Motion

Introduction to Computer Vision
CSE 152
Lecture 20

Announcements

- HW 4 due Friday at Midnight
- Final Exam: Tuesday, 6/12 at 8:00AM-11:00AM, regular classroom
- Extra Office Hours: Monday 6/11 9:00AM-10:00AM
- Fill out your CAPES

CSE152, Spring 2012
Intro Computer Vision

Motion

What are problems that we solve using motion
1. Correspondence: Where have elements of the image moved between image frames
2. Ego Motion: How has the camera moved.
3. Reconstruction: Given correspondence, what is 3-D geometry of scene
4. Segmentation: What are regions of image corresponding to different moving objects
5. Tracking: Where have objects moved in the image? related to correspondence and segmentation.

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

Structure-from-Motion (SFM)

Goal: Take as input two or more images or video w/o any information on camera position/motion, and estimate camera position and 3-D structure of scene.

Two Approaches
1. Discrete motion (wide baseline)
   1. Orthographic (affine) vs. Perspective
   2. Two view vs. Multi-view
   3. Calibrated vs. Uncalibrated
2. Continuous (Infinitesimal) motion

Discrete Motion: Some Counting

Consider \( M \) images of \( N \) points, how many unknowns?
1. Camera locations: Affix coordinate system to location of first camera location: \((M-1)*6\) Unknowns
2. 3-D Structure: \(3*N\) Unknowns
3. Can only recover structure and motion up to scale. Why?

Total number of unknowns: \((M-1)*6+3*N-1\)
Total number of measurements: \(2*M*N\)
Solution is possible when \((M-1)*6+3*N-1 \leq 2*M*N\)
\( M=2 \Rightarrow N \geq 5 \)
\( M=3 \Rightarrow N \geq 4 \)

The Eight-Point Algorithm (Longuet-Higgins, 1981)

Let \( F \) denote the Essential Matrix here

\[
\begin{bmatrix}
(a',a,b,1) & F_{11} & F_{12} & F_{13} \\
F_{21} & F_{22} & F_{23} \\
F_{31} & F_{32} & F_{33}
\end{bmatrix}
\]

\[
F_1F_2F_3 = 0
\]

Set \( F_{33} \) to 1

Solve For \( F \)

Solve For \( R \) and \( t \)
Sketch of Two View SFM Algorithm

Input: Two images
1. Detect feature points
2. Find 8 matching feature points (easier said than done, but usually done with RANSAC Algorithm)
3. Compute the Essential Matrix E using Normalized 8-point Algorithm
4. Compute R and T (recall that E=RS where S is skew symmetric matrix)
5. Perform stereo matching using recovered epipolar geometry expressed via E.
6. Reconstruct 3-D geometry of corresponding points.

Continuous Motion

- Consider a video camera moving continuously along a trajectory (rotating & translating).
- How do points in the image move
- What does that tell us about the 3-D motion & scene structure?

Motion

“When objects move at equal speed, those more remote seem to move more slowly.”
- Euclid, 300 BC

Background Subtraction

- Gather image I(x,y,t₀) of background without objects of interest (perhaps computed over average over many images).
- At time t, pixels where |I(x,y,t)-I(x,y,t₀)| > τ are labeled as coming from foreground objects

The Motion Field

Where in the image did a point move?

Down and left
Motion Field

What causes a motion field?
1. Camera moves (translates, rotates)
2. Object’s in scene move rigidly
3. Objects articulate (pliers, humans, quadripeds)
4. Objects bend and deform (fish)
5. Blowing smoke, clouds

Motion Field Yields 3-D Motion Information
The “instantaneous” velocity of points in an image
LOOMING
The Focus of Expansion (FOE)
Intersection of velocity vector with image plane

Is motion estimation inherent in humans?
Demo

Rigid Motion and the Motion Field
How do we relate 3-D motion of a camera to the 2-D motion of points in the image

