Human Activity and Vision Summer School

Probabilistic tracking

jean-marc odobez

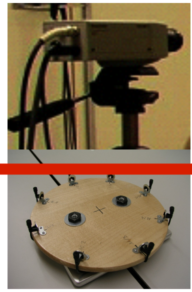
03.10.2012

idiap RESEARCH INSTITUTE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

the overall goal: to infer relevant information from audio-visual human scenes

audio-visual scenes



- **representation** (what is a person?)
- **detection** (are there any people?)
- **localization** (where are they?)
- **tracking** (where do they go?)
- **identification** (who are they?)
- **activity recognition & discovery** (what do they do? what do they look at?, do they interact? who do they interact with? what do they do together?, …)

Goal and outline

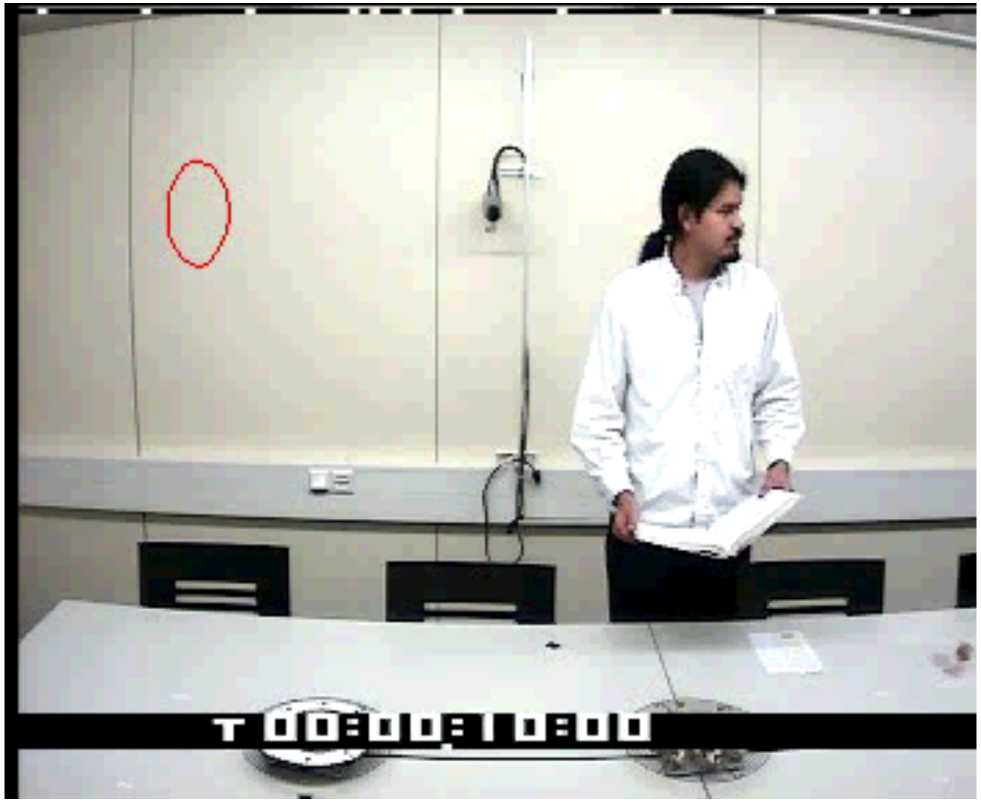
- Introduction
  - State-space example
  - Dynamic models

- Bayesian approaches
  - Kalman filter
  - Sampling methods (Particle filter)

- Note
  - many slides in the presentation (available on website)
  - not all presented here => they provide more complementary information

Introduction

- Visual tracking: a visual tracking-based definition…



"Tracking is the problem of **generating an inference about the motion of an object given a sequence of images**. Good solutions of this problem have a variety of applications…"

Forsyth and Ponce, *Computer Vision: A modern approach*, 2003.

