Motion

Computer Vision I
CSE252A
Lecture 18

Announcements

• TA evaluations today
• HW 4 due Friday at midnight
• Will go over final on thurs.

Some problems of motion
1. Correspondence: Where have elements of the image moved between image frames
2. Reconstruction: Given correspondence, what is the 3-D geometry of the scene
3. Segmentation: What are regions of image corresponding to different moving objects
4. Tracking: Where have objects moved in the image? related to correspondence and segmentation.

Variations:
- Small motion (video),
- Wide-baseline (multi-view)

Rigid Motion: General Case

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

\[ \mathbf{p} = \mathbf{T} + \mathbf{\omega} \times \mathbf{p} \]

Motion Field Equation

\[ \begin{align*}
    u &= \frac{T_{x}u - T_{f}f}{Z} - \omega_{z}f + \omega_{x}v + \frac{\omega_{x}uv}{f} - \frac{\omega_{y}u^{2}}{f} \\
    v &= \frac{T_{y}v - T_{f}f}{Z} + \omega_{x}f - \omega_{y}u - \frac{\omega_{y}uv}{f} + \frac{\omega_{y}v^{2}}{f}
\end{align*} \]

- \( \mathbf{T} \): Components of 3-D linear motion
- \( \mathbf{\omega} \): Angular velocity vector
- \((u,v)\): Image point coordinates
- \( Z \): depth
- \( f \): focal length

Optical Flow Constraint Equation

\[ \begin{align*}
    \mathbf{I}(x+u \delta t, y+v \delta t, t+\delta t) &= \mathbf{I}(x,y,t) \\
    \delta x \frac{\partial I}{\partial x} + \delta y \frac{\partial I}{\partial y} + \delta t \frac{\partial I}{\partial t} &= \mathbf{I}(x,y,t)
\end{align*} \]

- Assume brightness of patch remains same in both images:
- Assume small motion: (Taylor expansion of LHS up to first order)
Optical Flow Constraint Equation

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

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 \( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \)
- We want to solve for \( \frac{dx}{dt}, \frac{dy}{dt} \)
- One equation, two unknowns

Two ways to get flow

1. Think globally, and regularize over image
2. Look over window and assume constant motion in the window

Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

\[ E(u,v) = \sum_{(x,y)} \left( I(x,v) + I(x,v) - I(x,v) \right) \]

\[ \frac{dE(u,v)}{du} - \sum_{(x,y)} \left( I(x,y) + I(x,y) - I(x,y) \right) = 0 \]

\[ \frac{dE(u,v)}{dv} - \sum_{(x,y)} \left( I(x,y) + I(x,y) - I(x,y) \right) = 0 \]

Solve with:

\[ \left( \sum_{(x,y)} I(x,v), \sum_{(x,y)} I(x,v) \right) = -\left( \sum_{(x,y)} I(x,v), \sum_{(x,y)} I(x,v) \right) \]

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

\[ \sum \nabla I(x,v)^T \]
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 $\lambda_2$

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
**Coarse-to-fine optical flow estimation**

- Run iterative L-K on image H.
- Warp and upsample.
- Gaussian pyramid of image H.
- Repeat for image I.

**Multi-resolution Lucas Kanade Algorithm**

- Compute ‘simple’ LK at highest level.
- At level $i$:
  - Take flow $u^*, v^*$ from level $i-1$.
  - Bilinear interpolate it to create $u^*, v^*$ matrices of twice resolution for level $i$.
  - Multiply $u^*, v^*$ by 2.
  - Compute $f$ from a block displaced by $u^*, v^*$.
  - Apply LK to get $u^*, v^*$.
  - Add corrections $u^*, v^*$.

**Motion Model Example: Affine Motion**

- Affine: $\Lambda = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\mathbf{h} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$.

**Parametric (Global) Motion Models**

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

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

**Robust Estimation**

- Quadratic $\theta$ function gives too much weight to outliers.

- $\rho(r, \sigma) = \frac{r^2}{\sigma^2 + r^2}$
- $\Psi(r, \sigma) = \frac{2\sigma^2}{(\sigma^2 + r^2)^2}$
Main Challenges

1. 3-D Pose Variation
2. Occlusion of the target
3. Illumination variation
4. Camera jitter
5. Expression variation
etc.

Ho, Lee, Kriegman

Main tracking notions

• State: usually a finite number of parameters (a vector) that characterizes the “state” (e.g., location, size, pose, deformation of thing being tracked).
• Dynamics: How does the state change over time? How is that changed constrained?
• Representation: How do you represent the thing being tracked
• Prediction: Given the state at time $t-1$, what is an estimate of the state at time $t$?
• Correction: Given the predicted state at time $t$, and a measurement at time $t$, update the state.
• Initialization – what is the state at time $t=0$?

What is state?

• 2-D image location, $\Phi=(u,v)$
• Image location + scale $\Phi=(u,v,s)$
• Image location + scale + orientation $\Phi=(u,v,s,\theta)$
• Affine transformation
• 3-D pose
• 3-D pose plus internal shape parameters (some may be discrete).
• e.g., for a face, 3-D pose + facial expression using FACS + eye state (open/closed).
• Collections of control points specifying a spline
• Above, but for multiple objects (e.g. tracking a formation of airplanes).
• Augment above with temporal derivatives $(\dot{\Phi}, \ddot{\Phi})$

State Examples:

– object is ball, state is 3D position+velocity, measurements are stereo pairs
– object is person, state is body configuration, measurements are frames
– What is state here?
Example: Blob Tracker

- From input image $I(u,v)$ (color?) at time $t$, create a binary image by applying a function $f(I(u,v))$.
- Clean up binary image using morphological operators.
- Perform connected component exploration to find “blobs.” – connected regions.
- Compute their moments (mean and covariance of coordinates of region), and use as state.
- Using state estimate from $t-1$ and perform “data association” to identify state in from $t$.

Blob Tracking in IR Images

- Threshold about body temperature
- Connected component analysis
- Position, scale, orientation of regions
- Temporal coherence