Motion

Computer Vision I
CSE252A
Lecture 17

Some problems of motion
1. Correspondence: Where have elements of the image moved between image frames
2. Reconstruction: Given correspondence, what is 3-D geometry of 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)

Is motion estimation inherent in humans?
Demo

The Motion Field

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

Rigid Motion: General Case

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

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

Rigid Motion:
Velocity Vector: T
Angular Velocity Vector: \( \omega \) (or \( \Omega \))
General Motion

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

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

\[
\begin{bmatrix}
  \dot{u} \\
\dot{v}
\end{bmatrix} = \frac{f \begin{bmatrix}
x' \\
y'
\end{bmatrix}}{z} - \frac{f^2 \begin{bmatrix}
x'' \\
y''
\end{bmatrix}}{z^3}
\]

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

Motion Field Equation

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

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

Pure Translation

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

\( \omega = 0 \)

Pure Rotation: \( T = 0 \)

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

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

THE MOTION FIELD

The “instantaneous” velocity of points in an image

LOOMING

The Focus of Expansion (FOE)

Intersection of velocity vector with image plane

With just this information it is possible to calculate:
1. Direction of motion
2. Time to collision
3. Distance from FOE

Rotational MOTION FIELD

The “instantaneous” velocity of points in an image

PURE ROTATION

\[
\begin{align*}
\omega &= (0, 0, 1)^T \\
\dot{u} &= +\omega_2 v \\
\dot{v} &= -\omega_2 u
\end{align*}
\]
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:

\[ \frac{dI}{dt} = 0 \]

Optical flow constraint equation:

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

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

\[ \lambda_1 = \frac{dI}{dy} \]
\[ \lambda_2 = \frac{dI}{dx} \]
\[ \lambda_3 = \frac{dI}{dt} \]

The gradient constraint:

\[ I_u + I_v + I_t = 0 \]
\[ \nabla I \cdot \vec{U} = 0 \]

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

Flow vector

\[ u = \frac{dx}{dt}, \quad v = \frac{dy}{dt} \]

Normal Flow:

\[ \lambda_1 = \frac{I}{\sqrt{I^2 + \lambda_1^2}} \]

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

Lucas-Kanade: Integrate over a Patch

Assume a single velocity for all pixels within an image patch

\[ E(u, v) = \sum_{i \in \Omega} \left( I_i (x, y) u + I_i (x, y) v + I_i \right)^2 \]

\[ \frac{dE(u, v)}{du} = \sum 2I_i (I u + I v + I) = 0 \]
\[ \frac{dE(u, v)}{dv} = \sum 2I_i (I u + I v + I) = 0 \]

Solve with:

\[ \left( \sum I_i^2, \sum I_i I_x, \sum I_i I_y \right) \mu = \left( \sum I_i^2, \sum I_i I_x, \sum I_i I_y \right) \]

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

\[ \sum \nabla I I \cdot \vec{U} = -\sum \nabla I \]
Lucas-Kanade: Singularities and the Aperture Problem

Let \( M = \sum \nabla I_1 \nabla I_1^T \) and \( b = -\sum \nabla I_1 \).

- 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.

\[ \sum \nabla I \nabla I^T \]

- large gradients, all the same
  - large \( \lambda_1 \), small \( \lambda_2 \)

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
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

Revisiting the small motion assumption

- Is this motion small enough?
  - Probably not—it’s much larger than one pixel (2nd order terms dominate)

Pyramid / “Coarse-to-fine”

Coarse-to-fine optical flow estimation

- Run iterative L-K
- Warp & upsample

Multi-resolution Lucas Kanade Algorithm

- Compute ‘simple’ LK at highest level
- At level \(i\)
  - Take flow \(u_i, v_i\) from level \(i+1\)
  - Bilinear interpolate \(u, v\) to create \(u_i, v_i\), matrices of twice resolution for level \(i\)
  - Multiply \(u_i, v_i\) by 2
  - Compute \(f_i\) from a block displaced by \(u_i, v_i\)
  - Apply LK to get \(u_j, v_j\) (the correction in flow)
  - Add corrections \(u_j, v_j\), i.e., \(u_{j+1} = u_j + u_{j+1}\), \(v_{j+1} = v_j + v_{j+1}\)

Coarse-to-fine optical flow estimation

- Gaussian pyramid of image \(H\)
- Gaussian pyramid of image \(I\)
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: \[
A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad h = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}
\]

Robust Estimation

Quadratic \( \theta \) function gives too much weight to outliers.

\[
\rho(r, \sigma) = \frac{r^2}{\sigma^2 + r^2}, \quad \psi(r, \sigma) = \frac{2r\sigma^2}{(\sigma^2 + r^2)^2}
\]