Graphical Models and Their Applications

Tracking

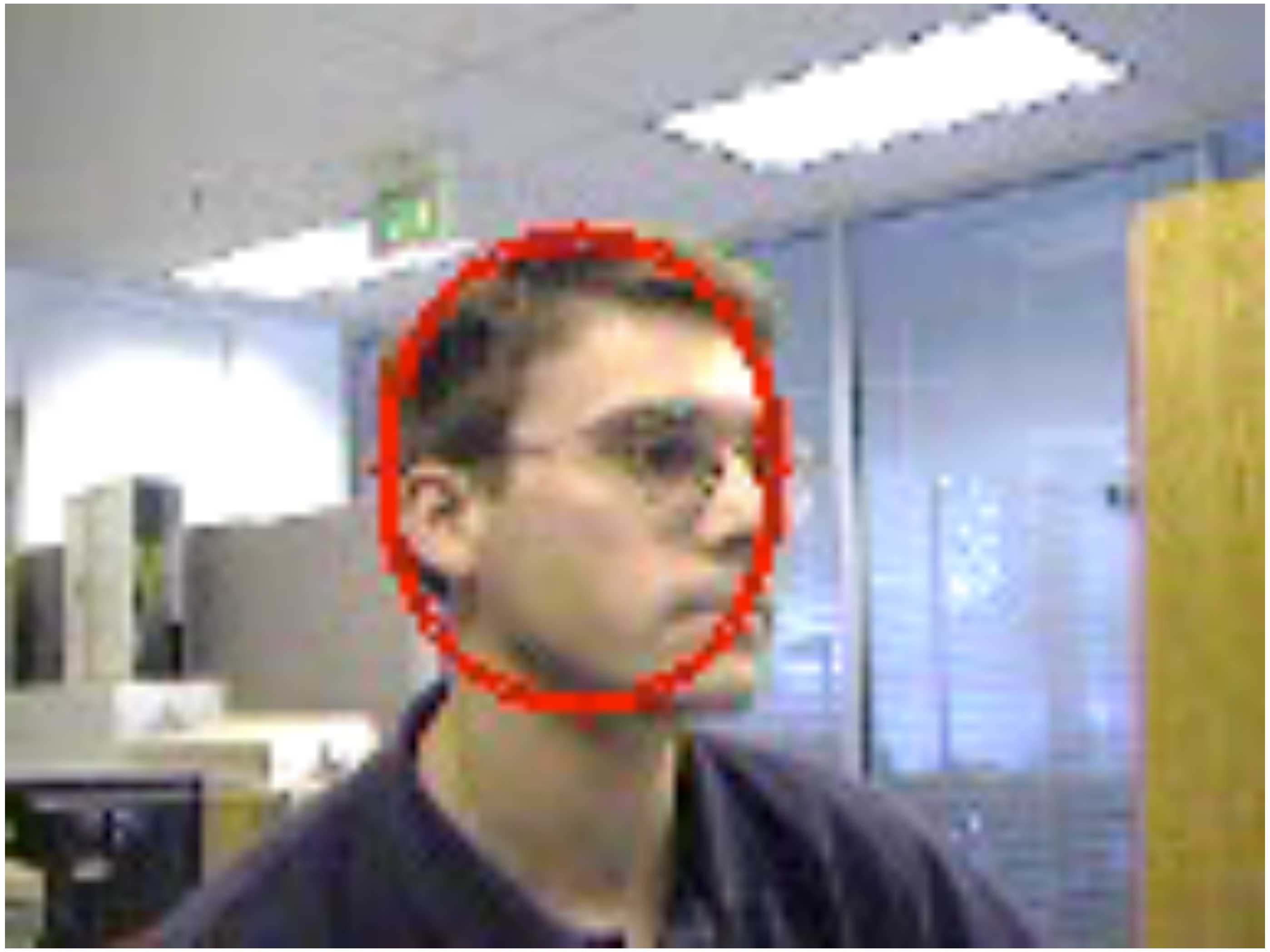
Jan 11, 2019

Gerard Pons-Moll

http://www.d2.mpi-inf.mpg.de/gm
slides adapted from Stefan Roth @ TU Darmstadt

Face Tracking

- Face tracking using color histograms and image gradients along contour:



- http://robotics.stanford.edu/~birch/headtracker/

Lane Tracking

- Lane tracking, e.g. for car navigation:

- http://path.berkeley.edu/~zuwhan/lanedetection/index.html

“Bee Tracking”

- Tracking is also very useful for facilitating behavioral research in animals.

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

Topic: Tracking

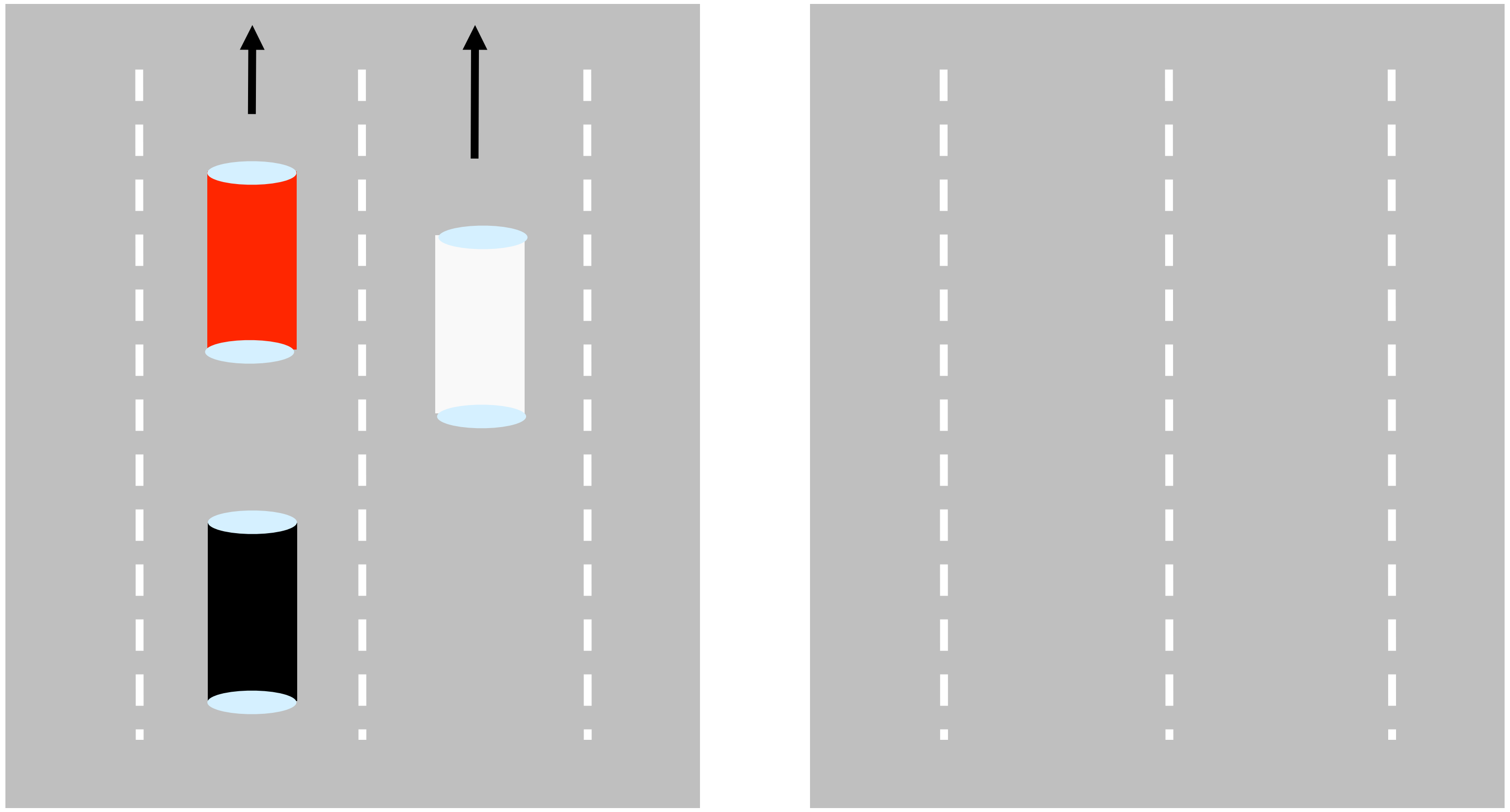
- Tracking is the problem of finding the motion of an object in an image sequence.

- Useful for a number of applications…
  - Animation & Interaction, Navigation, Video surveillance, Medical applications, Computer assisted living, etc.

