Image Registration Lecture 21: Hierarchical Model-Based Motion Estimation

Prof. Charlene Tsai

Why Multi-Resolution

- There are several ways for constructing image pyramid
  - Laplacian pyramid
  - Gaussian pyramid
- Reasons for doing multi-resolution
  - For efficiency
  - For robustness

Categories of Motion Models

- Parametric (global)
  - Affine
  - Projective
- Non-parametric (local)
  - Effective over small blocks of pixels
  - Often need some type of a smoothness or uniformity constraint
- Quasi-parametric (obselete)
  - Between full- and non-parametric

Brightness Constancy Constraint

- Assumption: Projection of a point in the world will produce the same intensity value in each image of the motion sequence.
- This implies:

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

- Assuming the I is differentiable. Taking the 1st order Taylor expansion of the RHS, we get

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

(cont)

- Neglecting h.o.t., we get

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

- Setting $\vec{u}=(dx/dt, dy/dt)^T$ we get

velocity

$$\vec{u}\bullet\nabla I+I_t=0$$

(Optical flow constraint eqn)

I_t: $\frac{\partial I}{\partial t}$

- One equation two unknowns => under-constrained (aperture problem)

Optical Flow Field (local)

- Assuming $\vec{u}$ to be locally constant, approximate $\vec{u}$ at a single location $\vec{x}_0$ from the neighborhood, i.e. least-squares formulation

$$E(\vec{u})=\sum_{\vec{x}\in N(\vec{x}_0)}(\vec{u}\bullet\nabla I+I_t)^2 \qquad \text{Eqn\#1}$$

- For better robustness, we can use weighted-least-squares instead

$$E(\vec{u})=\sum_{\vec{x}\in N(\vec{x}_0)}w(\vec{x})(\vec{u}\bullet\nabla I+I_t)^2$$

Affine Motion Model (global)

- Using affine motion model (higher order), we're able to describe motion of a larger region R:

$$\vec{u} = A\vec{x} + \vec{t} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = X(\vec{x})\vec{a}$$

where $X(\vec{x}) = \begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix}$ and

$\vec{a} = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad t_x \quad t_y]^T$

(cont)

- Replacing $\vec{u}$ with $X\vec{a}$, and plug into equn#1 we get

$$E(\vec{a}) = \sum_{\vec{x} \in R} \left(\nabla I^T X\vec{a} + I_t\right)^2$$

- The solution is given by

$$\frac{\partial E}{\partial \vec{a}} = 2\sum_{\vec{x} \in R} \left(\nabla I^T X\vec{a} + I_t\right)\left(X^T \nabla I\right)$$

$$= 2\sum_{\vec{x} \in R} \left(X^T(\nabla I)(\nabla I)^T X\vec{a} + X^T(\nabla I)I_t\right) = 0$$

(cont)

$$\vec{a} = -\left(\sum_{\vec{x}\in R} X^T(\nabla I)(\nabla I)^T X\right)^{-1}\left(\sum_{\vec{x}\in R} X^T(\nabla I)I_t\right)$$

- This is not a close form!
- Why?

Overall Algorithm for Motion Estimation

- Step1: Building Laplacian pyramid
- Step2: For each resolution, starting at coarsest
  - Apply current motion to $I_{(t-\Delta t)}$ (deforming image) to get $I'_{(t-\Delta t)}$
  - Estimate new affine motion between $I_{(t)}$ and $I'_{(t-\Delta t)}$
  - Compose new affine estimate with current estimate
  - Iterate until converge.