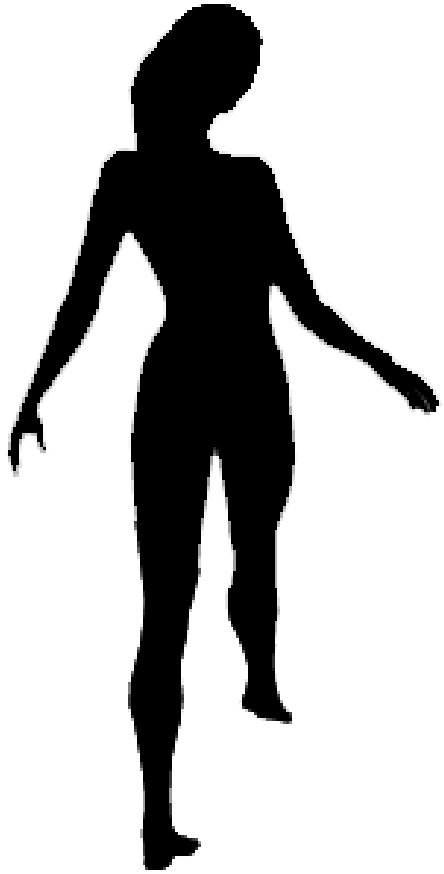
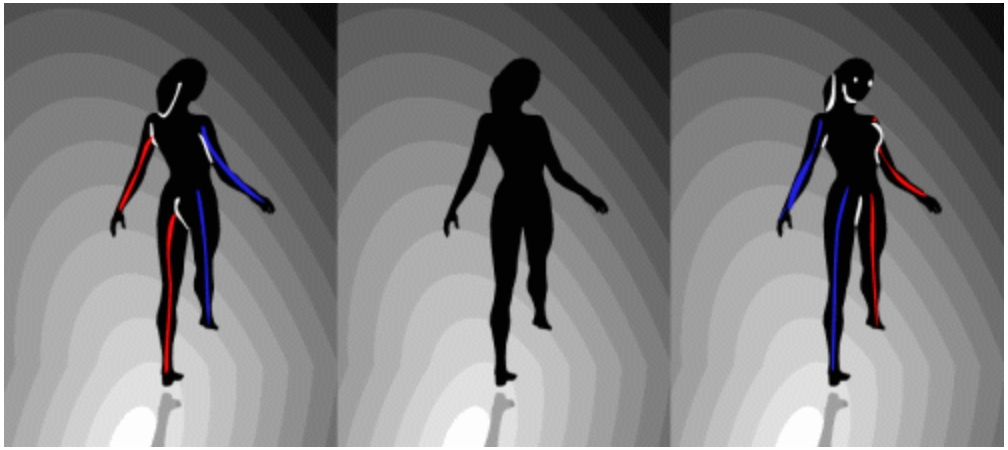
Lecture 10:
Motion and optical flow



Spinning dancer illusion, Nobuyuki Kayahara

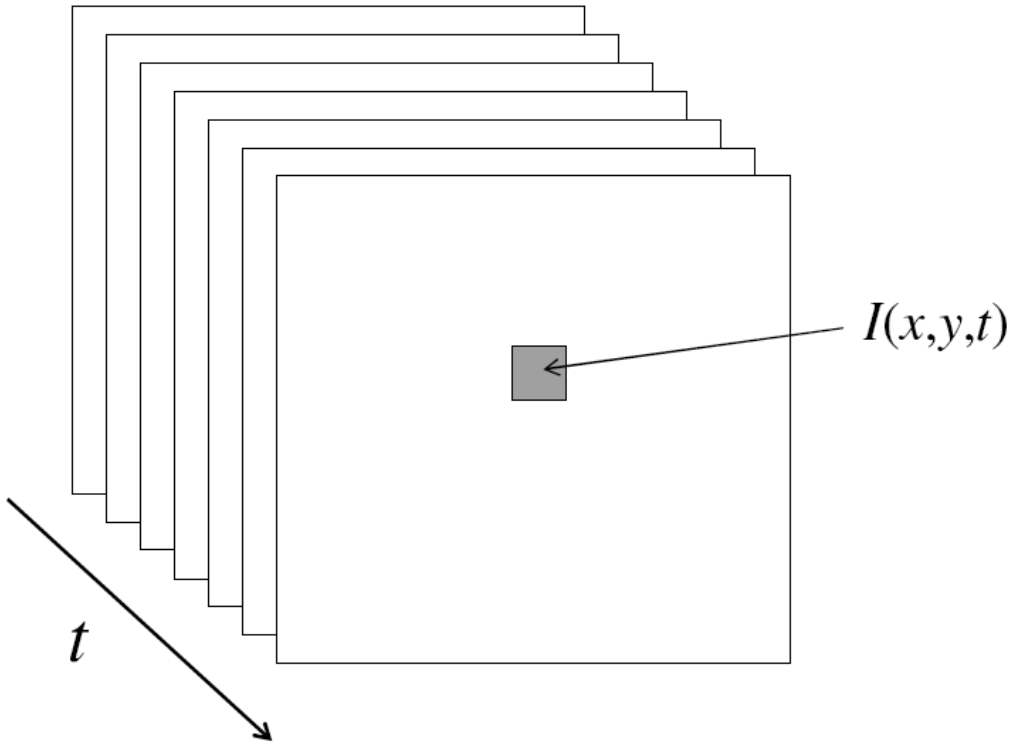


Now

- Optical flow: estimating motion in video
- Background subtraction

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)

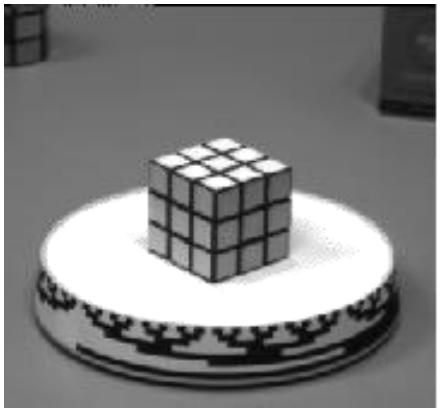


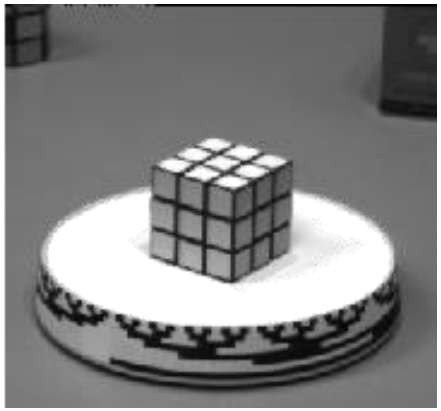
Uses of motion

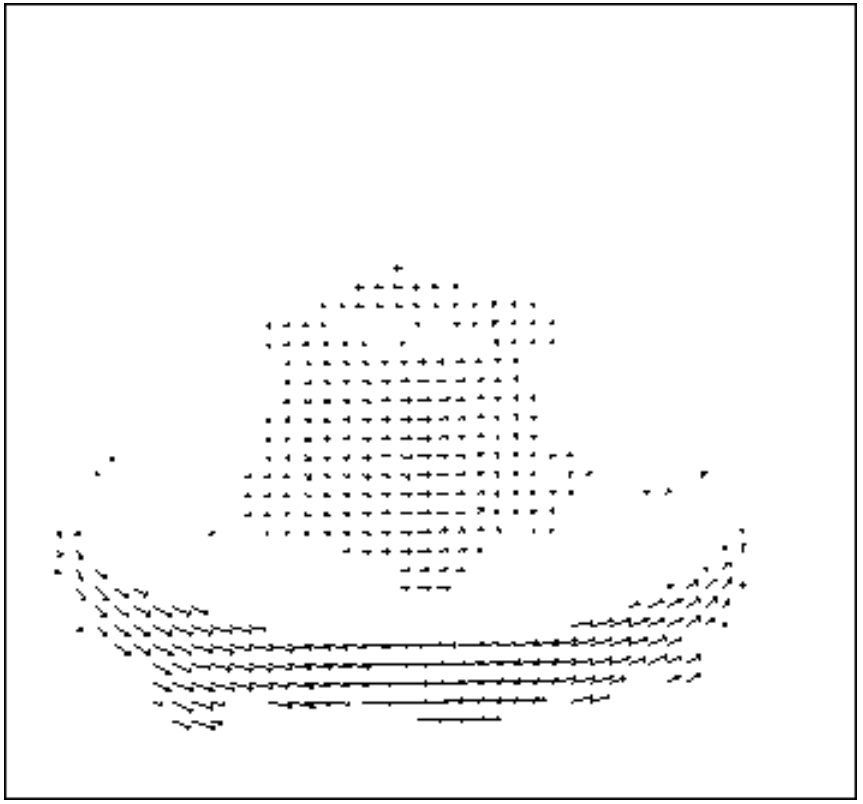
- Estimating 3D structure
- Segmenting objects based on motion cues
- Recognizing events and activities
- Improving video quality (motion stabilization)

