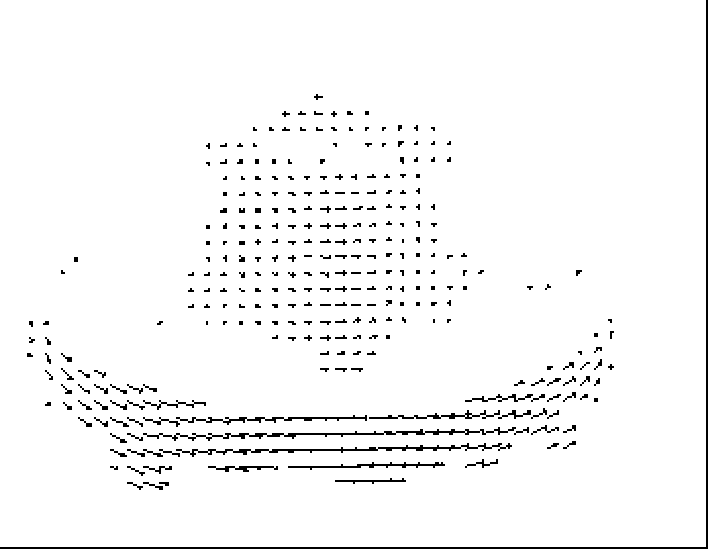


Motion and optical flow

Thursday, Nov 20

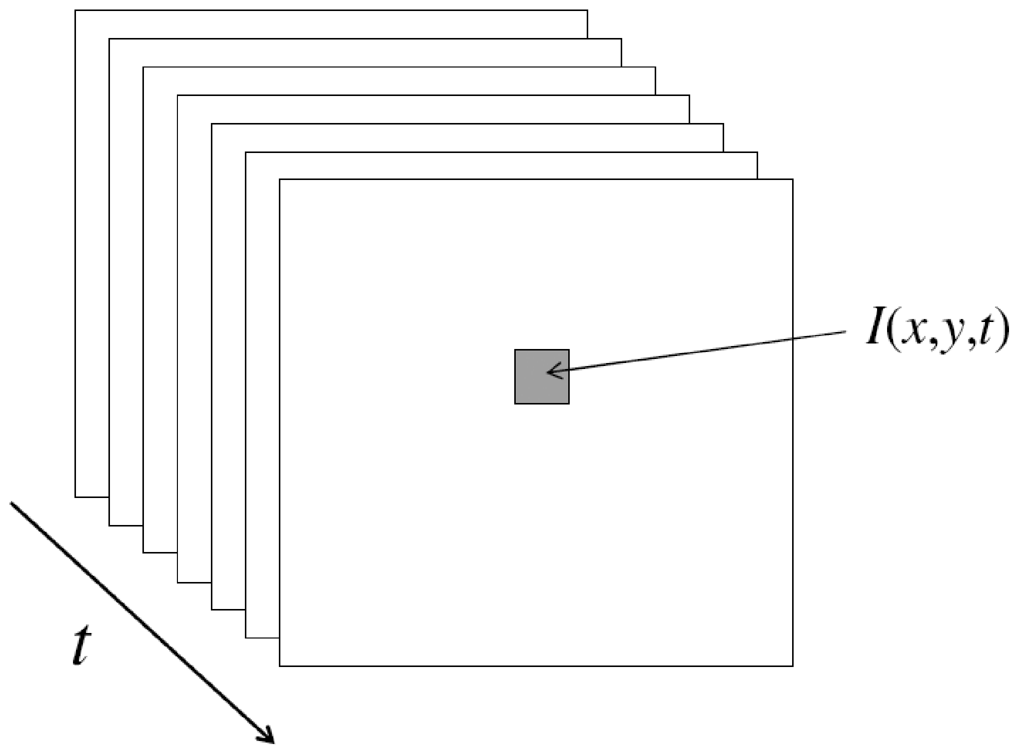
Many slides adapted from S. Seitz, R. Szeliski, M. Pollefeys, S. Lazebnik

Today

- Pset 3 solutions
- Introduction to motion
- Motion fields
- Feature-based motion estimation
- Optical flow

Video

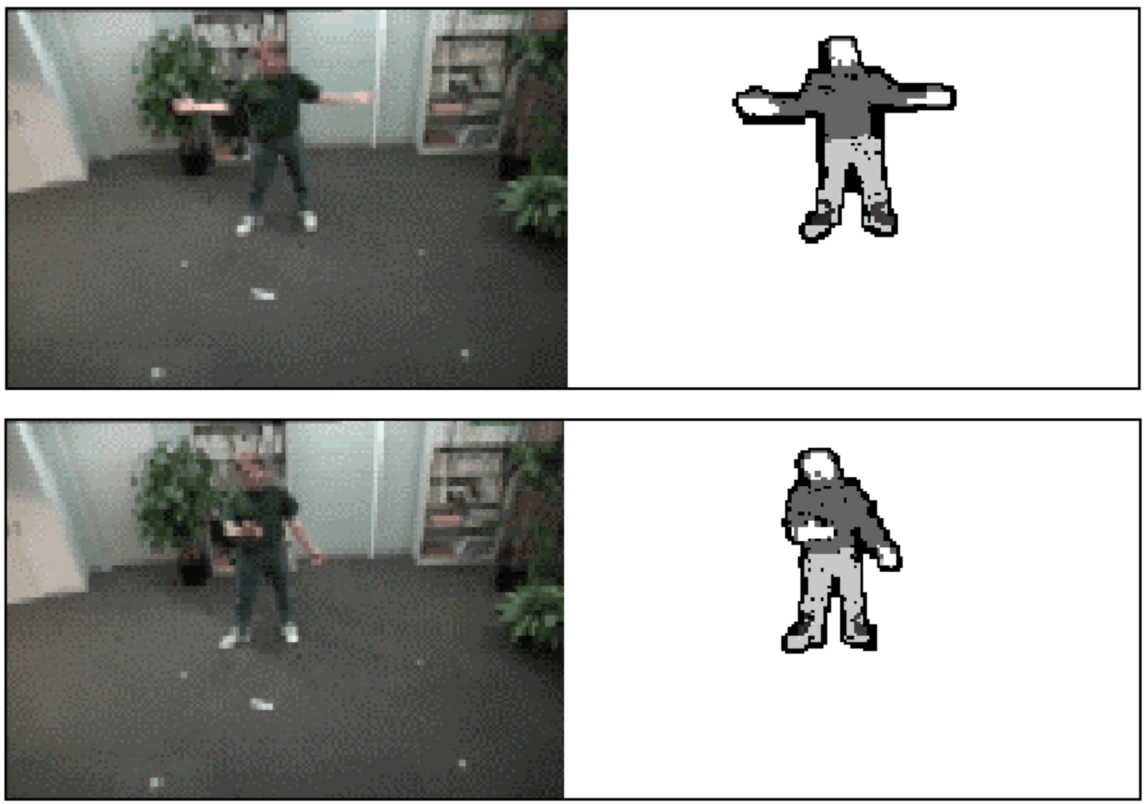
- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Applications of segmentation to video

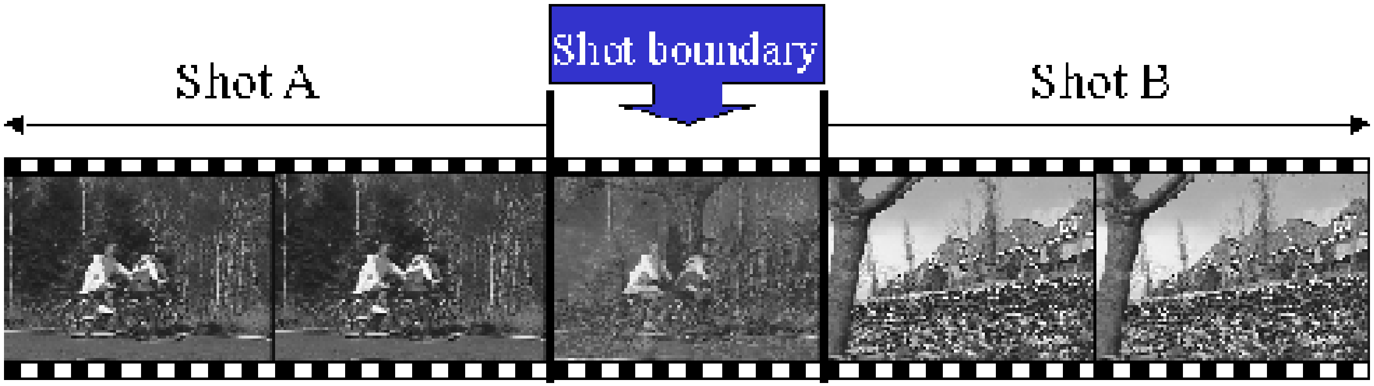
- Background subtraction
  - A static camera is observing a scene
  - Goal: separate the static *background* from the moving *foreground*

How to come up with background frame estimate without access to "empty" scene?



