Optical flow and keypoint tracking

Many slides adapted from S. Seitz, R. Szeliski, M. Pollefeys
Motion is a powerful perceptual cue

- Sometimes, it is the only cue
Motion is a powerful perceptual cue

• Even “impoverished” motion data can evoke a strong percept

Motion is a powerful perceptual cue

- Even “impoverished” motion data can evoke a strong percept

Uses of motion in computer vision

- 3D shape reconstruction
- Object segmentation
- Learning and tracking of dynamical models
- Event and activity recognition
- Self-supervised and predictive learning
Preview: Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates.
Keypoint tracking

Motion field

- The motion field is the projection of the 3D scene motion into the image
Optical flow

• **Definition**: optical flow is the *apparent* motion of brightness patterns in the image

• Ideally, optical flow would be the same as the motion field

• Have to be careful: apparent motion can be caused by lighting changes without any actual motion
  
  • Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination
Estimating optical flow

Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them.

Key assumptions

- **Brightness constancy**: projection of the same point looks the same in every frame
- **Small motion**: points do not move very far
- **Spatial coherence**: points move like their neighbors
The brightness constancy constraint

\[ (x, y) \text{ displacement} = (u, v) \]

\[ I(x, y, t - 1) \]

Brightness Constancy Equation:

\[ I(x, y, t - 1) = I(x + u(x, y), y + v(x, y), t) \]

Linearizing the right side using Taylor expansion:

\[ I(x, y, t - 1) \approx I(x, y, t) + I_x u(x, y) + I_y v(x, y) \]

Hence,

\[ I_x u + I_y v + I_t \approx 0 \]
The brightness constancy constraint

\[ I_x u + I_y v + I_t = 0 \]

- How many equations and unknowns per pixel?
  - One equation, two unknowns

- What does this constraint mean?
  \[ \nabla I \cdot (u, v) + I_t = 0 \]

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown!
The brightness constancy constraint

\[ I_x u + I_y v + I_t = 0 \]

• How many equations and unknowns per pixel?
  • One equation, two unknowns

• What does this constraint mean?
  \[ \nabla I \cdot (u, v) + I_t = 0 \]

• The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown!

If \((u, v)\) satisfies the equation, so does \((u+u', v+v')\) if \(\nabla I \cdot (u', v') = 0\)
The aperture problem

Perceived motion
The aperture problem

Actual motion
The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion
The barber pole illusion

http://en.wikipedia.org/wiki/Barberpole_illusion
Solving the aperture problem

• How to get more equations for a pixel?
• **Spatial coherence constraint:** pretend the pixel’s neighbors have the same \((u, v)\)
  • E.g., if we use a 5x5 window, that gives us 25 equations per pixel

\[ \nabla I(x_i) \cdot [u, v] + I_t(x_i) = 0 \]

\[
\begin{bmatrix}
I_x(x_1) & I_y(x_1) \\
I_x(x_2) & I_y(x_2) \\
\vdots & \vdots \\
I_x(x_n) & I_y(x_n)
\end{bmatrix}
\begin{bmatrix}
u \\ v
\end{bmatrix}
= -
\begin{bmatrix}
I_t(x_1) \\
I_t(x_2) \\
\vdots \\
I_t(x_n)
\end{bmatrix}
\]

Solving the aperture problem

• Least squares problem:

\[
\begin{bmatrix}
I_x(x_1) & I_y(x_1) \\
I_x(x_2) & I_y(x_2) \\
\vdots & \vdots \\
I_x(x_n) & I_y(x_n)
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
\vdots \\
v
\end{bmatrix}
=
\begin{bmatrix}
I_t(x_1) \\
I_t(x_2) \\
\vdots \\
I_t(x_n)
\end{bmatrix}
\]

• When is this system solvable?
  • What if the window contains just a single straight edge?

Conditions for solvability

- “Bad” case: single straight edge
Conditions for solvability

- “Good” case
**Lucas-Kanade flow**

Linear least squares problem

$$
\begin{bmatrix}
I_x(x_1) & I_y(x_1) \\
I_x(x_2) & I_y(x_2) \\
\vdots & \vdots \\
I_x(x_n) & I_y(x_n)
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= 
\begin{bmatrix}
I_t(x_1) \\
I_t(x_2) \\
\vdots \\
I_t(x_n)
\end{bmatrix}
$$

Solution given by

$$
(A^T A)d = A^T b
$$

$$
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= 
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
$$

The summations are over all pixels in the window

Lucas-Kanade flow

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\v
\end{bmatrix} = -\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}
\]

- Recall the Harris corner detector: \( M = A^T A \) is the second moment matrix.
- We can figure out whether the system is solvable by looking at the eigenvalues of the second moment matrix.
Recall: second moment matrix

Classification of windows using eigenvalues of the second moment matrix:

- "Corner": $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$
- "Edge": $\lambda_1 \gg \lambda_2$
- "Edge": $\lambda_2 \gg \lambda_1$
- "Flat" region: $\lambda_1$ and $\lambda_2$ are small
“Flower garden” example

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
“Cube” example

Input frames

Output

Source: MATLAB Central File Exchange
“Flower garden” example

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
“Flower garden” example

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Errors in Lucas-Kanade

• The motion is large (larger than a pixel)
• A point does not move like its neighbors
• Brightness constancy does not hold
Fixing the errors in Lucas-Kanade

- The motion is large (larger than a pixel)
  - Multi-resolution estimation
  - Iterative refinement
Multi-resolution estimation

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Multi-resolution estimation

Lucas-Kanade with Pyramids

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Fixing the errors in Lucas-Kanade

- The motion is large (larger than a pixel)
  - Multi-resolution estimation
  - Iterative refinement
  - Exhaustive neighborhood search (feature matching)
- A point does not move like its neighbors
  - Motion segmentation

Figure 11: (a) The optic flow from multi-scale gradient method. (b) Segmentation obtained by clustering optic flow into affine motion regions. (c) Segmentation from consistency checking by image warping. Representing moving images with layers.

Fixing the errors in Lucas-Kanade

- The motion is large (larger than a pixel)
  - Multi-resolution estimation
  - Iterative refinement
  - Exhaustive neighborhood search (feature matching)
- A point does not move like its neighbors
  - Motion segmentation
- Brightness constancy does not hold
  - Exhaustive neighborhood search with normalized correlation
Feature tracking

• If we have more than two images, we can track a feature from one frame to the next by following the optical flow

• Challenges
  • Finding good features to track
  • Adding and deleting tracks
Shi-Tomasi feature tracker

• Find good features using eigenvalues of second-moment matrix
  • Key idea: “good” features to track are the ones whose motion can be estimated reliably
• From frame to frame, track with Lucas-Kanade
  • This amounts to assuming a translation model for frame-to-frame feature movement
• Check consistency of tracks by affine registration to the first observed instance of the feature
  • Affine model is more accurate for larger displacements
  • Comparing to the first frame helps to minimize drift

Tracking example

Figure 1: Three frame details from Woody Allen’s *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

Figure 2: The traffic sign windows from frames 1, 6, 11, 16, 21 as tracked (top), and warped by the computed deformation matrices (bottom).