Tracking II

04.06.2014

Announcements

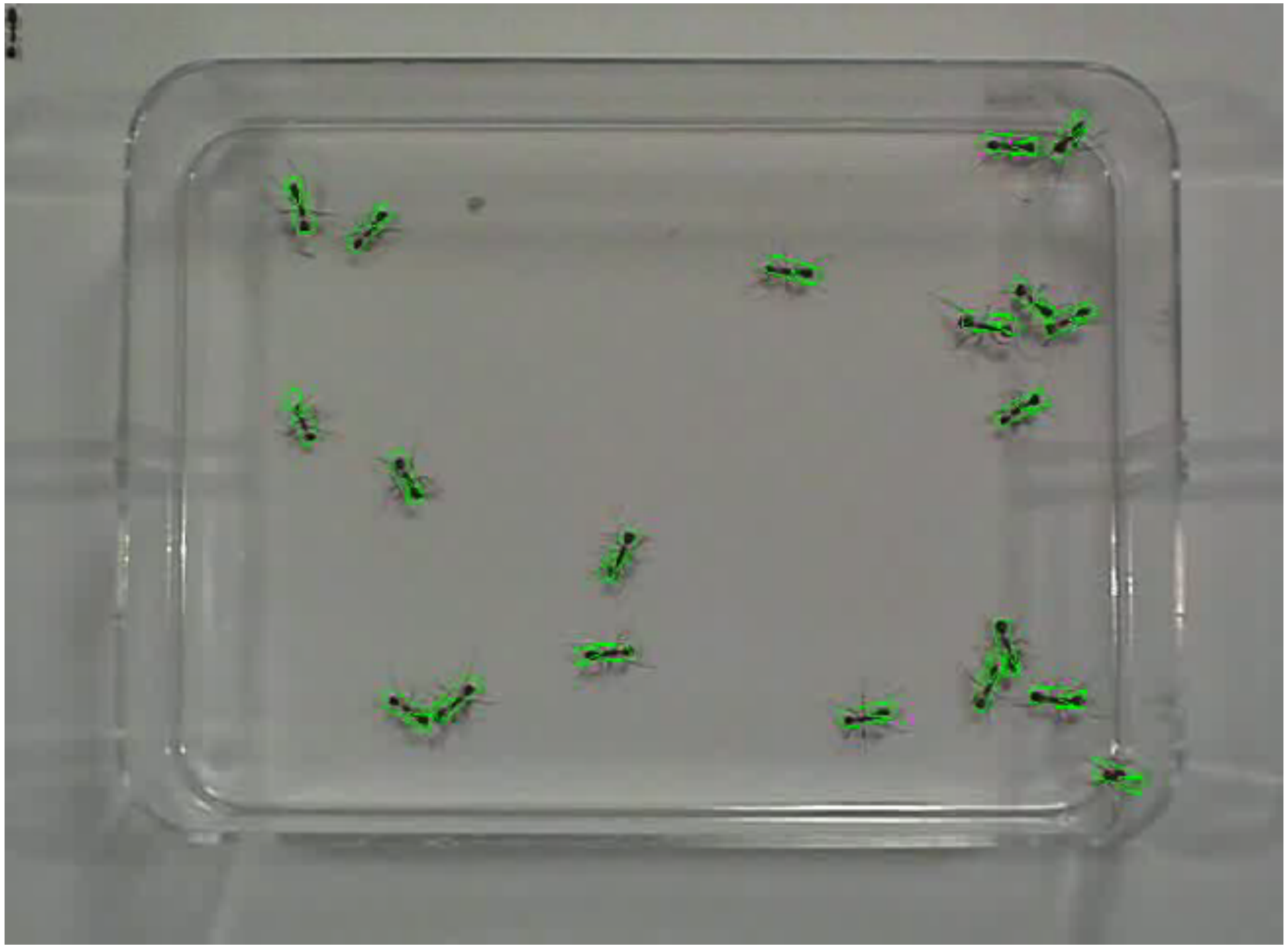
- Office hours today have to be cancelled

Tracking

- We typically distinguish 3 cases:
  - Tracking rigid objects
  - Tracking articulated objects, e.g. humans or animals
  - Tracking fully non-rigid objects
- Today:
  - Rigid objects
- Next time:
  - Tracking articulated objects, specifically humans

Example: "Ant Tracking"

- Tracking is, e.g., very useful for facilitating behavioral research in animals.



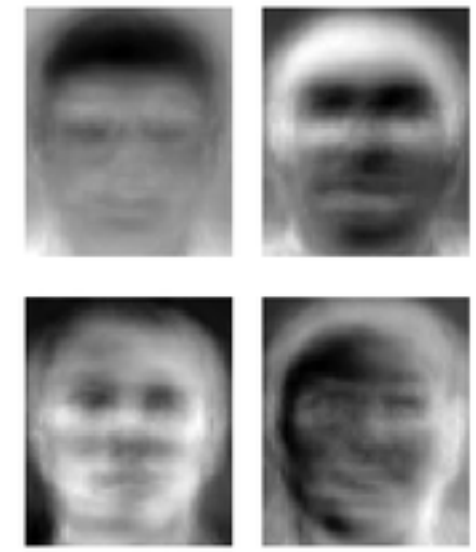
http://www.cc.gatech.edu/~borg/biotracking/recent-results.html

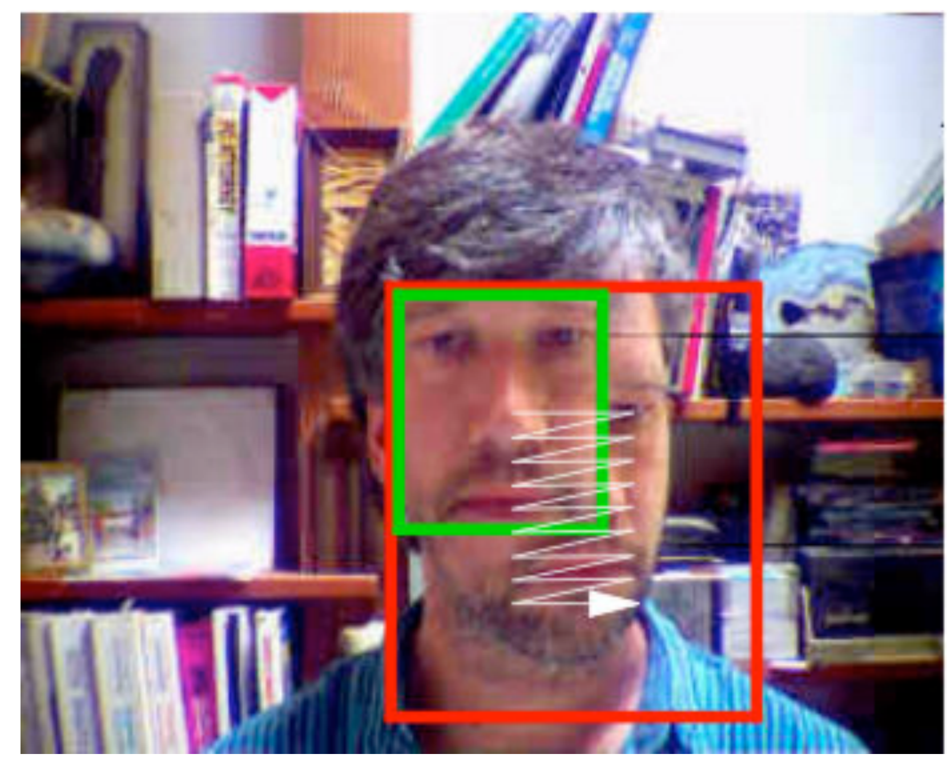
Challenges

- Fast motions
- Changing appearance
- Changing object pose
- Dynamic backgrounds
- Lots of clutter
- Occlusions
- Poor image quality
- ...

Simple Solution

- If the object we want to track is easy to find, we can detect it in every frame ("Tracking by detection").
- Example: Face tracking
  - If there is only one person in the scene and is always in a frontal view, we could use a PCA model of the face to find it.
  - But we may not want to search the entire image, because that is inefficient.
  - Use the position from the previous frame to constrain search.



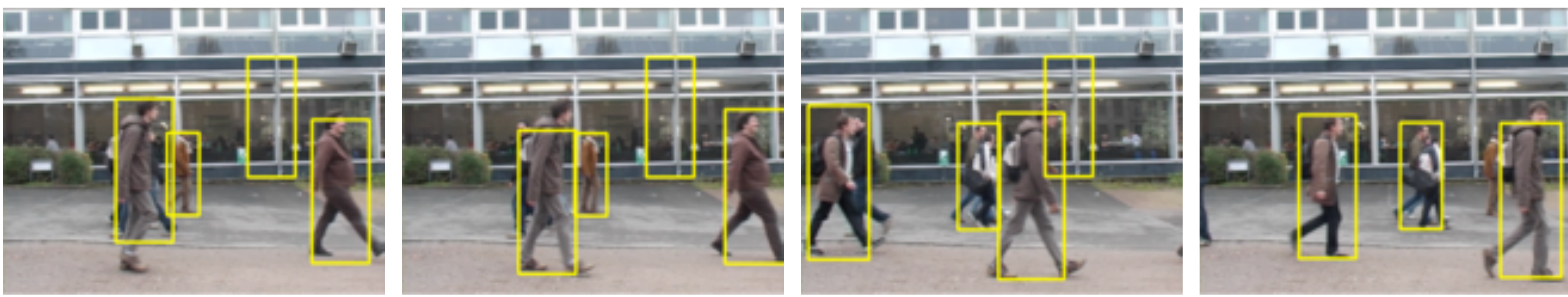


[IBM Vision Group]

Simple (?) Solution

TECHNISCHE UNIVERSITÄT DARMSTADT

- If the object we want to track is easy to find, we can just detect it in every frame (“Tracking by detection”).
- Problem: Need data association!
  - Many objects may be present & the detector may misfire

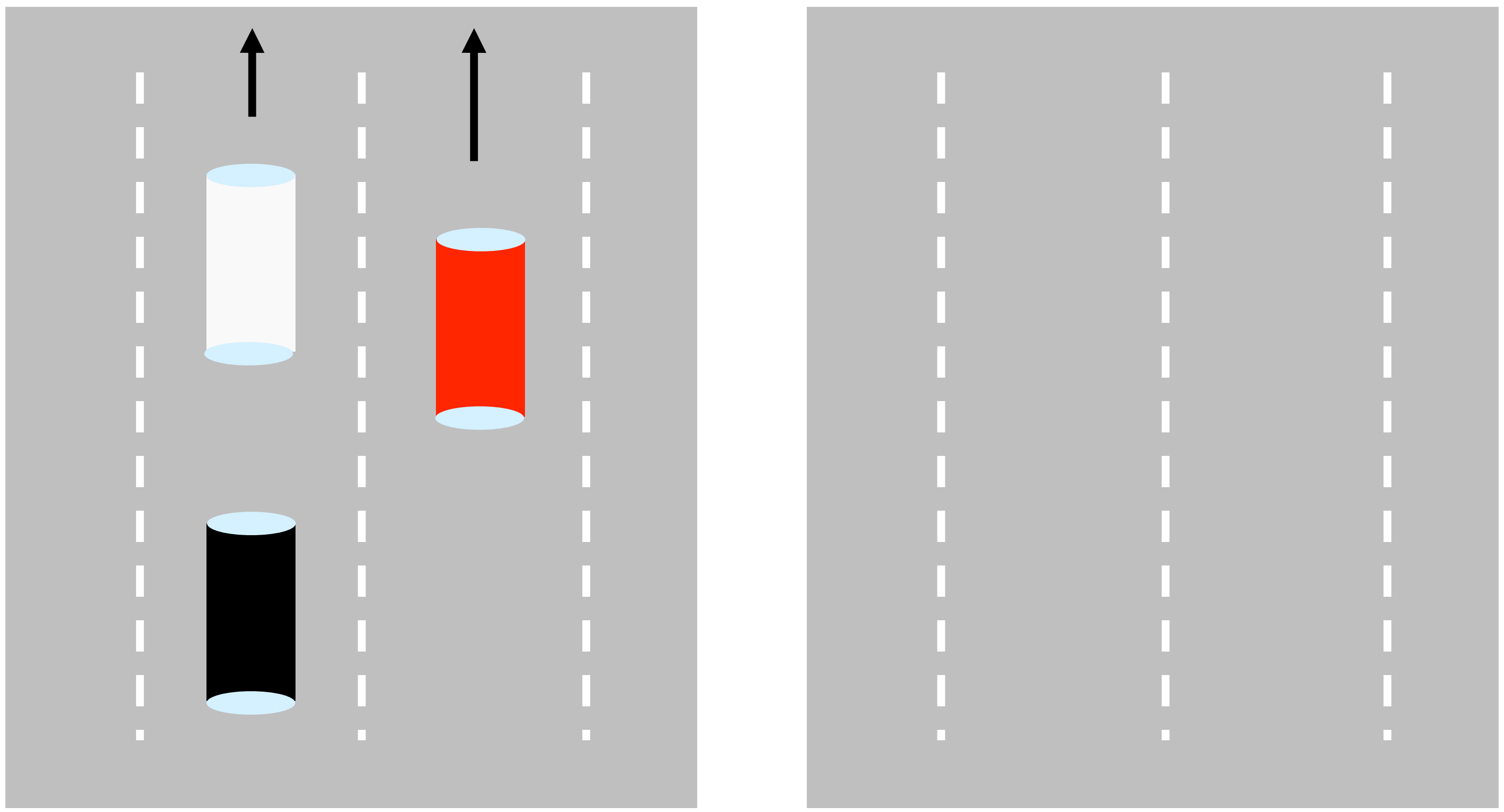


  - How do we know which detection is which?
  - In general, attempt data association
    - For now, sidestep this issue

Typical Models of Dynamics

- Constant position:
  - I.e. no real dynamics, but if the velocity of the object is sufficiently small, this can work.
- Constant velocity (possibly unknown):
  - We assume that the velocity does not change over time.
  - As long as the object does not quickly change velocity or direction, this is a quite reasonable model.
- Constant acceleration (possibly unknown):
  - Also captures the acceleration of the object.
  - This may include both the velocity, but also the directional acceleration.