Applications of segmentation to video

- Background subtraction
- Shot boundary detection
  - Commercial video is usually composed of *shots* or sequences showing the same objects or scene
  - Goal: segment video into shots for summarization and browsing (each shot can be represented by a single keyframe in a user interface)
  - Difference from background subtraction: the camera is not necessarily stationary

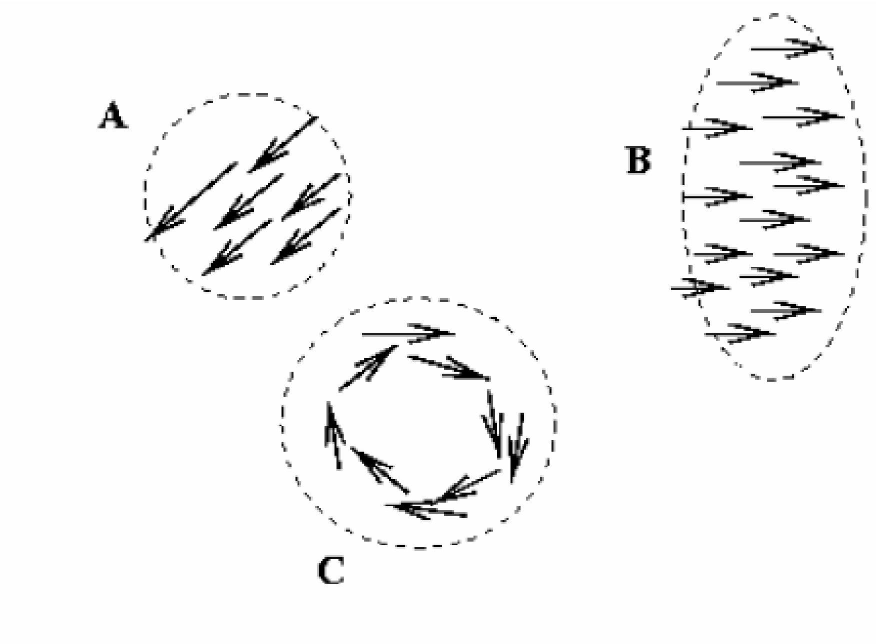


Applications of segmentation to video

- Background subtraction
- Shot boundary detection
  - For each frame
    - Compute the distance between the current frame and the previous one
      - » Pixel-by-pixel differences
      - » Differences of color histograms
      - » Block comparison
    - If the distance is greater than some threshold, classify the frame as a shot boundary

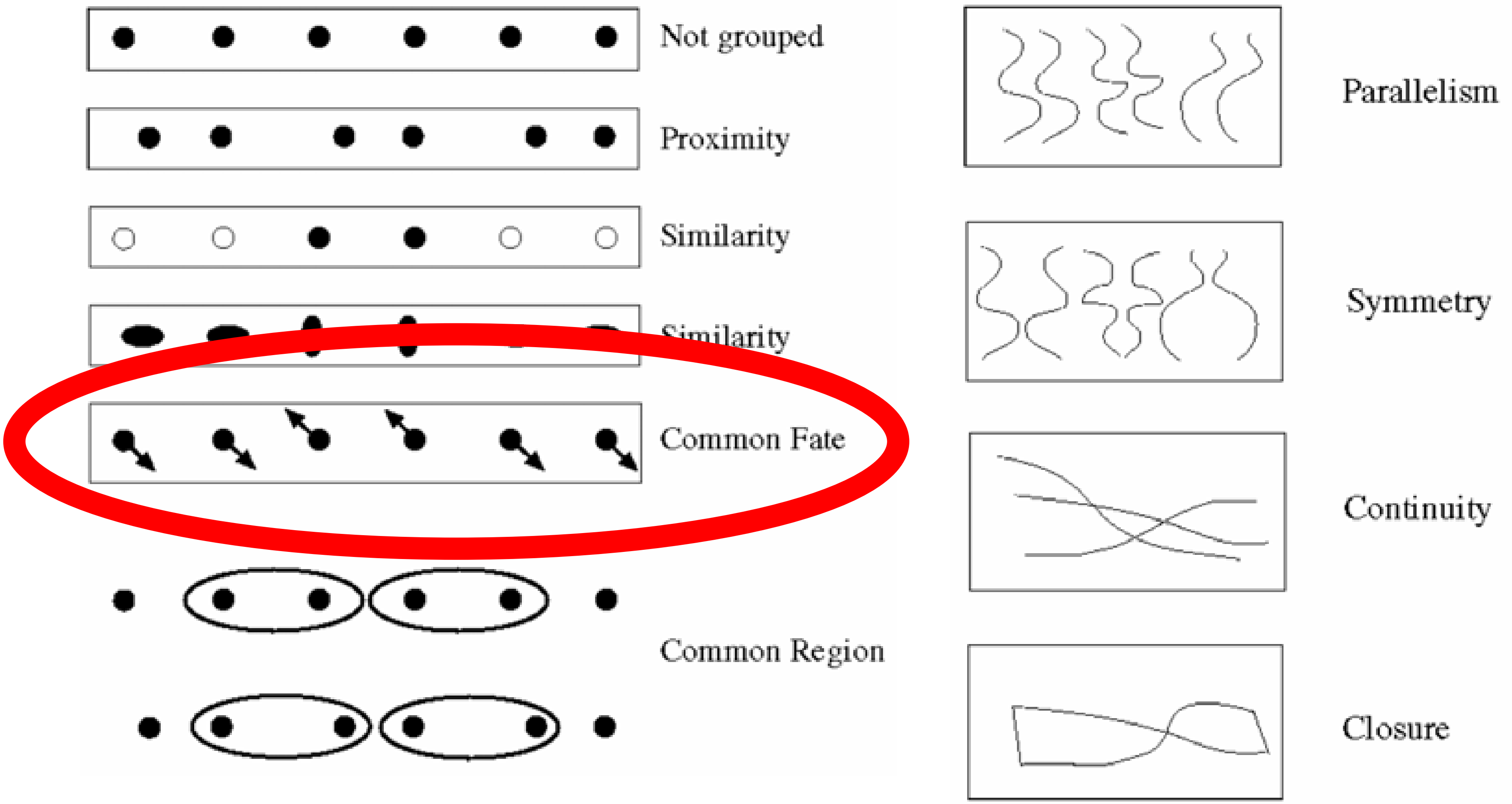
Applications of segmentation to video

- Background subtraction
- Shot boundary detection
- Motion segmentation
  - Segment the video into multiple *coherently* moving objects



