Announcements

- more panorama slots available now
  - you can sign up for a 2nd time if you'd like

Motion Estimation

http://www.sandlotscience.com/Ambiguous/Barberpole_Illusion.htm

http://www.sandlotscience.com/Distortions/Breathing_Square.htm

Today's Readings

- Szeliski Chapters 7.1, 7.2, 7.4

Why estimate motion?

Lots of uses

- Track object behavior
- Correct for camera jitter (stabilization)
- Align images (mosaics)
- 3D shape reconstruction
- Special effects

Motion estimation

Input: sequence of images

Output: point correspondence

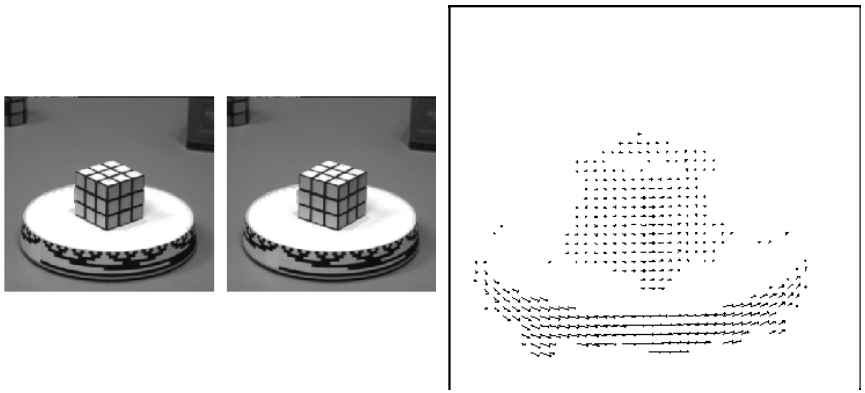
Feature correspondence: "Feature Tracking"

- we've seen this already (e.g., SIFT)
- can modify this to be more accurate/efficient if the images are in sequence (e.g., video)

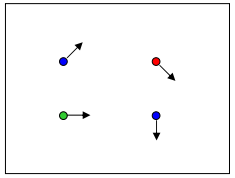
Pixel (dense) correspondence: "Optical Flow"

- today's lecture

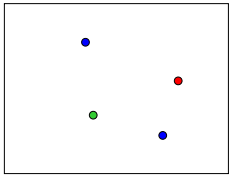
Optical flow



Problem definition: optical flow



$H(x, y)$



$I(x, y)$

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

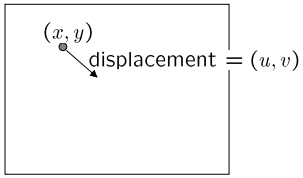
- Solve pixel correspondence problem
  - given a pixel in H, look for nearby pixels of the same color in I

Key assumptions

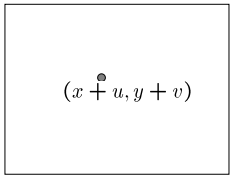
- **color constancy:** a point in H looks the same in I
  - For grayscale images, this is **brightness constancy**
- **small motion**: points do not move very far

This is called the **optical flow** problem

Optical flow constraints (grayscale images)



$H(x, y)$



$I(x, y)$

Let's look at these constraints more closely

- brightness constancy: Q: what's the equation?

- 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

shorthand: $I_x = \frac{\partial I}{\partial x}$

Optical flow equation

Combining these two equations

shorthand: $I_x = \frac{\partial I}{\partial x}$

$$
\begin{aligned}
0 &= I(x+u, y+v) - H(x,y) \\
&\approx I(x,y) + I_x u + I_y v - H(x,y) \\
&\approx (I(x,y) - H(x,y)) + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot [u \ \ v]
\end{aligned}
$$

In the limit as u and v go to zero, this becomes exact

$$0 = I_t + \nabla I \cdot [\tfrac{\partial x}{\partial t} \ \ \tfrac{\partial y}{\partial t}]$$

Optical flow equation

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

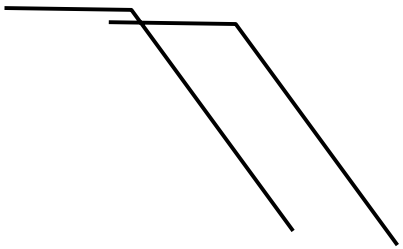
Q: how many unknowns and equations per pixel?

Intuitively, what does this constraint mean?

