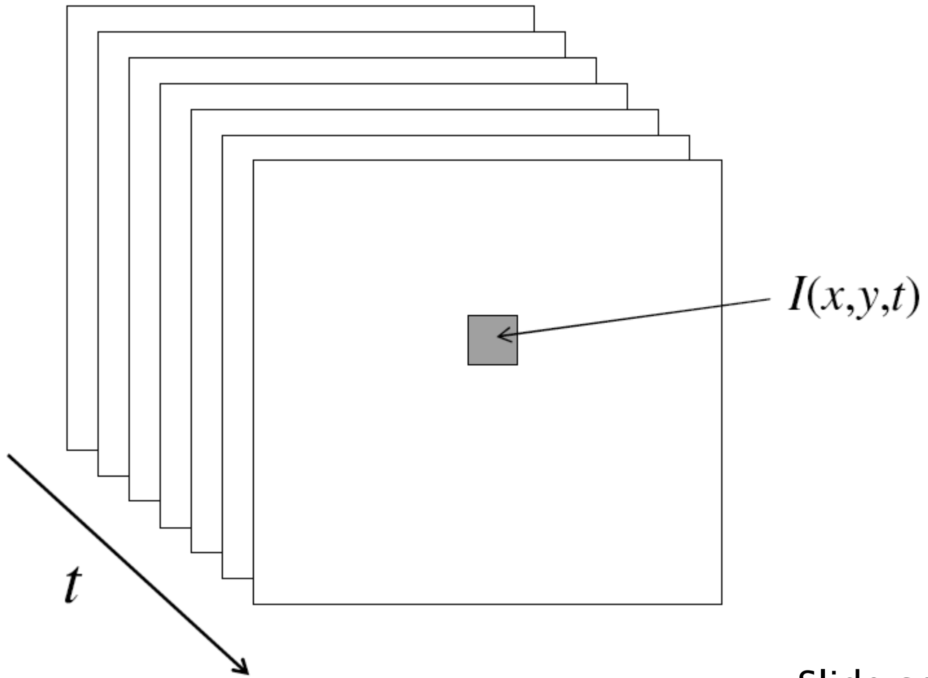
Motion

Optical Flow

Video

- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



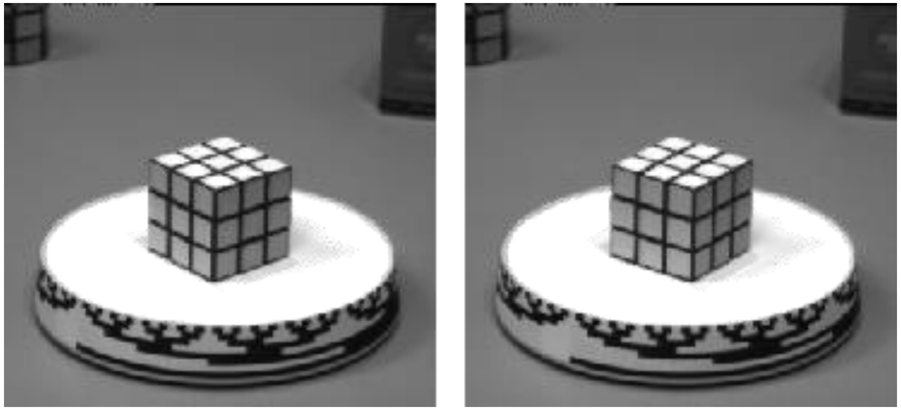
Slide credit: Kristen Grauman

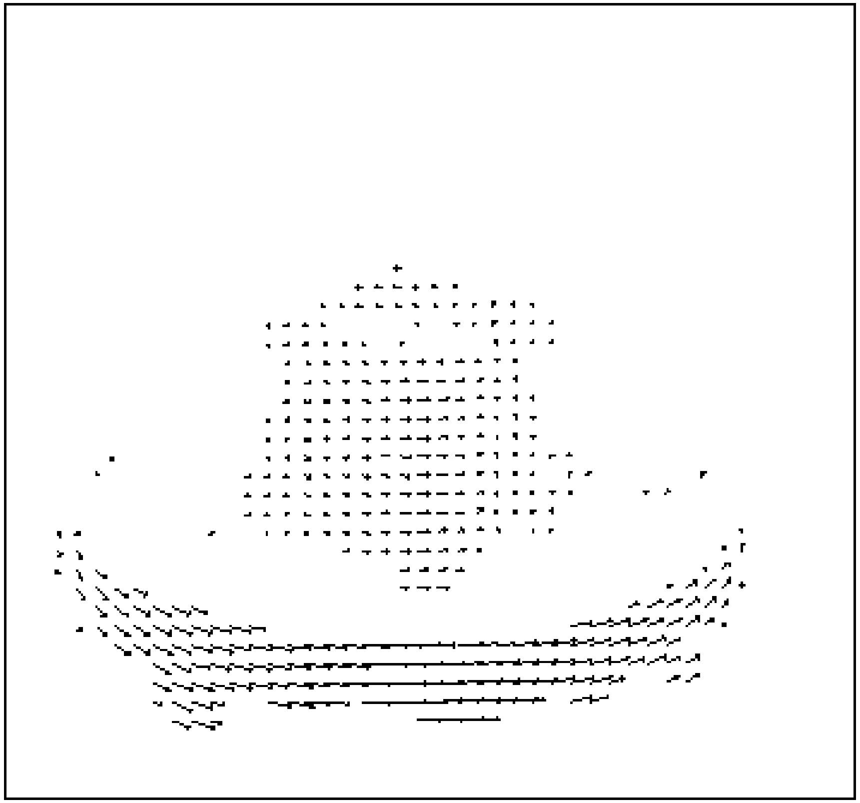
Uses of motion in computer vision

- 3D structure reconstruction
- Object segmentation from motion
- Learning and tracking of dynamical models
- Event and activity recognition
- Improving video quality