Sources of trouble

- Why is it harder that it might seem?
  - dimension loss
  - low image quality: low contrast, noise, motion blur
  - variability of visual appearance
  - occlusions, partial to total
  - clutter
  - unpredictable motions
  - constraints on computational complexity

- A lot in common with other computer vision tasks!

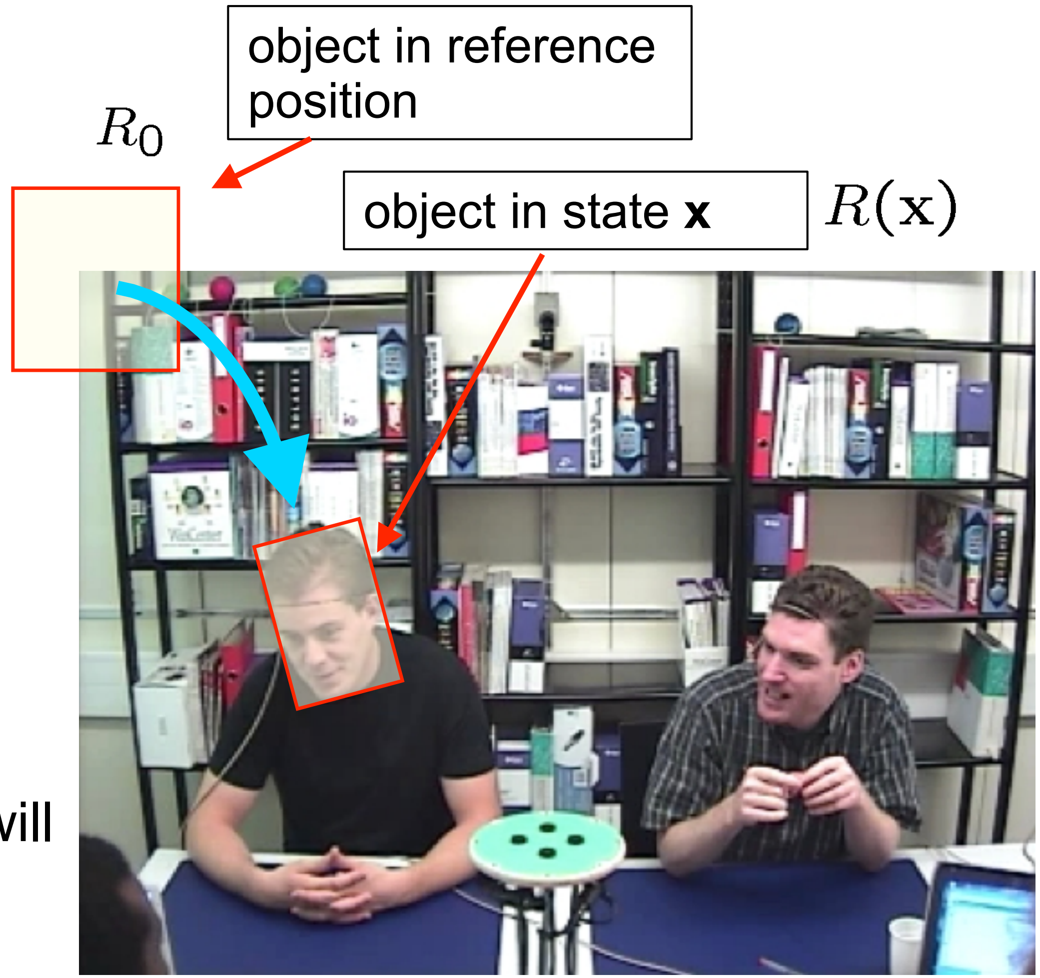
What do we want to estimate ? object state space $\mathbf{x}_t$

- In most cases:

  geometric state space

  - maps an object model from a reference position into the image

  - e.g. box:
    - mapping
      - translation
      - scaling, rotation, shear
    - allows to define a region of the images where measurements will be made