Illustration

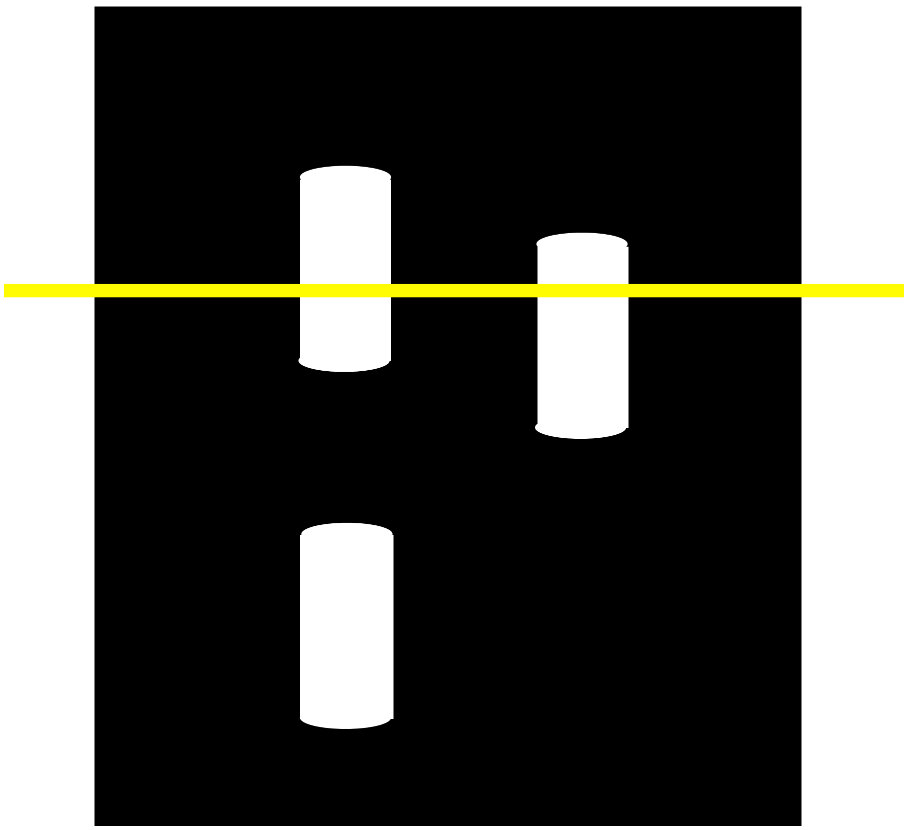


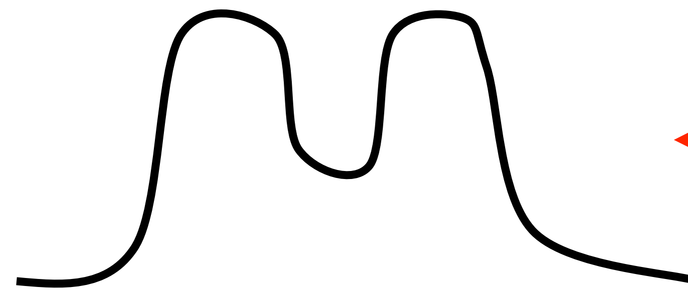
- Goal: Estimate car position at each time instant (say, of the red car).
- Observations: Image sequence and known background.

[Michael Black]

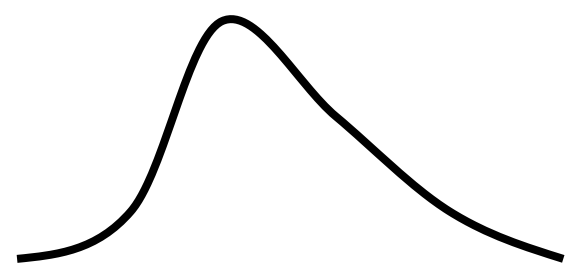
Bayesian Tracking

TECHNISCHE UNIVERSITÄT DARMSTADT

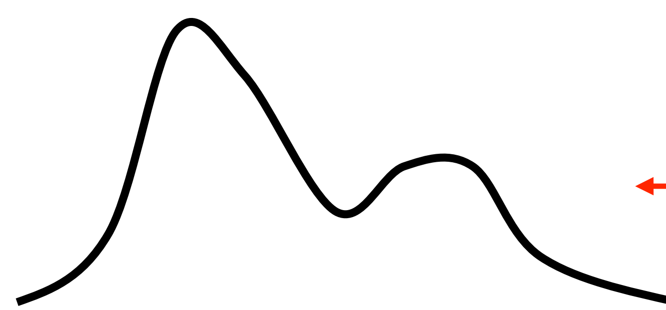




Likelihood:
noisy observation
$p(FG|\text{car} = (x, y))$



Prior:
$p(\text{car} = (x, y))$



Posterior:
Bayesian update
$p(\text{car} = (x, y)|FG)$

system state: car position

observations: images

[Michael Black]

Notation

- $\boldsymbol{x}_k \in \mathbb{R}^d$: internal state at k-th frame (hidden random variable, e.g., position of the object in the image).

  $\mathbf{X}_k = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k]^\mathrm{T}$: history up to time step k

- $\boldsymbol{z}_k \in \mathbb{R}^c$: measurement at k-th frame (observable random variable, e.g. the given image).

  $\mathbf{Z}_k = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k]^\mathrm{T}$: history up to time step k

[Michael Black]

Goal

Estimating the posterior probability $p(\boldsymbol{x}_k | \mathbf{Z}_k)$

How ???

One idea:
Recursion

$$p(\boldsymbol{x}_{k-1} | \mathbf{Z}_{k-1}) \quad \Rightarrow \quad p(\boldsymbol{x}_k | \mathbf{Z}_k)$$

- How to realize the recursion ?
- What assumptions are necessary ?

[Michael Black]

Recursive Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\begin{aligned}
& p(\boldsymbol{x}_k|\mathbf{Z}_k) \\
&= p(\boldsymbol{x}_k|\boldsymbol{z}_k, \mathbf{Z}_{k-1}) \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbf{Z}_{k-1}) \cdot p(\boldsymbol{x}_k|\mathbf{Z}_{k-1}) \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_k|\mathbf{Z}_{k-1}) \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1} \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{Z}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1} \\
&\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1}
\end{aligned}
$$

Bayes rule:

$$p(a|b) = p(b|a)p(a)/p(b)$$

Assumption:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbf{Z}_{k-1}) = p(\boldsymbol{z}_k|\boldsymbol{x}_k)$$

Marginalization:

$$p(a) = \int p(a,b) \, \mathrm{d}b$$

Assumption:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{Z}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

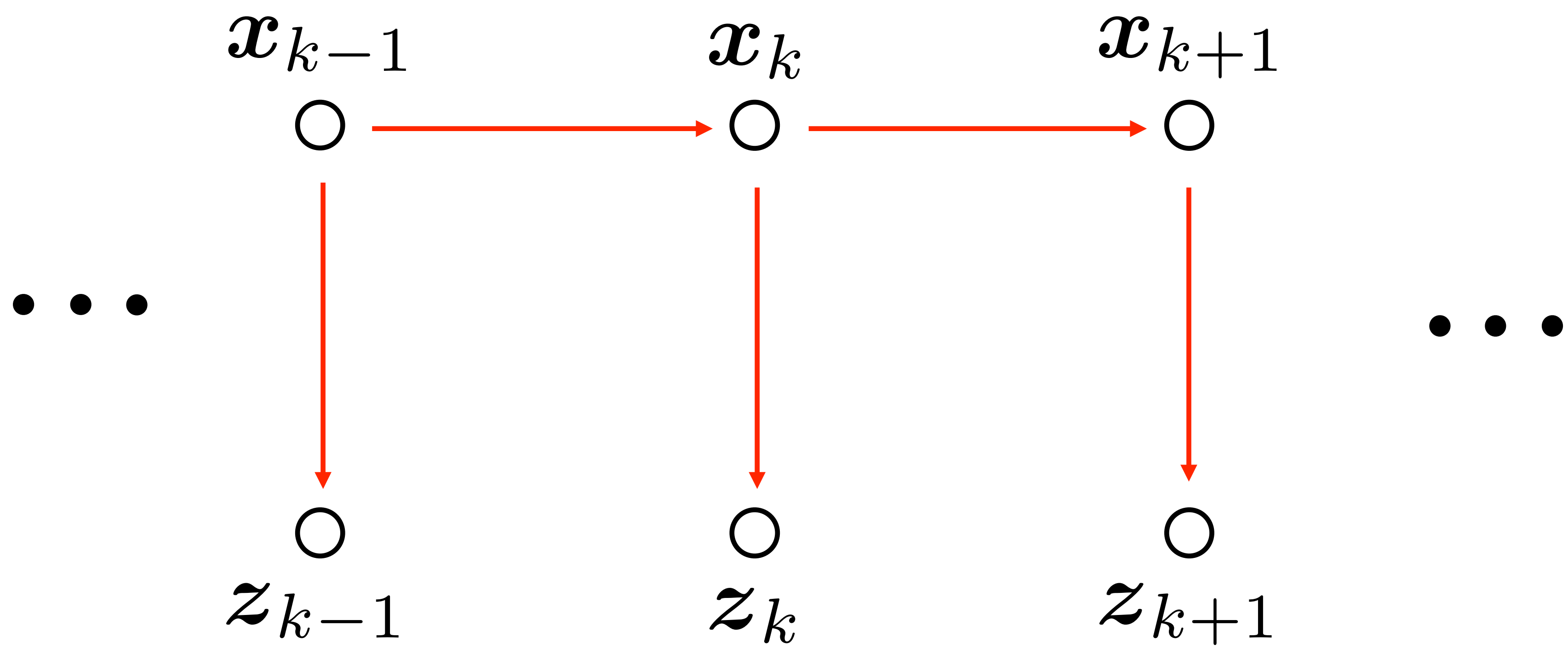
Bayesian Formulation

$$\boxed{p(\boldsymbol{x}_k | \mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k | \boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1} | \mathbf{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1}}$$

$p(\boldsymbol{x}_k \| \mathbf{Z}_k)$	posterior probability at current time step
$p(\boldsymbol{z}_k \| \boldsymbol{x}_k)$	likelihood
$p(\boldsymbol{x}_k \| \boldsymbol{x}_{k-1})$	temporal prior
$p(\boldsymbol{x}_{k-1} \| \mathbf{Z}_{k-1})$	posterior probability at previous time step
κ	normalizing term

Bayesian Graphical Model

- Hidden Markov model:



Assumptions:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k, \mathbf{Z}_{k-1}) = p(\boldsymbol{z}_k|\boldsymbol{x}_k) \qquad p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathbf{Z}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

$$p(\boldsymbol{x}_k|\mathbf{X}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

Dynamical Models Revisited

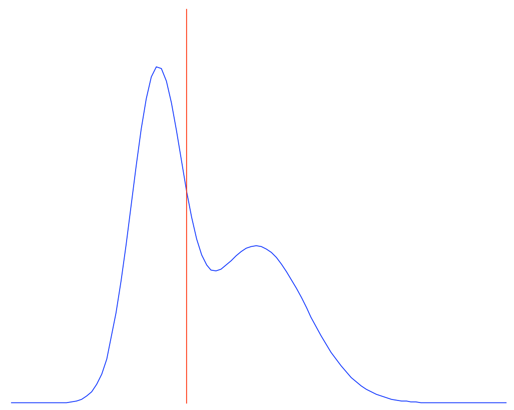
- If we make a first order Markov assumption $p(\boldsymbol{x}_k|\mathbf{X}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$, how can we still model the dynamics of the object?
    - Can we only model constant position or at most a assume a known drift of the object?
- Simple workarounds:
    - We let $\boldsymbol{x}_k$ represent the position of the object in frames $k-1$ and k. Then by comparing $\boldsymbol{x}_{k-1}$and $\boldsymbol{x}_k$, we can see of the velocity has changed.
    - Essentially equivalently, we can also let $\boldsymbol{x}_k$ represent both the position and the velocity (incl. direction) at frame k.

