Optical Flow
EECS 442 – David Fouhey
Fall 2019, University of Michigan
http://web.eecs.umich.edu/~fouhey/teaching/EECS442_F19/
Optical Flow

Idea first introduced by psychologist JJ Gibson in ~1940s to describe how to perceive opportunities for motion
Video

Video: sequence of frames over time
Image is function of space \((x,y)\) and time \(t\) (and channel \(c\))

\[ I(x,y,c,t) \]

- \(x,y\) – location
- \(c\) – channel
- \(t\) – time
Motion Perception

Not grouped

Proximity

Similarity

Similarity

Common Fate

Common Region

Gestalt psychology
Max Wertheimer
1880-1943
Motion and perceptual organization

Sometimes motion is the only cue

Slide Credit: S. Lazebnik, but idea of random dot stereogram is due to B. Julesz
Motion and perceptual organization

Sometimes motion is the only cue

Slide Credit: S. Lazebnik, but idea of random dot stereogram is due to B. Julesz
Motion and perceptual organization

Even impoverished motion data can create a strong percept

Slide Credit: S. Lazebnik
Motion and perceptual organization

Even impoverished motion data can create a strong percept
Motion and perceptual organization

Even impoverished motion data can create a strong percept

Fritz Heider & Marianne Simmel. 1944
Animation from:

An experimental study of apparent behavior.
American Journal of Psychology, 57, 343-359.

Courtesy of:
Department of Psychology,
University of Kansas, Lawrence.
Problem Definition: Optical Flow

Want to estimate pixel motion from image $I(x,y,t)$ to image $I(x,y,t+1)$
Optical flow

Optical flow is the *apparent* motion of objects

Will start by estimating motion of each pixel separately
Then will consider motion of entire image
Optical Flow

Solve correspondence problem: given pixel at time $t$, find **nearby** pixels of the **same color** at time $t+1$

Key assumptions:
- **Color/brightness constancy**: point at time $t$ looks same at time $t+1$
- **Small motion**: points do not move very far
Optical Flow

displacement = (u,v)

Brightness constancy:

I(x, y, t) = I(x + u, y + v, t + 1)

Wrong way to do things: brute force match
Optical Flow

Brightness constancy: \[ I(x, y, t) = I(x + u, y + v, t + 1) \]

Recall Taylor Expansion: \[ I(x + u, y + v, t) = I(x, y, t) + I_x u + I_y v + \ldots \]
Optical Flow Equation

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

\[ 0 \approx I(x + u, y + v, t + 1) - I(x, y, t) = I(x, y, t + 1) + I_x u + I_y v - I(x, y, t) \]

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

If you had to guess, what would you call this?

Adapted from S. Lazebnik slides
Optical Flow Equation

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

\[ 0 \approx I(x + u, y + v, t + 1) - I(x, y, t) \]

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

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

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

\[ = I_t + \nabla I \cdot [u, v] \]

When is this approximation exact?

\[ [u, v] = [0, 0] \]

When is it bad?

\[ u \text{ or } v \text{ big.} \]
Optical Flow Equation

Brightness constancy equation

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

What do static image gradients have to do with motion estimation?
Brightness Constancy Example

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

What's \( u \)?
Optical Flow Equation

Have: \[ I_x u + I_y v + I_t = 0 \quad I_t + \nabla I \cdot [u, v] = 0 \]

How many equations and unknowns per pixel? 1 (single equation), 2 (u and v)

One nasty problem:

Suppose \( \nabla I^T [u', v'] = 0 \)

\[ I_t + \nabla I^T [u + u', v + v'] = 0 \]

Can only identify the motion along gradient and \textbf{not} motion perpendicular to it

Adapted from S. Lazebnik slides
Aperture problem
Aperture problem

Slide credit: S. Lazebnik
Aperture problem
Other Invisible Flow
Other Invisible Flow
Solving Ambiguity – Lucas Kanade

2 unknowns \([u, v]\), 1 eqn per pixel
How do we get more equations?
Assume *spatial coherence*: pixel’s neighbors have *move together* / have same \([u, v]\)
5x5 window gives 25 new equations

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

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
\vdots & \vdots \\
I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
I_t(p_1) \\
\vdots \\
I_t(p_{25})
\end{bmatrix}
\]

Solving for \([u,v]\)

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
\vdots & \vdots \\
I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix} u \\ v \end{bmatrix} = -
\begin{bmatrix}
I_t(p_1) \\
\vdots \\
I_t(p_{25})
\end{bmatrix}
\]

What's the solution?

\[
(A^T A) d = A^T b \
\rightarrow \quad d = (A^T A)^{-1} A^T b
\]

Intuitively, need to solve (sum over pixels in window)

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

\[
A^T A \quad A^T b
\]

Adapted from S. Lazebnik slides
Solving for \([u,v]\)

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

\(A^T A\) \hspace{1cm} \(A^T b\)

What does this remind you of? Harris corner detection!