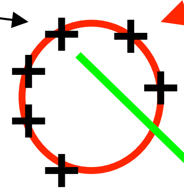


Object state: space of geometric transformations

point p of shape $p = \begin{pmatrix} x \\ y \end{pmatrix}$

$$p' = Ap + b = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} p + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

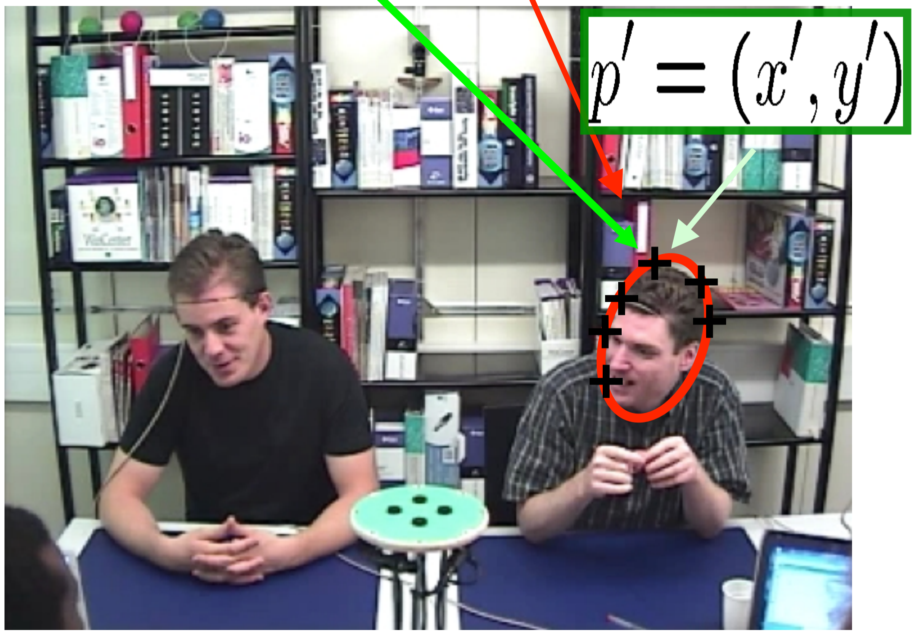
R_0



object in reference position

object in state **x**

- may needs mapping of individual points
- object
  - shape template: set of 2D points
  - image patch: points + image value inside a region (e.g. box, ellipse)
- mapping
  - b : translation
  - A : linear components (scalings, rotation, shear)