Motion field

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







Motion field + camera motion

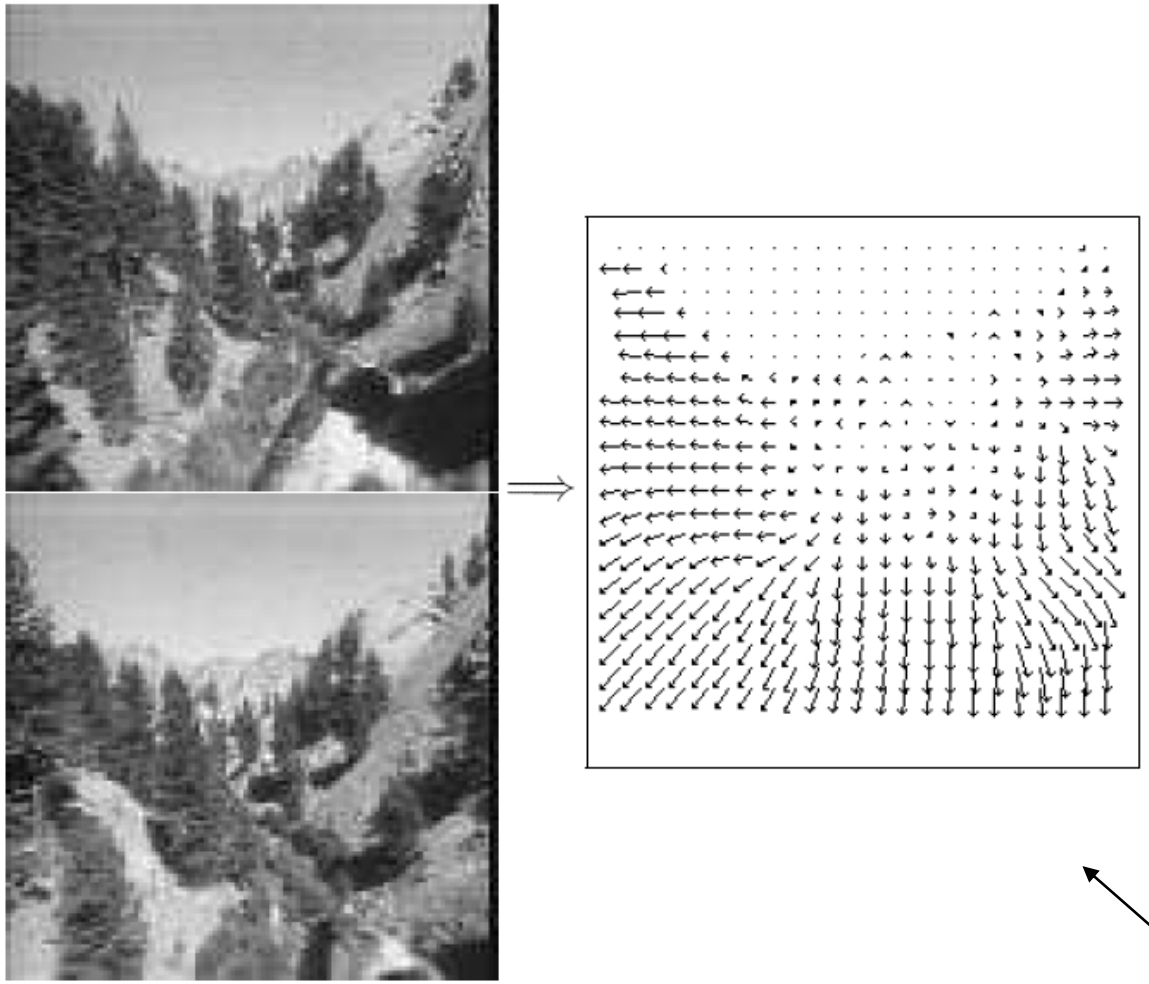


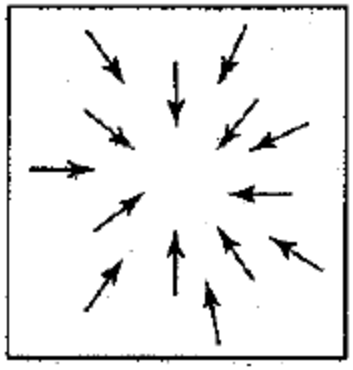
Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

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

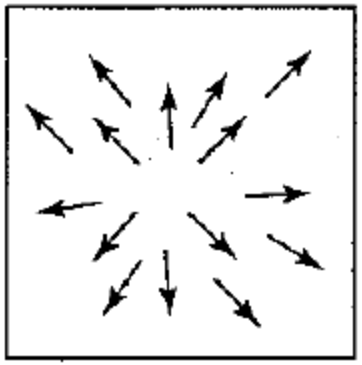
points closer to the camera move more quickly across the image plane

Figure from Michael Black, Ph.D. Thesis

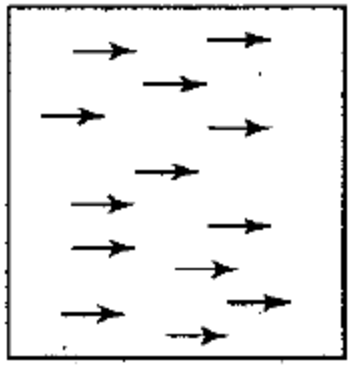
Motion field + camera motion



Zoom out



Zoom in



Pan right to left

Motion estimation techniques

- 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 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)

Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image

Thomas Brox, large displacement optical flow.
https://lmb.informatik.uni-freiburg.de/research/opticalflow/

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

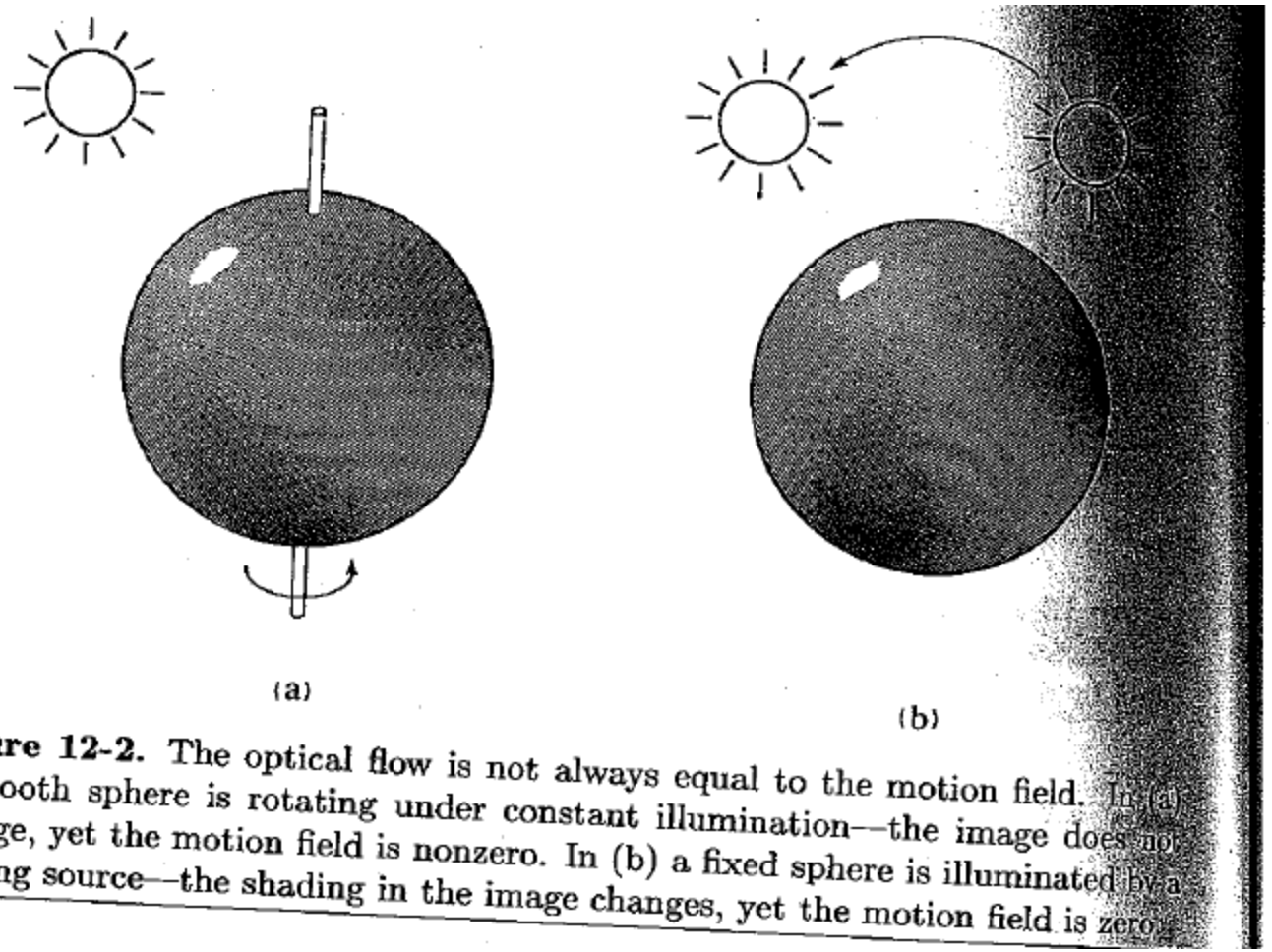
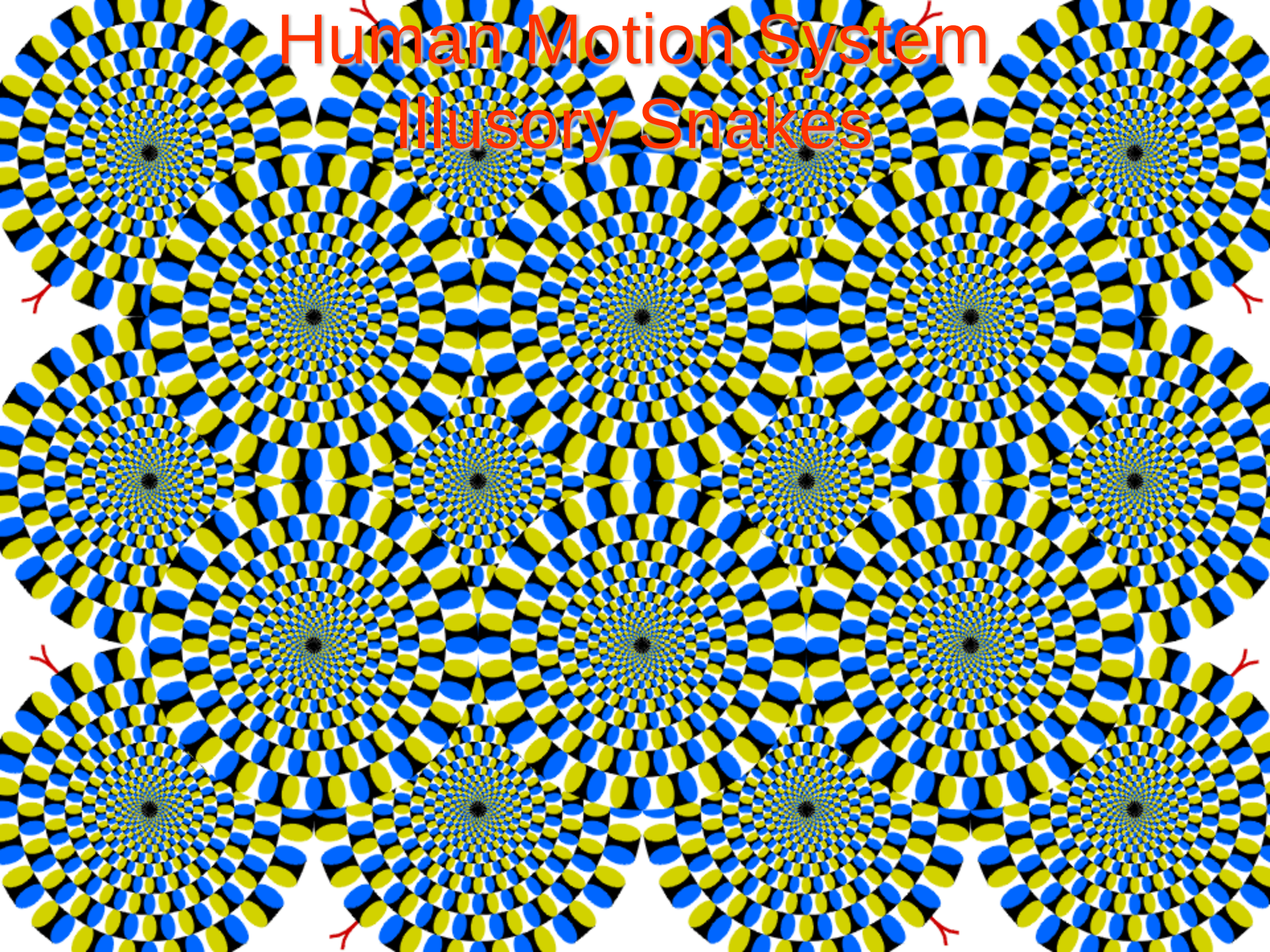


Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field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

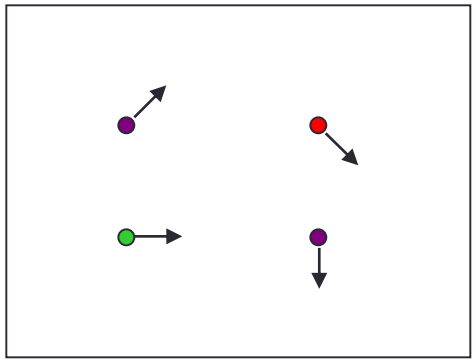
Figure from Horn book

Human Motion System

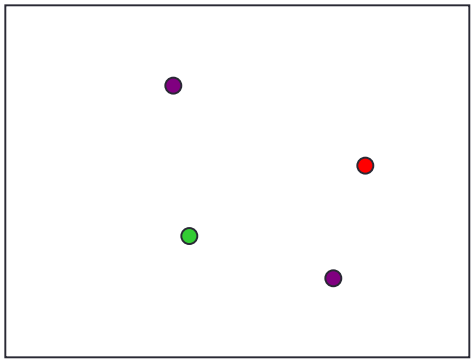
Illusory Snakes



Problem definition: optical flow



$I(x, y, t)$



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

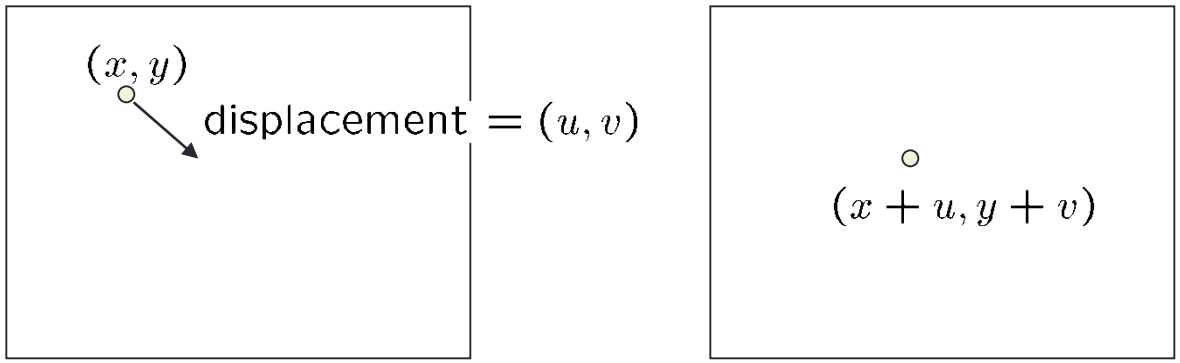
How to estimate pixel motion from image $I(x,y,t)$ to $I(x,y,t+1)$?

- Solve pixel correspondence problem
  - Given a pixel in I(x,y,t), look for nearby pixels of the same color in *I(x,y,t+1)*

Key assumptions

- Small motion: Points do not move very far
- Color constancy: A point in *I(x,y,t)* looks the same in *I(x,y,t+1)*
  - For grayscale images, this is brightness constancy

Optical flow constraints (grayscale images)



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

- Let's look at these constraints more closely
  - Brightness constancy constraint (equation)

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

  - Small motion: (u and v are less than 1 pixel, or smoothly varying)
    Taylor series expansion of I:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + [\text{higher order terms}]$$

$$\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

Optical flow equation

- Combining these two equations

$$0 = I(x+u, y+v, t+1) - I(x, y, t)$$

(Short hand: $I_x = \frac{\partial I}{\partial x}$ for t **or** $t+1$)

Optical flow equation

- Combining these two equations

$$
\begin{aligned}
0 &= I(x+u, y+v, t+1) - I(x, y, t) \\
&\approx I(x, y, t+1) + I_x u + I_y v - I(x, y, t) \\
&\approx [I(x, y, t+1) - I(x, y, t)] + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot < u, v >
\end{aligned}
$$

(Short hand: $I_x = \frac{\partial I}{\partial x}$ for t **or** $t+1$)

Optical flow equation

- Combining these two equations

$$
\begin{aligned}
0 &= I(x+u, y+v, t+1) - I(x,y,t) \\
&\approx I(x,y,t+1) + I_x u + I_y v - I(x,y,t) \\
&\approx [I(x,y,t+1) - I(x,y,t)] + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot < u, v >
\end{aligned}
$$

(Short hand: $I_x = \frac{\partial I}{\partial x}$ for t **or** $t+1$)

In the limit as u and v go to zero, this becomes exact

$$0 = I_t + \nabla I \cdot < u, v >$$

Brightness constancy constraint equation

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

How does this make sense?

Brightness constancy constraint equation

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

The brightness constancy constraint

Can we use this equation to recover image motion (u,v) at each pixel?

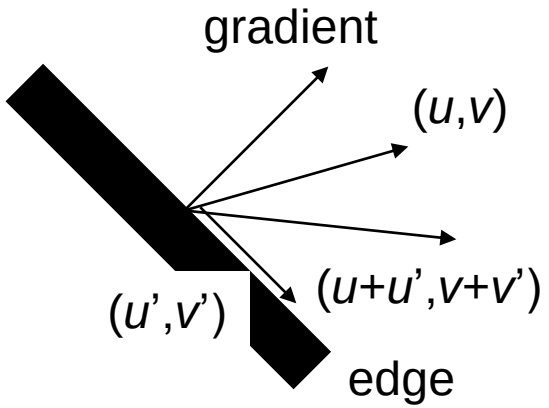
$$0 = I_t + \nabla I \cdot < u, v > \quad \text{or} \quad I_x u + I_y v + I_t = 0$$

- How many equations and unknowns per pixel?
  - One equation (this is a scalar equation!), two unknowns (u,v)

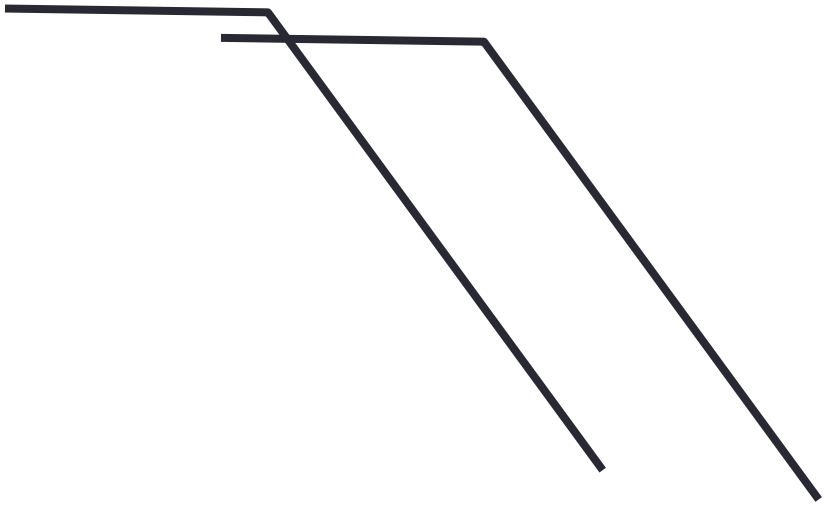
The component of the motion perpendicular to the gradient (i.e., parallel to the edge) cannot be measured

If (*u*, *v*) satisfies the equation, so does (*u*+*u'*, *v*+*v'*) if

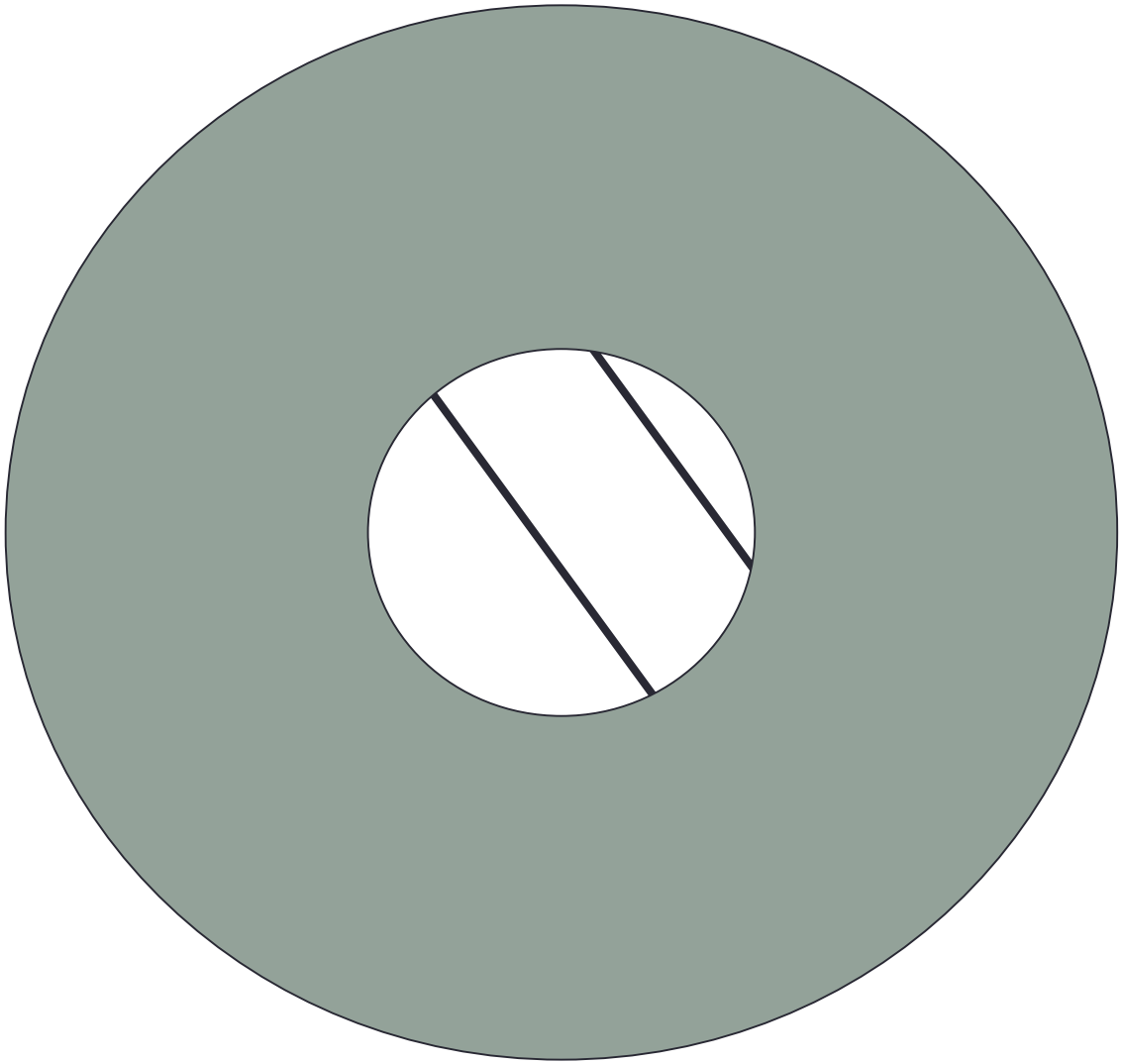
$$\nabla \mathrm{I} \cdot \left[\mathrm{u'}\ \mathrm{v'}\right]^{\mathrm{T}} = 0$$



