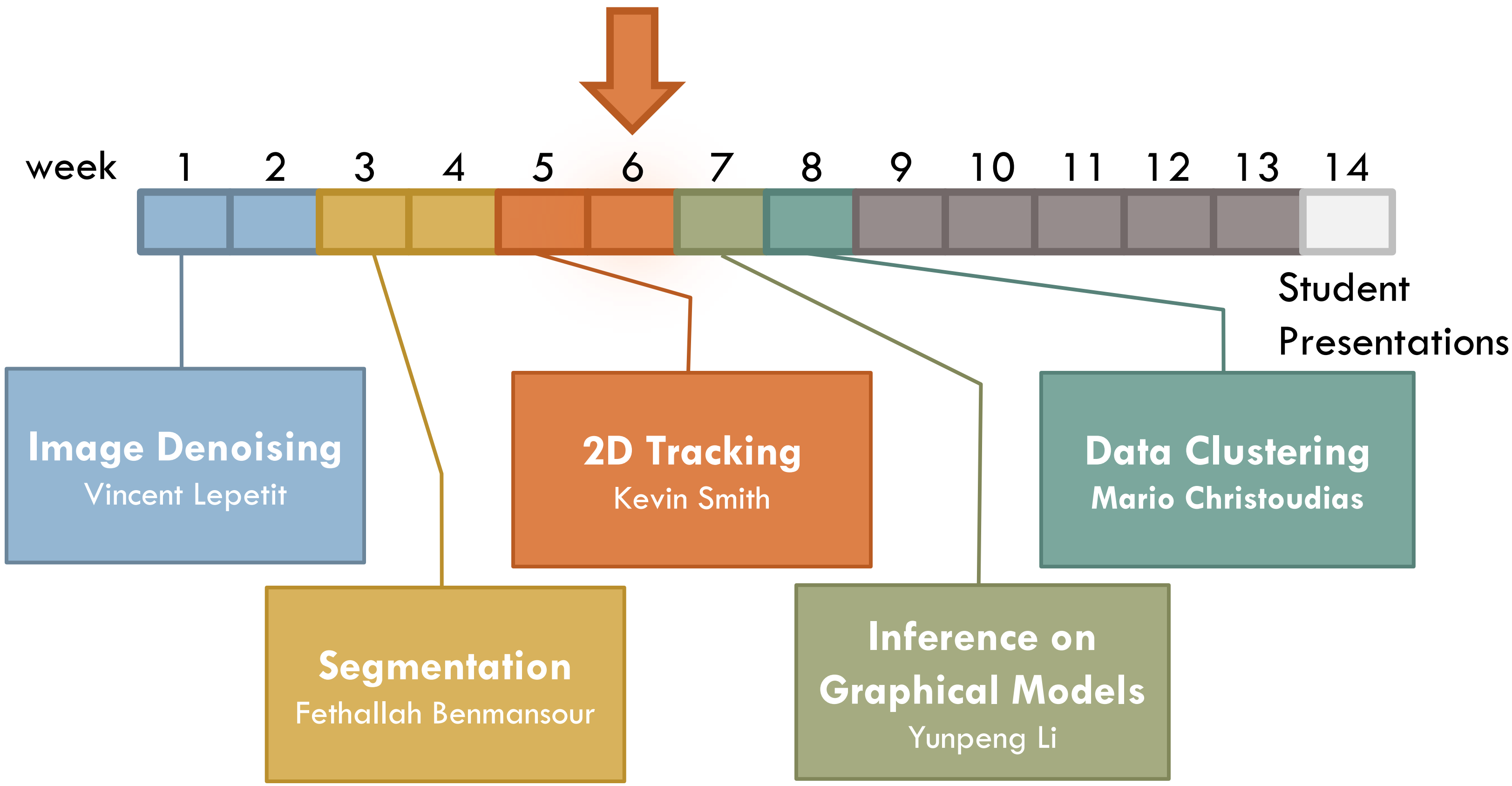
COM-711
SELECTED TOPICS IN COMPUTER VISION
2D TRACKING PART 2/2

Oct 28, 2011

Kevin Smith kevin.smith@lmc.biol.ethz.ch

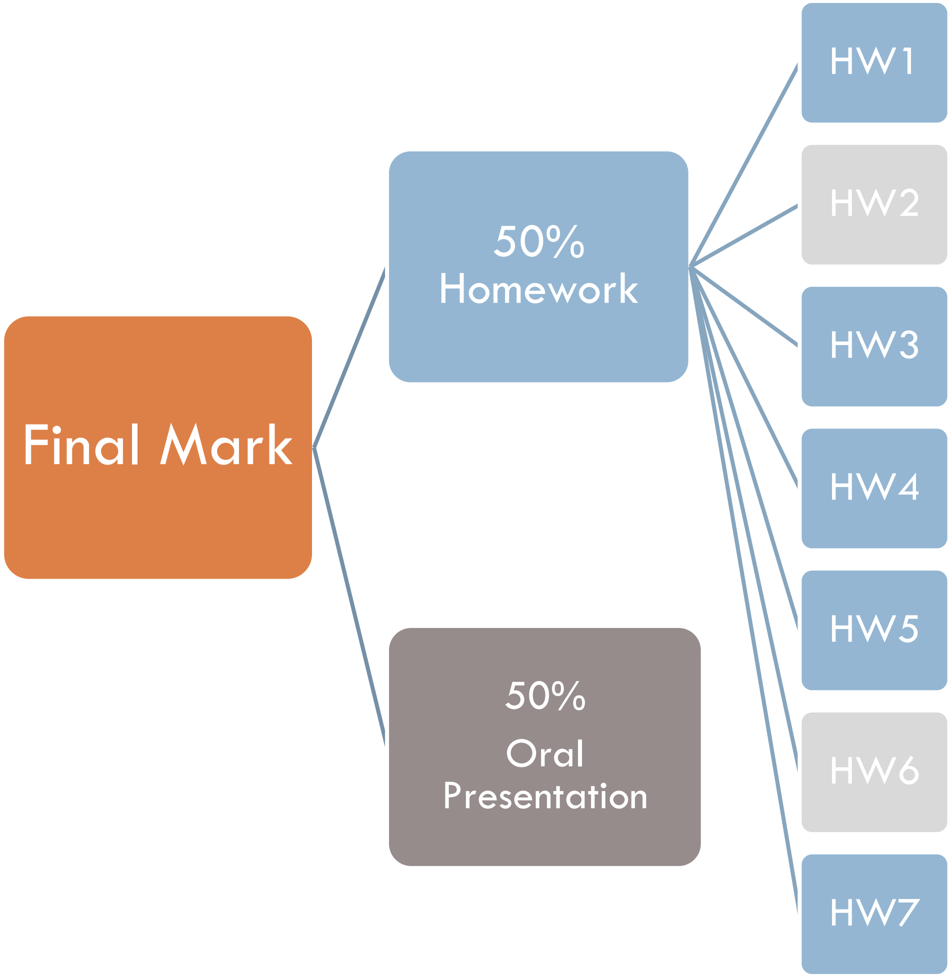
Course update

- Lectures



Course update

- Final mark
  - 50% homework
  - 50% presentation
- Homework mark
  - Considers only the best(*N*-2) scores from *N* total assignments

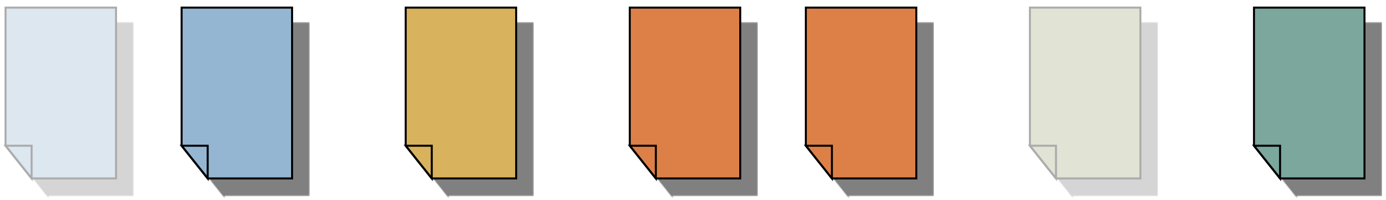


Oral presentations

- Each student **will present a published paper** on topics covered in the course to the rest of the class
  - Each student has approx **20 minutes** to speak (including questions)
  - A list of papers you may select from will be posted on the web site http://cvlab.epfl.ch/teaching/topics/index.php
  - Alternatively, you may propose a paper to present (subject to approval)
  - Instructors and other students will ask questions about the work
  - Presentations will be held during on Nov 18, Nov 25, Dec 2, Dec 9, Dec 16. Time slots will be assigned on a first-come-first serve basis, after the list is posted. A web site will be made available to sign up with your selected paper and time slot

Course update

- Homework: 7 total assignments



- Only the best N-2 scores from N total assignments will be considered (you can "skip" two assignments)

- First 4 assignments available on course web site
  http://cvlab.epfl.ch/teaching/topics/index.php

Outline

Introduction to the tracking problem

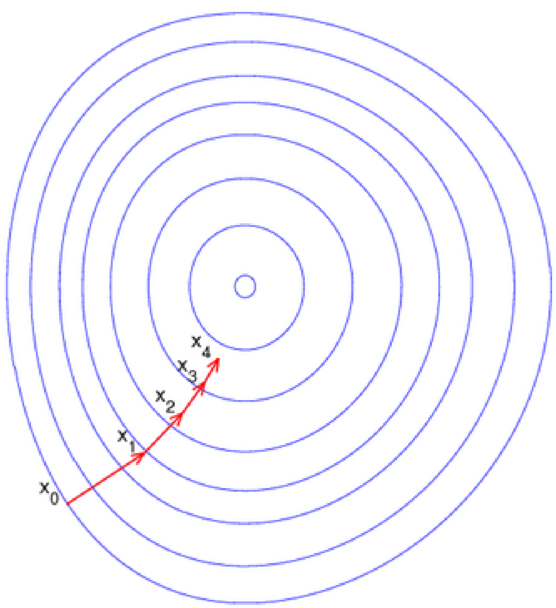
- What is tracking?
- Approaches , assumptions, &applications
- State of the art & challenges

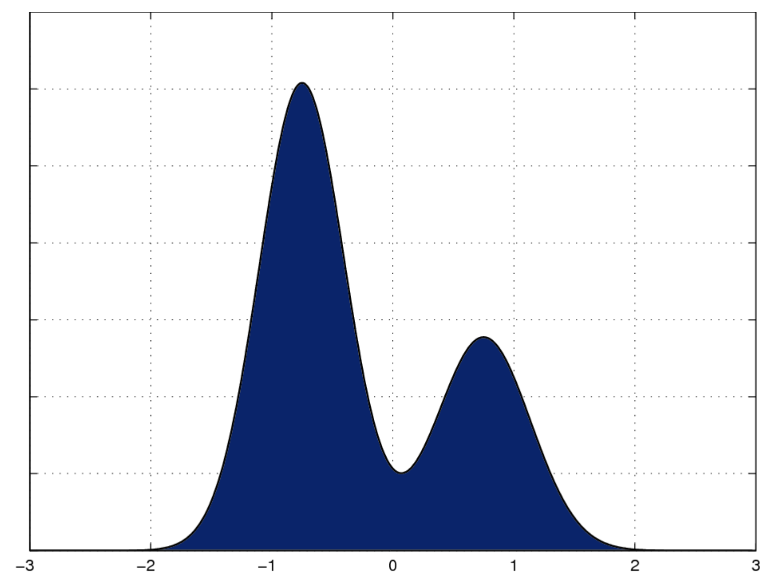
Recursive Bayesian filtering

- Background & formulation
- Kalman filter
- Particle filter

Recap: approaches to tracking

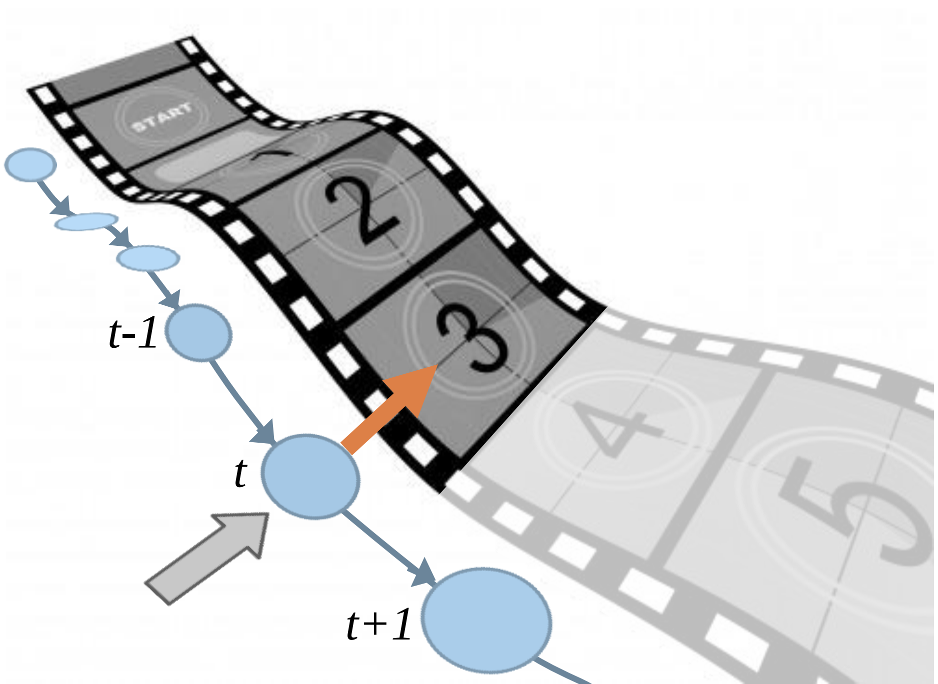
- Non-probabilistic
  - + quick convergence*
  - + efficient
  - - stuck in local max/min
  - - modeling multiple objects
- Probabilistic
  - + flexible, principled
  - + multi-modal
  - - slower
  - - interpretation





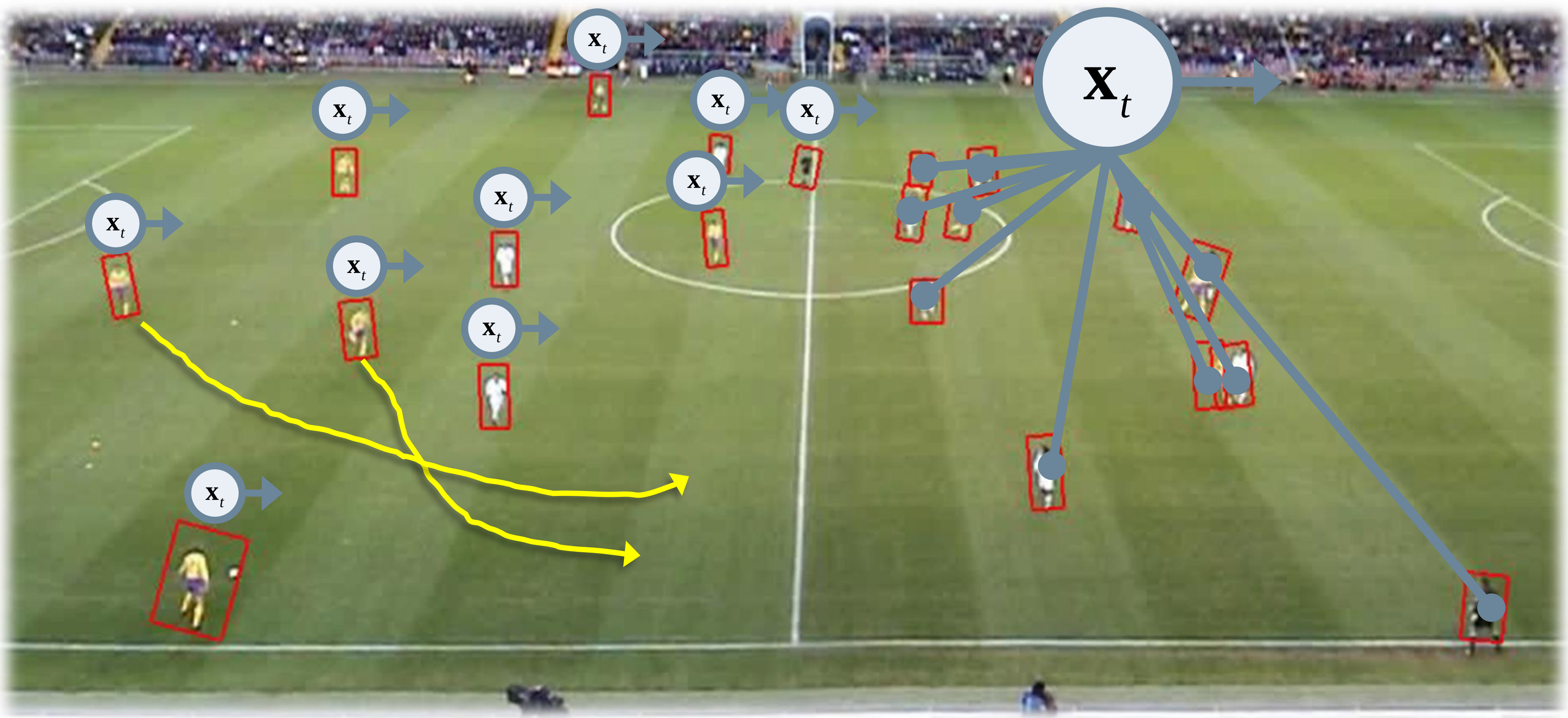
Recap: approaches to tracking

- Sequential
  - (recursive, online)
  - + Inexpensive ⟶ real-time
  - - no future information
  - - cannot revisit past errors
- Batch Processing
  - (offline)
  - - Expensive ⟶ not real-time*
  - + considers all information
  - + can correct past errors



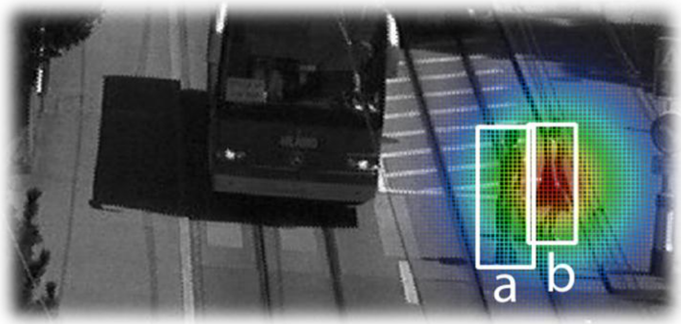
Recap: approaches to tracking

- Parallel trackers
  - several single-object trackers
  - computationally less expensive
  - ad-hoc interaction
- Joint state
  - single multi-object representation
  - computationally expensive
  - explicit principled interaction

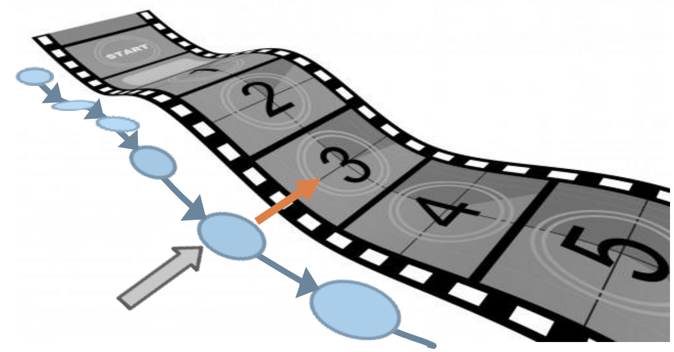


Recap: recursive Bayesian filtering

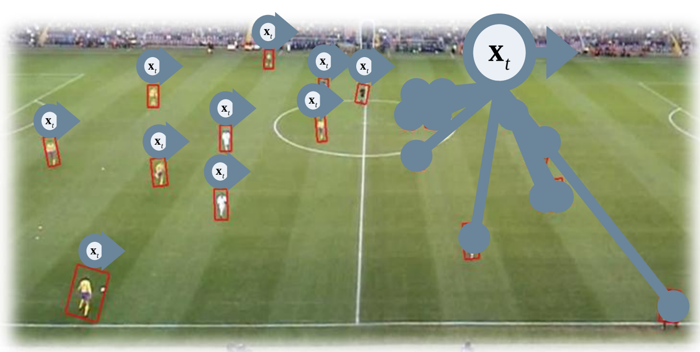
- **Probabilistic Formulation**



- **Sequential**



- **Multiple Objects**



Recap: recursive Bayesian filtering

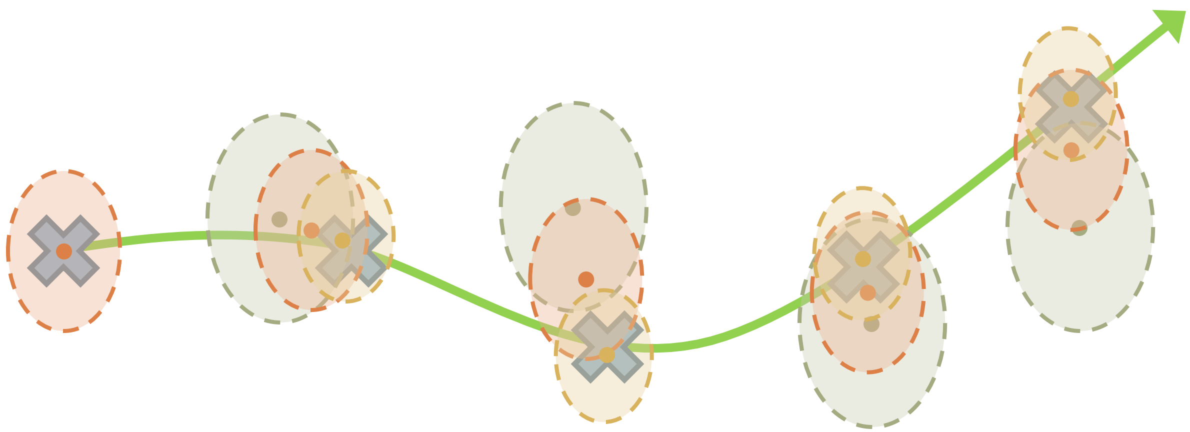
- **Filtering equation**

$$\underbrace{p(\mathbf{x}_t \mid Z_t)}_{\text{posterior estimate}} \propto \underbrace{p(\mathbf{z}_t \mid \mathbf{x}_t)}_{\text{likelihood or observation model}} \int_{\mathbf{x}_{t-1}} \underbrace{p(\mathbf{x}_t \mid \mathbf{x}_{t-1})}_{\text{motion model or dynamic model}} \underbrace{p(\mathbf{x}_{t-1} \mid Z_{t-1})}_{\text{posterior estimated at t-1}}$$

- Definitions
  - State from 1 to time t: $X_t = \{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t\}$
  - Observations from 1 to time t: $Z_t = \{\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}, \mathbf{z}_t\}$

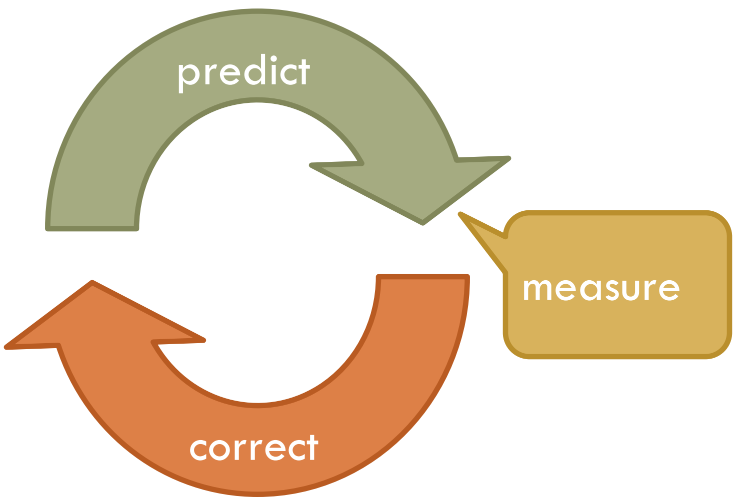
Recap: recursive Bayesian filtering

- **Key idea 1.** PDFs represent our belief as to the state of the object



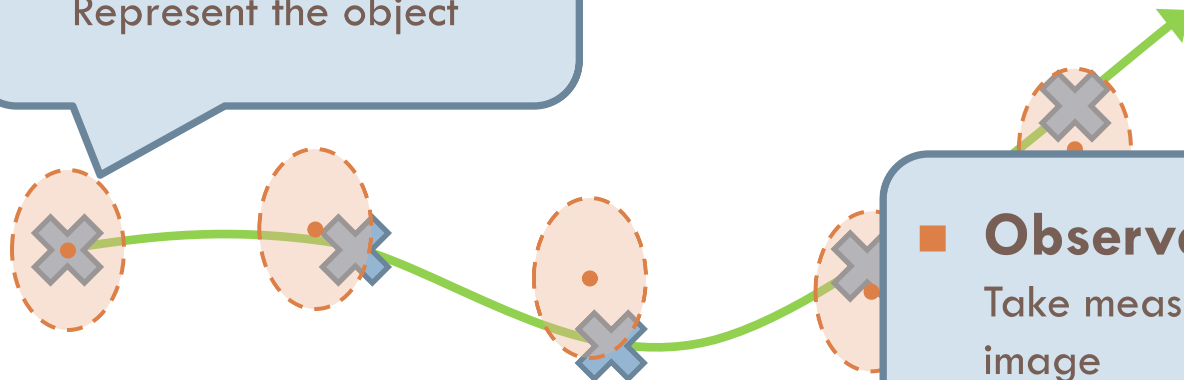
- **Key idea 2:** Recursive cycle
  1. **Predict** from motion model
  2. **Measurement** from image
  3. **Correct** the prediction

  ...repeat



Recap: recursive Bayesian filtering

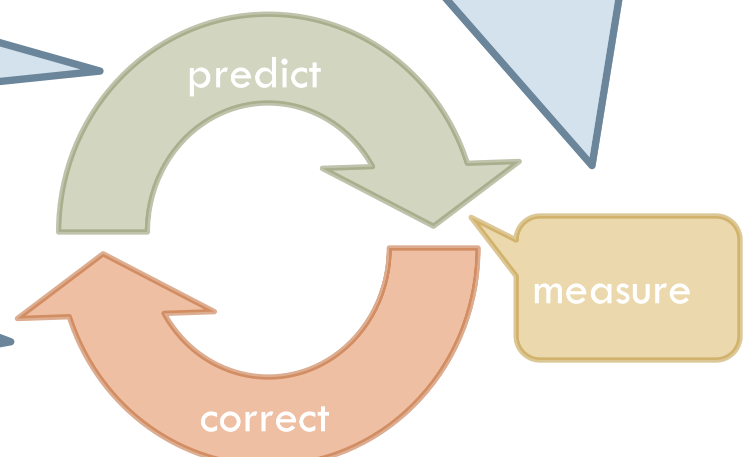
- **State definition**
  Represent the object



- **Observation model**
  Take measurements from the image

- **Dynamic model**
  Predict the next state given current state

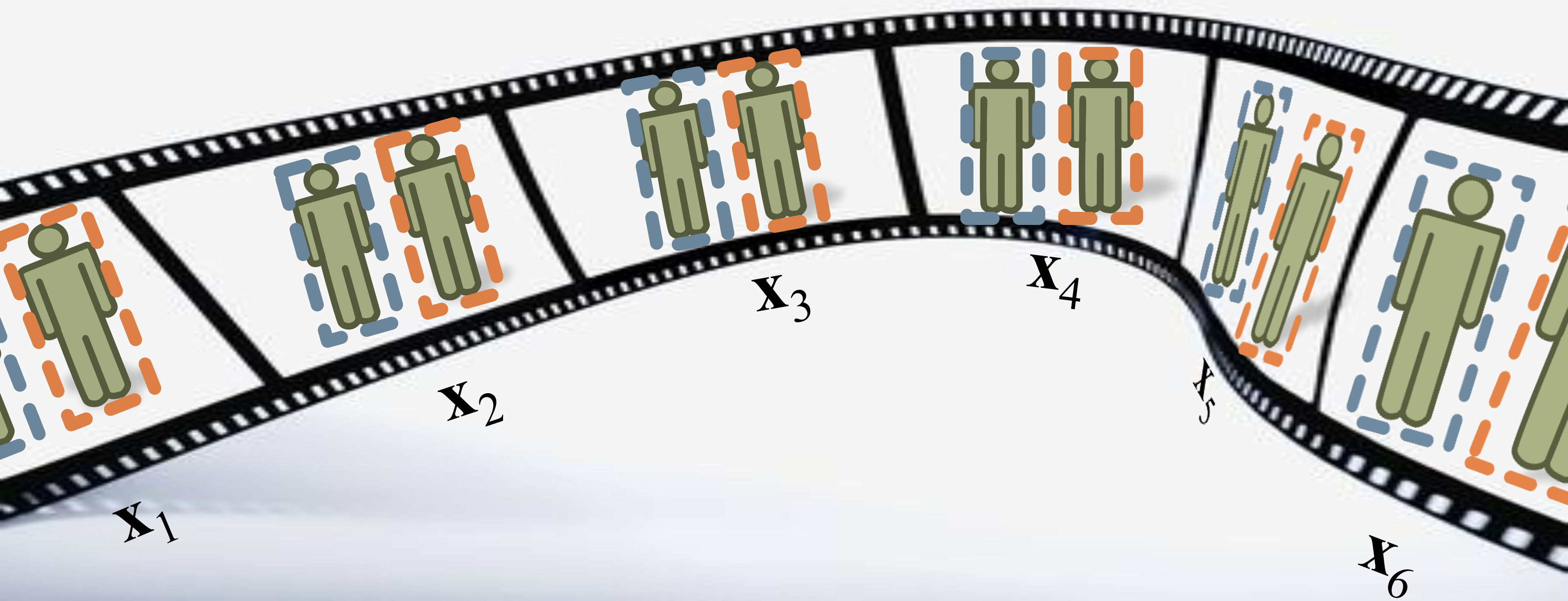
- **Inference method**
  Solution estimation



