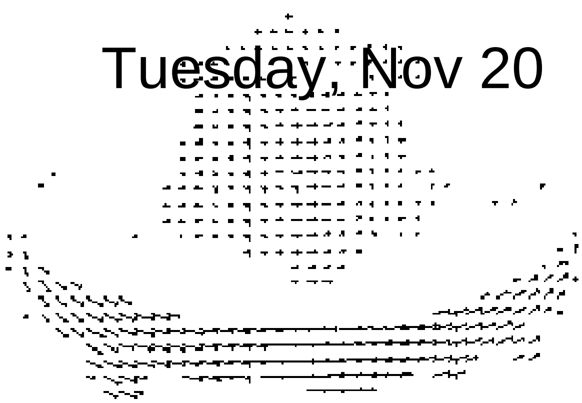
Lecture 19: Motion

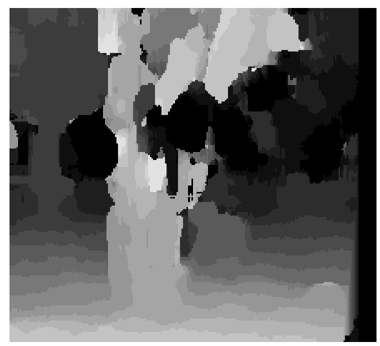
Tuesday, Nov 20



- Review Problem set 3
  - Dense stereo matching
  - Sparse stereo matching
  - Indexing scenes

Effect of window size

W = 3



W = 20

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

Figures from Li Zhang

Sources of error in correspondences

- Low-contrast / textureless image regions
- Occlusions
- Camera calibration errors
- Poor image resolution
- Violations of brightness constancy (specular reflections)
- Large motions

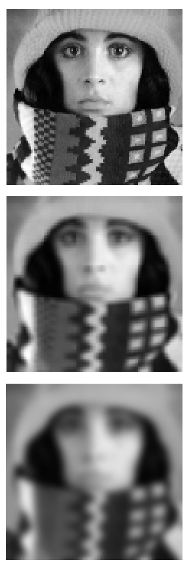
Sparse matching

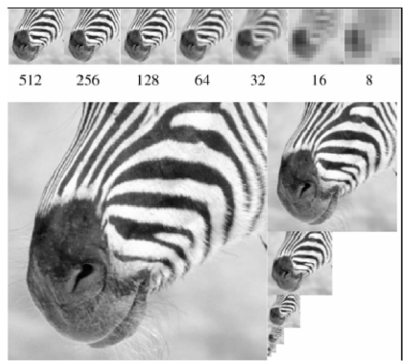
Indexing scenes

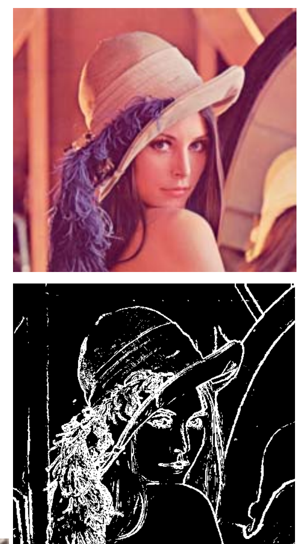
So far

- Features and filters
- Grouping, segmentation, fitting
- Multiple views, stereo, matching
- Recognition and learning

So far: Features and filters

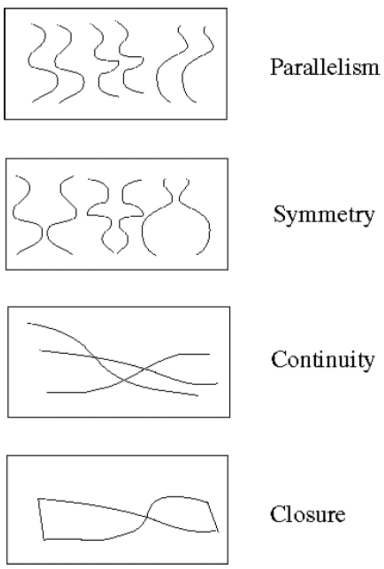


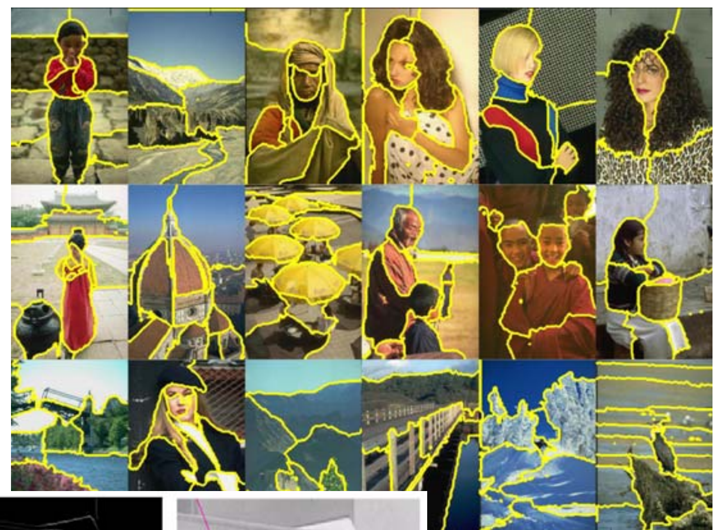




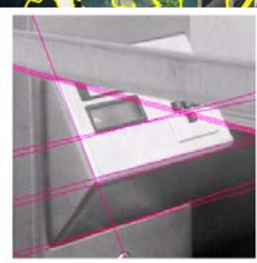
Transforming and describing images; textures and colors

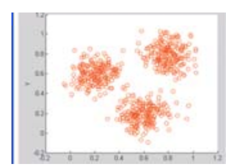
So far: Grouping





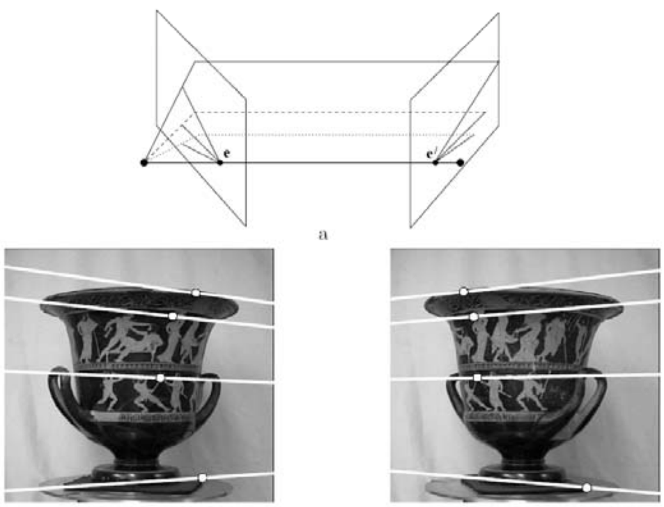
[fig from Shi et al]



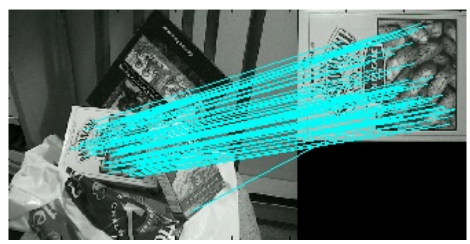


Clustering, segmentation, fitting; what parts belong together?

