Tracking and Motion

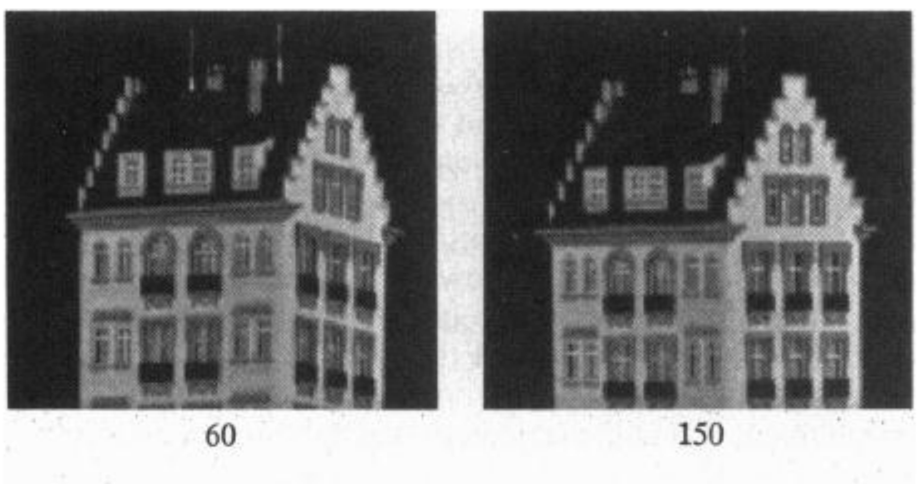
Book: Forsyth 11.

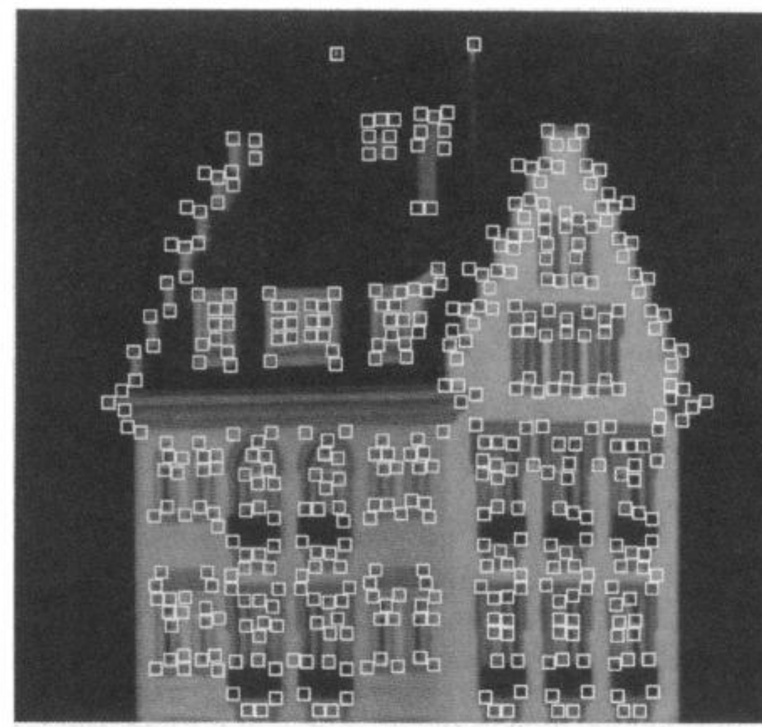
Overview

- 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.
- Single Object tracking
- Multi-object tracking
  - Track multiple possibly similarly looking objects.

Feature tracking

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

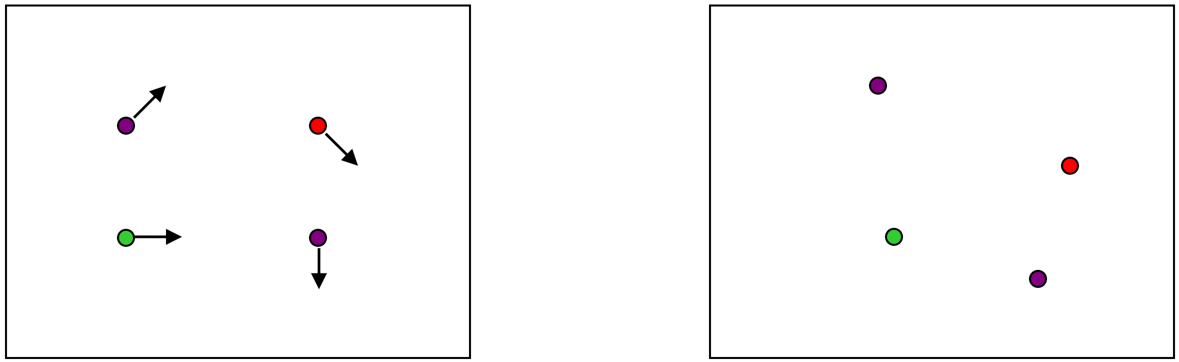




Feature tracking

- Challenges
  - Figure out which features can be tracked
  - Efficiently track across frames
  - Some points may change appearance over time (e.g., due to rotation, moving into shadows, etc.)
  - Drift: small errors can accumulate as appearance model is updated
  - Points may appear or disappear: need to be able to add/delete tracked points