Motion and perceptual organization

- Sometimes, motion is the only cue

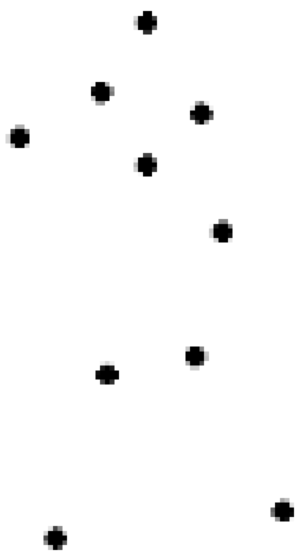


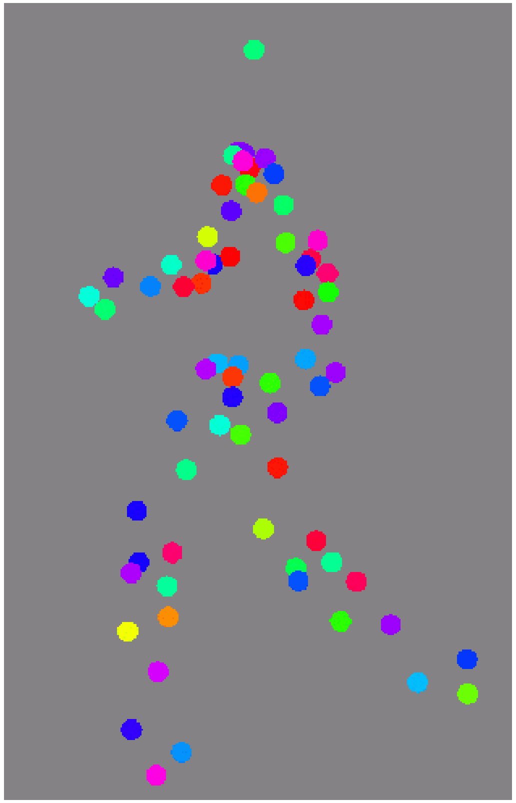
Motion and perceptual organization

- Sometimes, motion is foremost cue

Motion and perceptual organization

- Even “impoverished” motion data can evoke a strong percept





https://perswww.kuleuven.be/~u0064325/Talks/2013_Bremen_DescartesSelfishError/images/wwd_ervin_movieUp15.gif

Uses of motion

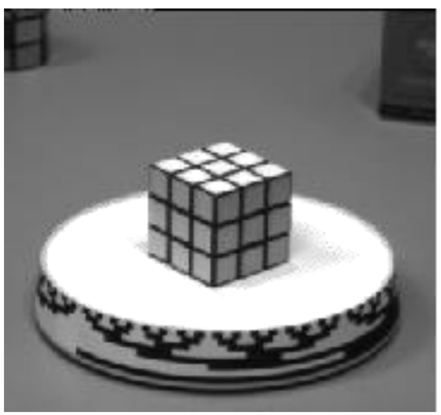
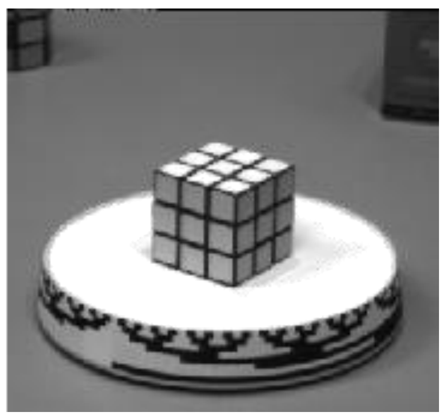
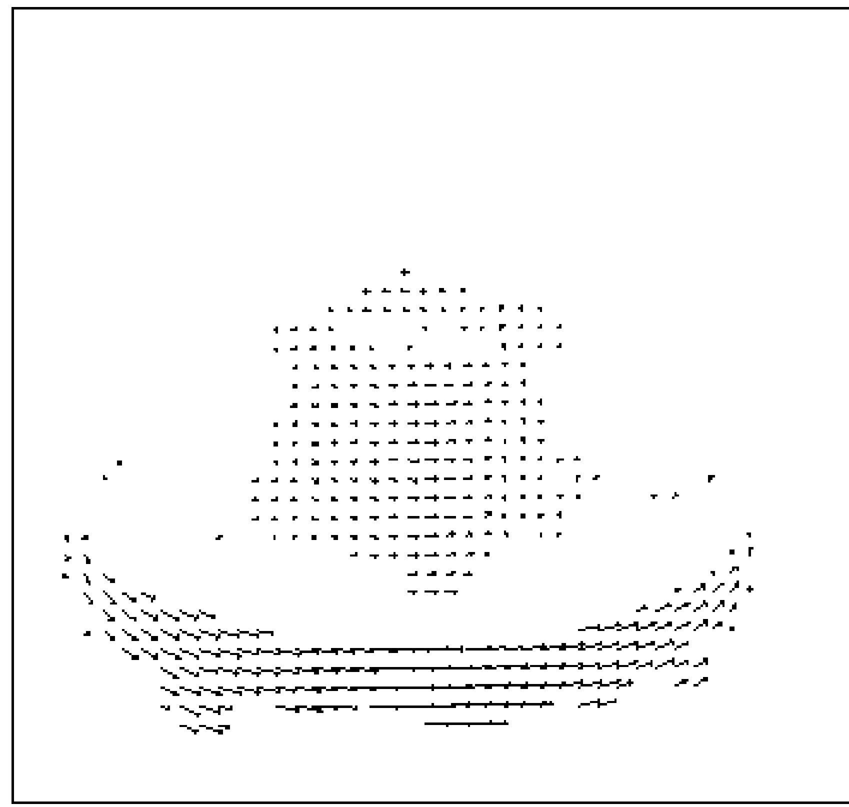
- Estimating 3D structure
- Segmenting objects based on motion cues
- Learning dynamical models
- Recognizing events and activities
- Improving video quality (motion stabilization)

Today

- Pset 3 solutions
- Introduction to motion
- Motion fields
- Feature-based motion estimation
- Optical flow

