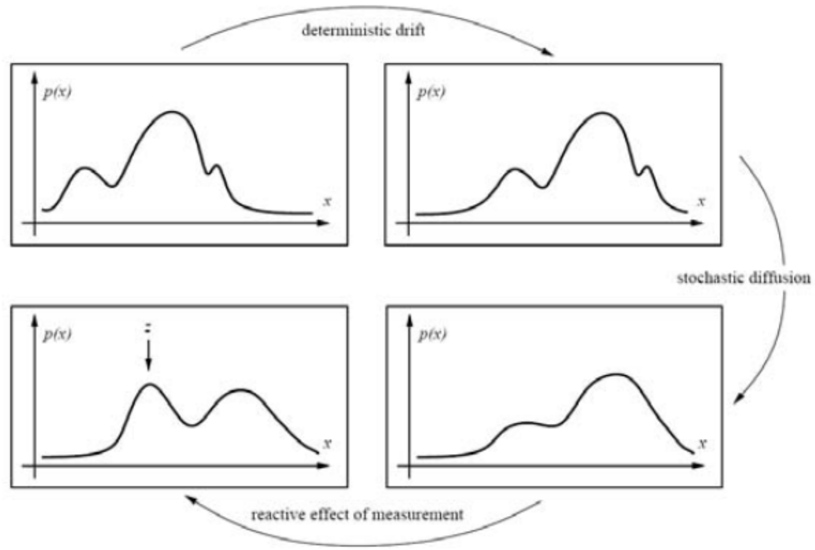
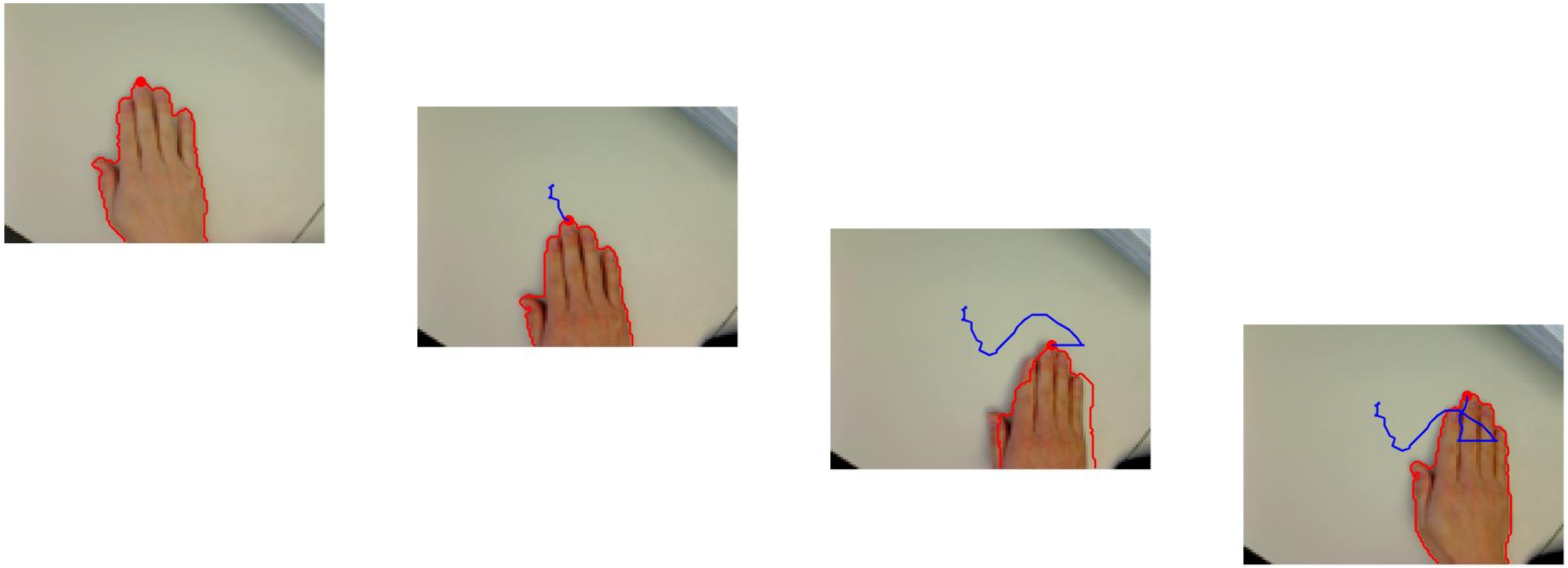


COMP9517: Computer Vision

Tracking

Motion Tracking

- Tracking is the problem of generating an inference about the motion of an object given a sequence of images



