
Planar projective transform (Homography)

3D Models:
Instantaneous camera motion models
Homography+epipole
Plane+Parallax

Motion Model Example: Affine Motion



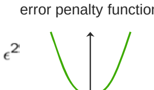
Affine: $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\mathbf{h} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$

Robust Estimation

Problem: Least-squares estimators penalize deviations between data & model with quadratic error f^n (extremely sensitive to outliers)

error penalty function | influence function

$\rho(\epsilon) = \epsilon^2$

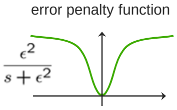


$\psi(\epsilon) = \frac{\partial \rho(\epsilon)}{\partial \epsilon} = 2\epsilon$

Redescending error functions (e.g., Geman-McClure) help to reduce the influence of outlying measurements.

error penalty function | influence function

$\rho(\epsilon; s) = \frac{\epsilon^2}{s + \epsilon^2}$



$\psi(\epsilon; s) = \frac{2\epsilon s}{(s + \epsilon^2)^2}$