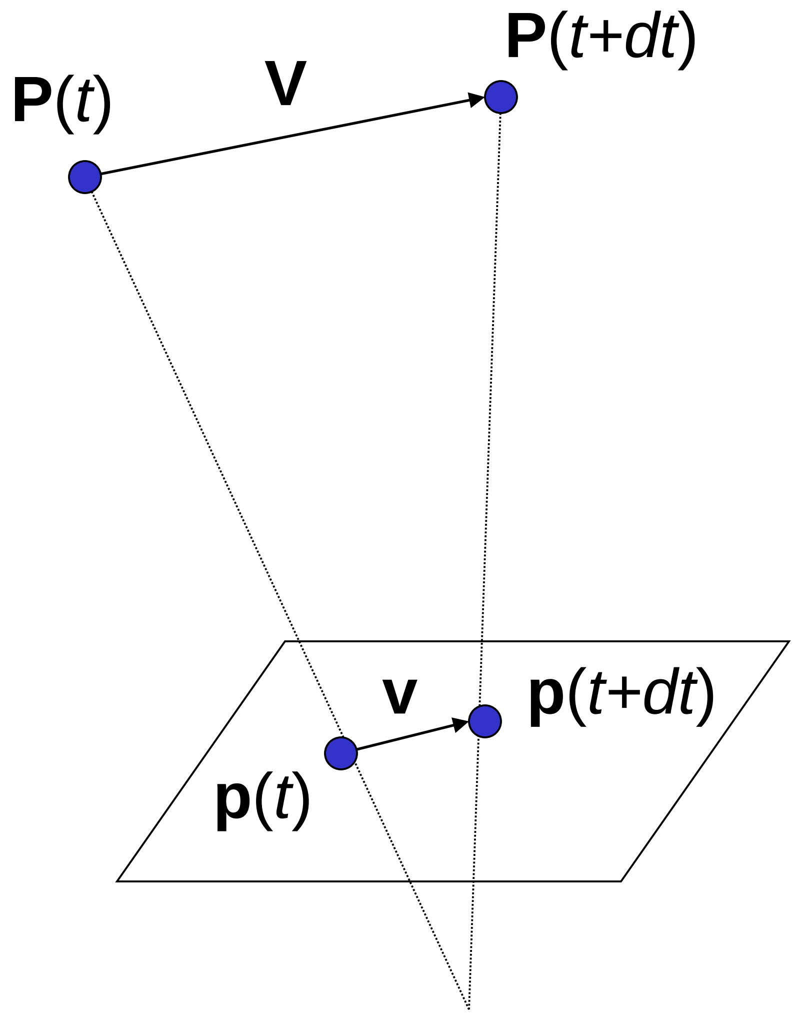
Motion field

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

Motion field and parallax

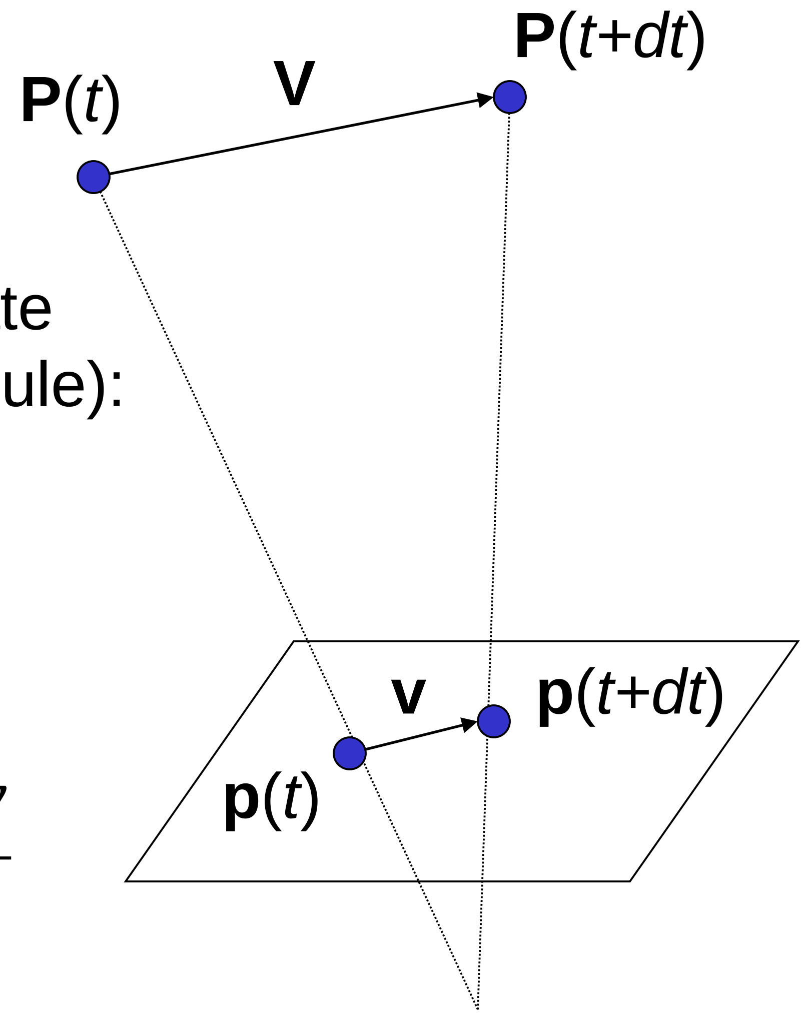
- $\mathbf{P}(t)$ is a moving 3D point
- Velocity of scene point: $\mathbf{V} = \mathrm{d}\mathbf{P}/\mathrm{d}t$
- $\mathbf{p}(t) = (x(t),y(t))$ is the projection of $\mathbf{P}$ in the image
- Apparent velocity $\mathbf{v}$ in the image: given by components $v_x = \mathrm{d}x/\mathrm{d}t$ and $v_y = \mathrm{d}y/\mathrm{d}t$
- These components are known as the *motion field* of the image



Motion field and parallax

Quotient rule:
D(f/g) = (g f' – g' f)/g^2

$$\mathbf{V} = (V_x, V_y, V_Z) \qquad \mathbf{p} = f\,\frac{\mathbf{P}}{Z}$$



To find image velocity **v**, differentiate **p** with respect to *t* (using quotient rule):

$$\mathbf{v} = f\,\frac{Z\mathbf{V} - V_z\mathbf{P}}{Z^2}$$

$$v_x = \frac{fV_x - V_z x}{Z} \qquad v_y = \frac{fV_y - V_z y}{Z}$$

Image motion is a function of both the 3D motion (**V**) and the depth of the 3D point (*Z*)

Motion field and parallax

- Pure translation: **V** is constant everywhere

$$v_x = \frac{fV_x - V_z x}{Z}$$

$$v_y = \frac{fV_y - V_z y}{Z}$$

$$\mathbf{v} = \frac{1}{Z}(\mathbf{v}_0 - V_z \mathbf{p}),$$

$$\mathbf{v}_0 = \left(fV_x, fV_y\right)$$

Motion field and parallax

- Pure translation: **V** is constant everywhere

$$\mathbf{v} = \frac{1}{Z}(\mathbf{v}_0 - V_z \mathbf{p}),$$

$$\mathbf{v}_0 = \left(f V_x, f V_y \right)$$

- V_z is nonzero:
  - Every motion vector points toward (or away from) $\mathbf{v}_0$, the vanishing point of the translation direction

Motion field and parallax

- Pure translation: **V** is constant everywhere

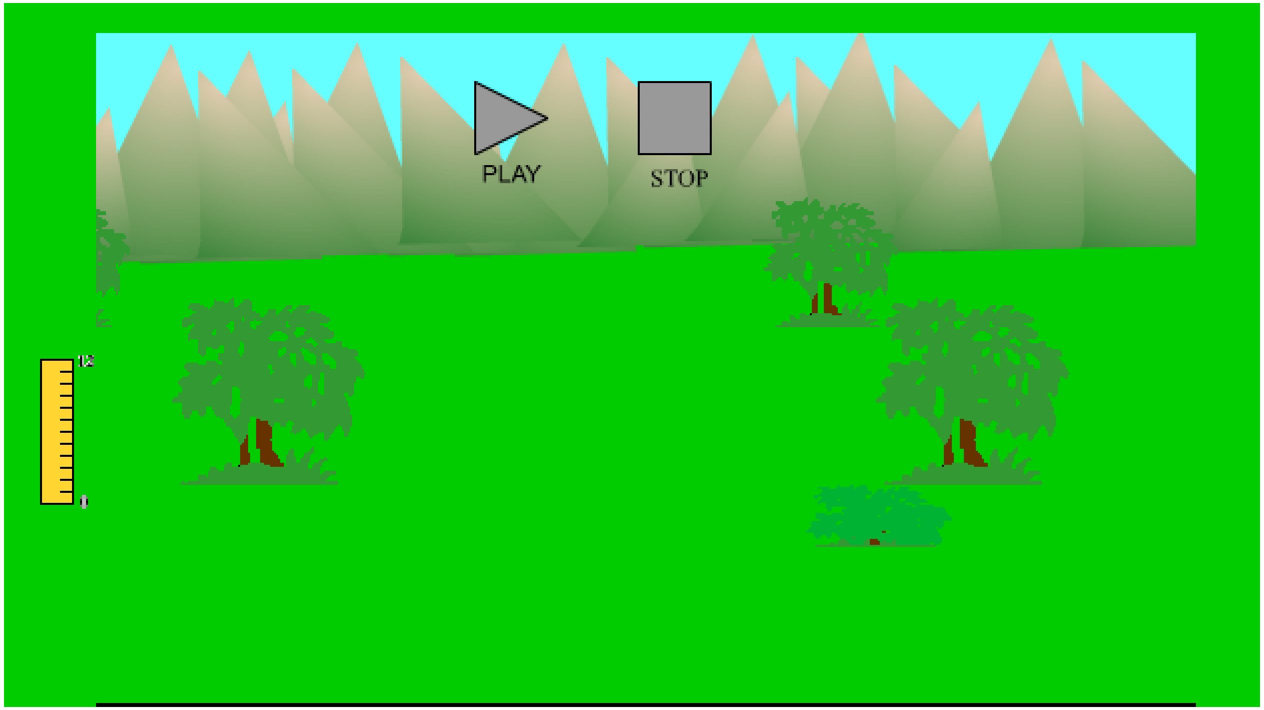
$$\mathbf{v} = \frac{1}{Z}(\mathbf{v}_0 - V_z \mathbf{p}),$$

$$\mathbf{v}_0 = \left(f V_x, f V_y \right)$$

- V_z is nonzero:
  - Every motion vector points toward (or away from) $\mathbf{v}_0$, the vanishing point of the translation direction
- V_z is zero:
  - Motion is parallel to the image plane, all the motion vectors are parallel
- The length of the motion vectors is inversely proportional to the depth Z