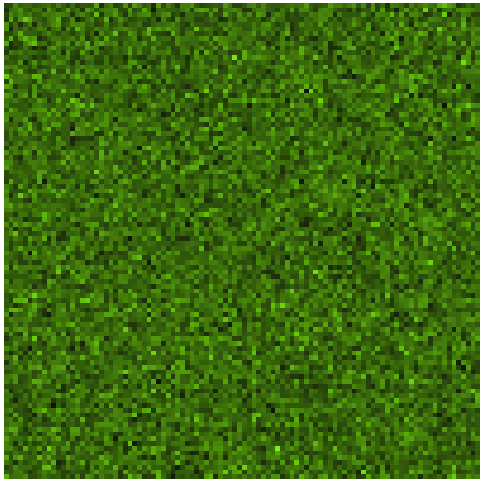
Applications

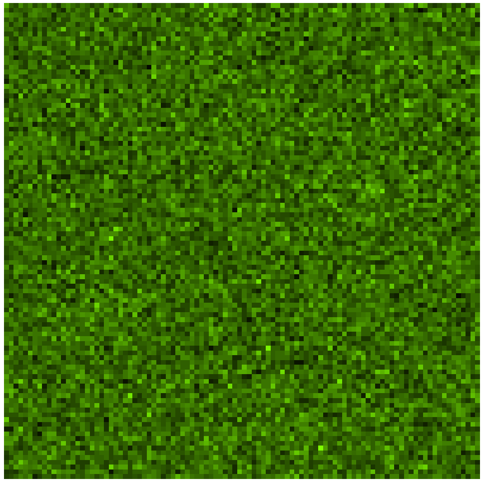
- **Motion capture**
  - Record motion of people to control cartoon characters in animations
  - Modify the motion record to obtain slightly different behaviours
- **Recognition from motion**
  - Determine the identity of a moving object
  - Assess what the object is doing
- **Surveillance**
  - Detect and track objects in a scene for security
  - Monitor their activities and warn if anything suspicious happens
- **Targeting**
  - Decide which objects to shoot in scene
  - Make sure the objects get hit

Difficulties in Tracking

- **Loss of information** caused by projection of the 3D world on a 2D image
- **Noise** in images
- Complex object **motion**
- **Non-rigid** or articulated nature of objects
- Partial and full object **occlusions**
- Complex object **shapes**
- Scene **illumination** changes
- **Real-time** processing requirements

Example Tracking Problem





Single moving microscopic particle

- Imaged with signal-to-noise ratio (SNR) of 1.5

Human visual motion perception

- Not so accurate and reproducible in quantification
- Good at integrating spatial and temporal information
- Powerful in making associations and predictions

Computer vision challenges

- Integration of spatial and temporal information
- Modeling and incorporation of prior knowledge
- Probabilistic rather than deterministic approach

Bayesian estimation methods…

Motion Assumptions

- When moving objects do not have unique texture or colour, the characteristics of the motion itself must be used to connect detected points into trajectories

- Assumptions about each moving object:
  - Location changes smoothly over time
  - Velocity (speed and direction) changes smoothly over time
  - Can be at only one location in space at any given time
  - Not in same location as another object at the same time

Topics

- **Bayesian inference**

  Using *probabilistic models* to perform tracking

- **Kalman filtering**

  Using *linear model assumptions* for tracking

- **Particle filtering**

  Using *nonlinear models* for tracking

Bayesian Inference

Problem Definition

- A moving object has a **state** which evolves over time

Random variable: X_i

Specific value: x_i

can contain any quantities of interest (position, velocity, acceleration, shape, intensity, colour, …)

- The state is **measured** at each time point

Random variable: Y_i

Specific value: y_i

in computer vision the measurements are typically features computed from the images

- Measurements are combined to estimate the state

Three Main Steps

- **Prediction**: use the measurements $(y_0, y_1, \ldots, y_{i-1})$ up to time $i-1$ to predict the state at time i

$$P(X_i \mid Y_0 = y_0, Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1})$$

- **Association**: select the measurements at time i that are related to the object state

- **Correction**: use the incoming measurement y_i to update the state prediction

$$P(X_i \mid Y_0 = y_0, Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}, Y_i = y_i)$$