Estimators

Assume the posterior probability $p(\boldsymbol{x}_k|\mathbf{Z}_k)$ is known:

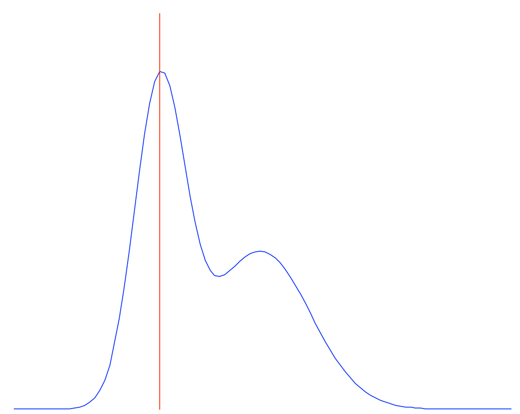
- posterior mean

$$\hat{\boldsymbol{x}}_k = E(\boldsymbol{x}_k|\mathbf{Z}_k)$$



- maximum a posteriori (MAP)

$$\hat{\boldsymbol{x}}_k = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\mathbf{Z}_k)$$



$p(\boldsymbol{x}_k|\mathbf{Z}_k)$

[Michael Black]

General Model

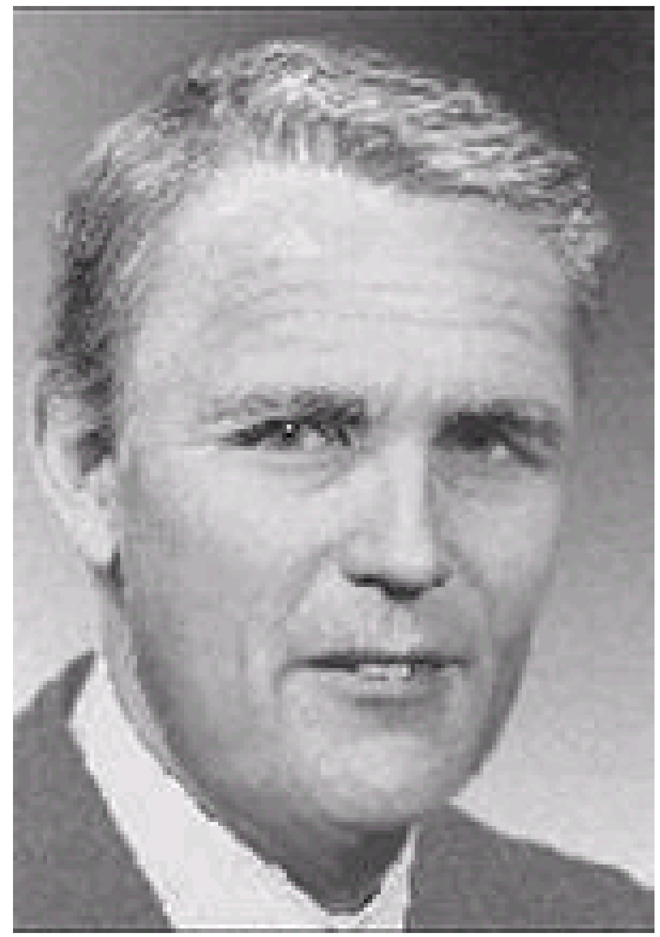
- $p(\boldsymbol{x}_k|\mathbf{Z}_k)$ can be an arbitrary, non-Gaussian, multi-modal distribution.
- The recursive equation often has no explicit solution, but can be numerically approximated using Monte Carlo techniques.
- If both likelihood and prior are Gaussian, the solution has closed form and the two estimators (posterior mean & MAP) are the same. Such a model is known as the **Kalman filter**. [Kalman, 1960]

[Michael Black]

Kalman Filter Introduction

- Slides from Wei Wu (formerly of Brown University, now at Florida State University)

Rudolph Emil Kalman



Born 1930 in Hungary

Professor Emeritus,
University of Florida

D.Sc., 1957,
Columbia University

His seminal paper:

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME, Journal of Basic Engineering*, *82*, 35–45

Broad Applications of Kalman Filter

- Engineering
  - Robotics, spacecraft, aircraft, automobiles
- Computer
  - Tracking, real-time graphics, computer vision
- Others
  - Forecasting economic indicators
  - Telephone and electricity loads
  - Encoding/decoding neural signals

Mathematical Properties of Kalman filter

- has a sound probabilistic framework
- makes explicit assumptions about the data and noise
- indicates the uncertainty of the estimate
- provides efficient estimation (closed-form and in real time)

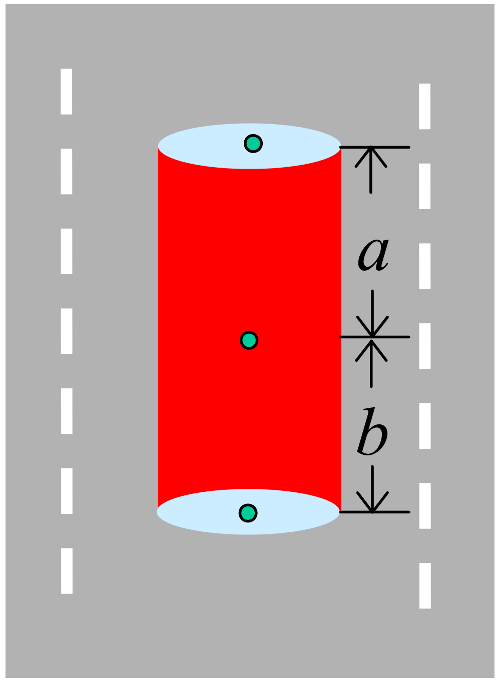
Likelihood Model

Generative model for the observation:

$$\boxed{\mathbf{z}_k = \mathbf{H}_k\ \mathbf{x}_k + \mathbf{q}_k}$$

$$\mathbf{H}_k \in \mathbf{R}^{c\times d}, \quad \mathbf{q}_k \sim N(0, \mathbf{Q}_k), \quad \mathbf{Q}_k \in \mathbf{R}^{c\times c}, \quad k = 1, 2, \cdots, M.$$

For example:



(x_k, y_k): centroid of car

(x_k^f, y_k^f): centroid of front bumper

(x_k^r, y_k^r): centroid of rear bumper

$$\begin{pmatrix} x_k^f \\ y_k^f \\ x_k^r \\ y_k^r \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & a \\ 1 & 0 & 0 \\ 0 & 1 & -b \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ 1 \end{pmatrix} + \begin{pmatrix} q_{k,1} \\ q_{k,2} \\ q_{k,3} \\ q_{k,4} \end{pmatrix}$$

Explicit Form

The likelihood model is equivalent to that

$$\mathbf{z}_k \sim N(\mathbf{H}_k\ \mathbf{x}_k, \mathbf{Q}_k)$$

The conditional probability has explicit form:

$$p(\mathbf{z}_k | \mathbf{x}_k) = \frac{1}{((2\pi)^c \det(\mathbf{Q}_k))^{1/2}} \exp(-\frac{1}{2}(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k)^T \mathbf{Q}_k^{-1}(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k))$$

Temporal Prior

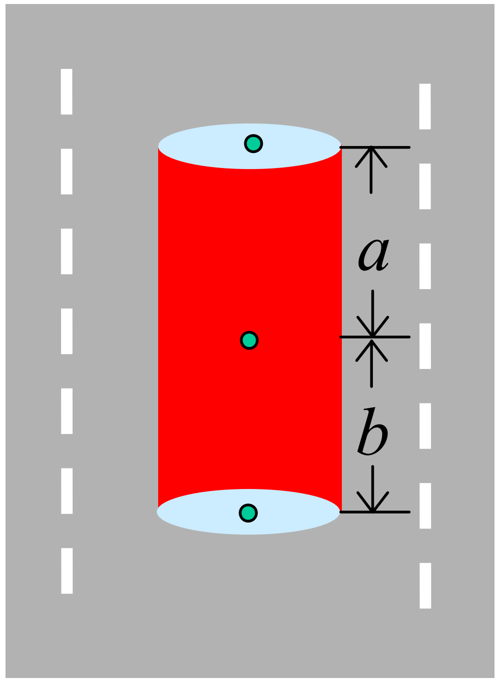
Temporal prior of the state:

$$\boxed{\mathbf{x}_k = \mathbf{A}_k \, \mathbf{x}_{k-1} + \mathbf{w}_k}$$

$$\mathbf{A}_k \in \mathbf{R}^{d\times d}, \quad \mathbf{w}_k \sim N(0, \mathbf{W}_k), \quad \mathbf{W}_k \in \mathbf{R}^{d\times d}, \quad k = 2,3,\cdots M.$$

Same example:

(x_k, y_k): centroid of car



Assume the velocity is constant (v_x, v_y),

$$\begin{pmatrix} x_k \\ y_k \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & v_x\Delta t \\ 0 & 1 & v_y\Delta t \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_{k-1} \\ y_{k-1} \\ 1 \end{pmatrix} + \begin{pmatrix} w_{k,1} \\ w_{k,2} \\ 0 \end{pmatrix}$$

Explicit Form

The prior model is equivalent to that

$$\mathbf{x}_{k+1} \sim N(\mathbf{A}_k\ \mathbf{x}_k, \mathbf{W}_k)$$

The conditional probability has explicit form:

$$p(\mathbf{x}_{k+1} | \mathbf{x}_k) = \frac{1}{((2\pi)^d \det(\mathbf{W}_k))^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}_{k+1} - \mathbf{A}_k \mathbf{x}_k)^T \mathbf{W}_k^{-1} (\mathbf{x}_{k+1} - \mathbf{A}_k \mathbf{x}_k))$$

Kalman Filter Model

Definition:

System Equation:

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k, \qquad \mathbf{w}_k \in N(0, \mathbf{W}_k)_{k=2,3,\cdots}$$

Measurement Equation:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k, \qquad \mathbf{q}_k \in N(0, \mathbf{Q}_k)_{k=1,2,\cdots}$$

Assumption:

All random variables have Gaussian distributions and they are linearly related

Learning Kalman Model

- In practice, the parameters in the model need to be estimated from training data. (In training data, we know both hidden states and measurements.)

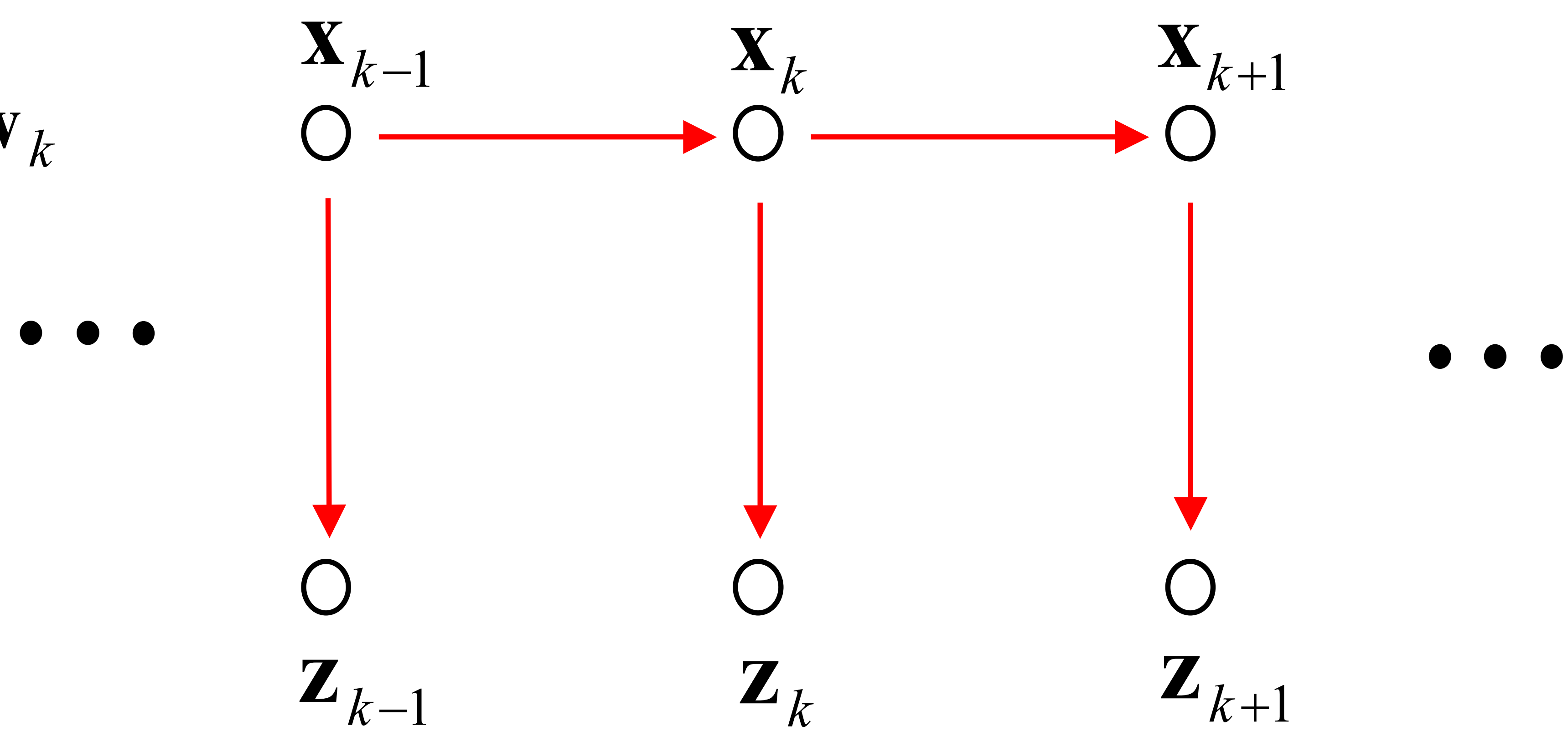
- Common simplification: $\mathbf{A}_k, \mathbf{H}_k, \mathbf{W}_k, \mathbf{Q}_k$ are constant over time (independent of *k*).

