Vision Review: Motion & Estimation

Course web page:
www.cis.udel.edu/~cer/arv
Announcements

- Homework 1 graded
- Homework 2 due next Tuesday
- Papers: Students without partners should go ahead alone, with the write-up only 2 pages and the presentation 15 minutes long
Computer Vision
Review Outline

- Image formation
- Image processing
- Motion & Estimation
- Classification
Outline

- **Multiple views** (Chapter on this in Hartley & Zisserman is online)
  - Epipolar geometry
  - Structure estimation
- **Optical flow**
- **Temporal filtering**
  - Kalman filtering for tracking
  - Particle filtering
Two View Geometry

- Stereo or one camera over time
- Epipolar geometry
- Fundamental Matrix
  - Properties
  - Estimating
Epipolar Geometry

- **Epipoles**: Where baseline intersects image planes
- **Epipolar plane**: Any plane containing baseline
- **Epipolar line**: Intersection of epipolar plane with image plane

![Diagram of epipolar geometry]
Example: Epipolar Lines

Known epipolar geometry constrains search for point correspondences

from Hartley & Zisserman
Focus of Expansion

- Epipoles coincide for pure translation along optical axis
- Not the same as vanishing point

from Hartley & Zisserman
The Fundamental Matrix $\mathbf{F}$

- Maps points in one image to their epipolar lines in another image for uncalibrated cameras
- Definition: $\mathbf{x}^T \mathbf{F} \mathbf{x} = 0$ ; $3 \times 3$, rank 2, not invertible
- Essential matrix $\mathbf{E}$: Fundamental matrix when calibration matrices known:

$$\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}$$
Estimating $F$

- Same general approach as DLT method for homography estimation
- Need 8 point correspondences for linear method
- Normalization/denormalization
  - Translate, scale image so that centroid of points is at origin, RMS distance of points to origin is $\sqrt{2}$
- Enforce singularity constraint
- Degeneracies
  - Points related by homography
  - Points and camera on ruled quadric (one hyperboloid, two planes/cones/cylinders)
Structure from Motion (SFM)

- Camera matrices $\mathbf{P}$, $\mathbf{P}'$ can be computed from $\mathbf{E}$, from which we can triangulate to deduce 3-D locations.

- Limits
  - Uncalibrated camera(s): Best we can do is reconstruction up to a projection.
  - Calibrated camera(s): Can reconstruct up to a similarity transform (i.e., could be a house 10 m away or a dollhouse 1 m away).
Reconstruction Ambiguities

Projective reconstruction
Affine reconstruction
Metric reconstruction

Two views

from Hartley & Zisserman
More Than Two Views

• Analogues of the fundamental matrix:
  – Trifocal tensor: 3 views
  – Quadrifocal tensor: 4 views

• Reconstruction methods
  – Bundle adjustment: Projective reconstruction from $n$ views taking all into account simultaneously
  – Factorization: Affine reconstruction for $n$ affine cameras (Tomasi & Kanade, 1992)

from Hartley & Zisserman
SFM from Sequences

• Feature tracking makes point correspondences easier

• Problems
  – Small baseline between successive images—only compute structure at intervals
  – Forward translation not good for structure estimation because rays to points nearly parallel
  – Many methods batch → Must have all frames before computing
Szeliski’s Projective Depth, Revisited

- Approach: Decompose motion of scene points into two parts:
  - 2-D homography (as if all points coplanar)
  - Plane-induced parallax
- Signed distance $\rho$ along epipolar line from point to where it would be on homography plane is parallax relative to $H$
- Parallax is proportional to 3-D distance from plane—the *projective depth*

from Hartley & Zisserman
Plane-Induced Parallax

Left view superimposed on right using homography induced by plane of paper

from Hartley & Zisserman
Differential Motion:
Dense Flow

- **Scene flow**: 3-D velocities of scene points: Derivative of rigid transformation between views with respect to time
- **Motion field**: 2-D projection of scene flow
- **Optical flow**: Approximation of motion field derived from apparent motion of image points
Brightness Constancy Assumption

- Assume pixels just move—i.e., that they don’t appear and disappear. This is equivalent to $\frac{dI(x, y, t)}{dt} = 0$, which by the chain rule yields:

$$I_{xu} + I_{yu} + I_t = 0$$

- Caveats
  - Lighting may change
  - Objects may reflect differently at different angles
Optical Flow

- Aperture problem: Can only determine optical flow component in gradient direction.
- Brightness constancy insufficient to solve for general optical flow vector field, so other constraints necessary:
  - Assume flow field is smoothly varying (Horn, 1986)
  - Assume low-dimensional function describes motion
    - Swinging arm, leg (Yamamoto & Koshikawa, 1991; Bregler, 1997)
    - Turning head (Basu, Essa, & Pentland, 1996)
Example: Optical Flow

