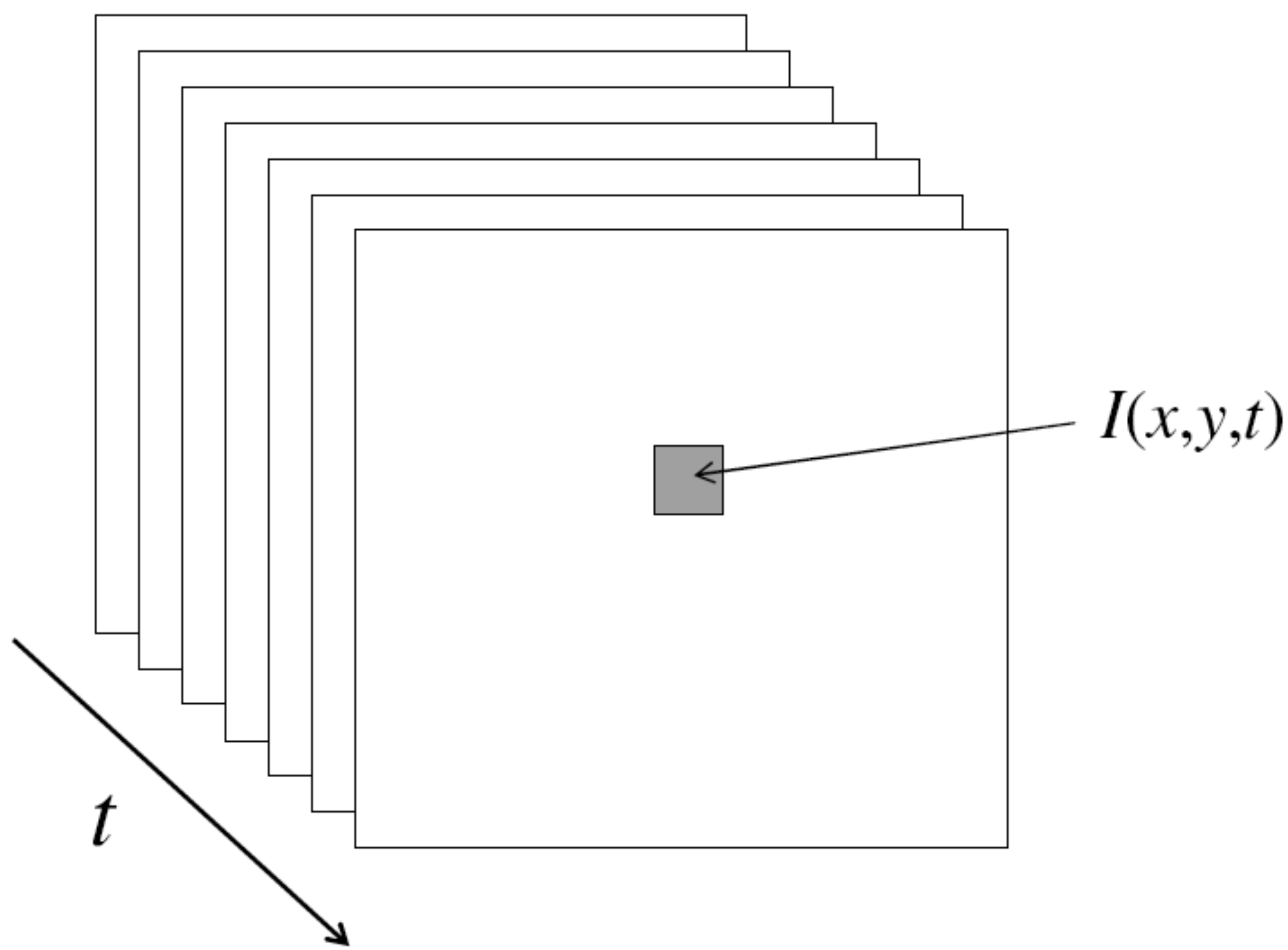
Video + Motion

Last Lecture of CV Course

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 taken from Kristen Grauman

Vision algorithm performance over time

Algorithm Complexity

High Quality 3d shape extraction

2000

Immersive Telepresence

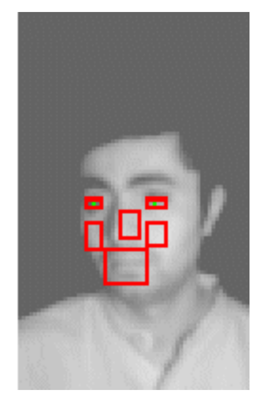
Video registration to 3D site models

1998

Geo-registration visual databases

Coarse 3D Depth Recovery

1995



Face Finding for Iris Recognition

