Lucas-Kanade Motion Estimation

Thanks to Steve Seitz, Simon Baker, Takeo Kanade, and anyone else who helped develop these slides.
Why estimate motion?

We live in a 4-D world

Wide applications
- Object Tracking
- Camera Stabilization
- Image Mosaics
- 3D Shape Reconstruction (SFM)
- Special Effects (Match Move)
Optical flow
Problem definition: optical flow

How to estimate pixel motion from image $H$ to image $I$?

- Solve pixel correspondence problem
  - given a pixel in $H$, look for nearby pixels of the same color in $I$

Key assumptions

- **color constancy**: a point in $H$ looks the same in $I$
  - For grayscale images, this is **brightness constancy**
- **small motion**: points do not move very far

This is called the **optical flow** problem
Optical flow constraints (grayscale images)

Let’s look at these constraints more closely

- brightness constancy: Q: what’s the equation?
  
  \[ H(x, y) = I(x+u, y+v) \]

- small motion: (u and v are less than 1 pixel)
  
  - suppose we take the Taylor series expansion of I:
    
    \[ I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \text{higher order terms} \]
    
    \[ \approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v \]
Optical flow equation

Combining these two equations

\[ 0 = I(x + u, y + v) - H(x, y) \]
\[ \approx I(x, y) + I_x u + I_y v - H(x, y) \]
\[ \approx (I(x, y) - H(x, y)) + I_x u + I_y v \]
\[ \approx I_t + I_x u + I_y v \]
\[ \approx I_t + \nabla I \cdot [u \ v] \]

What is \( I_t \) ? The time derivative of the image at \((x,y)\)

How do we calculate it?
Optical flow equation

\[ 0 = I_t + \nabla I \cdot [u \ v] \]

Q: how many unknowns and equations per pixel?

1 equation, but 2 unknowns (u and v)

Intuitively, what does this constraint mean?

• The component of the flow in the gradient direction is determined
• The component of the flow parallel to an edge is unknown

This explains the Barber Pole illusion
http://www.sandlotscience.com/Ambiguous/barberpole.htm
Aperture problem
Aperture problem
Solving the aperture problem

Basic idea: assume motion field is smooth

Lukas & Kanade: assume locally constant motion
  • pretend the pixel’s neighbors have the same (u,v)
    – If we use a 5x5 window, that gives us 25 equations per pixel!

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

Many other methods exist. Here’s an overview:
Lukas-Kanade flow

How to get more equations for a pixel?

• Basic idea: impose additional constraints
  – most common is to assume that the flow field is smooth locally
  – one method: pretend the pixel’s neighbors have the same \((u,v)\)
    » If we use a 5x5 window, that gives us 25 equations per pixel!

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

\[
\begin{bmatrix}
  I_x(p_1) & I_y(p_1) \\
  I_x(p_2) & I_y(p_2) \\
  \vdots & \vdots \\
  I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
  u \\
  v
\end{bmatrix}
= -
\begin{bmatrix}
  I_t(p_1) \\
  I_t(p_2) \\
  \vdots \\
  I_t(p_{25})
\end{bmatrix}
\]

\[
A \quad \text{25x2} \\
\begin{bmatrix}
  d \\
  \text{2x1}
\end{bmatrix} \\
\begin{bmatrix}
  b \\
  \text{25x1}
\end{bmatrix}
\]
RGB version

How to get more equations for a pixel?