- The **A, H, W, Q** can be estimated by maximizing the joint probability $p(\mathbf{X}_M, \mathbf{Z}_M)$.

Bayesian Graphical Model

$$\mathbf{x}_k = \mathbf{A}_k \, \mathbf{x}_{k-1} + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H}_k \, \mathbf{x}_k + \mathbf{q}_k$$



$$p(\mathbf{X}_M, \mathbf{Z}_M) = p(\mathbf{X}_M)\, p(\mathbf{Z}_M \mid \mathbf{X}_M)$$

$$= [\, p(\mathbf{x}_1) \prod_{k=2}^{M} p(\mathbf{x}_k | \mathbf{x}_{k-1})][\prod_{k=1}^{M} p(\mathbf{z}_k | \mathbf{x}_k)]$$

Splitting the Joint Distribution

Define: argmax & argmin

$$\hat{x} = \arg\max_x f(x) \Leftrightarrow f(\hat{x}) = \max f(x).$$

$$\arg\max_{A,W,H,Q} p(\mathbf{X}_M, \mathbf{Z}_M)$$

$$= \arg\max_{A,W} p(\mathbf{X}_M) \arg\max_{H,Q} p(\mathbf{Z}_M | \mathbf{X}_M)$$

$$= \arg\min_{A,W} f(\mathbf{A}, \mathbf{W}) \arg\min_{H,Q} g(\mathbf{H}, \mathbf{Q})$$

where

$$f(\mathbf{A}, \mathbf{W}) = -\alpha \log p(\mathbf{X}_M) = \sum_{k=2}^{M} [\log(\det \mathbf{W}) + (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{W}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})],$$

$$g(\mathbf{H}, \mathbf{Q}) = -\beta \log p(\mathbf{Z}_M | \mathbf{X}_M) = \sum_{k=1}^{M} [\log(\det \mathbf{Q}) + (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)^T \mathbf{Q}^{-1} (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)].$$

How to optimize functions with matrix variables ???

Matrix Calculus

Define: assume $\mathbf{X} = (x_{ij})_{mn}$, then

$$d/d\mathbf{X} = \begin{pmatrix} d/dx_{11} & \cdots & d/dx_{1n} \\ \vdots & \ddots & \vdots \\ d/dx_{m1} & \cdots & d/dx_{mn} \end{pmatrix}$$

- Quadratic Products:

$$d/d\mathbf{X}\ ((\mathbf{Xa+b})^T\mathbf{C}(\mathbf{Xa+b})) = (\mathbf{C}+\mathbf{C}^T)(\mathbf{Xa+b})\mathbf{a}^T$$

- Determinant: $d/d\mathbf{X}\ (\log(\det(\mathbf{X}))) = \mathbf{X}^{-T}$
- Inverse: $d/d\mathbf{X}\ (\mathbf{a}^T\mathbf{X}^{-1}\mathbf{b}) = -\mathbf{X}^{-T}\mathbf{ab}^T\mathbf{X}^{-T}$

(upper case: matrix, lower case: column vector)

http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html

Detailed Steps

Show one example on prior probability:

$$f(\mathbf{A},\mathbf{W}) = \sum_{k=2}^{M} [\log(\det \mathbf{W}) + (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{W}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})]$$

i) $$\frac{\partial}{\partial \mathbf{A}} f(\mathbf{A},\mathbf{W}) = \sum_{k=2}^{M} (\mathbf{W}^{-1} + \mathbf{W}^{-\mathrm{T}})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(-\mathbf{x}_{k-1})^T = 0$$

$$\Rightarrow \sum_{k=2}^{M} \mathbf{x}_k \mathbf{x}_{k-1}^T = \mathbf{A} \sum_{k=2}^{M} \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T \quad \Rightarrow \mathbf{A} = \sum_{k=2}^{M} \mathbf{x}_k \mathbf{x}_{k-1}^T \left(\sum_{k=2}^{M} \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T \right)^{-1}$$

ii) $$\frac{\partial}{\partial \mathbf{W}} f(\mathbf{A},\mathbf{W}) = \sum_{k=2}^{M} (\mathbf{W}^{-1} - \mathbf{W}^{-1} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T \mathbf{W}^{-1}) = 0$$

$$\Rightarrow \mathbf{W} = \sum_{k=2}^{M} (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^T / (M-1)$$

Closed-form Solutions:

$$\mathbf{A} = \left(\sum_{k=2}^{M} \mathbf{x}_k \mathbf{x}_{k-1}^T\right)\left(\sum_{k=2}^{M} \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T\right)^{-1},$$

$$\mathbf{W} = \frac{1}{M-1}\left(\sum_{k=2}^{M} \mathbf{x}_k \mathbf{x}_k^T - \mathbf{A}\sum_{k=2}^{M} \mathbf{x}_{k-1} \mathbf{x}_k^T\right),$$

$$\mathbf{H} = \left(\sum_{k=1}^{M} \mathbf{z}_k \mathbf{x}_k^T\right)\left(\sum_{k=1}^{M} \mathbf{x}_k \mathbf{x}_k^T\right)^{-1},$$

$$\mathbf{Q} = \frac{1}{M}\left(\sum_{k=1}^{M} \mathbf{z}_k \mathbf{z}_k^T - \mathbf{H}\sum_{k=1}^{M} \mathbf{x}_k \mathbf{z}_k^T\right).$$

Recursive Estimation

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \kappa\ p(\mathbf{z}_k|\mathbf{x}_k)\int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1}$$

Time update:

posterior at previous step:

$$p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$$

temporal prior:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1})$$

prior distribution:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1}$$

Measurement update:

prior distribution:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1})$$

likelihood:

$$p(\mathbf{z}_k|\mathbf{x}_k)$$

posterior distribution:

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \kappa\ p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})$$

O. L. R. Jacobs, *Introduction to Control Theory, 2nd Edition*, 1993

Basic Properties of Gaussian Distribution

$\mathbf{x}, \mathbf{y} \in \mathbf{R}^d$ independent random vectors, $\mathbf{x} \sim N(0, \mathbf{A})$, $\mathbf{y} \sim N(0, \mathbf{B})$,

then, i) for any matrix $\mathbf{C} \in \mathbf{R}^{d \times d}$, $\mathbf{Cx} \sim N(0, \mathbf{CAC}^T)$;

ii) $\mathbf{x} + \mathbf{y} \sim N(0, \mathbf{A} + \mathbf{B})$.

Main ideas in the proof:

i)
$$
\begin{aligned}
p(\mathbf{x}) &\propto \exp(-\frac{1}{2}\mathbf{x}^T \mathbf{A}^{-1}\mathbf{x}) \\
&\propto \exp(-\frac{1}{2}(\mathbf{x}^T \mathbf{C}^T)(\mathbf{C}^{-T}\mathbf{A}^{-1}\mathbf{C}^{-1})(\mathbf{Cx})) \\
&\propto \exp(-\frac{1}{2}(\mathbf{Cx})^T(\mathbf{CAC}^T)^{-1}(\mathbf{Cx})).
\end{aligned}
$$

$\boxed{(\mathbf{uv})^T = \mathbf{v}^T\mathbf{u}^T \ \& \ (\mathbf{uv})^{-1} = \mathbf{v}^{-1}\mathbf{u}^{-1}}$

ii) $Cov(\mathbf{x} + \mathbf{y}) = Cov(\mathbf{x}) + Cov(\mathbf{y}) = \mathbf{A} + \mathbf{B}$.

Kalman Filtering
Step I: Time Update

Assume: $\mathbf{x}_{k-1} \,|\, \mathbf{Z}_{k-1} \sim N(\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})$

$$\Leftrightarrow \quad \mathbf{x}_{k\text{-}1} = \hat{\mathbf{x}}_{k-1} + \mathbf{e}_{k-1}, \;\; \mathbf{e}_{k-1} \sim N(0, \mathbf{P}_{k-1})$$

System equation: $\mathbf{x}_k = \mathbf{A}_k \, \mathbf{x}_{k-1} + \mathbf{w}_k, \;\; \mathbf{w}_k \sim N(0, \mathbf{W}_k)$

$$\Rightarrow \quad \mathbf{x}_k = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} + \mathbf{A}_k \mathbf{e}_{k-1} + \mathbf{w}_k$$

Use properties i) and ii):

$$\mathbf{A}_k \mathbf{e}_{k\text{-}1} + \mathbf{w}_k \sim N(0, \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k)$$

Let

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k\text{-}1}, \;\; \mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k,$$

then,

$$\mathbf{x}_k \,|\, \mathbf{Z}_{k-1} \sim N(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-)$$

Kalman Filtering
Step II: Measurement Update

Time update:

$$p(\mathbf{x}_k \mid \mathbf{Z}_{k-1}) \propto \exp(-\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{P}_k^-)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-))$$

Measurement equation:

$$p(\mathbf{z}_k \big| \mathbf{x}_k) \propto \exp(-\frac{1}{2}(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k)^T \mathbf{Q}_k^{-1}(\mathbf{z}_k - \mathbf{H}_k \mathbf{x}_k))$$

Recursive update:

(details omitted)

$$p(\mathbf{x}_k \mid \mathbf{Z}_k) \propto p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{Z}_{k-1})$$

$$\propto \exp(-\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T (\mathbf{P}_k)^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k))$$

where, $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$, $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-$,

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{Q}_k)^{-1}.$$

That is,

$$\mathbf{x}_k \mid \mathbf{Z}_k \sim N(\hat{\mathbf{x}}_k, \mathbf{P}_k)$$

Kalman Filter Algorithm

Time Update

Prior estimate

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k\text{-}1}$$

Error covariance

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k$$

Measurement Update

Posterior estimate

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$$

Error covariance

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{Q}_k)^{-1}$$

previous estimate of $\hat{\mathbf{x}}_{k-1}$ and $\mathbf{P}_{k-1}$

Welch & Bishop, *An Introduction to the Kalman Filter*, 2002

Conclusion

- The Kalman filter is an extremely useful technique and is a fundamental and important tool in stochastic control theory.
- It is regularly rediscovered, and appears in different guises in different fields.
- It can be generalized to various non-linear and/or non-Gaussian models: extended Kalman filter, unscented Kalman filter, switching Kalman filter, mixture Kalman filter, particle filter, and so on.

Kalman Filtering in Action

- Some example I found on the web:

http://www.youtube.com/watch?v=U1L0X4cts8o

Does the Kalman filter address our challenges?