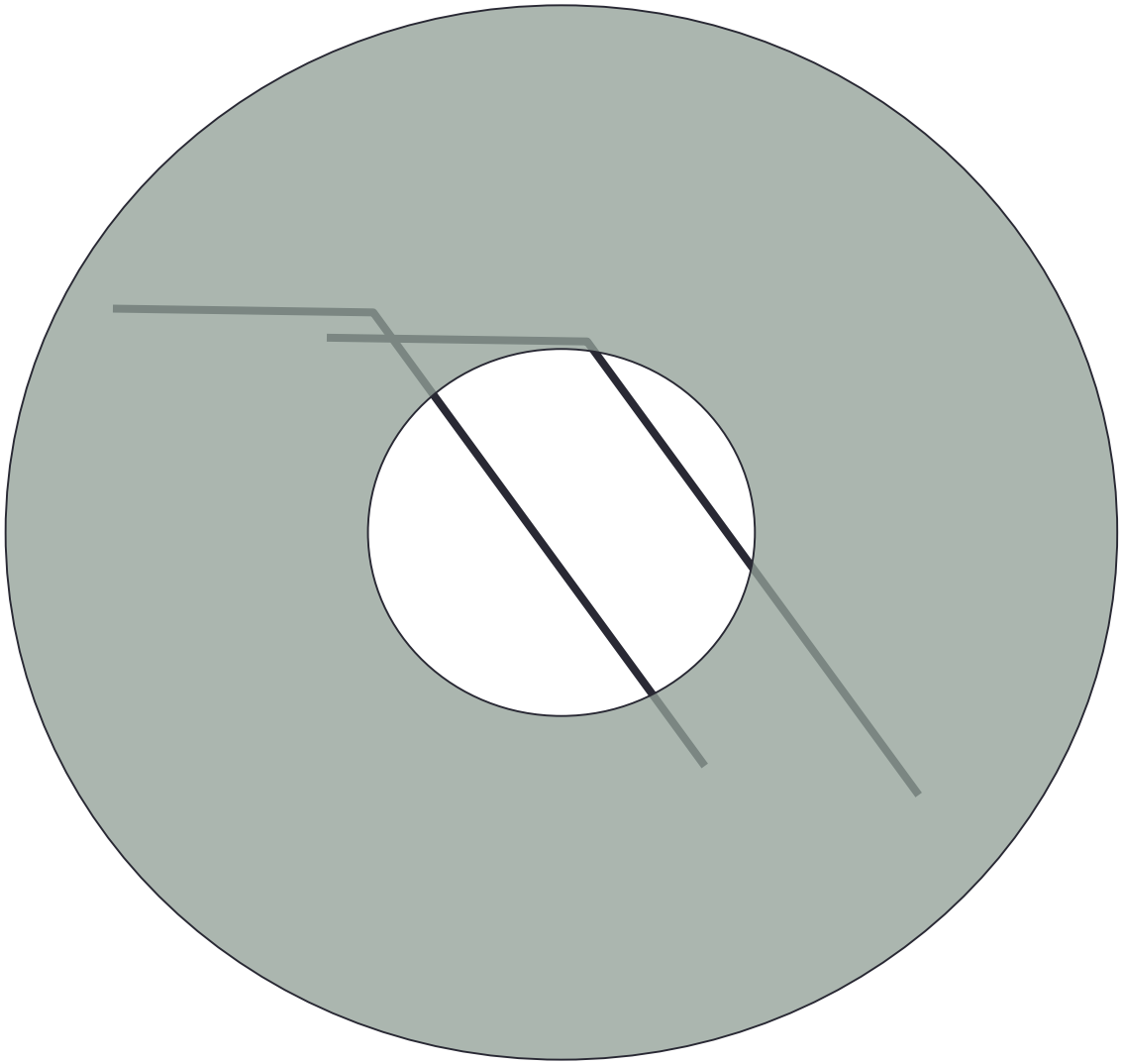
Aperture problem



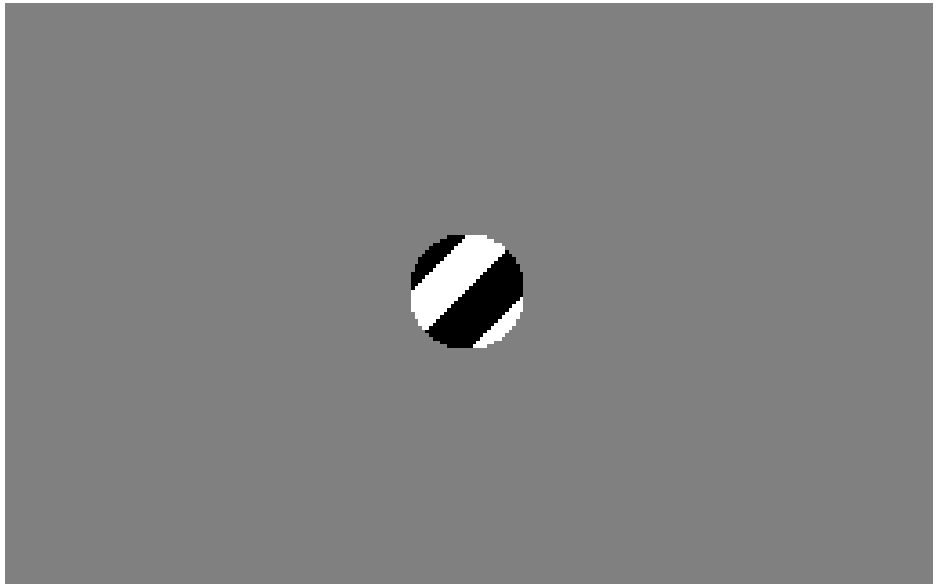
Aperture problem



Aperture problem

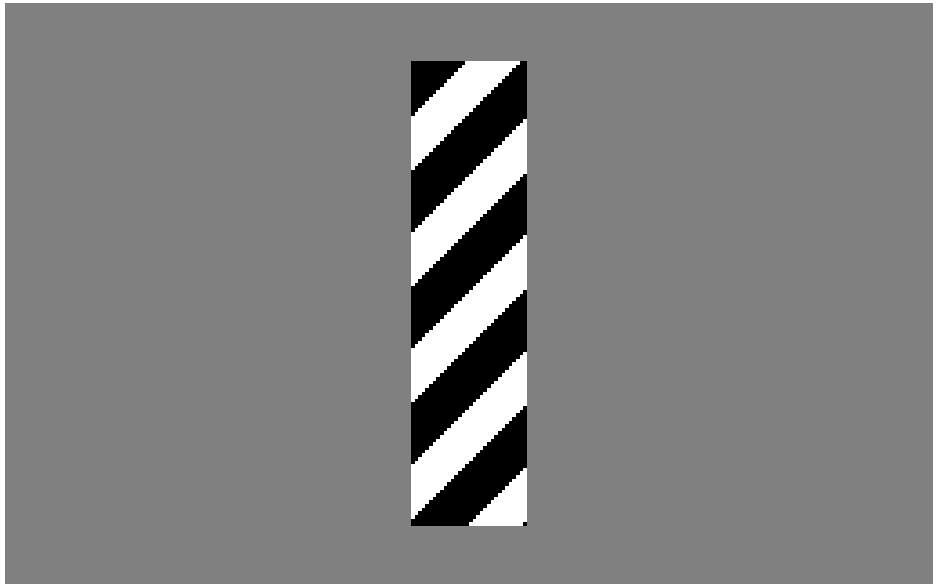


The barber pole illusion



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

The barber pole illusion



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

Solving the ambiguity…

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 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

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

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

Solving the ambiguity…

- Least squares problem:

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

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

Matching patches across images

- Overconstrained linear system

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

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

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

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

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

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

Conditions for solvability

Optimal (u, v) satisfies Lucas-Kanade equation

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

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

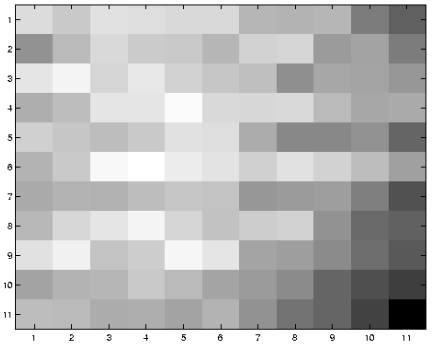
When is this solvable? What are good points to track?

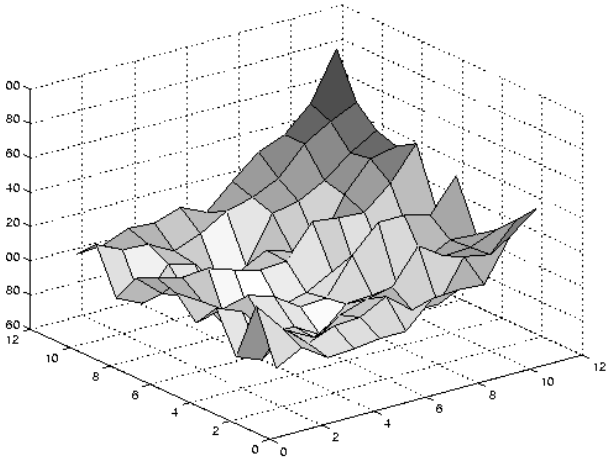
- **A**^{T}**A** should be invertible
- **A**^{T}**A** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **A**^{T}**A** should not be too small
- **A**^{T}**A** should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Does this remind you of anything?