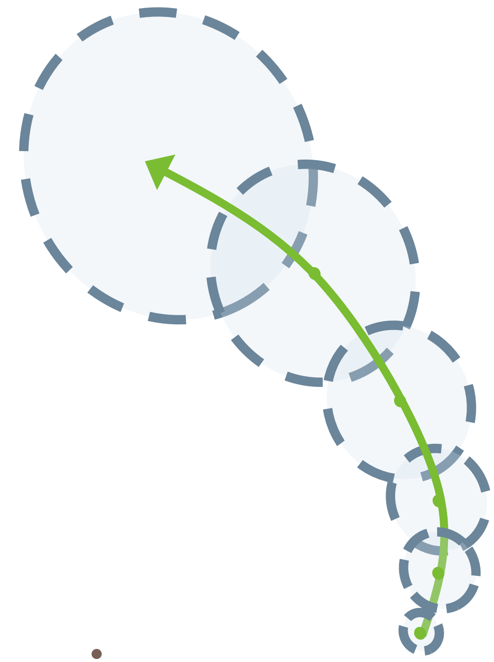
Recap: state definition

- State vector $\mathbf{x}_t$ describes object(s) at an instant in time
- Defines solution space

$$X_t = \{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t\}$$



Recap: dynamic model



- Predicts new state $\mathbf{x}_t$ based on previous state $\mathbf{x}_{t\text{-}1}$

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = N\left(\mathbf{F}_t \mathbf{x}_{t-1}, \Sigma_{F_t}\right)$$

- Often a linear transform + Gaussian noise

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{w}_t$$

Predicted state time t

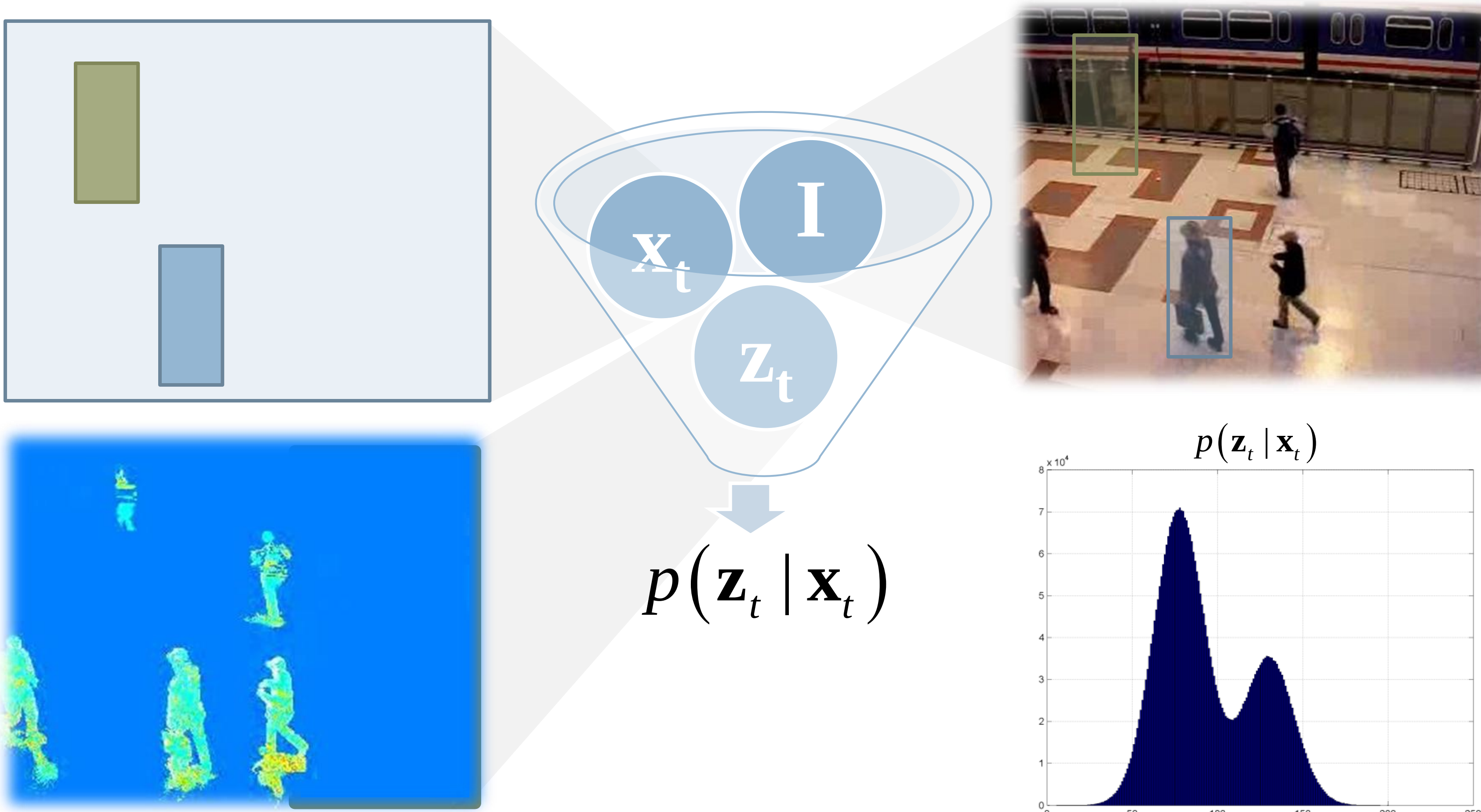
State transition model

Previous state time $t\text{-}1$

Gaussian noise $\mathbf{w}_{\mathrm{t}} \sim N(0, \mathbf{Q}_t)$

Recap: observation model

- Models the likelihood that a state estimate $\mathbf{x}_t$ gave rise to the observed image data $\mathbf{z}_t$



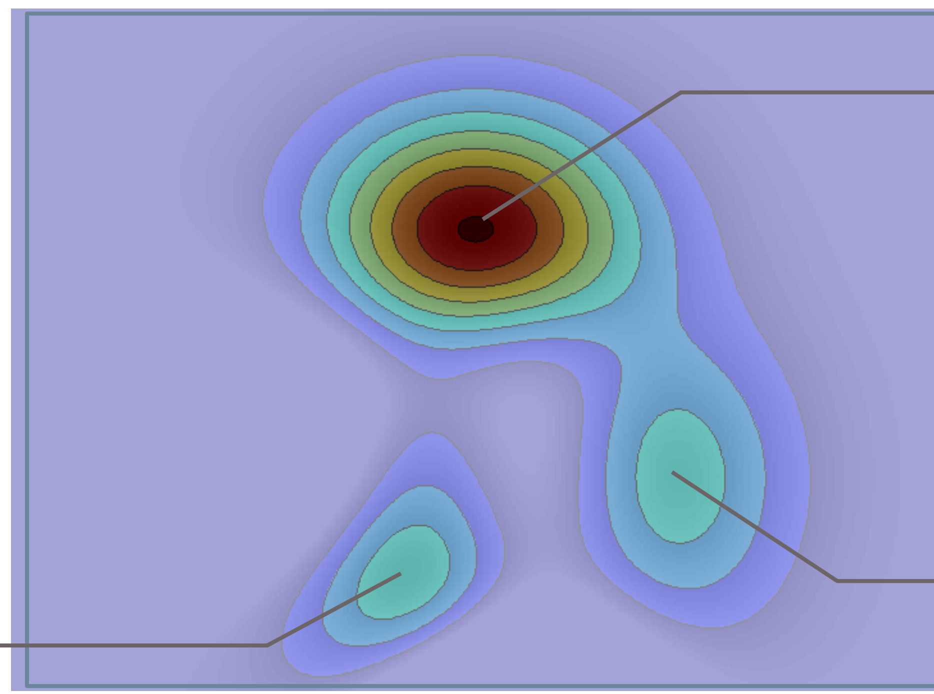
Probability distribution to model belief in object location

- Tracking faces in frame t

Probability distribution to model belief in object location

- *Posterior* or *target* distribution – models belief as to the state of the system given the observations up to t

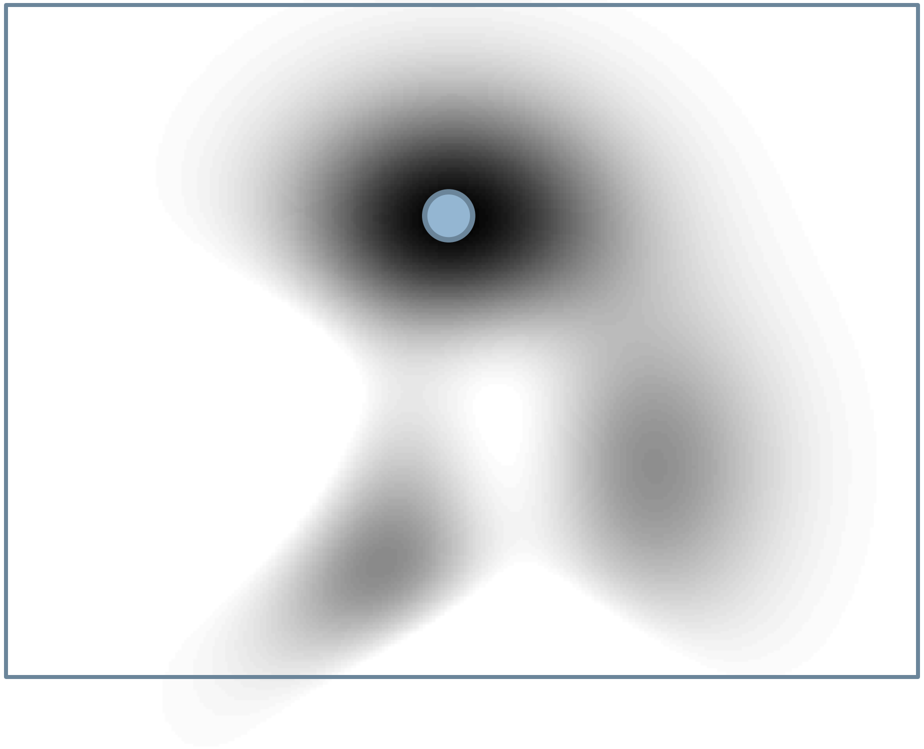
- $p(\mathbf{x}_t \mid Z_t)$



Representing the posterior

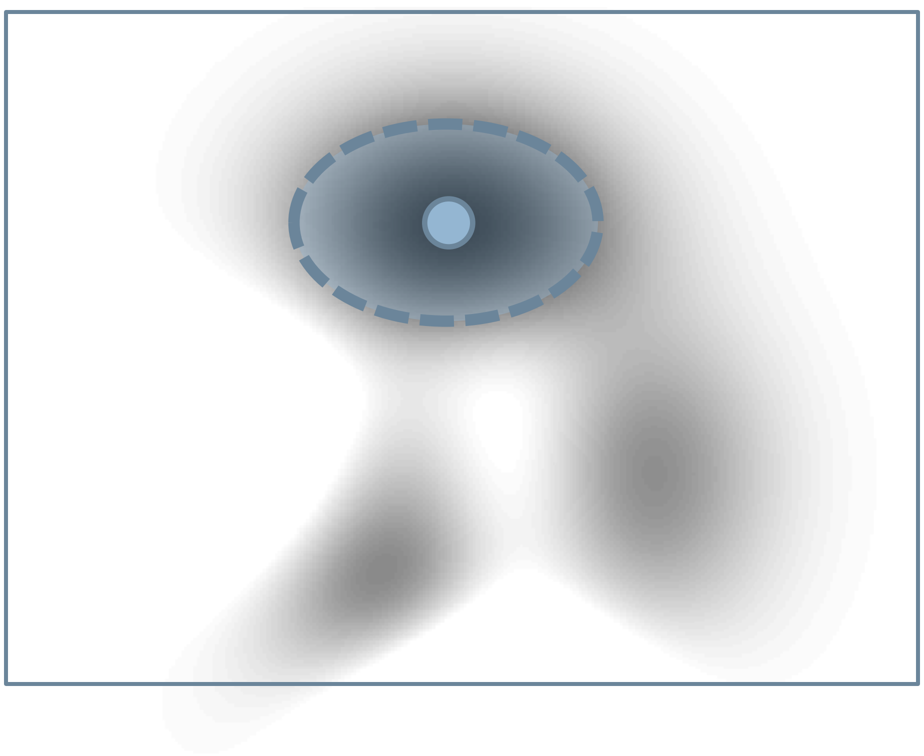
- A point (dirac)

$$p\left(\mathbf{x}_t \mid Z_t\right) = \begin{cases} 1 & \text{if } \mathbf{x}_t = \mu \\ 0 & otherwise \end{cases}$$



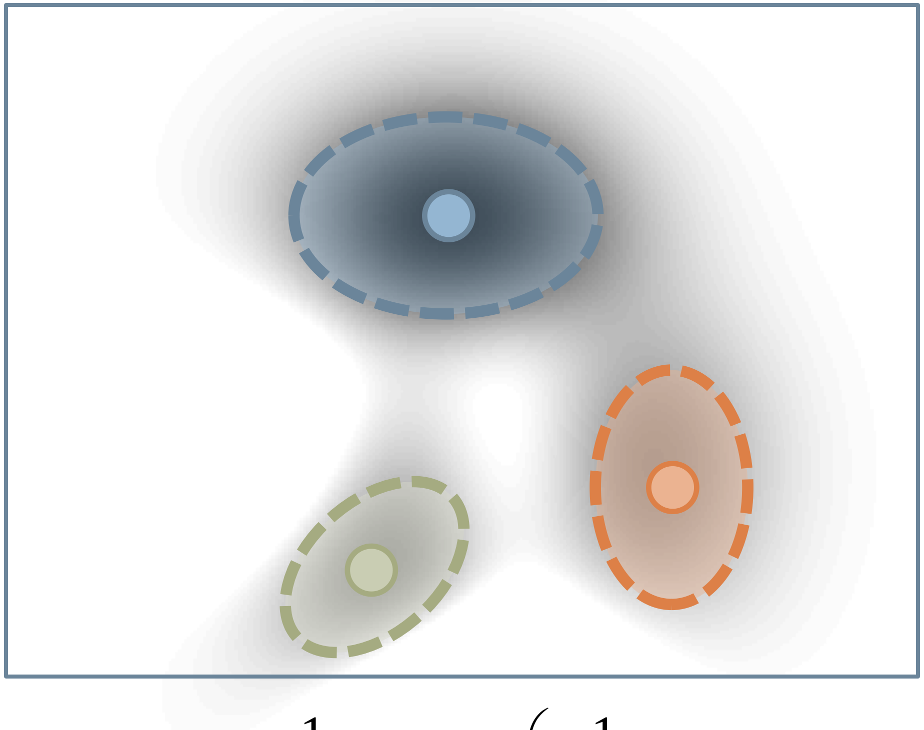
Representing the posterior

- Gaussian $p\left(\mathbf{x}_t \mid Z_t\right) = N\left(\mu, \Sigma\right) = \dfrac{1}{\sqrt{(2\pi)^n \mid \Sigma \mid}} \exp\left(-\dfrac{1}{2}(\mathbf{x}_t - \mu)^{\mathrm{T}} \Sigma^{-1} (\mathbf{x}_t - \mu)\right)$



Representing the posterior

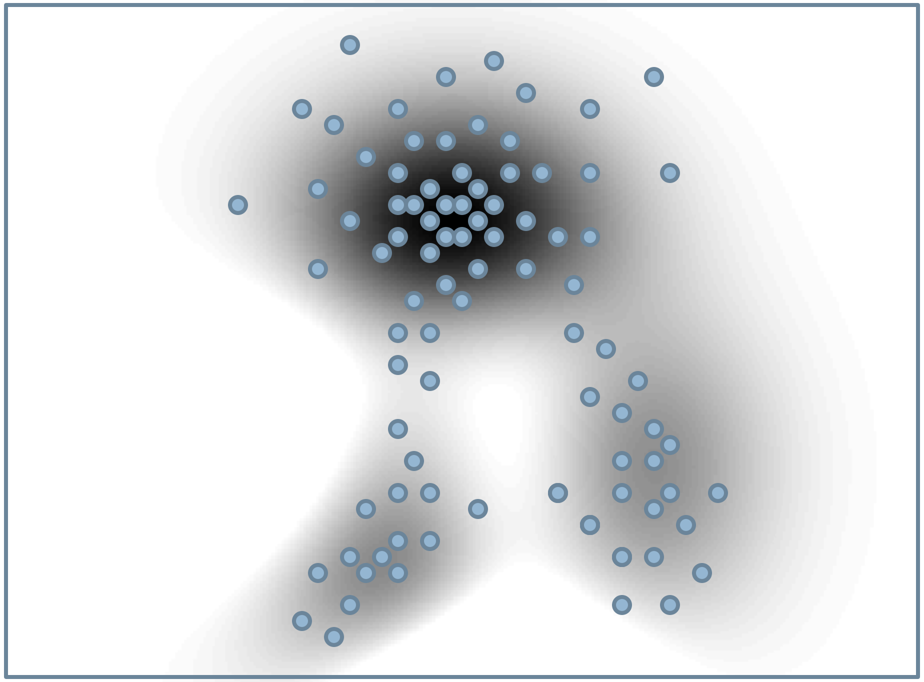
- Mixture of Gaussians $\left\{ \left(\mu_1, \Sigma_1\right), \left(\mu_2, \Sigma_2\right), \ldots \right\}$



$$p\left(\mathbf{x}_t \mid Z_t\right) \propto \sum_i \frac{1}{\sqrt{(2\pi)^n \mid \Sigma_i \mid}} \exp\left(-\frac{1}{2} (\mathbf{x}_t - \mu_i)^{\mathrm{T}} \Sigma_i^{-1} (\mathbf{x}_t - \mu_i) \right)$$

Representing the posterior

- Set of discrete samples (particles) $\left\{x_t^{(n)}, n=1,\ldots,N\right\}$

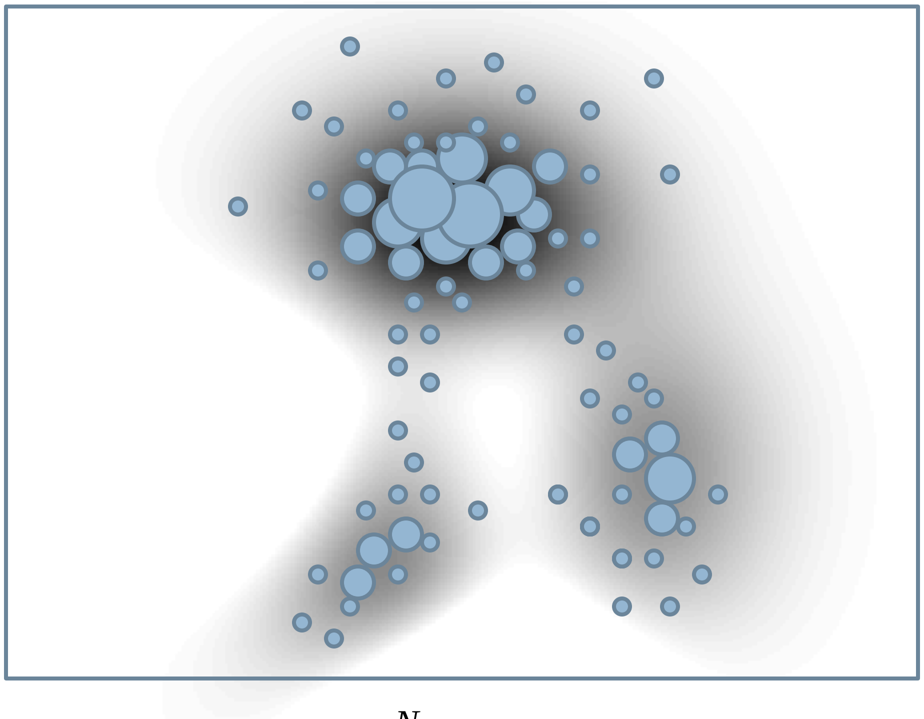


$$p\left(\mathbf{x}_t \mid Z_t\right) \approx \sum_{n=1}^{N} \delta\left(x_t - x_t^{(n)}\right)$$

Representing the posterior

- Set of weighted samples (particles) $\left\{x_t^{(n)}, w_t^{(n)}\right\}_{n=1}^{N}$

$$w_t^{(n)} \in [0,1] \qquad \sum_n w_t^{(n)} = 1$$



$$p\left(\mathbf{x}_t \mid Z_t\right) \approx \sum_{n=1}^{N} w_t^{(n)} \, \delta\left(x_t - x_t^{(n)}\right)$$

Recursive Bayesian filtering

- Models belief about the current state X_t given past and present observed data $Z_{1:t}$.

Kalman filter

exact solution

[1] Kalman, R.E. *A new approach to linear filtering and prediction problems.* ASME, Journal of Basic Engineering, 1960.

SIR particle filter

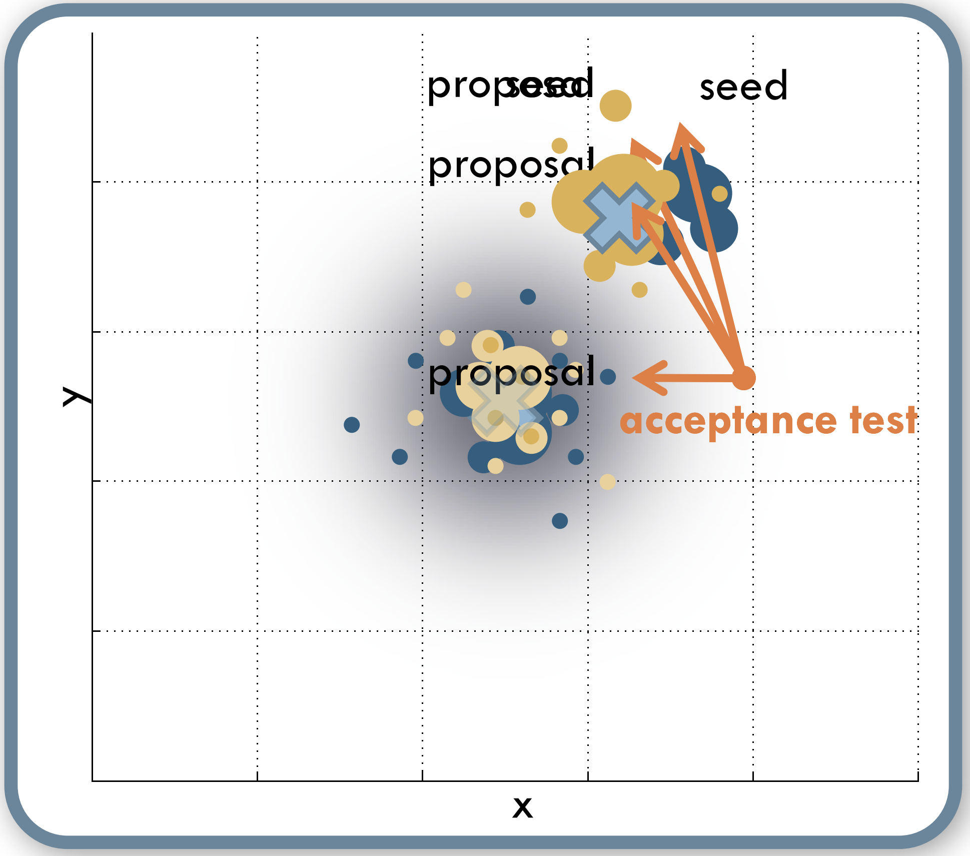
discrete approximation

[2] M. Isard and A. Blake. *Condensation*, International Journal of Computer Vision, 1998.

MCMC particle filter

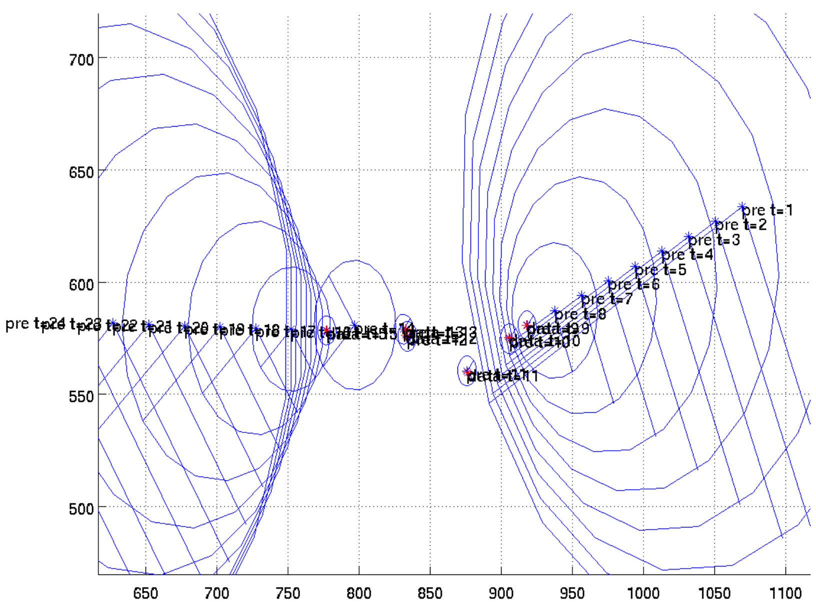
discrete approximation

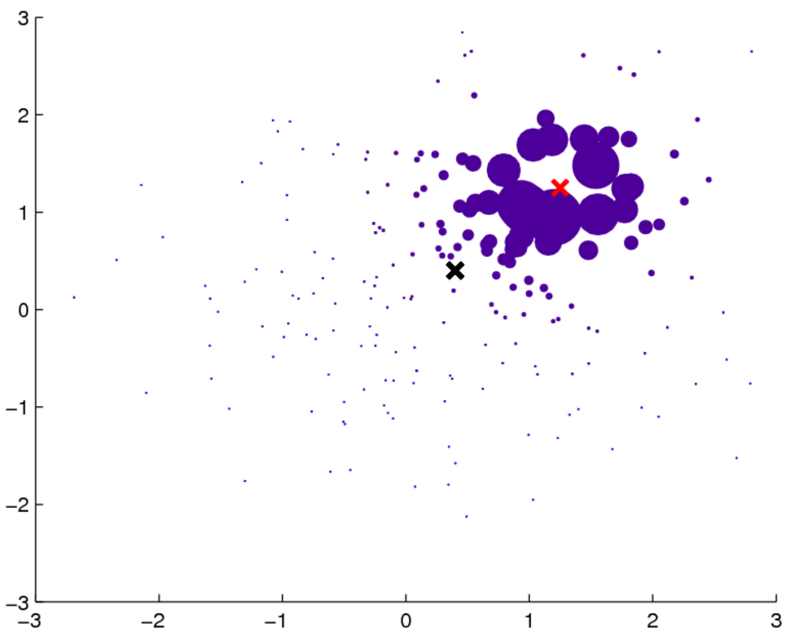
[3] Z. Khan, T. Balch, and F. Dellaert, *An MCMC-based particle filter for tracking multiple interacting targets*, ECCV, 2004.



Recursive bayesian filtering

- Kalman filter
  exact solution
  - Continuous state space
  - Linear dynamics
  - Gaussian observation density
- Particle filter
  approximate solution
  - Continuous, discrete, or mixed state space
  - Arbitrary dynamics
  - Arbitrary observation density





Outline

Introduction to the tracking problem

- What is tracking?
- Approaches , assumptions, &applications
- State of the art & challenges

Recursive Bayesian filtering

- Background & formulation
- Kalman filter
- Particle filter

Kalman filter

- Published in 1960

  Kalman, R. E. 1960. "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).

- Used for many problems
  - Guidance
  - Navigation
  - Autopilots
  - Radar
  - Satellite
  - Weather forecasting

Kalman filter: Gaussians!