Motion parallax

http://psych.hanover.edu/KRANTZ/MotionParallax/MotionParallax.html



Motion field + camera motion

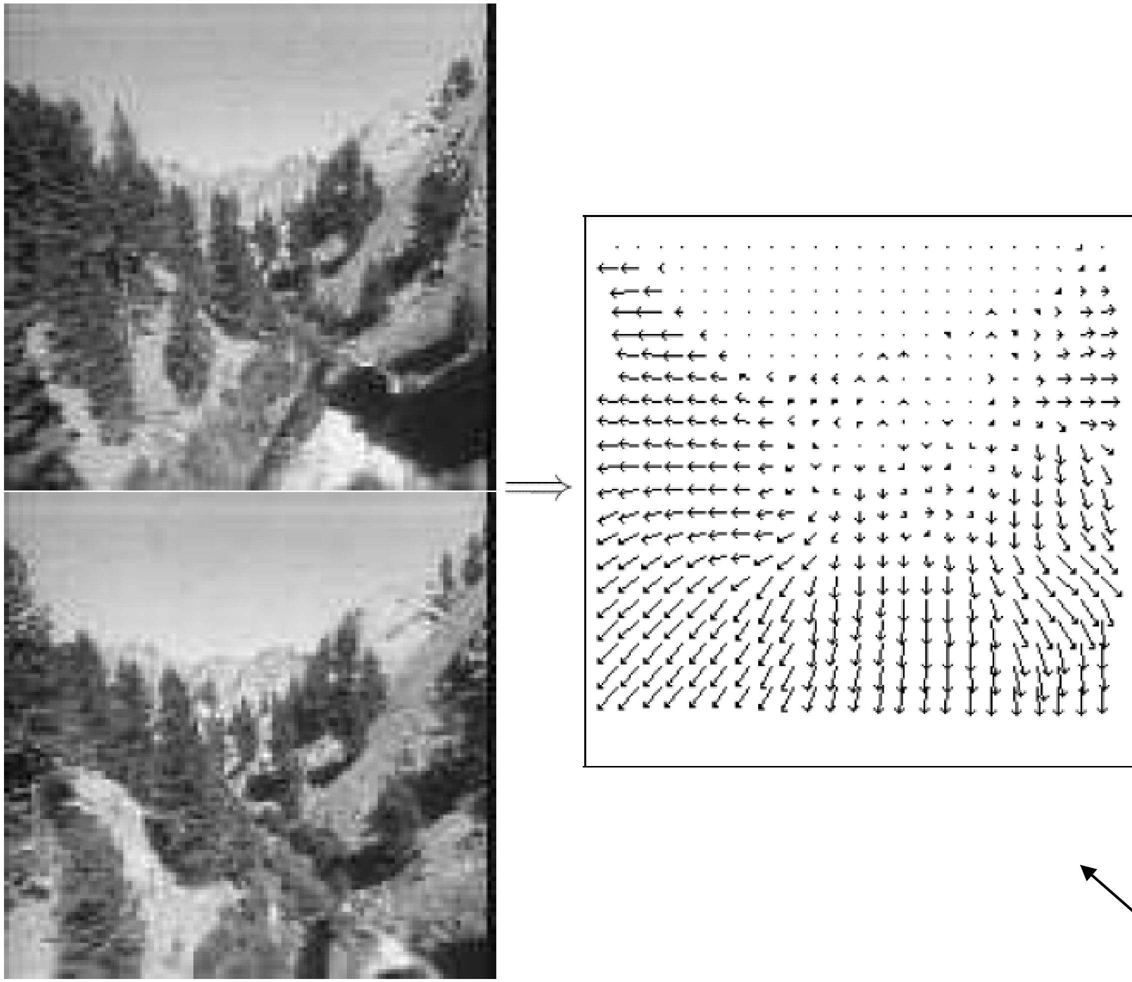


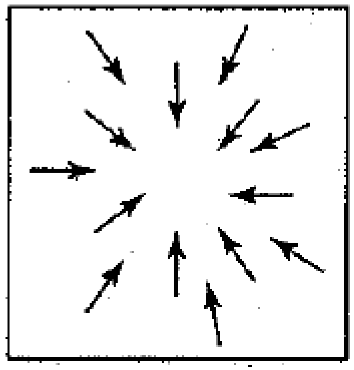
Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

Length of flow vectors inversely proportional to depth Z of 3d point

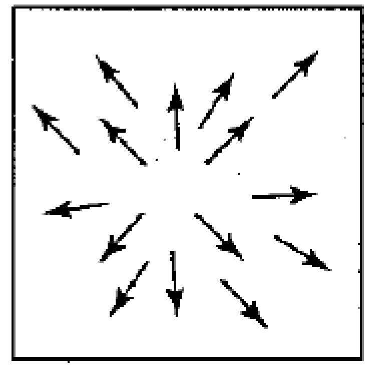
points closer to the camera move more quickly across the image plane

Figure from Michael Black, Ph.D. Thesis

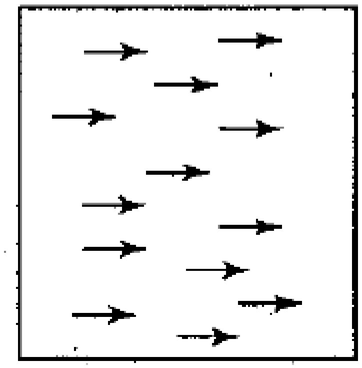
Motion field + camera motion



Zoom out



Zoom in



Pan right to left

Motion estimation techniques

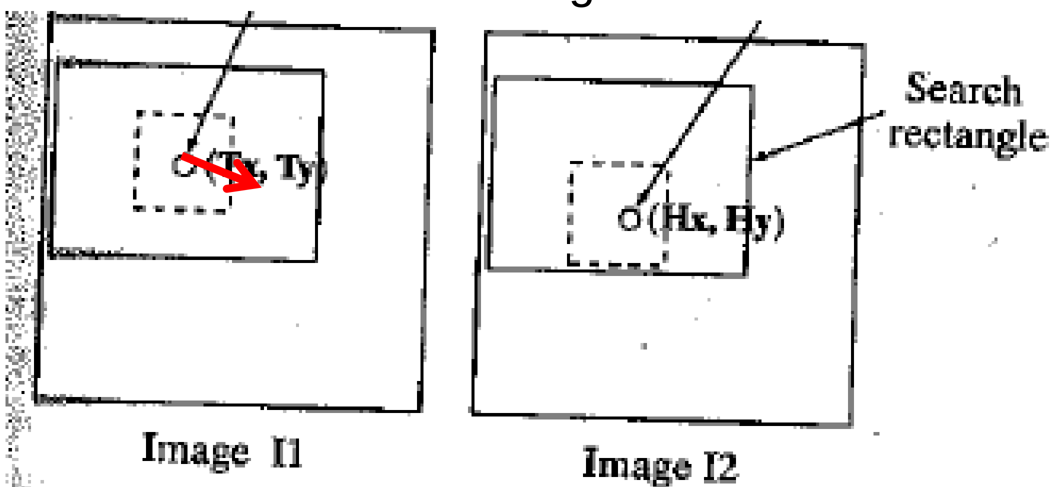
- Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small

