Announcements

Photo shoot next Wednesday in class!

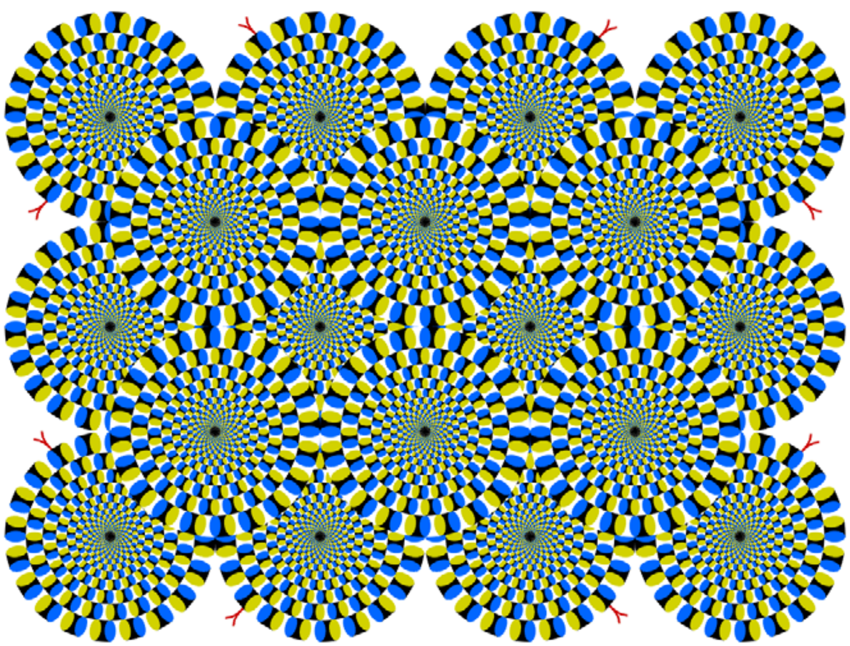
Motion Estimation

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

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

Today's Readings

- Szeliski Chapters 7.1, 7.2, 7.4
- Newton's method Wikpedia page



Copyright A.Kitaoka 2003

Why estimate motion?

Lots of uses

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

Motion estimation

Input: sequence of images

Output: point correspondence

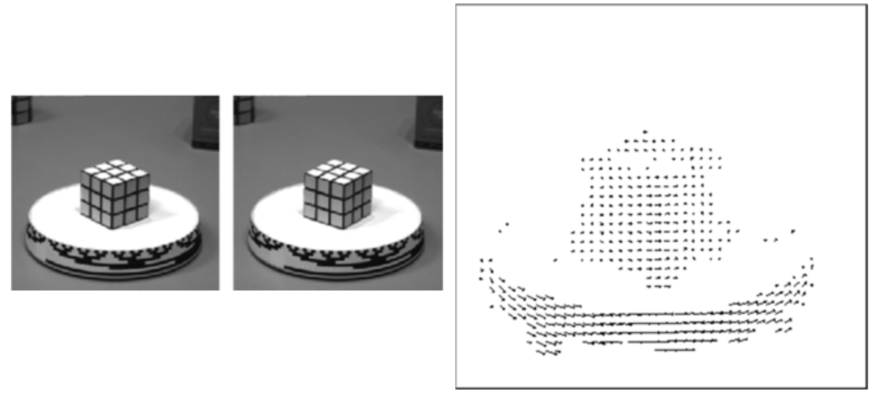
Feature tracking

- we've seen this already (e.g., SIFT)
- can modify this to be more efficient

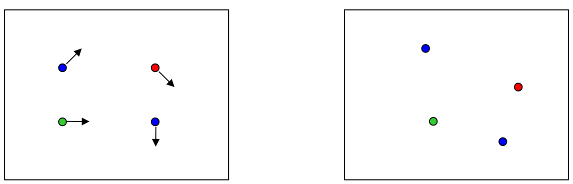
Pixel tracking: "Optical Flow"

- 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?

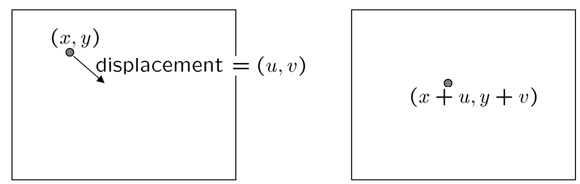
- Solve pixel correspondence problem
  - given a pixel in H, look for nearby pixels of the same color in I

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?