- in bayesian filtering terms
  - posterior

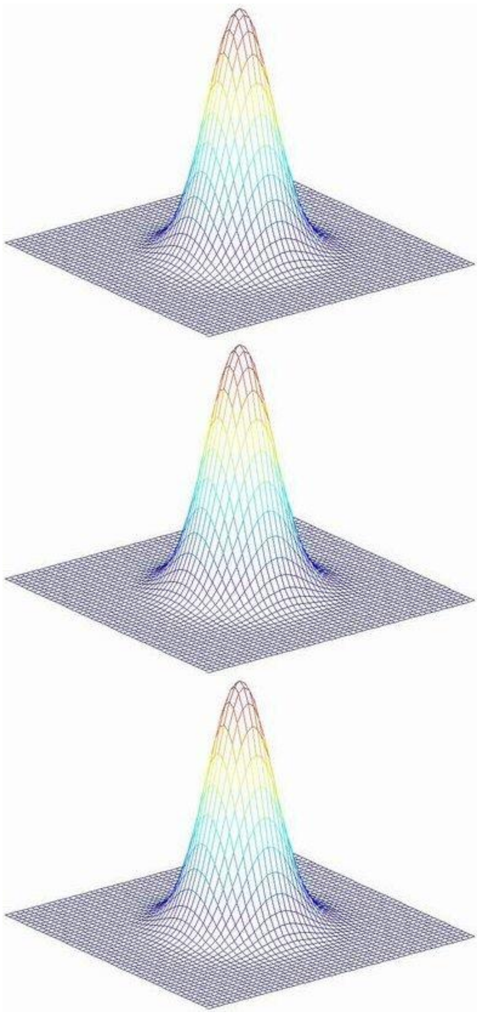
$$p\left(\mathbf{x}_t \mid Z_t\right) = N\left(\hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}\right)$$

  - motion model

$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = N\left(\mathbf{F}_t \mathbf{x}_{t-1}, \mathbf{Q}_t\right)$$

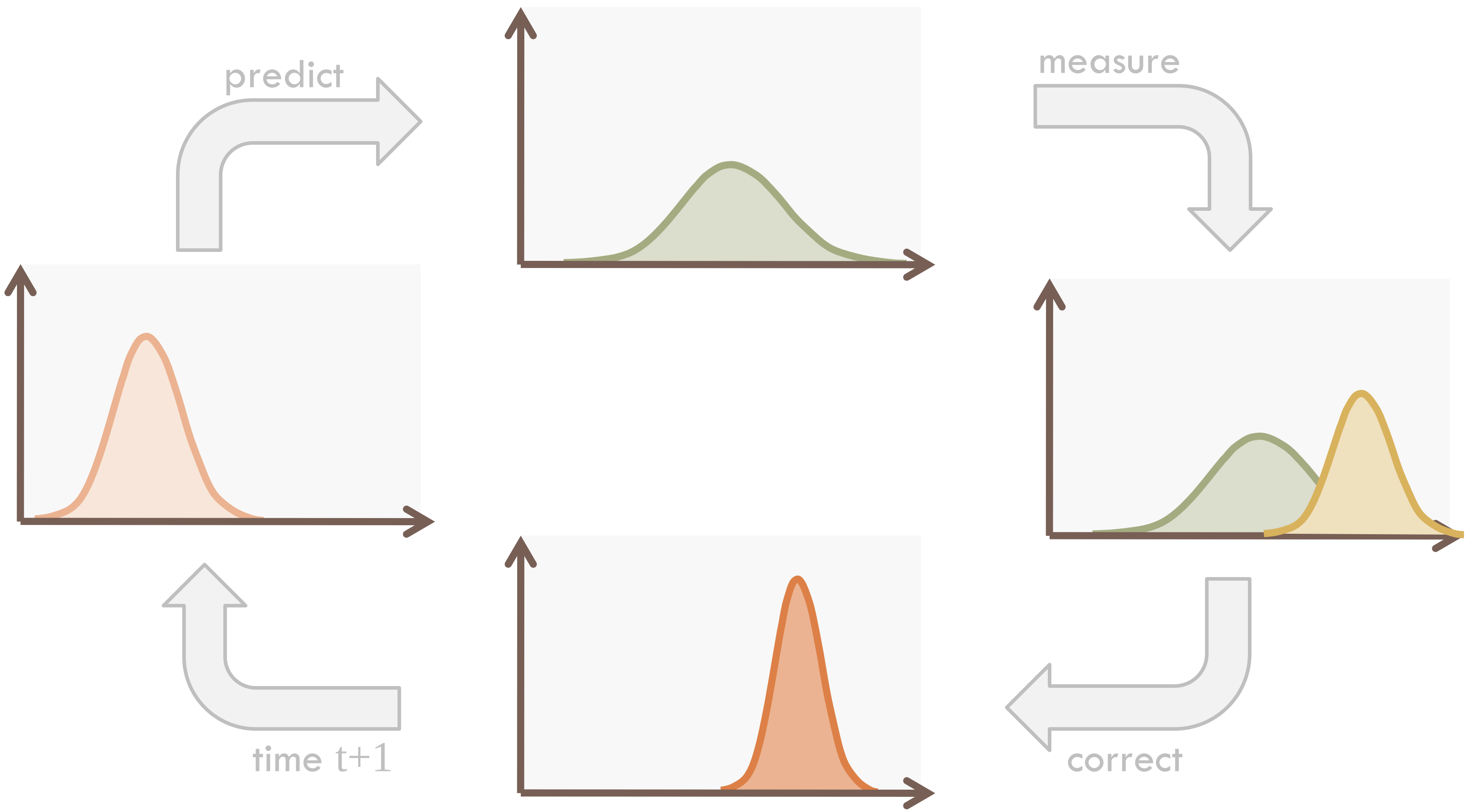
  - observation model

$$p\left(\mathbf{z}_t \mid \mathbf{x}_t\right) = N\left(\mathbf{H}_t \mathbf{x}_t, \mathbf{R}_t\right)$$

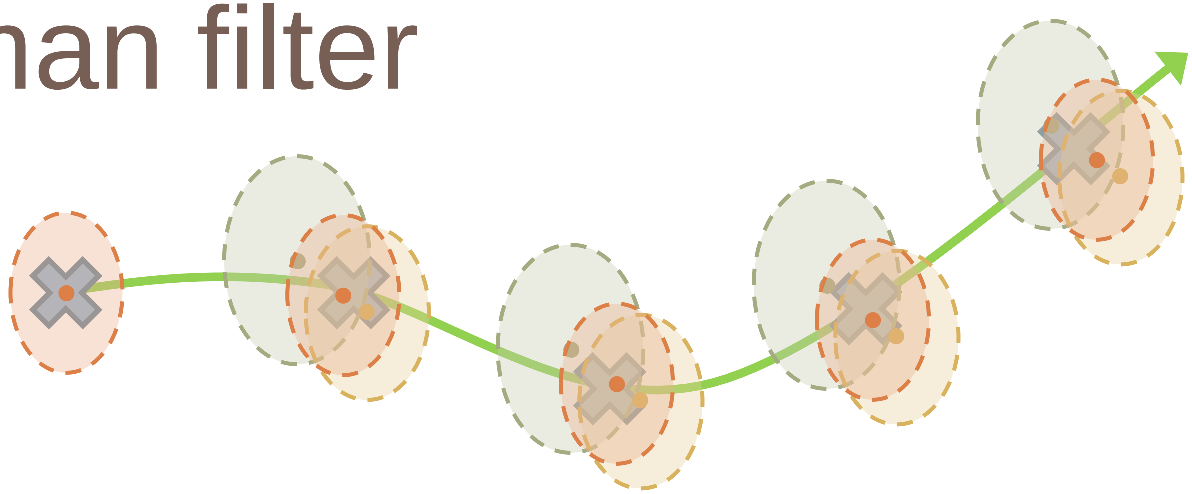


Probability density propagation

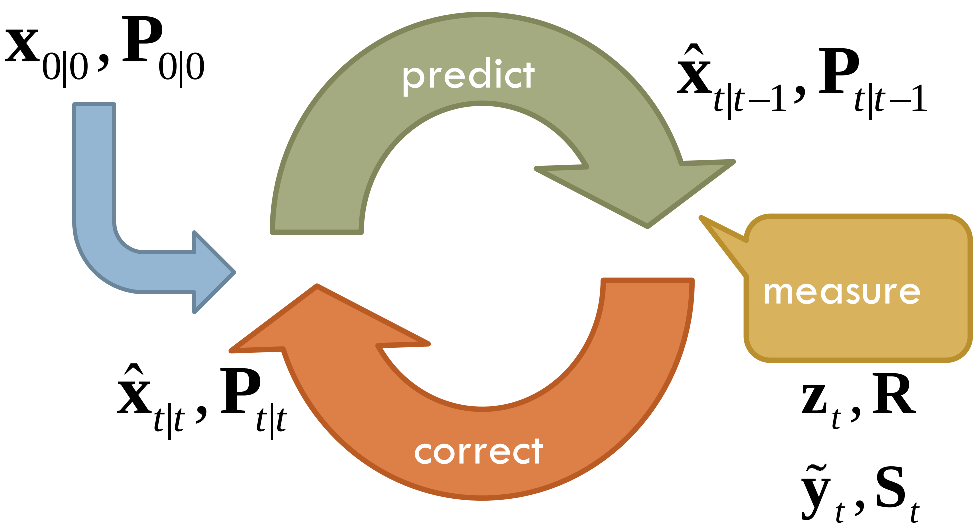
- Kalman filter uses Gaussians



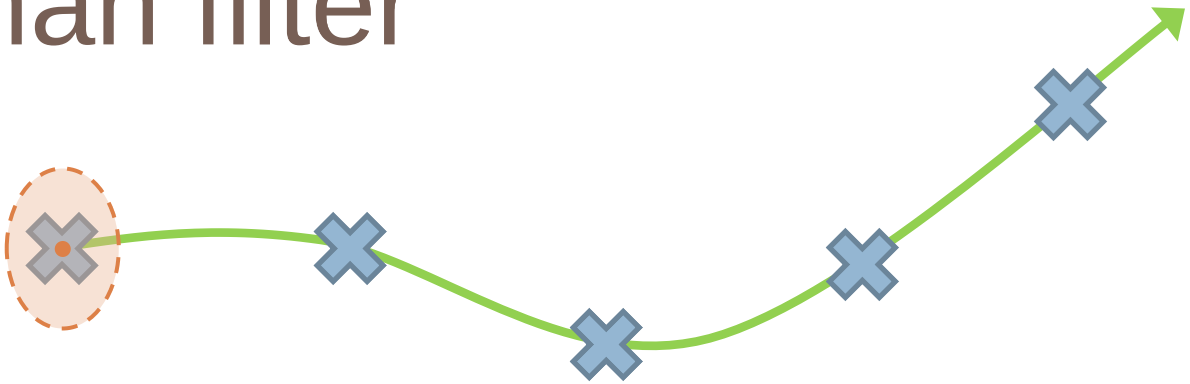
Kalman filter



- Predict, measure, correct cycle iteratively estimates the state at each time step



Kalman filter



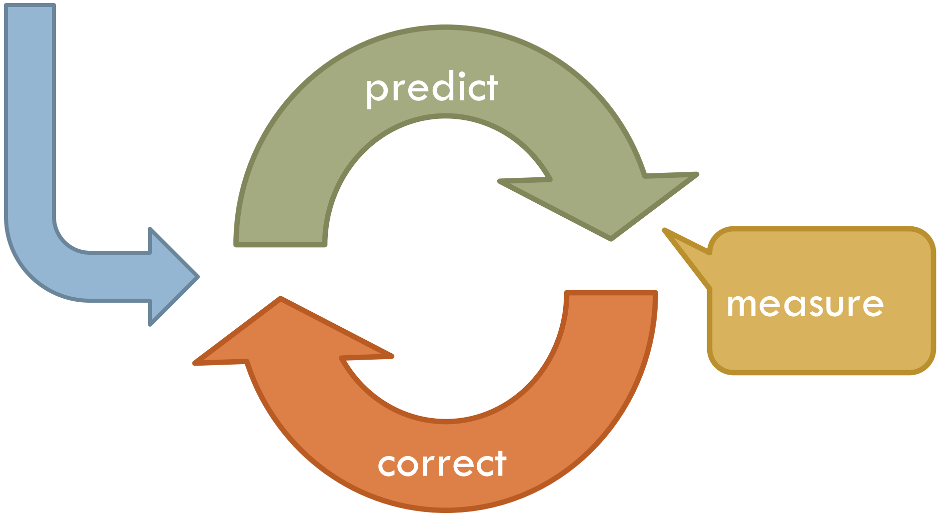
- State vector

$$\mathbf{x}_t = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}$$

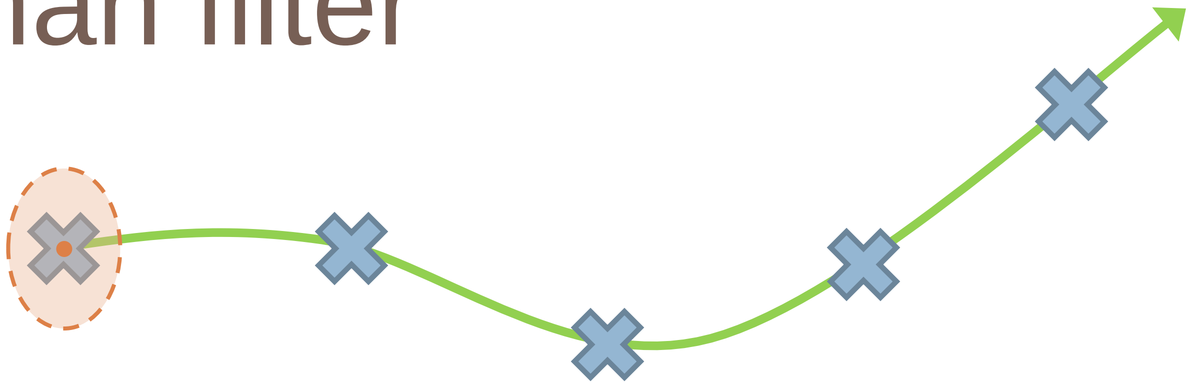
- Measurement

$$\mathbf{z}_t = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\mathbf{x}_{0|0}, \mathbf{P}_{0|0}$$



Kalman filter

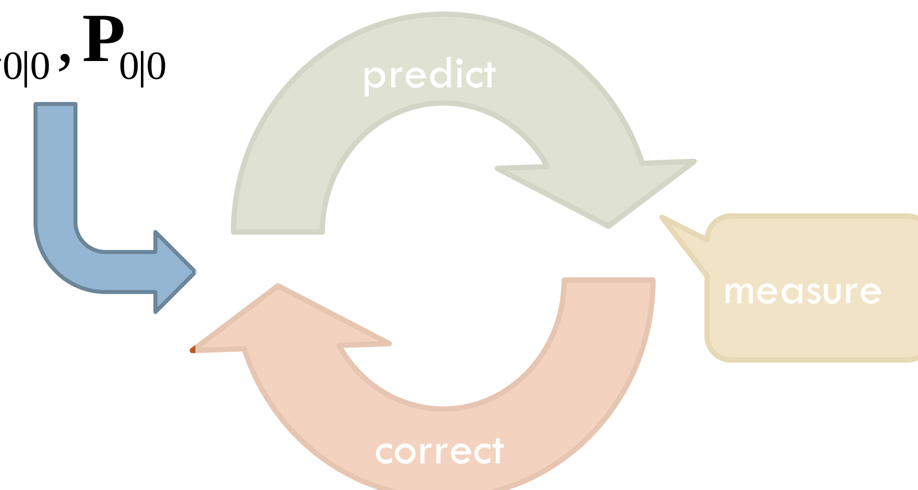


- Initial state

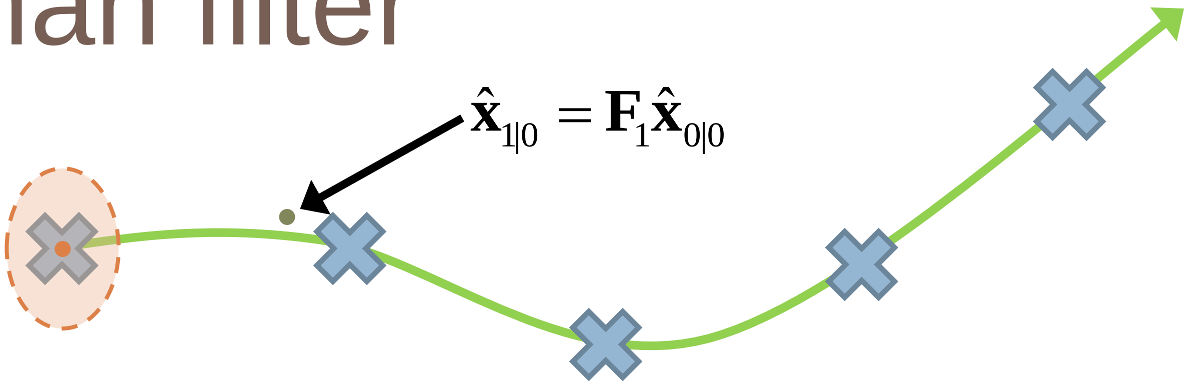
$$\mathbf{x}_{0|0} = \begin{pmatrix} x_0 \\ y_0 \\ \dot{x}_0 \\ \dot{y}_0 \end{pmatrix}$$

$$\mathbf{P}_{0|0} = \begin{pmatrix} L & & & \\ & L & & \\ & & L & \\ & & & L \end{pmatrix}$$

$\mathbf{x}_{0|0}, \mathbf{P}_{0|0}$



Kalman filter

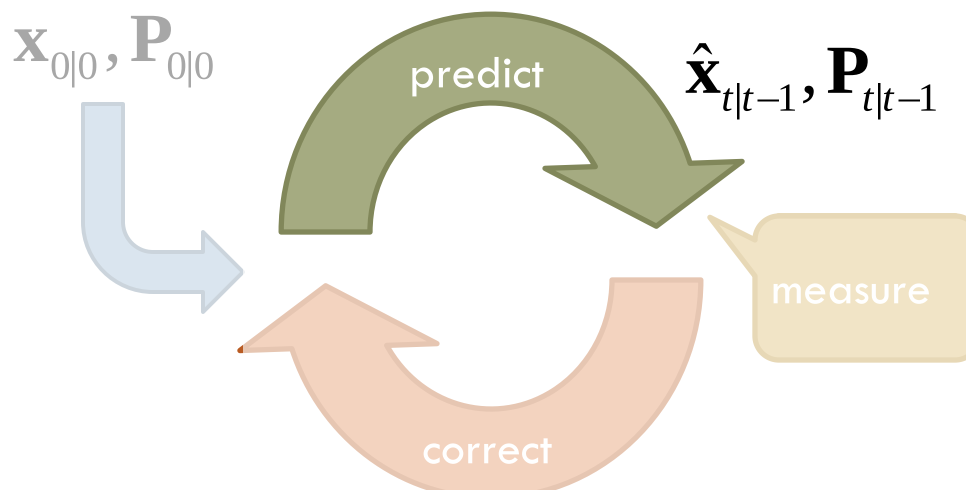


- Prediction from the motion model

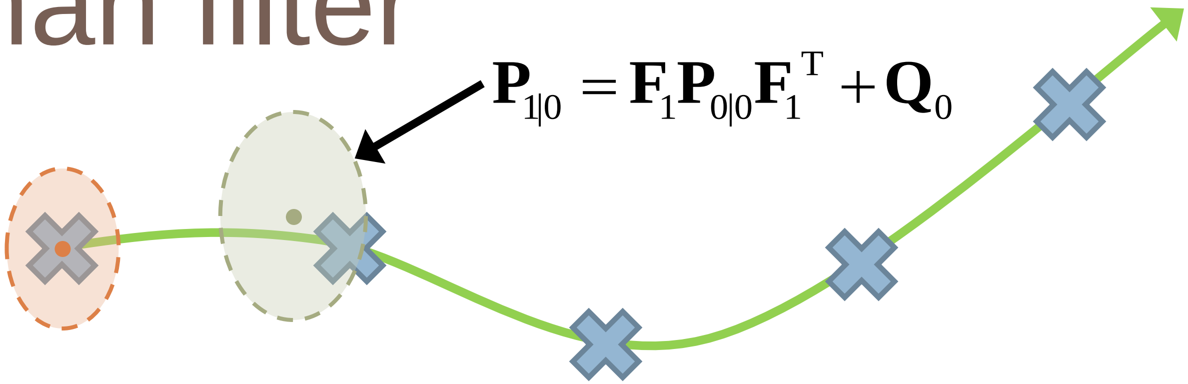
$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = N\left(\mathbf{F}_t \mathbf{x}_{t-1}, \mathbf{Q}_t\right)$$

- Update the mean

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1}$$



Kalman filter

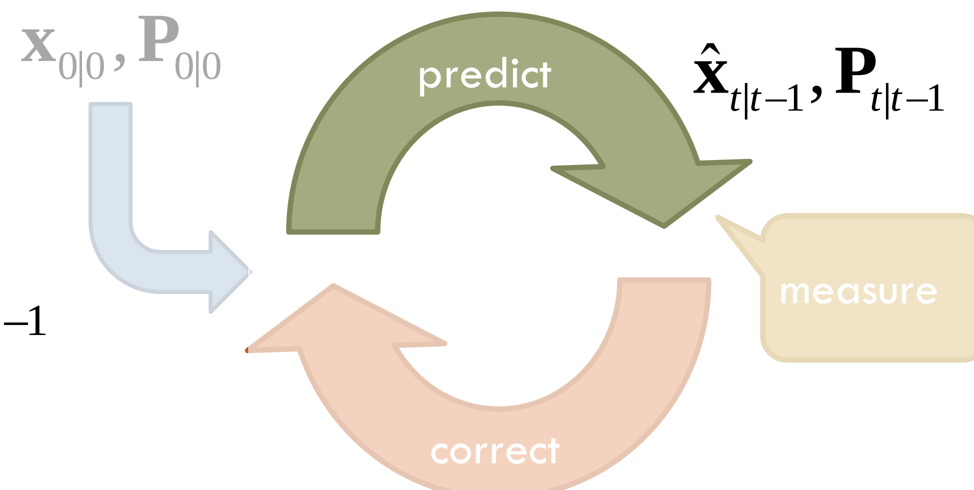


- Prediction from the motion model

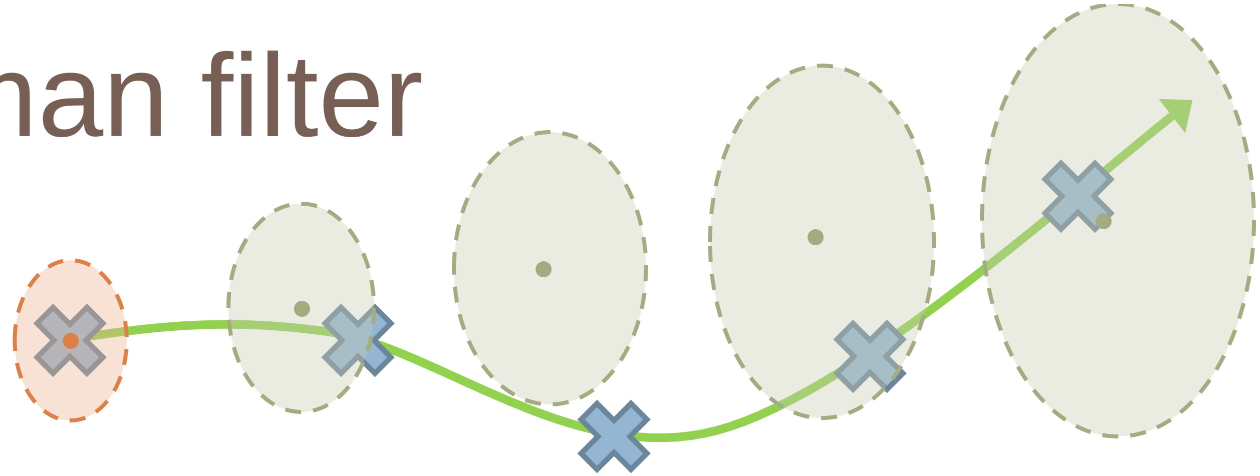
$$p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = N\left(\mathbf{F}_t \mathbf{x}_{t-1}, \mathbf{Q}_t\right)$$

- Update covariance

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^{\mathrm{T}} + \mathbf{Q}_{t-1}$$



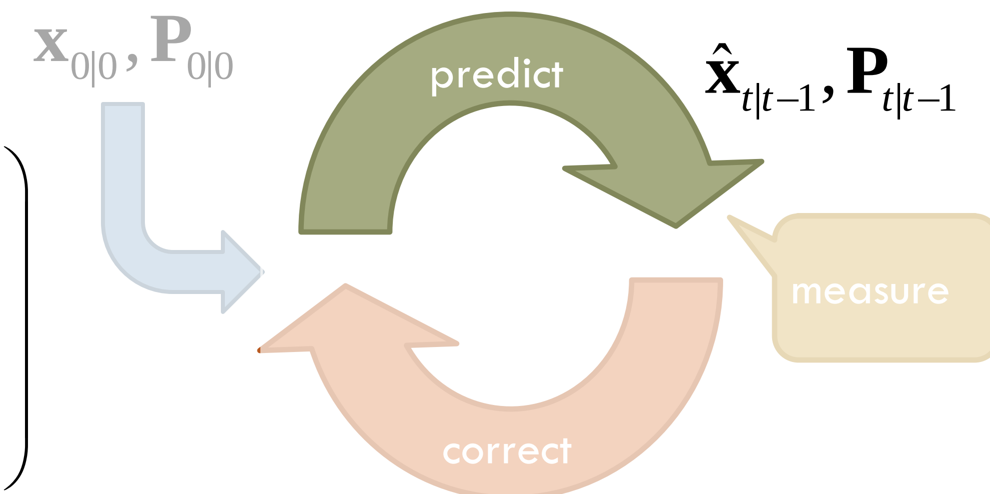
Kalman filter



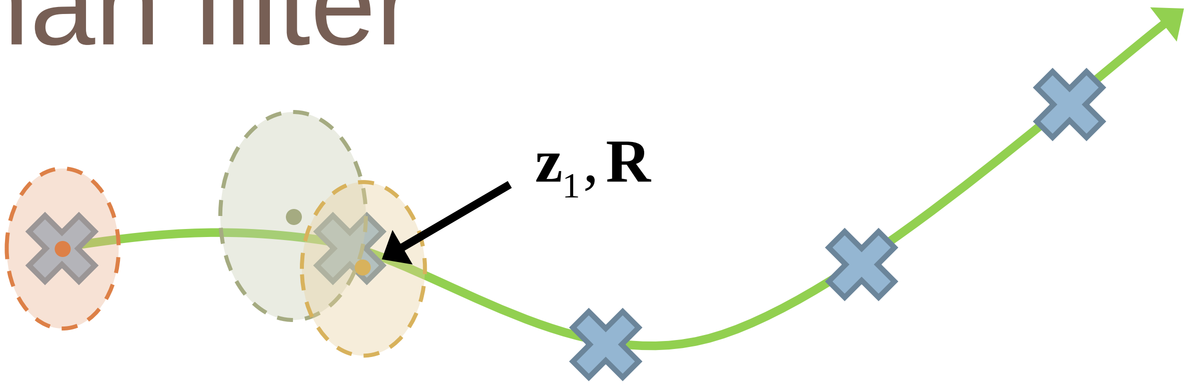
- Prediction from the motion model

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1}$$

$$\mathbf{x}_t = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \quad \mathbf{F}_t = \begin{pmatrix} 1 & & \Delta t & \\ & 1 & & \Delta t \\ & & 1 & \\ & & & 1 \end{pmatrix}$$



Kalman filter

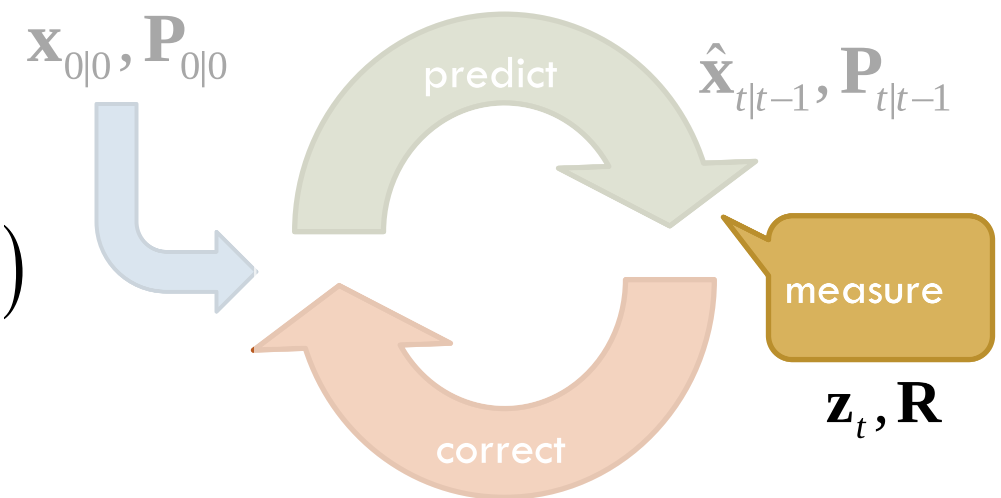


- Receive a *noisy* measurement (observation)