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

  - We will talk only about: Rigid objects

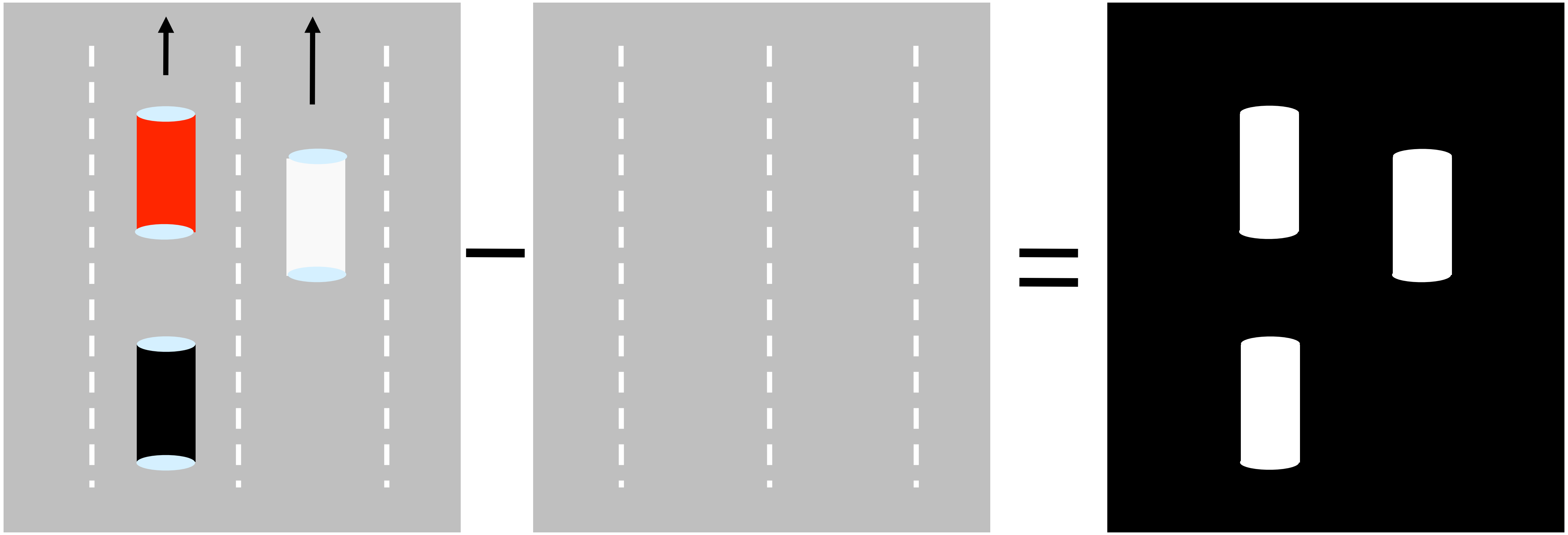
Illustration



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

[Michael Black]

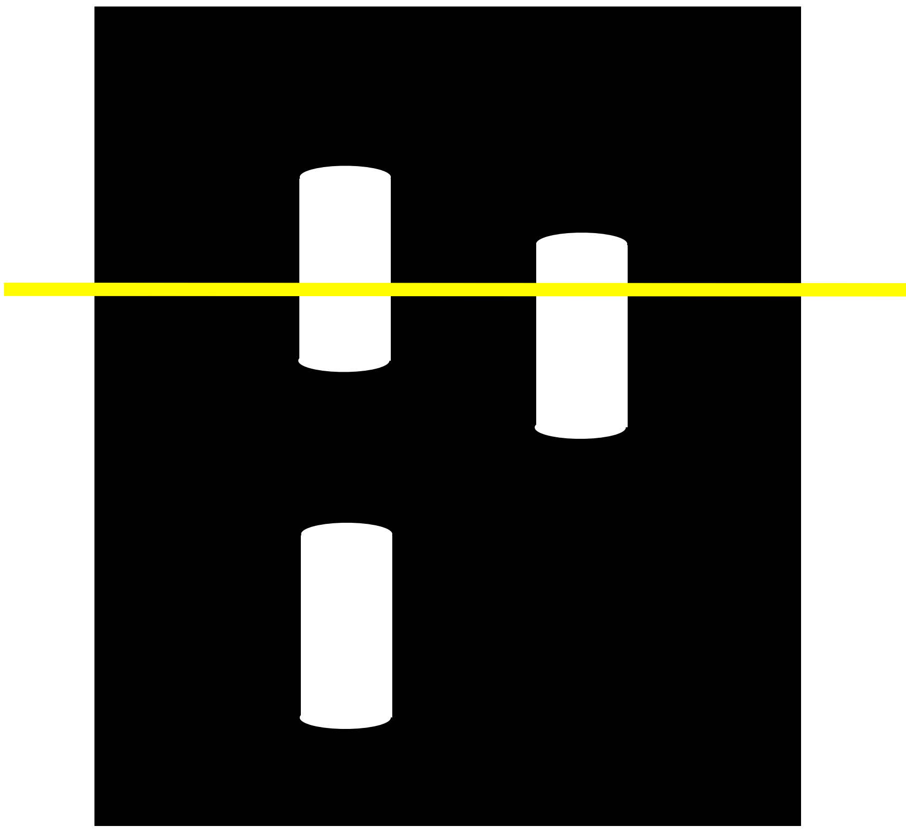
Illustration

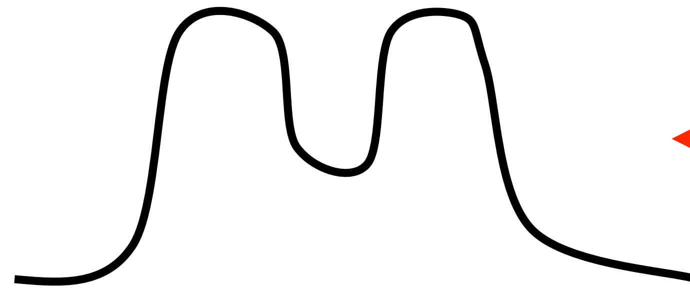


- Perform background subtraction.
- Obtain binary map of possible cars.
- But which one is the one we want to track?

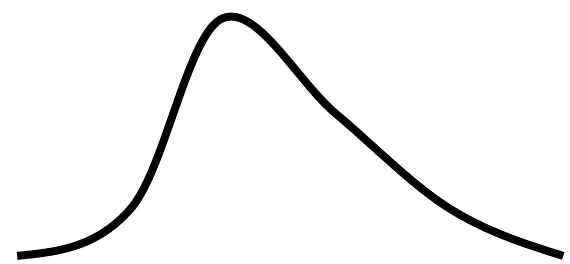
[Michael Black]