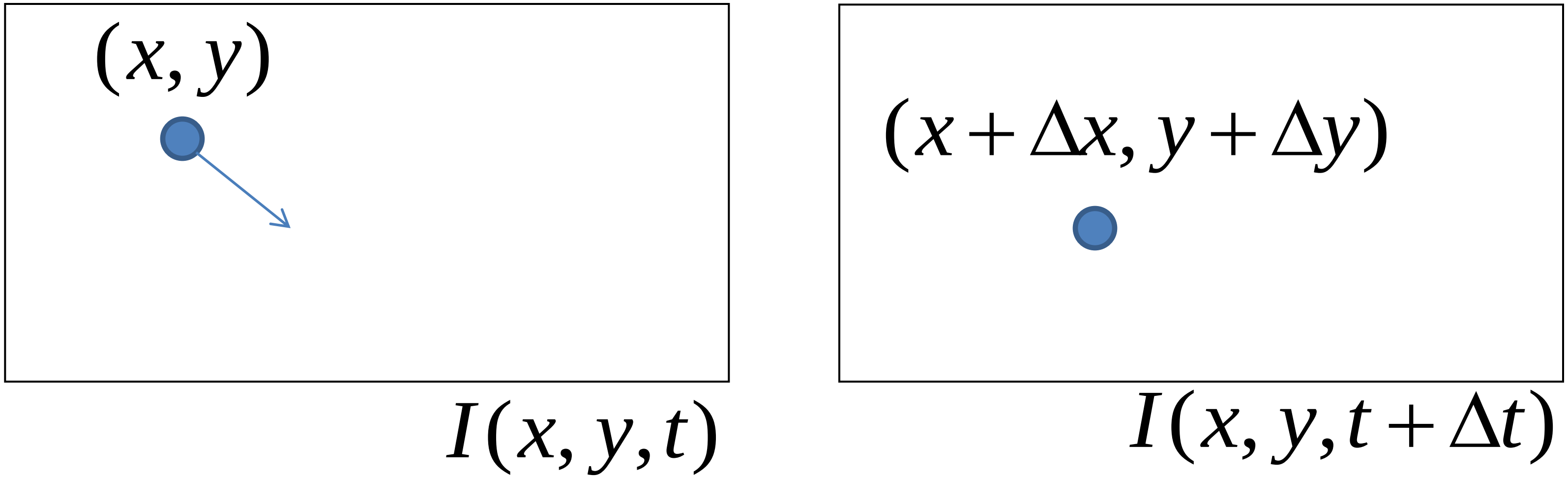
Feature tracking: KLT Algorithm



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

- Given two subsequent frames, estimate the point translation
- Assumptions of Lucas-Kanade Tracker (KLT)
  - **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

The brightness constancy constraint



- Brightness Constancy Equation:

$$I(x, y, t) = I(x+\Delta x, y+\Delta y, t+\Delta t)$$

- Taylor expansion:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t) + I_x \Delta x + I_y \Delta y + I_t \Delta t$$

The brightness constancy constraint

- Brightness Constancy Equation:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

- Taylor expansion:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + I_x \Delta x + I_y \Delta y + I_t \Delta t$$

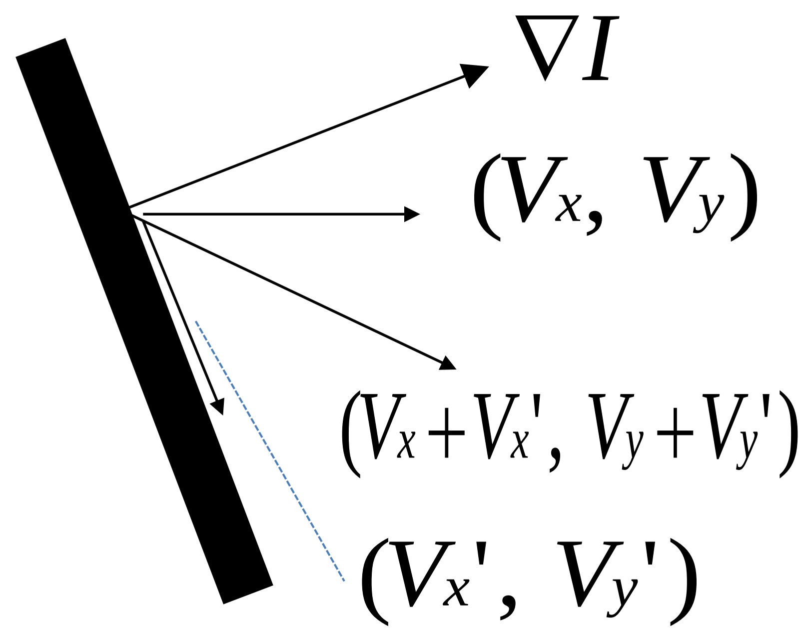
$$I_x \Delta x + I_y \Delta y + I_t \Delta t = 0$$