(b,A); 2D affine state space to be estimated

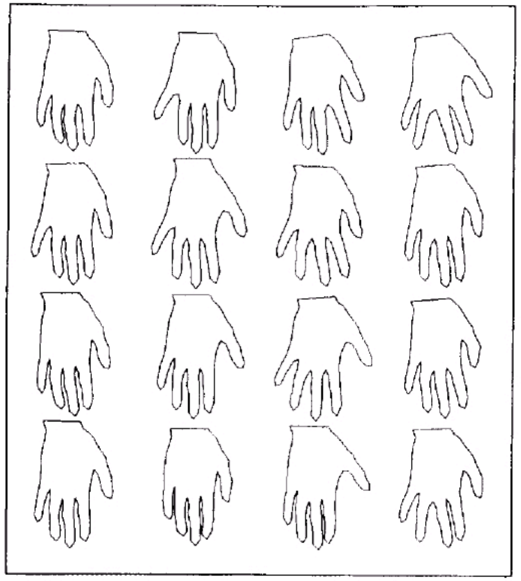
More complex models: eigen Shapes

- Object: shape represented by a set of point

$$s = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}$$

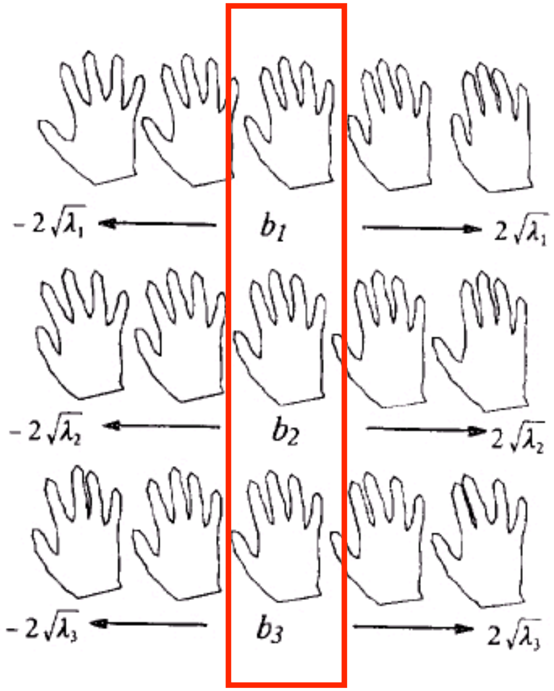
[Taylor and Cootes's *Active Shapes*]

- Model: using training data, principal shape variations (modes) learnt off-line (PCA) : provide a linear parameterization of the shape



Training data

$$s = \bar{s} + \sum_i b_i \phi_i$$

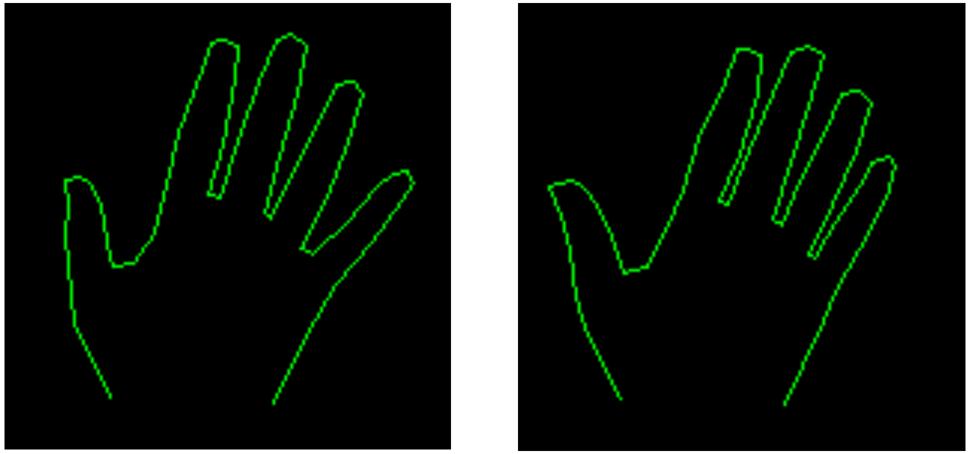


mean

Eigen Shapes

- State: few deformation modes around average shape template, plus affinity parameters (to move the shape into the image)

$$\mathbf{x} = (\mathbf{b}, A, \phi_{1:m}) \in \mathbb{R}^{6+m}$$



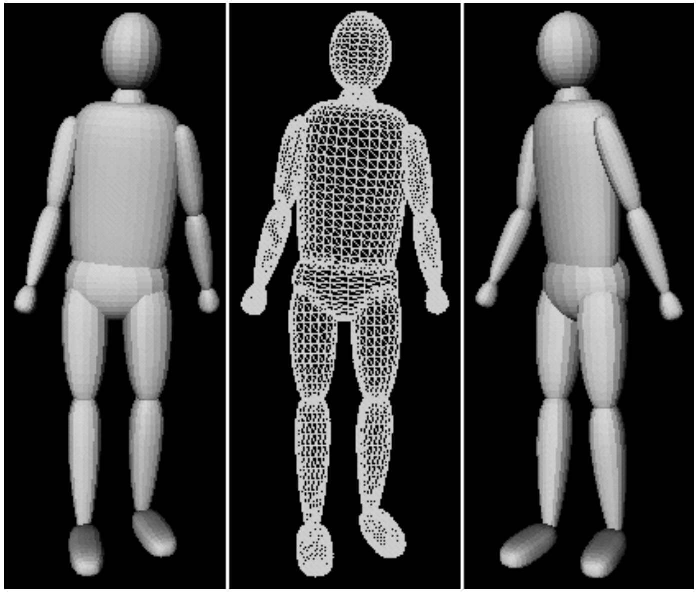
[Taylor and Cootes's *Active Shapes*]