• Basic idea: impose additional constraints
  – most common is to assume that the flow field is smooth locally
  – one method: pretend the pixel's neighbors have the same \((u,v)\)
    » If we use a 5x5 window, that gives us 25*3 equations per pixel!

\[
0 = I_t(p_i)[0, 1, 2] + \nabla I(p_i)[0, 1, 2] \cdot [u \ v]
\]

\[
\begin{bmatrix}
I_x(p_1)[0] & I_y(p_1)[0] \\
I_x(p_1)[1] & I_y(p_1)[1] \\
I_x(p_1)[2] & I_y(p_1)[2] \\
\vdots & \vdots \\
I_x(p_{25})[0] & I_y(p_{25})[0] \\
I_x(p_{25})[1] & I_y(p_{25})[1] \\
I_x(p_{25})[2] & I_y(p_{25})[2]
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= 
\begin{bmatrix}
I_t(p_1)[0] \\
I_t(p_1)[1] \\
I_t(p_1)[2] \\
\vdots \\
I_t(p_{25})[0] \\
I_t(p_{25})[1] \\
I_t(p_{25})[2]
\end{bmatrix}
\]

\[
A_{75 \times 2} \quad \begin{bmatrix} d \end{bmatrix}_{2 \times 1} \quad b_{75 \times 1}
\]
Lukas-Kanade flow

Prob: we have more equations than unknowns

$A_{25x2} \ d_{2x1} = b_{25x1}$

$\rightarrow$ minimize $\|Ad - b\|^2$

Solution: solve least squares problem

- minimum least squares solution given by solution (in $d$) of:

$\begin{bmatrix} A^T A & d \end{bmatrix} d = A^T b$

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

$A^T A$ $A^T b$

- The summations are over all pixels in the $K \times K$ window
- This technique was first proposed by Lukas & Kanade (1981)
Conditions for solvability

- Optimal \((u, v)\) satisfies Lucas-Kanade equation

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

\(A^T A\)

\(A^T b\)

When is This Solvable?

- \(A^T A\) should be invertible
- \(A^T A\) should not be too small due to noise
  - eigenvalues \(\lambda_1\) and \(\lambda_2\) of \(A^T A\) should not be too small
- \(A^T A\) should be well-conditioned
  - \(\lambda_1/\lambda_2\) should not be too large (\(\lambda_1 = \) larger eigenvalue)
Edges cause problems

\[ \sum \nabla I (\nabla I)^T \]
- large gradients, all the same
- large $\lambda_1$, small $\lambda_2$
Low texture regions don’t work

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

- gradients have small magnitude
- small \( \lambda_1 \), small \( \lambda_2 \)
High textured region work best

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

- gradients are different, large magnitudes
- large \( \lambda_1 \), large \( \lambda_2 \)
Errors in Lukas-Kanade

What are the potential causes of errors in this procedure?

• Suppose $A^TA$ is easily invertible
• Suppose there is not much noise in the image

When our assumptions are violated

• Brightness constancy is **not** satisfied
• The motion is **not** small
• A point does **not** move like its neighbors
  – window size is too large
  – what is the ideal window size?
Revisiting the small motion assumption

Is this motion small enough?
- Probably not—it’s much larger than one pixel (2\textsuperscript{nd} order terms dominate)
- How might we solve this problem?
Reduce the resolution!
Coarse-to-fine optical flow estimation

Gaussian pyramid of image H

Gaussian pyramid of image I

\( u = 10 \text{ pixels} \)

\( u = 5 \text{ pixels} \)

\( u = 2.5 \text{ pixels} \)

\( u = 1.25 \text{ pixels} \)
Coarse-to-fine optical flow estimation

---

Gaussian pyramid of image H

Run iterative L-K

Warp & upsample

Run iterative L-K

---

Gaussian pyramid of image I

---
Optical flow result
The Flower Garden Video

What should the optical flow be?
Results from Ming Ye’s Algorithm (2003 EE)
TAXI: Hamburg Taxi

256x190, (Barron 94)
max speed 3.0 pix/frame

Ours

Error map

Smoothness error
Traffic

512x512
(Nagel)
max speed: 6.0 pix/frame

Ours
Error map
Smoothness error
Pepsi Can

201x201 (Black)
Max speed: 2pix/frame

Ours

BA

Smoothness error
FG: Flower Garden

360x240 (Black)
Max speed: 7pix/frame

BA

LMS

Ours

Error map

Smoothness error