$$I_x V_x + I_y V_y + I_t = 0$$

$$\nabla I^T \cdot \vec{V} + I_t = 0$$

The brightness constancy constraint

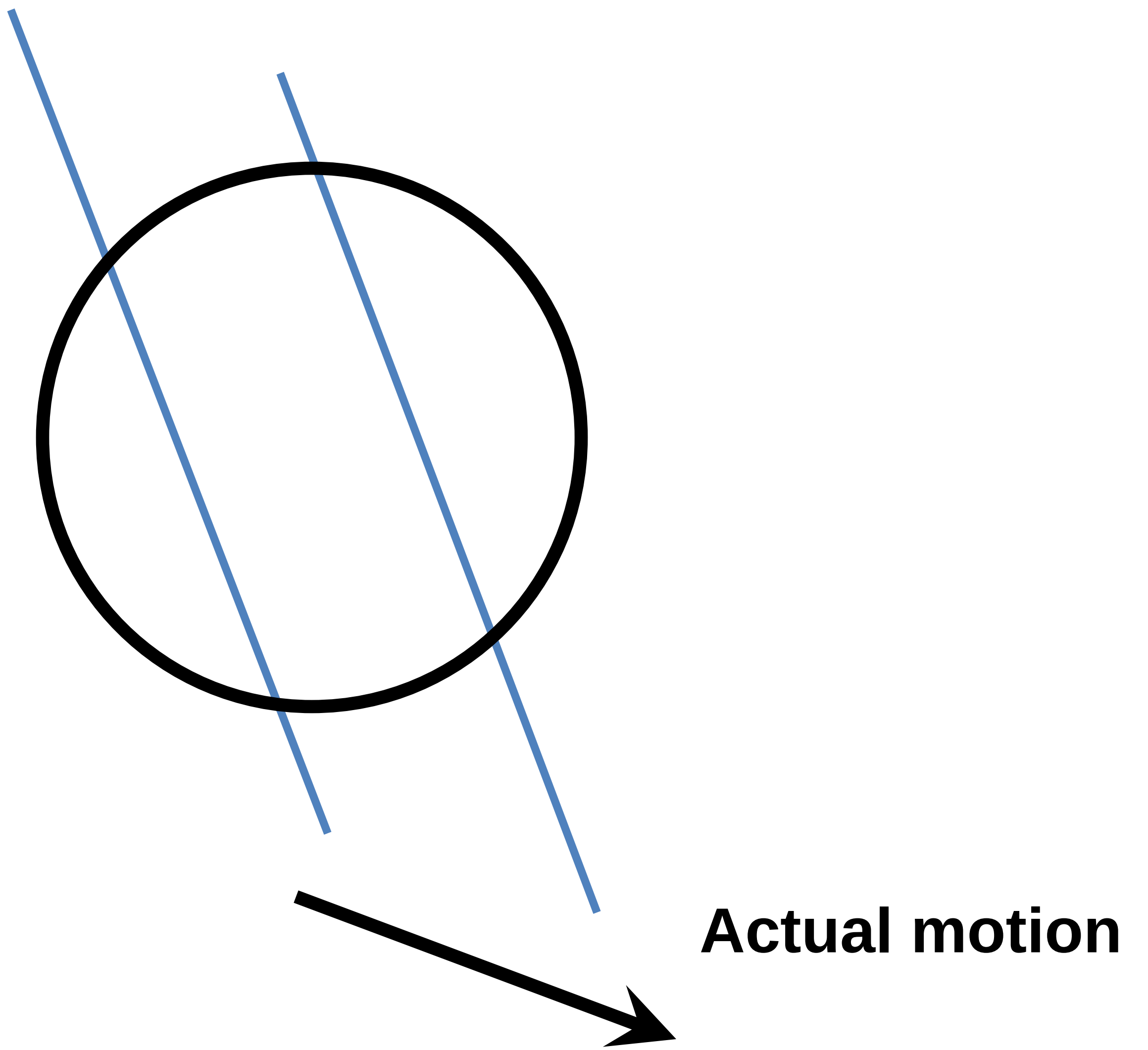
- Can we use this equation to recover image motion (V_x,V_y) at each pixel?
- NO. One equation, two unknowns (V_x,V_y).
- The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured.



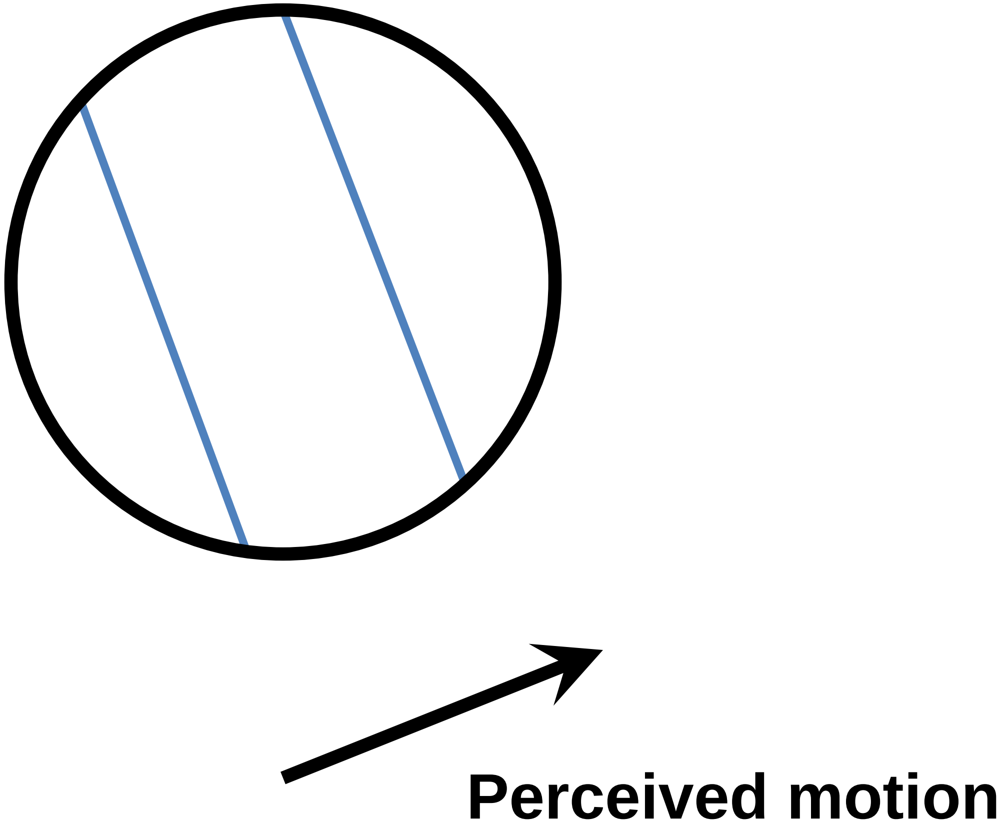
If (V_x,V_y) satisfies the equation, so does $(V_x + V_x', V_y + V_y')$ if:

$$\nabla I^T \cdot \left[V_x' \ V_y'\right] = 0$$

The aperture problem



The aperture problem



Solving the ambiguity

- **Spatial coherence constraint**
- Assume the pixel's neighbors have the same velocity (so a 5x5 window, that gives us 25 equations per pixel).

$$\nabla I^T \cdot [V_x \ V_y] = -I_t$$

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \dots & \dots \\ I_x(p_N) & I_y(p_N) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_1) \\ \dots \\ I_t(p_N) \end{bmatrix}$$

Matching patches across images

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \cdots & \cdots \\ I_x(p_N) & I_y(p_N) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_1) \\ \cdots \\ I_t(p_N) \end{bmatrix}$$

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

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

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

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

Low-texture region

– gradients have small magnitude

– small λ_1, small λ_2

Edge

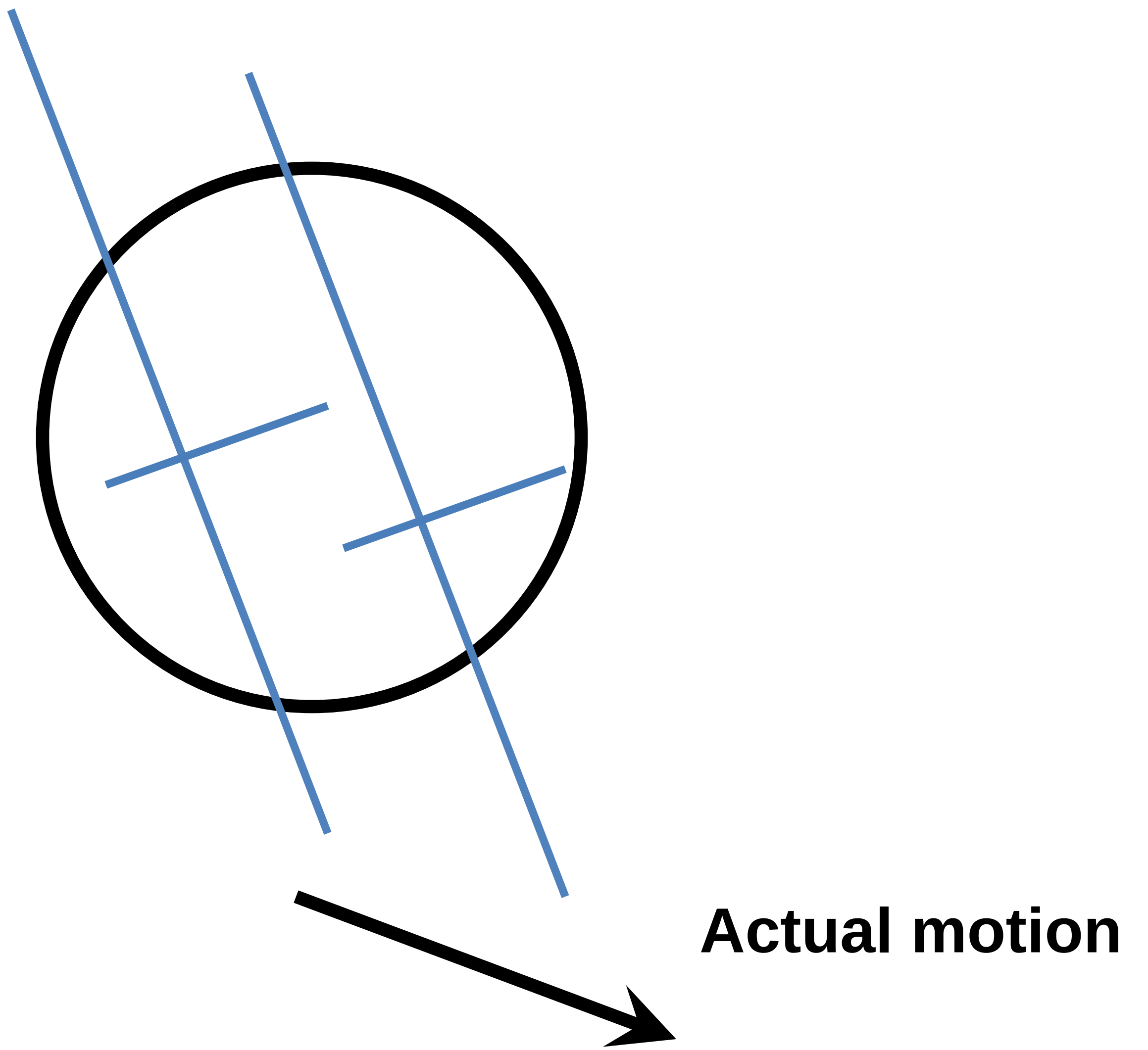
– gradients very large or very small

– large λ_1, small λ_2

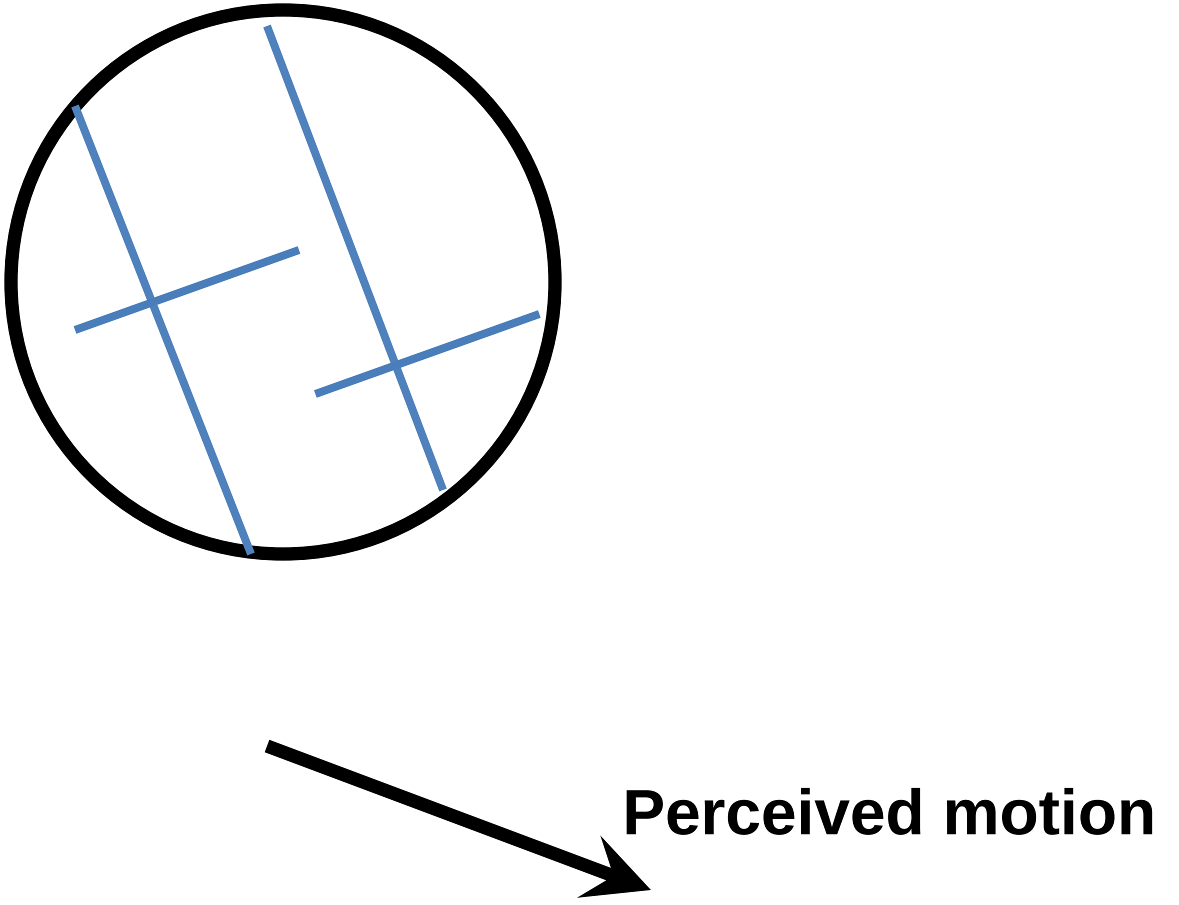
High-texture region

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

The aperture problem resolved



The aperture problem resolved



Summary of KLT algorithm

- Find a good point to track (harris corner)
- Use intensity second moment matrix and difference across frames to find displacement
- Iterate and use coarse-to-fine search to deal with larger movements
- When creating long tracks, check appearance of registered patch against appearance of initial patch to find points that have drifted