Bayesian Tracking

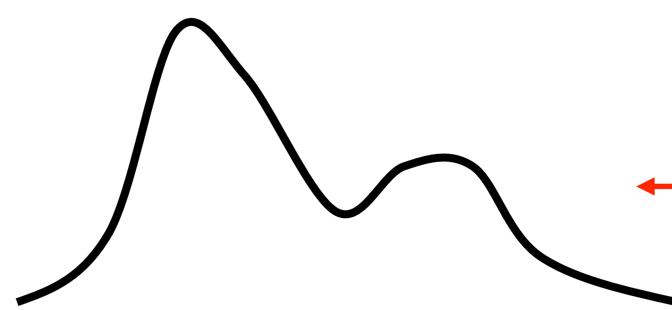




Likelihood:
noisy observation
$p(FG|\mathrm{car} = (x, y))$



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



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

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

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

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

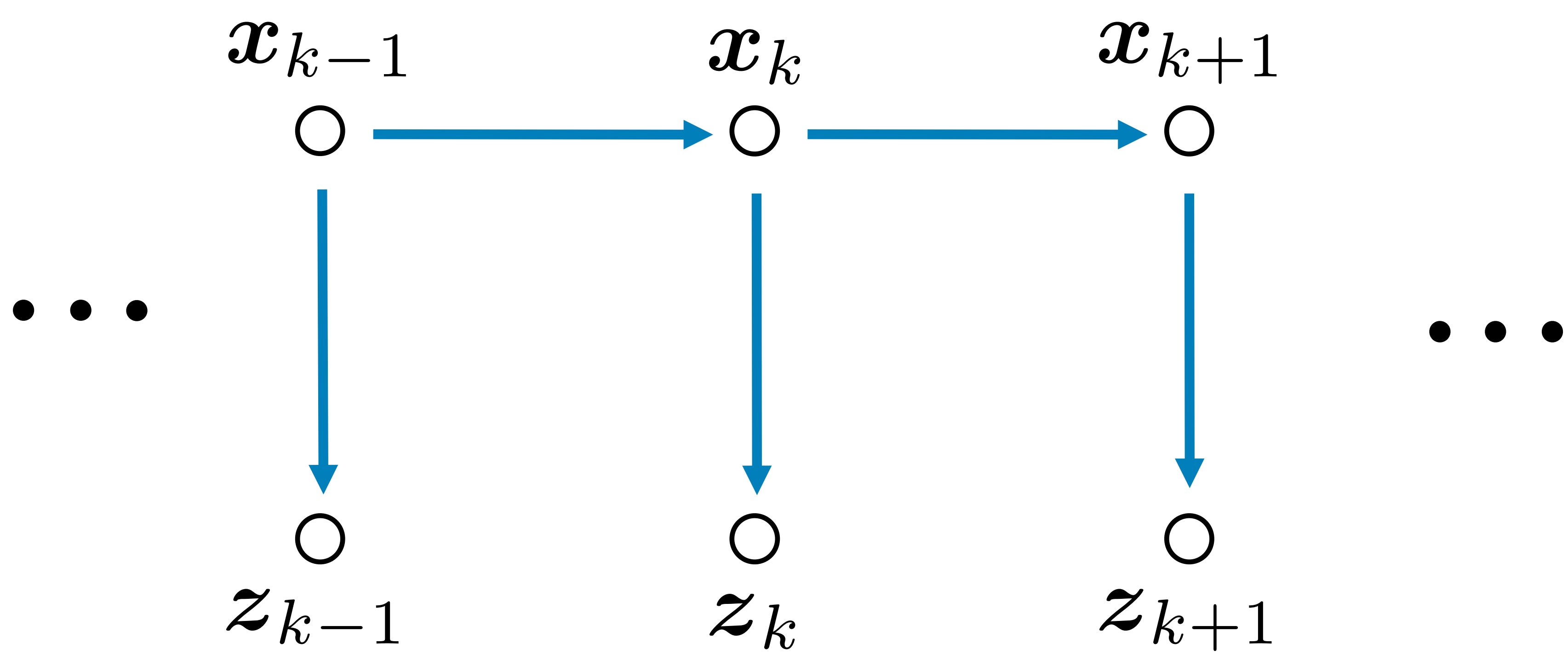
Bayesian Formulation

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

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

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

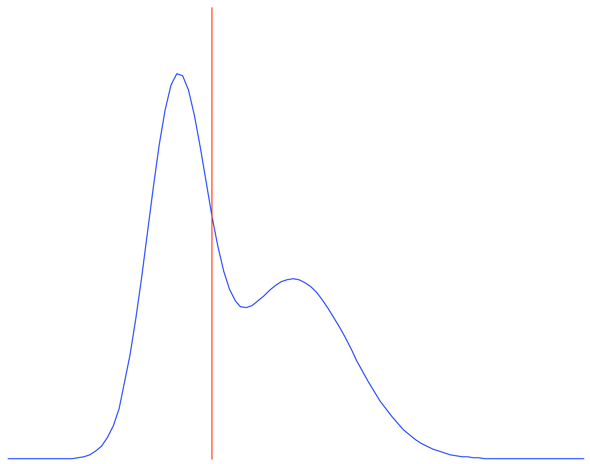
$$p(\boldsymbol{x}_k|\mathbf{X}_{k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

Estimators

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

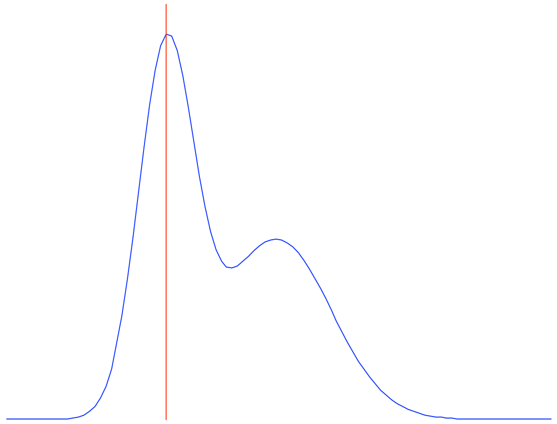
- posterior mean

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



- 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

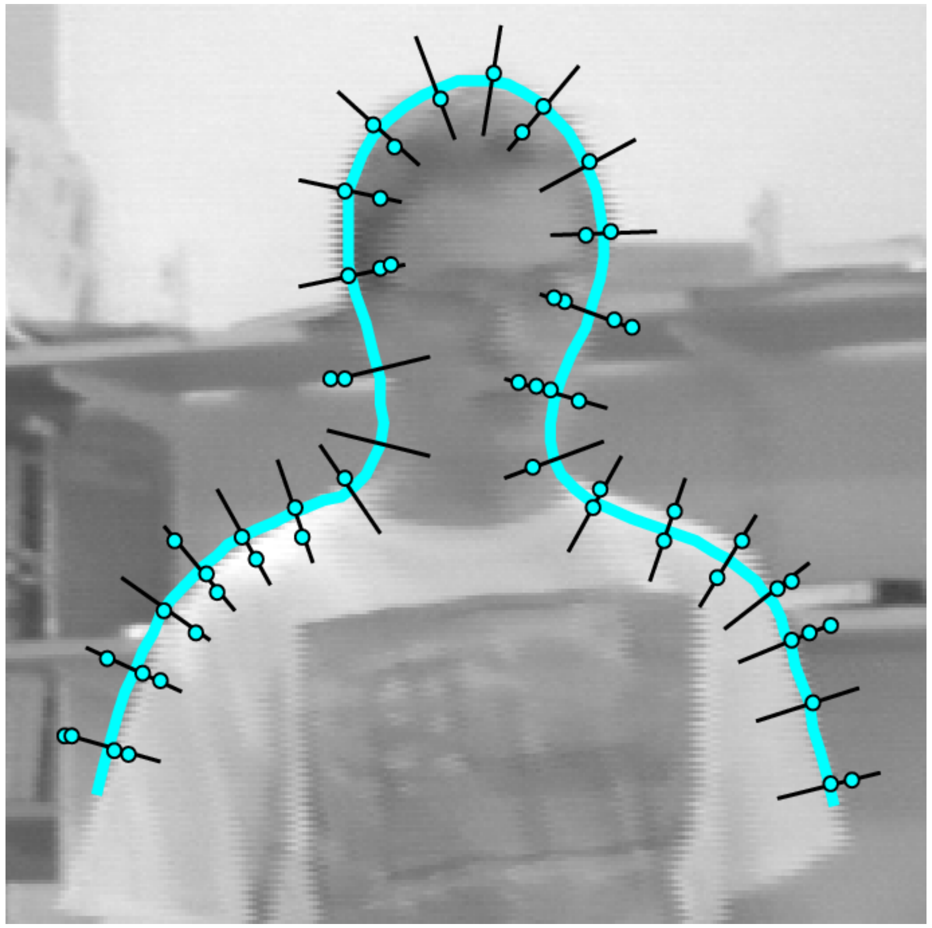
- $p(\boldsymbol{x}_k|\mathbf{Z}_k)$ can be an arbitrary, non-Gaussian, multi-modal distribution.
- The recursive equation often has no explicit solution, but can be numerically approximated using Monte Carlo techniques.

- Special Case - Kalman filter [Kalman, 1960]
  - If both likelihood and prior are Gaussian, the solution has closed form and the two estimators (posterior mean & MAP) are the same.
  - 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.
- We thus discuss only a more general version: **Particle filtering**
  - more general recursive estimation technique.
  - But also computationally much harder, and tricky to implement correctly...

[Michael Black]

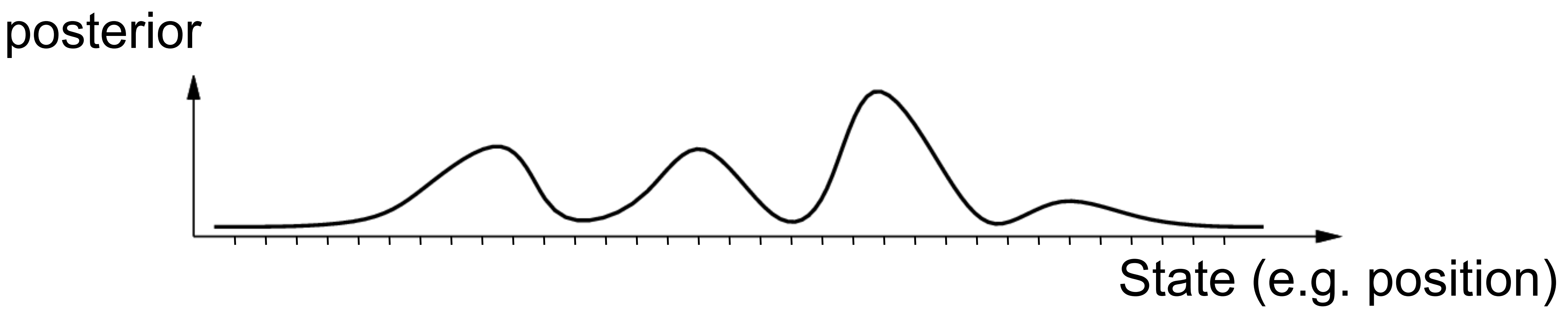
Multi-Modal Posteriors

- 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.
  - This rules out many parametric distributions, including the Gaussian.