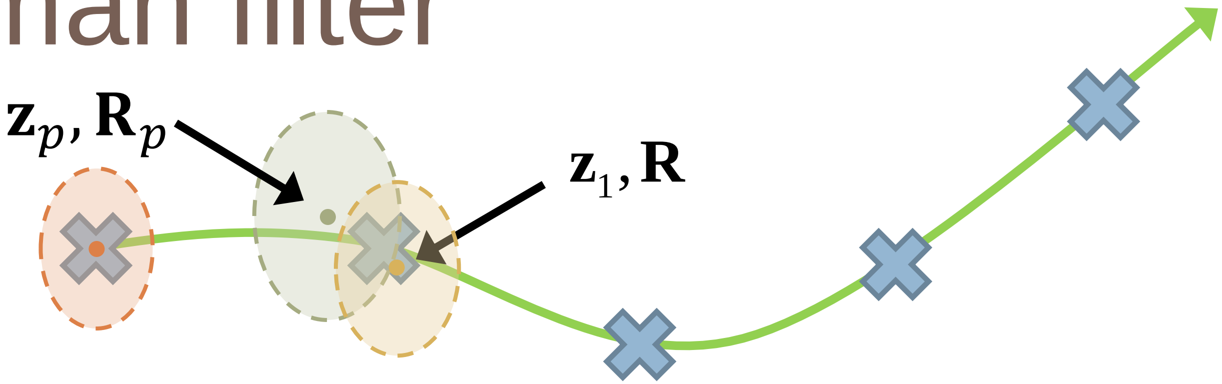
$$\mathbf{z}_1 = \begin{pmatrix} x \\ y \end{pmatrix}$$

- Observation model

$$p\left(\mathbf{z}_t \mid \mathbf{x}_t\right) = N\left(\mathbf{H}_t \mathbf{x}_{t|t-1}, \mathbf{R}_t\right)$$



Kalman filter

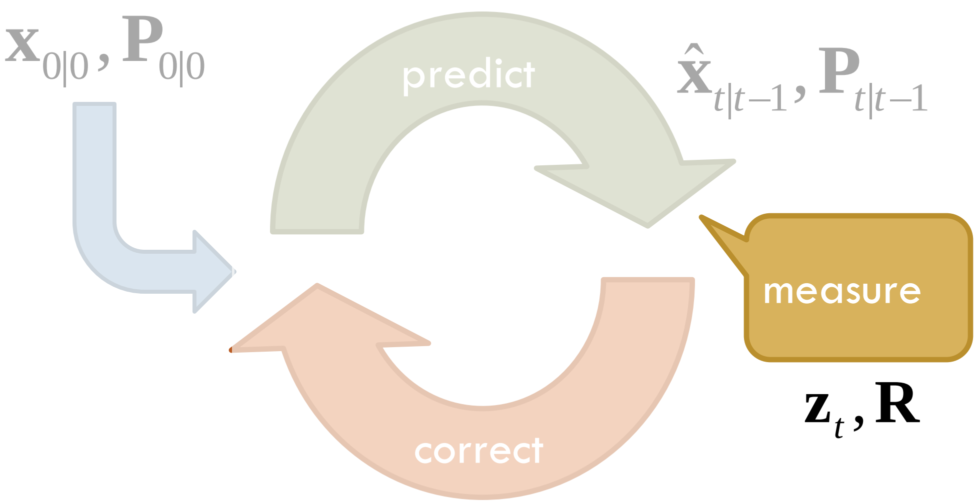


- Predicted observation

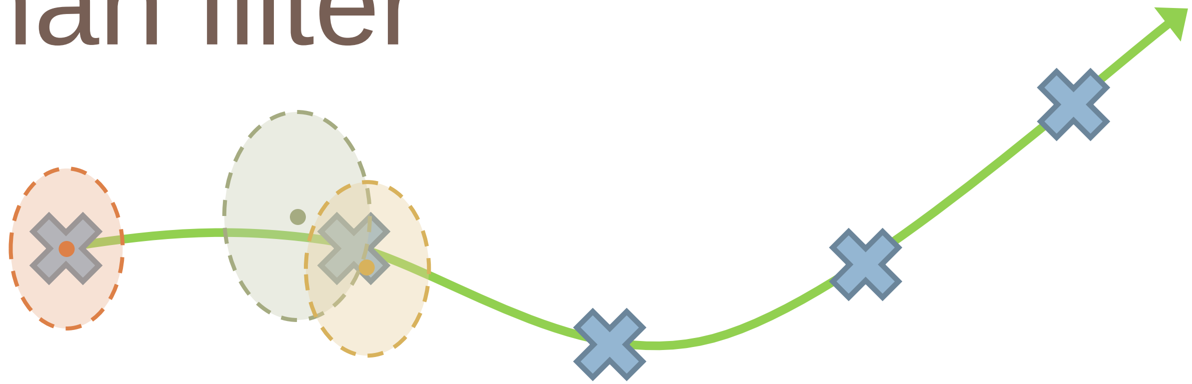
$$\mathbf{z} = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\mathbf{z}_p = \mathbf{H}_t \mathbf{x}_t$$

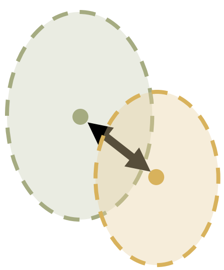
$$\underbrace{\begin{pmatrix} x_p \\ y_p \end{pmatrix}}_{\mathbf{z}_p} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{\mathbf{H}_t} \underbrace{\begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}}_{\mathbf{x}_t}$$



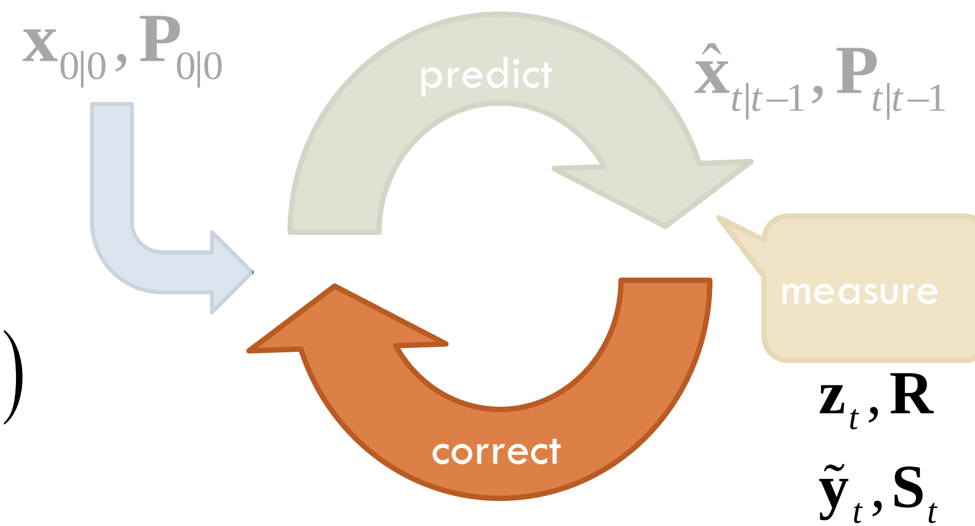
Kalman filter



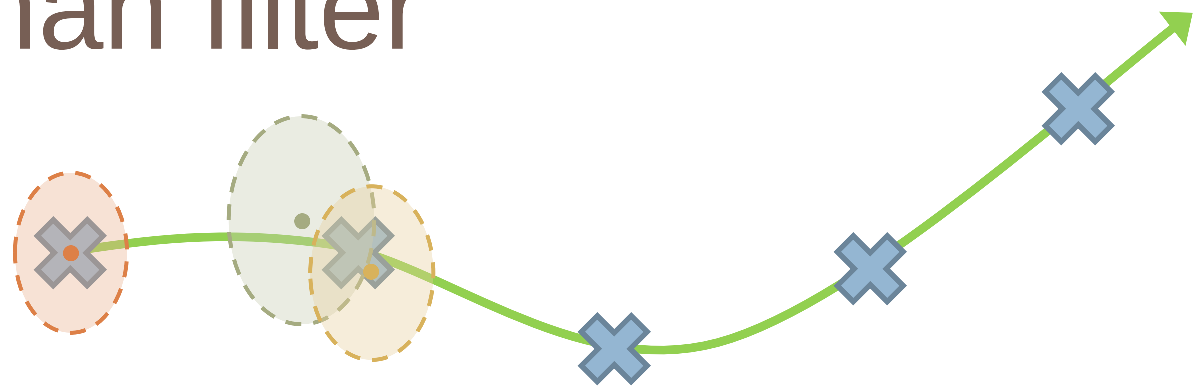
- Observation model – how likely is the observation given the prediction?



$$p(\mathbf{z}_t \mid \mathbf{x}_t) = N\left(\mathbf{H}_t \mathbf{x}_{t|t-1}, \mathbf{R}_t\right)$$



Kalman filter



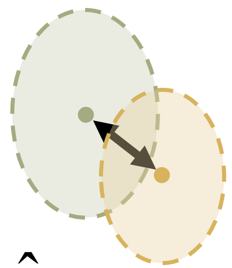
- The residual (innovation), $\tilde{y}_t, \boldsymbol{S}_t$

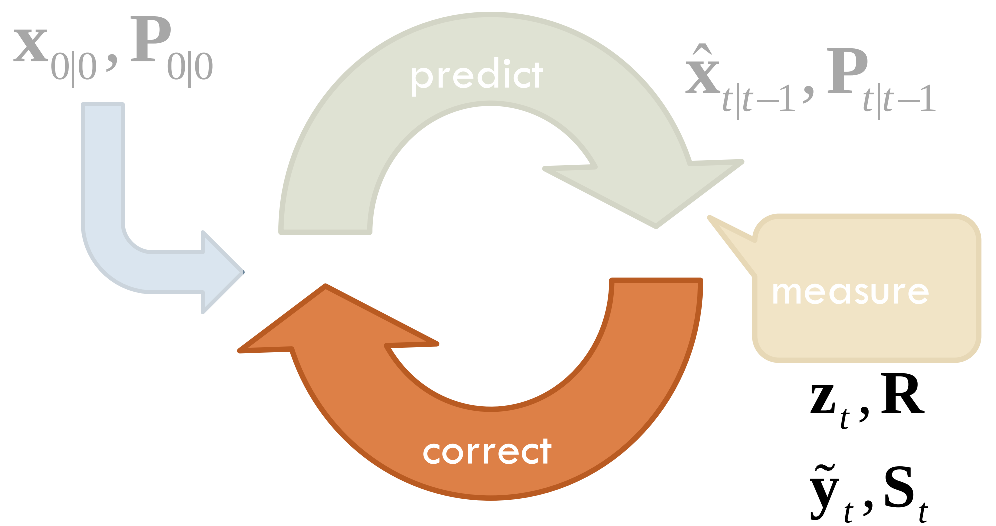
predicted measurement
actual measurement

$$\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1}$$

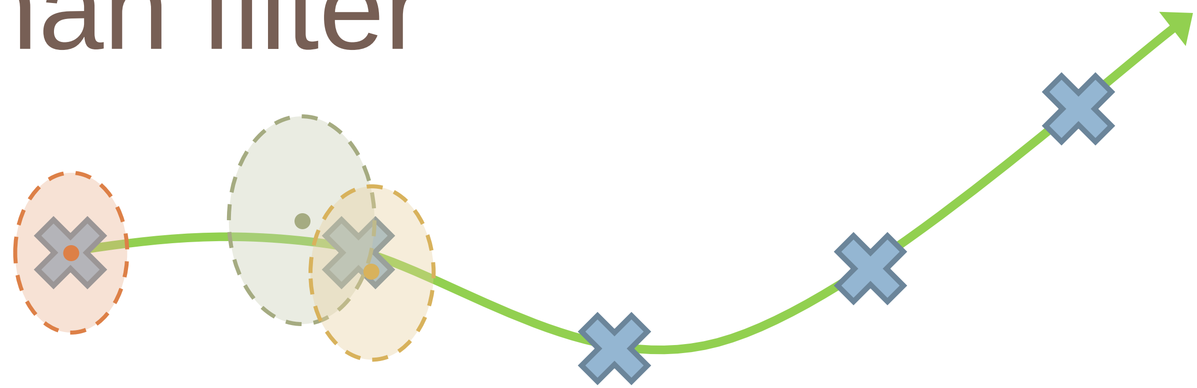
$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^{\mathrm{T}} + \mathbf{R}_t$$

predicted covariance
measurement noise





Kalman filter

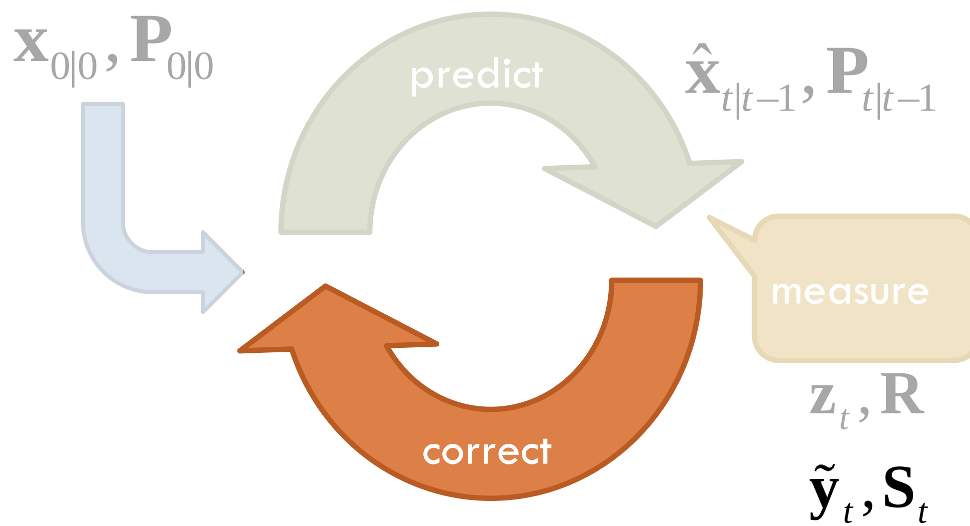


- Correct the prediction using measurement
  - Kalman gain, $\mathbf{K}$ – specifies how much the correction considers the prediction $\hat{\mathbf{x}}_{(t|t-1)}, \mathbf{P}_{t|t-1}$ or the measurement $\tilde{y}_t, \mathbf{S}_t$

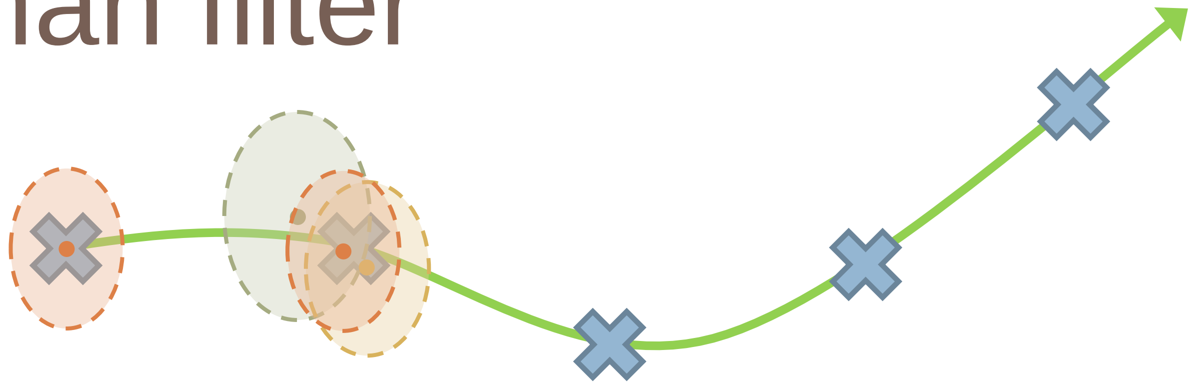
$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{S}_t^{-1}$$

predicted covariance

residual covariance



Kalman filter



- Correct the prediction using measurement

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \tilde{\mathbf{y}}_k$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{H} \hat{\mathbf{x}}_{t|t-1} \right)$$

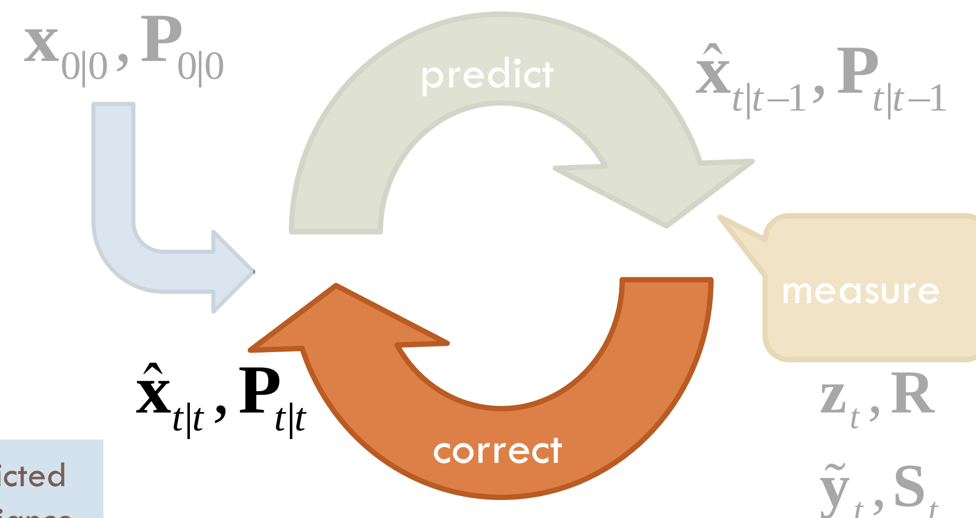
state prediction

residual

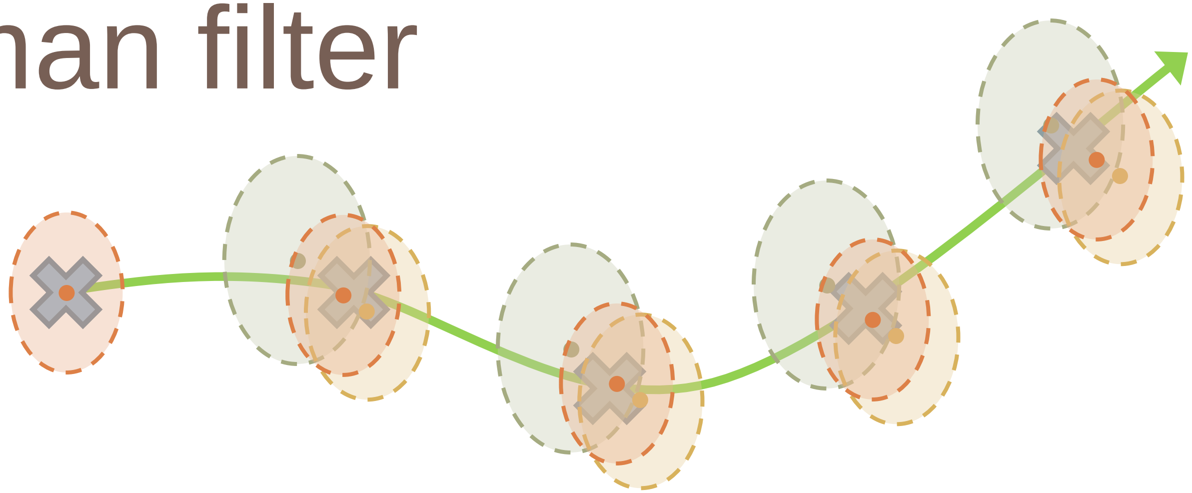
$$\mathbf{P}_{t|t} = \left(I - \mathbf{K}_t \mathbf{H}_t \right) \mathbf{P}_{t|t-1}$$

includes residual covariance

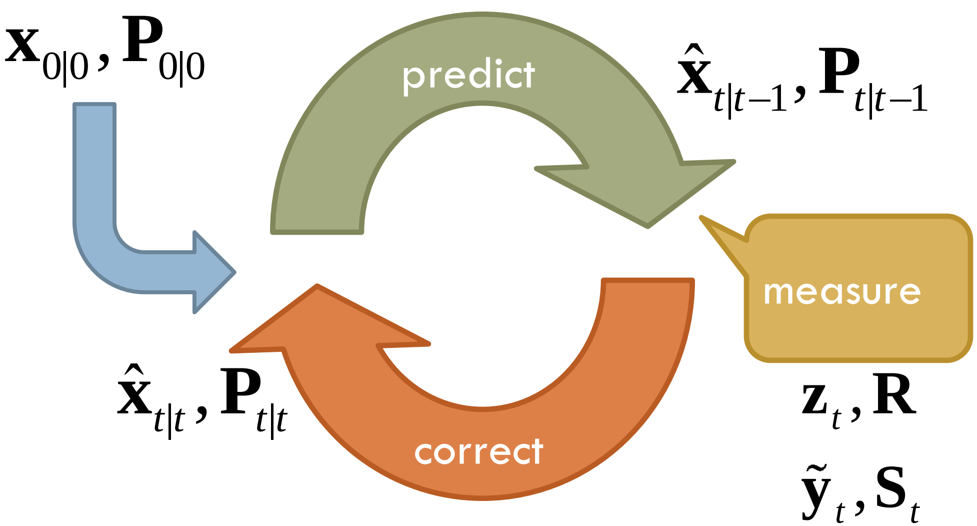
predicted covariance



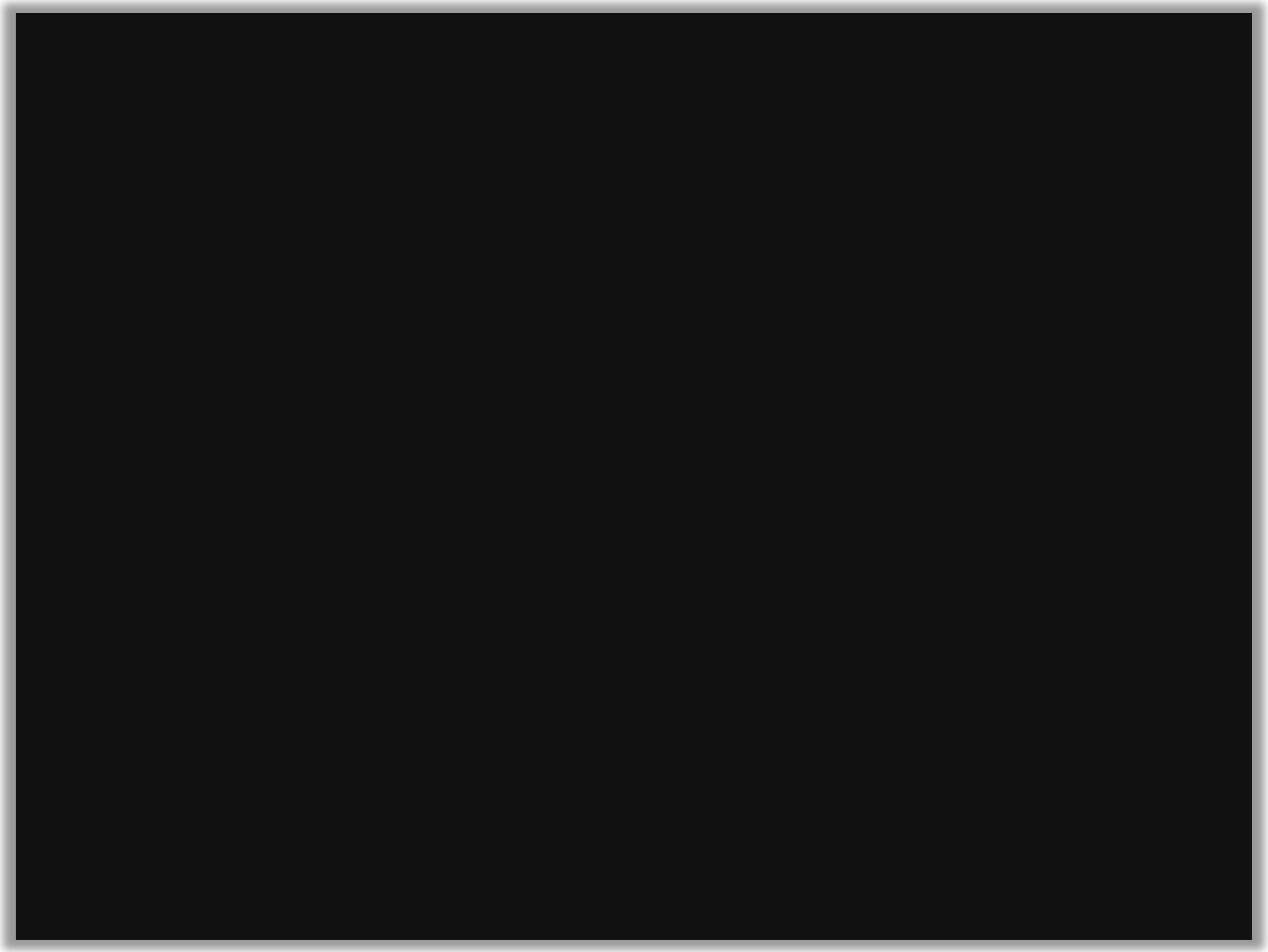
Kalman filter



- Predict, measure, correct cycle iteratively estimates the state at each time step



Kalman filter



Kalman filter smoothing of accelerometer measurements.

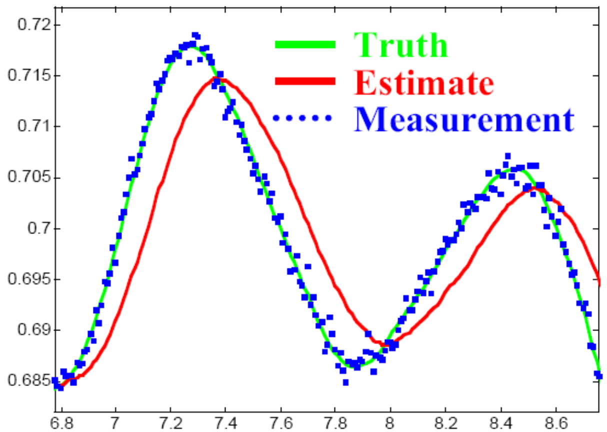
Kalman filter

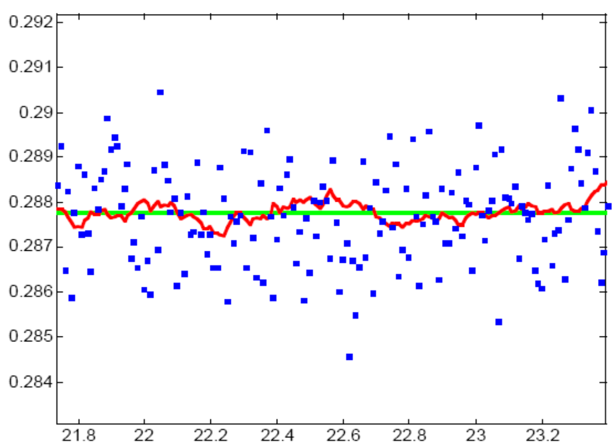
Kalman filter tracking an aircraft.

Kalman filter tracking an aircraft.