So far: Multiple views

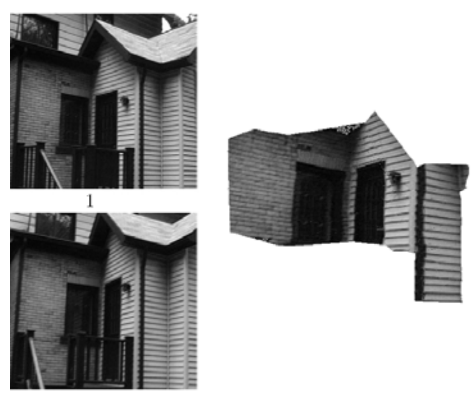


Hartley and Zisserman



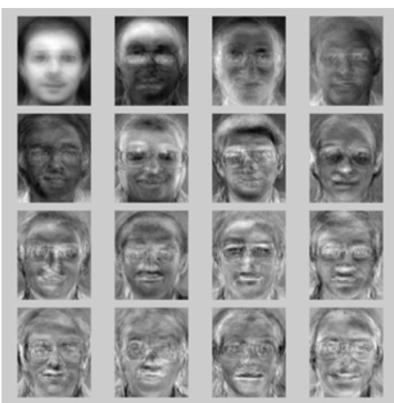
Lowe

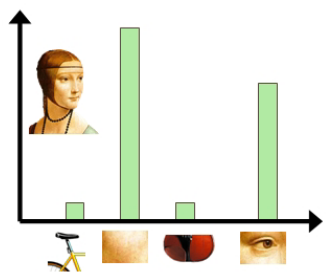
Multi-view geometry and matching, stereo

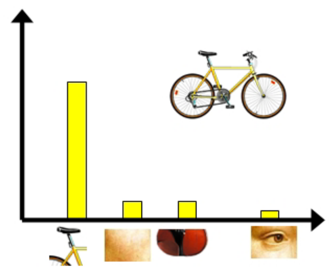


Tomasi and Kanade

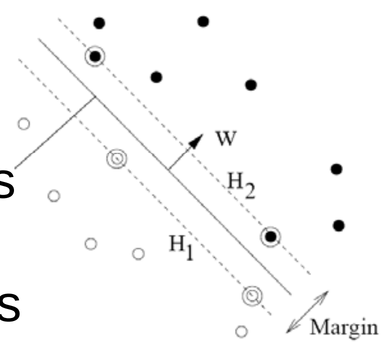
So far: Recognition and learning







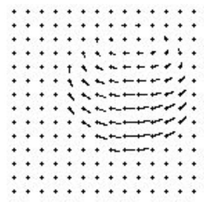
Shape matching,
recognizing objects
and categories,
learning techniques

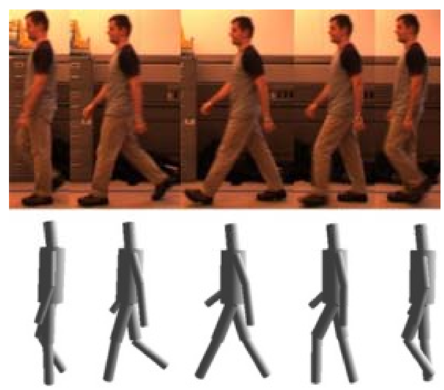


11/13 11/15		Rapid Object Detection using a Boosted Cascade of Simple Features, by P. Viola and M. Jones, 2001. FP 22.5: SVMs Learning Gender with Support Faces, by B. Moghaddam and M. Yang. TPAMI 2002, F&G 2000 Unsupervised Learning of Models for Recognition, by M. Weber, M. Welling, and P. Perona, ECCV 2000. Object Class Recognition by Unsupervised Scale-Invariant Learning, by R. Fergus, P. Perona, and A. Zisserman, CVPR 2003.	slides fullpage (faces part 2, detection, boosting) slides fullpage (SVMs, unsupervised model learning)	Pset 4 files
11/20 11/27 11/29 12/4	Motion, optical flow, tracking	Trucco & Verri handout		**Pset 4 due 12/4**
12/6	Wrap-up			Graduate students' reviews and extensions due
12/13	*Final exam*			

Motion and tracking

Tracking objects, video analysis, low level motion





Tomas Izo

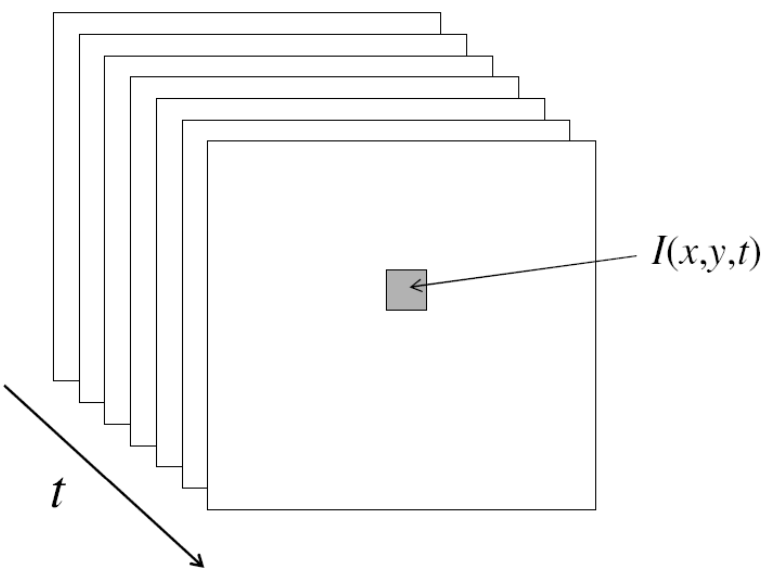
Outline

- Motion field and parallax
- Optical flow, brightness constancy
- Aperture problem
- Constraints on image motion

Uses of motion

- Analyzing motion can be useful for
  - Estimating 3d structure
  - Segmentation of moving objects
  - Tracking objects, features over time

Image sequences



A digital video is a sequence of images (frames) captured over time.