Tracking example

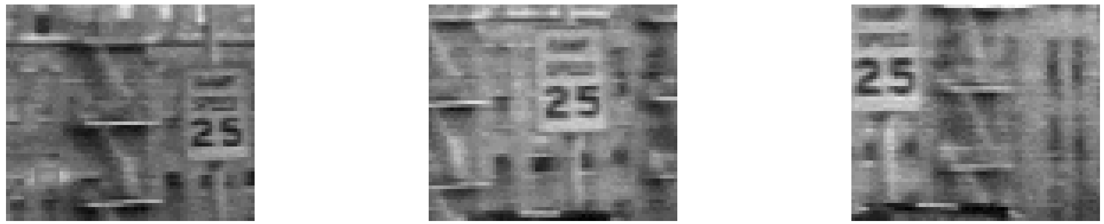


Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

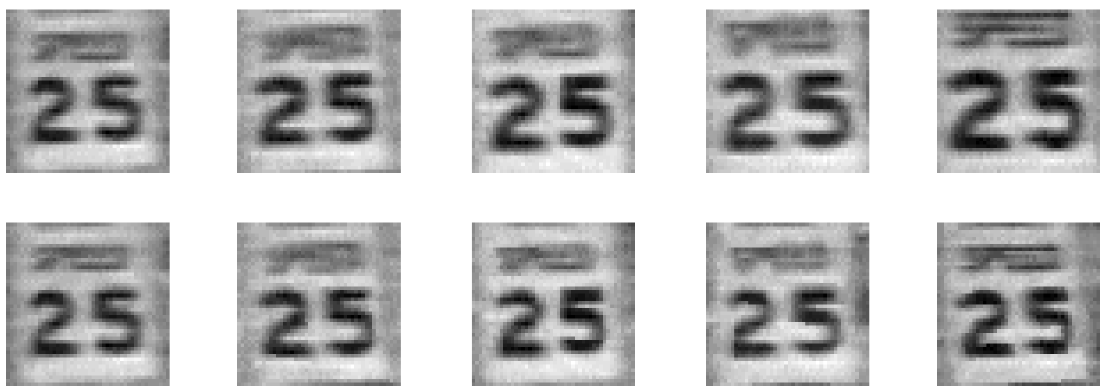
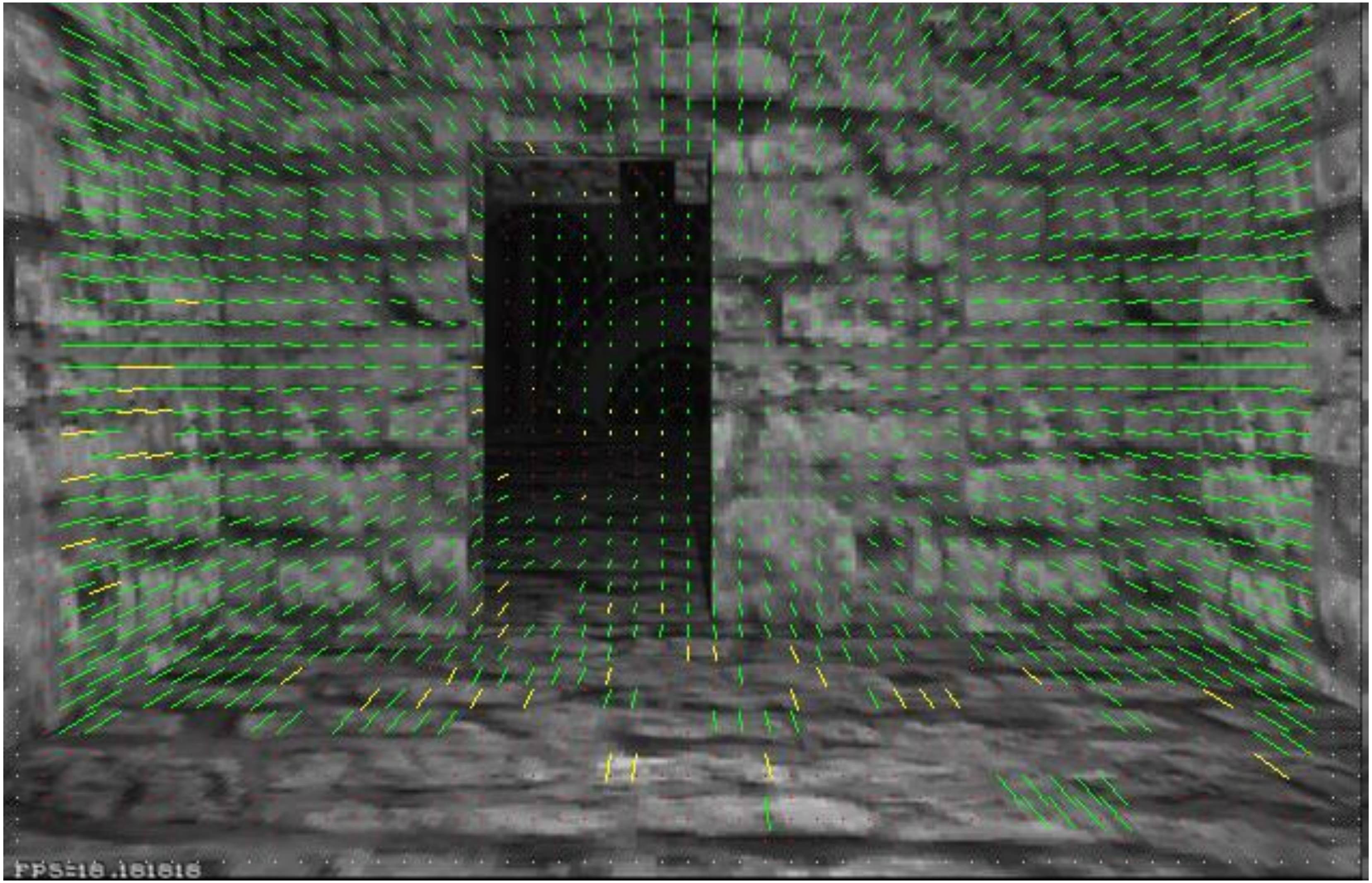


Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

Optical flow



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
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

Uses of motion

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

KLT Optical Flow

- Same as KLT feature tracking, but for each pixel:
  - As we saw, works better for textured pixels
- Operations can be done one frame at a time, rather than pixel by pixel.