- Fast motions:
  - Yes, we can adapt the motion prior to our needs (within the extent of the Gaussian modeling assumption).
- Changing appearance:
  - To some extent. We can include the appearance as part of the state variable.
- Changing object pose:
  - Not really. The fact that the model is linear makes it difficult to model different poses.
- …

Restrictions

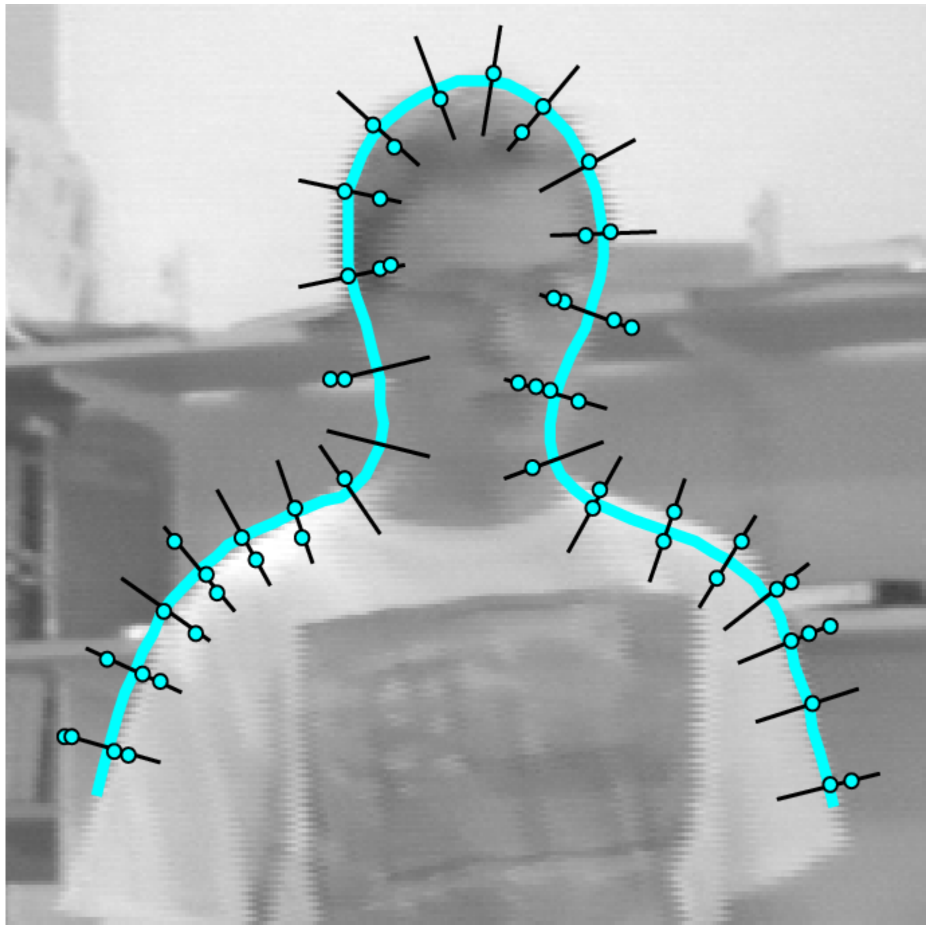
- The important restrictions of the Kalman filter are that it assumes linear state and output transformations, as well as Gaussian noise.
- There are many cases where this is inappropriate.
- Next: Particle filtering
  - A much more general recursive estimation technique.
  - But also computationally much harder, and tricky to implement correctly…

Particle Filtering

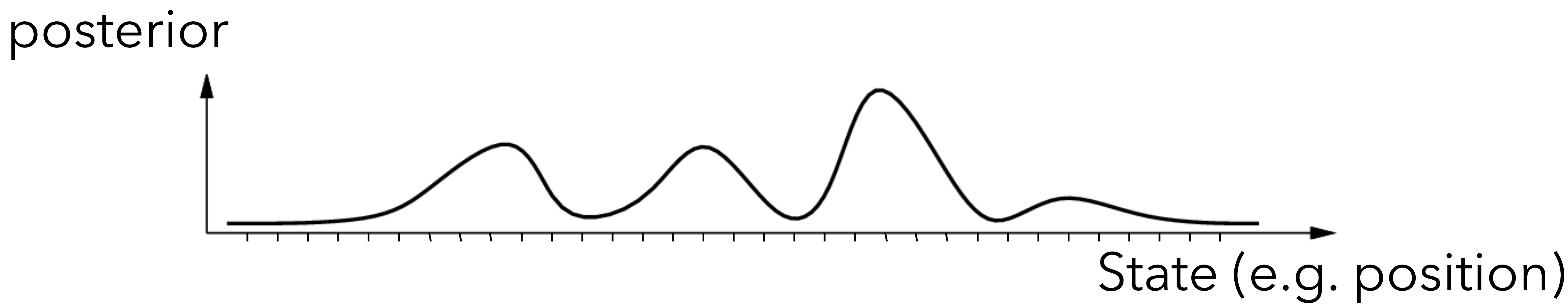
04.06.2014

Multi-Modal Posteriors

- Measurement clutter in natural images causes likelihood functions to have multiple, local maxima.
  - In a particular frame, the observation may be poor so that there are multiple promising looking locations.
  - We cannot resolve these ambiguities until we have seen more data (additional frames).
- To do that, we have to allow for the posterior at each frame to be multi-modal.



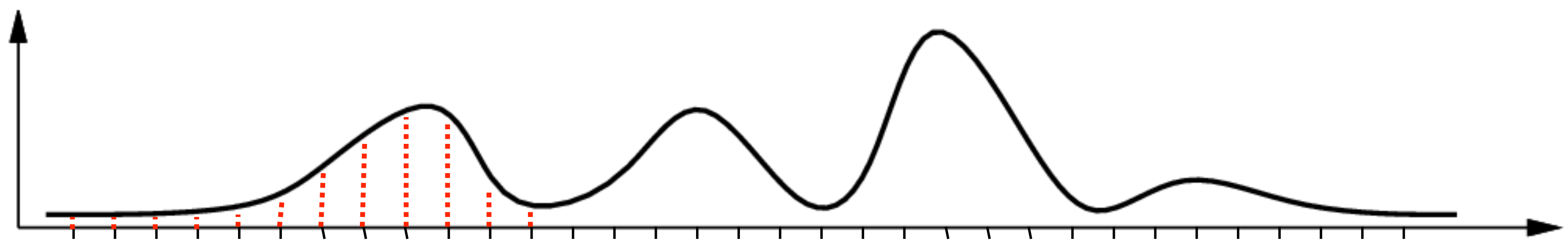
Multi-Modal Posteriors



- How can we represent the posterior at each time step in a flexible way that allows for:
  - Multiple modes
    - To encode multiple promising locations.
  - Varying number of modes
    - Modes may appear and disappear again when they are ruled out.

Non-Parametric Approximation

- We could sample at regular intervals.



- Instead of representing a continuous function, we approximate it using a **discrete set of samples** (or particles) each of which has a **weight**.

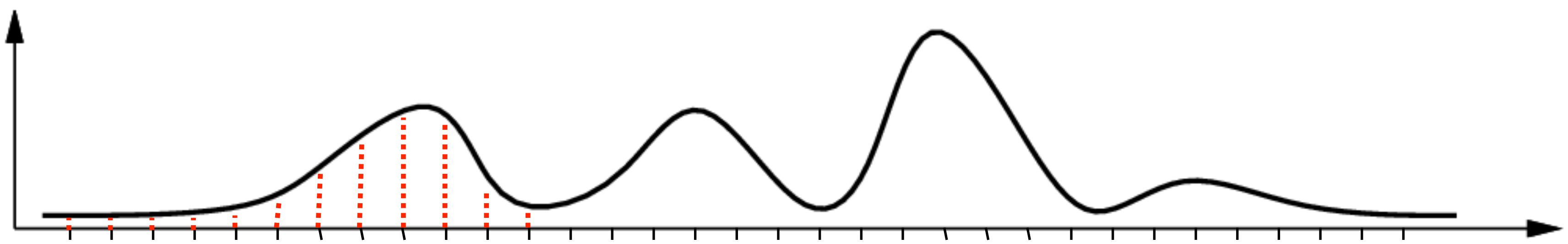
$$S = \{(\boldsymbol{x}^{(i)}, w^{(i)});\ i = 1, \ldots, N\}$$

- We usually use normalized weights: $$\sum_{i=1}^{N} w^{(i)} = 1$$

Non-Parametric Approximation

TECHNISCHE UNIVERSITÄT DARMSTADT

- We could sample at regular intervals.



Since there is no underlying parametric form, we call this a non-parametric representation or approximation.

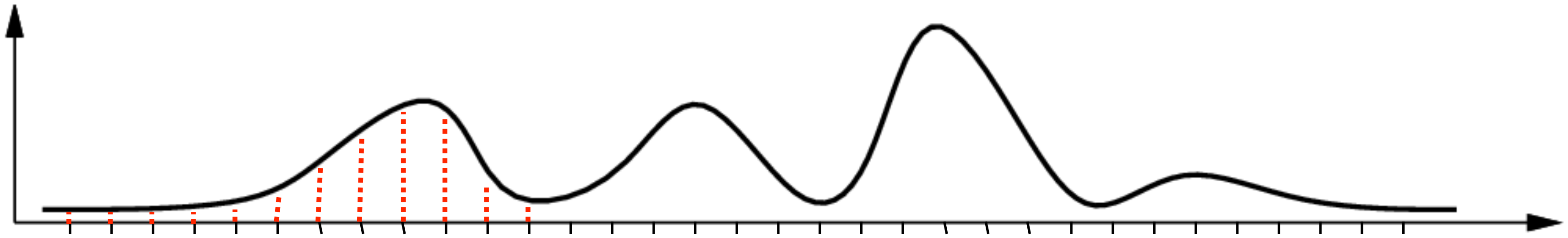
- If needed, we can convert this back to a continuous density by assuming that each sample is represented by a small Gaussian mixture component:

$$\tilde{p}(\boldsymbol{x}) = \sum_i w^{(i)} \mathcal{N}(\boldsymbol{x};\ \boldsymbol{x}^{(i)}, \sigma^2)$$

  - Note though that this is normally not necessary!

Non-Parametric Approximation

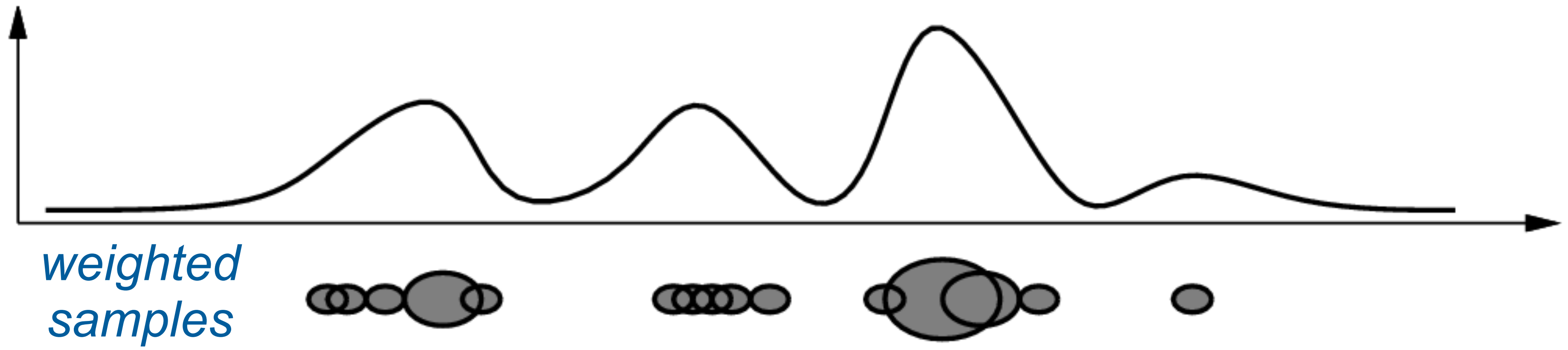
- We could sample at regular intervals.



- Problems?
  - Most samples have low weight – wasted computation.
  - How finely do we need to discretize?
  - High dimensional space – discretization impractical.

Non-Parametric Approximation

- Idea: Sample at irregular intervals and (optionally) weigh samples.



- Weighted samples: $$S = \left\{(\boldsymbol{x}^{(i)}, w^{(i)});\ i = 1, \ldots, N\right\}$$
- Normalized weights $$\sum_{i=1}^{N} w^{(i)} = 1$$

How does this help us?

- Remember the filtering recursion:

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\,\mathrm{d}\boldsymbol{x}_{k-1}$$

- We need to be able to compute integrals of the type:

$$\int f(\boldsymbol{x}) \cdot p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

- **Monte-Carlo approximation:**

$$\int f(\boldsymbol{x}) \cdot p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \sum_i f(\boldsymbol{x}^{(i)}), \qquad \boldsymbol{x}^{(i)} \sim p(\boldsymbol{x})$$

Monte-Carlo Approximation

$$\int f(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \approx \sum_i f(\boldsymbol{x}^{(i)}), \qquad \boldsymbol{x}^{(i)} \sim p(\boldsymbol{x})$$

- In other terms, the $\boldsymbol{x}^{(i)}$ are a sample representation of the density $p(\boldsymbol{x})$.
- What if we have a weighted sample representation?
  - Just as easy...

$$\int f(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \approx \sum_i w^{(i)} f(\boldsymbol{x}^{(i)})$$

  - Note however that in these cases the $\boldsymbol{x}^{(i)}$ are usually not the same as before.