Criteria for Harris corner detector

Low texture region

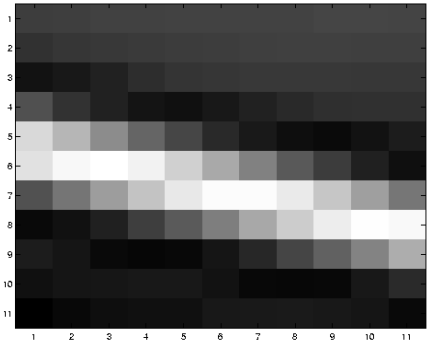


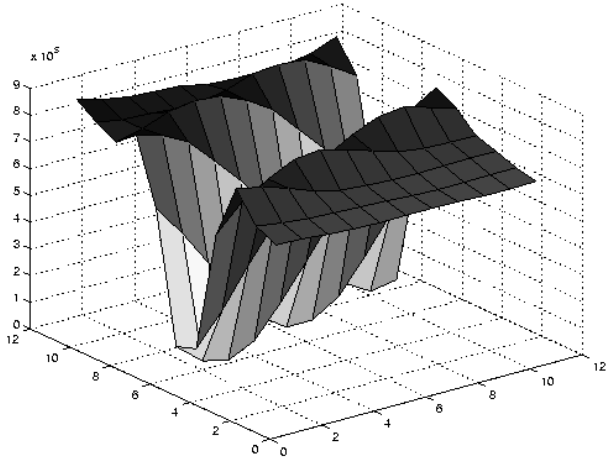


$$\sum \nabla I (\nabla I)^T$$

- gradients have small magnitude
- small λ_1, small λ_2

Edge

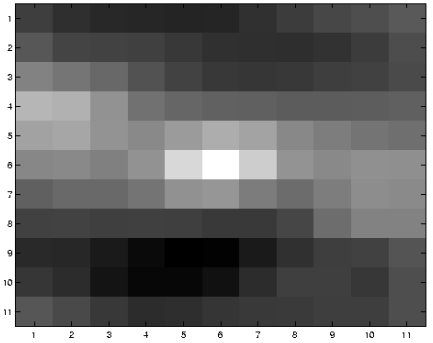


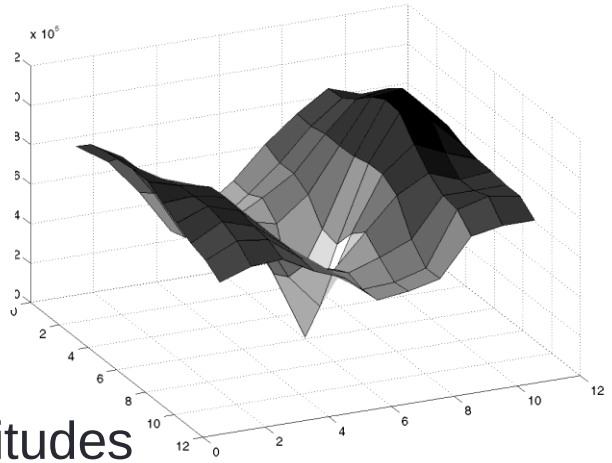


$$\sum \nabla I (\nabla I)^T$$

- large gradients, all the same
- large λ_1, small λ_2

High textured region





$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

– large λ_1, large λ_2

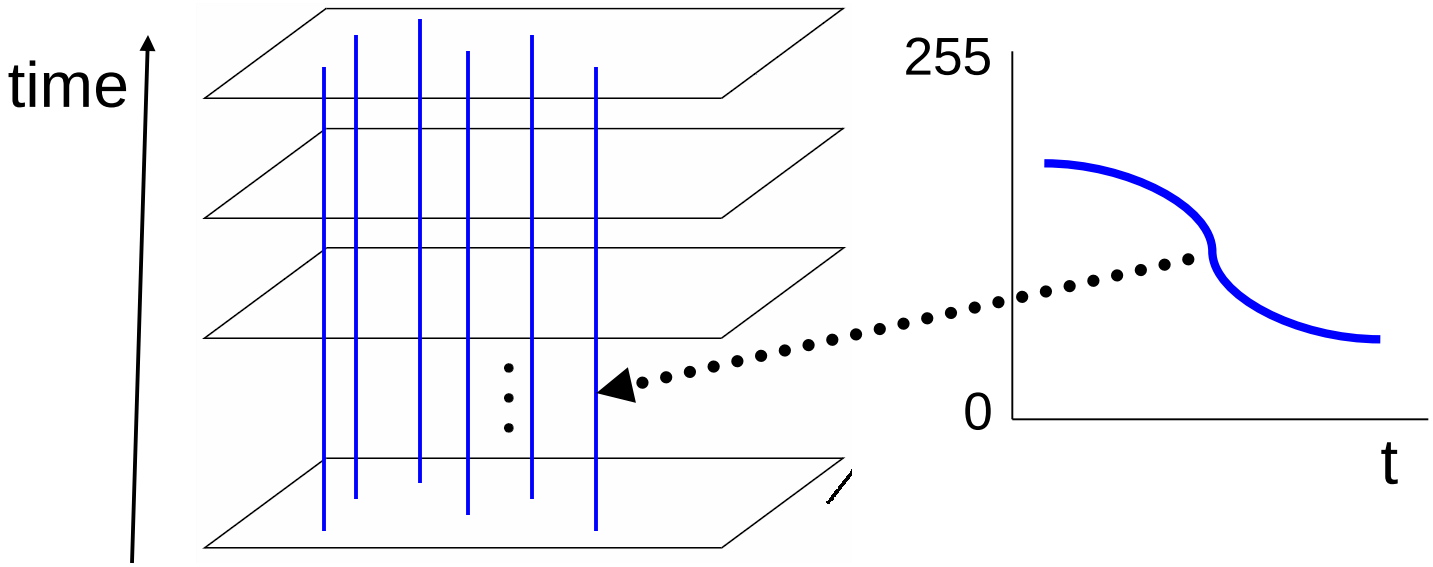
Motion estimation techniques

- 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 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)

Today

- Optical flow: estimating motion in video
- 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

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: $I(x, y, t)$

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

$$| \quad - \quad | > Th$$

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**.

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 Tamersoy

Frame Differencing

$Th = 25$

$Th = 50$

$Th = 100$

$Th = 200$

Slide credit: Birgi Tamersoy

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 Tamersoy

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

Background Subtraction

-

=

Pros and cons

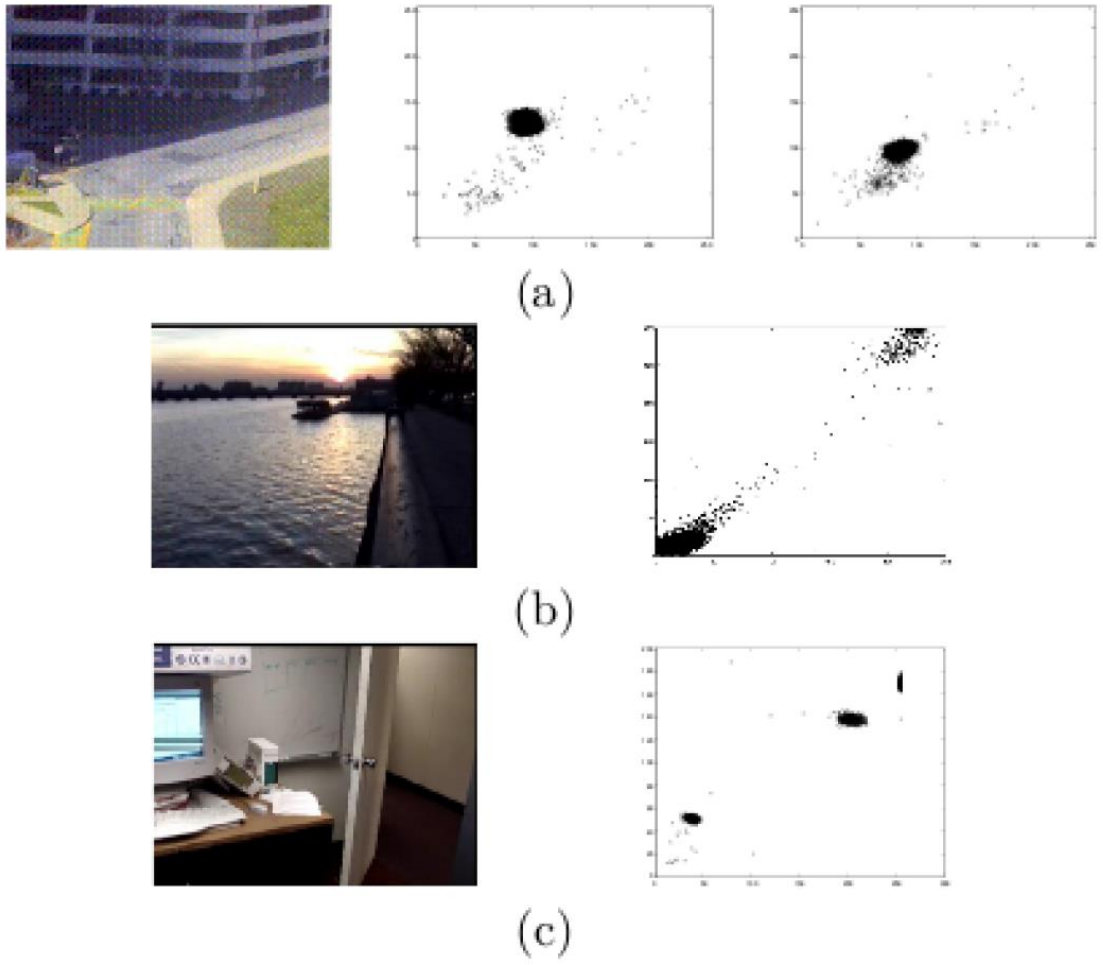
Advantages:

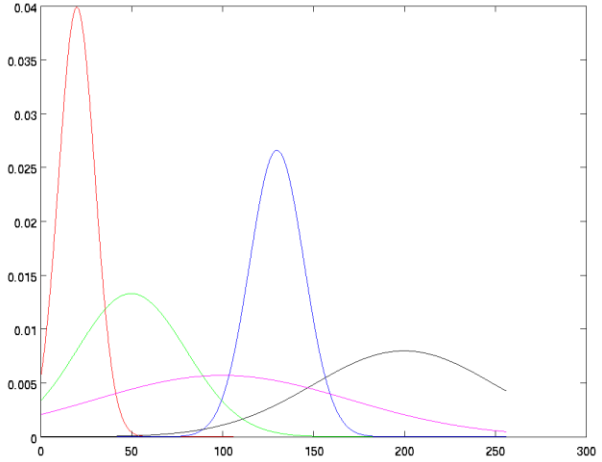
- Extremely easy to implement and use!
- All pretty fast.
- Corresponding background models need not be constant, they change over time.

Disadvantages:

- Accuracy of frame differencing depends on object speed and frame rate
- Median background model: relatively high memory requirements.
- Setting global threshold Th…

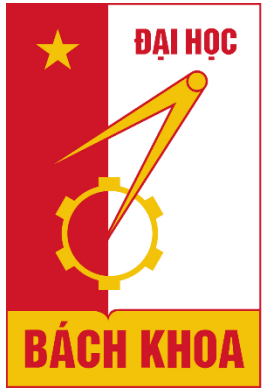
Background mixture models





Idea: model each background pixel with a *mixture* of Gaussians; update its parameters over time.