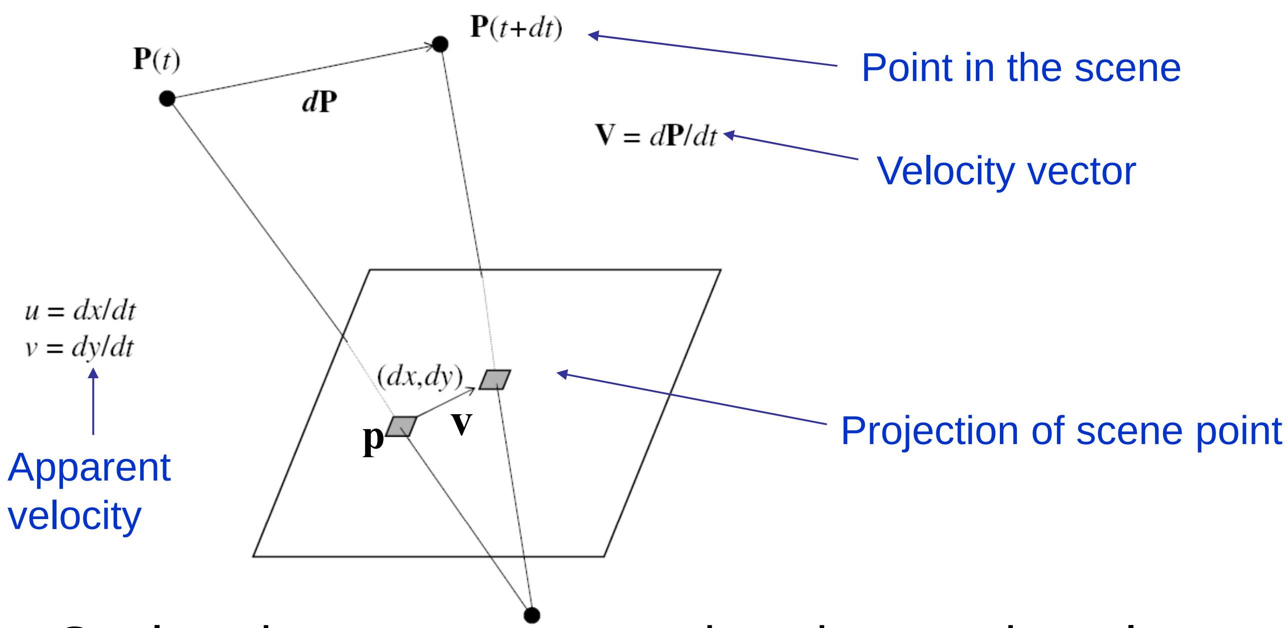
Now we consider image as a function of both *position* and *time*.

Figure by Martial Hebert, CMU

Types of motion in video

- Considering rigid objects – they can rotate and translate in the scene.
- Motion may be due to
  - Movement in scene
  - Movement of camera (ego motion)
- Geometrically equivalent, however illumination effects can make one appear different than the other.

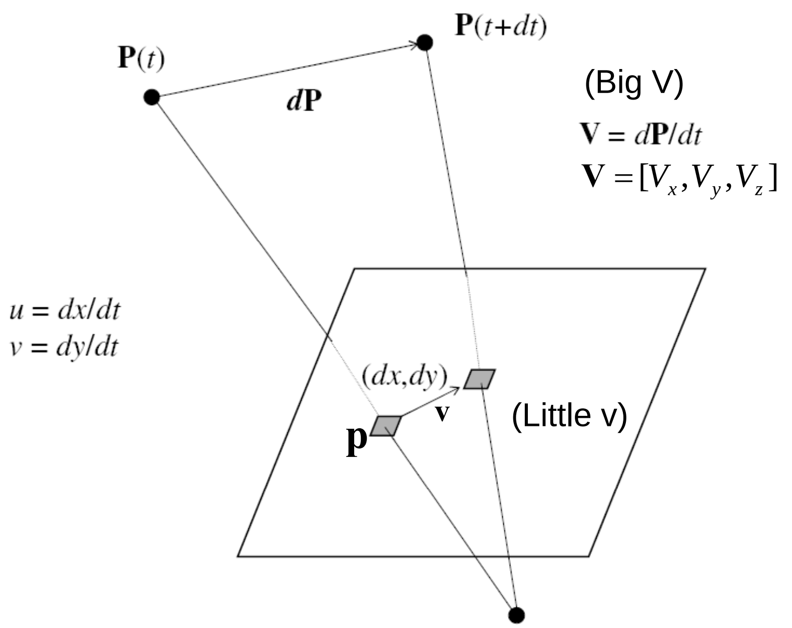
Motion field and apparent motion



Goal: estimate apparent motion, the u and v values at each pixel x,y, i.e., u(x,y), v(x,y)

Figure by Martial Hebert, CMU

Motion field equations



$$\mathbf{p} = f\,\frac{\mathbf{P}}{Z}$$

Take the time derivative of both sides:

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

Figure by Martial Hebert, CMU

Motion

$$\mathbf{V} = [V_x, V_y, V_z]$$
$$\boldsymbol{\omega} = \left[\omega_x, \omega_y, \omega_z\right]$$
$$\mathbf{P} = \left[X, Y, Z\right]$$

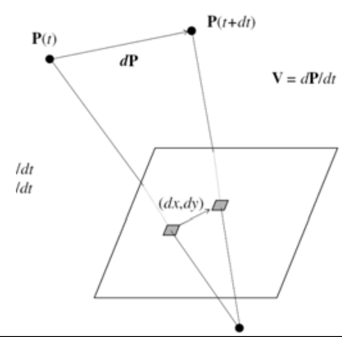
Translational motion

Angular velocity

Velocity of scene point described as

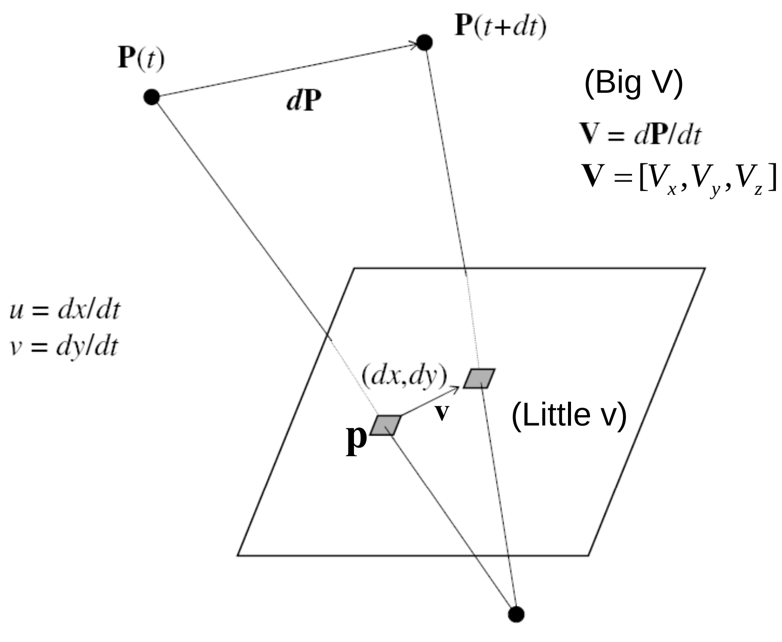
$$\mathbf{V} = -\mathbf{T} - \boldsymbol{\omega} \times \mathbf{P}$$

$$V_x = -T_x - \omega_y Z + \omega_z Y$$
$$V_y = -T_y - \omega_z X + \omega_x Z$$
$$V_z = -T_z - \omega_x Y + \omega_y X$$



Using this and the motion field equation, can give expressions for the components of the image velocity **v...**

Motion field equations



$$\mathbf{p} = f\,\frac{\mathbf{P}}{Z}$$

Take the time derivative of both sides:

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

Figure by Martial Hebert, CMU

Motion field equations

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

$$V_x = -T_x - \omega_y Z + \omega_z Y$$
$$V_y = -T_y - \omega_z X + \omega_x Z$$
$$V_z = -T_z - \omega_x Y + \omega_y X$$

$$v_x = \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}$$
$$v_y = \frac{T_z y - T_y f}{Z} - \omega_x f + \omega_z x + \frac{\omega_y xy}{f} - \frac{\omega_x y^2}{f}$$