2D Video Insertion

1993

Real-time insertion in Live TV

2D Stabilization

1990

Mosaicing for entertainment & surveillance

Time

A FRAMEWORK FOR VIDEO PROCESSING

ALIGN

2D & 3D MODELS OF MOTION & STRUCTURE

MODEL-BASED IMAGE SEQUENCE ALIGNMENT

TEST

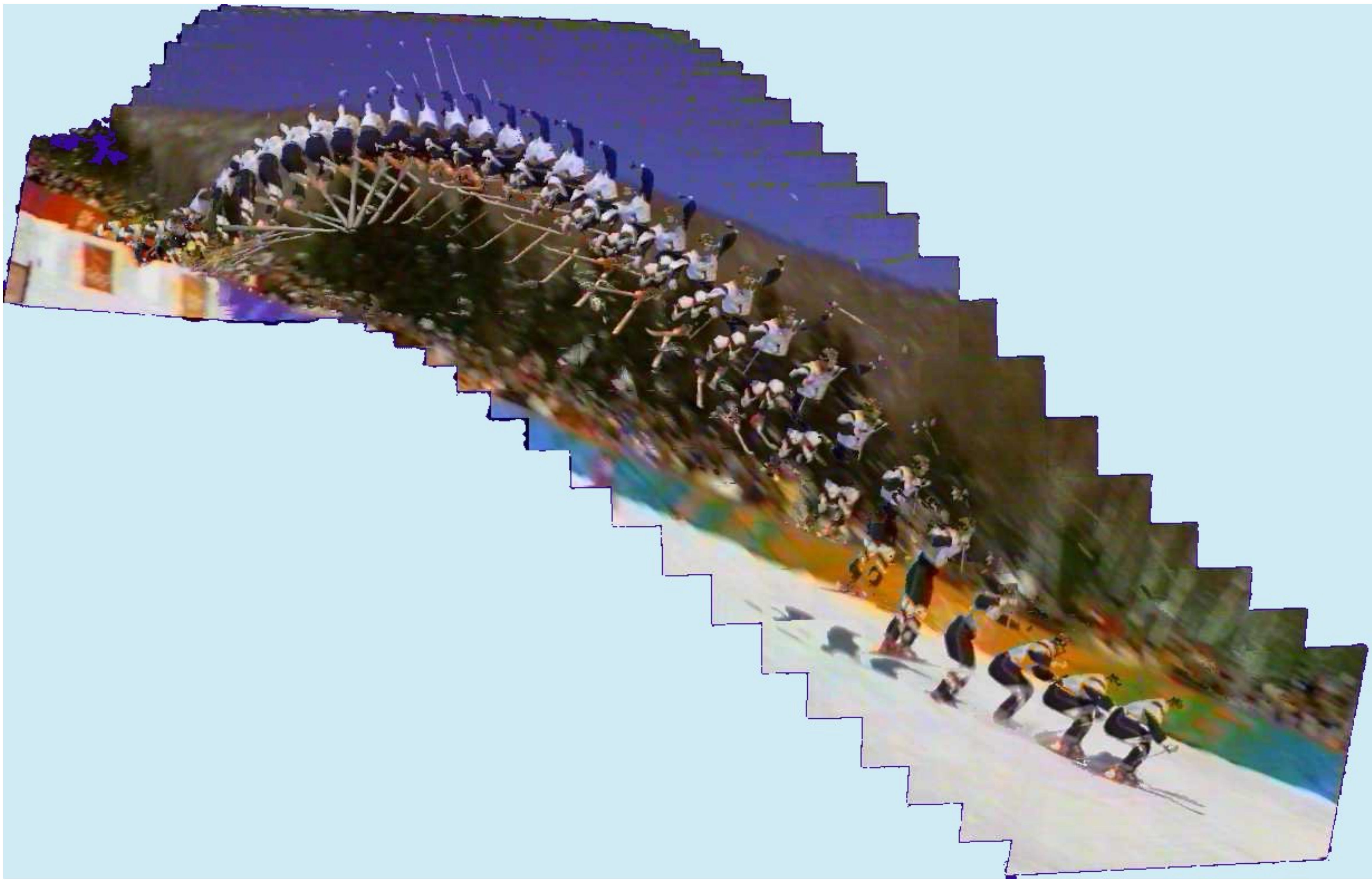
WARP/RENDER WITH 2D/3D MODELS

TEST ALIGNMENT QUALITY

SYNTHESIZE

CREATE OUTPUT REPRESENTATIONS

SYNOPISIS MOSAICS



Motion: Why is it useful?

Derek Hoiem

Motion: Why is it useful?