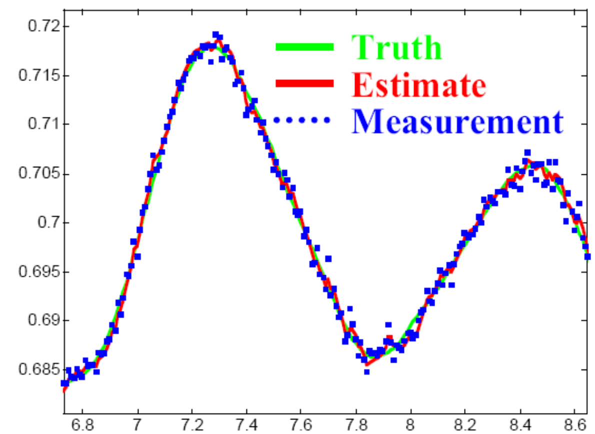
Kalman filter limitations

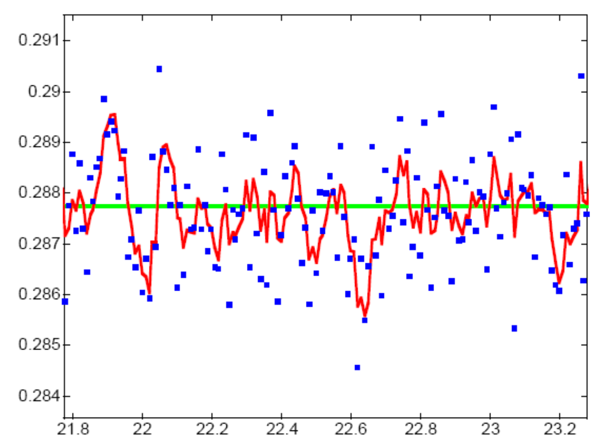
- Position only $\mathbf{x}_t = \begin{pmatrix} x \\ y \end{pmatrix}$





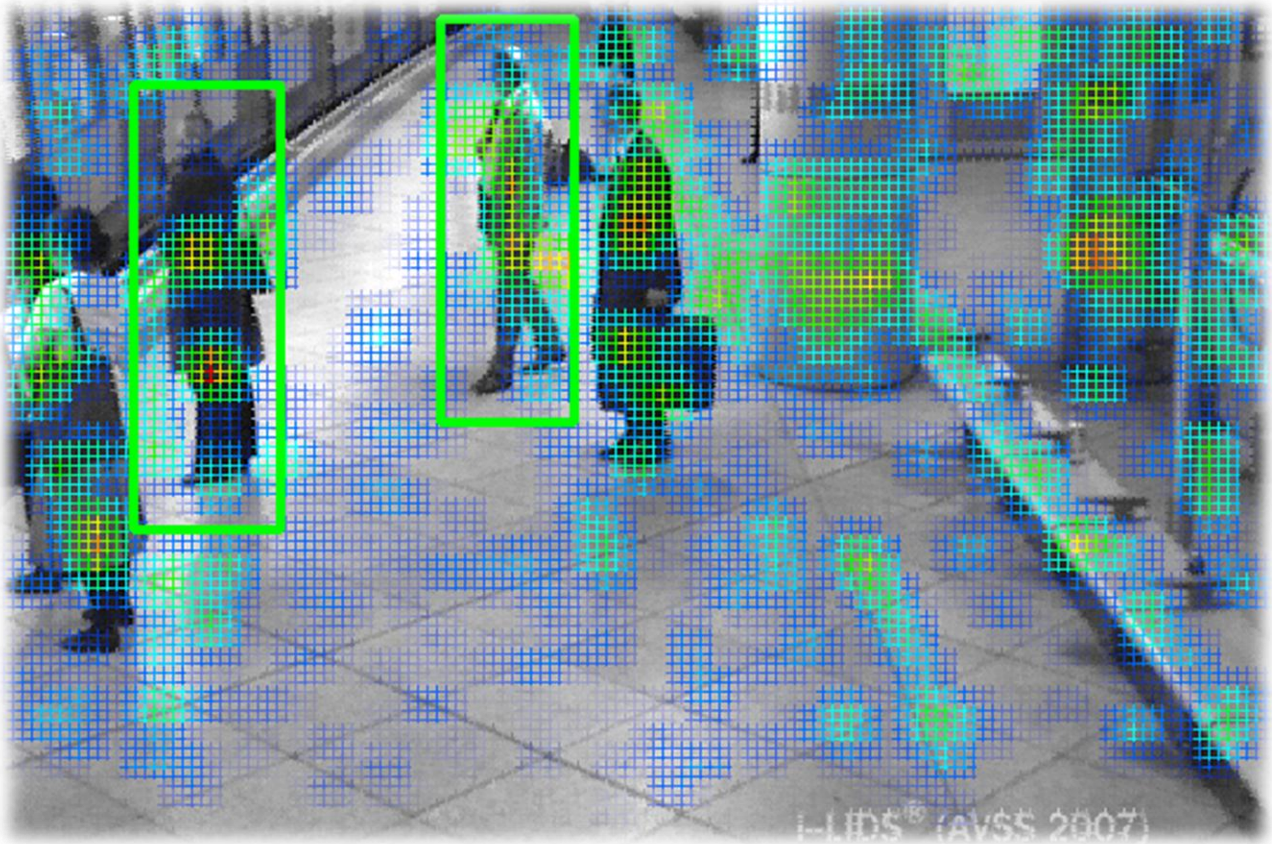
- Constant velocity model $\mathbf{x}_t = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}$





Kalman limitations

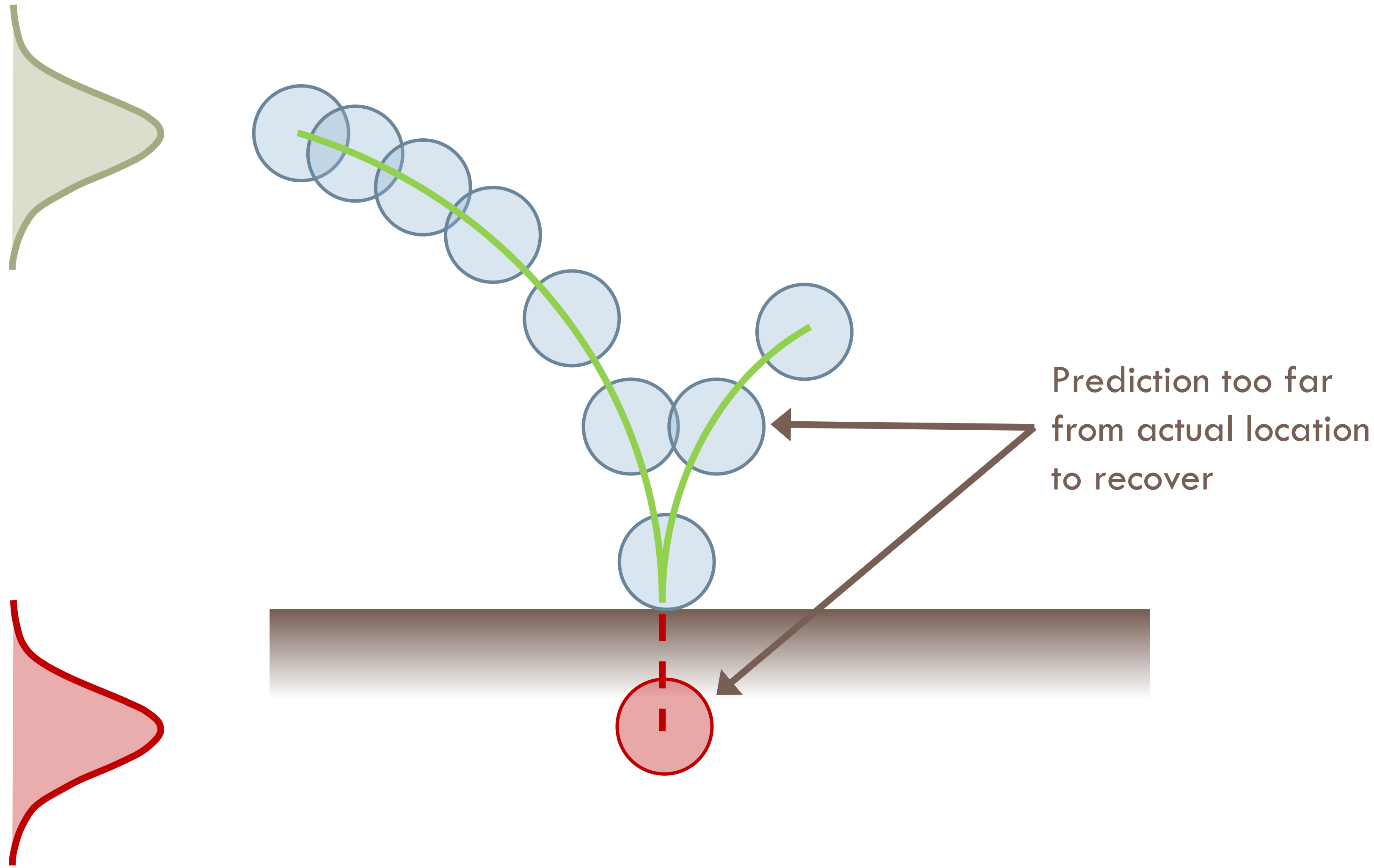
- No non-Gaussian observation models



M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, Robust Tracking-by-Detection using a Detector Confidence Particle Filter , International Conference on Computer Vision (ICCV), 2009

Kalman limitations

- Uni-modal distributions fail for unpredicted motion



Summary: Kalman filter

- **Pros +**
  - Gaussian densities easy to work with
  - Exact solution
  - Well established method
- **Cons -**
  - Restricted to Gaussian densities
  - Uni-modal distribution: single hypothesis
  - Only linear, continuous dynamic model

Outline

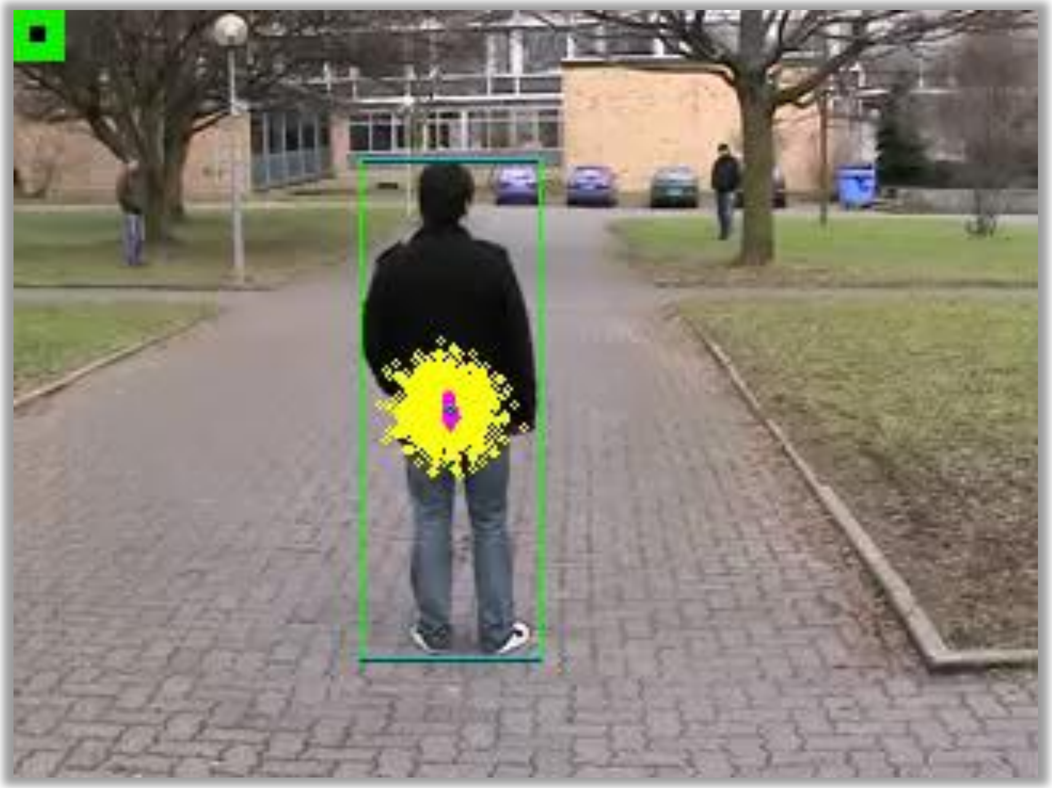
Introduction to the tracking problem

- What is tracking?
- Approaches , assumptions, &applications
- State of the art & challenges

Recursive Bayesian filtering

- Background & formulation
- Kalman filter
- Particle filter

Particle filter



D. Klein, D. Schulz, S. Frintrop, and A. Cremers, Adaptive Real-Time Video Tracking for Arbitrary Objects , International Conference on Intelligent Robots and Systems (IROS), 2010

Particle filters

- Go by many names:
  - Sequential Monte Carlo Methods
  - Sequential importance resampling (SIR)
  - Bootstrap filters
  - Condensation trackers
  - Survival of the fittest

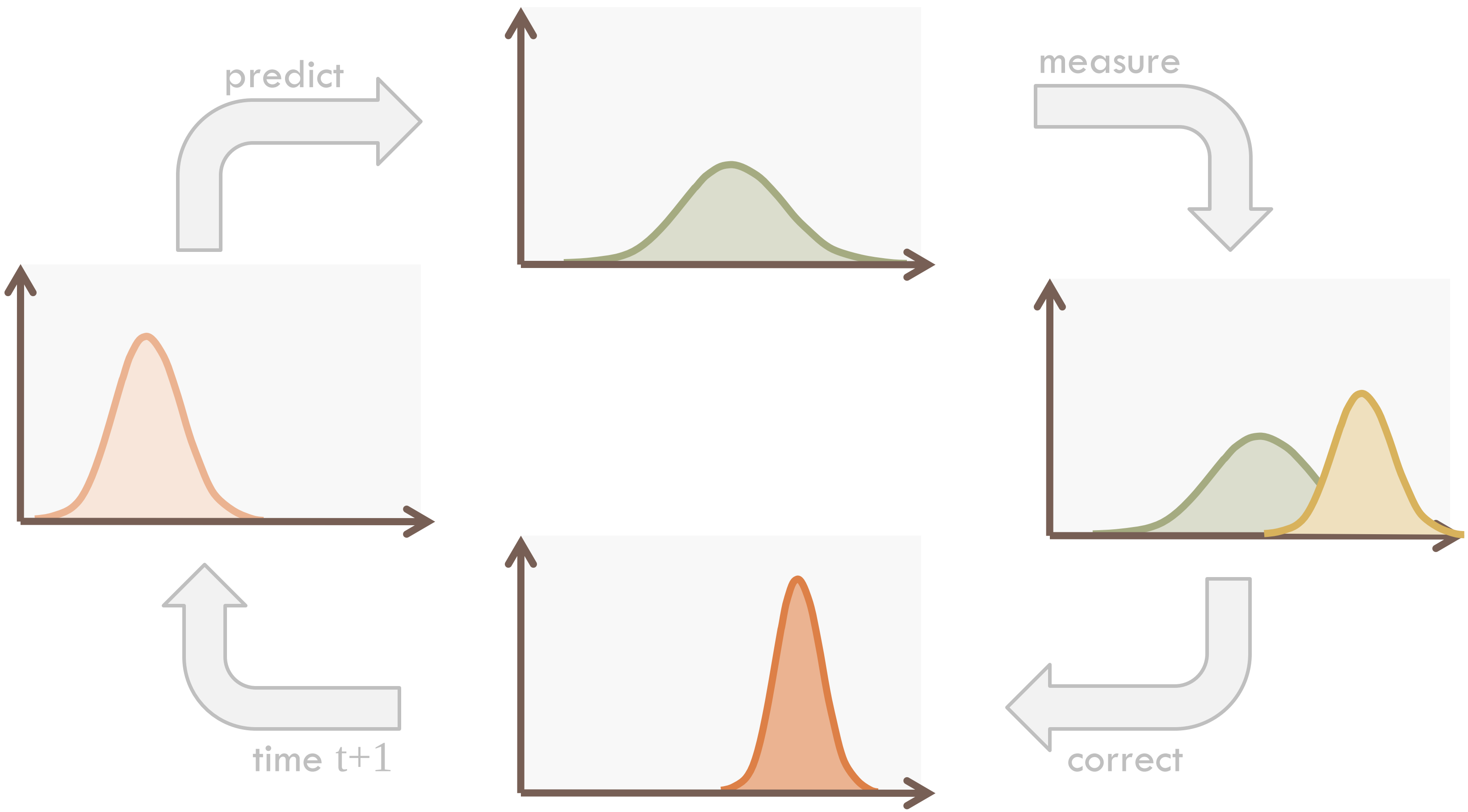
- Originally used for problems in
  - Statistics
  - Fluid mechanics
  - Statistical mechanics
  - Signal processing

- Introduced to the Computer Vision community by

Michael Isard and Andrew Blake, CONDENSATION -- Conditional Density Propagation for Visual Tracking, International Journal of Computer Vision (IJCV), 29, 1, 5--28, (1998)

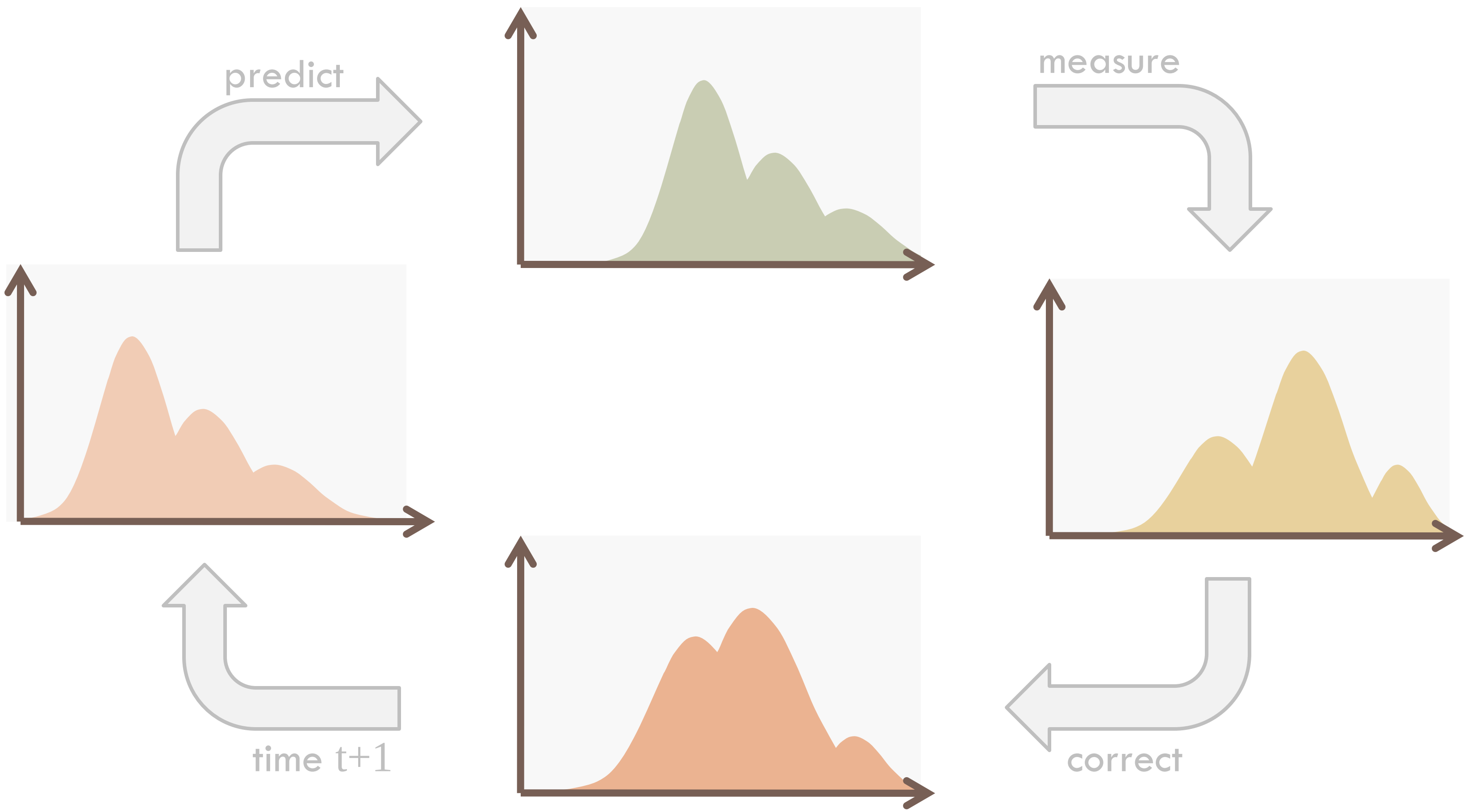
Probability density propagation

- **Gaussian densities** → Kalman filter



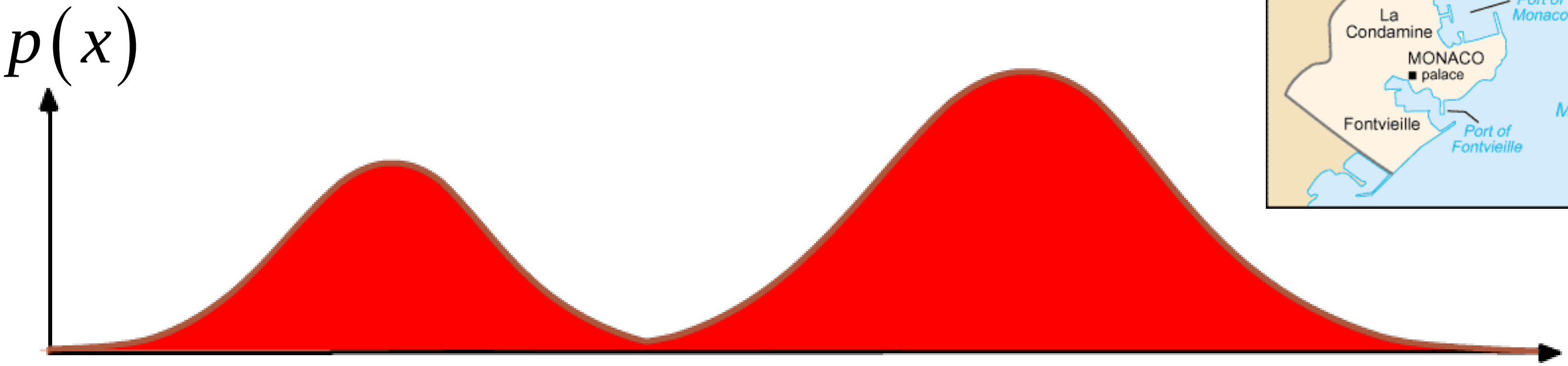
Probability density propagation

- **General densities** → particle filter



Monte Carlo approximation

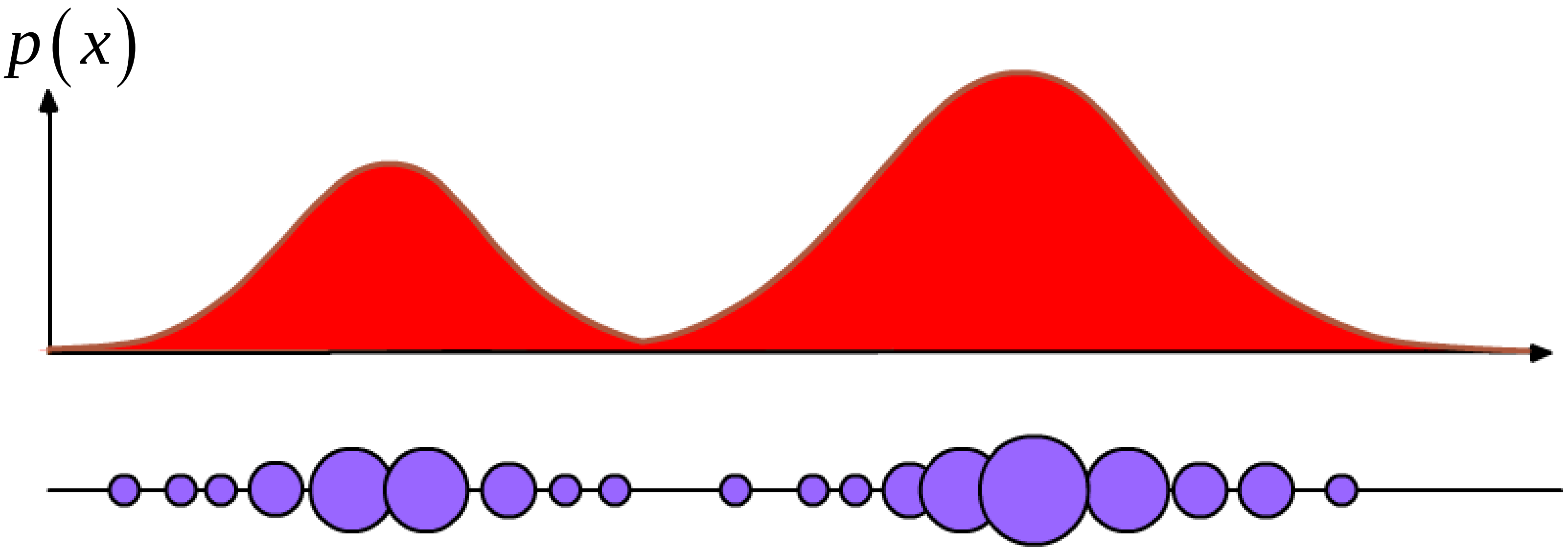
- How can we represent an arbitrary probability density?



A complicated density we'd like to represent with particles

Monte Carlo approximation

- Represent the density non-parametrically, as a set of (weighted) samples!