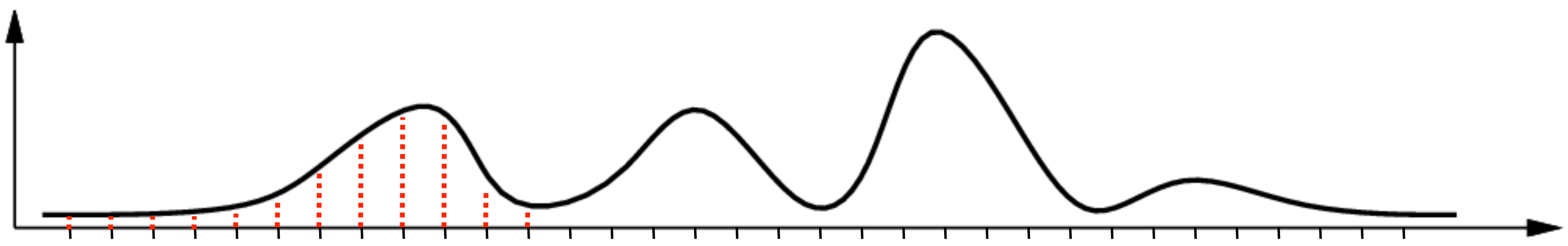
Multi-Modal Posteriors



- 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.

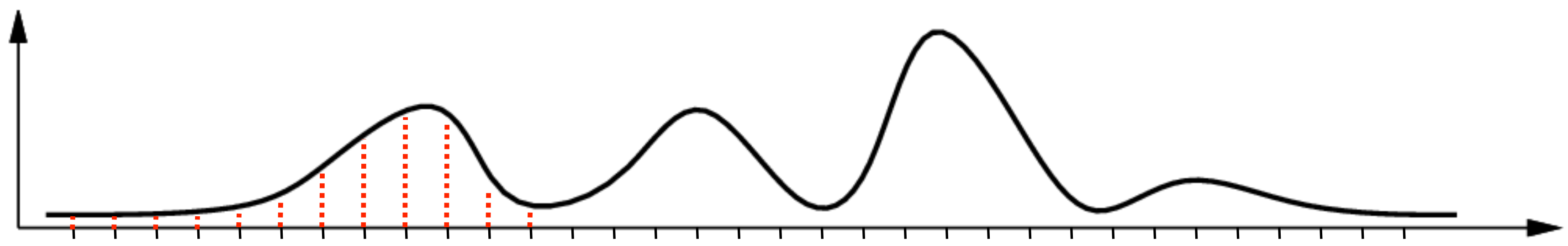
$$S = \left\{ (\boldsymbol{x}^{(i)}, w^{(i)});\ i = 1, \ldots, N \right\}$$

  - We usually use normalized weights:

$$\sum_{i=1}^{N} w^{(i)} = 1$$

Non-Parametric Approximation

- We could sample at regular intervals.



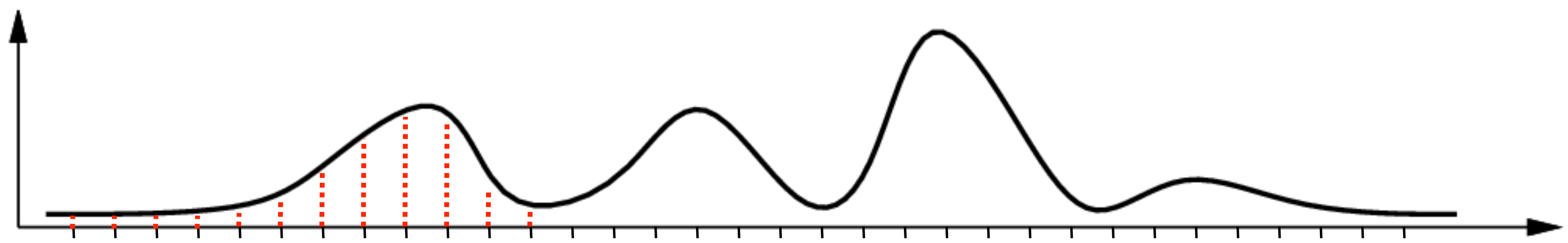
- Since there is no underlying parametric form, we call this a non-parametric representation or approximation.
- If needed, we can convert this back to a continuous density by assuming that each sample is represented by a small Gaussian mixture component:

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

  - Note though that this is typically not necessary!

Non-Parametric Approximation

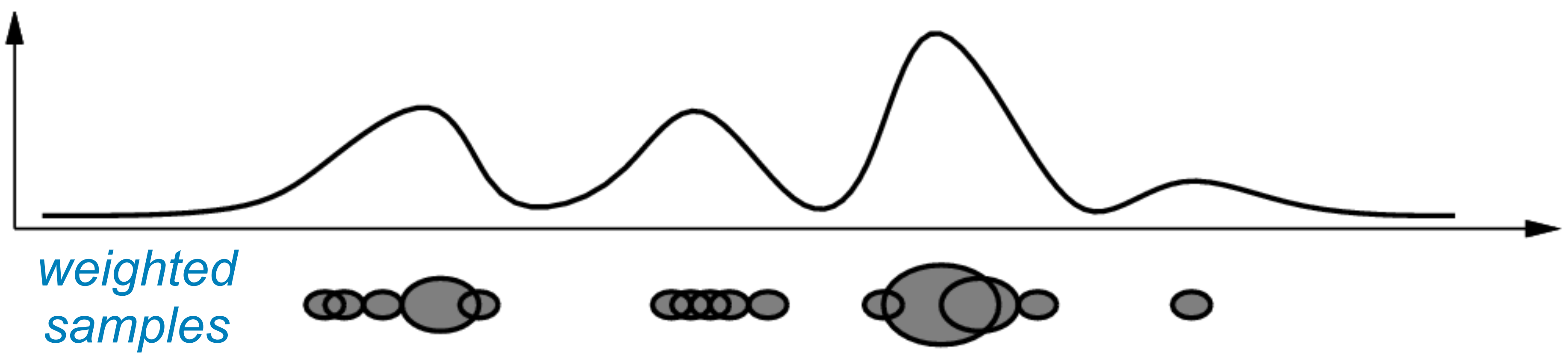
- We could sample at regular intervals.



- Problems?
  - Most samples have low weight – wasted computation.
  - How finely do we need to discretize?
  - High dimensional space – discretization impractical (exponential in the number of dimensions).

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 $$\sum_{i=1}^{N} w^{(i)} = 1$$

Importance Sampling

- Approach:
  - approximate expectation directly
  - goal:

$$\mathbb{E}[f] = \int f(\mathbf{z}) p(\mathbf{z})$$

- grid-sampling:
  - discretize z-space into a uniform grid
  - evaluate the integrand as a sum of the form:

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} f(\mathbf{z}^{(l)}) p(\mathbf{z}^{(l)})$$

  - but: number of terms grows exponentially with number of dimensions

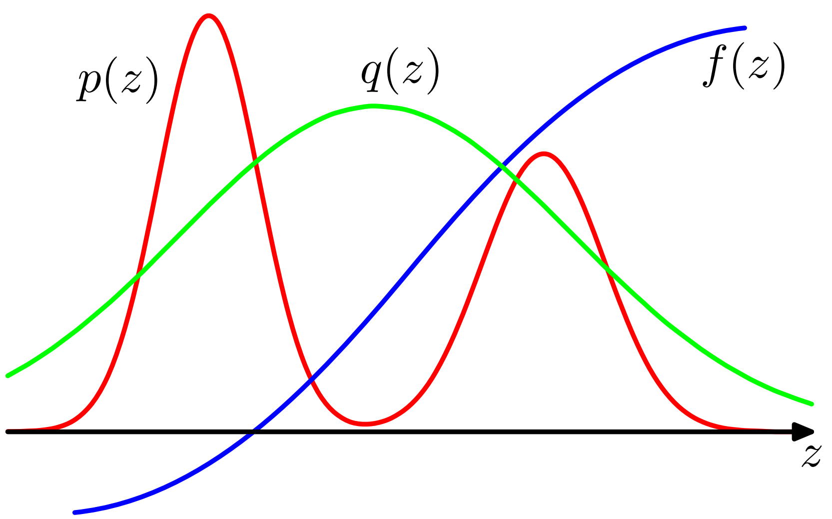
Importance Sampling

- Idea:
  - use a proposal distribution q(**z**) from which it is easy to draw samples
  - express expectation in the form of a finite sum over samples $\{\mathbf{z}^{(l)}\}$ drawn from q(**z**)

