Optical Flow

Computer Vision I
CSE252A
Lecture 17

Announcements
• Homework 4 Posted
• Today: Tracking
• Reading: Sections 17.1-17.3

Stereo Assignment

Helmholtz reciprocity Stereo

\[ \rho(\theta_{in}, \phi_{in}; \theta_{out}, \phi_{out}) = \rho(\theta_{out}, \phi_{out}; \theta_{in}, \phi_{in}) \]

[Helmholtz, 1910], [Minnaert, 1941], [Nicodemus et al, 1977]

Motion
Rigid Motion: General Case

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

Rigid Motion Equation

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

Motion Field Equation

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

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

Pure Translation

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

\( \omega = 0 \)

Pure Rotation: \( \mathbf{T} = 0 \)

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

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

Definition of optical flow

OPTICAL FLOW = apparent motion of brightness patterns

Ideally, the optical flow is the projection of the three-dimensional velocity vectors on the image

Mathematical formulation

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

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:

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

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

\[ I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t} \]

Flow vector

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

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

What is the correspondence of P & P’

Contour plots of image intensity in two images
Normal Flow

Illusion Works Barber Pole Illusion

Two ways to get flow

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

Horn & Schunck algorithm

Additional smoothness constraint:

\[ e_c = \int \left( (u_i^2 + u_j^2) + (v_i^2 + v_j^2) \right) dx dy, \]

besides optical flow constraint equation term

\[ e_c = \int (I_{xx}u + I_{xy}v + I_{yx}v + I_{yy}v)^2 dx dy, \]

Find u(x,y) that minimizes \( e_c + \lambda e_c \)

Solved as calculus of variations problem – See B.K.P. Horn’s book Robot Vision

Lukas Kanade: Integrate over a window

Assume a single velocity \((u,v)\) for all pixels within an image patch, we can write a cost function indicating for a given \((u,v)\), the extent that the optical flow equation is violated over a window \( \Omega \)

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

\[ \frac{dE(u,v)}{du} = \sum 2I_x I_x + I_y I_y = 0 \]

Condition for a local minimum

\[ \frac{dE(u,v)}{dv} = \sum 2I_x I_y = 0 \]

Solve with:

\[ \frac{\sum I_x^2}{\sum I_x} \frac{\sum I_y}{\sum I_y} \left[ \begin{array}{c} u \\ v \end{array} \right] = -\frac{\sum I_x}{\sum I_x} \left[ \sum I_x \right] \]

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

\[ \sum \nabla I \nabla I^T \mathbf{U} = -\nabla \mathbf{H} \]

Lukas-Kanade: Singularities and the Aperture Problem

Let \( M = \sum \nabla I \nabla I^T \) and \( \mathbf{b} = -\sum \nabla I \nabla I^T \)

- Algorithm: At each pixel compute \( \mathbf{U} \) by solving \( M \mathbf{U} = \mathbf{b} \)

- \( M \) 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
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

Some variants

- Coarse to fine (image pyramids)
- Local/global motion models
- Robust estimation

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

Coarse-to-Fine Estimation

$\nabla I_x \cdot u + \nabla I_y \cdot v + I_z = 0 \implies \text{small } u \text{ and } v ...$

Pyramid / “Coarse-to-fine”

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.

\[ \rho(r, \sigma) = \frac{r^3}{\sigma^2 + r^2} \quad \psi(r, \sigma) = \frac{2r\sigma^3}{(\sigma^2 + r^2)^2} \]