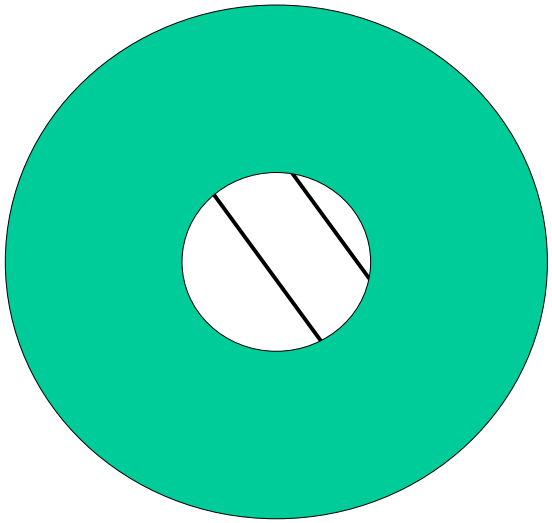
- The component of the flow in the gradient direction is determined
- The component of the flow parallel to an edge is unknown

This explains the Barber Pole illusion
http://www.sandlotscience.com/Ambiguous/Barberpole_Illusion.htm

Aperture problem



Aperture problem



Solving the aperture problem

Basic idea: assume motion field is smooth

Horn & Schunk: add smoothness term

$$\int\int (I_t + \nabla I \cdot [u\ v])^2 + \lambda^2(\|\nabla u\|^2 + \|\nabla v\|^2)\ dx\ dy$$

Lucas & Kanade: assume locally constant motion

- pretend the pixel's neighbors have the same (u,v)

Many other methods exist. Here's an overview:

- S. Baker, M. Black, J. P. Lewis, S. Roth, D. Scharstein, and R. Szeliski. *A database and evaluation methodology for optical flow*. In Proc. ICCV, 2007
- http://vision.middlebury.edu/flow/

Lucas-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!

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u\ v]$$

$$\underset{25\times2}{\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix}} \underset{2\times1}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{25\times1}{\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}}$$

$$\underset{25\times2}{A} \qquad \underset{2\times1}{d} \qquad \underset{25\times1}{b}$$

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!

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

$$\underset{75\times2}{\begin{bmatrix} I_x(\mathbf{p_1})[0] & I_y(\mathbf{p_1})[0] \\ I_x(\mathbf{p_1})[1] & I_y(\mathbf{p_1})[1] \\ I_x(\mathbf{p_1})[2] & I_y(\mathbf{p_1})[2] \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}})[0] & I_y(\mathbf{p_{25}})[0] \\ I_x(\mathbf{p_{25}})[1] & I_y(\mathbf{p_{25}})[1] \\ I_x(\mathbf{p_{25}})[2] & I_y(\mathbf{p_{25}})[2] \end{bmatrix}} \underset{2\times1}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{75\times1}{\begin{bmatrix} I_t(\mathbf{p_1})[0] \\ I_t(\mathbf{p_1})[1] \\ I_t(\mathbf{p_1})[2] \\ \vdots \\ I_t(\mathbf{p_{25}})[0] \\ I_t(\mathbf{p_{25}})[1] \\ I_t(\mathbf{p_{25}})[2] \end{bmatrix}}$$

$$\underset{75\times2}{A} \qquad \underset{2\times1}{d} \qquad \underset{75\times1}{b}$$

Lucas-Kanade flow

Prob: we have more equations than unknowns

$$\underset{25\times2}{A}\ \underset{2\times1}{d} = \underset{25\times1}{b} \quad \longrightarrow \quad \text{minimize } \|Ad - b\|^2$$

Solution: solve least squares problem

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

$$\underset{2\times2}{(A^TA)}\ \underset{2\times1}{d} = \underset{2\times1}{A^Tb}$$

$$\underset{A^TA}{\begin{bmatrix} \sum I_xI_x & \sum I_xI_y \\ \sum I_xI_y & \sum I_yI_y \end{bmatrix}} \begin{bmatrix} u \\ v \end{bmatrix} = - \underset{A^Tb}{\begin{bmatrix} \sum I_xI_t \\ \sum I_yI_t \end{bmatrix}}$$

- The summations are over all pixels in the K x K window
- This technique was first proposed by Lucas & Kanade (1981)

Conditions for solvability

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

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

$A^T A$ $\qquad$ $A^T b$

When is This Solvable?

- **A^TA** should be invertible
- **A^TA** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **A^TA** should not be too small
- **A^TA** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Does this look familiar?

- **A^TA** is the Harris matrix

Observation

This is a two image problem BUT

- Can measure sensitivity by just looking at one of the images!
- This tells us which pixels are easy to track, which are hard
  - very useful for feature tracking...