Translational components | **Rotational components**

Trucco & Verri Section 8.2.1

Motion field equations

- Translational part of image motion depends on (unknown) depth of the point
- *Motion parallax*: image motion is a function of both motion in space and depth of each point.

$$v_x = \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}$$

$$v_y = \frac{T_z y - T_y f}{Z} - \omega_x f + \omega_z x + \frac{\omega_y xy}{f} - \frac{\omega_x y^2}{f}$$

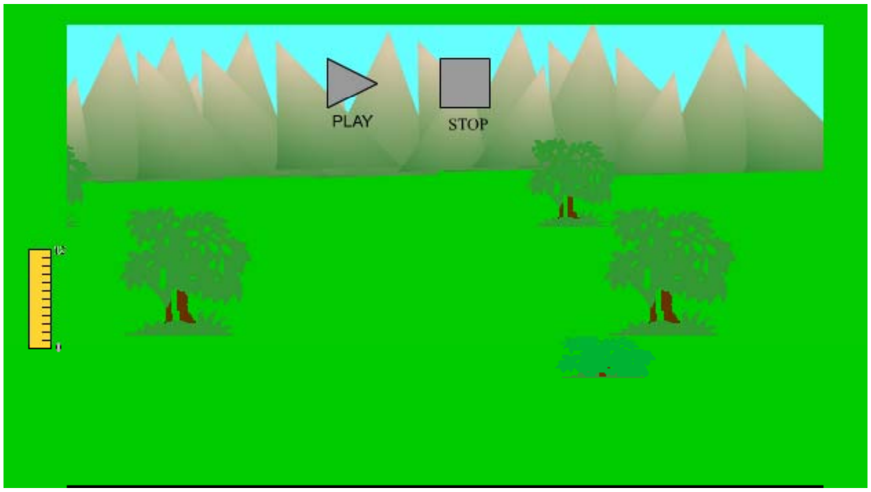
Translational components

Rotational components

Trucco & Verri Section 8.2.1

Motion parallax

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



Translational motion

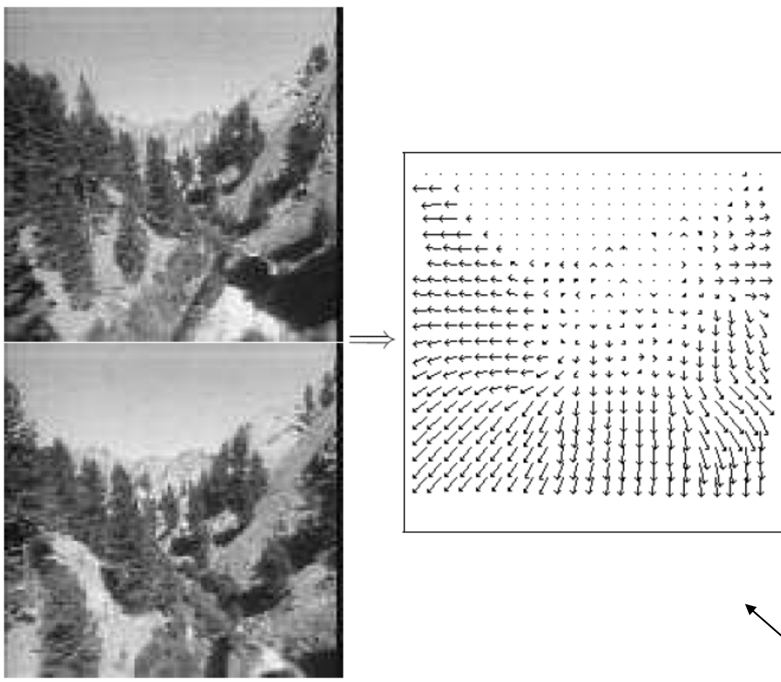


Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

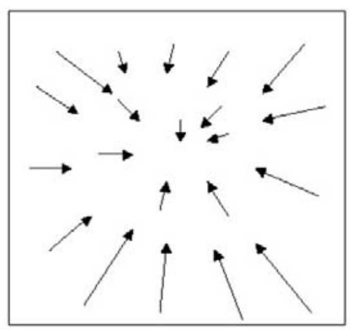
Radial motion field if T_z nonzero.

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

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

Figure from Michael Black, Ph.D. Thesis

Translational motion

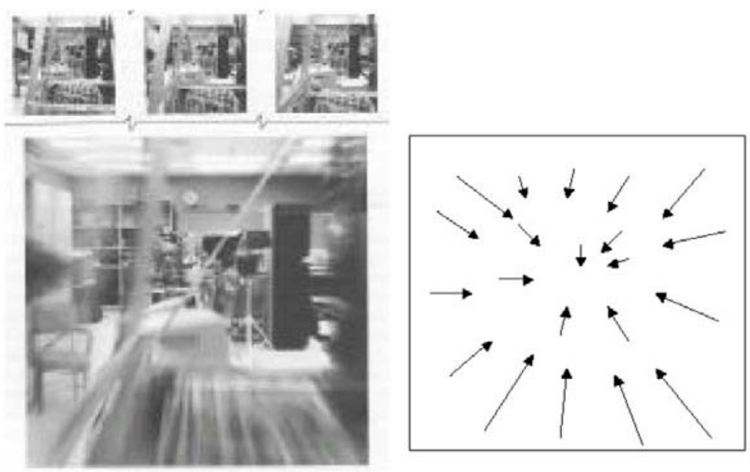


Radial motion field if T_z nonzero.

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

Figure from Michael Black, Ph.D. Thesis

Translational motion



Radial motion field if T_z nonzero.

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

Figure from Michael Black, Ph.D. Thesis

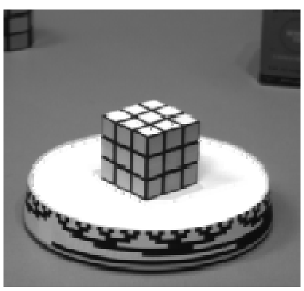
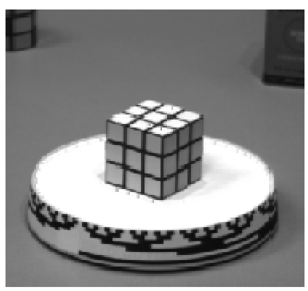
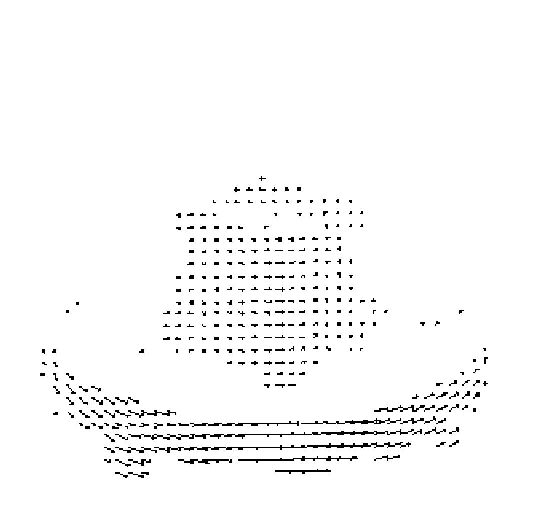
Motion vs. Stereo: Similarities