Into 3D Models

- State: 3D pose of a set of r parameterized parts (possibly articulated)

$$\mathbf{x} = (T_0, R_{1:r}, \boldsymbol{\theta}_{1:r}) \in \mathbb{R}^{3+r(1+m)}$$

- Context: pose tracking of objects of known type (manufactured objects, human body) whose geometry is known, assumed, or learnt



[Sminchisescu's body model] [Sidenbladh's body tracker]

More generally

- State captures various aspects of tracked objects
  - 3D pose/shape [cont.]
  - 2D pose/shape [cont. or disc.]
  - Auxiliary variables:
    - color [cont. or disc.] : histogram template
    - identity [disc.] : for multi object tracking
    - activity [disc.] : is the person walking, running, speaking ?

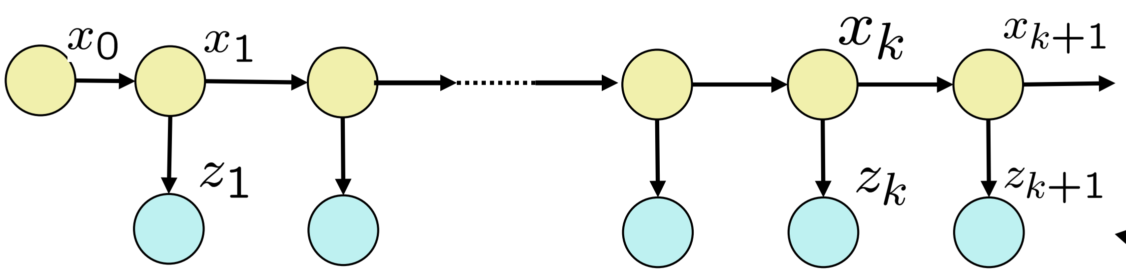
    => help in defining
      - better (more precise), simpler observation/motion models
      - cf introduction of latent variables for distribution modeling
- Note:
  - state parameters should be ‘observable, measurable’:
    - parameters should have an impact on the measurements

Outline

- Bayesian tracking
  - Model
  - Parameter dynamics modeling
  - Sequential estimation
    - Kalman filter
    - Particle filter

Probabilistic approach

- Dynamical state-space model



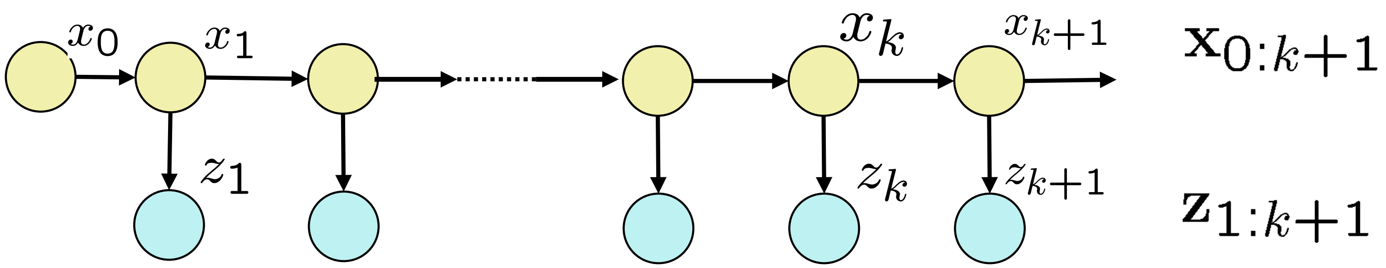
e.g. the true location of the object in the image

e.g. the observed location measured in the image (e.g. as given by a detector)