Rigid Motion: General Case
Position & Orientation
\[
\dot{p} = T + \Omega \times p
\]
Position and orientation of a rigid body
Rotation Matrix & Translation vector
General Motion

\[
\begin{bmatrix}
\dot{u} \\
\dot{v}
\end{bmatrix} = \frac{f}{z} \begin{bmatrix}
x \\
y
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{\dot{u}} \\
\dot{\dot{v}}
\end{bmatrix} = \frac{f}{z} \begin{bmatrix}
x' \\
y'
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{\dot{u}} \\
\dot{\dot{v}}
\end{bmatrix} = \frac{f}{z} \begin{bmatrix}
x' \\
y'
\end{bmatrix}
\]

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

Motion Field Equation

\[
\dot{u} = \frac{T_u - T_f}{Z} - \omega_x f + \omega_y v + \frac{\omega_y v f}{f} - \frac{\omega_x u^2}{f}
\]

\[
\dot{v} = \frac{T_v - T_f}{Z} + \omega_x f - \omega_y u - \frac{\omega_y v f}{f} - \frac{\omega_x v^2}{f}
\]

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

Pure Translation

\[
\dot{u} = \frac{T_u - T_f}{Z} - \omega_x f + \omega_y v + \frac{\omega_y v f}{f} - \frac{\omega_x u^2}{f}
\]

\[
\dot{v} = \frac{T_v - T_f}{Z} + \omega_x f - \omega_y u - \frac{\omega_y v f}{f} - \frac{\omega_x v^2}{f}
\]

\( \omega = 0 \)

Forward Translation & Focus of Expansion

[Gibson, 1950]

Pure Rotation: \( T=0 \)

\[
\dot{u} = -\frac{T_u - T_f}{Z} - \omega_x f + \omega_y v + \frac{\omega_y v f}{f} - \frac{\omega_x u^2}{f}
\]

\[
\dot{v} = -\frac{T_v - T_f}{Z} + \omega_x f - \omega_y u - \frac{\omega_y v f}{f} - \frac{\omega_x v^2}{f}
\]

- Independent of \( T_x, T_y, T_z \)
- Independent of \( Z \)
- Only function of \( (u,v), f \) and \( \omega \)
Rotational MOTION FIELD
The “instantaneous” velocity of points in an image

PURE ROTATION
\[ \omega = (0,0,1)^T \]

Motion Field Equation: Estimate Depth

\[ \begin{align*}
\dot{u} &= \frac{T_x T_x f}{Z} - \omega_y f + \omega_y v + \frac{\omega_y}{f} \\
\dot{v} &= \frac{T_y T_y f}{Z} + \omega_x f - \omega_x u - \frac{\omega_x^2}{f} \\
\end{align*} \]

If \( T, \omega, \) 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 \((u,v)\).

Optical Flow:
Where do pixels move to?

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)
Mathematical formulation

\[ I(x, y; t) = \text{brightness at image point } (x, y) \text{ at time } t \]

Consider scene (or camera) to be moving, so \((x, y)\) is a function of time (i.e., \(x(t), y(t)\)), and point is moving with velocity \((dx/dt, dy/dt)\)

**Brightness constancy assumption:**

\[ I(x + \frac{dx}{dt}, y + \frac{dy}{dt}, t + \frac{dt}{dt}) = I(x, y, t) \Rightarrow \frac{dI}{dt} = 0 \]

**Optical flow constraint equation:**

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

---

Normal Flow

Illusion Works Barber Pole Illusion

---

Optical Flow Constraint

Barber pole illusion

---

Final Exam

- Closed book
- No Calculator
- One cheat sheet
  - Single piece of paper, handwritten, no photocopying, no physical cut & paste.
  - You can start with your sheet from the midterm.
- What to study
  - Basically material presented in class, and supporting material from text
  - If it was in text, but NEVER mentioned in class, it is very unlikely to be on the exam
- Question style:
  - Short answer
  - Some longer problems to be worked out.