- Adaptive Background Mixture Models for Real-Time Tracking, Chris Stauffer & W.E.L. Grimson



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Object Tracking

Intro to Object Tracking

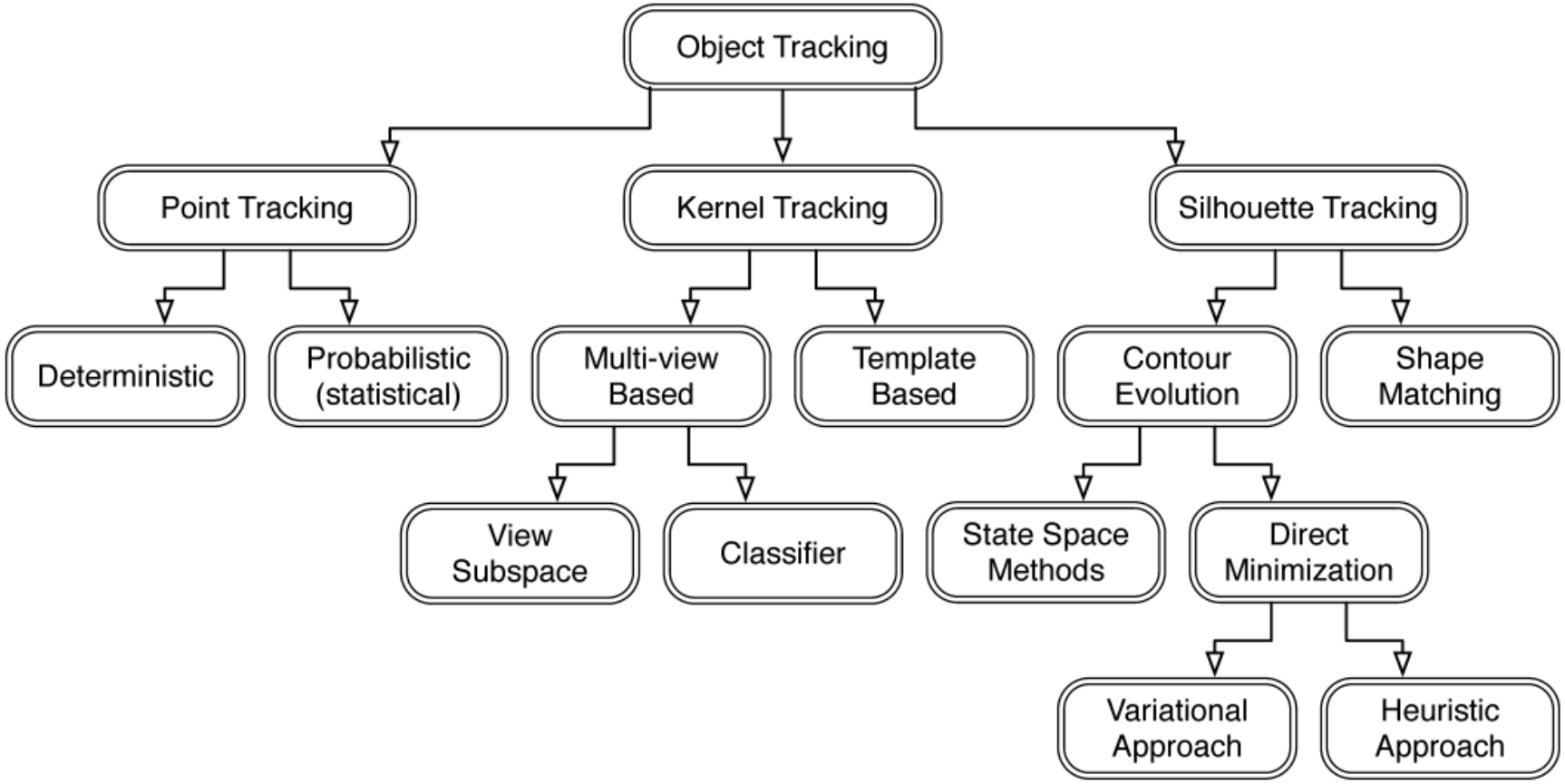
Objective: generate the trajectory of an object over time by locating its position in every frame of the video

Two key steps:

- Detecting object

- Establishing correspondence between the object instances across frames

Object Tracking



Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

- Objects detected in consecutive frames are represented by points

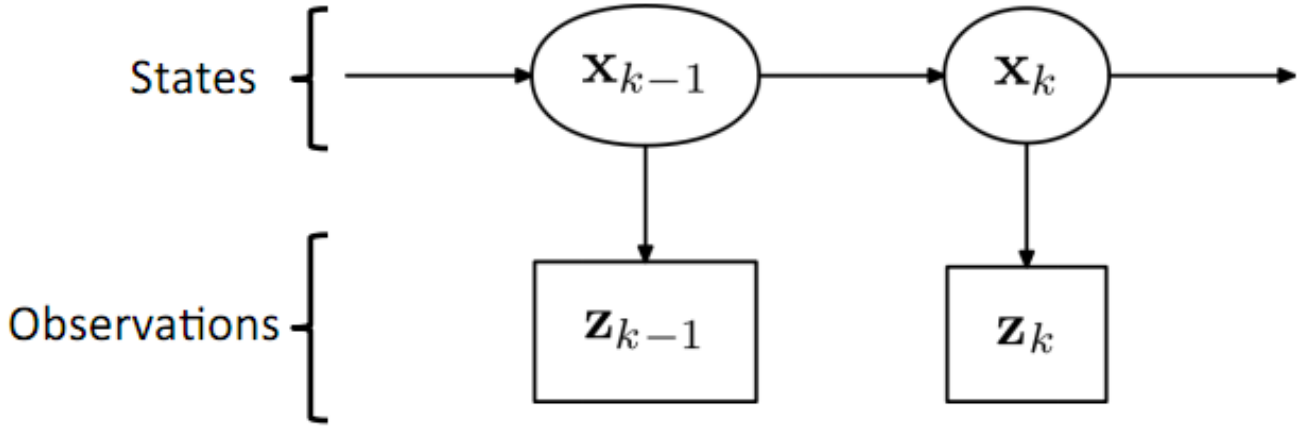
- requires an external mechanism to detect the objects in every frame

Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

Bayesian filter

- Hidden Markov Model



- Markov assumptions

$$p(x_k \mid x_{1:k-1}) = p(x_k \mid x_{k-1})$$

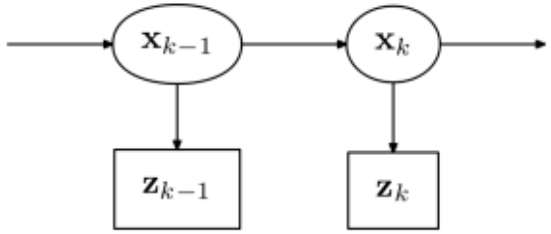
$$p(z_k \mid x_{1:k}) = p(z_k \mid x_k)$$

Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

Bayesian filter

- Recursive Bayes filters
- Find posterior $p(x_k \mid z_{1:k})$
- State eq. (motion dynamics) $f(x_k \mid x_{k-1})$
- Observation eq. (image) $g(z_k \mid x_k)$



- Prediction

Previous posterior

$$p(x_k \mid z_{1:k-1}) = \int f(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1}$$

- Update

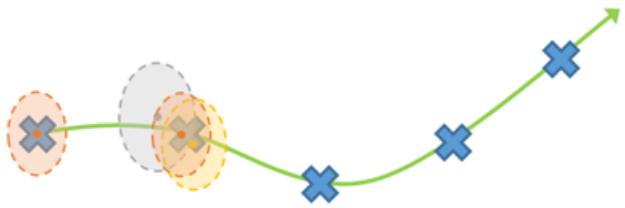
$$p(x_k \mid z_{1:k}) = \frac{g(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{\int g(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k}$$

Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

Kalman filter

- Linear system
- Gaussian noise
- give “optimal solution”

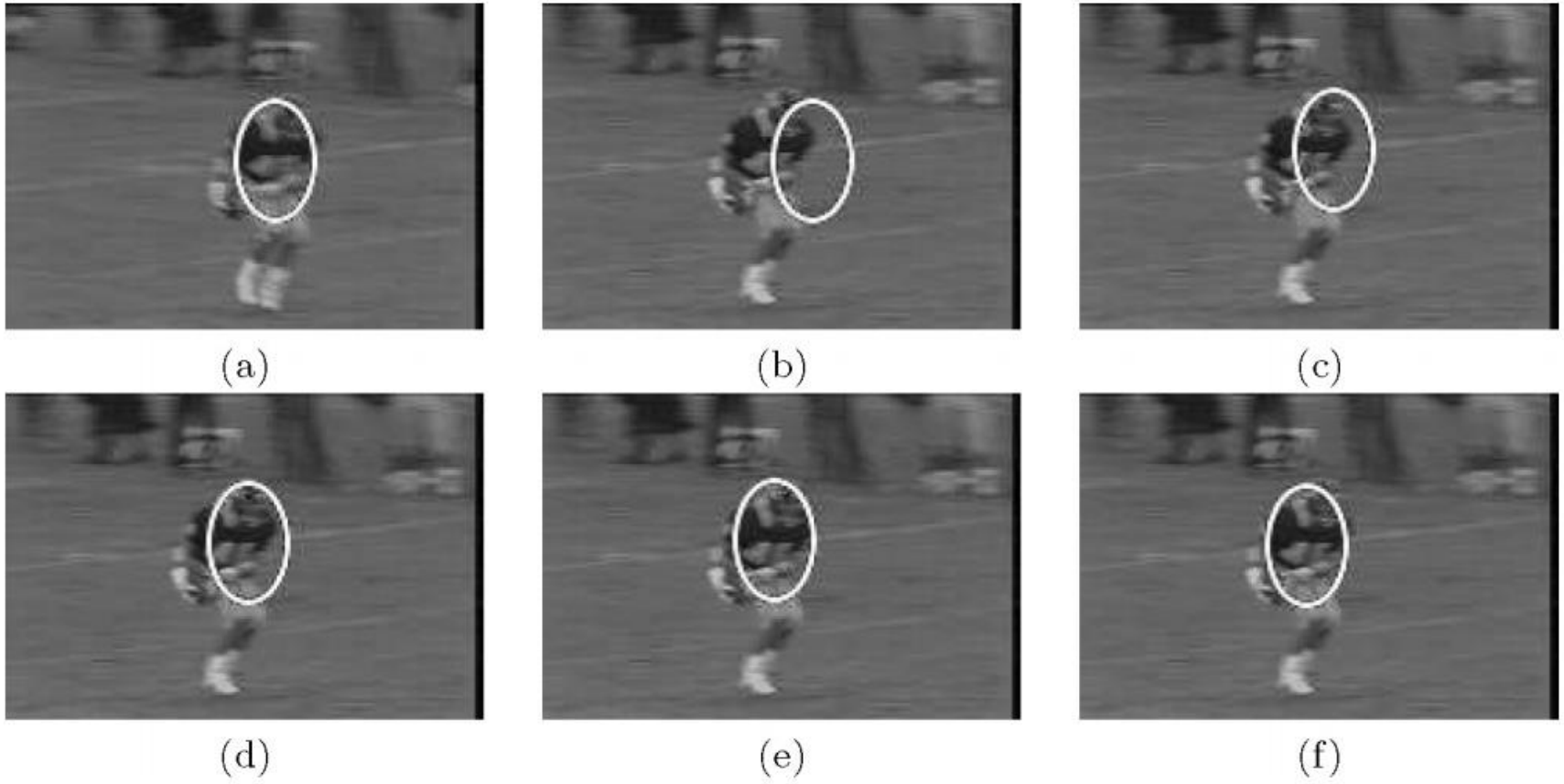


Prediction step (t)	Correction step (t+1)
1. State vector prediction $\vec{x}_{t+1\|t} = A \cdot \vec{x}_t + B \cdot \vec{u}_t$	1. Compute de Kalman gain $K_{t+1} = P_{t+1\|t} \cdot H^T (H.P_{t+1\|t}.H^T + R)^{-1}$
2. Error covariance matrix prediction $P_{t+1\|t} = A \cdot P_{t-1}.A^T + Q$	2. State vector correction $\vec{x}_{t+1} = \vec{x}_{t+1\|t} + K_{t+1} \cdot (\vec{y}_{t+1} - h(\vec{x}_t))$
3. Predicted measurements vector $\vec{y}_{t+1\|t} = H \cdot \vec{x}_t$	3. Error covariance matrix correction $P_{t+1} = P_{t+1\|t} - K_{t+1}.C.P_{t'+1\|t}$

Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

Template and Density-Based Appearance Models – Mean-shift



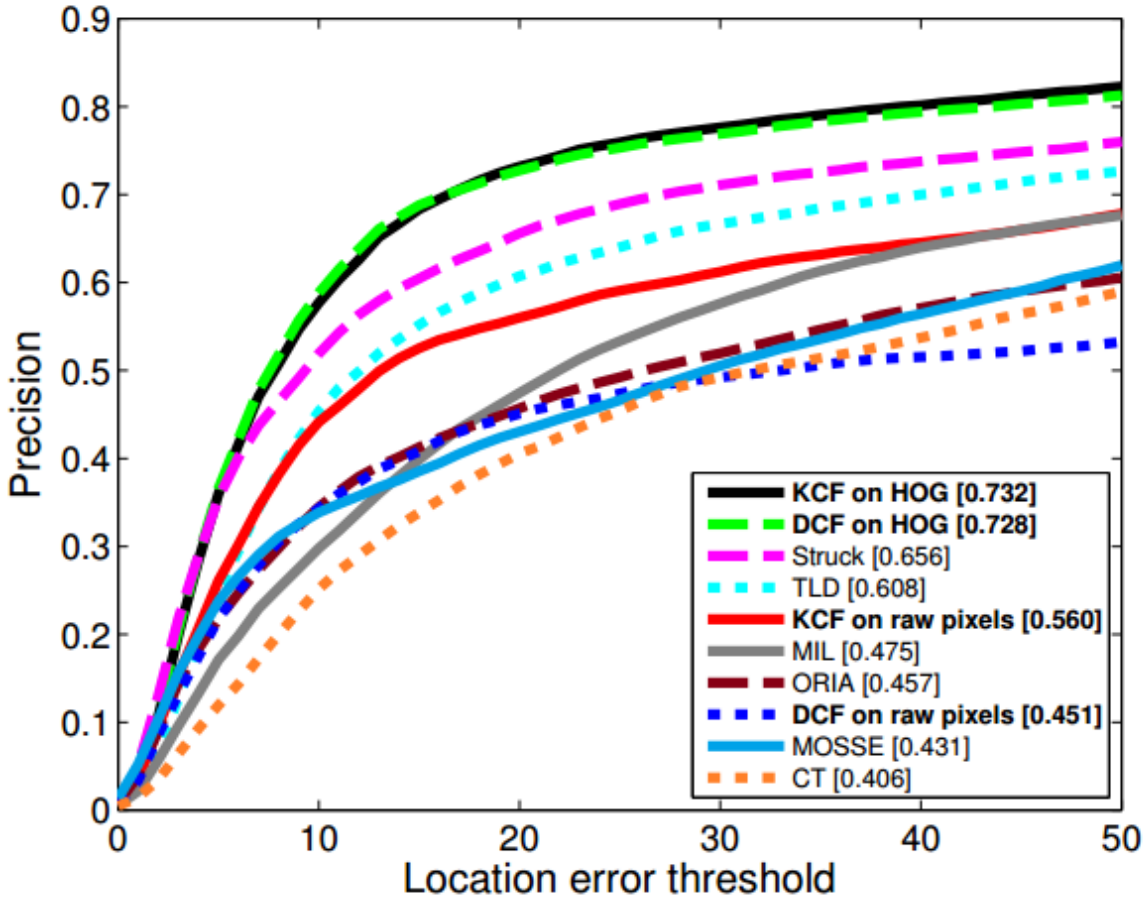
Object Tracking

Visual tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

- KCF 2014
- Struck 2014
- TLD 2010
- MIL 2009
- Online boosting 2006

...



Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

- find the object region in each frame by means of an object model (color histogram, object edges or the object contour …) generated using the previous frames

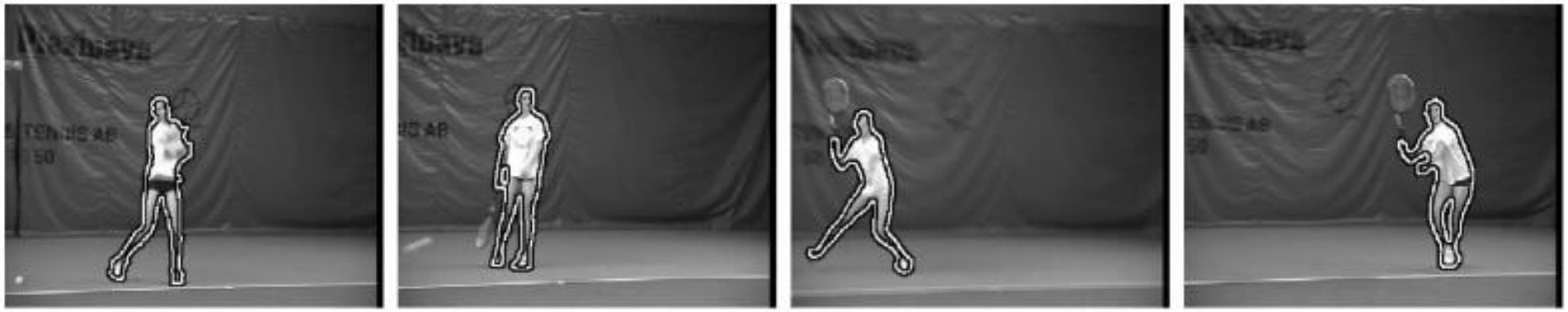
- tracking complete region of an object

- provide an accurate shape description for those objects with complex shapes

Object Tracking

Point Tracking | Kernel Tracking | Silhouette Tracking

Shape Matching and Contour Tracking



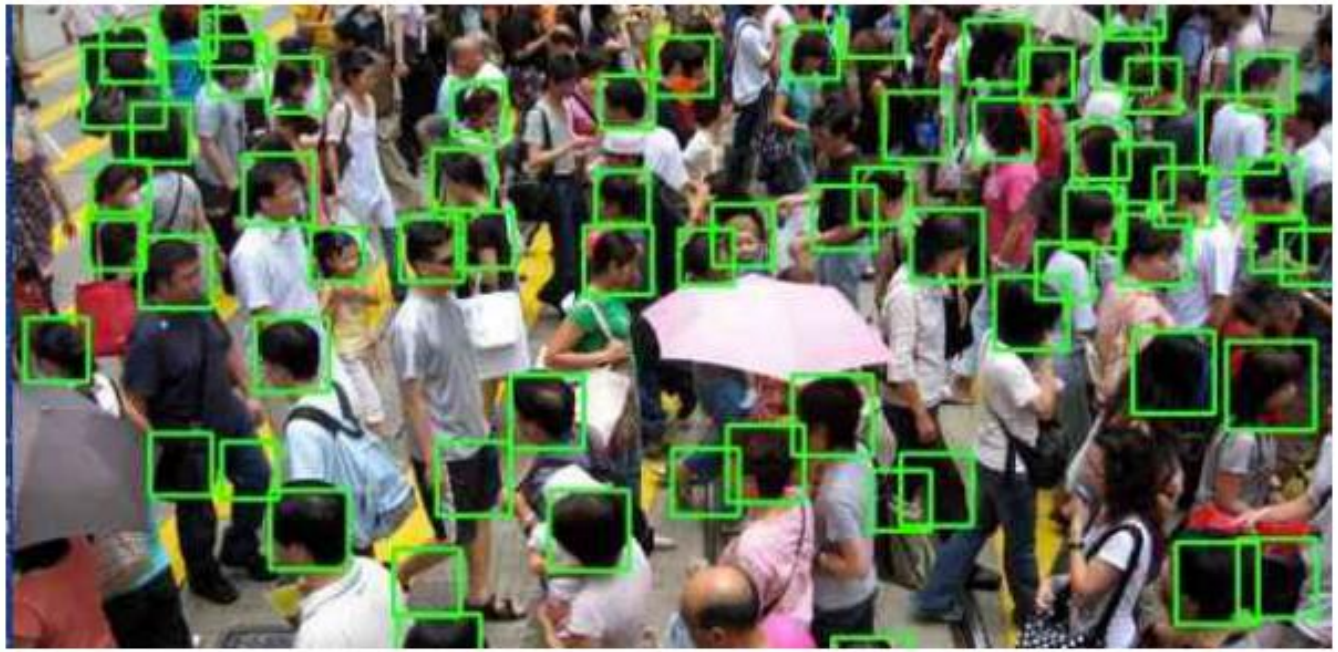
(a)

(b)