- What we expect
  - handle uncertainties (noise, ambiguities, clutter, crude modeling…)
  - more than a single point estimate => access to distribution

    e.g. make good prediction about where (region) to search for object in next frame
  - allows parameter learning
  - well established tools

Dynamical state-space model



- Assumptions
  - hidden process is a Markov chain

$$p(\mathbf{x}_{k+1}|\mathbf{x}_{0:k}) = \boxed{\qquad\qquad}$$

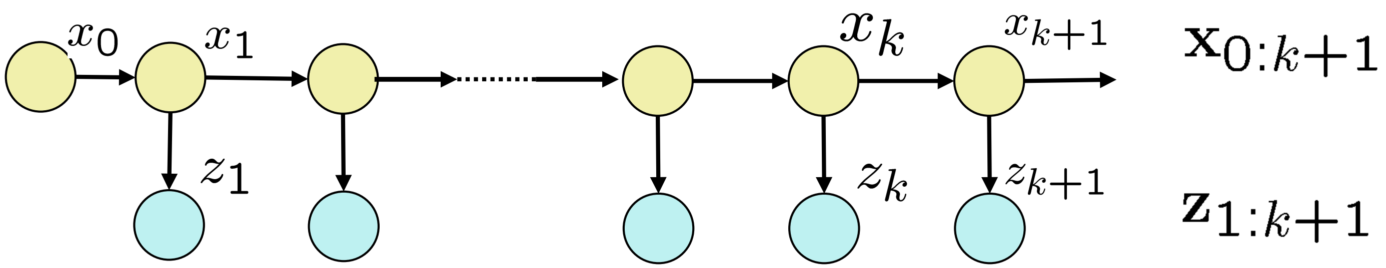
  - observations: conditionally independent given the state

$$p(\mathbf{z}_{k+1}|\mathbf{z}_{1:k}, \mathbf{x}_{0:k+1}) = \boxed{\qquad\qquad}$$

  - joint law up to time k

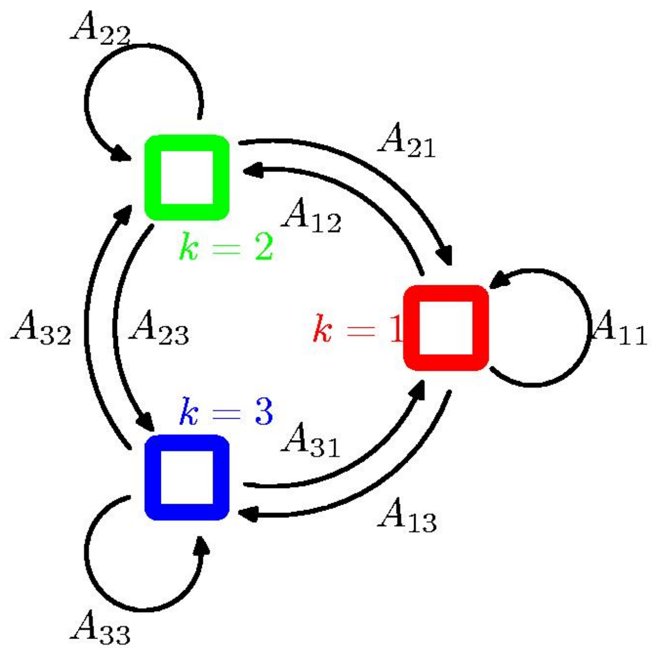
$$p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k}) = p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{x}_i|\mathbf{x}_{i-1}) p(\mathbf{z}_i|\mathbf{x}_i)$$