Motion Field

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





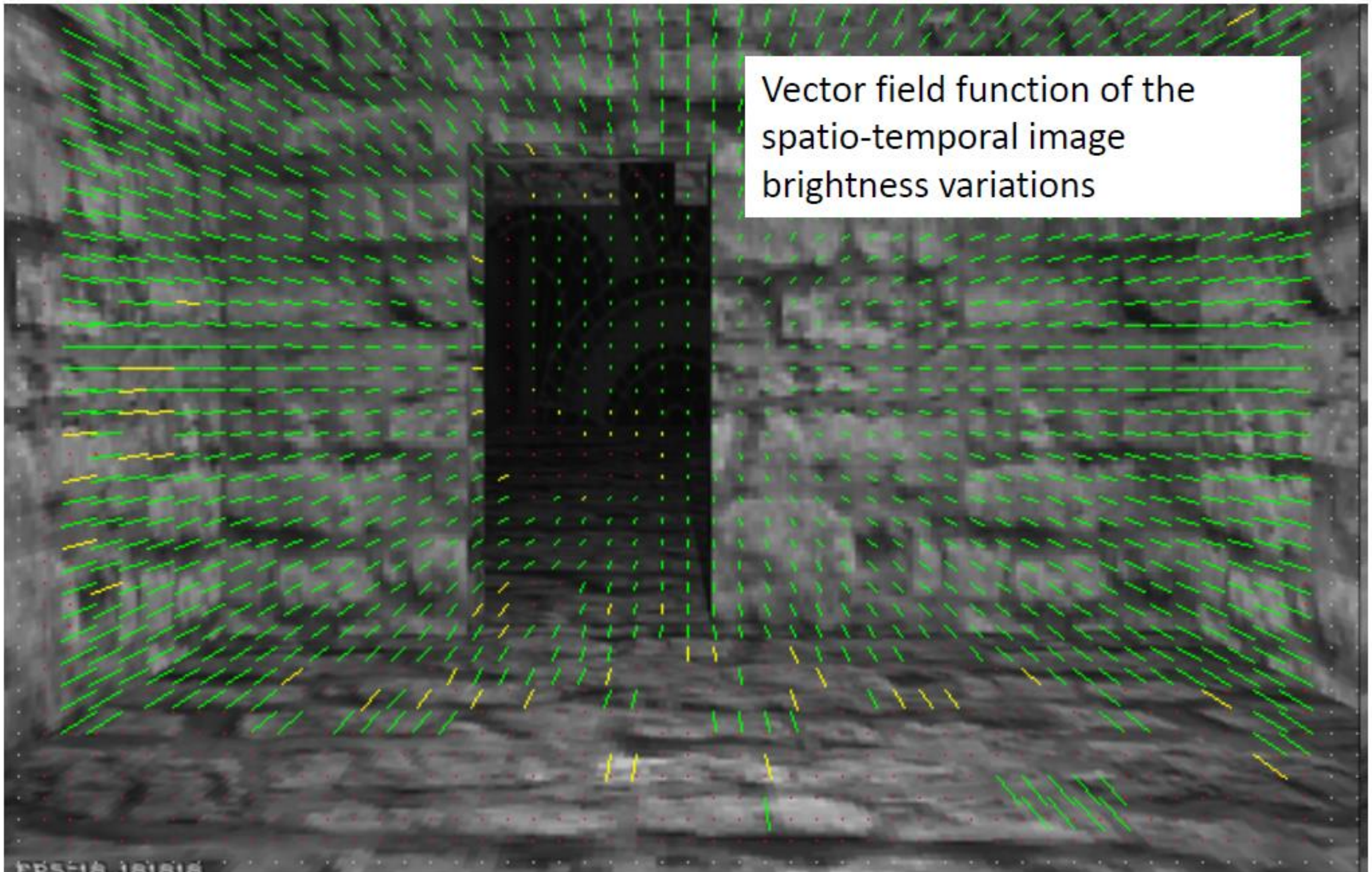
Slide credit: Kristen Grauman

Optical Flow

- **Definition**: optical flow is the *apparent* motion of brightness patterns in the image.
- Note: 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: Kristen Grauman

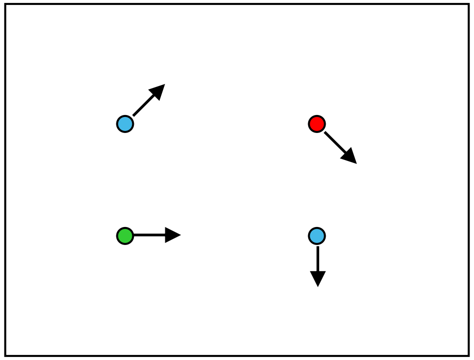
Optical Flow



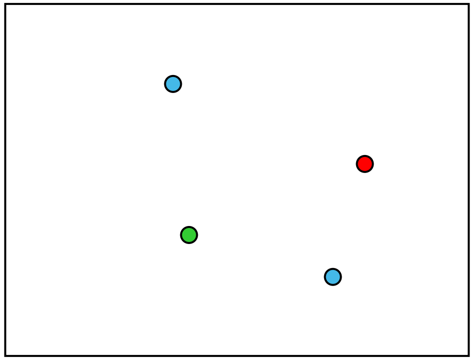
Picture courtesy of Selim Temizer - Learning and Intelligent Systems (LIS) Group, MIT