Best estimates where there are "corners"
Optical Flow for Time-to-Collision

- When will object we are headed toward (or one headed toward us) be at $Z = 0$?
- If object is at depth $Z_{obj}$ and the $Z$ component of the robot’s translational velocity is $t_Z$, then $TTC = Z_{obj}/t_Z$
- Divergence of a vector field $\mathbf{O}$ is defined as
  $$\text{div}(\mathbf{O}) = \nabla \cdot \mathbf{O} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$
- From motion field definition, we can show that $\text{div}(\mathbf{O}) = 2/TTC$ (Coombs et al., 1995)
Sparse Differential Motion: Feature Tracking

- Idea: Ignore everything but “corners”
- Feature detection, disappearance
- Tracking = Estimation over time + correspondence

- Tracking
  - Kalman Filter
    - Data association techniques: PDAF, JPDAF, MHF
  - Particle Filters
    - Stochastic estimation
Optimal Linear Estimation

• Assume: Linear system with uncertainties
  – State $\mathbf{x}$
  – Dynamical (system) model: $\mathbf{x} = \Phi \mathbf{x}_{t-1} + \xi$
  – Measurement model: $\mathbf{z} = \mathbf{Hx} + \mu$
  – $\xi, \mu$ indicate white, zero-mean, Gaussian noise with covariances $\mathbf{Q}, \mathbf{R}$ respectively

• Want best state estimate at each instant
Estimation variables

- Typical parameters in state $\mathbf{x}$:
  - Measurement-type parameters that we want to smooth
  - Time variables: Velocity, acceleration
  - Derived quantities: Depth, shape, curvature

- Measurement $\mathbf{z}$: What can be seen in one image
  - Position, orientation, scale, color, etc.

- Noise
  - $\mathbf{Q}, \mathbf{R}$: Set from real data if possible, but ad-hoc numbers may work
Kalman Filter

- Essentially an online version of least squares
- Provides best linear unbiased estimate

\[
\begin{align*}
\hat{x} &= \Phi x_{t-1} & \text{Predicted state} \\
\hat{z} &= H\hat{x} & \text{Predicted measurement} \\
\hat{P} &= \Phi P_{t-1}\Phi^T + Q & \text{State prediction covariance} \\
S &= H\hat{P}H^T + R & \text{Measurement prediction covariance} \\
\nu &= z - \hat{z} & \text{Innovation} \\
K &= \hat{P}H^TS^{-1} & \text{Filter gain} \\
x &= \hat{x} + Kn & \text{State estimate} \\
P &= (I - KH)\hat{P} & \text{State covariance estimate}
\end{align*}
\]
Example: 2-D position, velocity

- State \( \mathbf{x} = [x, y, \dot{x}, \dot{y}]' \)
- Observation \( \mathbf{z} = [x, y]' \)
- Dynamics \( \Phi = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \)
- Measurement \( \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \)
Example: 2-D position, velocity
Kalman-estimated states
courtesy of K. Murphy
Finding Measurements in Images

- Look for peaks in template-match function; most recent state estimate suggests where to search
- Gradient ascent [Shi & Tomasi, 1994; Terzopoulos & Szeliski, 1992]
  - Identifies nearby, good hypothesis
  - May pick incorrectly when there is ambiguity
  - Vulnerable to agile motions
- Random sampling [Isard & Blake, 1996]
  - Approximates local structure of image likelihood
  - Identifies alternatives
  - Resistant to agile motions
Handling Nonlinear Models

• Many system & measurement models can’t be represented by matrix multiplications (e.g., sine function for periodic motion)

• Kalman filtering with nonlinearities
  – Extended Kalman filter
    • Linearize nonlinear function with 1st-order Taylor series approximation at each time step
  – Unscented Kalman filter
    • Approximate distribution rather than nonlinearity
    • More efficient and accurate to 2nd-order
    • See http://cslu.ece.ogi.edu/nasel/research/ukf.html
Particle Filters

- Stochastic sampling approach for dealing with non-Gaussian posteriors
- Efficient, easy to implement, adaptively focuses on important areas of state space
- More on Thursday
Homework 2

- Implement a planar SSD template tracker using the Kalman filter to estimate homography at each time step.
- Given a sequence of a street sign in motion and a picture of it as a template.
- Manually initialize first frame, but must automatically extract measurements thereafter.
Template & Sequence
Kalman Filter Toolbox

• Web site: www.cs.berkeley.edu/~murphyk/Bayes/kalman.html

• Just need to plug correct parameters into the `kalman_update` function
Nonlinear Minimization in Matlab

- Function `lsqnonlin`
- Must write evaluation function `func` for `lsqnonlin` to call that returns a scalar (smaller numbers better)
- Example:

```matlab
% define 'func' with two parameters a & b
% set X0
opts = optimset('LevenbergMarquardt', 'on');
X = lsqnonlin('func', X0, [], [], [], opts, a, b);
```