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

G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

Derek Hoiem

Tracking: some applications

Body pose tracking, activity recognition

Censusing a bat population

Video-based interfaces

Medical apps

Surveillance

Kristen Grauman

Detection vs. tracking


t=1


t=2

…


t=20


t=21

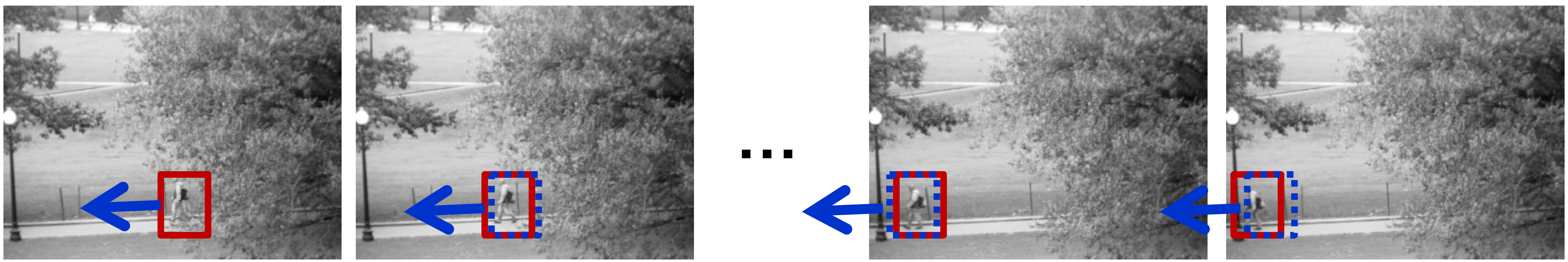
Kristen Grauman

Detection vs. tracking

Detection: We detect the object independently in each frame and can record its position over time, e.g., based on detection window coordinates

Adapted from Kristen Grauman

Detection vs. tracking

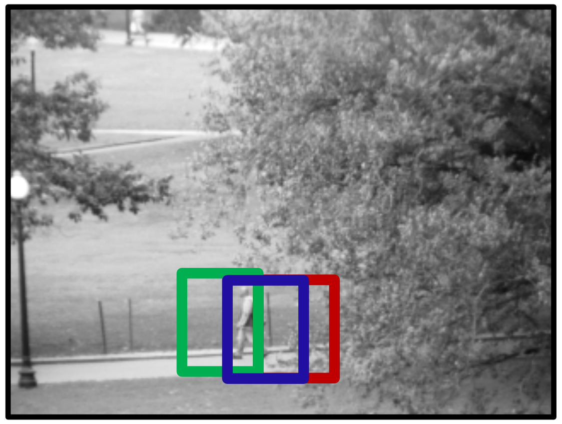


Tracking with *dynamics*: We use image measurements to estimate position of object, but also incorporate position predicted by dynamics, i.e., our expectation of the object's motion pattern