  $H(x,y) - I(x+u, y+v) = 0$

- 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

$$0 = I(x+u, y+v) - H(x,y)$$

$$\approx I(x,y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v - H(x,y)$$

$$= I(x,y) - H(x,y) + \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \cdot (u,v)$$

$$0 \approx \frac{d}{dt} I + \nabla I \cdot (u,v)$$

Optical flow equation

Combining these two equations

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

$$0 = I(x+u, y+v) - H(x,y)$$

$$\approx I(x,y) + I_x u + I_y v - H(x,y)$$

$$\approx (I(x,y) - H(x,y)) + I_x u + I_y v$$

$$\approx I_t + I_x u + I_y v$$

$$\approx I_t + \nabla I \cdot [u\ \ v]$$

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

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

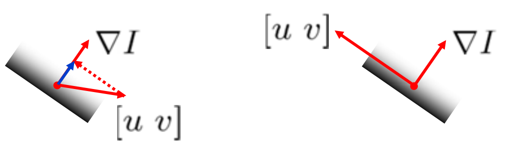
Optical flow equation

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

Q: how many unknowns and equations per pixel?

2 1

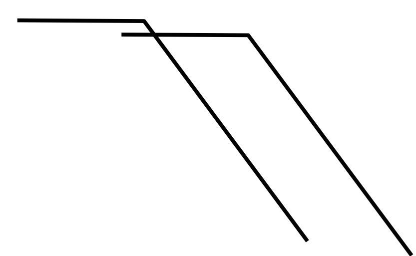
Intuitively, what does this constraint mean?



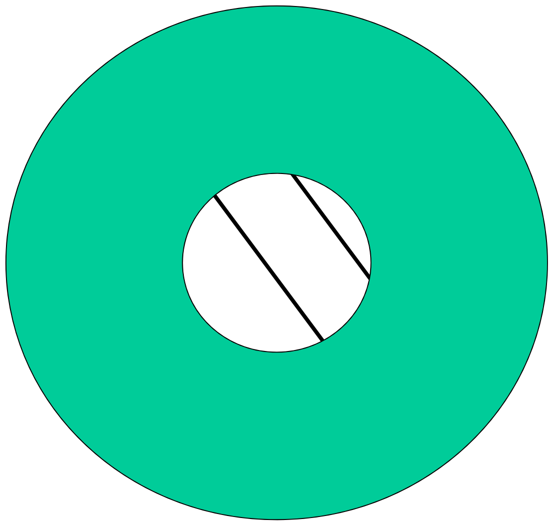
- The component of the flow in the gradient direction is determined
- The component of the flow parallel to an edge is unknown

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/

Solving the aperture problem

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!

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

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

Lucas-Kanade flow

Prob: we have more equations than unknowns

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

Solution: solve least squares problem

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

$$\underset{2\times2}{(A^T A)} \ \underset{2\times1}{d} = \underset{2\times1}{A^T b}$$

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

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

– described in Szesliski text (today's reading)

Conditions for solvability

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

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

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

When is This Solvable?

- $\mathbf{A^T A}$ should be invertible
- $\mathbf{A^T A}$ should not be too small due to noise
  – eigenvalues λ_1 and λ_2 of $\mathbf{A^T A}$ should not be too small
- $\mathbf{A^T A}$ should be well-conditioned
  – λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Does this look familiar?

- $\mathbf{A^T A}$ 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 $A^T A$ 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

$$0 = I(x+u, y+v) - H(x,y)$$

$$\approx I(x,y) + I_x u + I_y v - H(x,y)$$

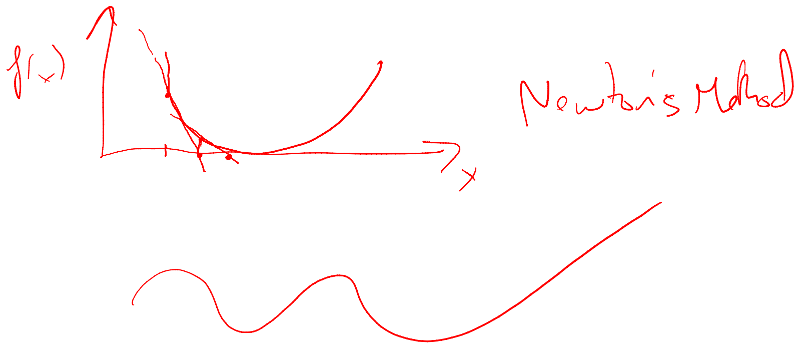
This is not exact

- 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

Root Finding



Improving accuracy

Recall our small motion assumption

$$0 = I(x+u, y+v) - H(x,y)$$

$$\approx I(x,y) + I_x u + I_y v - H(x,y)$$

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
  - – Today's reading (first four pages)
    - » http://www.library.cornell.edu/nr/bookcpdf/c9-4.pdf
- Approach so far does one iteration of Newton's method
  - – Better results are obtained via more iterations