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

Motion vs. Stereo: Differences

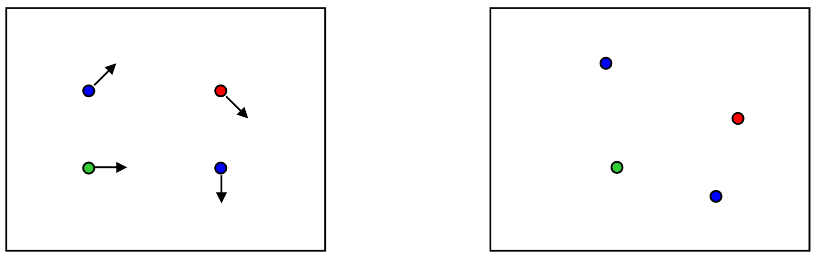
- Motion:
  - Uses velocity: consecutive frames must be close to get good approximate time derivative
  - 3d movement between camera and scene not necessarily single 3d rigid transformation
- Whereas with stereo:
  - Could have any disparity value
  - View pair separated by a single 3d transformation

Optical flow problem

Goal: estimate apparent motion, the u and v values at each pixel x,y, i.e., u(x,y), v(x,y)

Optical flow problem



$H(x, y)$ $I(x, y)$

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

Adapted from Steve Seitz, UW

- What might make it difficult to estimate apparent motion?

Brightness constancy

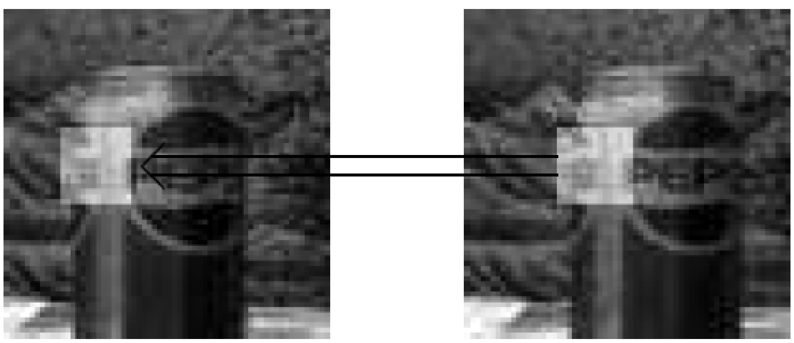


Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.

Figure by Michael Black

Spatial coherence

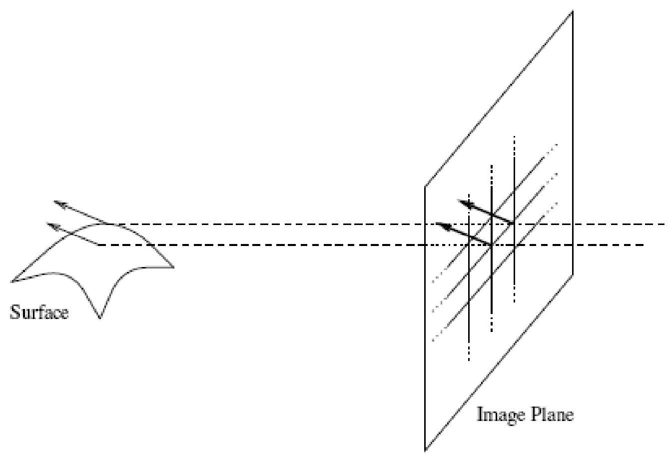


Figure 1.7: Spatial coherence assumption. Neighboring points in the image are assumed to belong to the same surface in the scene.

Figure by Michael Black

Temporal smoothness

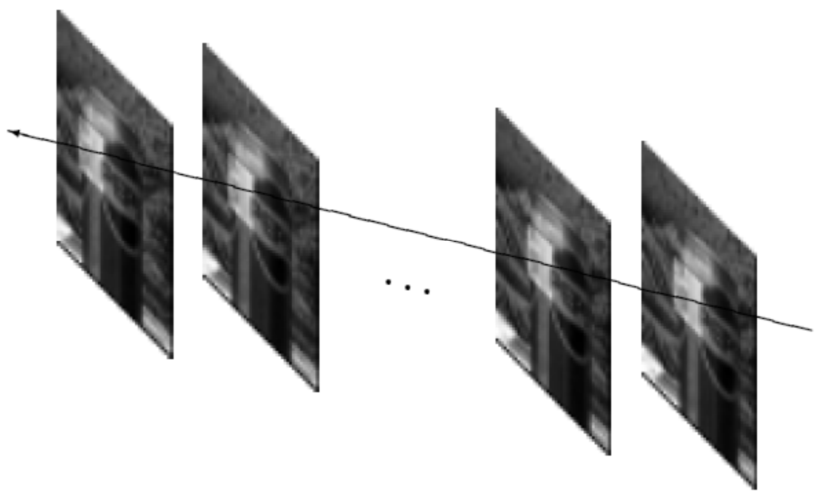


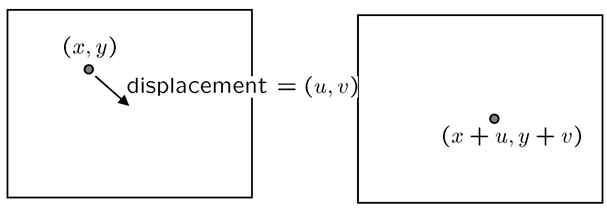
Figure 1.8: Temporal continuity assumption. A patch in the image is assumed to have the same motion (constant velocity, or acceleration) over time.

Figure by Michael Black

Motion constraints

- To recover optical flow, we need some constraints (assumptions)
  - *Brightness constancy*: in spite of motion, image measurement in small region will remain the same
  - *Spatial coherence*: assume nearby points belong to the same surface, thus have similar motions, so estimated motion should vary smoothly.
  - *Temporal smoothness*: motion of a surface patch changes gradually over time.

Brightness constancy equation



$$
\begin{aligned}
I(x, y, t) &= I(x + \delta x, y + \delta y, t + \delta t) \\
&= I(x + u\delta t, y + v\delta t, t + \delta t)
\end{aligned}
$$

$$\frac{dI}{dt} = 0$$

Total derivative: *x* and *y* are also functions of time *t*

$$= \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}$$

spatial image gradients

temporal derivatives, *u* and *v*

Brightness constancy equation

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

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

u *v*

Rewritten as: $$I_x u + I_y v + I_t = 0.$$

$$\nabla I^T \mathbf{u} + I_t = 0.$$

This is exactly true in the limit as *u* and *v* go to 0, for infinitesimal motions.