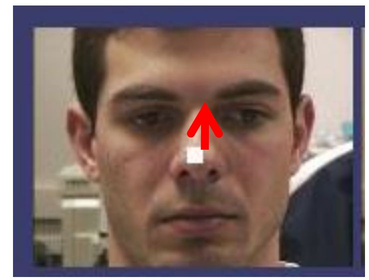
Feature-based matching for motion

Interesting point

Best matching neighborhood



Time t



Time t+1

A Camera Mouse

Video interface: use feature tracking as mouse replacement

- User clicks on the feature to be tracked
- Take the 15x15 pixel square of the feature
- In the next image do a search to find the 15x15 region with the highest correlation
- Move the mouse pointer accordingly
- Repeat in the background every 1/30th of a second

James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

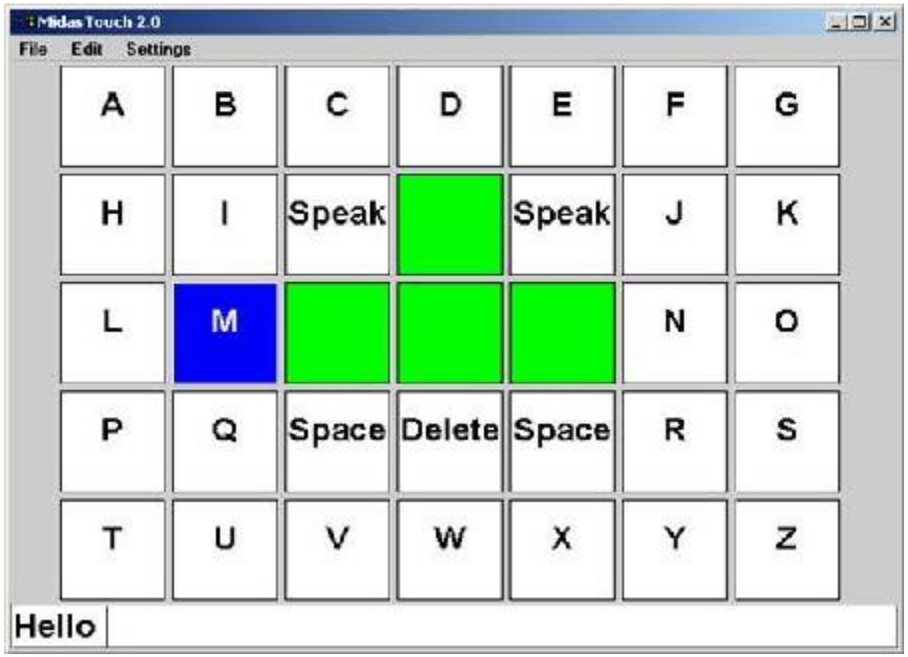
A Camera Mouse

Specialized software for communication, games

James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

A Camera Mouse

Specialized software for communication, games



James Gips and Margrit Betke
http://www.bc.edu/schools/csom/eagleeyes/

What are good features to track?

- Recall the Harris corner detector
- Can measure quality of features from just a single image
- Automatically select candidate “templates”

Motion estimation techniques

- Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small

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

Apparent motion ~= motion field

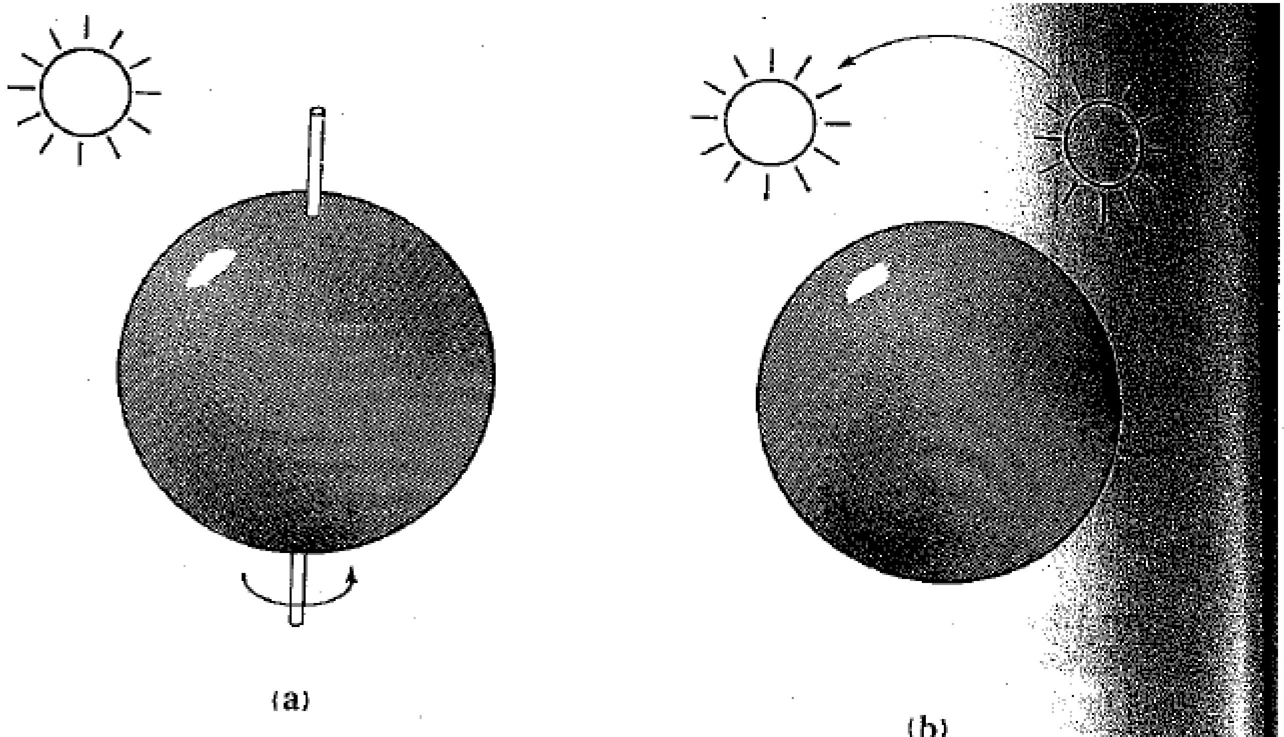
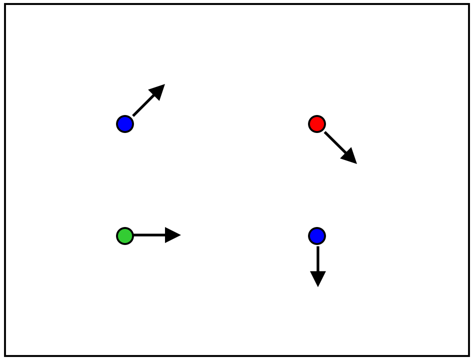


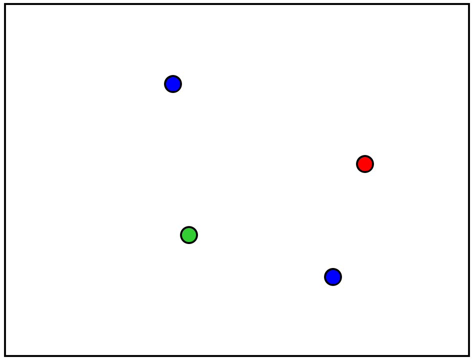
Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field is zero.