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



$I(x,y,t-1)$



$I(x,y,t)$

Slide credit: Kristen Grauman

Color/Brightness Constancy

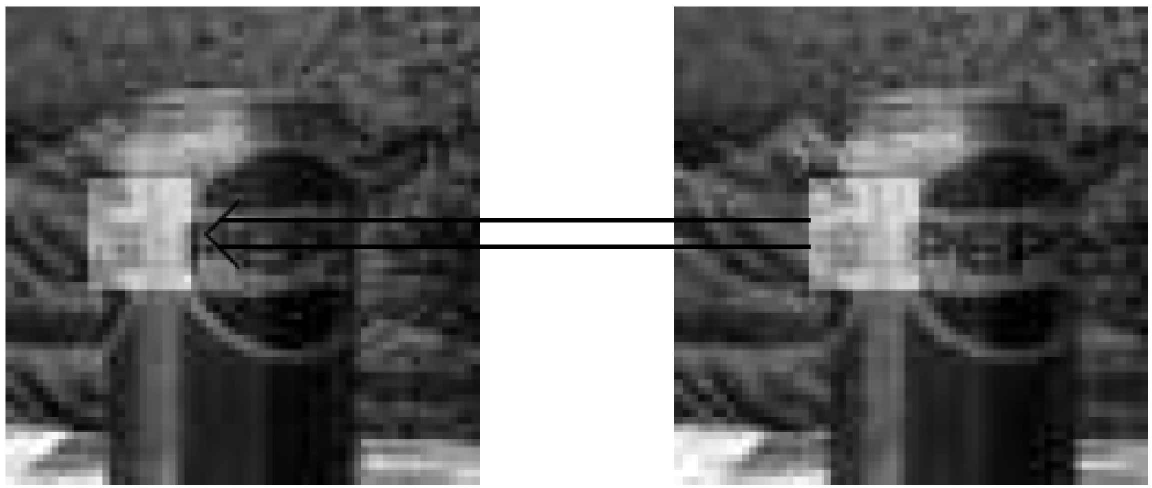


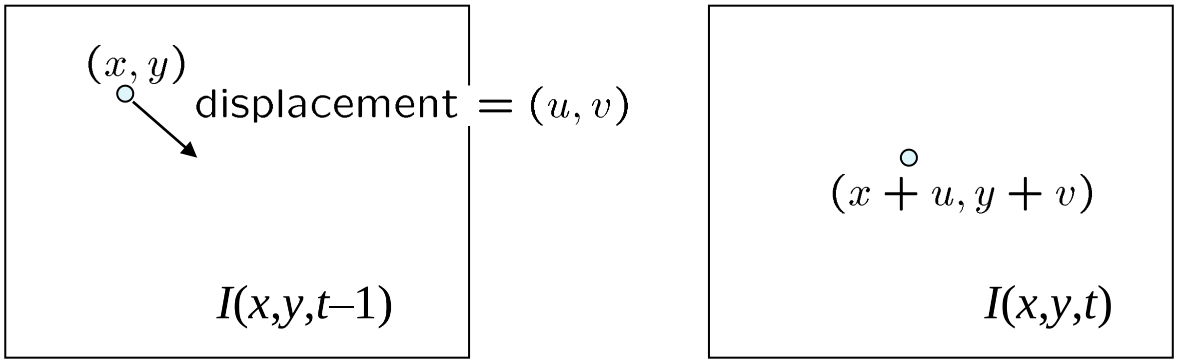
Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.

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

Figure by Michael Black

Slide credit: Kristen Grauman

The brightness constancy constraint