Iterative Refinement

Iterative Lucas-Kanade Algorithm

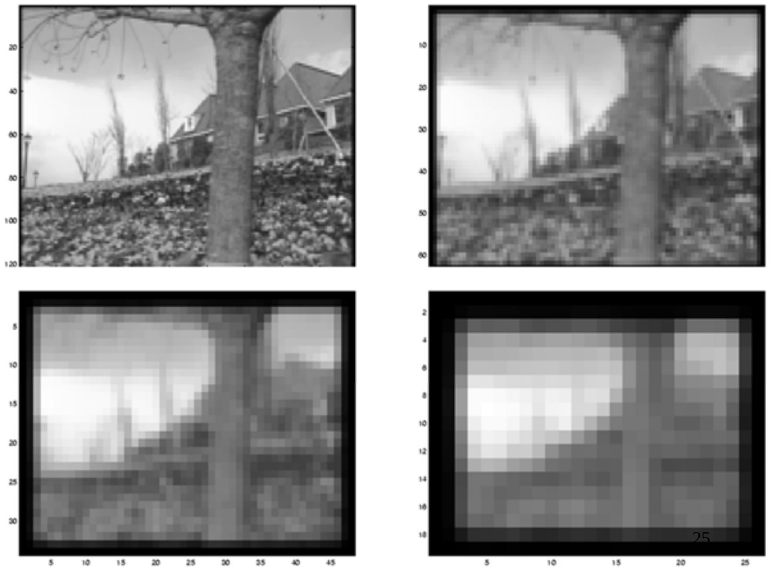
1. Estimate velocity at each pixel by solving Lucas-Kanade equations
2. Warp H towards I using the estimated flow field
   - use image warping techniques
3. Repeat until convergence

Revisiting the small motion assumption

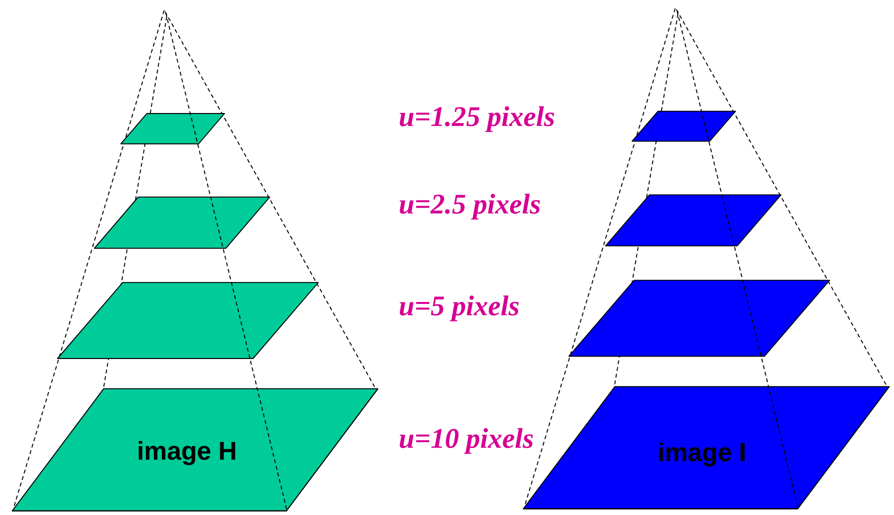
Is this motion small enough?

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

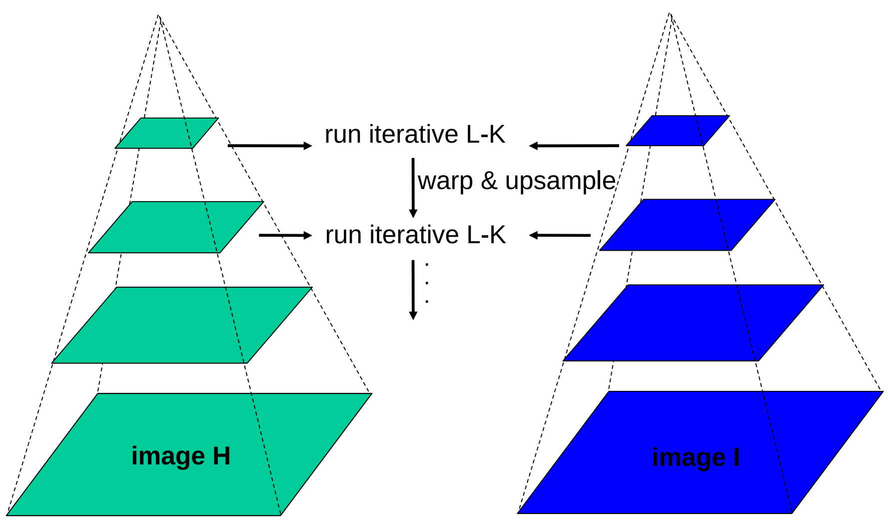
Reduce the resolution!