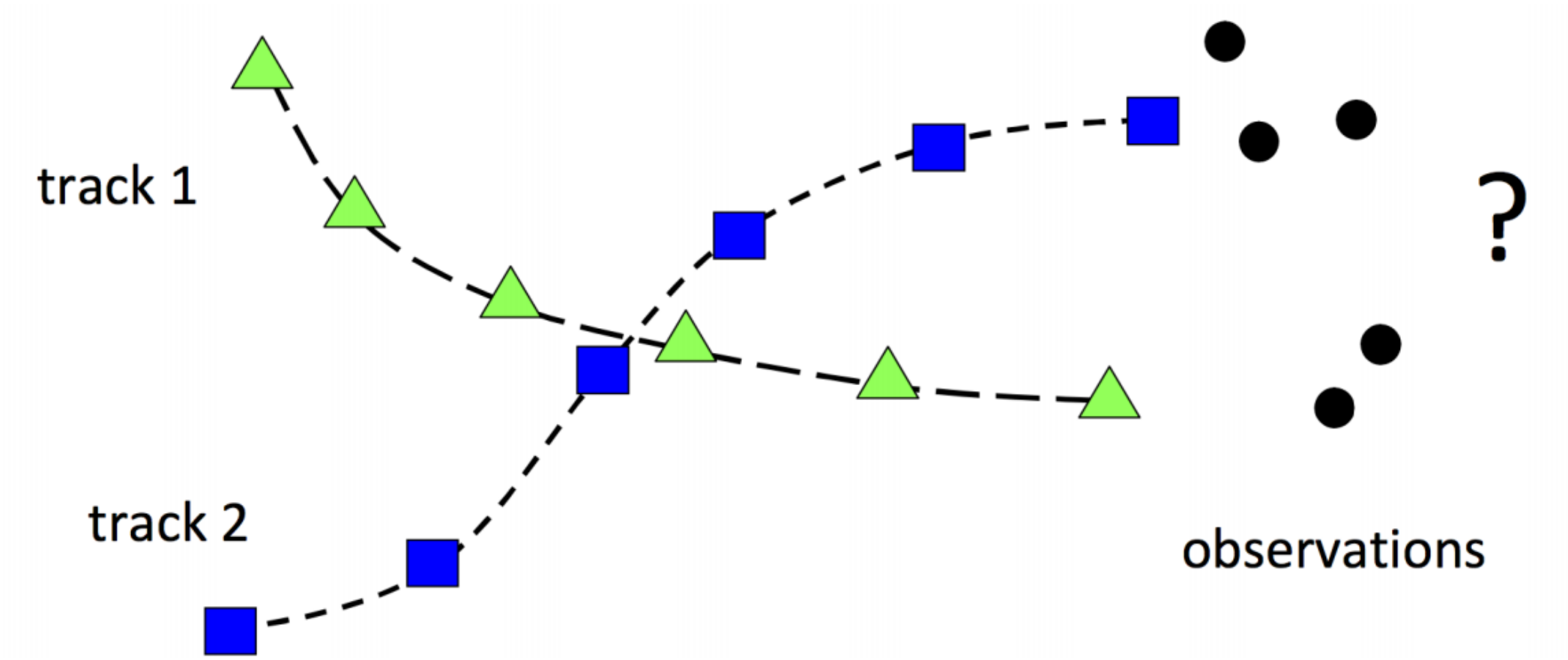
Object tracking – Multi target tracking

- Data association

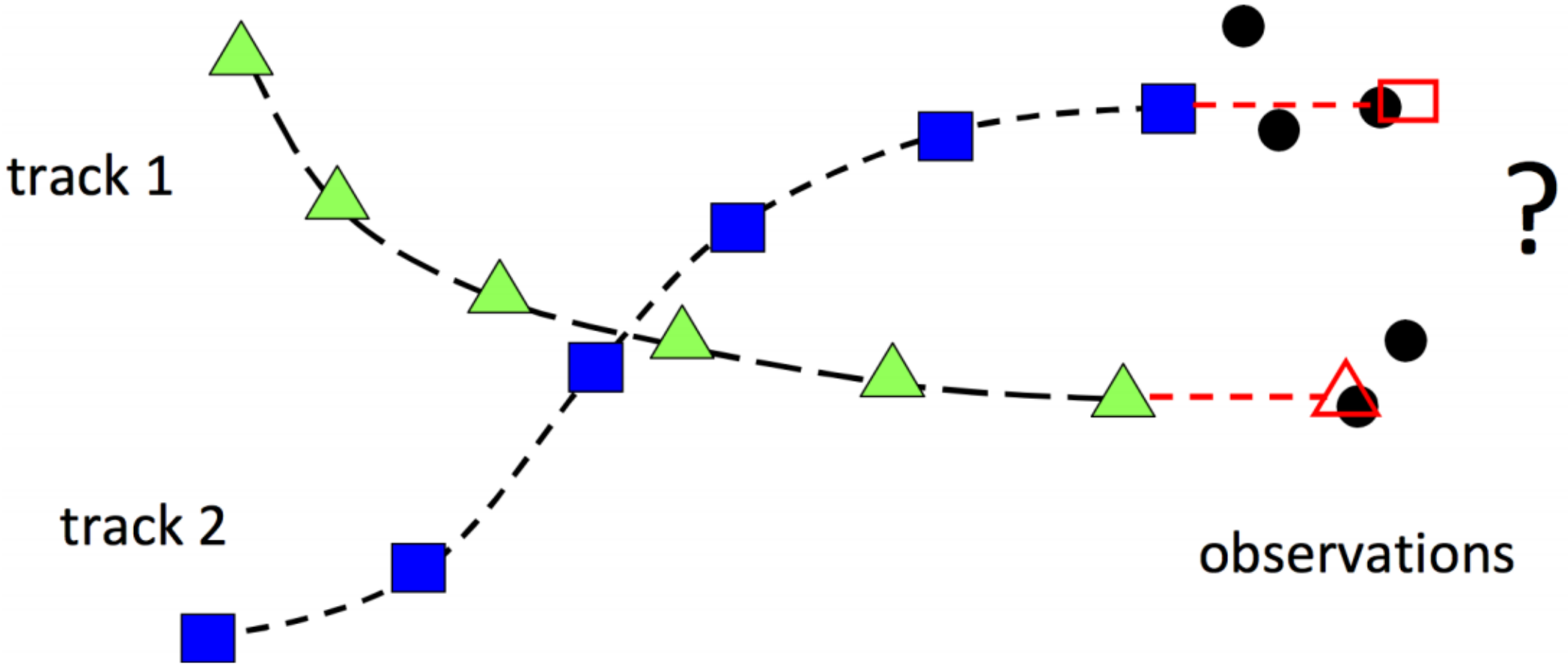
- Discrete combinatorial optimization



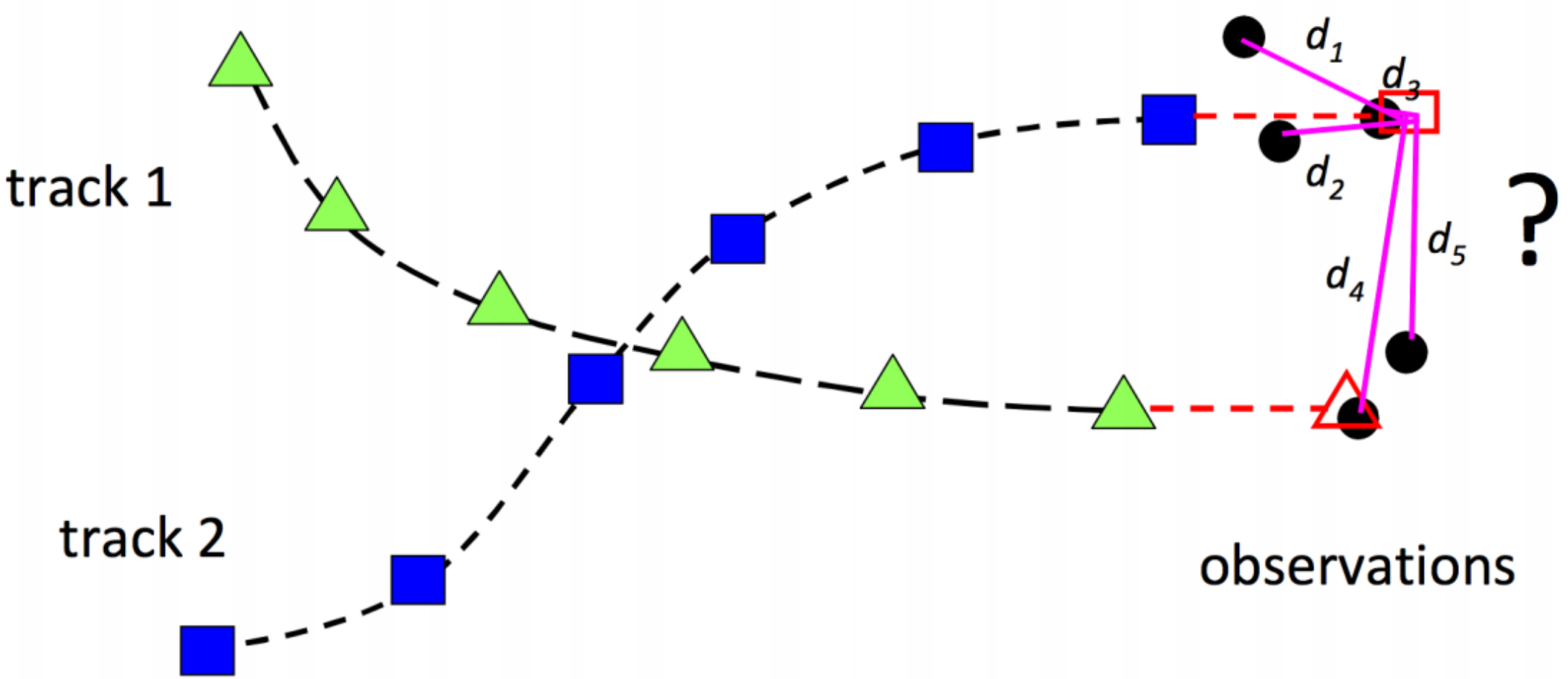
Object tracking – Multi target tracking



Object tracking – Multi target tracking



Object tracking – Multi target tracking



Object tracking – Multi target tracking

Global Nearest Neighbor Standard Filter (GNNSF)

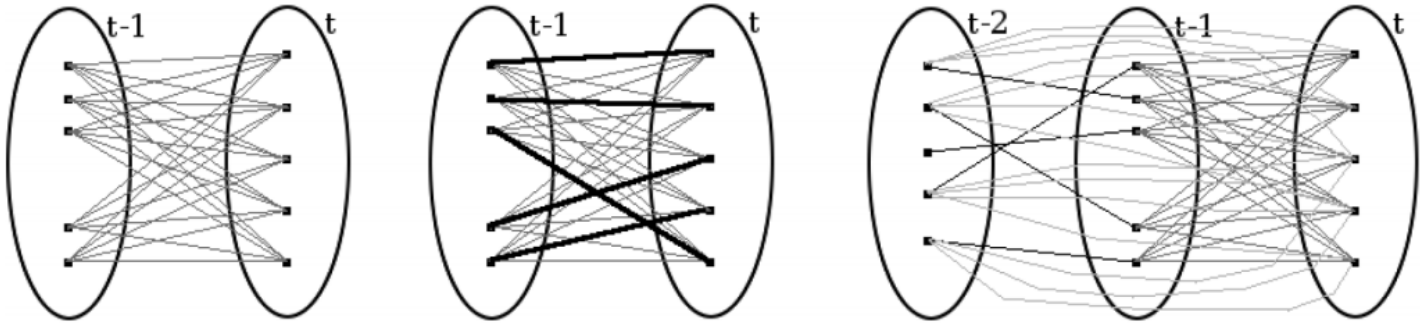
2D assignment problem (Bipartial matching problem)

$$\min_{x_{i,j}} \sum c_{i,j} x_{i,j}$$

$$s.t. \sum_{i:i>0} x_{i,j} = 1$$

$$\sum_{j:j>0} x_{i,j} = 1$$

$$x_{i,j} \in \{0, 1\}$$



- Hungarian method
- Auction method
- JVC method

Object tracking – Multi target tracking

- MCMC Data Association
- Network flow Data Association
- Discrete-Continuous Energy Minimization

https://www.youtube.com/watch?v=lZRzhZSDYKs

Discussion