Filtering Step-by-Step

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \,\mathrm{d}\boldsymbol{x}_{k-1}$$

- We start with assuming that we have a weighted sample representation for the posterior $p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})$ at the previous time step:

$$S_{k-1} = \{(\boldsymbol{x}_{k-1}^{(i)}, w_{k-1}^{(i)});\ i = 1, \ldots, N\}$$

Filtering Step-by-Step

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}$$

- We start with assuming that we have a weighted sample representation for the posterior $p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})$ at the previous time step:

$$S_{k-1} = \{(\boldsymbol{x}_{k-1}^{(i)}, w_{k-1}^{(i)});\ i = 1, \ldots, N\}$$

- Use this to carry out Monte-Carlo integration:

$$\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1} \approx \sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$$

Filtering Step-by-Step

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \boxed{\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}}$$

- Represent the approximation $\sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ again using another particle set:

$$\hat{S}_{k-1} = \{(\hat{\boldsymbol{x}}_{k-1}^{(i)}, \hat{w}_{k-1}^{(i)});\ i = 1, \ldots, N\}$$

Filtering Step-by-Step

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})\, \mathrm{d}\boldsymbol{x}_{k-1}$$

- Represent the approximation $\sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ again using another particle set:

$$\hat{S}_{k-1} = \{(\hat{\boldsymbol{x}}_{k-1}^{(i)}, \hat{w}_{k-1}^{(i)});\ i = 1, \ldots, N\}$$

- Take into account the likelihood by re-weighting the particles:

$$\begin{aligned} \boldsymbol{x}_k^{(i)} &= \hat{\boldsymbol{x}}_{k-1}^{(i)} \\ w_k^{(i)} &= p(\boldsymbol{z}_k|\hat{\boldsymbol{x}}_{k-1}^{(i)})\hat{w}_{k-1}^{(i)} \\ S_k &= \{(\boldsymbol{x}_k^{(i)}, w_k^{(i)});\ 1, \ldots, N\} \end{aligned}$$

Filtering Step-by-Step

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \,\mathrm{d}\boldsymbol{x}_{k-1}$$

- We obtain a weighted sample representation for the posterior $p(\boldsymbol{x}_k|\mathbf{Z}_k)$ at the current time step:

$$S_k = \{(\boldsymbol{x}_k^{(i)}, w_k^{(i)});\ 1, \ldots, N\}$$

Filtering Step-by-Step

$$p(\boldsymbol{x}_k|\mathbf{Z}_k) = \kappa \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \cdot p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1}) \, \mathrm{d}\boldsymbol{x}_{k-1}$$

- We obtain a weighted sample representation for the posterior $p(\boldsymbol{x}_k|\mathbf{Z}_k)$ at the current time step:

$$S_k = \left\{(\boldsymbol{x}_k^{(i)}, w_k^{(i)});\ 1, \ldots, N\right\}$$

- Remaining question: How do we represent $\sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ using a sample set?

Temporal Propagation

$$\sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(i)})$$

- The simplest way to deal with the problem of representing the result of the Monte-Carlo integration is to propagate each sample independently according to the temporal dynamics and keeping the weight:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k-1}^{(i)} &\sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(i)}) \\ \hat{w}_{k-1}^{(i)} &= w_{k-1}^{(i)} \end{aligned}$$

- Problem: Sample impoverishment
  - Solution: Resampling

Resampling

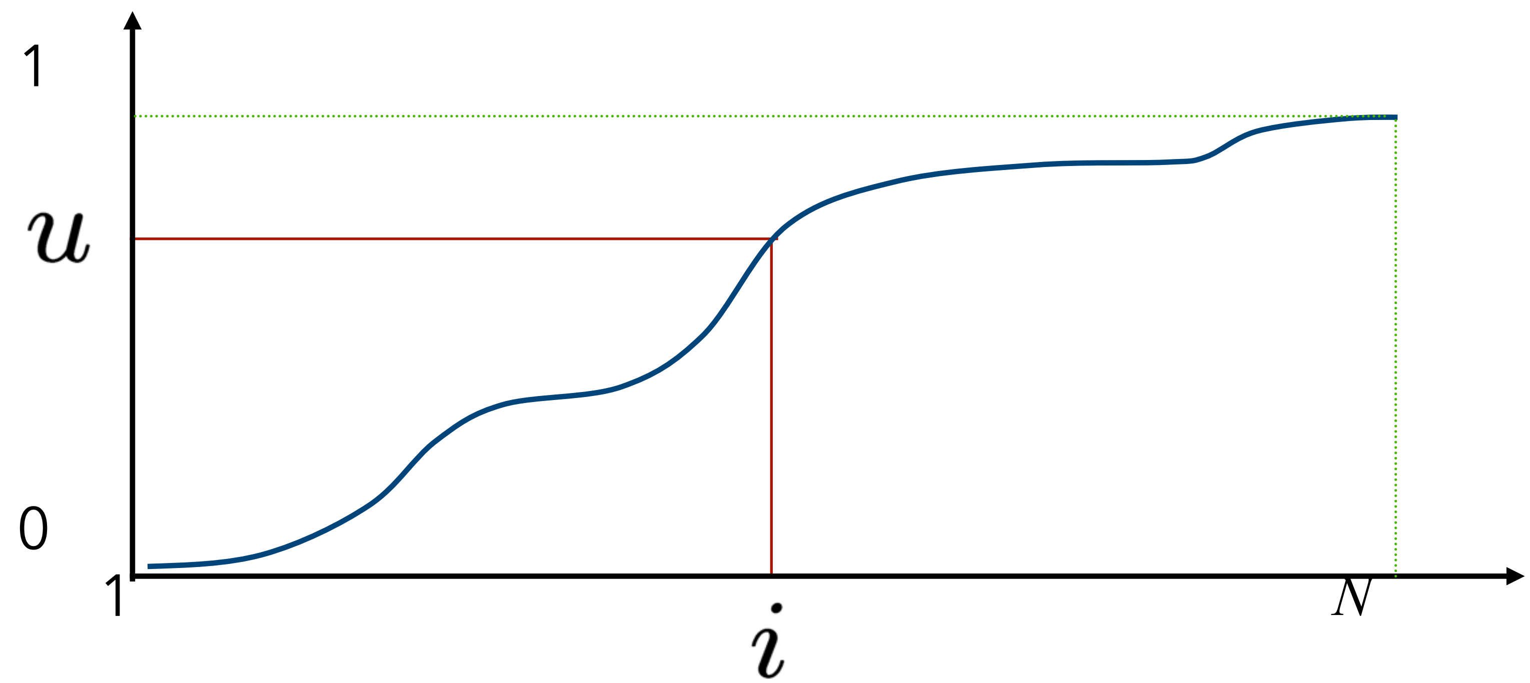
- Given weighted sample set

$$S_{k-1} = \{(\boldsymbol{x}_{k-1}^{(i)}, w_{k-1}^{(i)});\ i = 1, \ldots, N\}$$

- Draw unweighted samples by sampling from the weight distribution:

sample

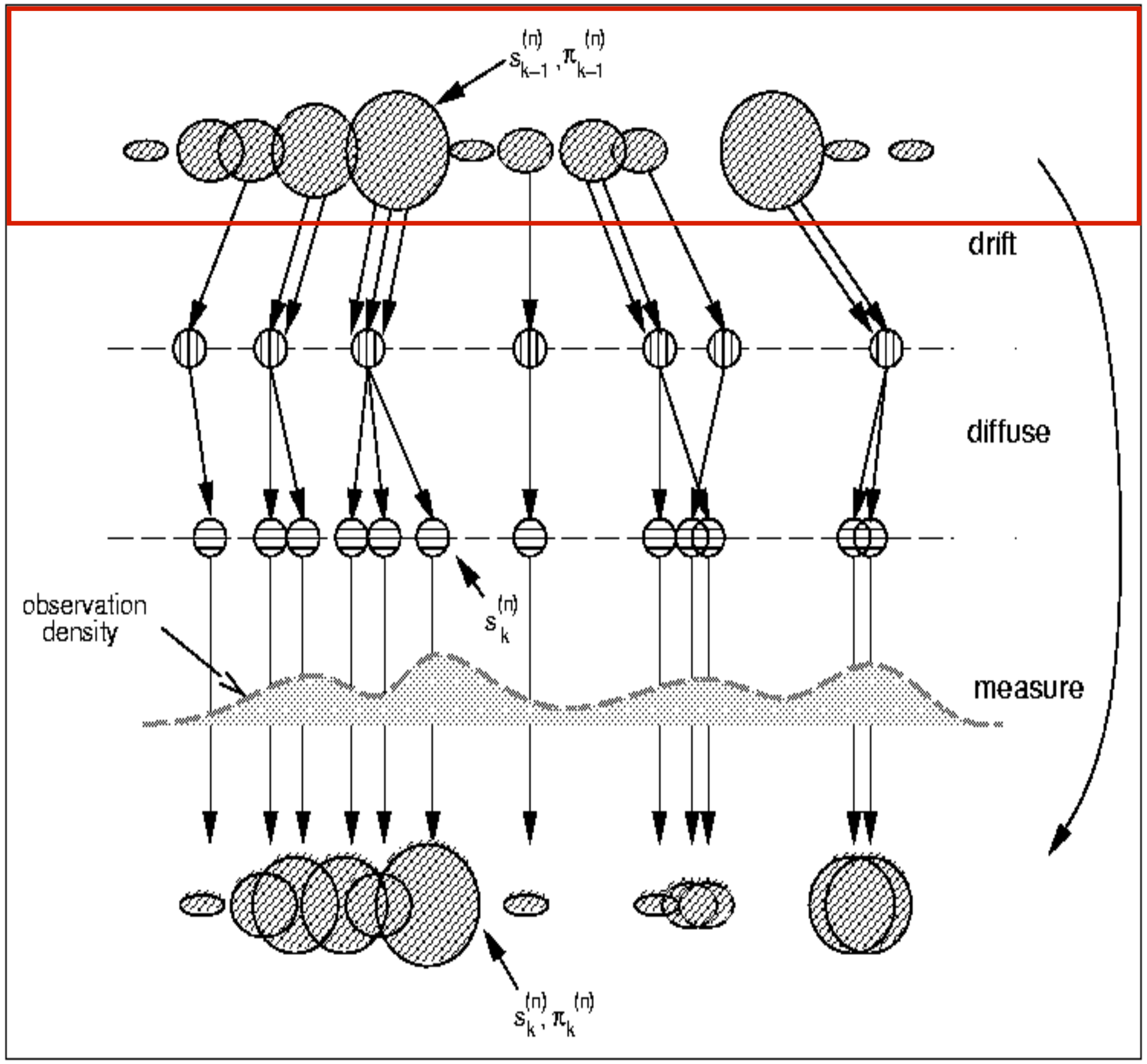
$u \sim U(0,1)$



Cumulative distribution of weights

Particle Filter

Posterior $p(\boldsymbol{x}_{k-1} | \mathbf{Z}_{k-1})$

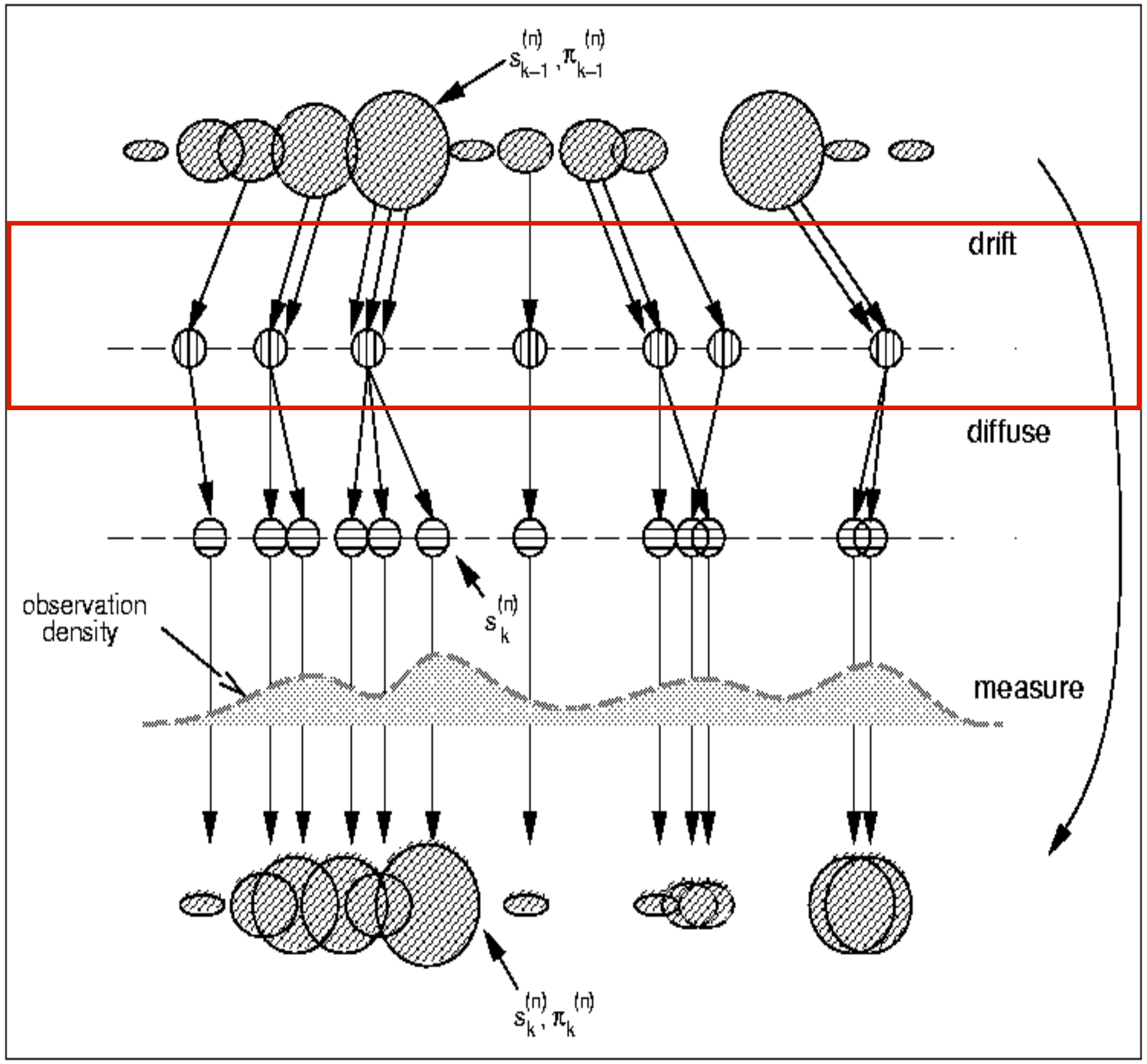


Isard & Blake '96

Particle Filter

Posterior $p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})$

resample



Isard & Blake '96

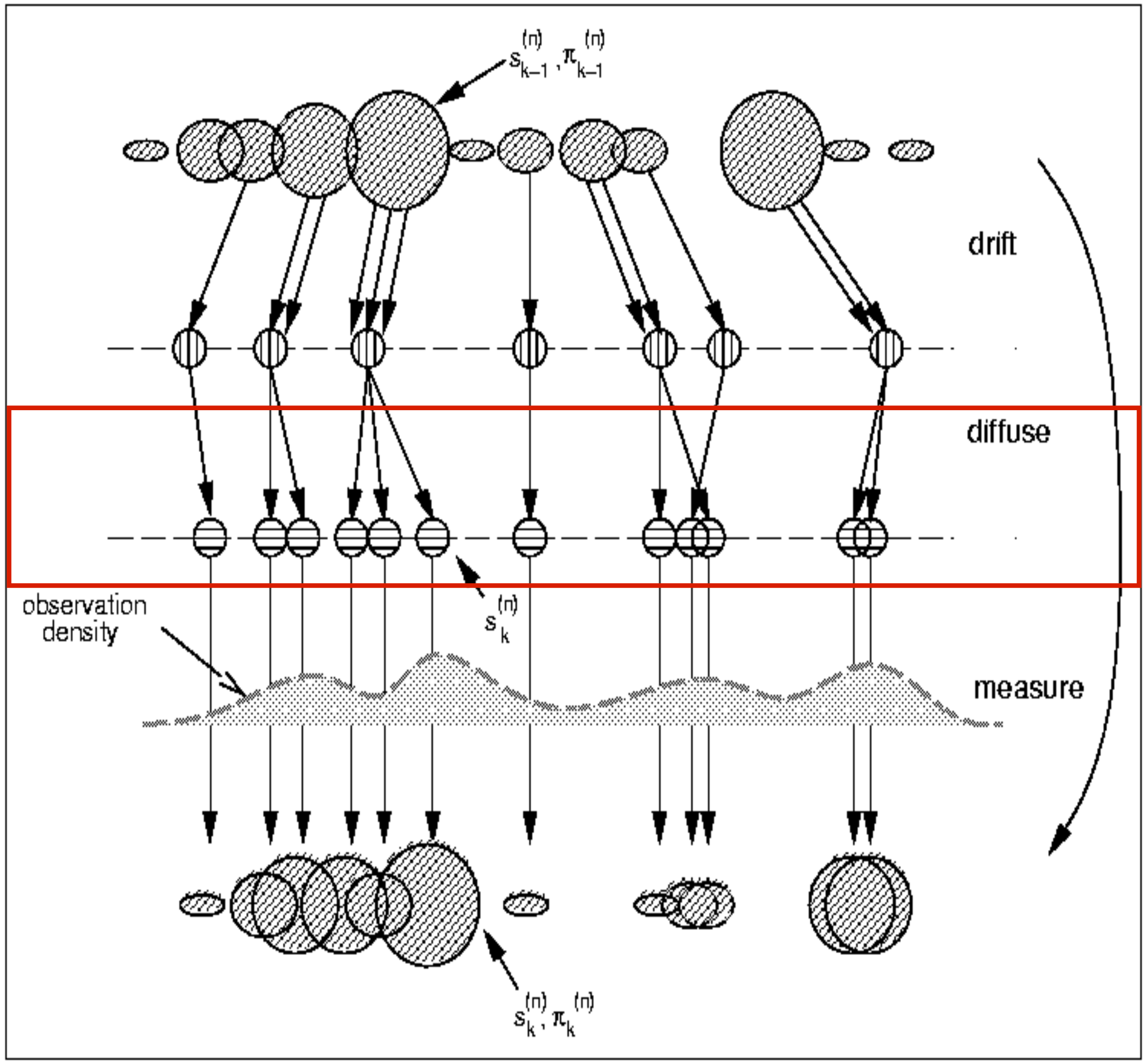
Particle Filter

Posterior $p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})$

resample

Apply temporal dynamics

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$



Isard & Blake '96

Particle Filter

Posterior $p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})$

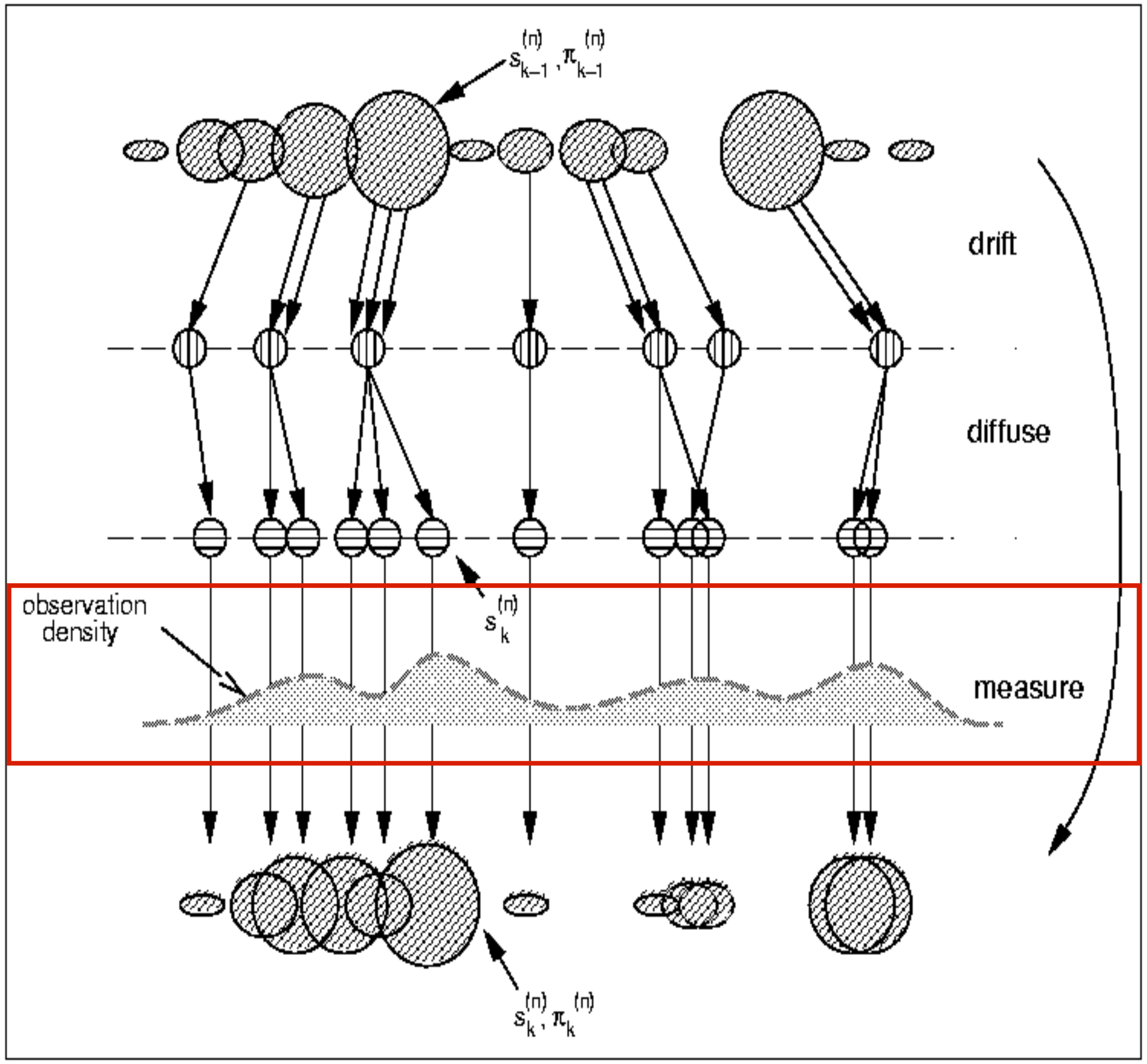
resample

Apply temporal dynamics

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

reweight

Likelihood $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$



Isard & Blake '96

Particle Filter

Posterior $p(\boldsymbol{x}_{k-1}|\mathbf{Z}_{k-1})$

resample

Apply temporal dynamics

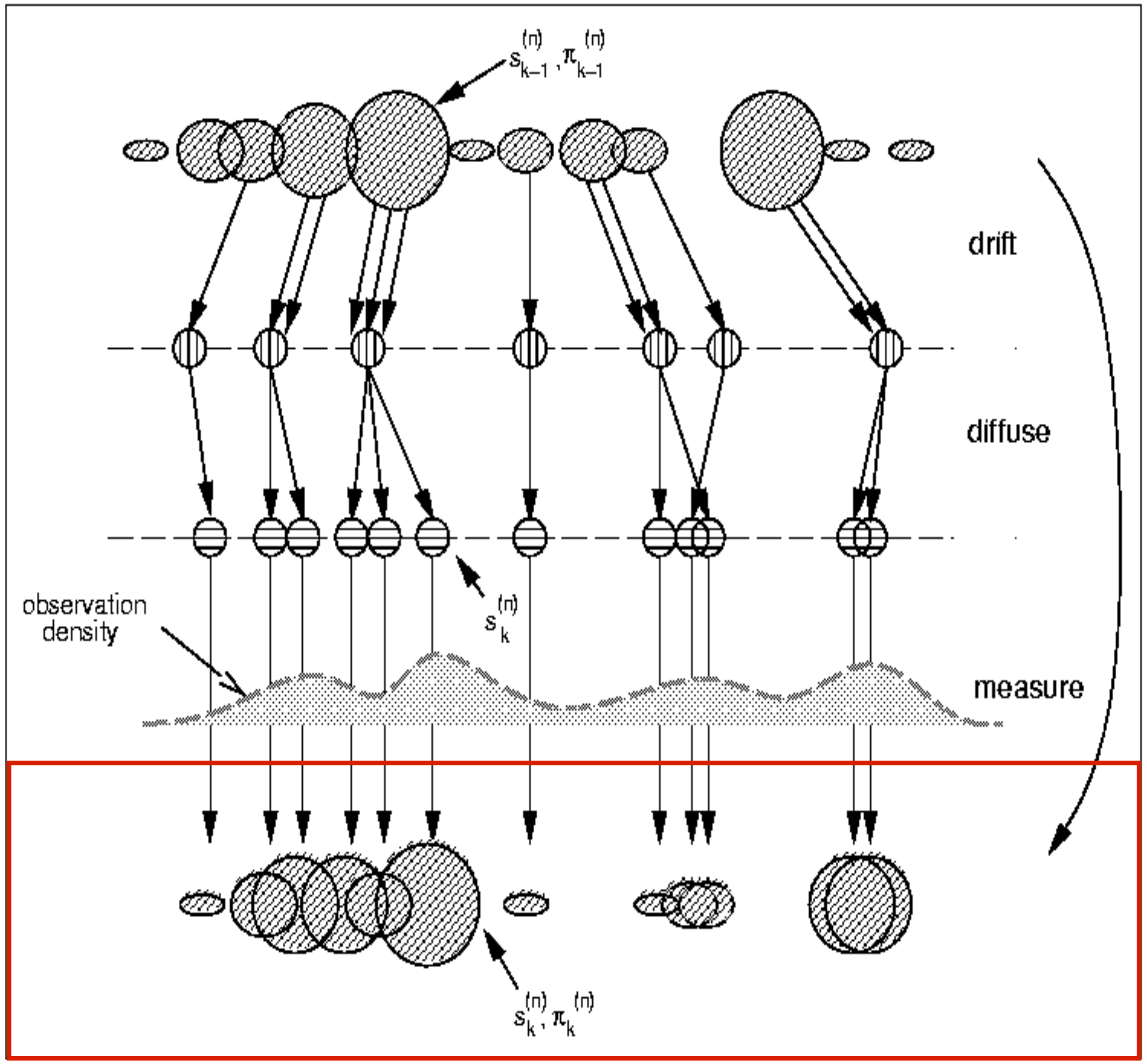
$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$