Brightness Constancy Equation:

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

Linearizing the right side using Taylor expansion:

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

Hence, $I_x u + I_y v + I_t \approx 0$

Slide credit: Kristen Grauman

The brightness constancy constraint

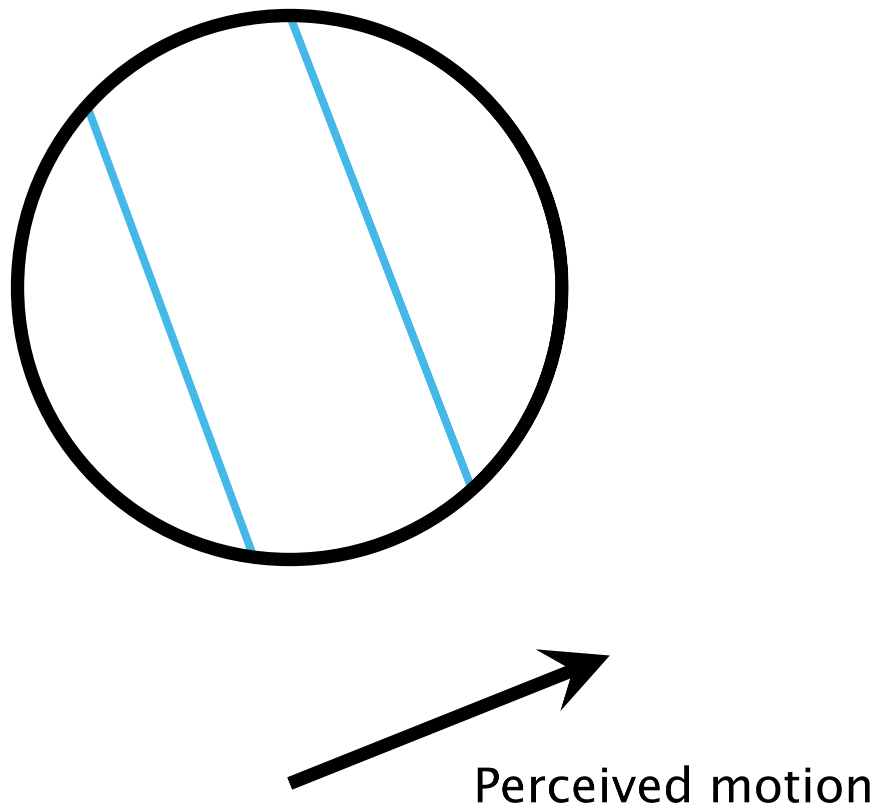
- How many equations and unknowns per pixel?
  - One equation per pixel
  - u and v are unknown!