Independence Assumptions

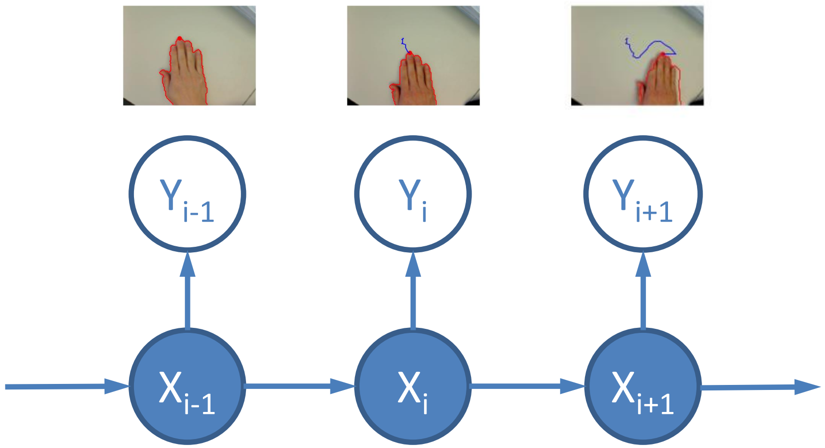
- Current state depends only on the immediate past

$$P(X_i \mid X_0, X_1, \ldots, X_{i-1}) = P(X_i \mid X_{i-1})$$

- Measurements depend only on the current state

$$P(Y_i, Y_j, \ldots, Y_k \mid X_i) = P(Y_i \mid X_i) P(Y_j, \ldots, Y_k \mid X_i)$$

These assumptions imply the tracking problem has the structure of inference on a hidden Markov model



Tracking by Bayesian Inference

- Prediction

$$P(X_i \mid y_0, y_1, \ldots, y_{i-1}) = \int P(X_i, X_{i-1} \mid y_0, y_1, \ldots, y_{i-1}) \, dX_{i-1}$$

$$= \int P(X_i \mid X_{i-1}, \cancel{y_0, y_1, \ldots, y_{i-1}}) P(X_{i-1} \mid y_0, y_1, \ldots, y_{i-1}) \, dX_{i-1}$$

$$= \int \underbrace{P(X_i \mid X_{i-1})}_{\text{dynamics model}} \underbrace{P(X_{i-1} \mid y_0, y_1, \ldots, y_{i-1})}_{\text{posterior of previous time}} \, dX_{i-1}$$

Tracking by Bayesian Inference

- Correction

$$P(X_i \mid y_0, y_1, \ldots, y_i) = \frac{P(X_i, y_0, y_1, \ldots, y_i)}{P(y_0, y_1, \ldots, y_i)}$$

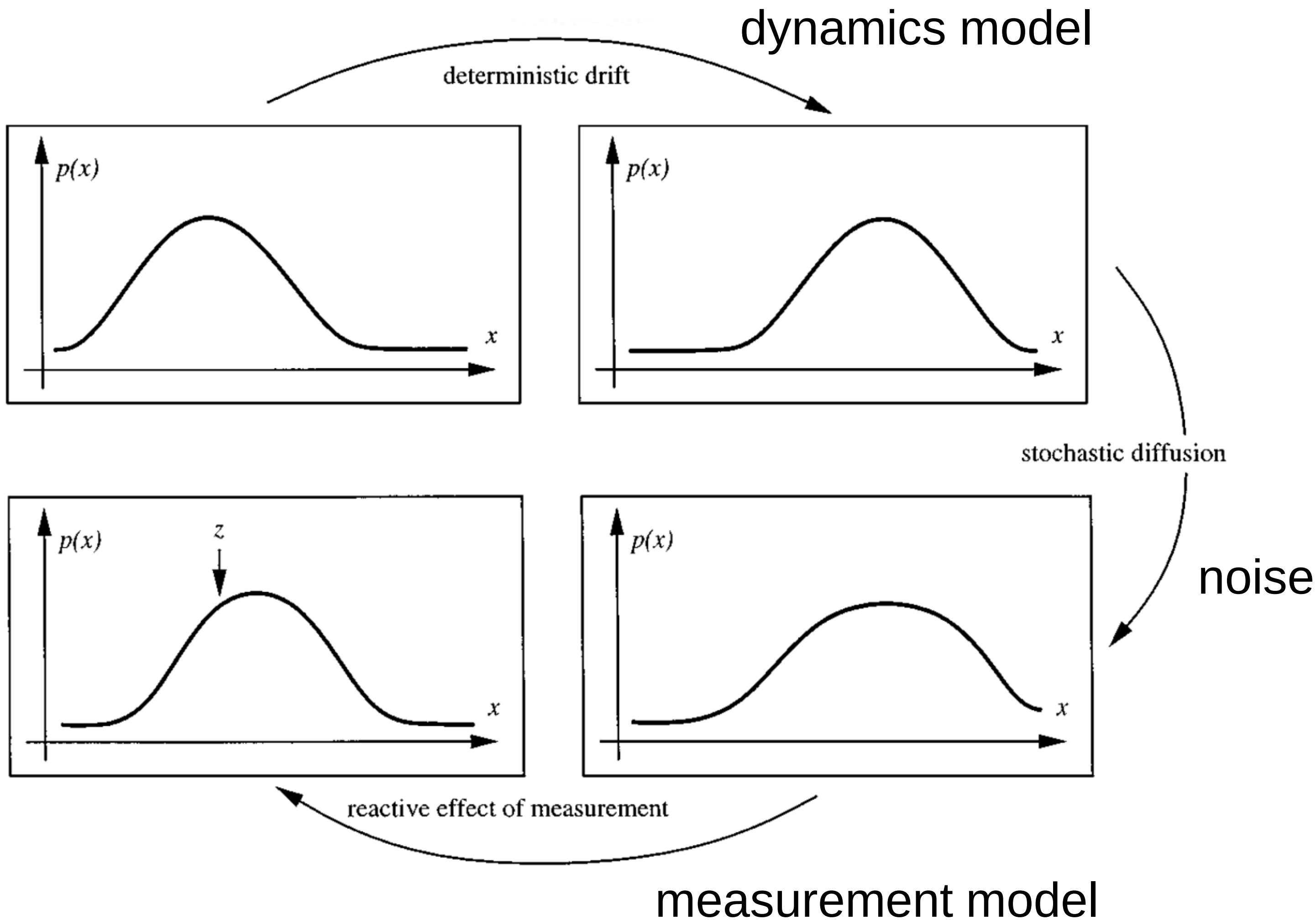
$$= \frac{P(y_i \mid X_i, \cancel{y_0, y_1, \ldots, y_{i-1}})P(X_i \mid y_0, y_1, \ldots, y_{i-1})P(y_0, y_1, \ldots, y_{i-1})}{P(y_0, y_1, \ldots, y_i)}$$