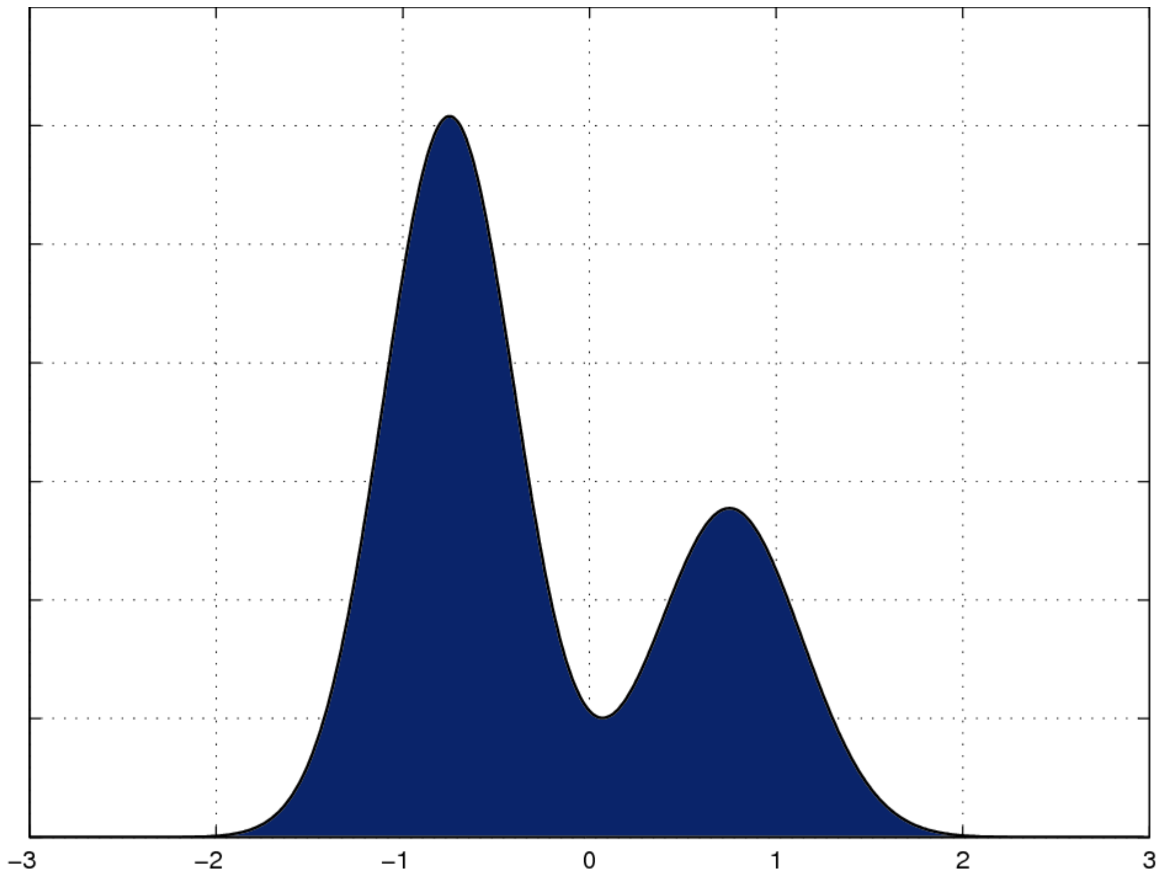


Monte Caro approximation

$$p(x) \approx \sum_{n=1}^{N} w_n \, \delta\left(x - x_n\right)$$

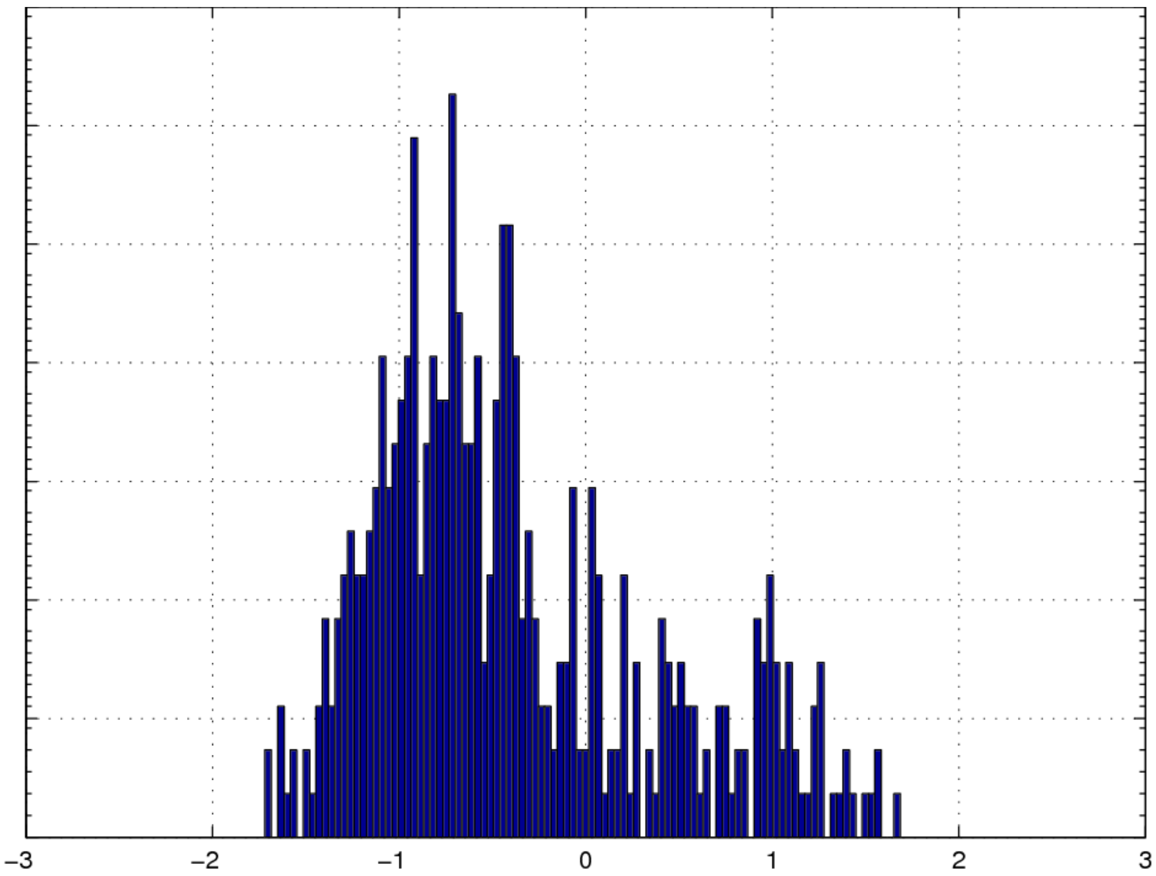
Particle approximation

- Target distribution



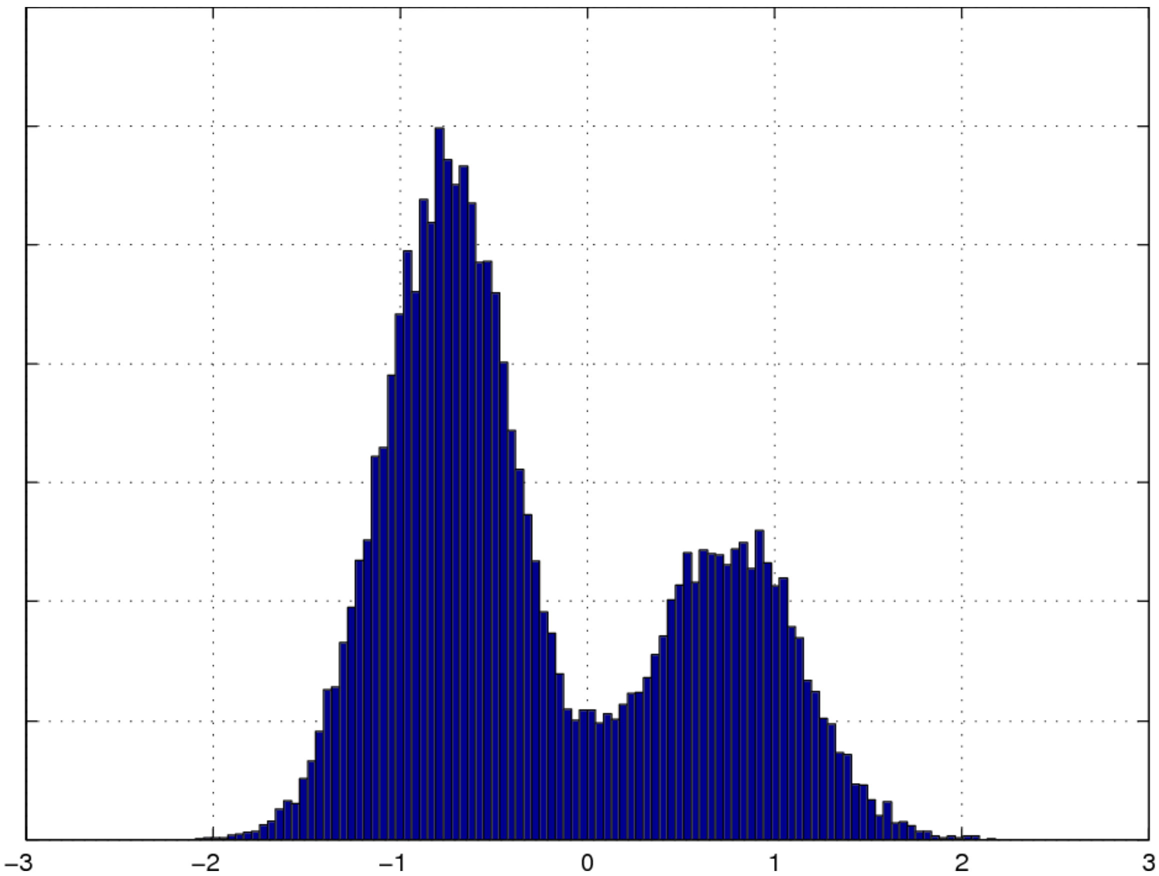
Particle approximation

- Monte Carlo approximation – **too few samples**

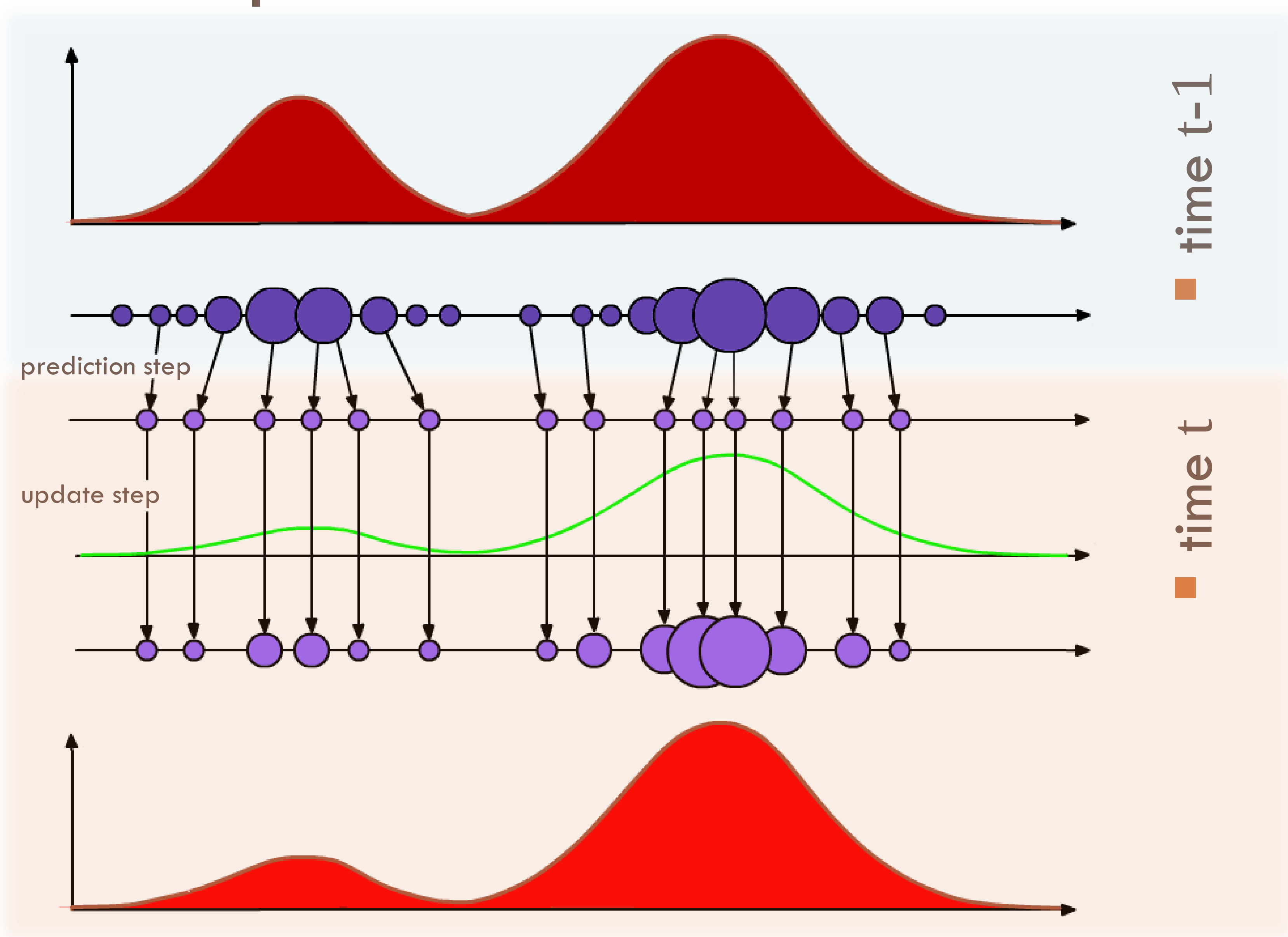


Particle approximation

- Monte Carlo approx – **added samples**



SIR particle filter



Homework assignment #5

- Implement an SIR Particle filter
  - Code – hand in electronically [7 pts]
  - Results on 3 sequences

[1 pts]

Sequence 1. Toy car

[1 pts]

Sequence 2. Girl in Pink

[1 pts]

Sequence 3. Head tracking

Homework assignment #5

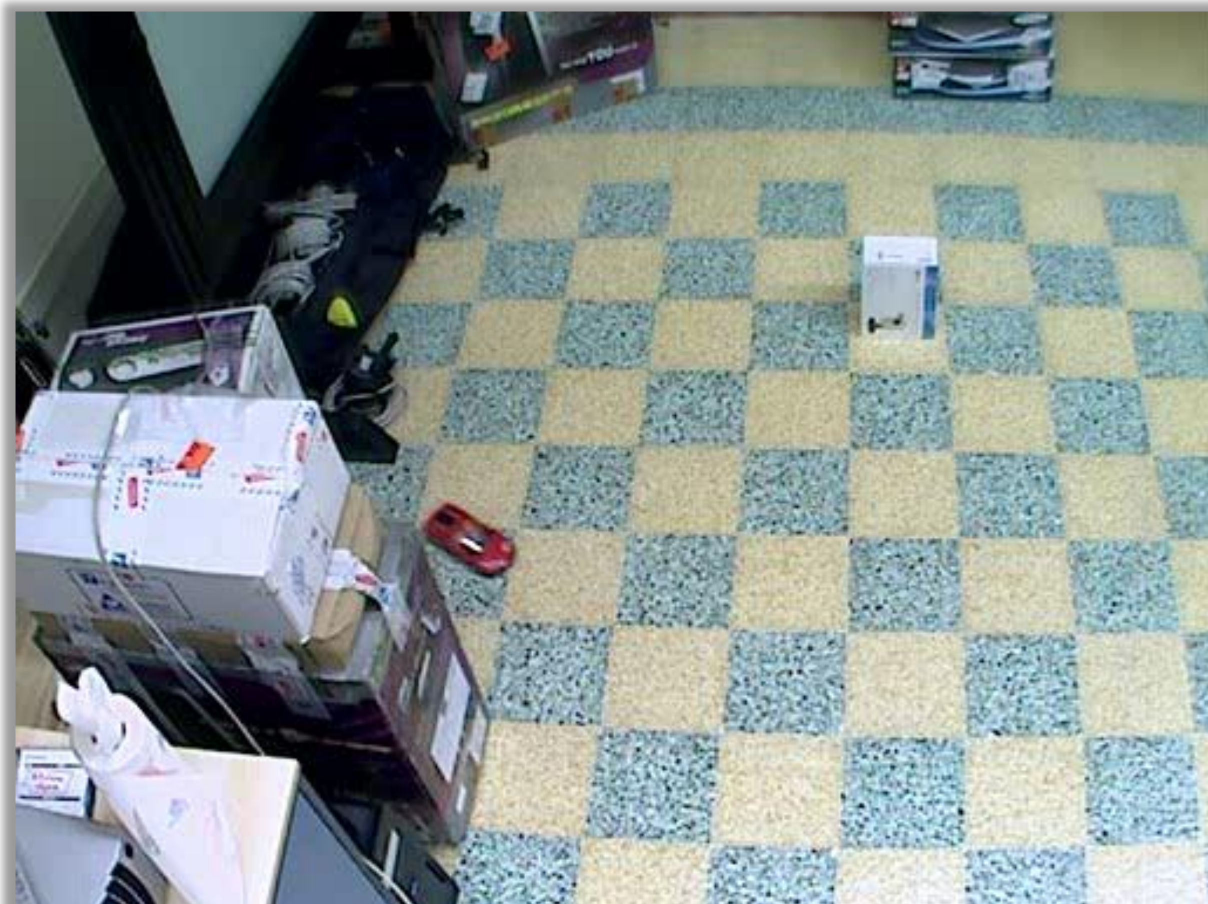
- **Due** on December 9th

- Skeleton code on course web site
  - Mundane tasks are already written
  - Hints provided

- Bonus points possible for
  - Original observation model, or
  - Automatic initialization

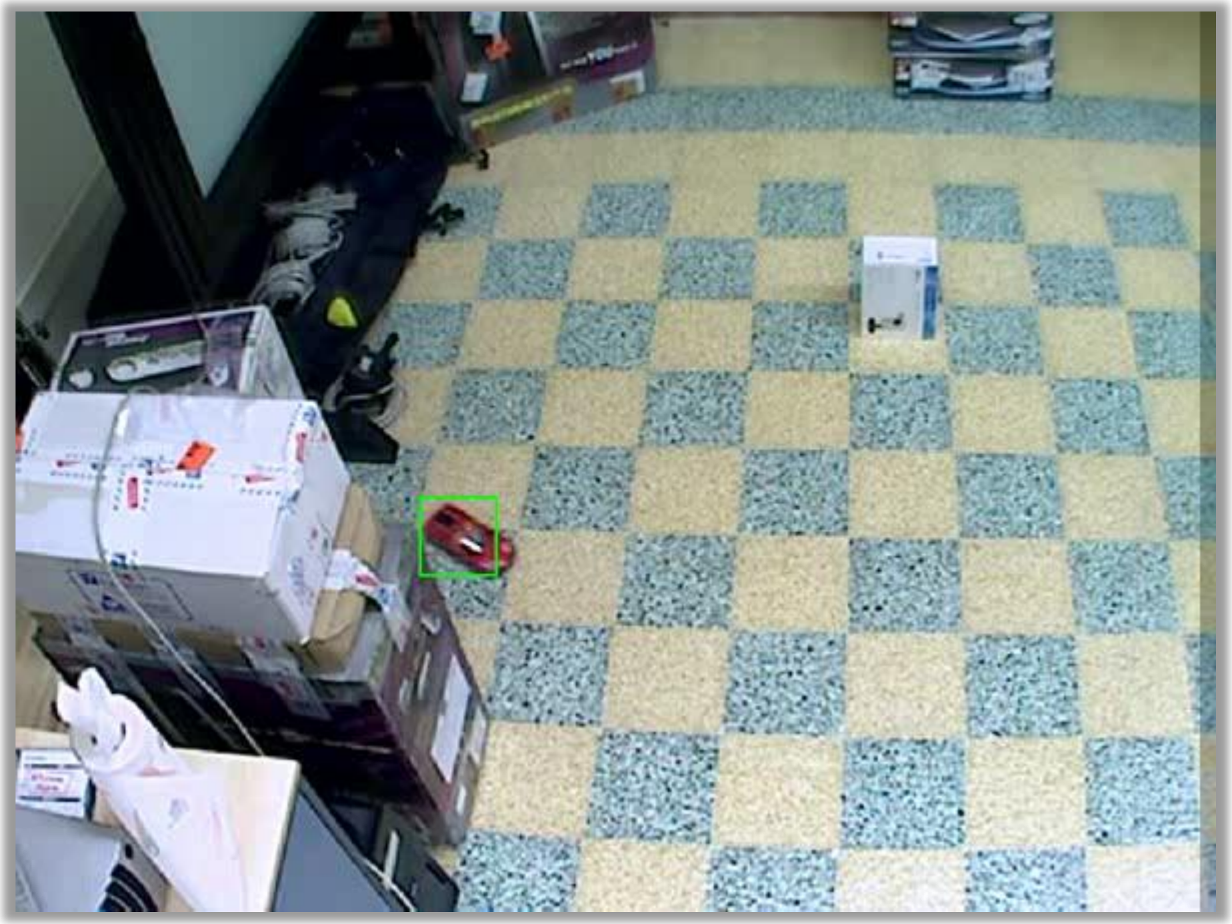
Homework assignment #5

- **Sequence 1** Track the red toy car



Homework assignment #5

- **Sequence 1** Red toy car – **my results**

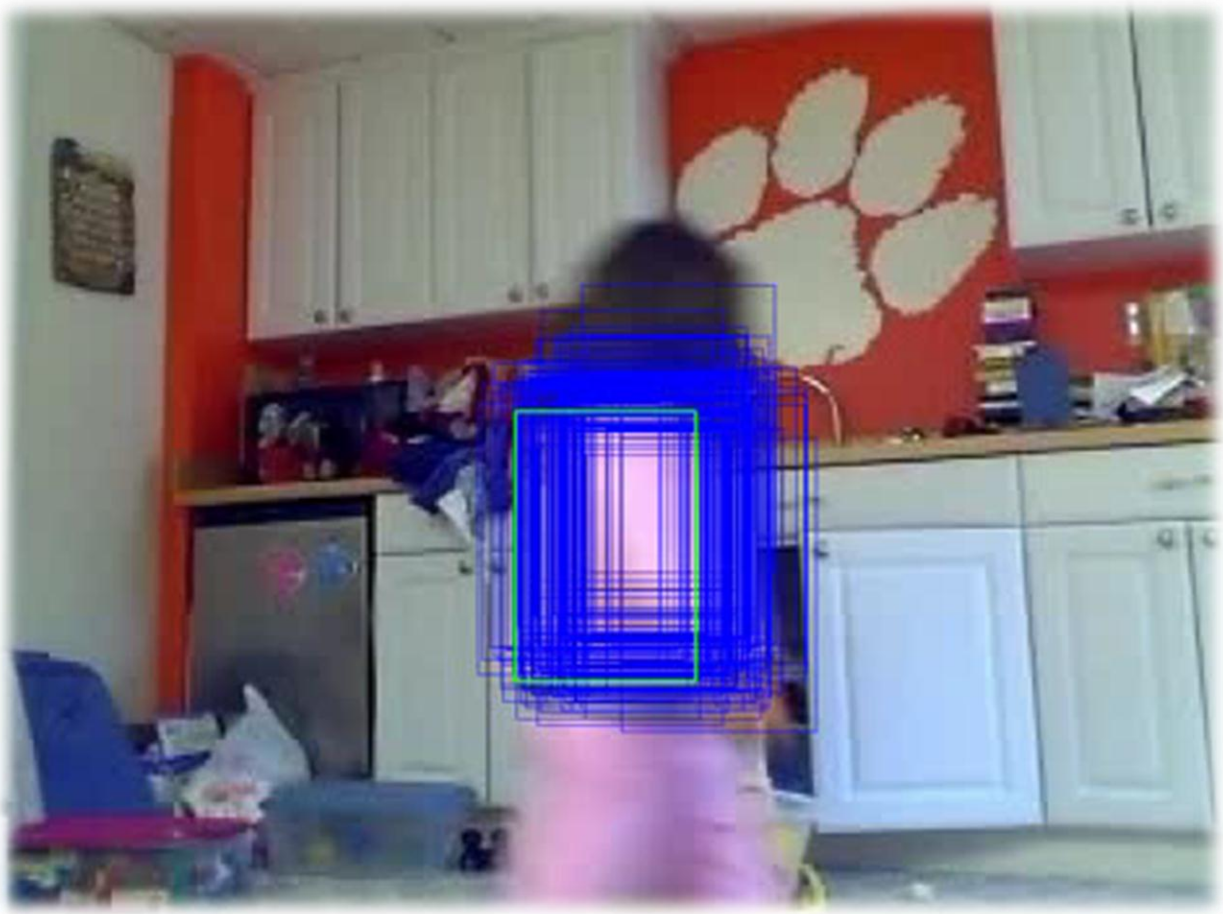


Homework assignment #5

- **Sequence 2** Track the girl in pink

Homework assignment #5

- **Sequence 2** Girl in pink– **my results**



Homework assignment #5

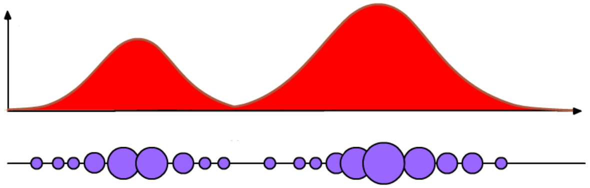
- **Sequence 3** Track the head of the person on the left

Homework assignment #5

- **Sequence 3** Head tracking– **my results**

What is a particle?

- A “sample” of the posterior

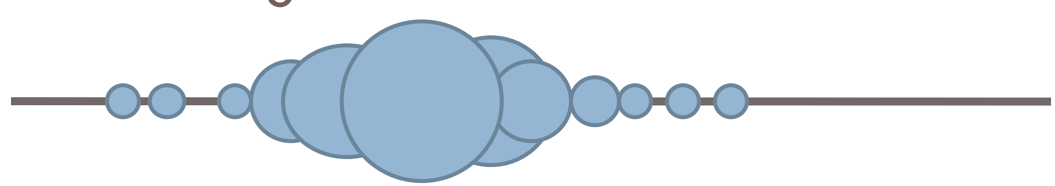


- Particles contain a
  - **state estimate**
  - **weight**

$$s_t^n \triangleq (\mathbf{x}_t^n, w_t^n)$$

- Summing the particles gives an approximation to the target distribution

$$p(\mathbf{x}_t \mid Z_t) \approx \sum_{n=1}^{N} w_{t-1}^n \, \delta\left(\mathbf{x}_t - \mathbf{x}_t^n\right)$$

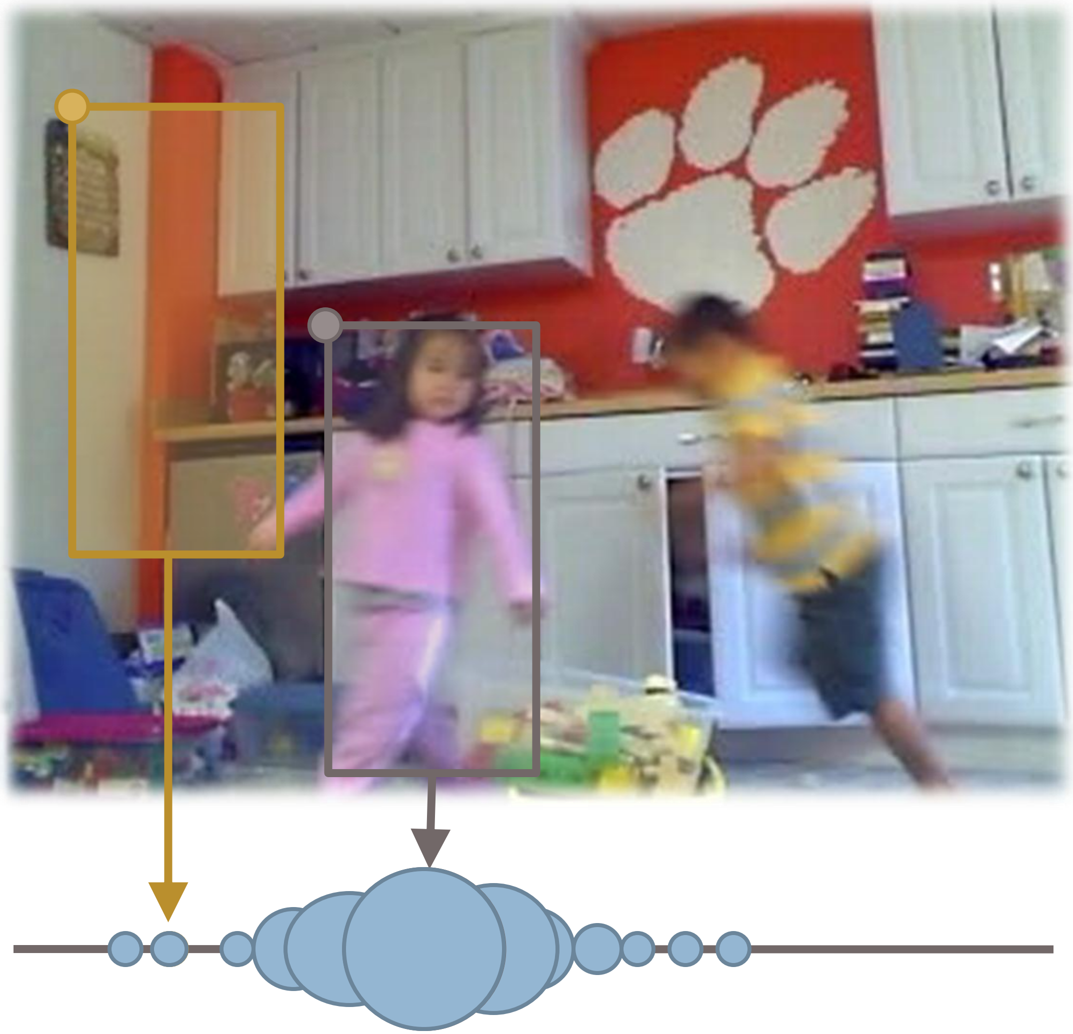


What is a particle?