Kristen Grauman

Tracking: prediction + correction

Time t



Time t+1

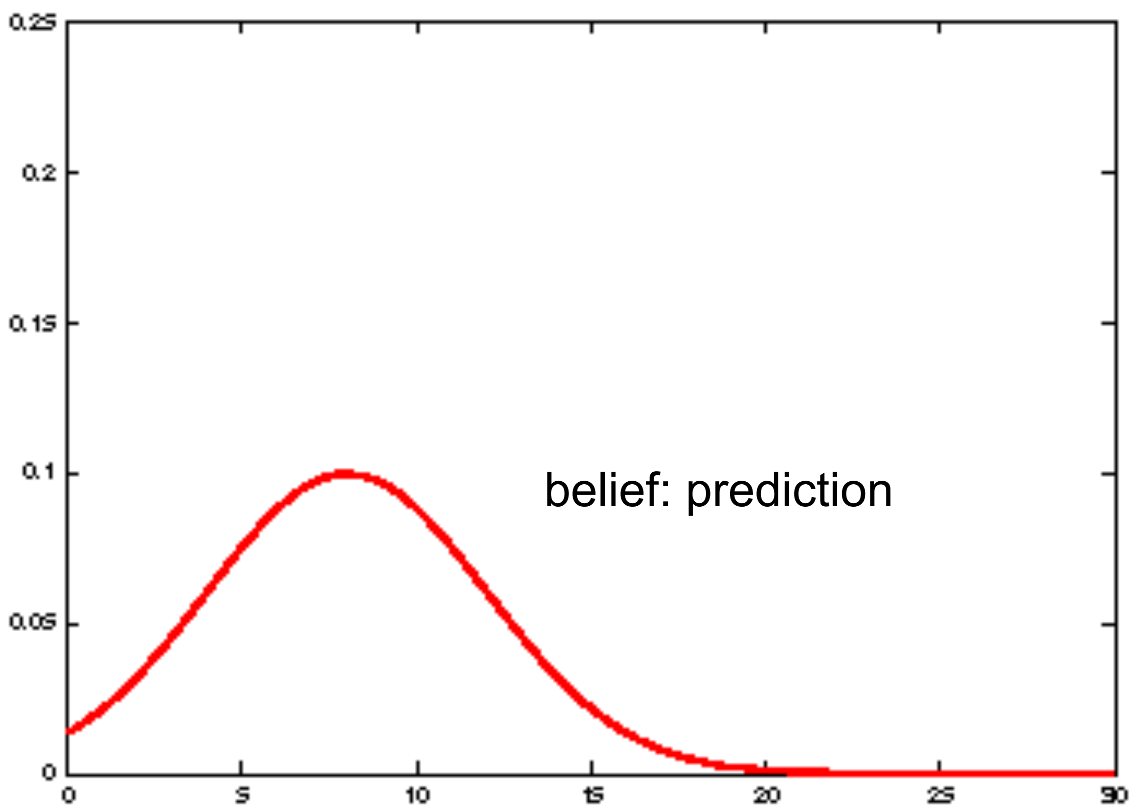
Belief

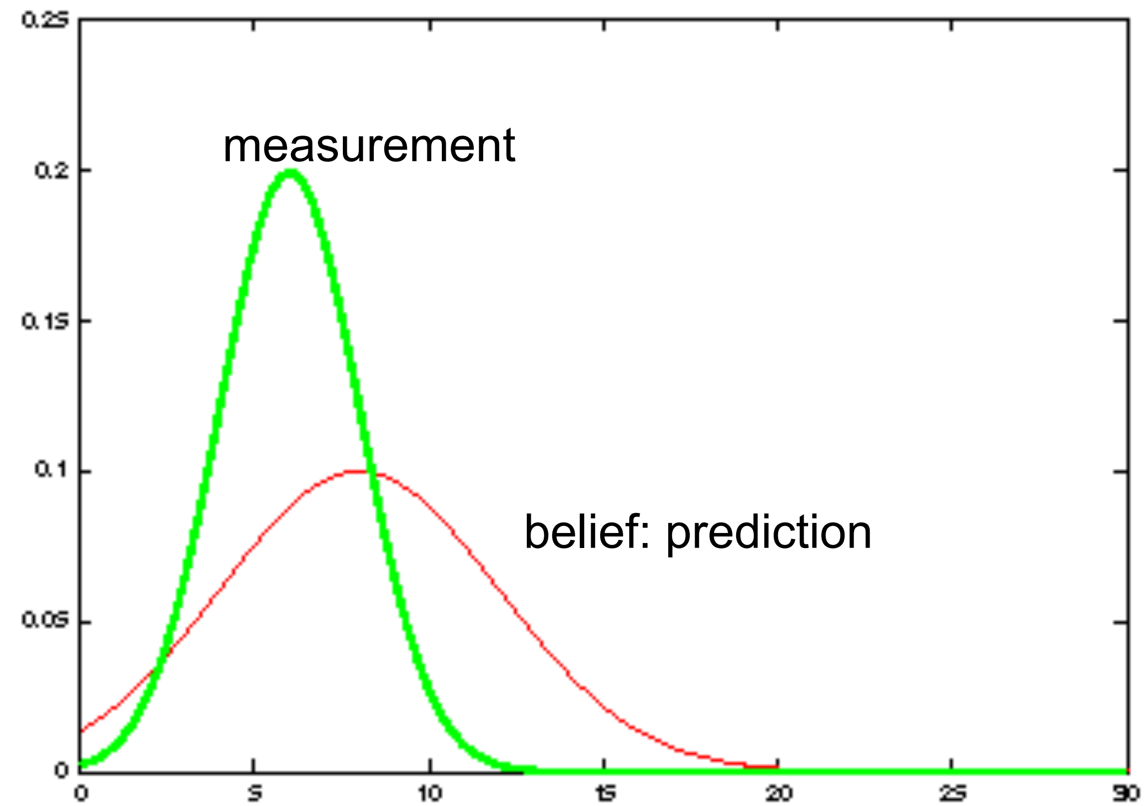
Measurement

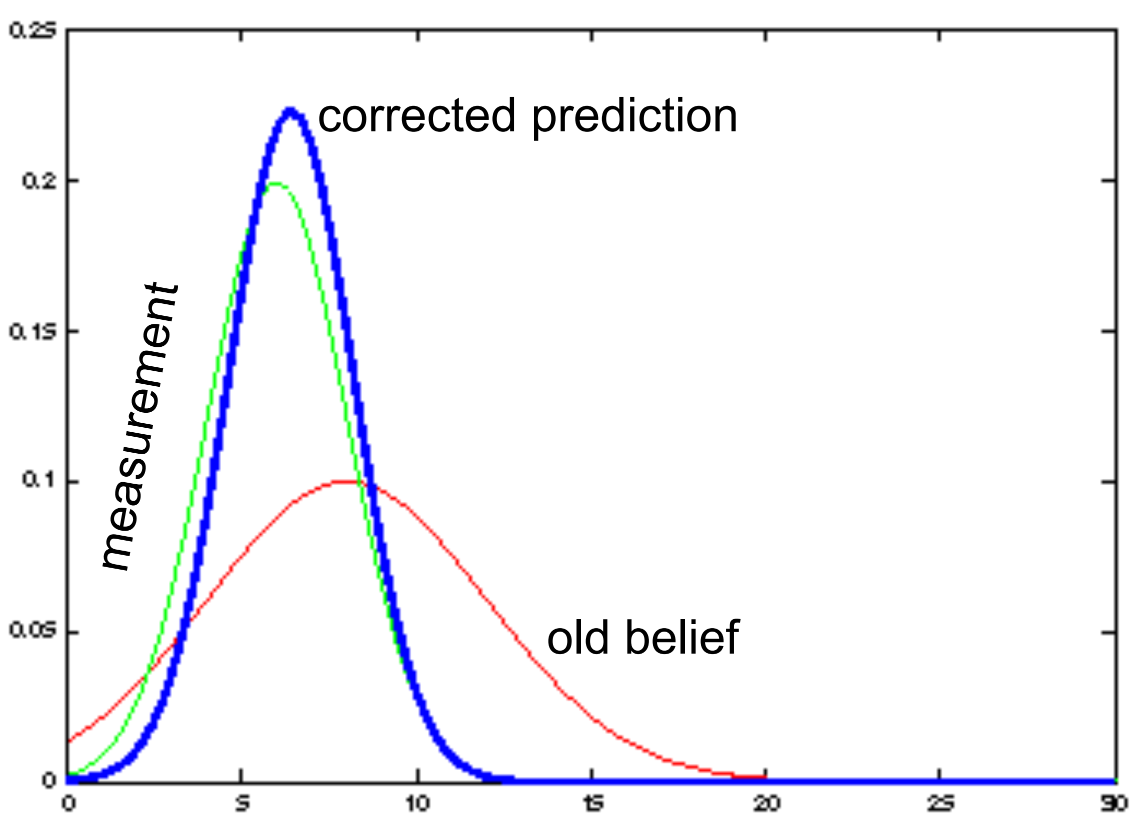
Corrected prediction