$$= \underbrace{P(y_i \mid X_i)}_{\text{measurement model}}\underbrace{P(X_i \mid y_0, y_1, \ldots, y_{i-1})}_{\text{prediction of current state}}\underbrace{\frac{P(y_0, y_1, \ldots, y_{i-1})}{P(y_0, y_1, \ldots, y_i)}}_{\text{constant}}$$

$$\propto P(y_i \mid X_i)P(X_i \mid y_0, y_1, \ldots, y_{i-1})$$

Kalman Filtering

Probability Density Propagation



Linear / Gaussian Assumption

If we assume the dynamics (state transition) model and the measurement model to be linear, and the noise to be additive Gaussian, then all the probability densities will be Gaussians

$$x \sim N(\mu, \Sigma)$$

- The state is advanced by multiplying with some known matrix and then adding a zero-mean normal random variable

$$x_i \sim N(D_i x_{i-1}, \Sigma_{d_i}) \qquad x_i = D_i x_{i-1} + w_{i-1}$$

- The measurement is obtained by multiplying the state by some matrix and then adding a zero-mean normal random variable

$$y_i \sim N(M_i x_i, \Sigma_{m_i}) \qquad y_i = M_i x_i + v_i$$

$$x_i \sim N(Ax_{i-1}, Q)$$

$$y_i \sim N(Hx_i, R)$$

Kalman Filtering

Prediction

1. Predict state

$$x_i^- = Ax_{i-1}$$

2. Predict covariance

$$P_i^- = AP_{i-1}A^T + Q$$

Correction

1. Compute Kalman gain

$$K_i = P_i^- H^T (H P_i^- H^T + R)^{-1}$$

2. Correct state with measurement