Brightness constancy equation

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

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

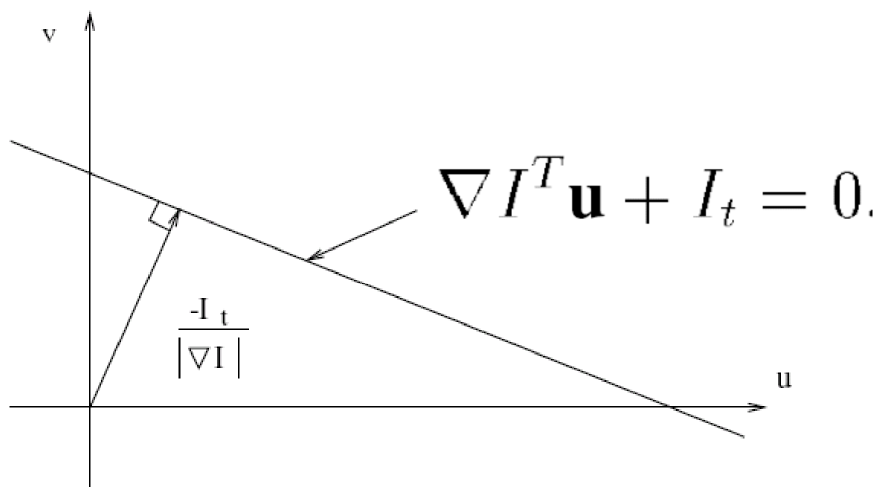
u *v*

Rewritten as: $$I_x u + I_y v + I_t = 0.$$

$$\nabla I^T \mathbf{u} + I_t = 0.$$

Which terms are measurable from images?
How many unknowns in this equation?

Aperture problem



According to brightness constancy constraint, motions that satisfy the optical flow equation are only constrained to lie along a line in *u,v* space.

Figure from Michael Black's Ph.D. Thesis

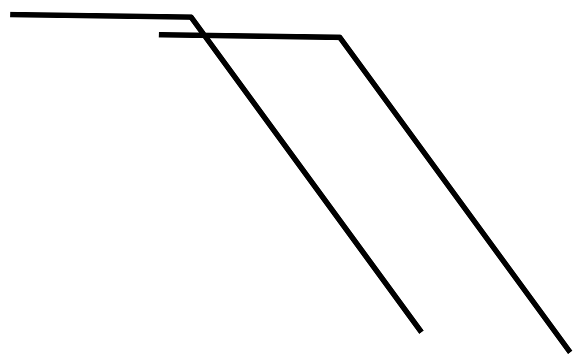
Aperture problem

$$\nabla I^T \mathbf{u} + I_t = 0.$$

- Brightness constancy equation: single equation, two unknowns; infinitely many solutions.

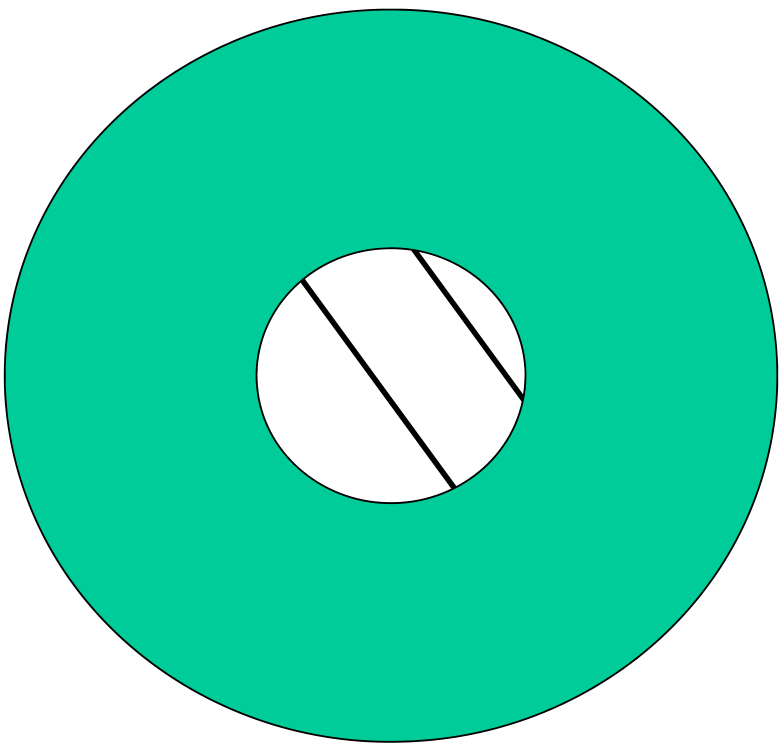
- Can only compute projection of actual flow vector [*u*,*v*] in the direction of the image gradient, that is, in the direction *normal* to the image edge.
  - Flow component in gradient direction determined
  - Flow component parallel to edge unknown.

Aperture problem



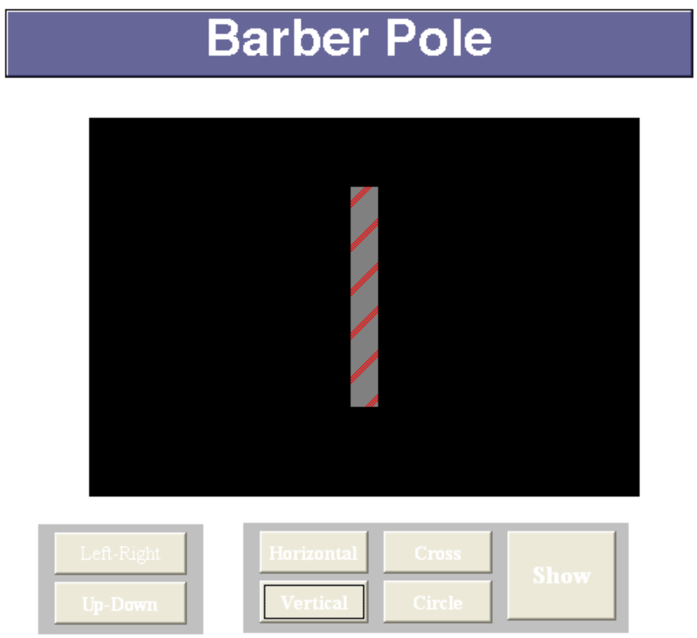
Slide by Steve Seitz, UW

Aperture problem



Slide by Steve Seitz, UW

Aperture problem



- http://www.psychologie.tu-dresden.de/i1/kaw/diverses%20Material/www.illusionworks.com/html/barber_pole.html

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!

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

$$\underset{25\times 2}{\underset{A}{\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix}}} \underset{2\times 1}{\underset{d}{\begin{bmatrix} u \\ v \end{bmatrix}}} = - \underset{25\times 1}{\underset{b}{\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}}}$$

Slide by Steve Seitz, UW

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!