Difference with HMM



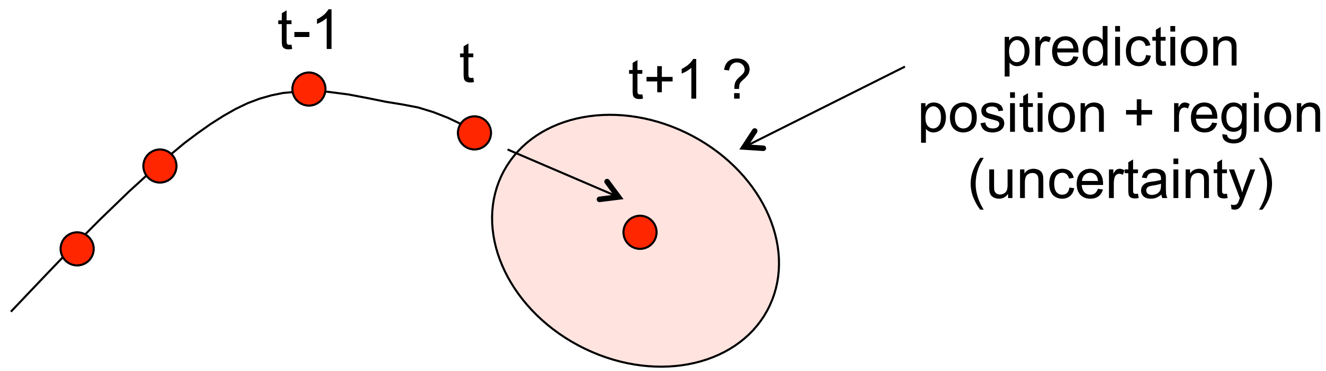
- HMM: hidden state **discrete**
  - Dynamical model: probability table
  - Example of transition

$$A_{ij} = p(\mathbf{x}_k = j | \mathbf{x}_{k-1} = i)$$



- Here, **state continuous**
  - How do we model state dynamical process ?

Dynamical model : intuition



Smooth trajectories

- predictions depends on past observations : auto-regressive process
  (can be driven by physics principles)
- includes uncertainty about prediction

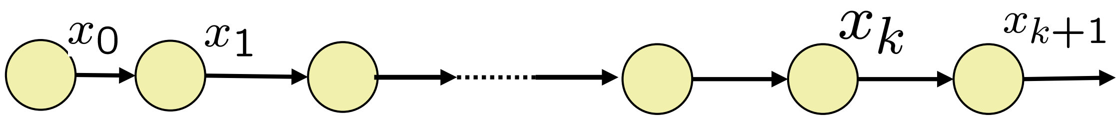
ARP order

$$\mathbf{x}_k = F(\mathbf{x}_{k-1}, ..., \mathbf{x}_{k-K}, w_k)$$

possibly non-linear

driven by
Independent noise $w_k \sim \mathcal{N}(0, I)$

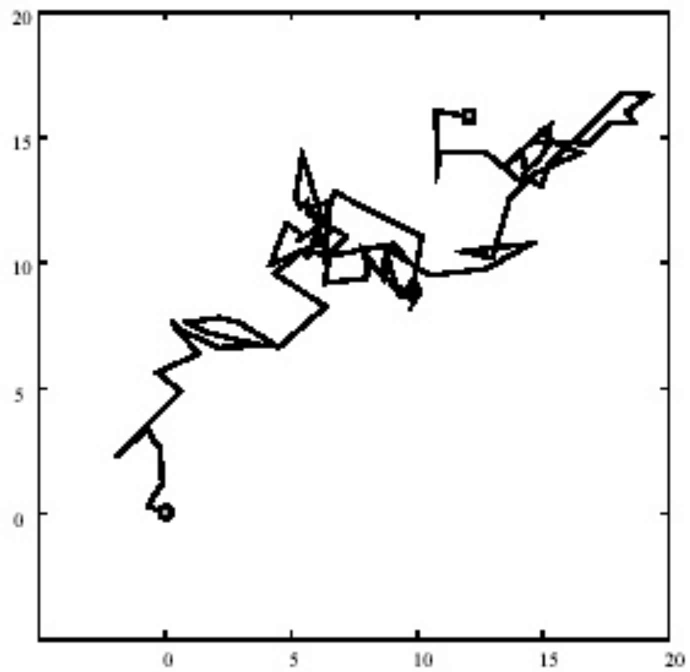
Dynamical model: auto-regressive (AR) process