$$x_i = x_i^- + K_i (y_i - H x_i^-)$$

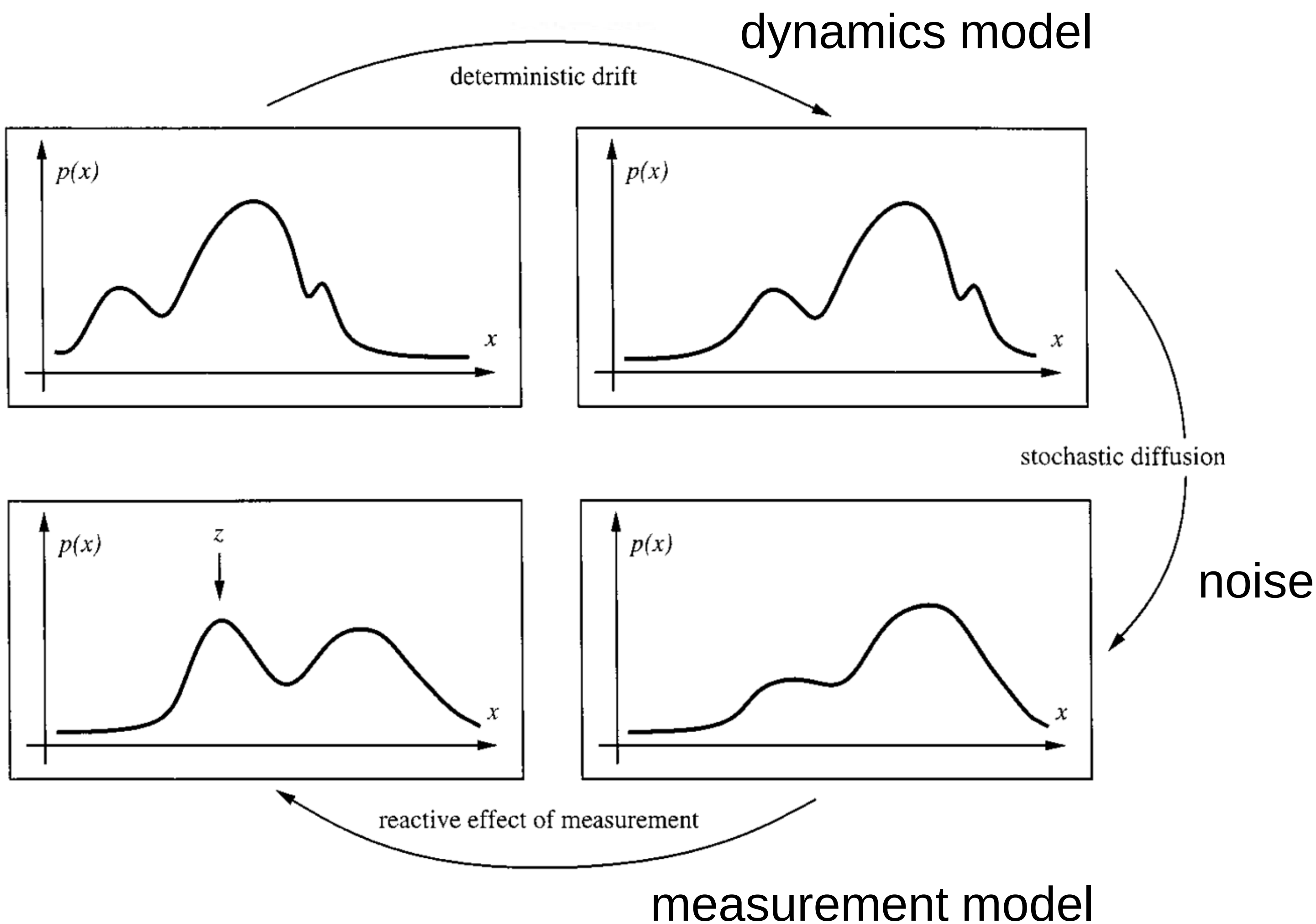
3. Correct covariance

$$P_i = (I - K_i H) P_i^-$$

$$i \rightarrow i+1$$

Particle Filtering

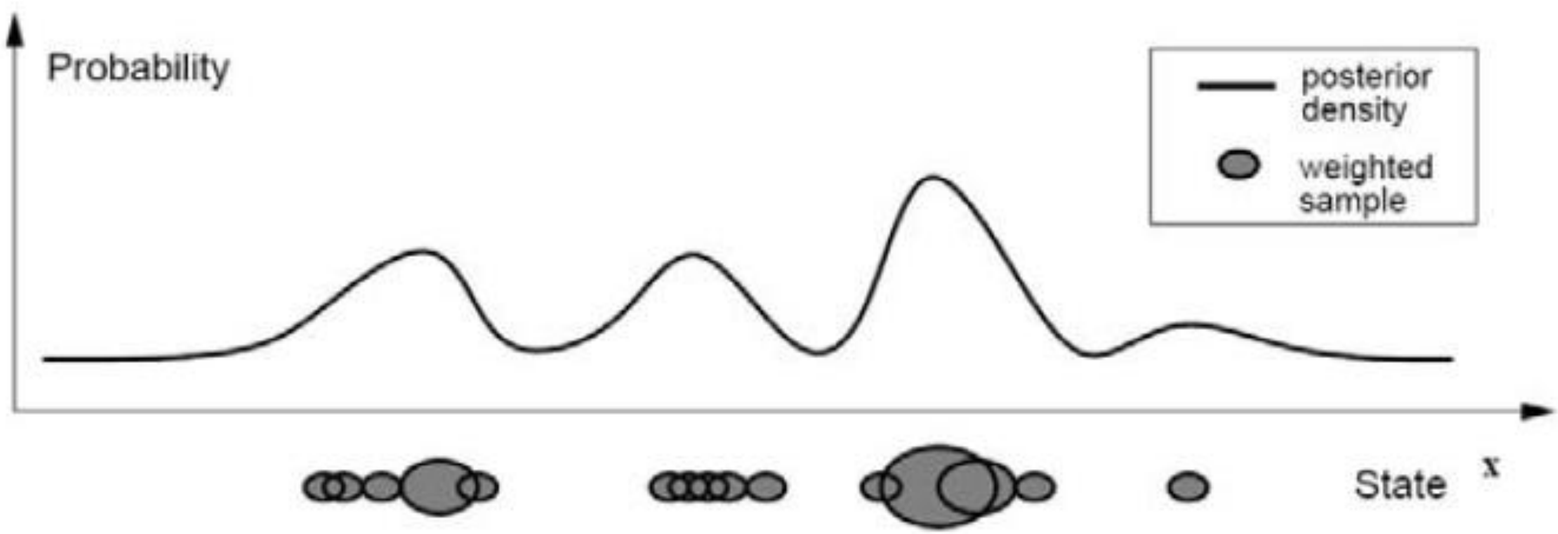
Probability Density Propagation



Non-Linear / Non-Gaussian Case

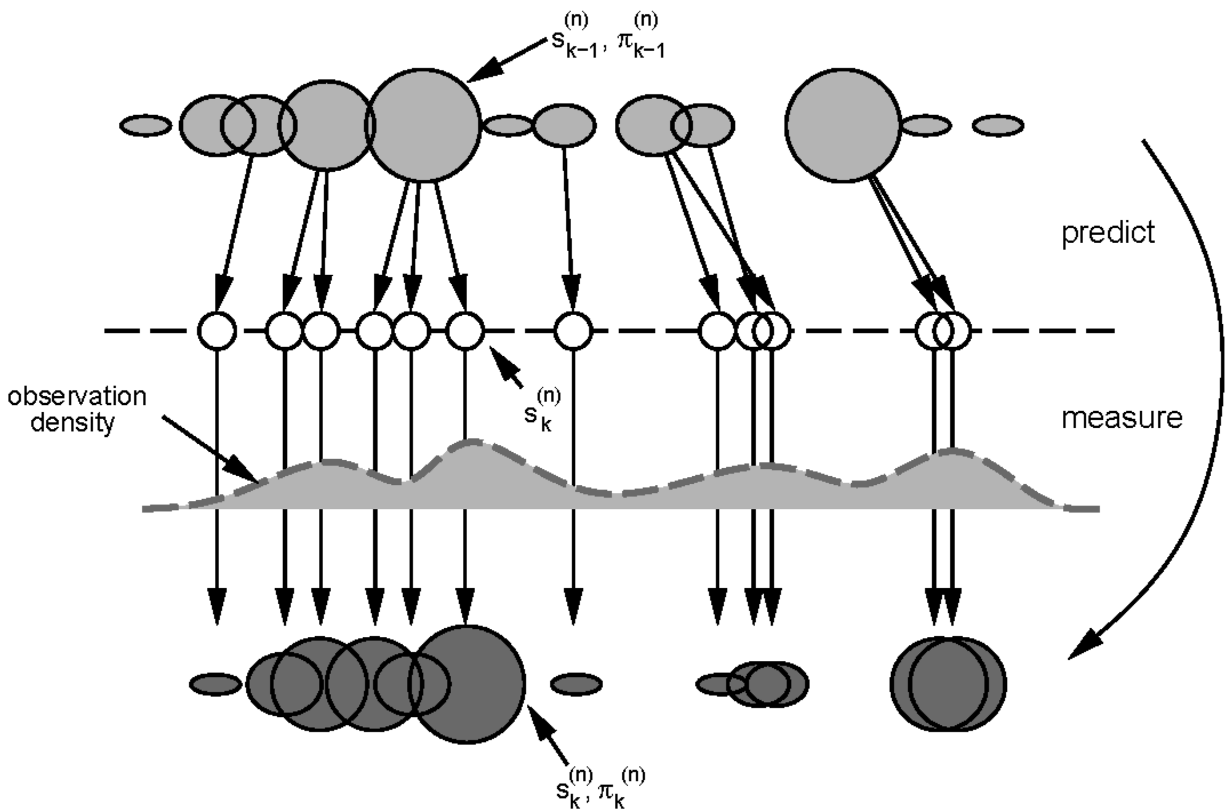
- Represent the conditional state density by a set of samples (particles) with corresponding weights (importance)

$$P(X_i \mid y_0, y_1, \ldots, y_i) \rightarrow \{s_i^{(n)}, \pi_i^{(n)}\}_{n=1}^{N}$$



Particle Filtering

- Propagate each sample using the dynamics model and obtain its new weight using the measurement model