Kristen Grauman

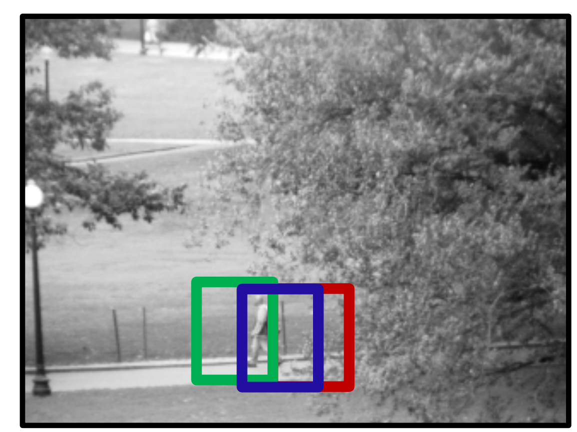
Tracking: prediction + correction







Time t



Time t+1

Kristen Grauman

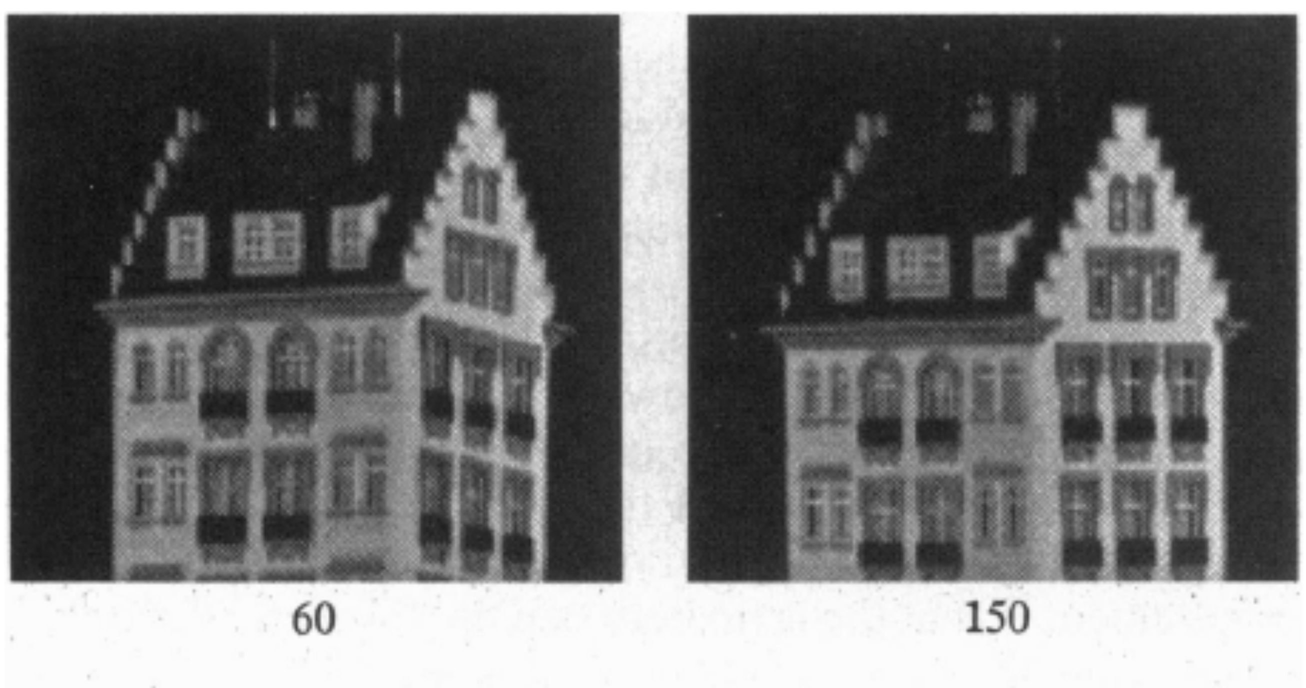
Recovering motion

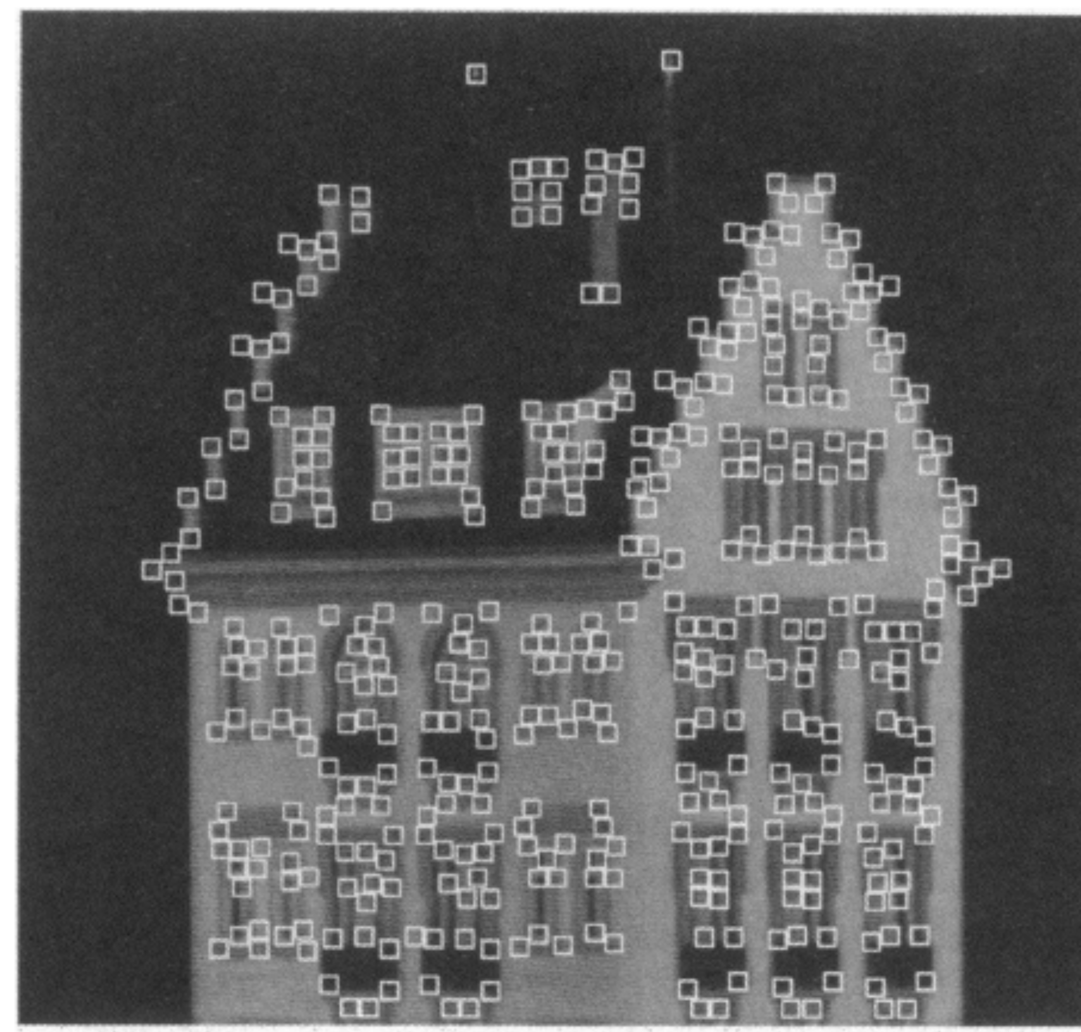
- Feature-tracking
  - Extract visual features (corners, textured areas) and “track” them over multiple frames
- Optical flow
  - Recover image motion at each pixel from spatio-temporal image brightness variations (optical flow)