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

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

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

Slide by Steve Seitz, UW

Lucas-Kanade flow

Prob: we have more equations than unknowns

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

Solution: solve least squares problem

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

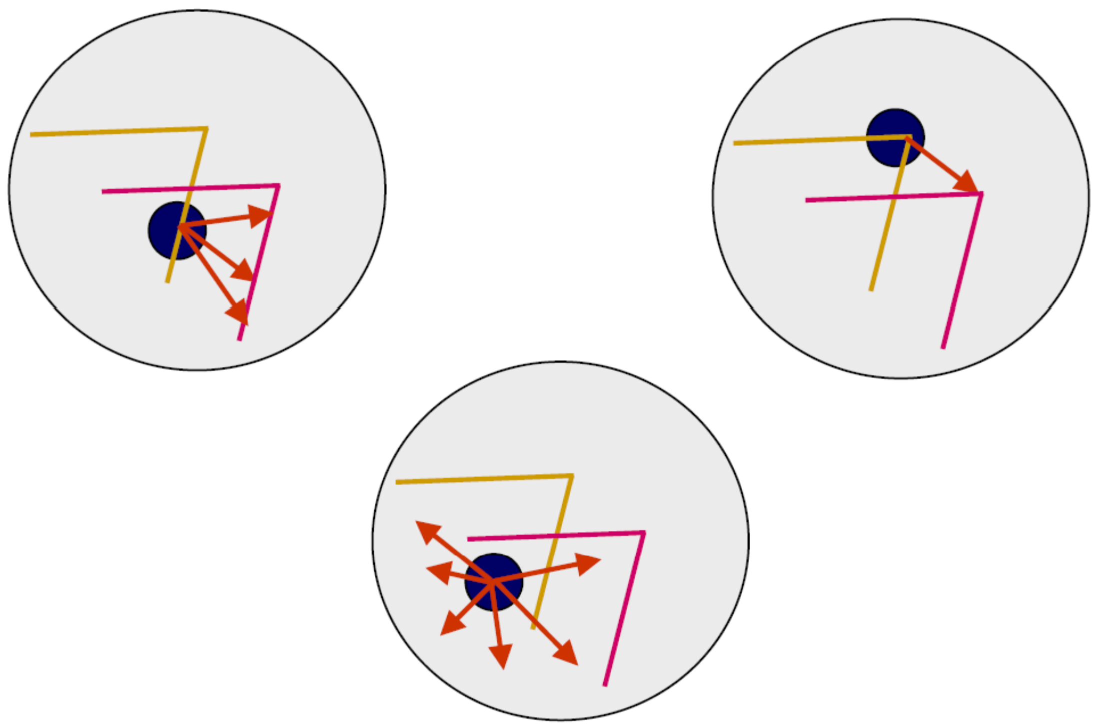
$$\underset{2\times 2}{(A^T A)}\ \underset{2\times 1}{d} = \underset{2\times 1}{A^T b}$$

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

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

Slide by Steve Seitz, UW

Windows and apparent motion



Slide from Trevor Darrell, MIT

Conditions for solvability

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

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

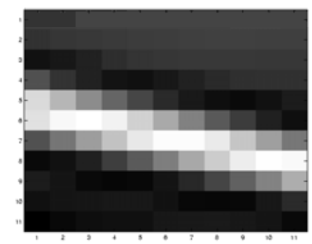
$$A^T A \qquad\qquad\qquad\qquad A^T b$$

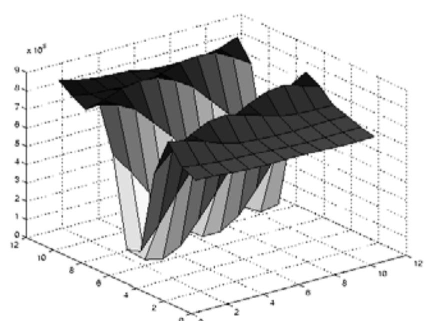
When is this solvable?

- **$A^T A$** 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



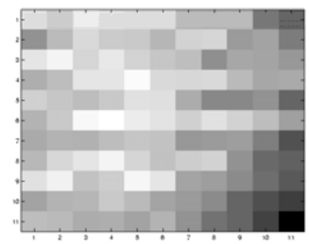


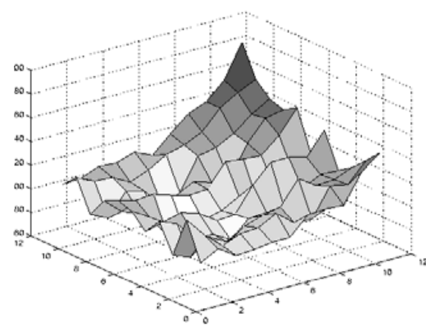
–gradient strong in one direction

–large λ_1, small λ_2

Adapted from Steve Seitz, UW

Low texture region

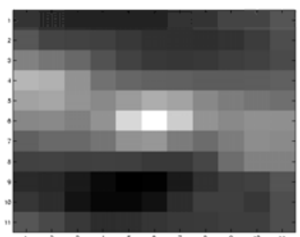


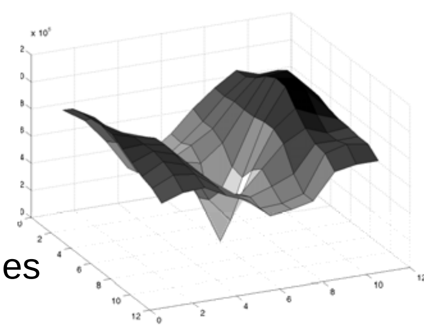


– gradients have small magnitude
– small λ_1, small λ_2

Slide by Steve Seitz, UW

High textured region





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

Slide by Steve Seitz, UW

Good conditions for solving flow

- Recall Harris corner detection
- Good feature windows to track in time can be detected independently in a single frame.

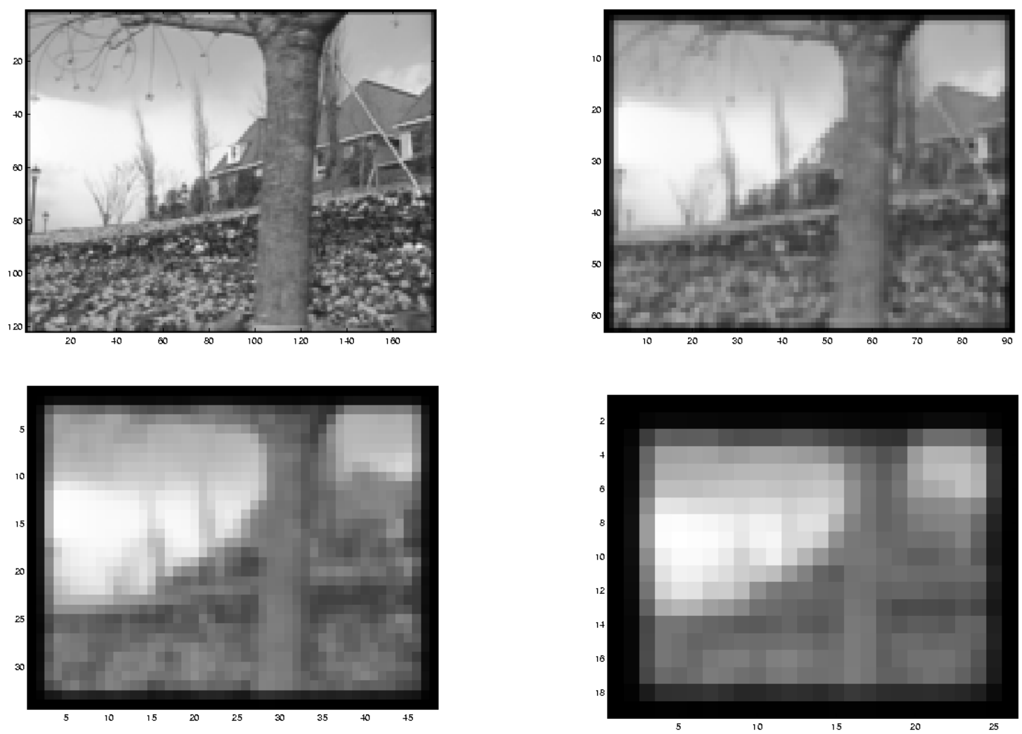
Revisiting the small motion assumption

Is this motion small enough?