$$I_x u + I_y v + I_t = 0$$

$$\nabla I \cdot (u, v) + I_t = 0$$

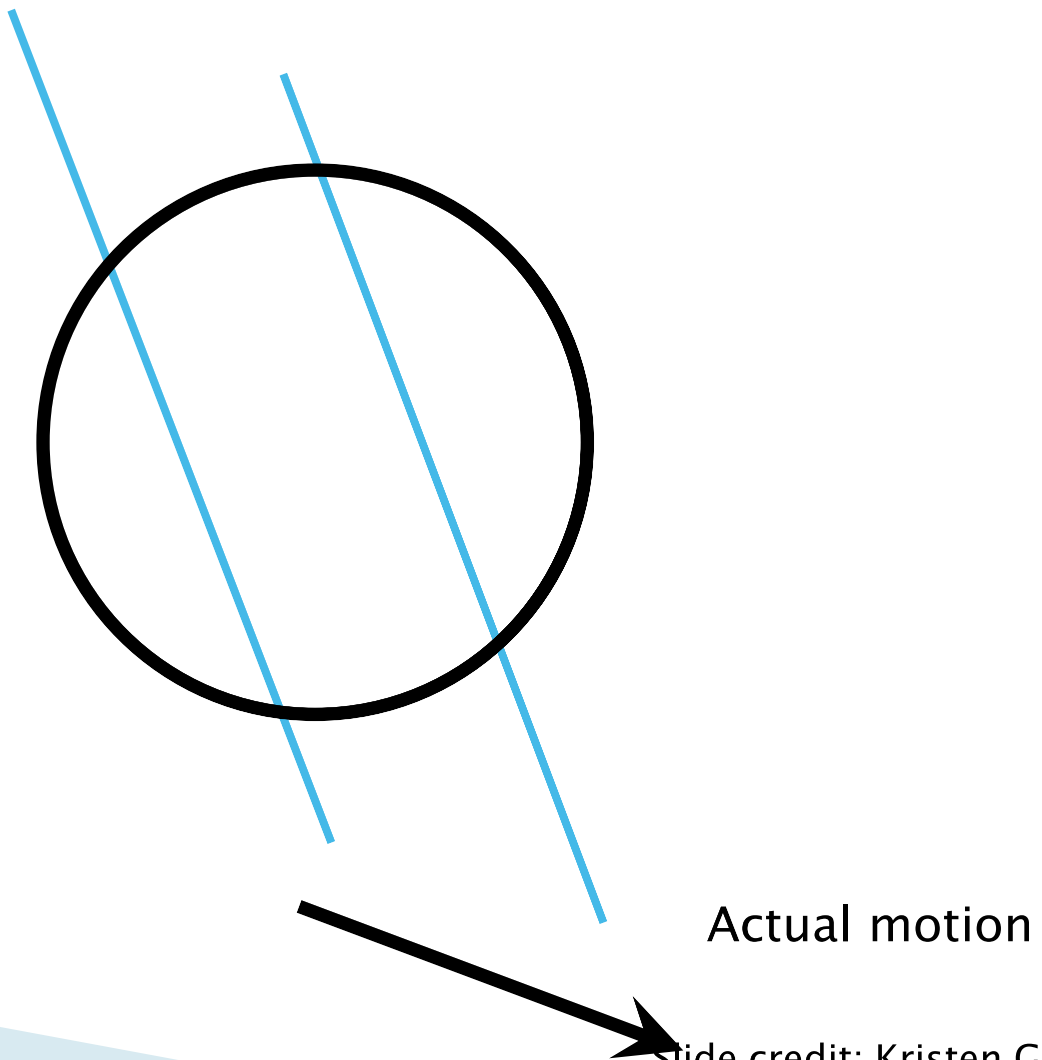
Slide credit: Kristen Grauman

The aperture problem



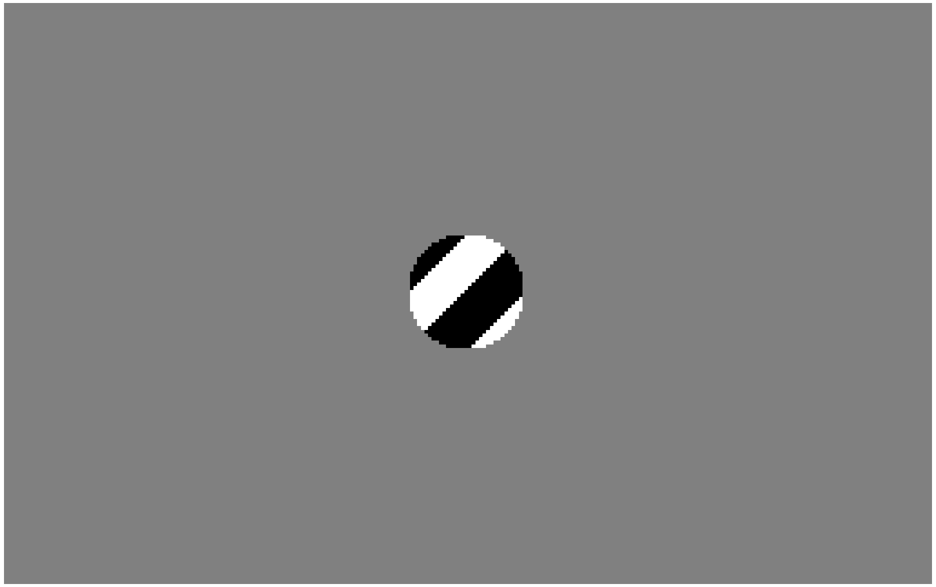
Slide credit: Kristen Grauman

The aperture problem



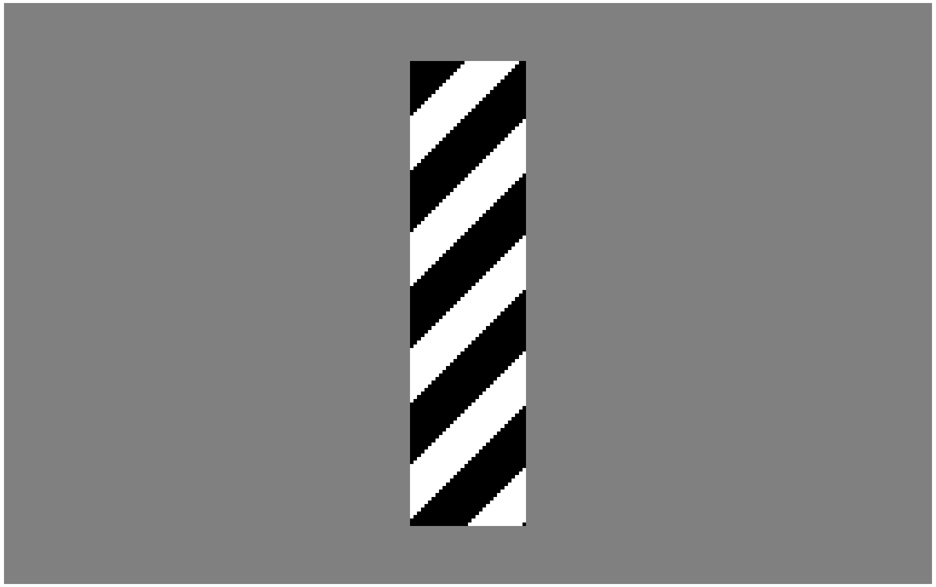
Slide credit: Kristen Grauman

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

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

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

$$\underset{25\text{x}2}{A} \;\; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Slide credit: Steve Seitz