$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

$$\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}f(\mathbf{z}^{(l)})$$

  - with importance weights: $r_l = \frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}$



Importance Sampling

- typical setting:
  - p(**z**) can be only evaluated up to a normalization constant (unknown):

$$p(\mathbf{z}) = \tilde{p}(\mathbf{z})/Z_p$$

  - q(**z**) can be also treated in a similar fashion

$$q(\mathbf{z}) = \tilde{q}(\mathbf{z})/Z_q$$

  - then:

$$\begin{aligned} \mathbb{E}[f] &= \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \frac{Z_q}{Z_p}\int f(\mathbf{z})\frac{\tilde{p}(\mathbf{z})}{\tilde{q}(\mathbf{z})}q(\mathbf{z})d\mathbf{z} \\ &\simeq \frac{Z_q}{Z_p}\frac{1}{L}\sum_{l=1}^{L}\tilde{r}_l f(\mathbf{z}^{(l)}) \end{aligned}$$

  - with: $\tilde{r}_l = \frac{\tilde{p}(\mathbf{z}^{(l)})}{\tilde{q}(\mathbf{z}^{(l)})}$

Importance Sampling

- Ratio of normalization constants can be evaluated:

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(\mathbf{z})d\mathbf{z} = \int \frac{\tilde{p}(\mathbf{z})}{\tilde{q}(\mathbf{z})}q(\mathbf{z})d\mathbf{z} \simeq \frac{1}{L}\sum_{l=1}^{L}\tilde{r}_l$$

  - and therefore:

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} w_l f(\mathbf{z}^{(l)})$$

  - with:

$$w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\frac{\tilde{p}(\mathbf{z}^{(l)})}{\tilde{q}(\mathbf{z}^{(l)})}}{\sum_m \frac{\tilde{p}(\mathbf{z}^{(m)})}{\tilde{q}(\mathbf{z}^{(m)})}}$$

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 again using another particle set (we discuss in a minute how…): $\sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$

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

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 again using another particle set (we discuss in a minute how…): $\sum_i w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$

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

- 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

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

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

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

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 at the current time step: $p(\boldsymbol{x}_k|\mathbf{Z}_k)$

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

- 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

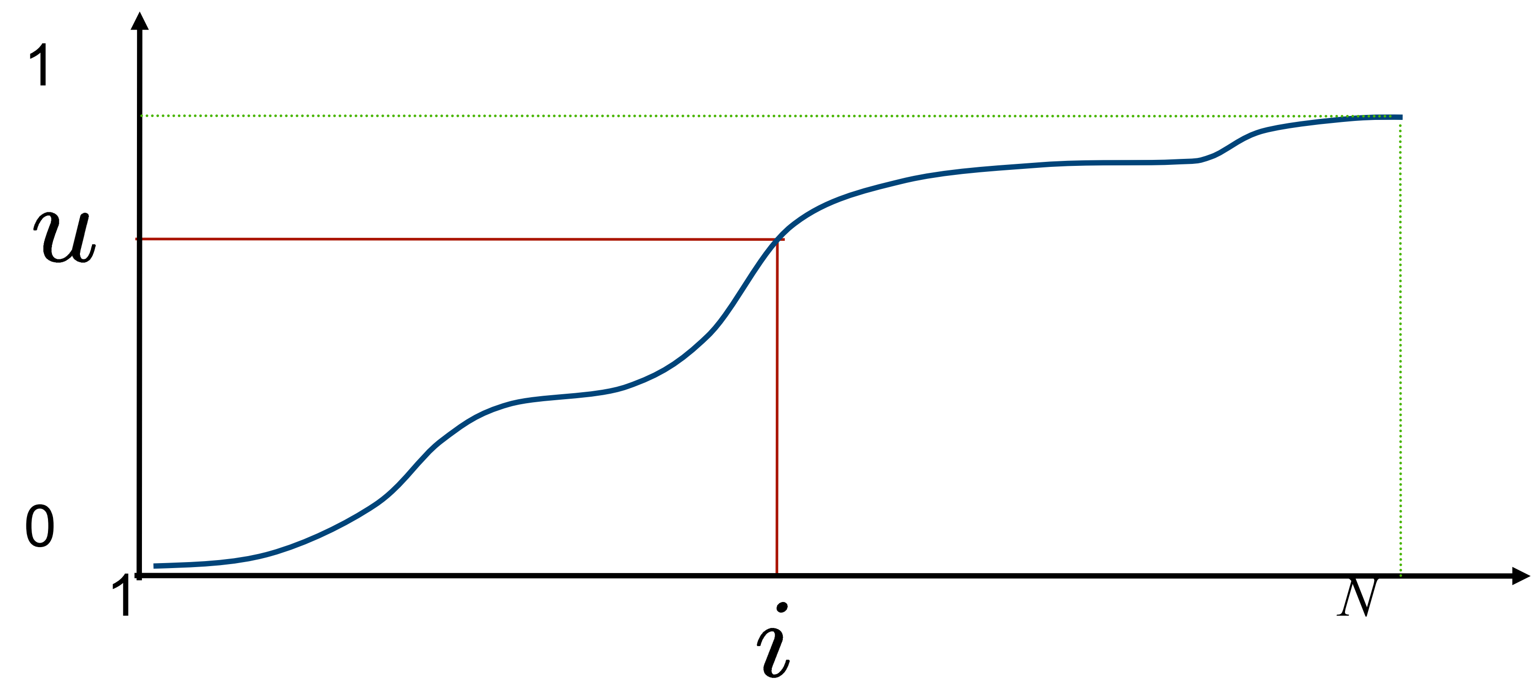
- Given weighted sample set

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

- Draw unweighted samples by sampling from the weight distribution:

sample

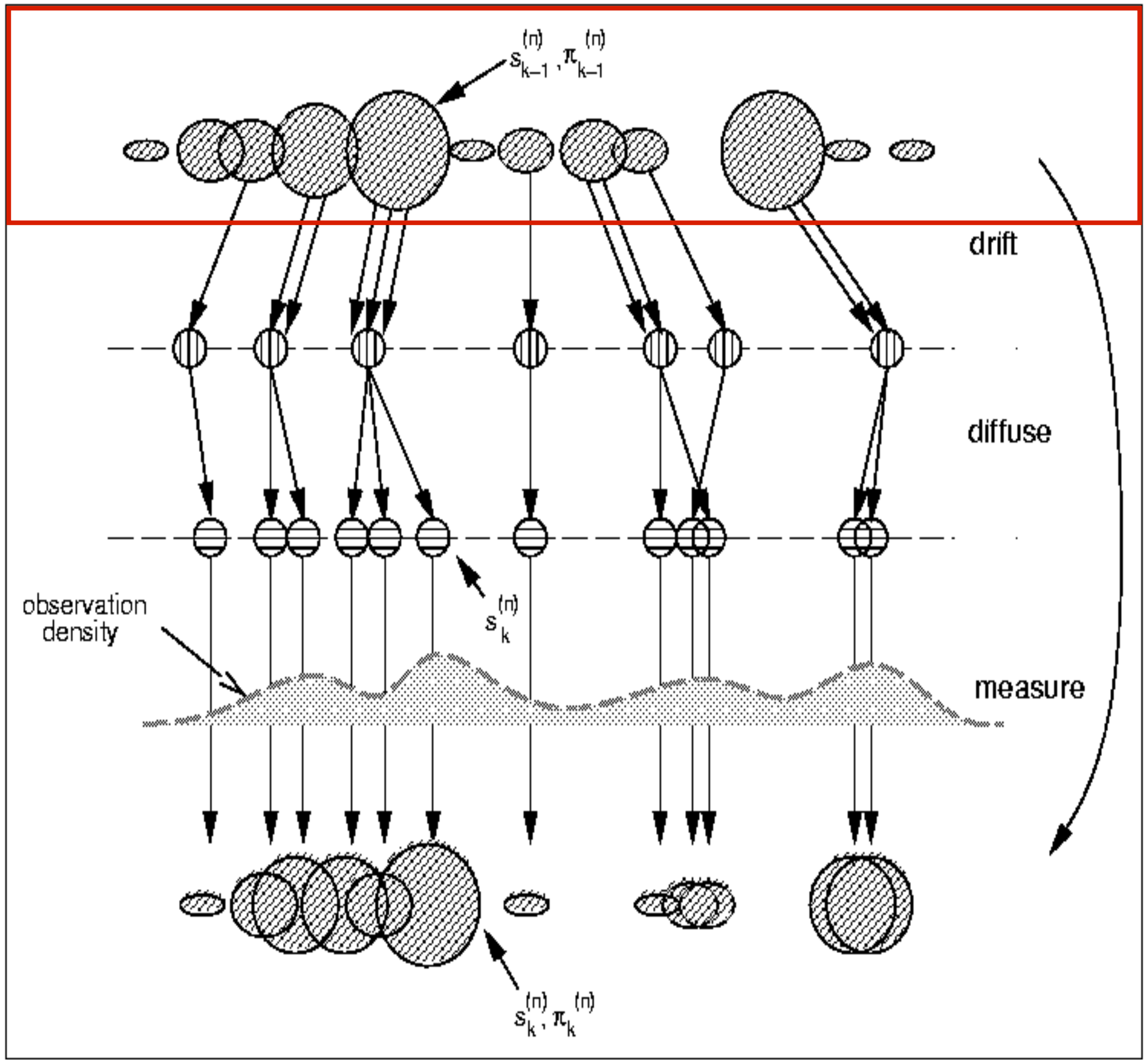
$$u \sim U(0,1)$$



Cumulative distribution of weights

Particle Filter

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

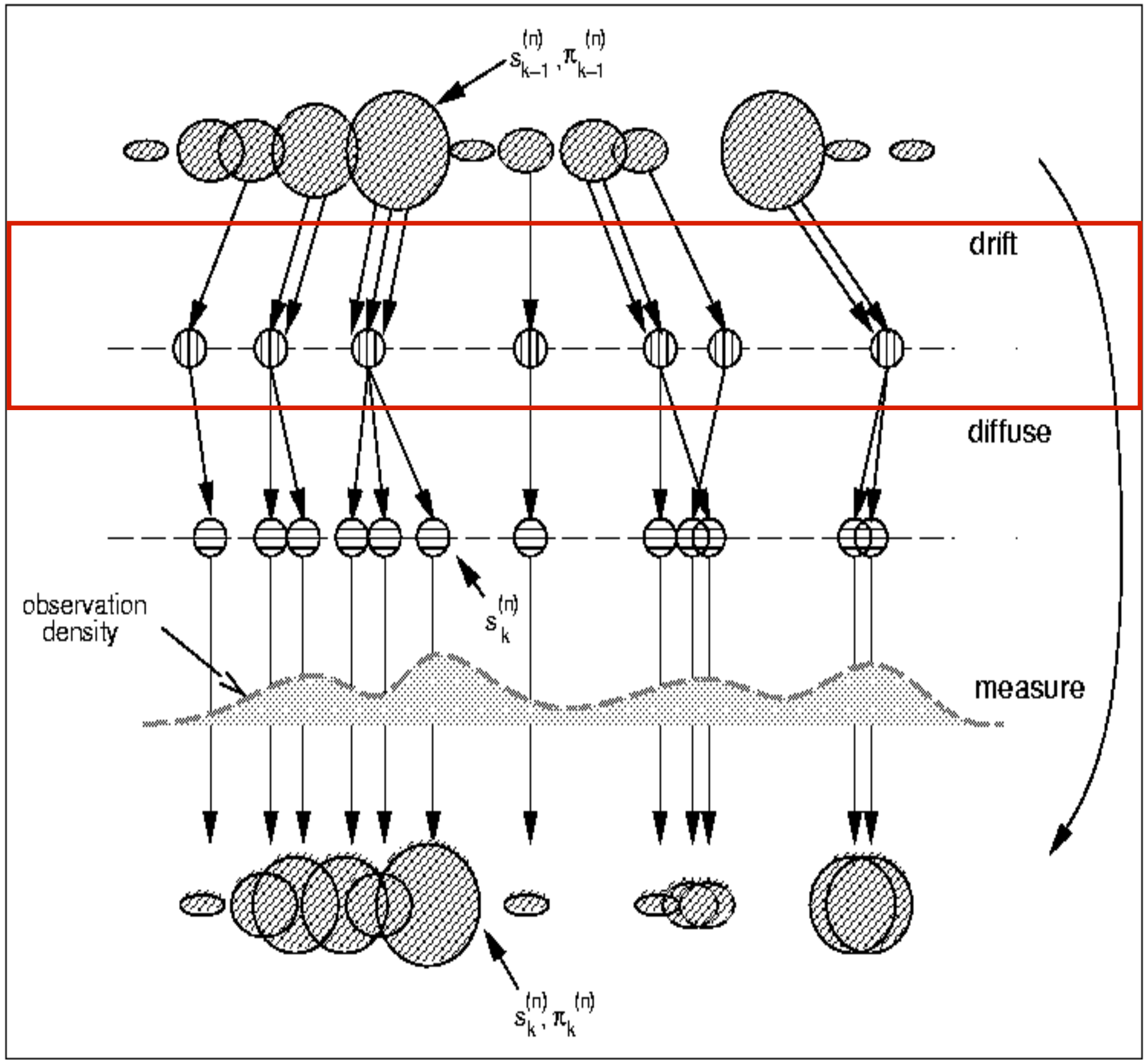


Isard & Blake '96

Particle Filter

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

resample