- Each particle contains a
  - **state estimate**
  - **weight**

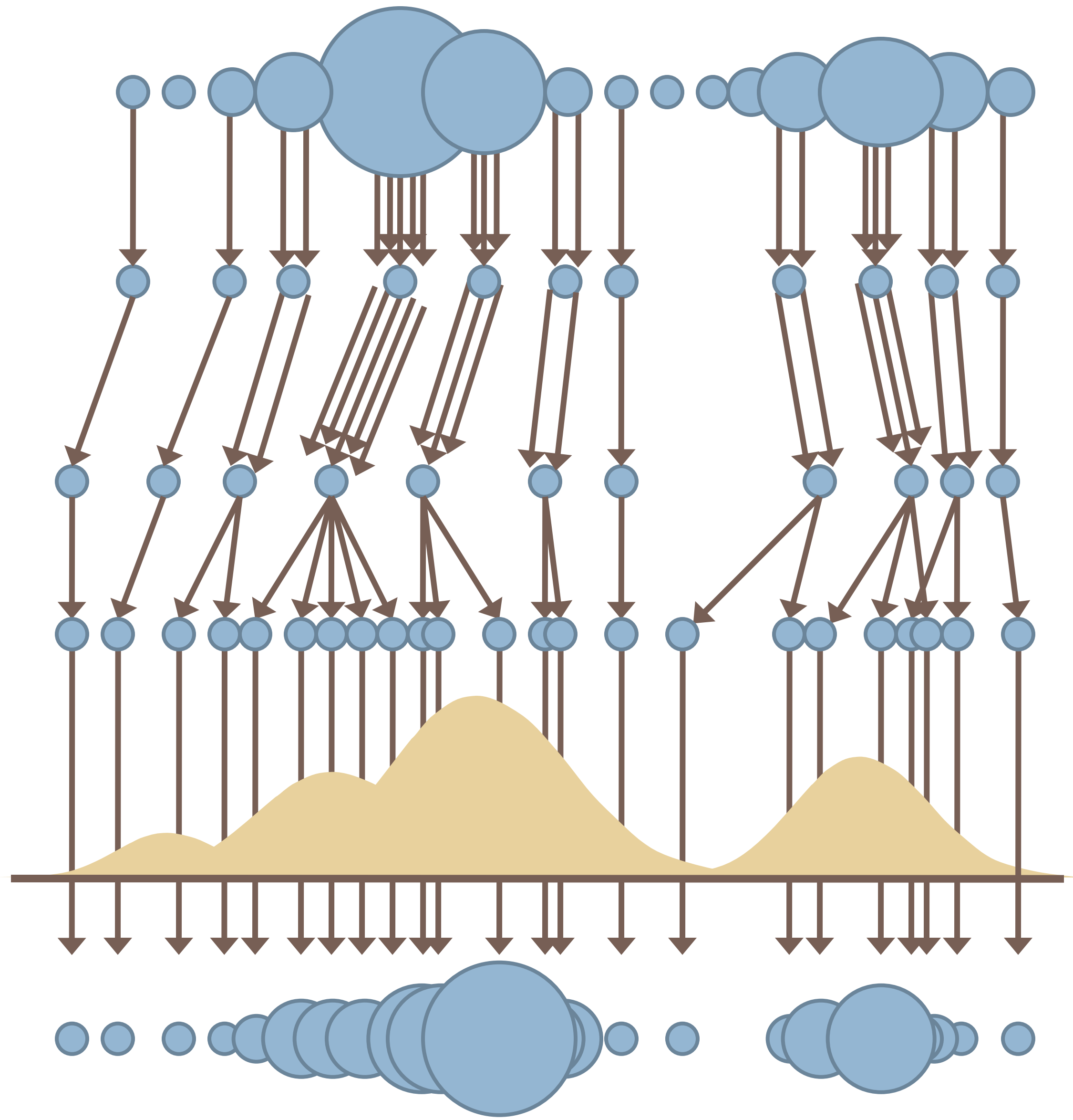
$$s_t^n \triangleq (\mathbf{x}_t^n, w_t^n)$$

$$s_t^n \triangleq \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ a \\ h \end{pmatrix}, w_t$$



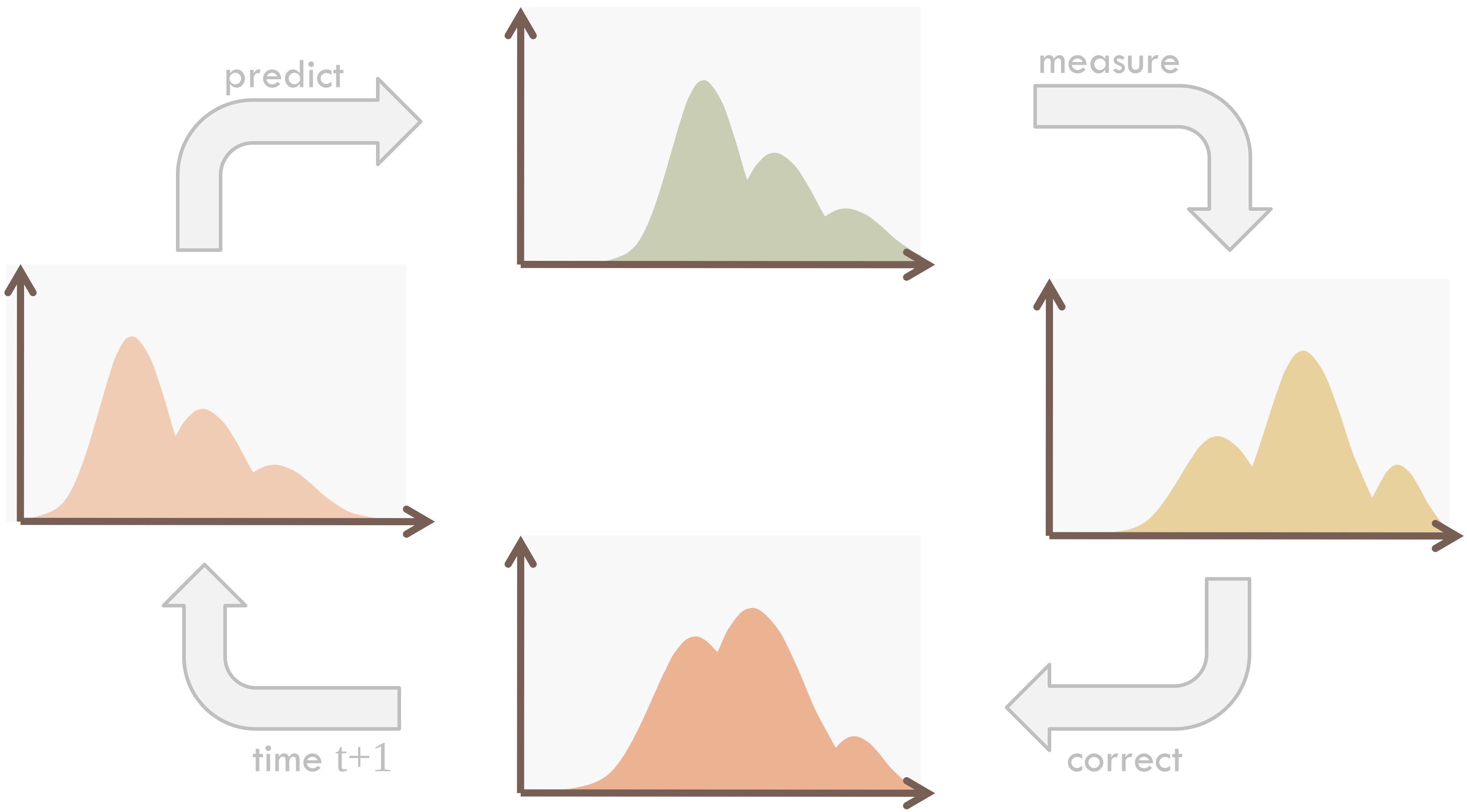
SIR particle filter

- **Begin** with weighted samples from t-1
- **Resample:** draw samples according to $\{w_{t-1}\}^{n=1:N}$
- **Drift:** apply motion model (no noise)
- **Diffuse:** apply noise to spread particles
- **Measure:** weights are assigned by likelihood response
- **Finish:** density estimate



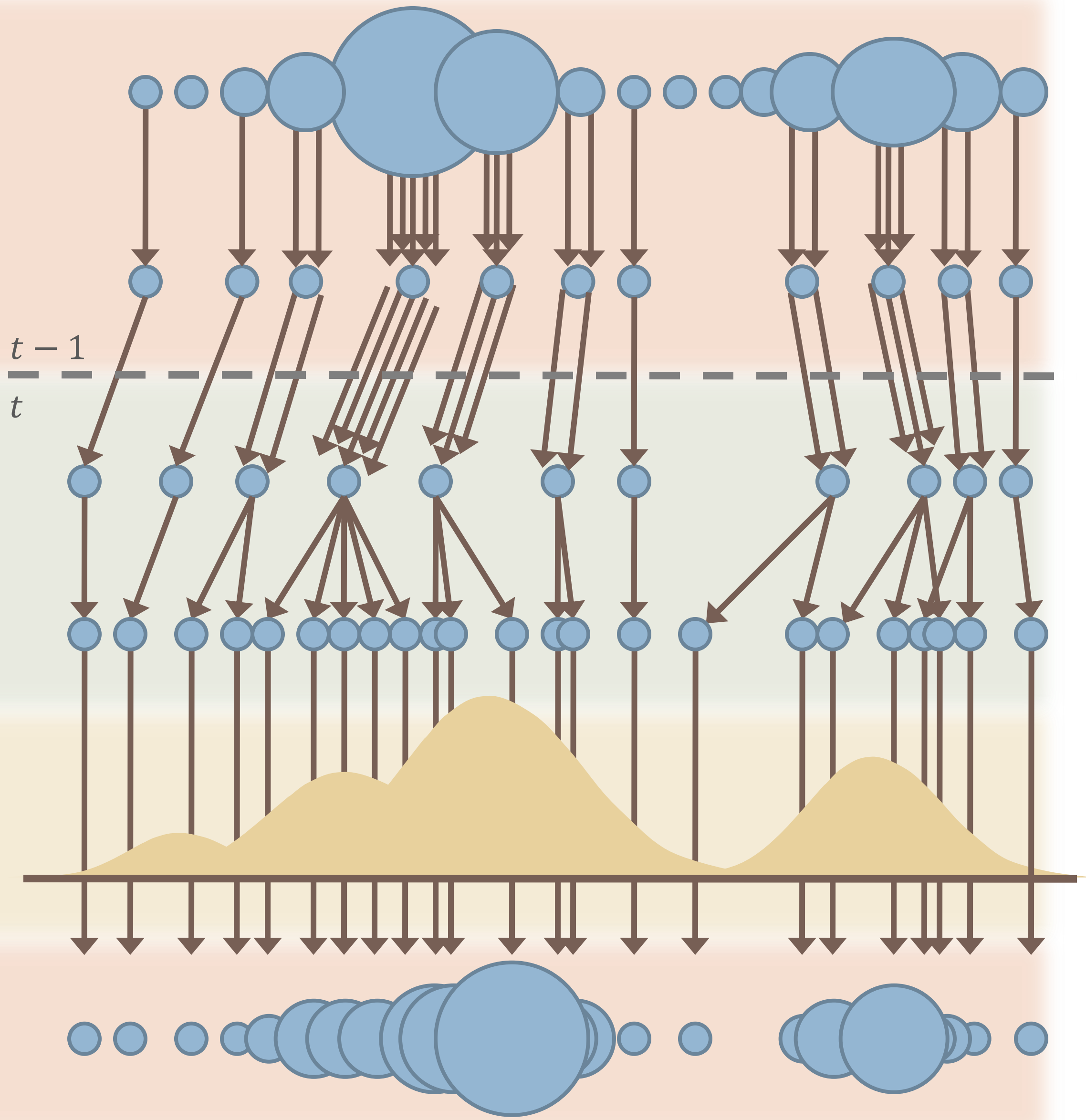
Probability density propagation

- **Notice** similarities to the familiar recursive process



SIR particle filter

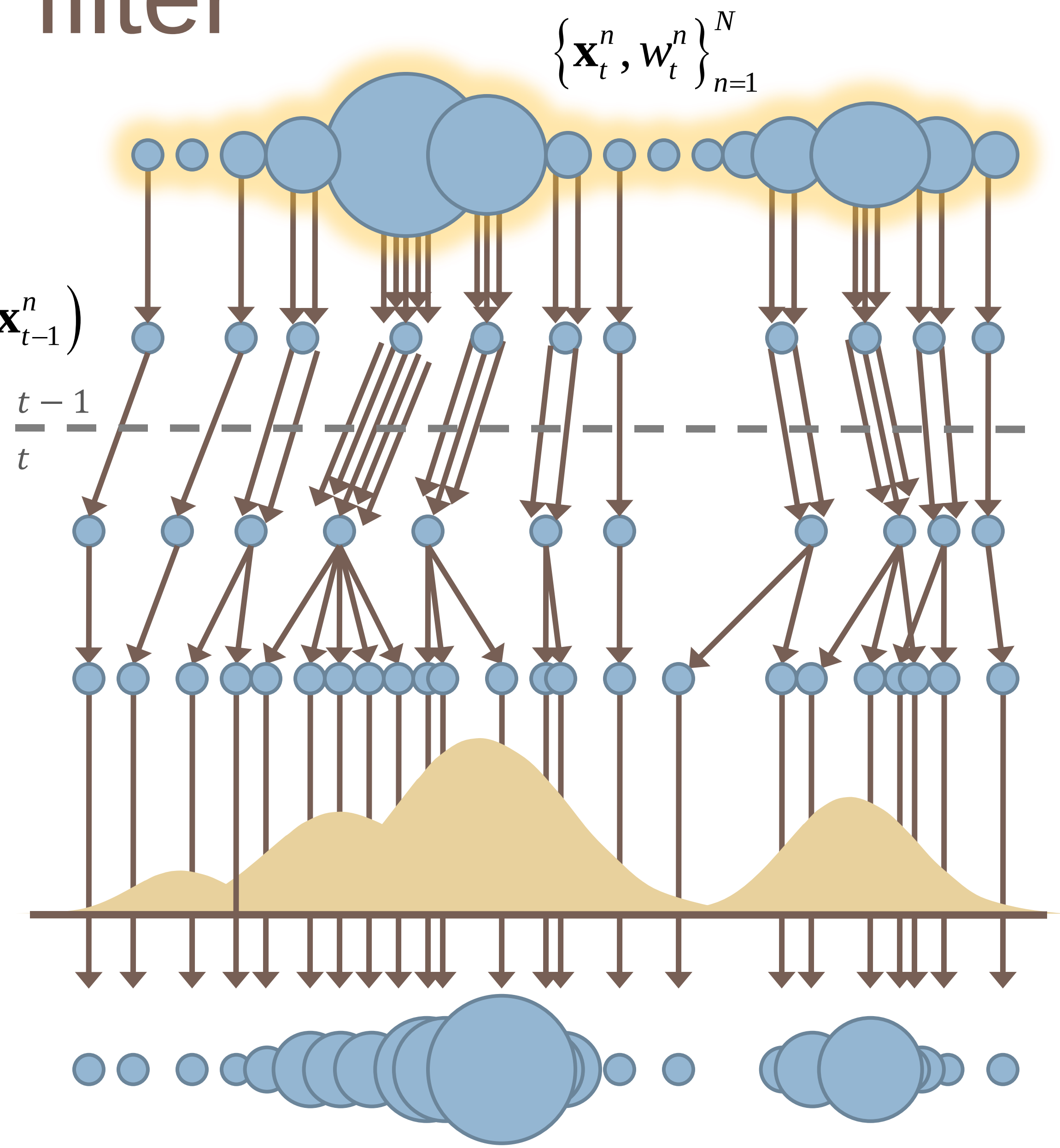
- **Begin** with weighted samples from t-1
- **Resample:** draw samples according to $\{w_{t-1}\}^{n=1:N}$
- **Drift:** apply motion model (no noise)
- **Diffuse:** apply noise to spread particles
- **Measure:** weights are assigned by likelihood response
- **Finish:** density estimate



SIR particle filter

- **Begin** with weighted samples from t-1

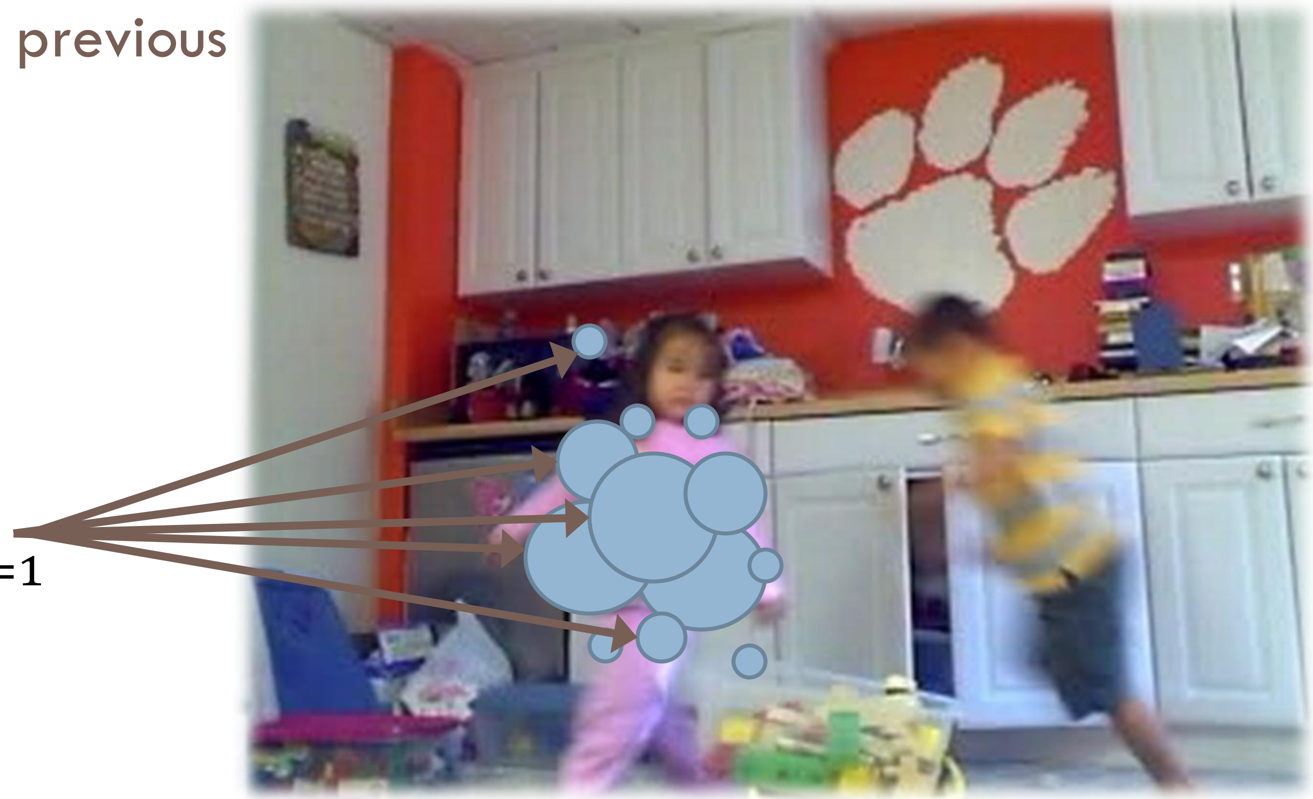
$$p\left(\mathbf{x}_{t-1} \mid Z_{t-1}\right) \approx \sum_{n=1}^{N} w_{t-1}^{n} \, \delta\left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{n}\right)$$



Previous estimate

- Receive posterior estimate from previous time step

$$\{\mathbf{x}_{t-1}^{n}, \mathbf{w}_{t-1}^{n}\}_{n=1}^{N}$$



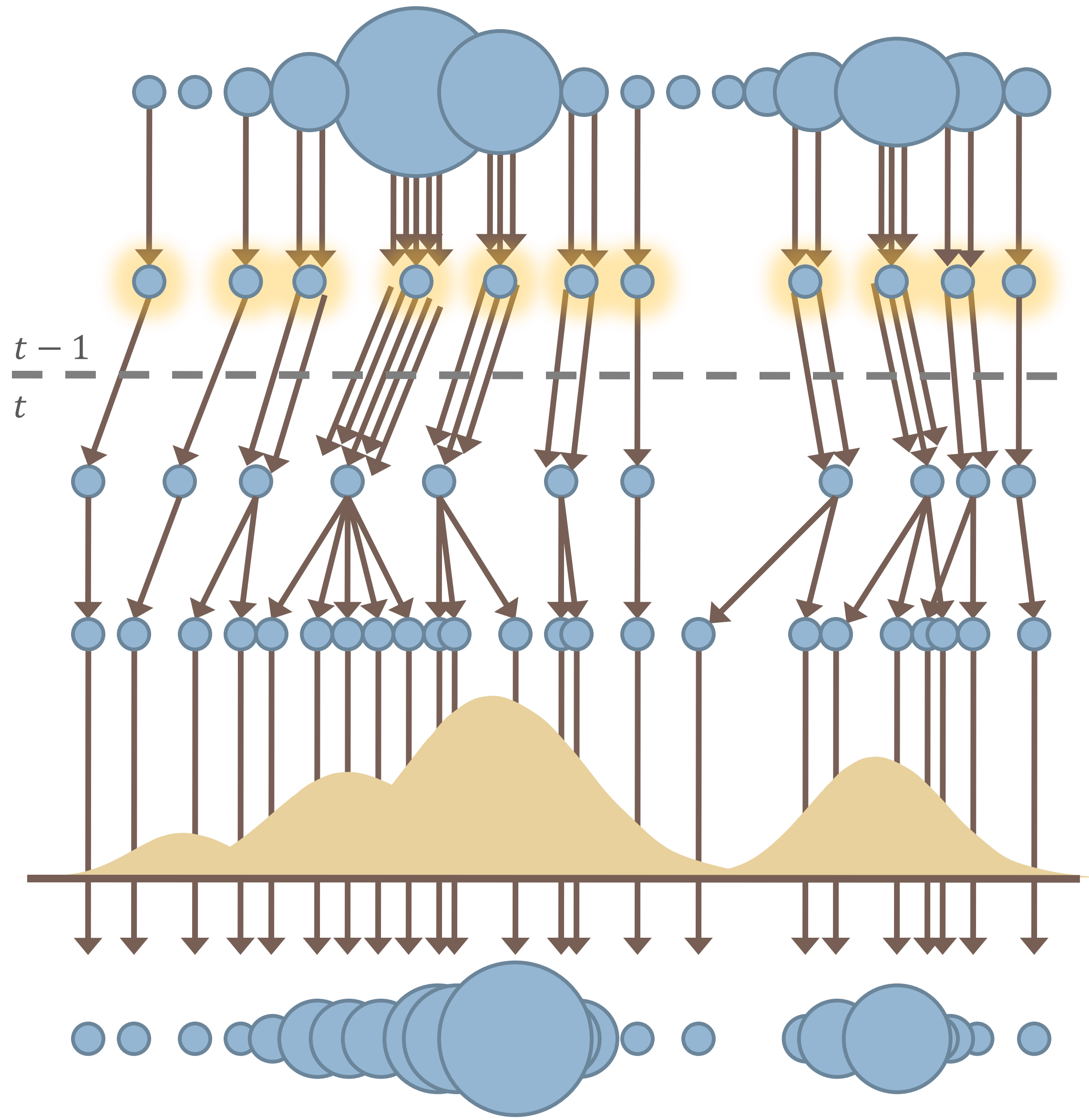
$$p\left(\mathbf{x}_{t-1} \mid Z_{t-1}\right) \approx \sum_{n=1}^{N} w_{t-1}^{n}\ \delta\left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{n}\right)$$

SIR particle filter

- **Resample:** draw samples according to $\{w_{t-1}\}^{n=1:N}$

  N new samples are drawn from the previous set **with replacement.**

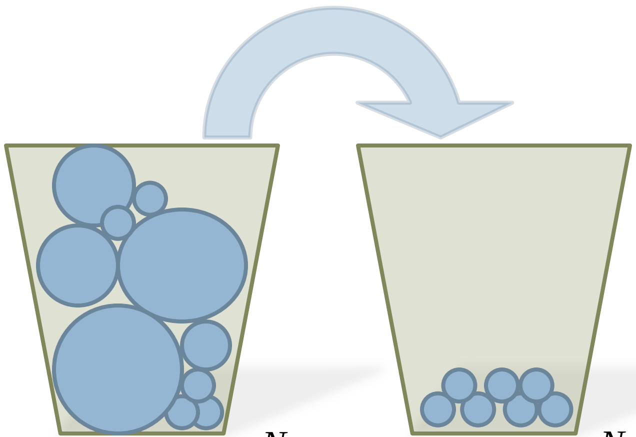
  New samples are assigned **uniform weights.**



Resample

- N new samples are drawn from the previous set **with replacement** to prevent **degeneracy.**
- Repeated samples occur by design.

Weighted sampling with replacement



$$\left\{ \mathbf{x}_{t-1}^{n}, w_{t-1}^{n} \right\}_{n=1}^{N} \qquad \left\{ \mathbf{x}_{t}^{n}, w_{t}^{n} \right\}_{n=1}^{N}$$

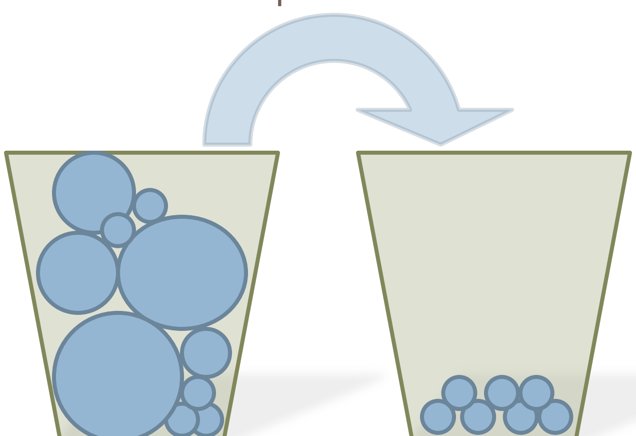
New sample set is given uniform weights

Resample

- N new samples are drawn from the previous set **with replacement** to prevent **degeneracy.**

- Repeated samples occur by design.

Weighted sampling with replacement

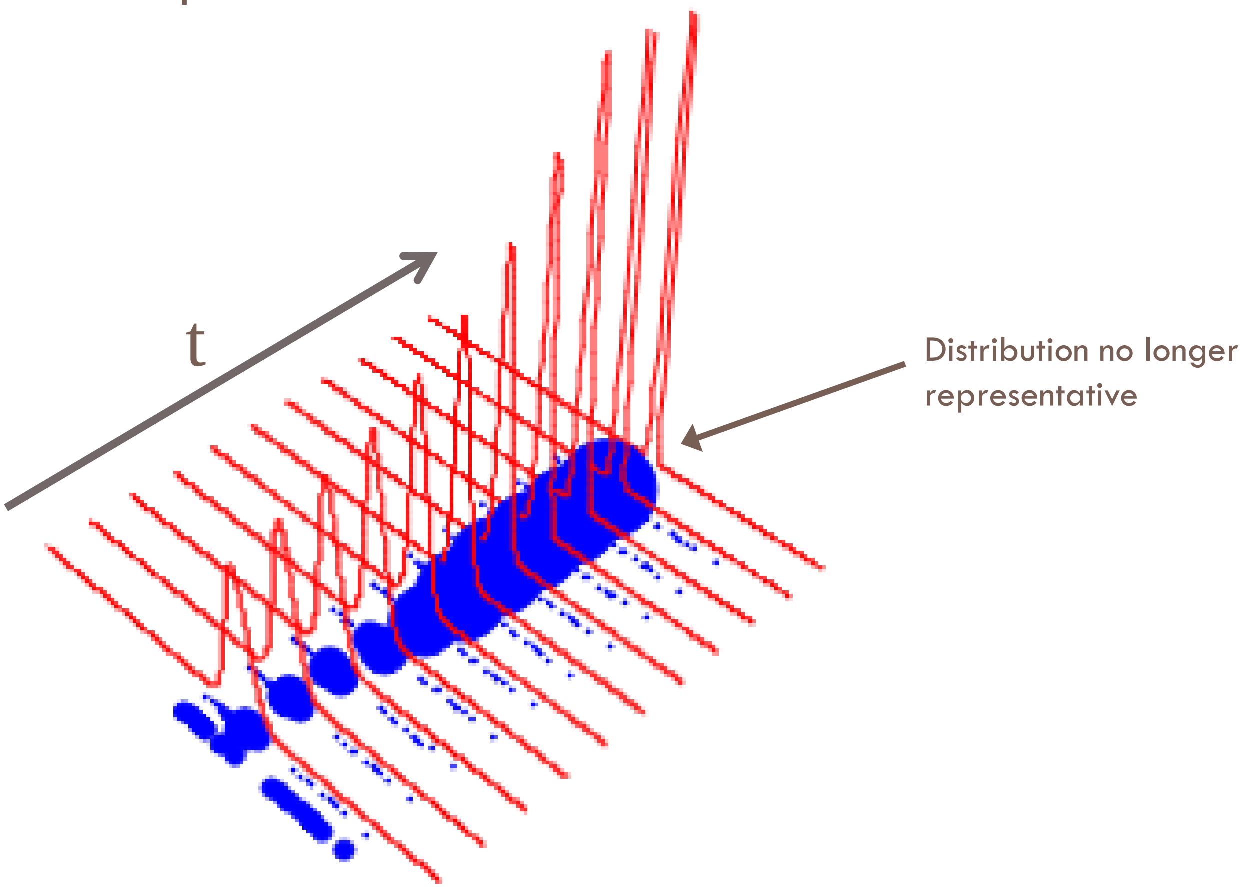


$\left\{ \mathbf{x}_{t-1}^{n}, w_{t-1}^{n} \right\}_{n=1}^{N}$ $\left\{ \mathbf{x}_{t}^{n}, w_{t}^{n} \right\}_{n=1}^{N}$

New sample set is given uniform weights

Degeneracy

- Failing to resample results in **degeneracy.**
  - Iteratively propagating the particles and assigning weights tends to make a few samples dominate the rest



SIR particle filter: predict

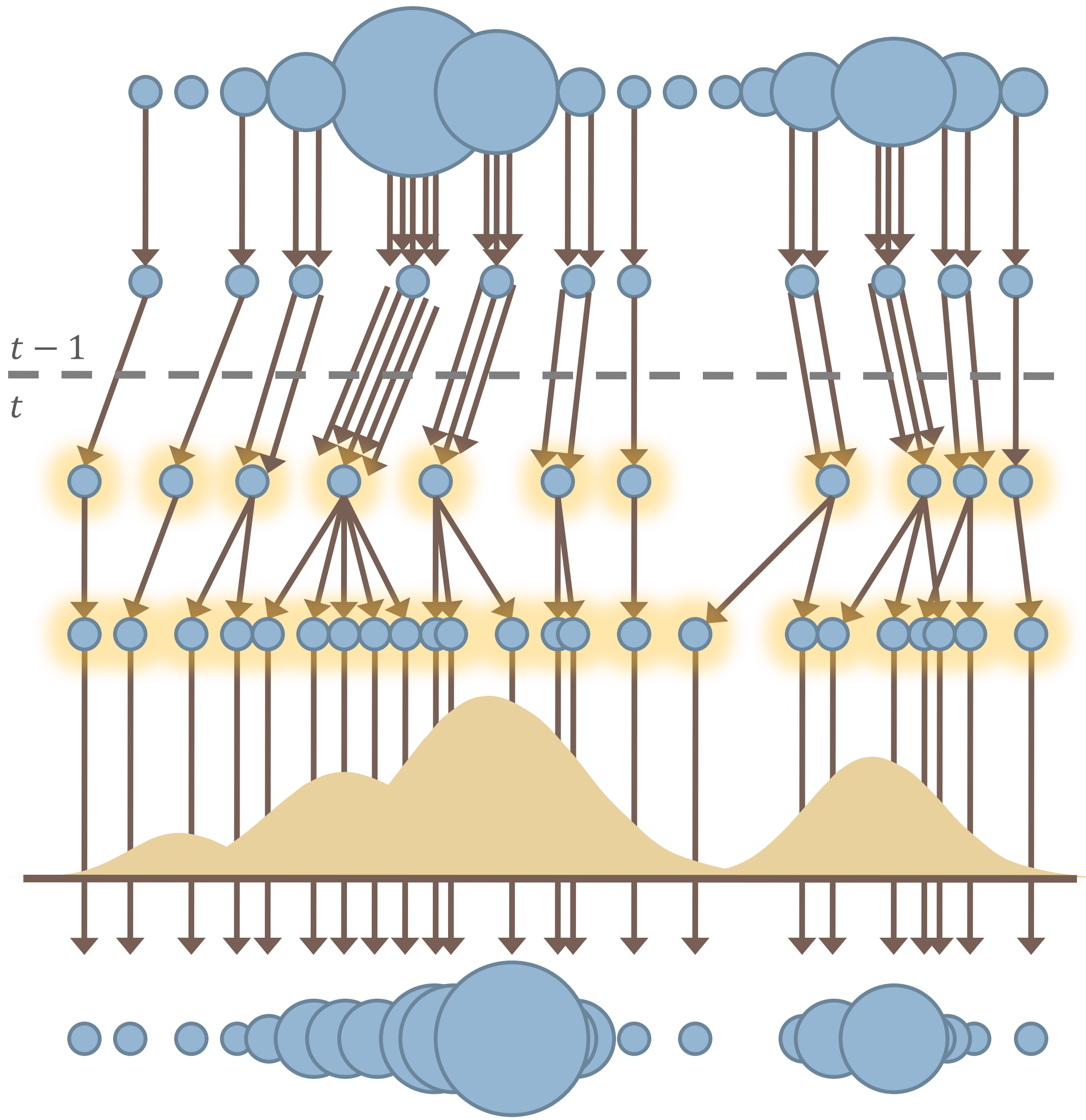
- Apply the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ to every particle!