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

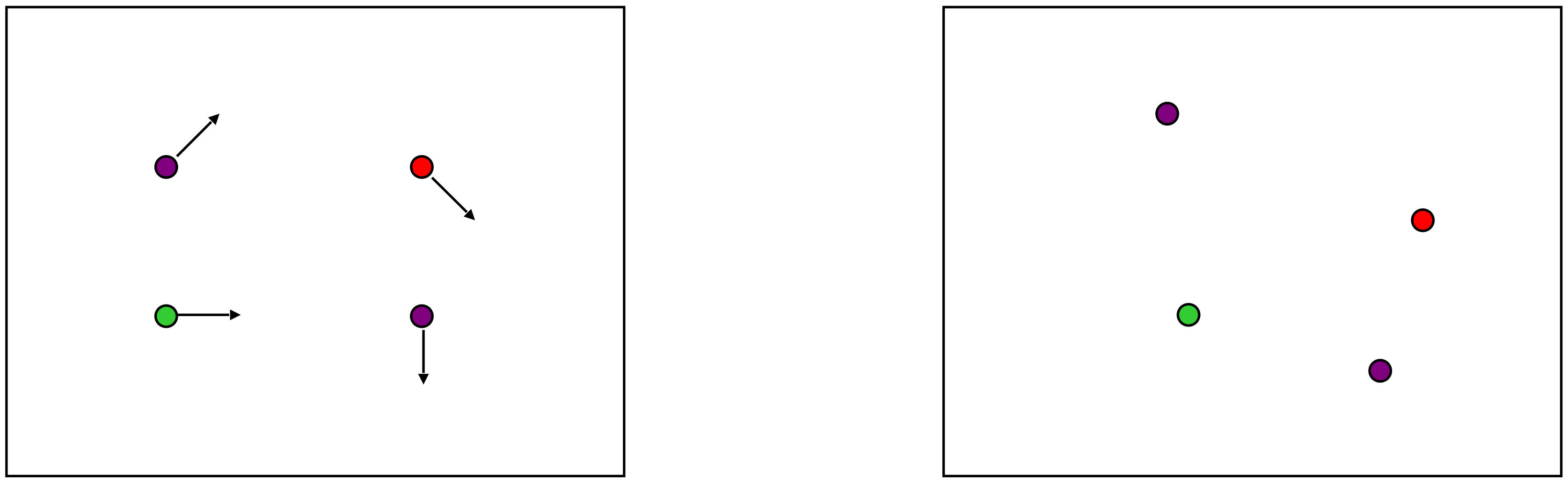
Feature tracking

- Many problems, such as structure from motion require matching points
- If motion is small, tracking is an easy way to get them





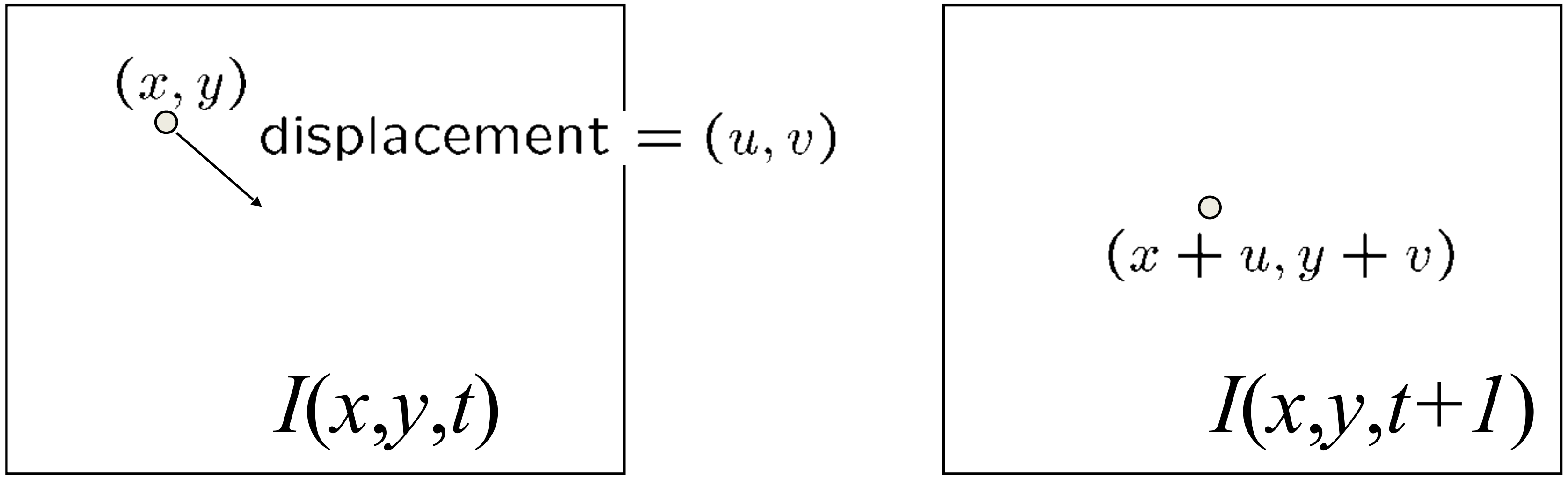
Feature tracking



$I(x,y,t)$ $\qquad$ $I(x,y,t+1)$

- Given two subsequent frames, estimate the point translation
- Key assumptions of Lucas-Kanade Tracker
  - **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

Feature tracking



- Brightness Constancy Equation:

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

Take Taylor expansion of $I(x+u, y+v, t+1)$ at (x,y,t) to linearize the right side:

Image derivative along x　　Difference over frames

$$I(x + u, y + v, t + 1) \approx I(x, y, t) + \boxed{I_x} \cdot u + I_y \cdot v + \boxed{I_t}$$

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

So: $$I_x \cdot u + I_y \cdot v + I_t \approx 0 \quad \rightarrow \nabla I \cdot \begin{bmatrix} u & v \end{bmatrix}^T + I_t = 0$$

Feature tracking

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

How many variables and equations?