reweight

Likelihood $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$

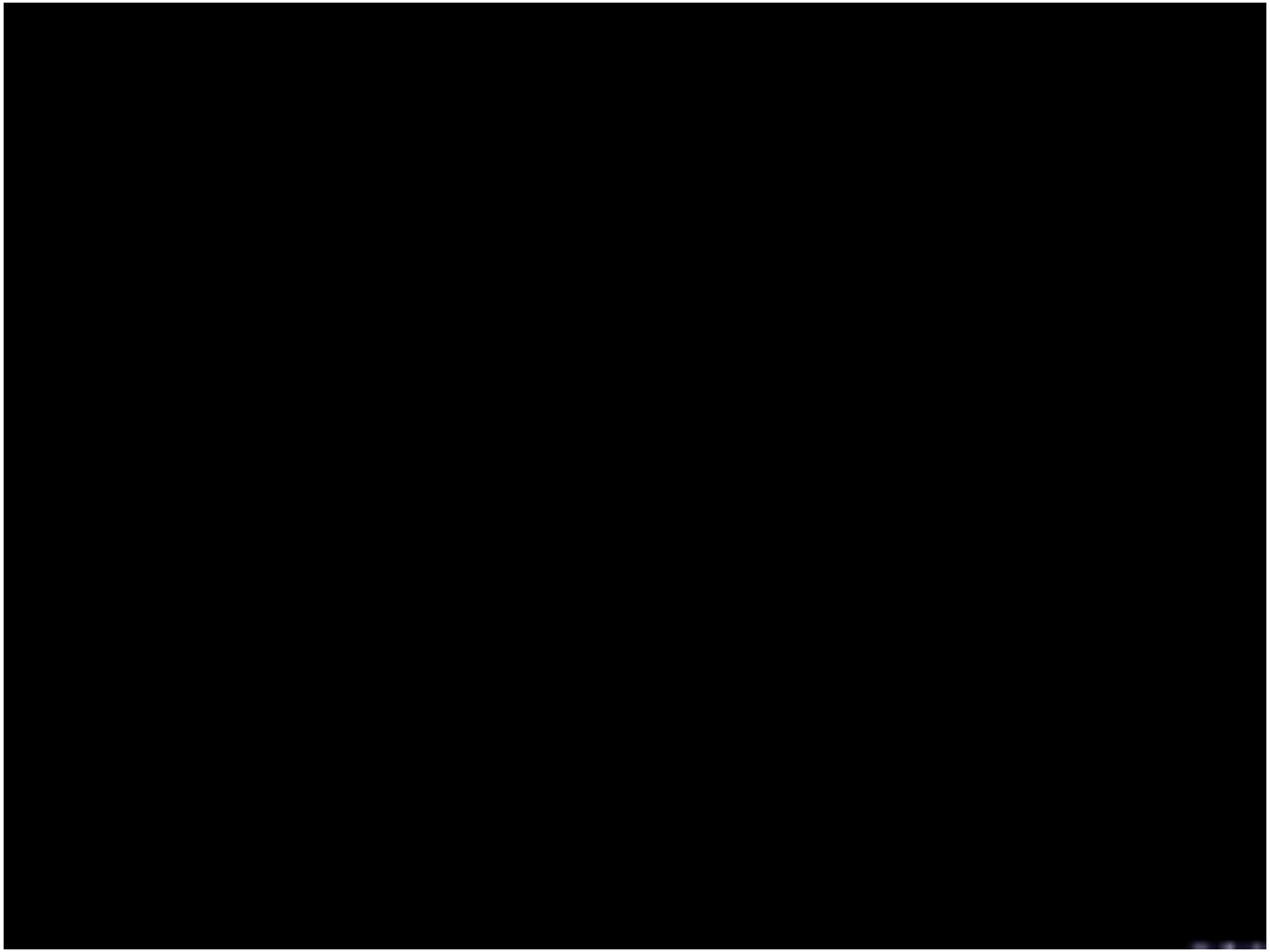
normalize

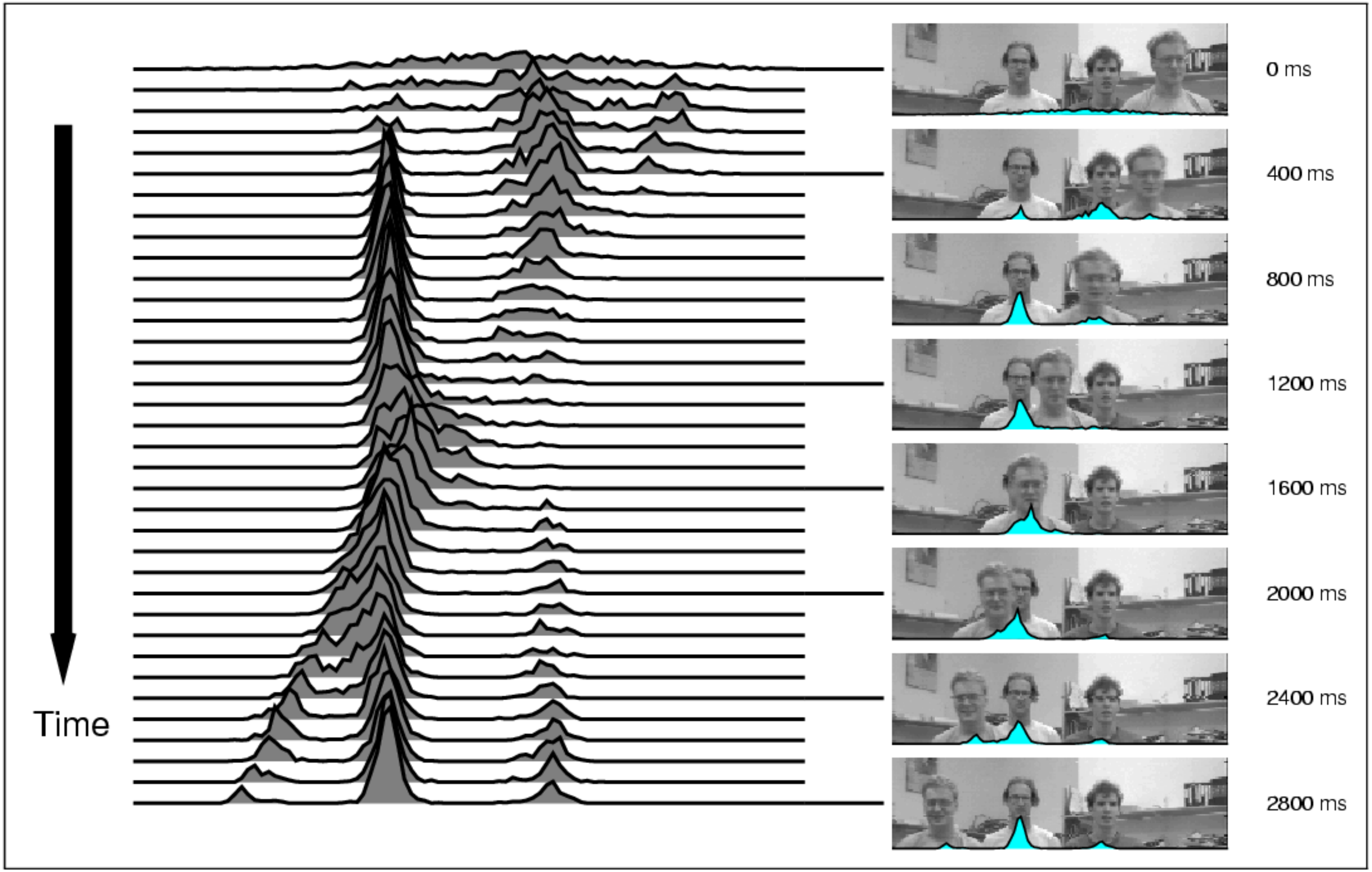
Posterior $p(\boldsymbol{x}_k|\mathbf{Z}_k)$



Isard & Blake '96

Particle Filtering in Action



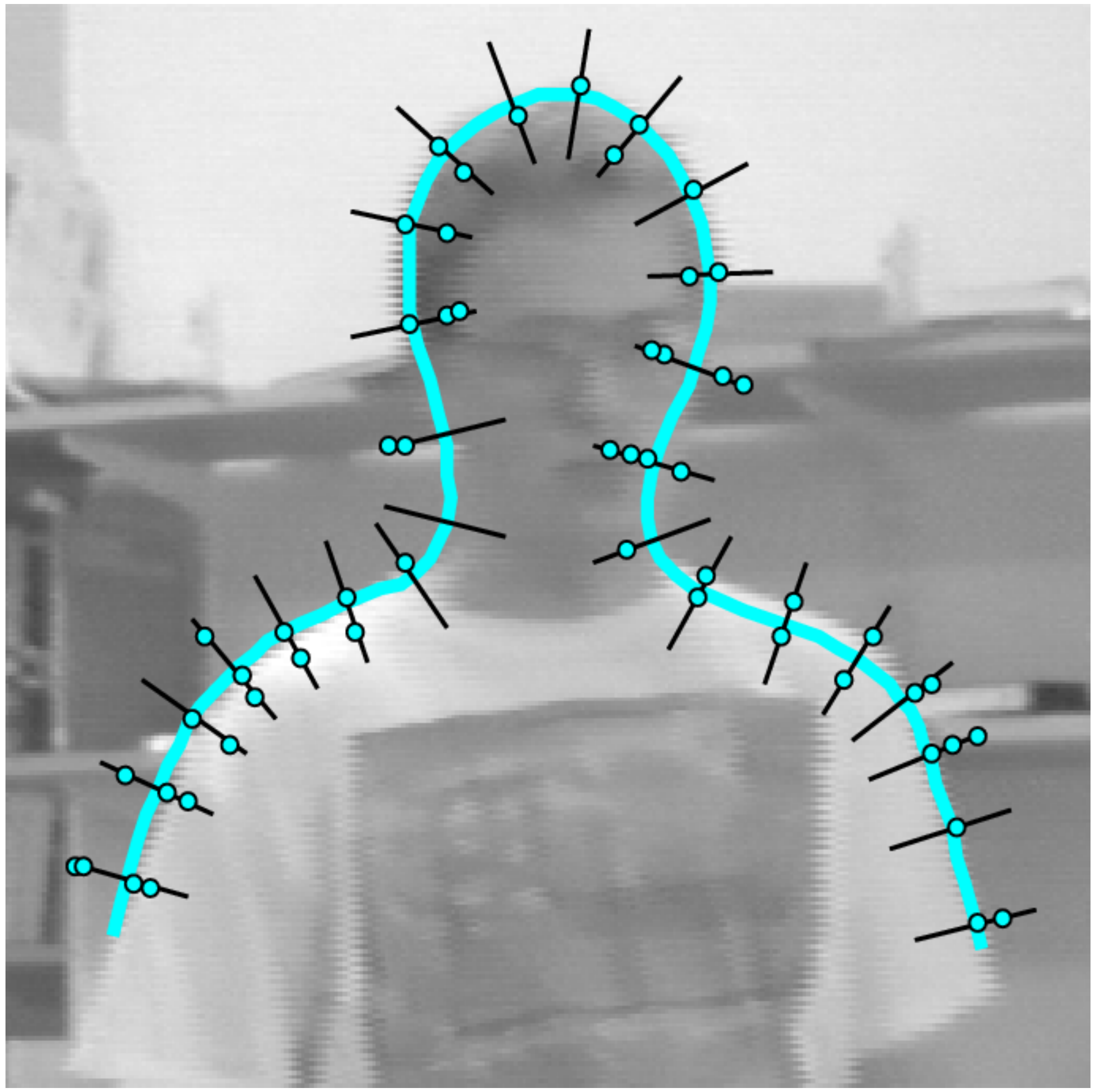


[Michael Isard]

Some Properties

- It can be shown that in the infinite particle limit this converges to the correct solution [Isard & Blake].
- In practice, we of course want to use a finite number.
  - In low-dimensional spaces we might only need 100s of particles for the procedure to work well.
  - In high-dimensional spaces sometimes 1000s or even 10000s particles are needed.
- There are many variants of this basic procedure, some of which are more efficient (e.g. need fewer particles)
  - See e.g.: Arnaud Doucet, Simon Godsill, Christophe Andrieu: On sequential Monte Carlo sampling methods for Bayesian filtering.

Contour Tracking



State: control points of spline-based contour representation

Measurements: edge strength perpendicular to contour

Dynamics: 2nd–order Markov (often learned)

[Isard & Blake, "Condensation - conditional density propagation for visual tracking." IJCV, 1998]

High-Dimensional State Spaces

- Tracking a hand (high-dimensional state space)

[Michael Isard]

Tracking in Clutter

- Tracking a leaf that moves fast in a very cluttered scene:

[Michael Isard]