Errors in Lucas-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?

Improving accuracy

Recall our small motion assumption

$$\begin{aligned} 0 &= I(x+u, y+v) - H(x,y) \\ &\approx I(x,y) + I_x u + I_y v - H(x,y) \end{aligned}$$

This is not exact

- To do better, we need to add higher order terms back in:

$$= I(x,y) + I_x u + I_y v + \text{higher order terms} - H(x,y)$$

This is a polynomial root finding problem

- Can solve using **Newton's method**
  - Also known as **Newton-Raphson** method
  - For more on Newton-Raphson, see (first four pages)
    - » http://www.ulib.org/webRoot/Books/Numerical_Recipes/bookcpdf/c9-4.pdf
- Lucas-Kanade method does one iteration of Newton's method
  - Better results are obtained via more iterations

1D case on board

Iterative Refinement

Iterative Lucas-Kanade Algorithm

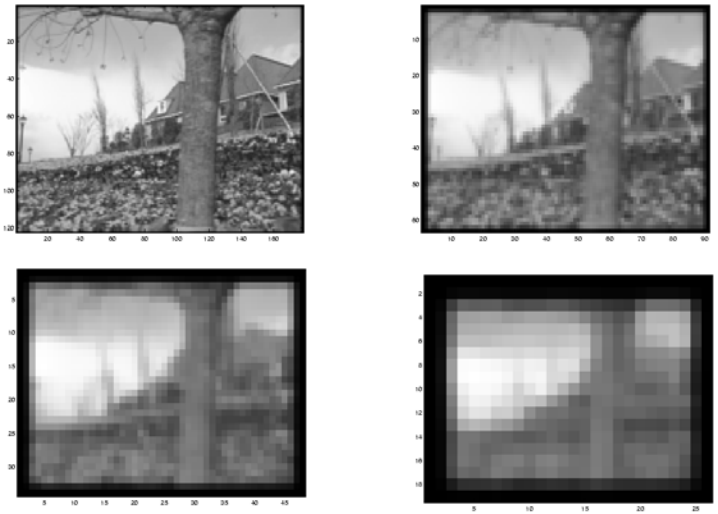
1. Estimate velocity at each pixel by solving Lucas-Kanade equations
2. Warp H towards I using the estimated flow field
   - *use image warping techniques*
3. Repeat until convergence

Revisiting the small motion assumption

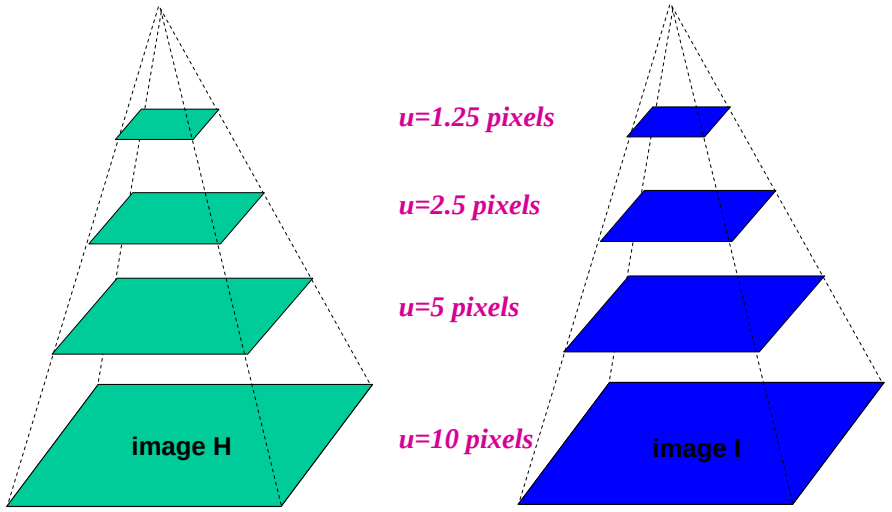
Is this motion small enough?

- Probably not—it's much larger than one pixel (2nd order terms dominate)
- How might we solve this problem?

Reduce the resolution!