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

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

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

Feature tracking

- Least squares problem:

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

Matching patches across images

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \qquad \underset{25\text{x}2}{A} \; \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Least squares solution for *d* given by $(A^T A)\ d = A^T b$

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

The summations are over all pixels in the K x K window

Conditions for solvability

Optimal (u, v) satisfies Lucas-Kanade equation

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

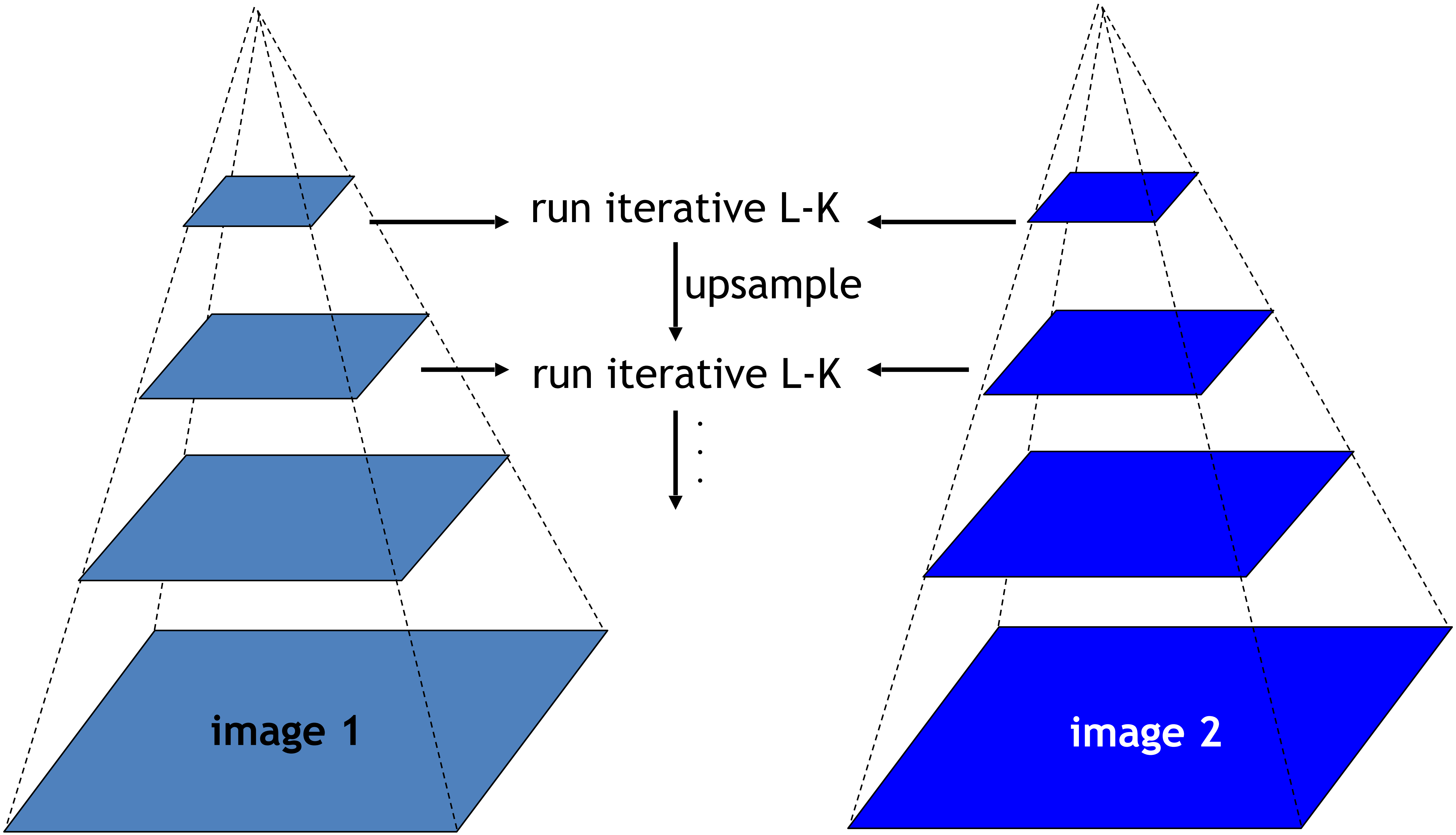
When is this solvable? I.e., what are good points to track?

- **$A^T A$** should be invertible
- **$A^T A$** should not be too small due to noise
  - 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)

Does this remind you of anything?

Criteria for Harris corner detector

Dealing with larger movements: coarse-to-fine registration



Gaussian pyramid of image 1 (t)

Gaussian pyramid of image 2 (t+1)

There are several other algorithms that you should know

- Optical flow (Lucas Kanade)
- Other advanced versions of motion and tracking

Let us look back

Lecture:1: Introduction to Computer Vision

Lecture:2: Basics of Image Processing

Lecture:3-4: Affine Transformations

Lecture:5-6: Image Filtering

Lecture 7: Learning to code with opencv+python

Lecture 8: Discrete Wavelet Transform [+PyWavelets]

Lecture 9: Camera Calibration

Lecture 10: Corner Detection

Lecture 11: Line+boundary detection (Hough) + Calibration (cont.)

Lecture 12: Camera Calibration (Cont.)

Lecture 13: Image Registration

Lecture 14: Segmentation

Lecture 15-20: Segmentation + Recognition

Lecture 21-23: Stereo + Mosaic

Lecture 24: Video+Motion

About End-Sem

- Two parts
  - Closed book (24th)- 1 hr
  - Open book/resources (on 24th)- 3 hrs + 1hr (bonus) - you can use any resource but should not discuss with any person (friend, senior, batch-mate, TAs, or professor)
  - Syllabus - entire course (but more emphasis on post exam-3)