Figure from Horn book

Estimating optical flow



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



$I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field 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

Brightness constancy

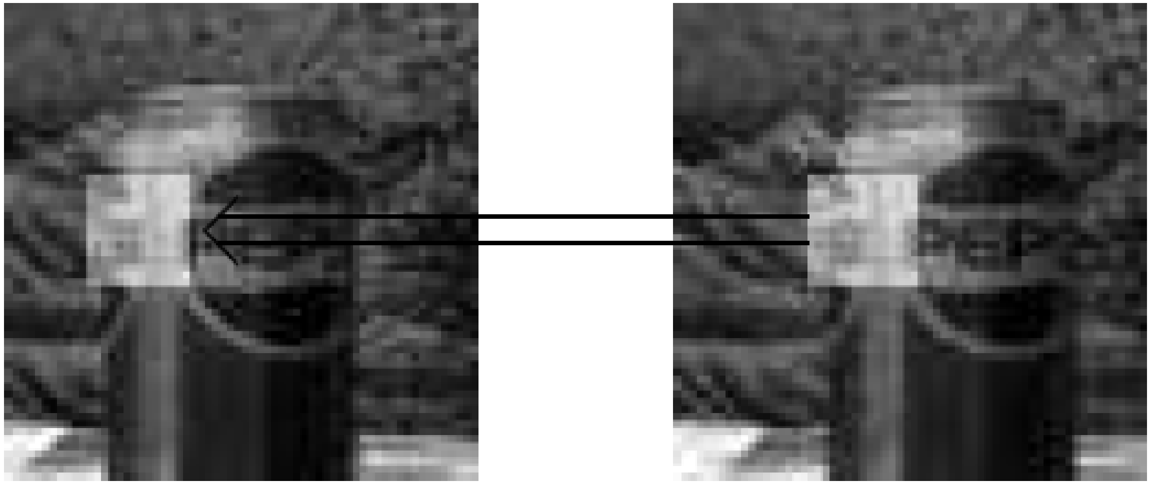
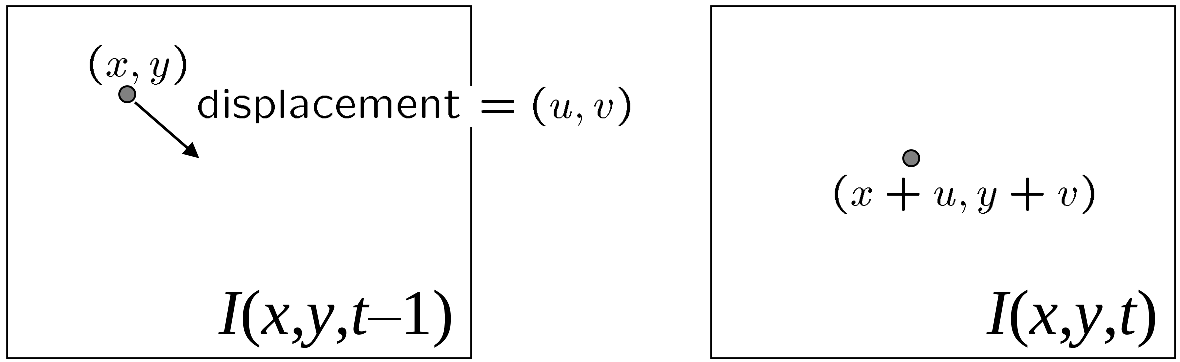


Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.

Figure by Michael Black

The brightness constancy constraint



Brightness Constancy Equation:

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

Can be written as:

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

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

So, $$I_x \cdot u + I_y \cdot v + I_t \approx 0$$

The brightness constancy constraint

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

- How many equations and unknowns per pixel?
  - One equation, two unknowns
- Intuitively, what does this constraint mean?

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

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

The brightness constancy constraint

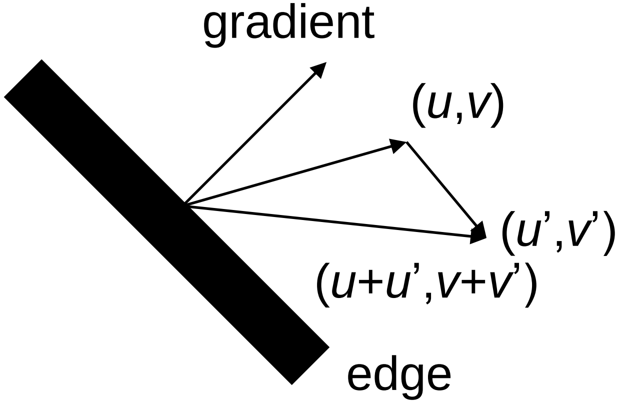
$$I_x \cdot u + I_y \cdot v + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation, two unknowns
- Intuitively, what does this constraint mean?

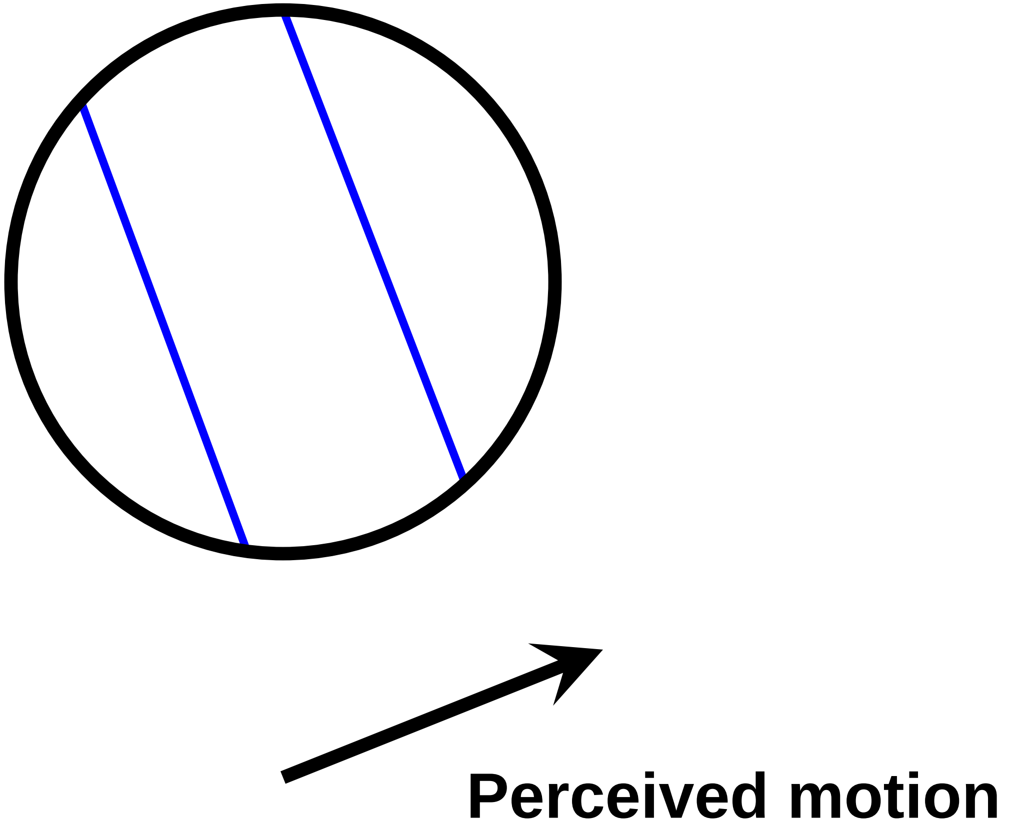
$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