Solving the aperture problem

Prob: we have more equations than unknowns

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

Solution: solve least squares problem

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

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

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

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

Slide credit: Steve Seitz

Conditions for solvability

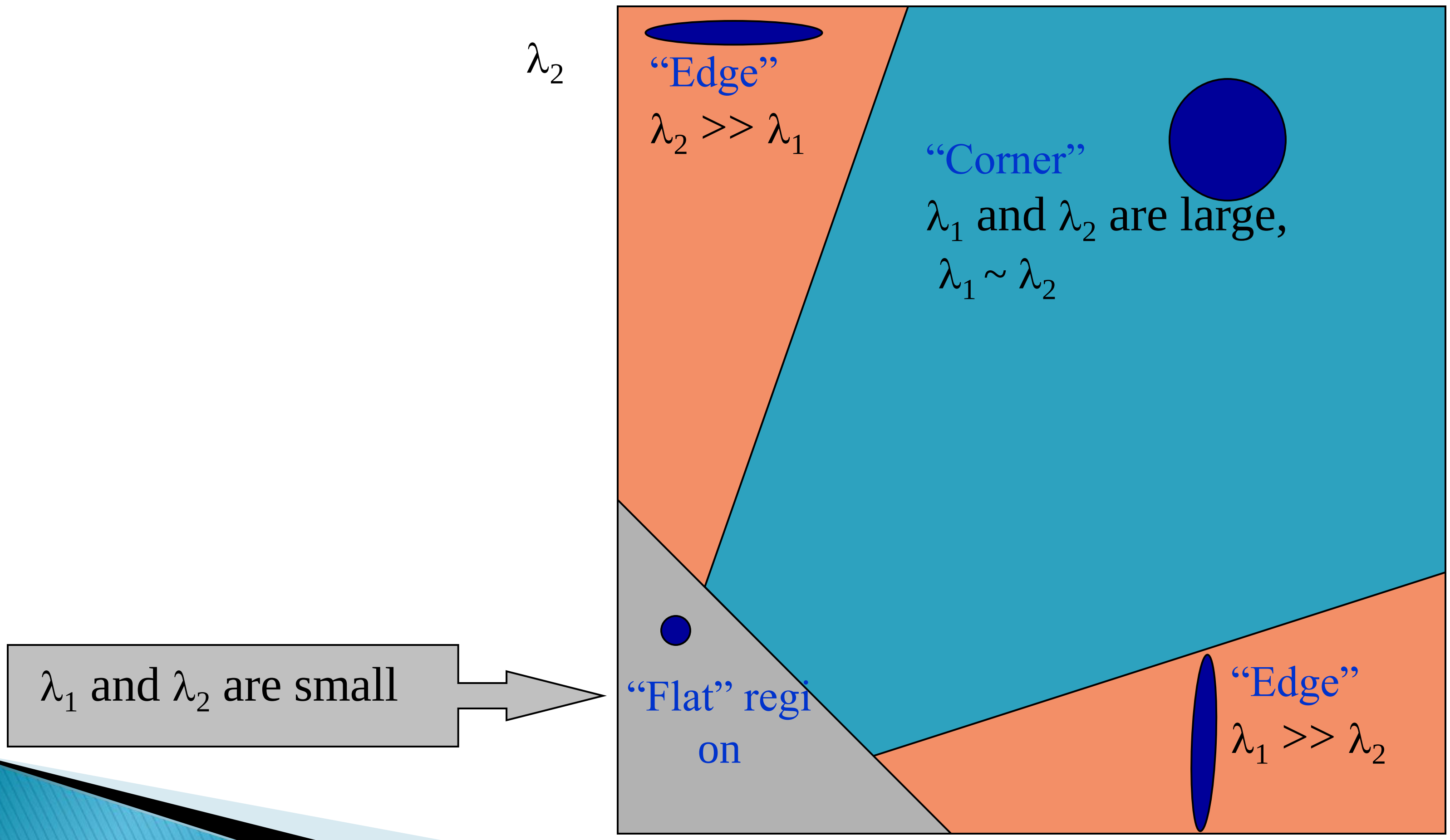
$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

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

When is this solvable?