Isard & Blake '96

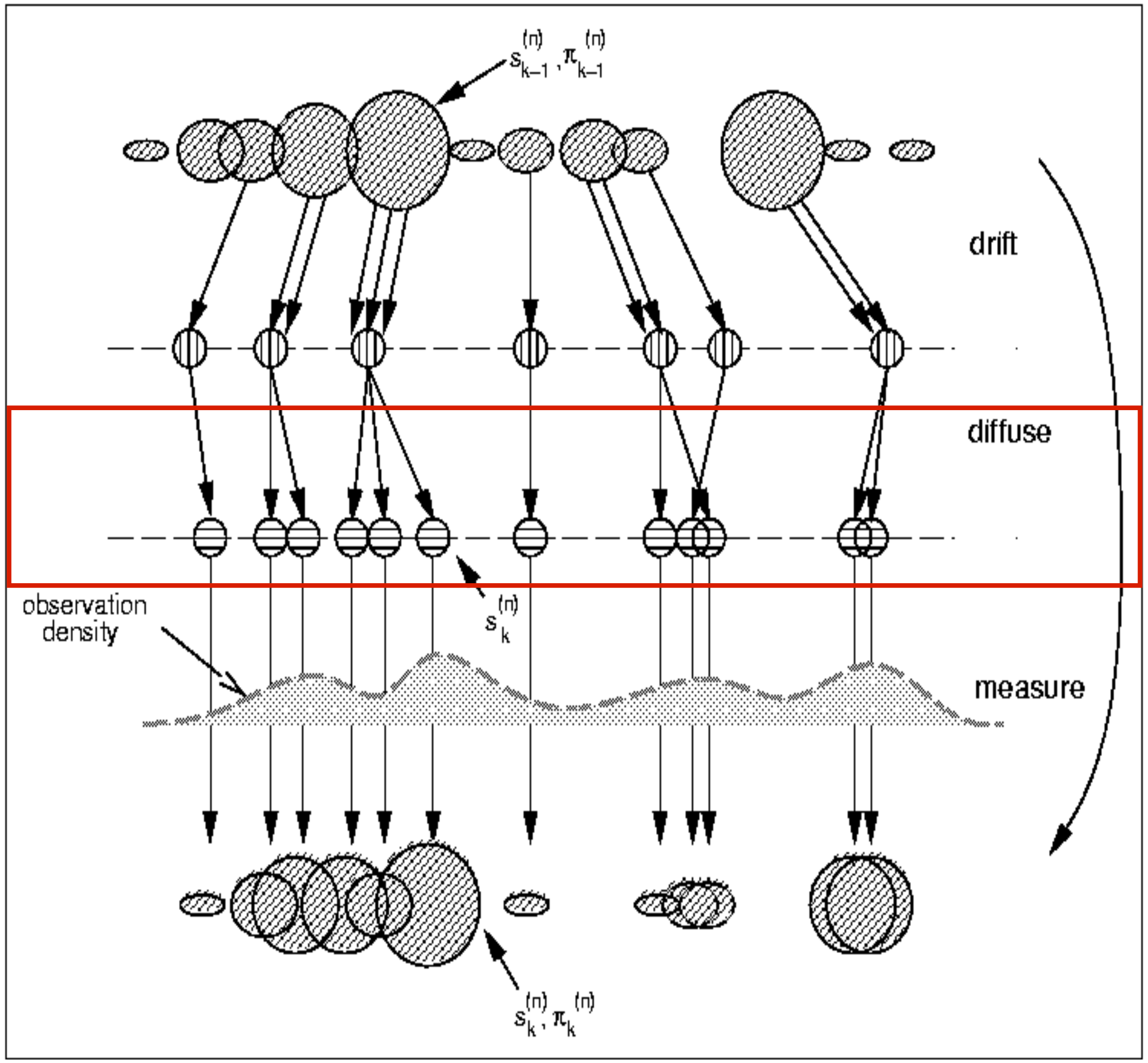
Particle Filter

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

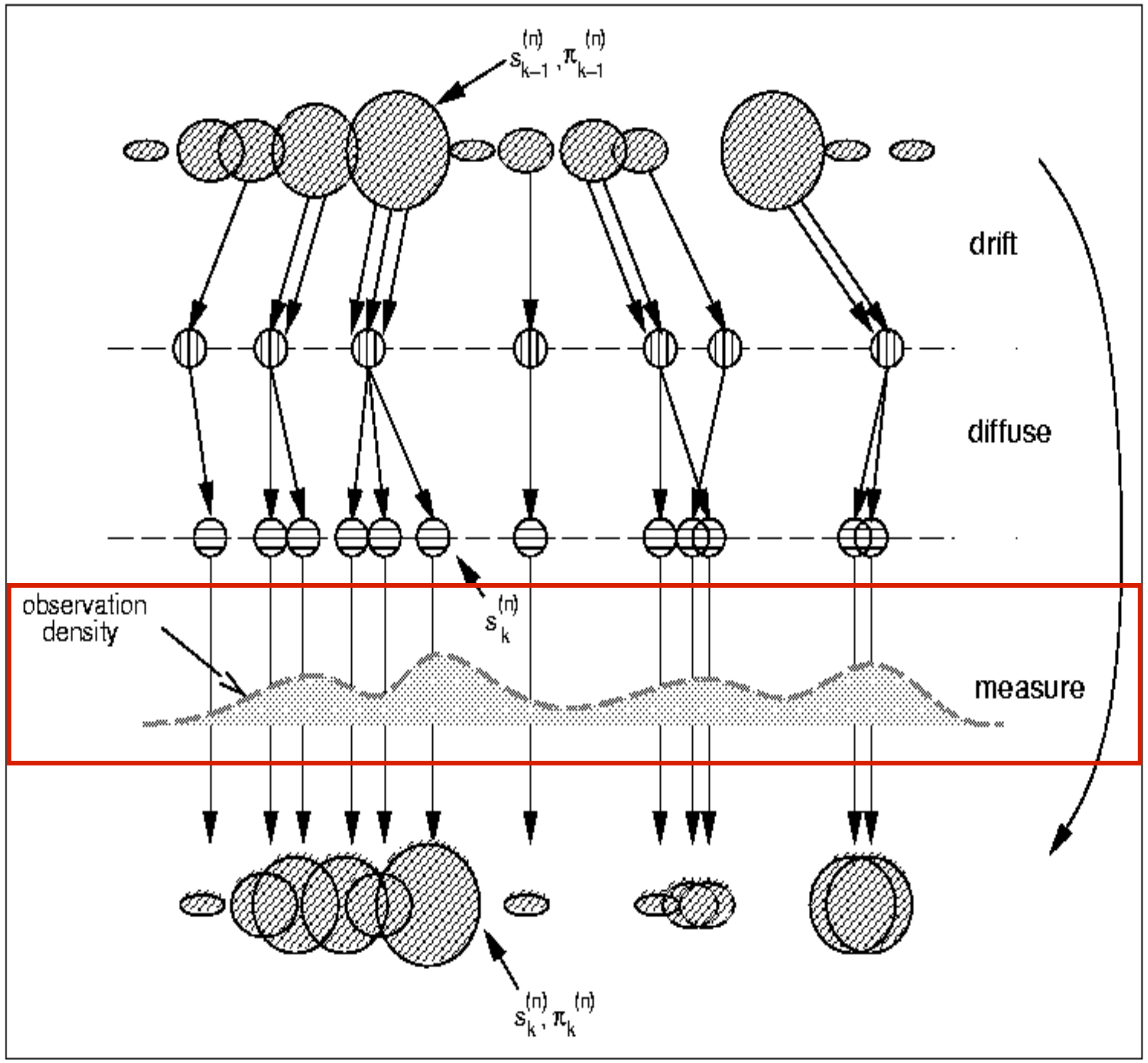
↓ **resample**

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

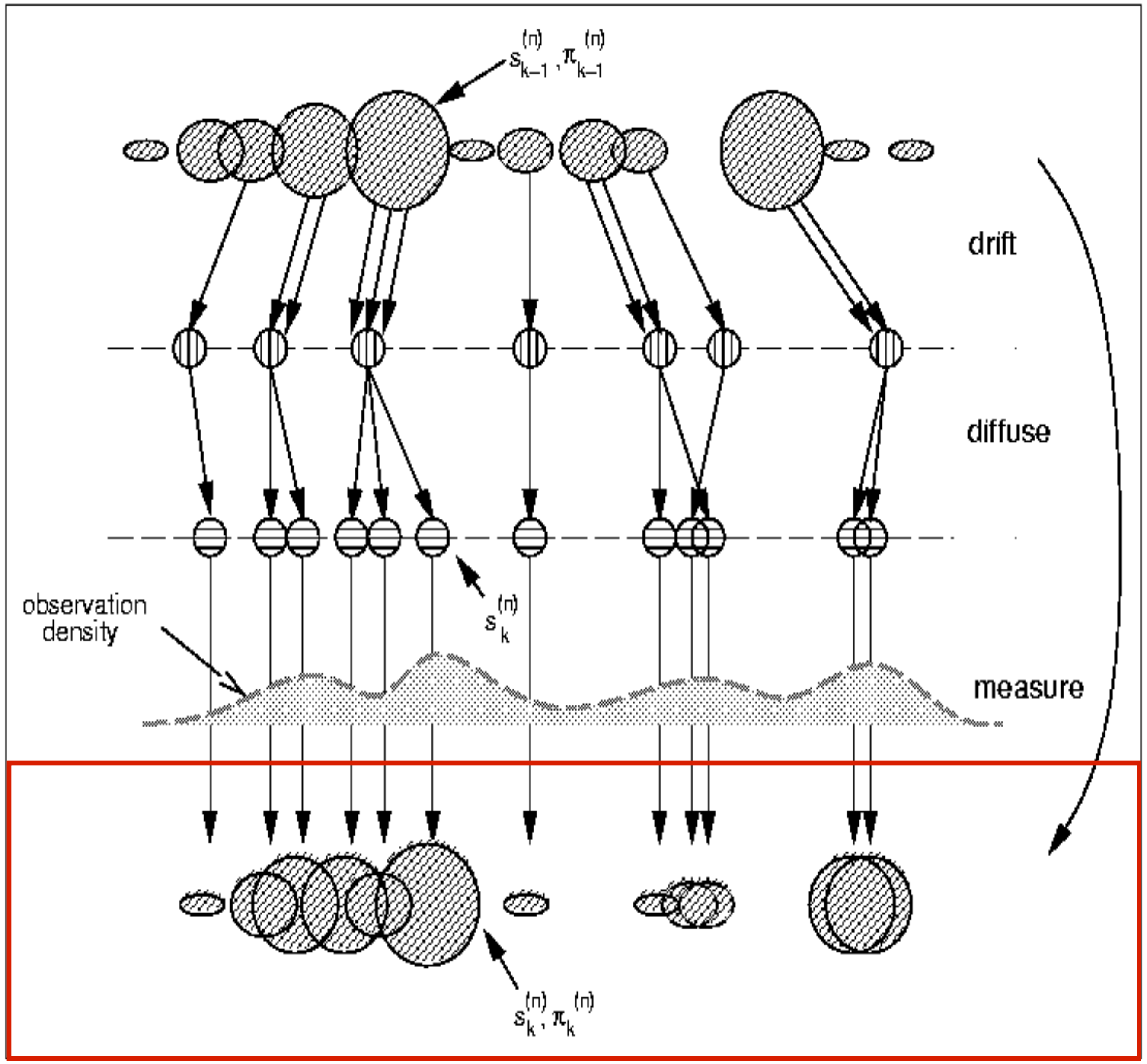
$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$