Coarse-to-fine optical flow estimation



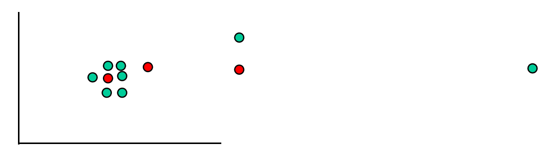
Coarse-to-fine optical flow estimation



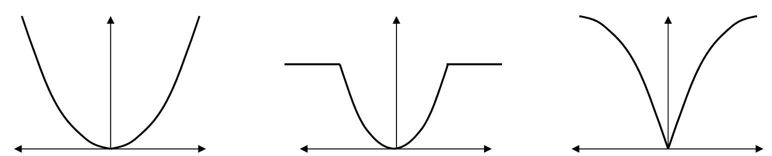
Robust methods

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

- least squares techniques overly sensitive to outliers



Error metrics



quadratic	truncated quadratic	lorentzian
$\rho(x) = x^2$	$\rho_{\alpha,\lambda}(x) = \begin{cases} \lambda x^2 & \text{if } \lvert x\rvert < \frac{\sqrt{\alpha}}{\sqrt{\lambda}} \\ \alpha & \text{otherwise} \end{cases}$	$\rho_\sigma(x) = log\left(1 + \frac{1}{2}(\frac{x}{\sigma})^2\right)$

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

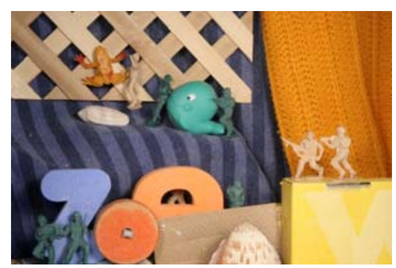
Flow quality evaluation

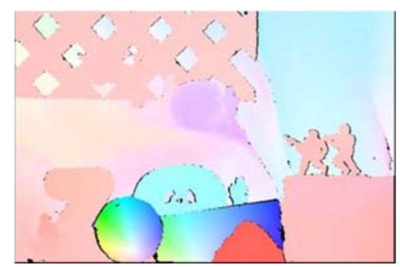
Flow quality evaluation

Flow quality evaluation

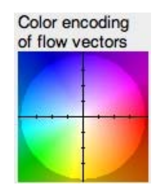
Middlebury flow page

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





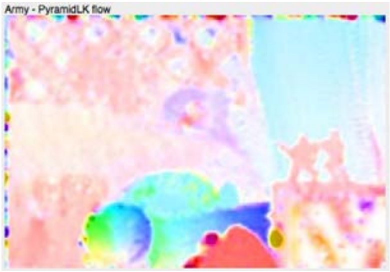
Ground Truth



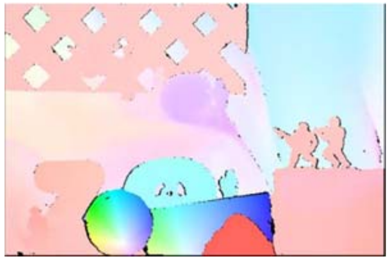
Flow quality evaluation

Middlebury flow page

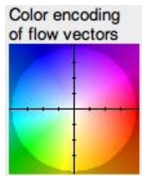
- http://vision.middlebury.edu/flow/



Lucas-Kanade flow



Ground Truth

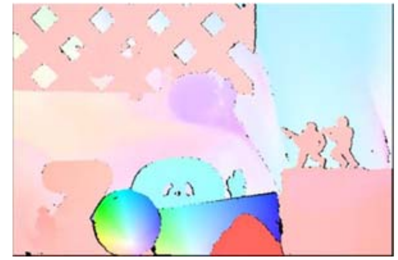


Flow quality evaluation

Middlebury flow page

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

Best-in-class alg (as of 2/26/12)



Ground Truth