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

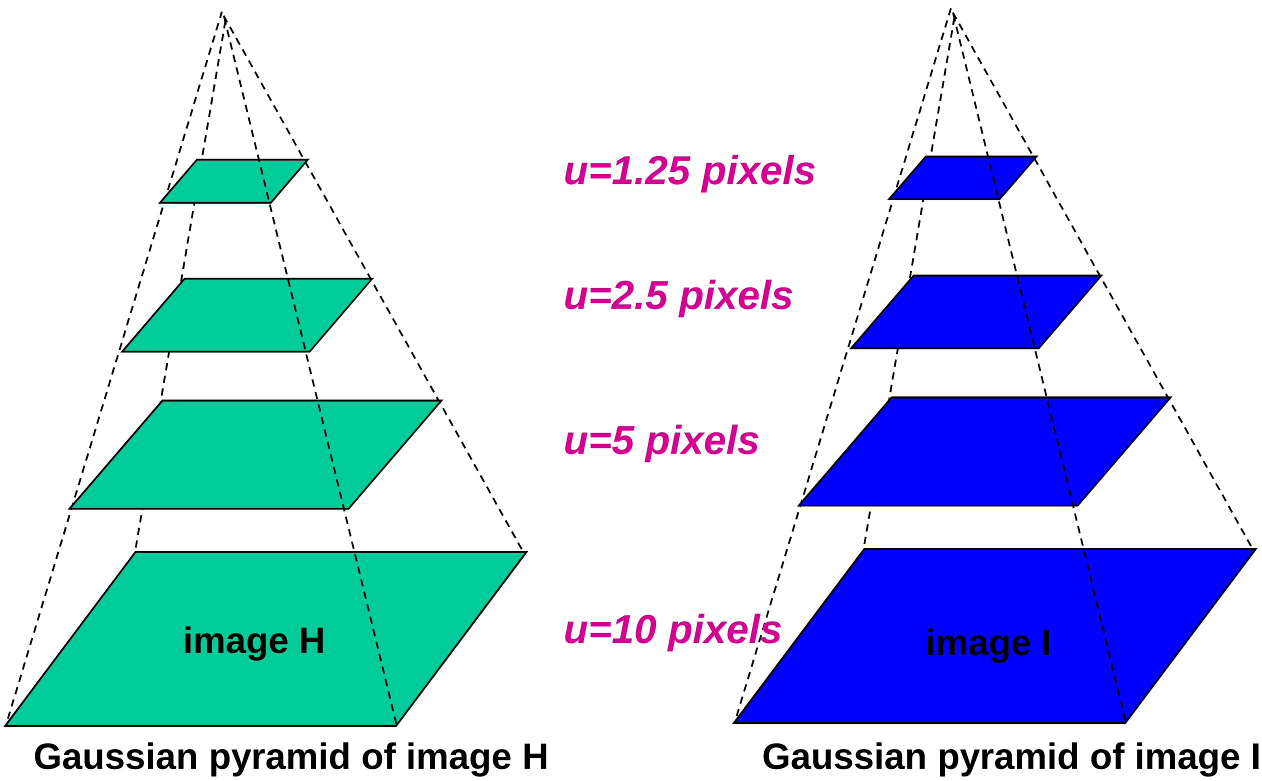
Slide by Steve Seitz, UW

Reduce the resolution!

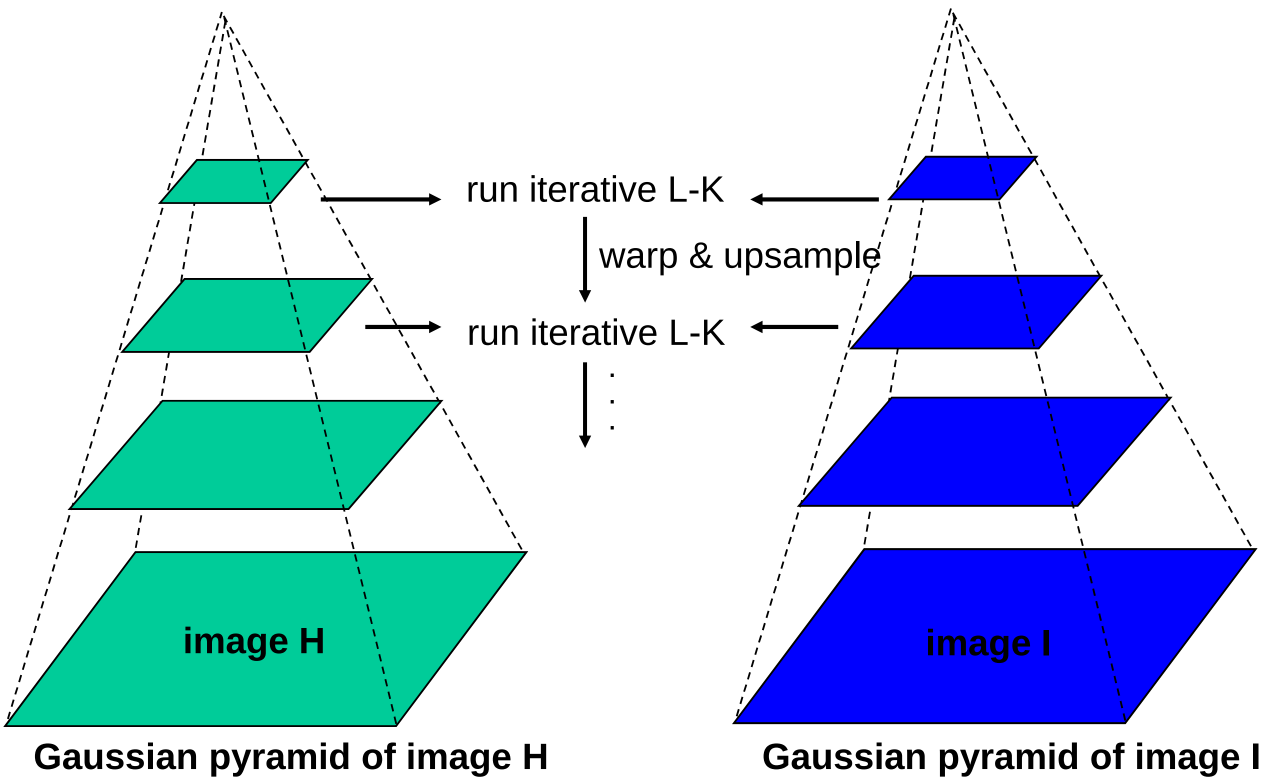


Slide by Steve Seitz, UW

Coarse-to-fine optical flow estimation



Coarse-to-fine optical flow estimation



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

Coming up

- Problem set 4 due 12/4

More on motion

- Multiple motions and segmentation
- Tracking
- SfM