reweight

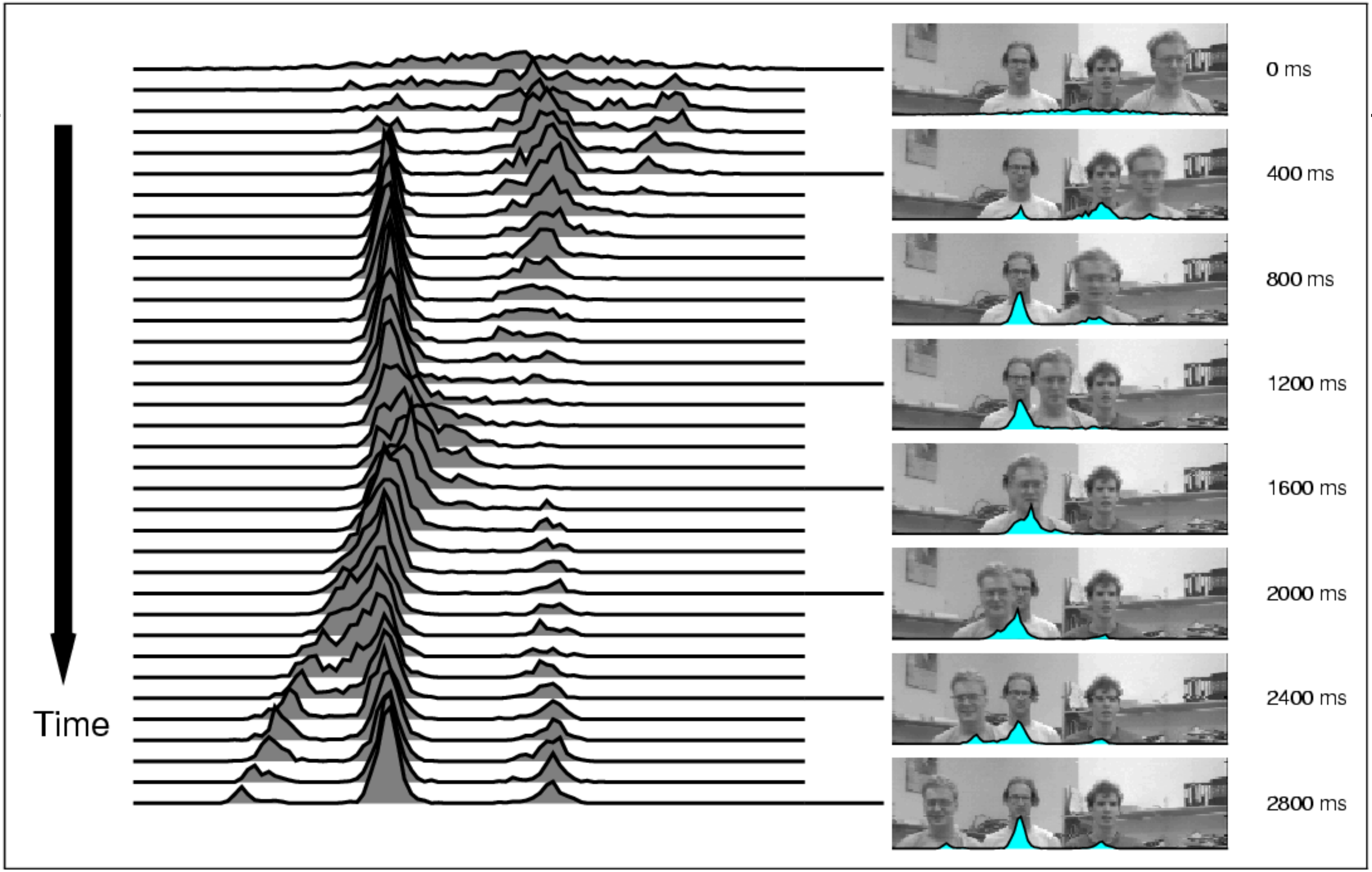
Likelihood $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$

normalize

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



Isard & Blake '96

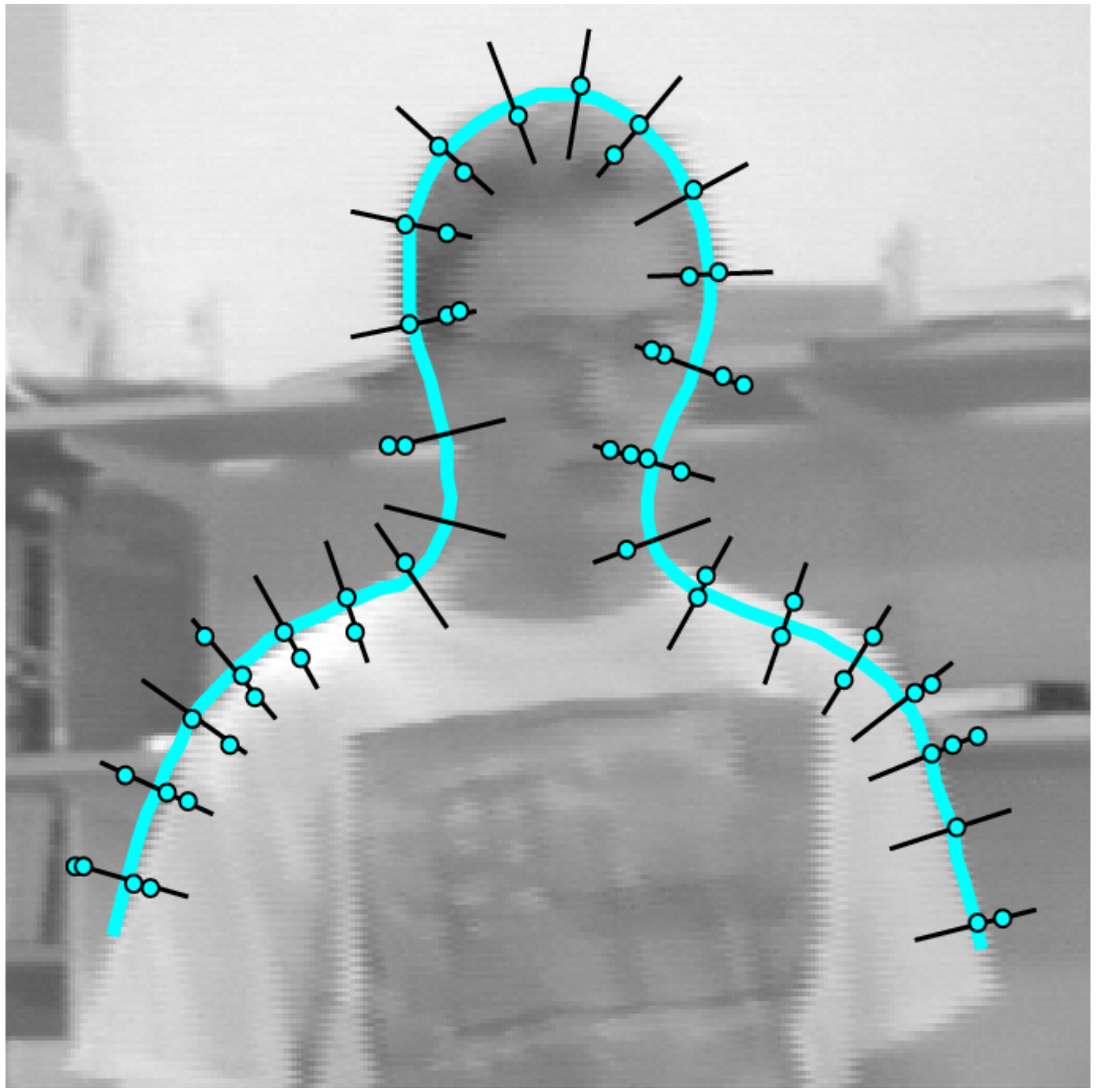


[Michael Isard]

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 1,000s or even 10,000s 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. Statistics and Computing, vol. 10, pp. 197-- 208, 2000.

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]

Tracking the unpredictable…

- They called it “dance”

[Michael Isard]

Summary

- Particle filtering is a very general tool for temporal inference that we can exploit for tracking.
  - Nonetheless, it applies in a variety of other applications as well.
  - It has problems in high-dimensional spaces, however, but there are a number of variants that alleviate some of these issues.
- Human tracking (“Marker-less mocap”) can be performed using particle filtering:
  - Wide range of applications, especially in entertainment.
  - Only a small part of the problems is solved to date.

- For those interested in the Kalman filter:
  - Greg Welch and Gary Bishop: An Introduction to the Kalman Filter
- Background for Particle Filtering:
  - Simon Maskell and Neil Gordon: A Tutorial on Particle Filtering for On-Line Nonlinear/Non-Gaussian Bayesian Tracking