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

Recall: Harris Corner Detector…



Edge

- gradients very large or very small
- large λ_1, small λ_2

Slide credit: Kristen Grauman

Low-texture region

- gradients have small magnitude
- small λ_1, small λ_2

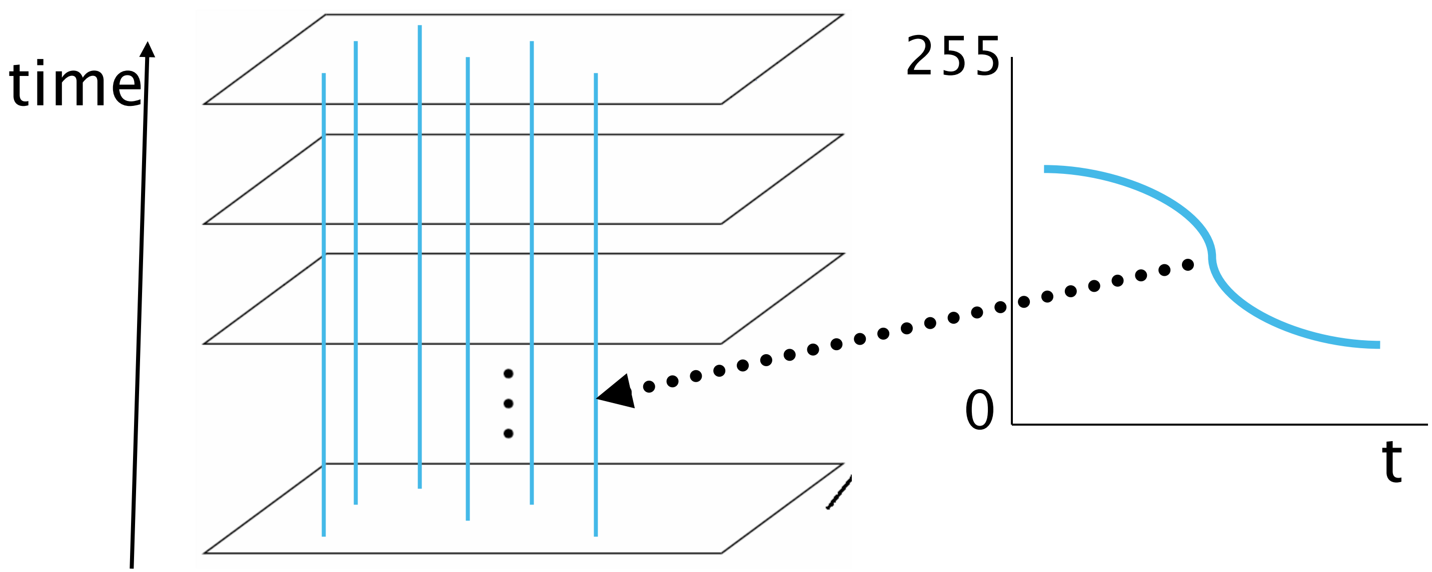
Slide credit: Kristen Grauman

High-texture region

- gradients are different, large magnitudes
- large λ_1, large λ_2

Background Subtraction

Video as an "Image Stack"



- Can look at video data as a spatio-temporal volume
  - If camera is stationary, each line through time corresponds to a single ray in space

Alyosha Efros, CMU

Background Subtraction

- Given an image (mostly likely to be a video frame), we want to identify the **foreground objects** in that image!

$\Rightarrow$

Motivation

- In most cases, objects are of interest, not the scene.
- Makes our life easier: less processing costs, and less room for error.

Slide credit: Birgi Tamersoy

Background subtraction

- Simple techniques can do ok with static camera
- …But hard to do perfectly

- Widely used:
  - Traffic monitoring (counting vehicles, detecting & tracking vehicles, pedestrians),
  - Human action recognition (run, walk, jump, squat),
  - Human-computer interaction
  - Object tracking

Simple Approach

Image at time t: $I(x, y, t)$ $\Downarrow$

Background at time t: $B(x, y, t)$ $\Downarrow$

$|$ 

$-$ 

$| > Th$

1. Estimate the background for time t.
2. Subtract the estimated background from the input frame.
3. Apply a threshold, Th, to the absolute difference to get the **foreground mask**.

But, how can we estimate the background?

Slide credit: Birgi Tamersoy

Frame Differencing

- Background is estimated to be the previous frame. Background subtraction equation then becomes:

$$B(x, y, t) = I(x, y, t-1)$$
$$\Downarrow$$
$$|I(x, y, t) - I(x, y, t-1)| > Th$$

- Depending on the object structure, speed, frame rate and global threshold, this approach may or may **not** be useful (usually **not**).

$|$ 

$-$ 

$| > Th$

Slide credit: Birgi Tamer

Frame Differencing

$Th = 25$

$Th = 50$

$Th = 100$

$Th = 200$

Slide credit: Birgi Tamer

Mean Filter

- In this case the background is the mean of the previous n frames:

$$B(x,y,t) = \frac{1}{n}\sum_{i=0}^{n-1} I(x,y,t-i)$$
$$\Downarrow$$
$$|I(x,y,t) - \frac{1}{n}\sum_{i=0}^{n-1} I(x,y,t-i)| > Th$$

- For $n = 10$:

Estimated Background

Foreground Mask

Slide credit: Birgi Tamer

Median Filter

- Assuming that the background is more likely to appear in a scene, we can use the median of the previous n frames as the background model:

$$B(x, y, t) = median\{I(x, y, t - i)\}$$
$$\Downarrow$$
$$|I(x, y, t) - median\{I(x, y, t - i)\}| > Th \text{ where}$$
$$i \in \{0, \ldots, n - 1\}.$$

- For $n = 10$:

Estimated Background

Foreground Mask

Slide credit: Birgi Tamersoy

Average/Median Image

Alyosha Efros, CMU