Errors in KLT:

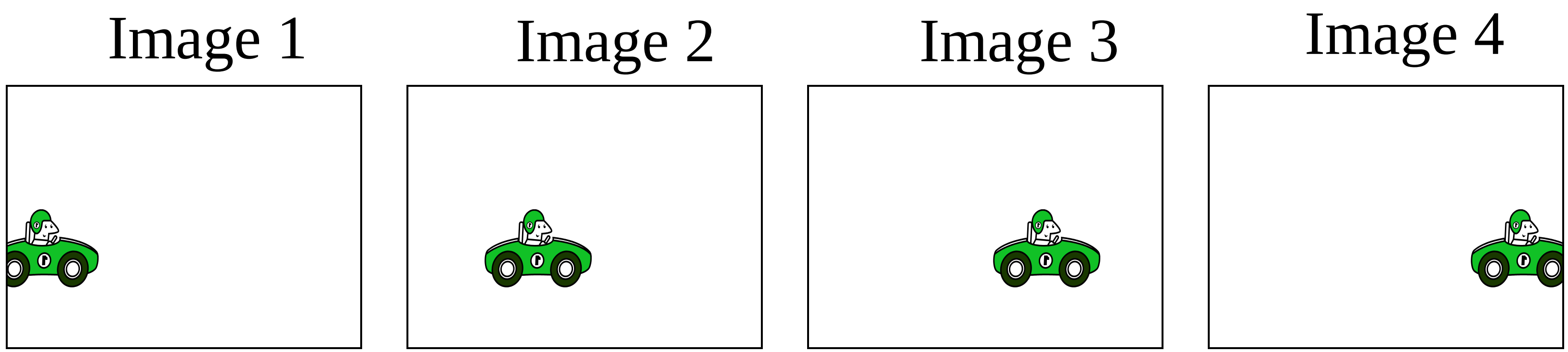
- The motion is large
  - Possible Fix: Keypoint matching
- A point does not move like its neighbors
  - Possible Fix: Region-based matching
- Brightness constancy does not hold
  - Possible Fix: Gradient constancy

KLT Feature Tracking Summary

- Major contributions from Lucas, Tomasi, Kanade (KLT)
  - Tracking feature points
  - Optical flow
- Key idea:
  - By assuming brightness constancy, truncated Taylor expansion leads to simple and fast patch matching across frames

Object Tracking

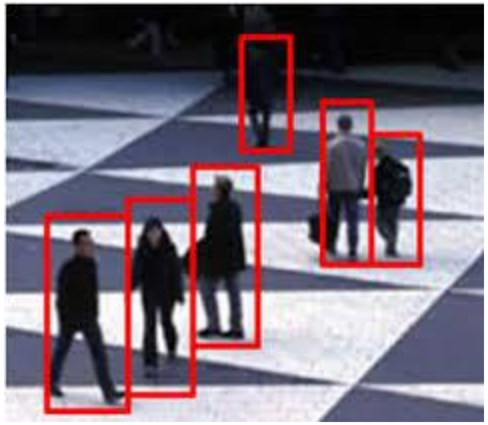
The Object Tracking Problem

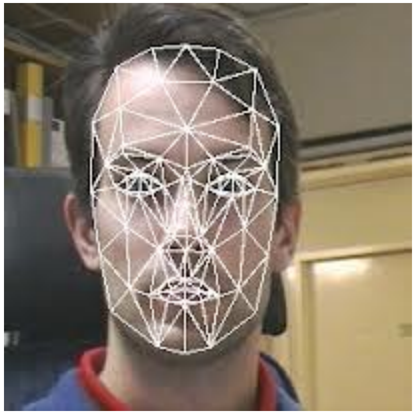


- Can we estimate the position of the object?
- Can we estimate its velocity?
- Can we predict future positions?