Particle Filtering Algorithm

Iterate

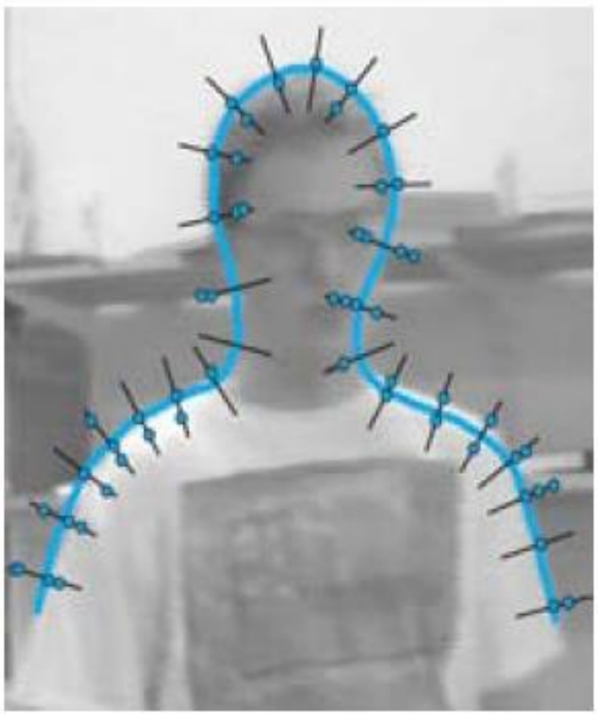
From the "old" sample set $\{\mathbf{s}_{k-1}^{(n)}, \pi_{k-1}^{(n)}, c_{k-1}^{(n)}, n = 1, \ldots, N\}$ at time-step t_{k-1}, construct a "new" sample set $\{\mathbf{s}_k^{(n)}, \pi_k^{(n)}, c_k^{(n)}, n = 1, \ldots, N\}$ for time t_k.

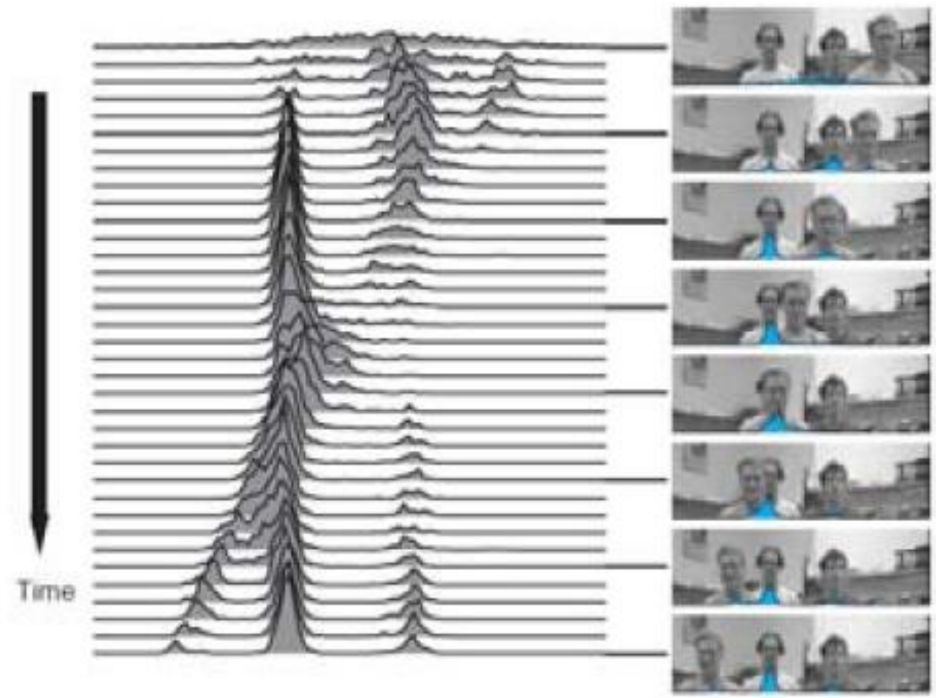
Construct the n^{th} of N new samples as follows:

1. **Select** a sample $\mathbf{s'}_k^{(n)}$ as follows:
   (a) generate a random number $r \in [0, 1]$, uniformly distributed.
   (b) find, by binary subdivision, the smallest j for which $c_{k-1}^{(j)} \geq r$
   (c) set $\mathbf{s'}_k^{(n)} = \mathbf{s}_{k-1}^{(j)}$
2. **Predict** by sampling from
$$p(\mathcal{X}_k | \mathcal{X}_{k-1} = \mathbf{s'}_k^{(n)})$$
   to choose each $\mathbf{s}_k^{(n)}$. For instance, in the case that the dynamics are governed by a linear AR process, the new sample value may be generated as: $\mathbf{s}_k^{(n)} = A\,\mathbf{s'}_k^{(n)} + (I - A)\overline{\mathcal{X}} + B\mathbf{w}_k^{(n)}$ where $\mathbf{w}_k^{(n)}$ is a vector of standard normal random variates, and BB^T is the process noise covariance.
3. **Measure** and weight the new position in terms of the measured features $\mathbf{Z}_k$:
$$\pi_k^{(n)} = p(\mathbf{Z}_k | \mathcal{X}_k = \mathbf{s}_k^{(n)})$$
   then normalise so that $\sum_n \pi_k^{(n)} = 1$ and store together with cumulative probability as $(\mathbf{s}_k^{(n)}, \pi_k^{(n)}, c_k^{(n)})$ where
$$\begin{aligned} c_k^{(0)} &= 0, \\ c_k^{(n)} &= c_k^{(n-1)} + \pi_k^{(n)} \quad \text{for} \quad n = 1, \ldots, N. \end{aligned}$$

Example Application

- Tracking of active contour representations of objects

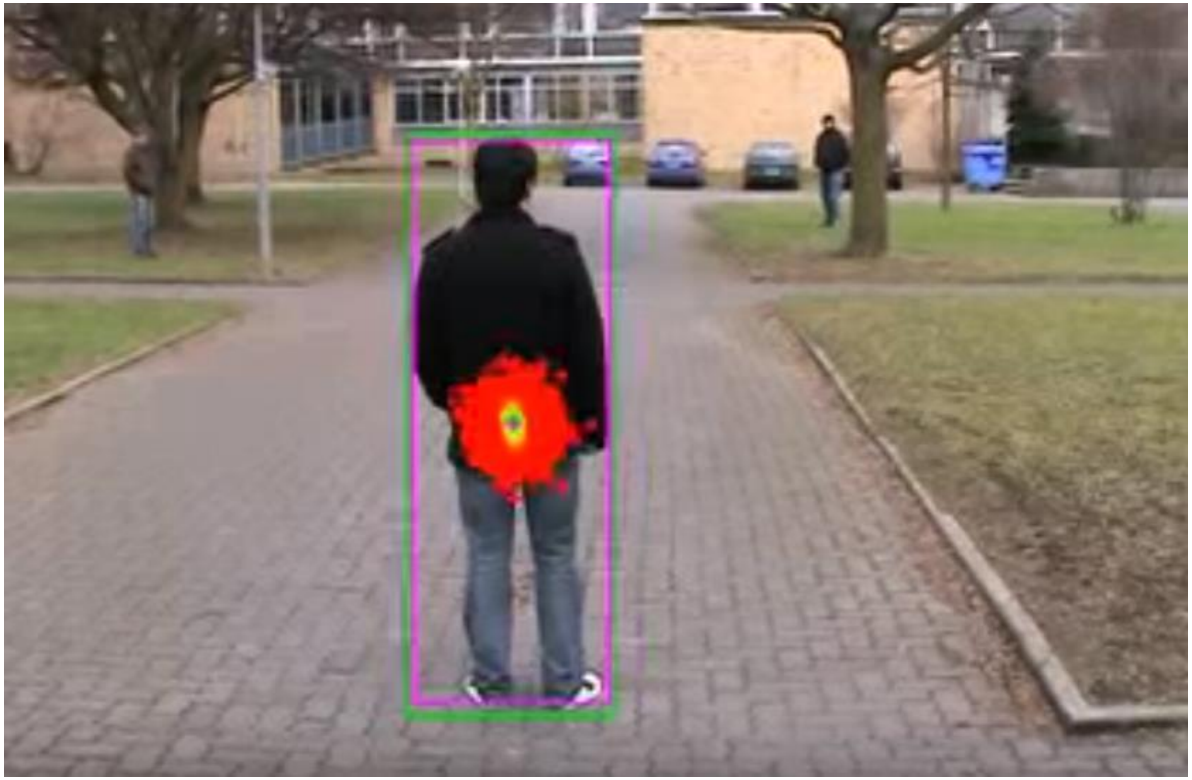




Particle filtering is also known variously as sequential Monte Carlo (SMC) filtering, bootstrap filtering, the condensation algorithm…

Example Application

- Tracking of object location in presence of clutter



https://www.youtube.com/watch?v=j-duyzShJ_o

References and Acknowledgements

- Chapters 5 and 8 of Szeliski 2010
- Chapter 18 of Forsyth and Ponce 2011
- Chapter 9 of Shapiro and Stockman 2001
- Paper by M. Isard and A. Blake 1998
  CONDENSATION: Conditional density propagation for visual tracking
  Available online via the UNSW Library
- Images drawn from the above references

This lecture is shorter than usual to give you extra time to check the above sources for more details