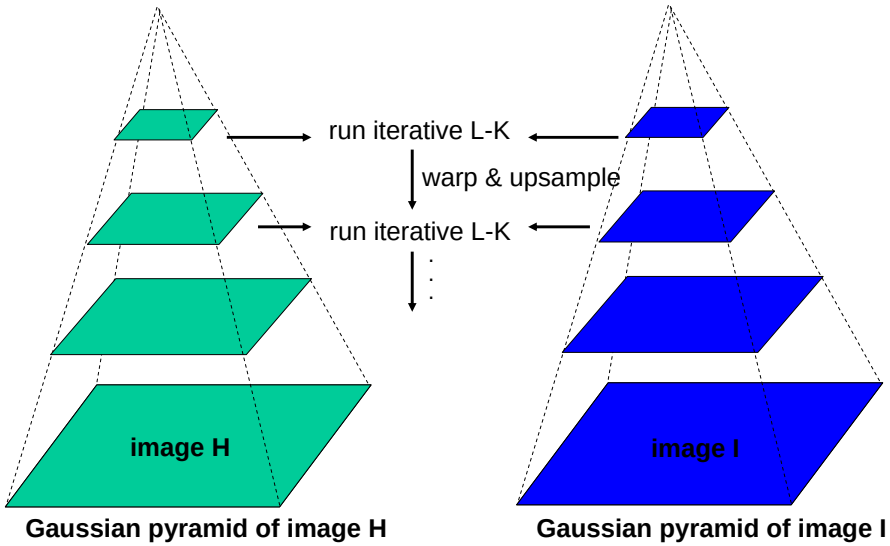


Coarse-to-fine optical flow estimation

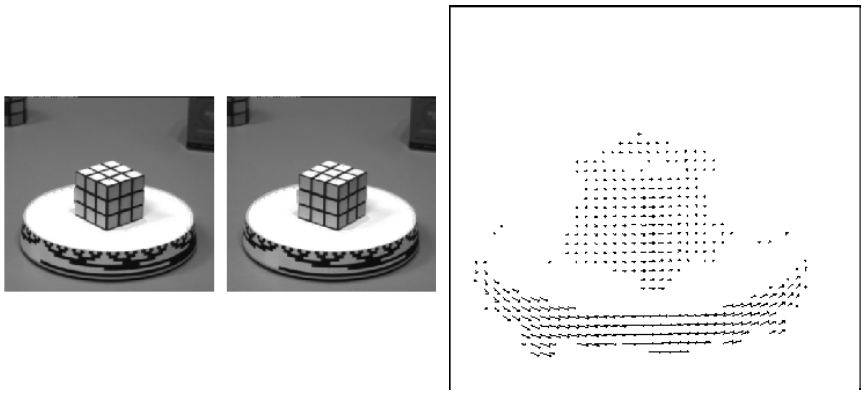


Gaussian pyramid of image H **Gaussian pyramid of image I**

Coarse-to-fine optical flow estimation



Optical flow result

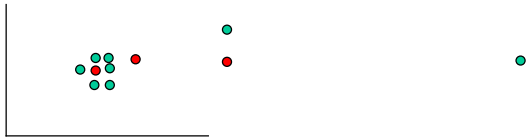


Dewey morph

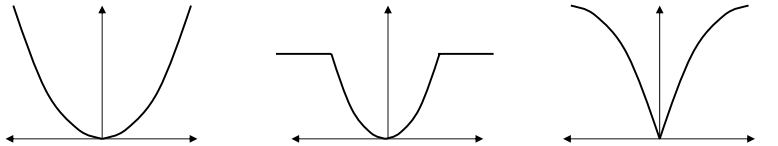
Robust methods

L-K minimizes a sum-of-squares error metric

- least squares techniques overly sensitive to outliers



Error metrics



Robust optical flow

Robust Horn & Schunk

$$\int\int \rho(I_t + \nabla I \cdot [u\ v]) + \lambda^2 \rho(\|\nabla u\|^2 + \|\nabla v\|^2)\ dx\ dy$$

Robust Lucas-Kanade

$$\sum_{(x,y)\in W} \rho(I_t + \nabla I \cdot [u\ v])$$

first image | quadratic flow | lorentzian flow | detected outliers

Reference

- Black, M. J. and Anandan, P., A framework for the robust estimation of optical flow, *Fourth International Conf. on Computer Vision* (ICCV), 1993, pp. 231-236 http://www.cs.washington.edu/education/courses/576/03sp/readings/black93.pdf

Benchmarking optical flow algorithms

Middlebury flow page

- http://vision.middlebury.edu/flow/

Discussion: features vs. flow?

Features are better for:

Flow is better for:

Advanced topics

Particles: combining features and flow

- Peter Sand et al.
- http://rvsn.csail.mit.edu/pv/

State-of-the-art feature tracking/SLAM

- Georg Klein et al.
- http://www.robots.ox.ac.uk/~gk/