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{w}_t$$

linear motion model

noise

- **Drift:** apply motion model (no noise)
- **Diffuse:** apply noise to spread particles



Motion model

- Apply the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ to every particle!

$$\mathbf{x}_t = \underset{\substack{\uparrow \\ \text{linear motion model}}}{\mathbf{F}_t} \mathbf{x}_{t-1} + \underset{\substack{\uparrow \\ \text{noise}}}{\mathbf{w}_t}$$

$$\begin{pmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & & \Delta t & \\ & 1 & & \Delta t \\ & & 1 & \\ & & & 1 \end{pmatrix}}_{\mathbf{F}_t} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{pmatrix} + \mathbf{w}_t$$

$$\mathbf{w}_t \sim N(0, \mathbf{Q}_t)$$

SIR particle filter: measure

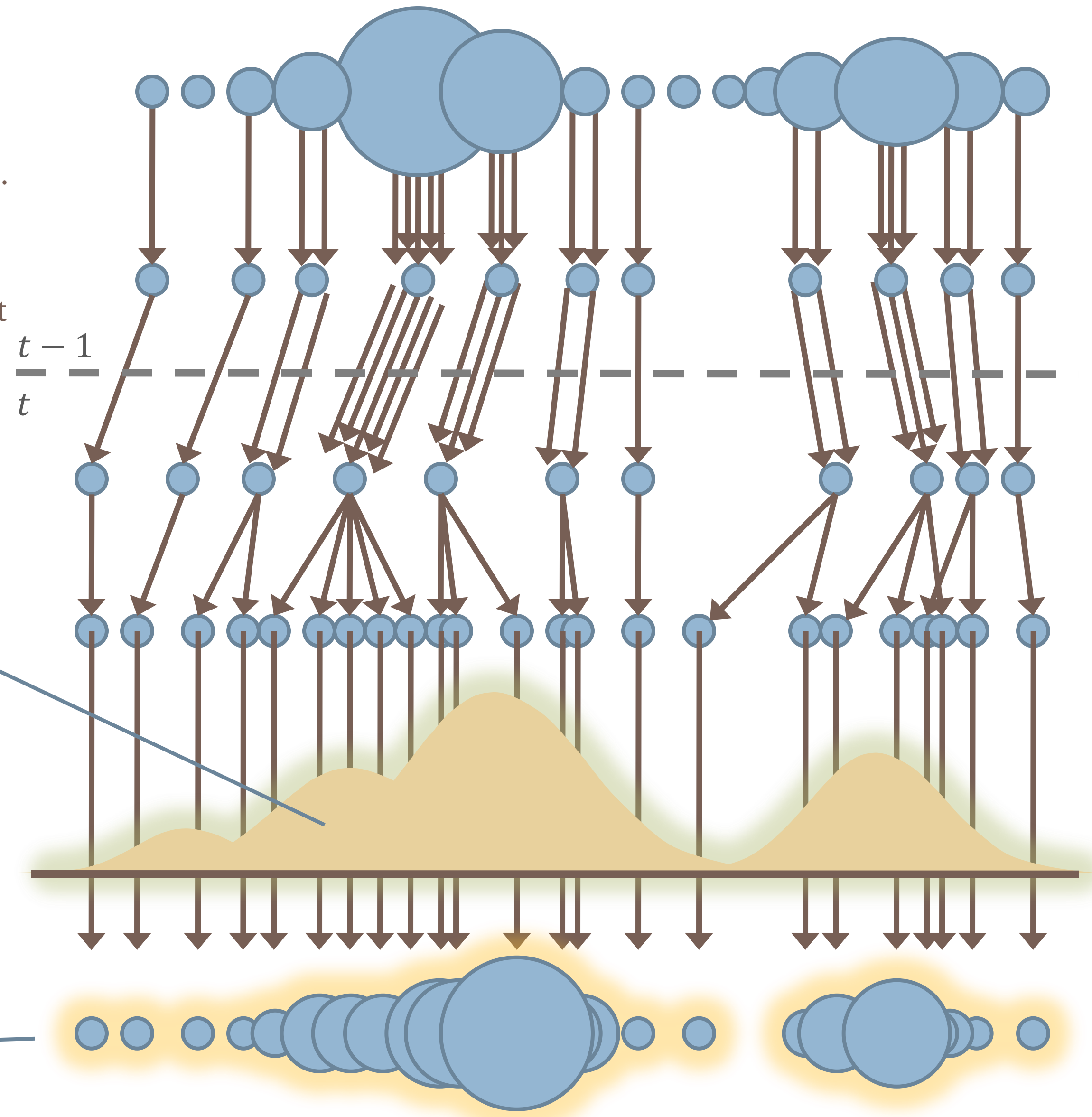
Obtain an observation $\mathbf{z}_t$ for each state estimate $\mathbf{x}_{t.}$

Evaluate likelihood that $\mathbf{x}_t$ gave rise to $\mathbf{z}_t$ using observation model.

$$p(\mathbf{z}_t \mid \mathbf{x}_t)$$

- **Measure:** weights are proportional to the observation likelihood

$$p(\mathbf{x}_t \mid Z_t)$$



Observation model

- **Obtain observation $\mathbf{z}_t$** for each state estimate $\mathbf{x}_t$.

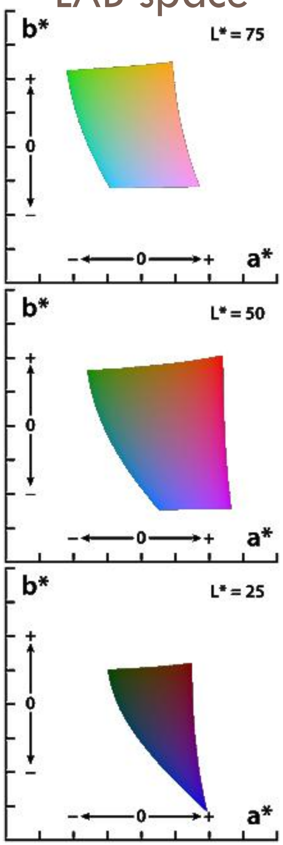
Observation model

- **Obtain observation** $\mathbf{z}_t$ for each state estimate $\mathbf{x}_t$.

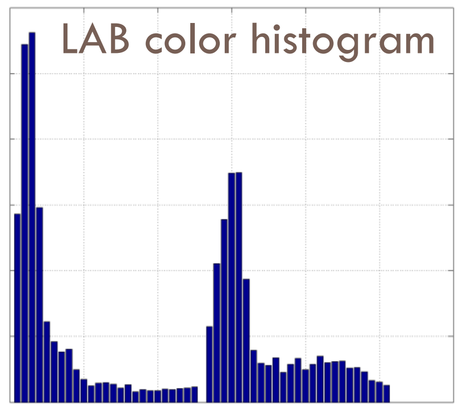
Observation model

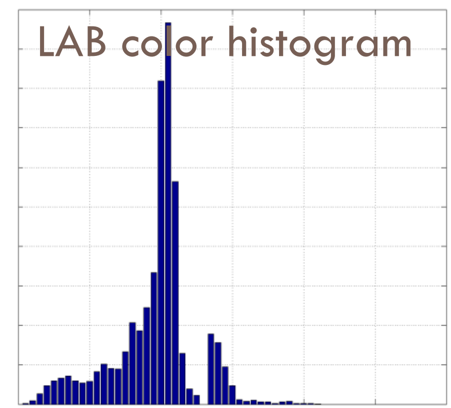
- **Obtain observation** $\mathbf{z}_t$ for each state estimate $\mathbf{x}_t$.

LAB space



$\mathbf{z}_t^1$





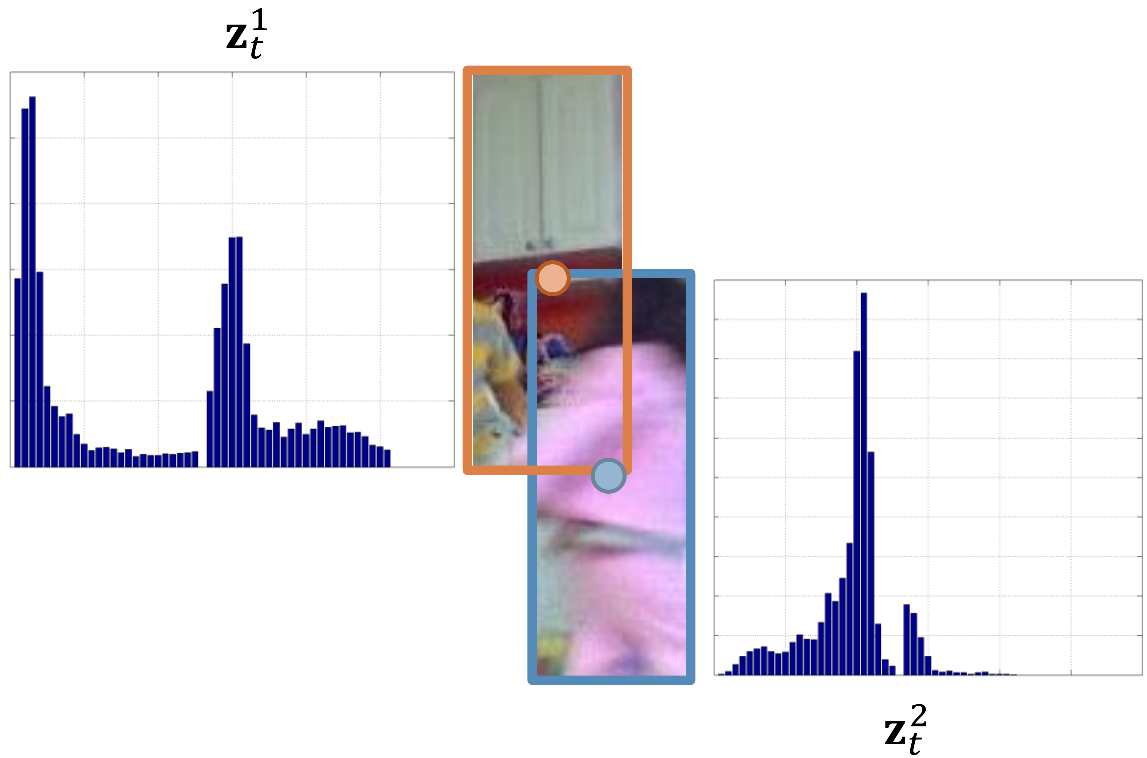
$\mathbf{z}_t^2$

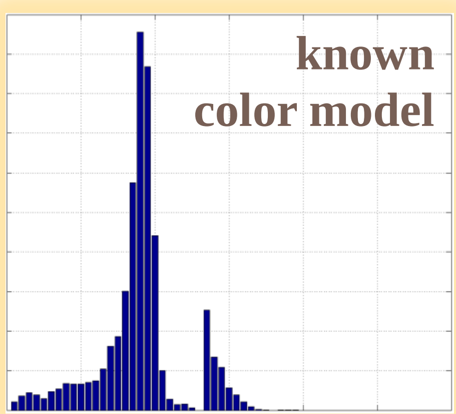
Observations are obtained by converting pixel values within bounding boxes to LAB color space, and concatenating to form an AB channel histogram

Observation model

- **Obtain observation** $\mathbf{z}_t$ for each state estimate $\mathbf{x}_t$.
- **Evaluate likelihood** that an $\mathbf{x}_t$ gave rise to $\mathbf{z}_t$ using observation model.

$$p(\mathbf{z}_t|\mathbf{x}_t^n) = e^{-\lambda\, dist(\mathbf{z}_t,\mathbf{c})}$$



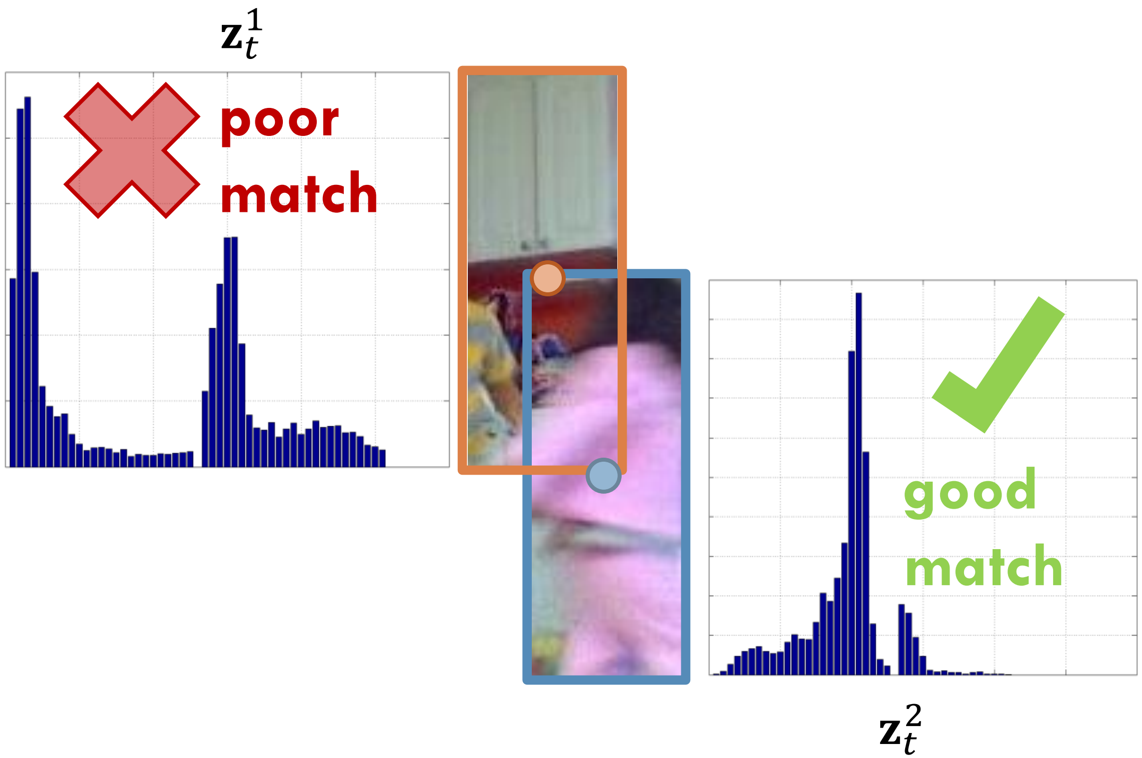


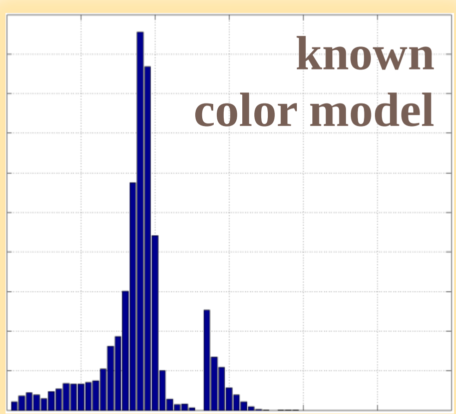
Observation model compares $\mathbf{z}_t$ to a known color model **c** using the **KL divergence.**

Observation model

- **Obtain observation** $\mathbf{z}_t$ for each state estimate $\mathbf{x}_t$.
- **Evaluate likelihood** that an $\mathbf{x}_t$ gave rise to $\mathbf{z}_t$ using observation model.

$$p(\mathbf{z}_t|\mathbf{x}_t^n) = e^{-\lambda\, dist(\mathbf{z}_t, \mathbf{c})}$$





Observation model compares $\mathbf{z}_t$ to a known color model **c** using the **KL divergence.**

Observation model

- **Obtain observation** $\mathbf{z}_t$ for each state estimate $\mathbf{x}_t$.
- **Evaluate likelihood** that an $\mathbf{x}_t$ gave rise to $\mathbf{z}_t$ using observation model.
- **Assign weights** are proportional to the likelihood response

$$w_t^n = p\left(\mathbf{z}_t \mid \mathbf{x}_t^n\right)$$

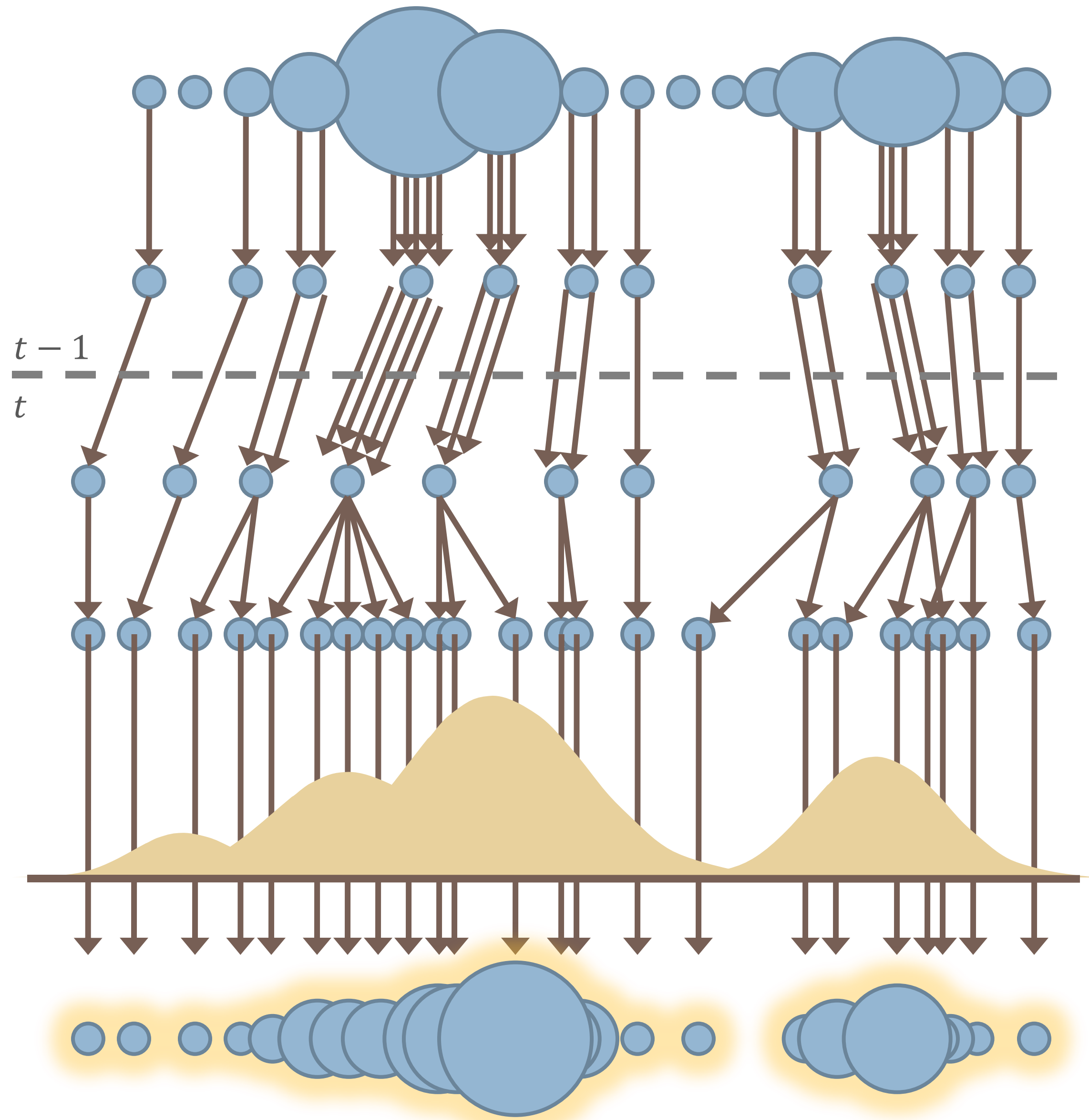
Observation model

- **Obtain observation** $\mathbf{z}_t$ for each state estimate $\mathbf{x}_t$.
- **Evaluate likelihood** that an $\mathbf{x}_t$ gave rise to $\mathbf{z}_t$ using observation model.
- **Assign weights** are proportional to the likelihood response

$$w_t^n = p\left(\mathbf{z}_t \mid \mathbf{x}_t^n\right)$$

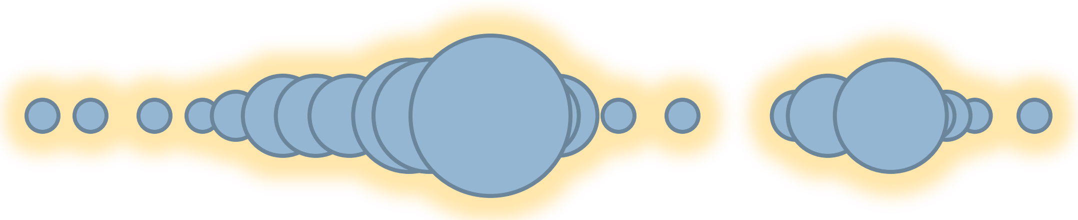
SIR particle filter

- **Begin** with weighted samples from t-1
- **Resample:** draw samples according to $\{w_{t-1}\}^{n=1:N}$
- **Drift:** apply motion model (no noise)
- **Diffuse:** apply noise to spread particles
- **Measure:** weights are assigned by likelihood response
- **Finish:** density estimate



Obtaining a solution

- So far, we do not have an explicit state estimate, we have a cloud of particles!

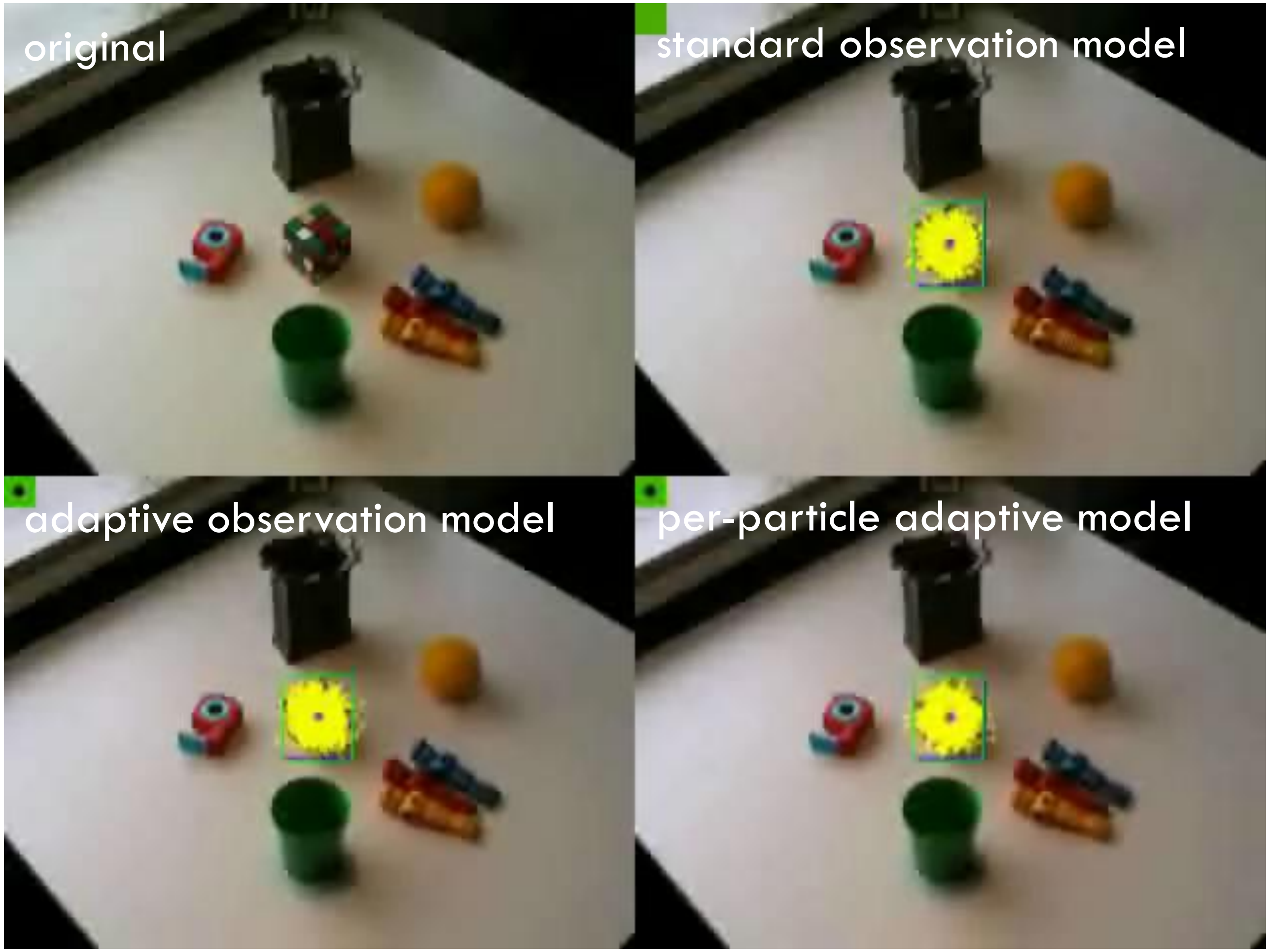


- How do we extract an answer? It depends…
  - Compute a **mean** or **median** particle
  - Confidence: inverse variance
  - For discrete labels, this does not work!
    - Use the mode?

Particle filter

D. Klein, D. Schulz, S. Frintrop, and A. Cremers, [Adaptive Real-Time Video Tracking for Arbitrary Objects]() , International Conference on Intelligent Robots and Systems (IROS), 2010

Particle filter



D. Klein, D. Schulz, S. Frintrop, and A. Cremers, Adaptive Real-Time Video Tracking for Arbitrary Objects , International Conference on Intelligent Robots and Systems (IROS), 2010

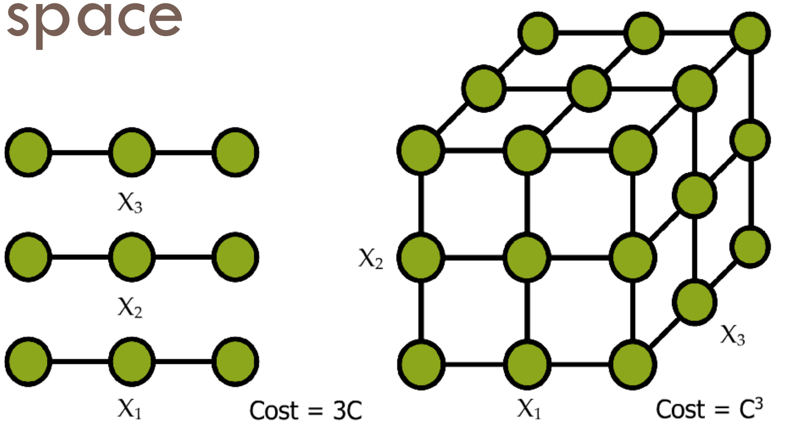
Summary: particle filters

- Represents arbitrary (multi-modal) densities
- Converges to true posterior for nonlinear, non-Gaussian systems
- Efficient: concentrates particles on interesting regions
- Works for many types of state spaces

Summary: particle filters

- Number of samples N is important
  - Use as few as necessary (for efficiency)
  - But use enough to do a good job exploring the state space

- Complexity grows exponentially with dimensionality of the state space



Things to think about…

- Initialization
  - By hand
  - Background subtraction
  - Detection
- Observation models
  - Generative -> render the state on top of the image and compare
  - Discriminative -> classifier or detector score
- Prediction vs Correction
  - If dynamics dominate, cues form the data may be ignored
  - If observation model dominates, tracking is not smooth
- Nonlinear Dynamics
  - Needed for multiple objects, discrete state elements, etc.

Particle filters in action

Michael Isard and Andrew Blake CONDENSATION -- conditional density propagation for visual tracking
International Journal of Computer Vision (IJCV), 29, 1, 5--28, (1998)

Particle filters in action

Homework #5, sequence 1

Particle filters in action

- tracking a ball