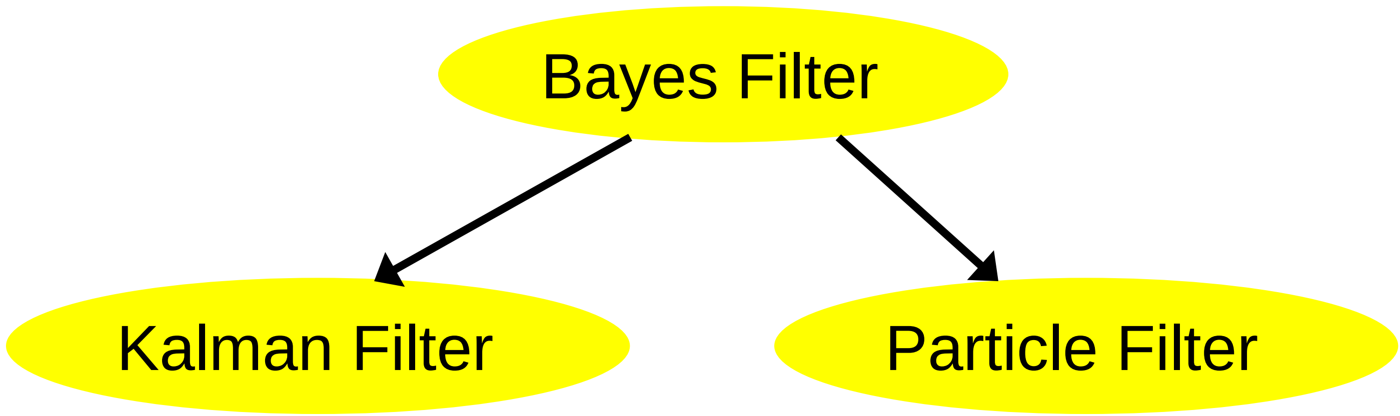
Image Features for Tracking

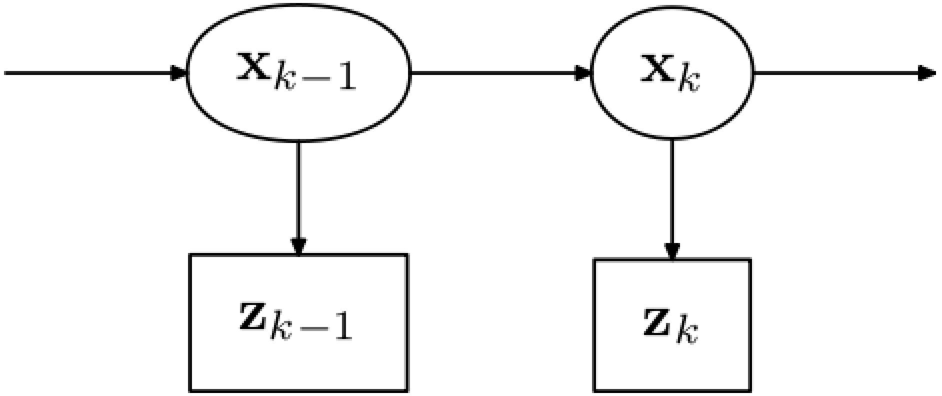
- Color
- Shape
- Motion (e.g. using local features)
- Object detector





Probabilistic Tracking Methods

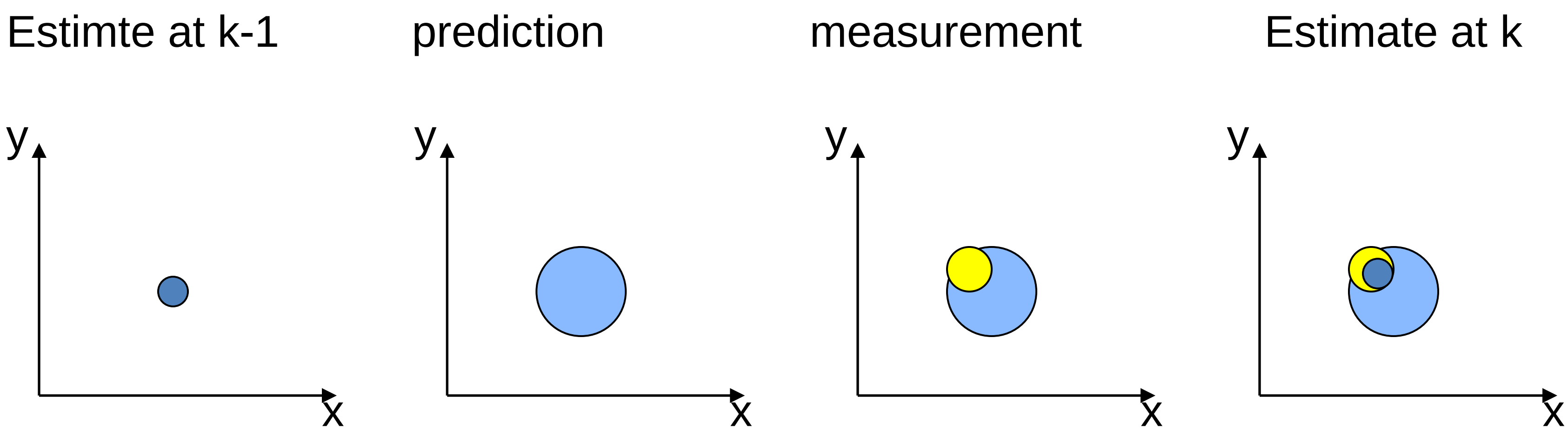




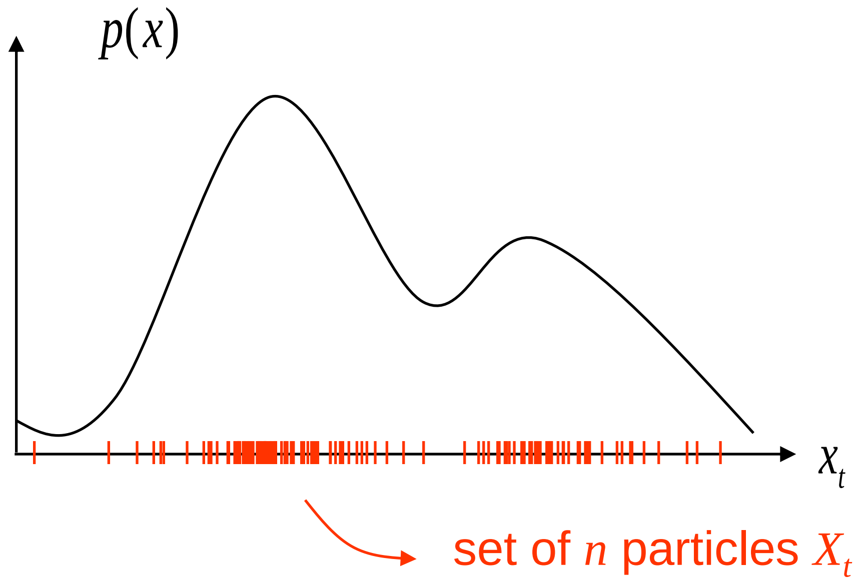
- Object state x_k
- Measurement z_k
- Estimate:

$$p(x_k \mid z_{1:k}) \propto p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})$$

Kalman Filter: Gaussian Modeling



Particle Filters: Basic Idea



$$p(x_t \in X_t) \approx p(x_t \mid z_{1\ldots t})$$

Tracking Examples

https://www.youtube.com/watch?v=wCMk-pHzScE&list=PLtrtHGBZ86GTl2nkP63cc1e2XXE9dn7y2&index=1

http://cvrlcode.ics.forth.gr/handtracking/