When can we find \([u,v]\)?

- \(A^T A\) invertible: precisely equal brightness isn’t
- \(A^T A\) not too small: noise + equal brightness
- \(A^T A\) well-conditioned: \(|\lambda_1|/ |\lambda_2|\) not large (edge)

Adapted from S. Lazebnik slides
Low texture region

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix} = \sum \nabla I (\nabla I)^T
\]

- gradients have small magnitude
- small \(\lambda_1\), small \(\lambda_2\)

Slide credit: S. Lazebnik
$$\begin{bmatrix}
\Sigma I_x I_x & \Sigma I_x I_y \\
\Sigma I_x I_y & \Sigma I_y I_y
\end{bmatrix} = \Sigma \nabla I (\nabla I)^T$$

- large gradients, all the same
- large $\lambda_1$, small $\lambda_2$
High texture region

\[
\begin{bmatrix}
\Sigma I_x I_x & \Sigma I_x I_y \\
\Sigma I_x I_y & \Sigma I_y I_y
\end{bmatrix} = \Sigma \nabla I (\nabla I)^T
\]

- gradients are different, large magnitudes
- large \( \lambda_1 \), large \( \lambda_2 \)

Slide credit: S. Lazebnik
Lucas-Kanade flow example

Input frames

Output

Source: MATLAB Central File Exchange
Aperture problem Take 2
Aperture problem Take 2
For Comparison

Slide credit: S. Lazebnik
For Comparison
So How Does This Fail?

• Point doesn’t move like neighbors:
  • Why would this happen?
  • Figure out which points move together, then come back and fix.
So How Does This Fail?

- Point doesn’t move like neighbors:
  - Why would this happen?
  - Figure out which points move together, then come back and fix

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.

So How Does This Fail?

- Point doesn’t move like neighbors:
  - Why would this happen?
  - Figure out which points move together, then come back and fix.

- Brightness constancy isn’t true
  - Why would this happen?
  - Solution: other form of matching (e.g. SIFT)

- Taylor series is bad approximation
  - Why would this happen?
  - Solution: Make your pixels big
Revisiting small motions

• Is this motion small enough?
  • Probably not—it’s much larger than one pixel
  • How might we solve this problem?

Slide credit: S. Lazebnik
Reduce the resolution!

Slide credit: S. Lazebnik
Coarse-to-fine optical flow estimation

Typically called Gaussian Pyramid

Slide credit: S. Lazebnik
Coarse-to-fine optical flow estimation

Do we start at bottom or top to align?

Slide credit: S. Lazebnik
Coarse-to-fine optical flow estimation

image 1

Flow
Warp, Upsample
Flow
...

image 2

Slide credit: S. Lazebnik
Optical Flow Results

Lucas-Kanade without pyramids

Fails in areas of large motion

Slide credit: K. Hassan-Shafique
Optical Flow Results

Lucas-Kanade with Pyramids

Slide credit: K. Hassan-Shafique
Applying This

• Would like tracks of where things move (e.g., for reconstruction)

Applying This

• Which features should we track?
  • Use eigenvalues of $A^T A$ to find corners
• Use flow to figure out $[u,v]$ for each “track”
• Register points to first frame by affine warp

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).

State-of-the-art optical flow, 2009

Start with something similar to Lucas-Kanade
+ gradient constancy
+ energy minimization with smoothing term
+ region matching

Region-based + Pixel-based + Keypoint-based

Large displacement optical flow, Brox et al., CVPR 2009
State-of-the-art optical flow

- Input: 6 channel input (RGB @ t, RGB @ t+1)
- Output: 2 channel input (u,v)
- Current best methods are learned

Training Data

Flying Chairs Dataset

Deep Optical Flow

Results on Sintel (standard benchmark)

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

Slide credit: S. Lazebnik
Motion Magnification

Idea: take flow, magnify it
Motion Magnification

Example credit: C. Liu
Motion Magnification

Example credit: C. Liu