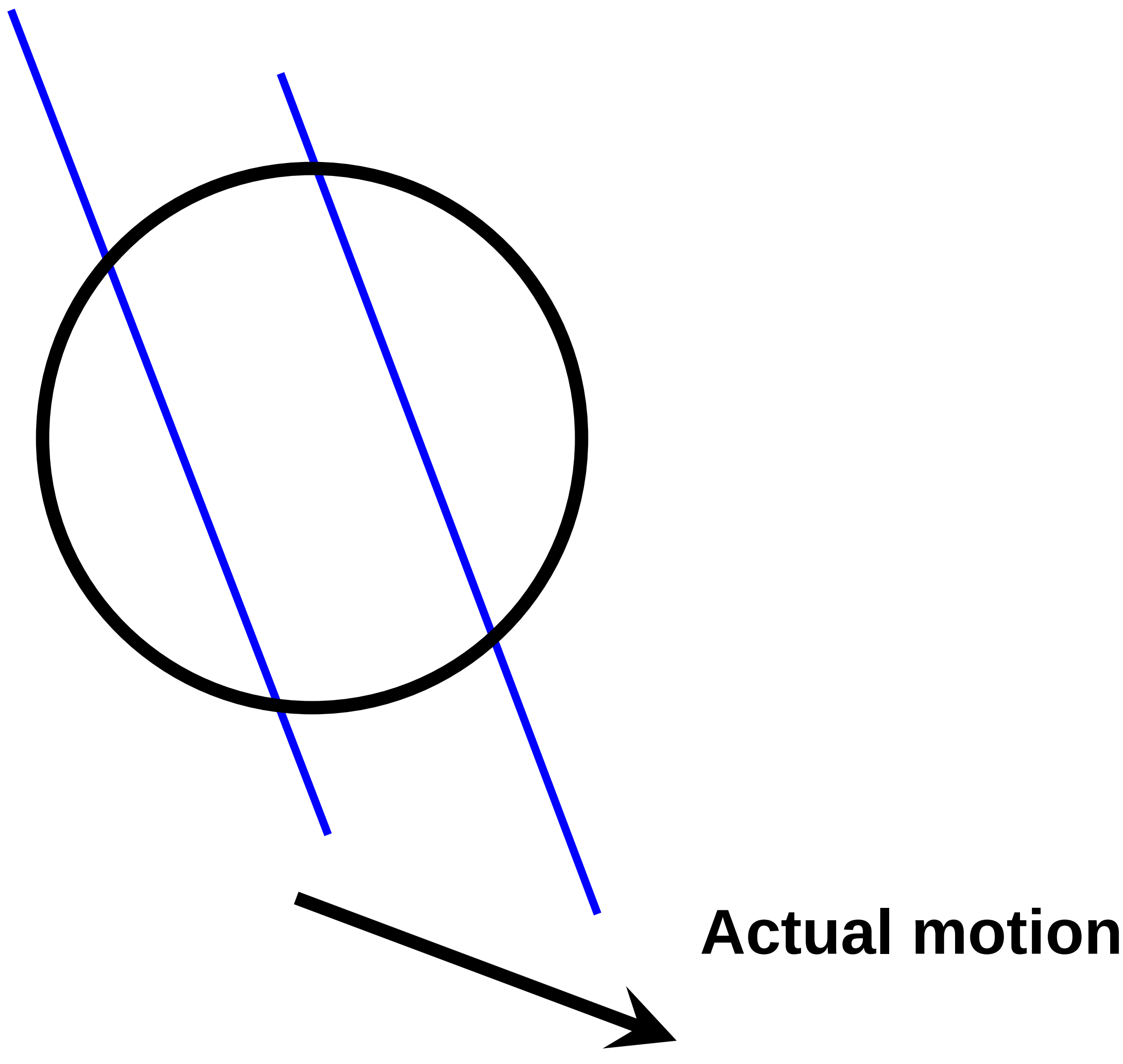
If (u, v) satisfies the equation,
so does ($u+u'$, $v+v'$) if $\nabla I \cdot (u', v') = 0$



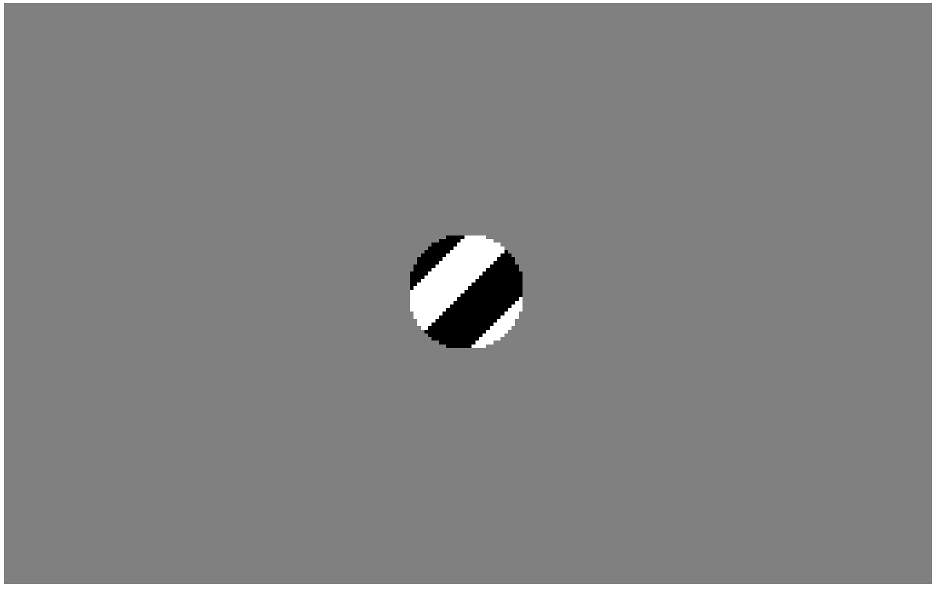
The aperture problem



The aperture problem

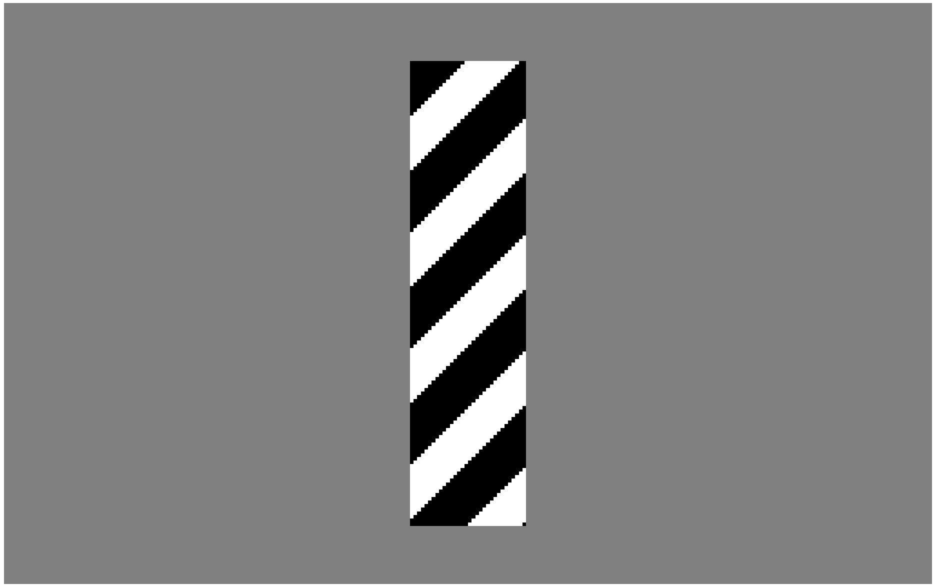


The barber pole illusion



http://en.wikipedia.org/wiki/Barberpole_illusion

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 (grayscale image)

- How to get more equations for a pixel?
- **Spatial coherence constraint:** pretend the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

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

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

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

Solving the aperture problem

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)

Conditions for solvability

$$\left[\begin{array}{cc} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{array} \right] \left[\begin{array}{c} u \\ v \end{array} \right] = - \left[\begin{array}{c} \sum I_x I_t \\ \sum I_y I_t \end{array} \right]$$

$$\underset{}{A^T A} \qquad\qquad\qquad\qquad A^T b$$

When is this solvable?

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

Slide by Steve Seitz, UW

Edge

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

Low-texture region

– gradients have small magnitude

– small λ_1, small λ_2

High-texture region

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

Example use of optical flow: Motion Paint

Use optical flow to track brush strokes, in order to animate them to follow underlying scene motion.

http://www.fxguide.com/article333.html

Motion vs. Stereo: Similarities

- Both involve solving
  - Correspondence: disparities, motion vectors
  - Reconstruction

Motion vs. Stereo: Differences

- Motion:
  - Uses velocity: consecutive frames must be close to get good approximate time derivative
  - 3d movement between camera and scene not necessarily single 3d rigid transformation
- Whereas with stereo:
  - Could have any disparity value
  - View pair separated by a single 3d transformation

Summary

- Motion field: 3d motions projected to 2d images; dependency on depth
- Solving for motion with
  - sparse feature matches
  - dense optical flow
- Optical flow
  - Brightness constancy assumption
  - Aperture problem
  - Solution with spatial coherence assumption