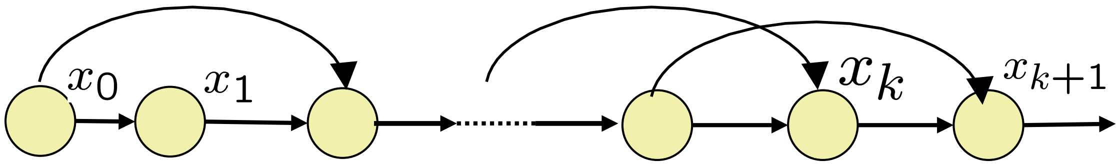
- E.g. assuming constant position
  - ⇒ Speed is noise

$$\dot{\mathbf{x}}_k = Bw_k \text{ with } w_k \sim \mathcal{N}(w_k|0, I)$$
$$\mathbf{x}_k = \mathbf{x}_{k-1} + Bw_k$$
$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k-1}, \Gamma = B^T B)$$

  - ⇒ Brownian motion
- Note
  - One can **simulate samples** from the process using **ancestral sampling**
  - AR model of order 1

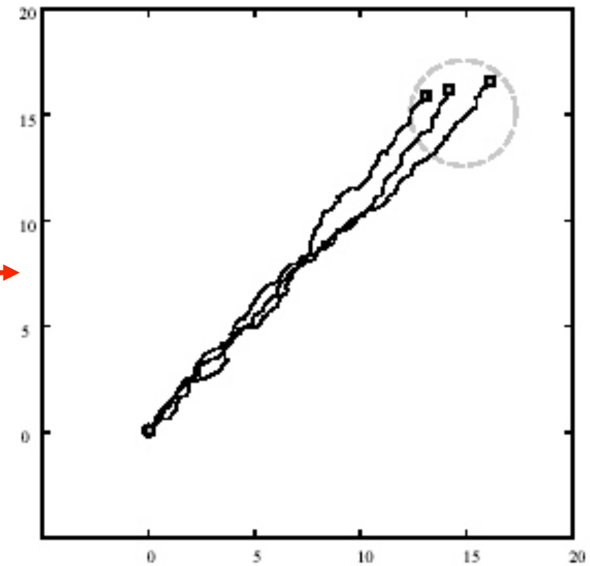


Dynamical model: auto-regressive (AR) process



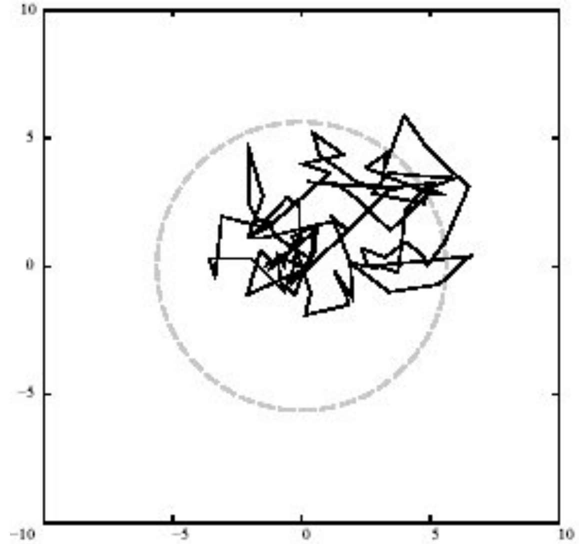
- More realistic: constant speed model
  => acceleration is noise

$$\ddot{\mathbf{x}} = Bw_k$$
$$\mathbf{x}_k = 2\mathbf{x}_{k-1} - \mathbf{x}_{k-2} + Bw_k$$
$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_{k-2}) = \mathcal{N}(\mathbf{x}_k|2\mathbf{x}_{k-1} - \mathbf{x}_{k-2}, \Gamma)$$



- Note:
  - State can grow without bound
    - maybe not adapted for other parameters (e.g. scale)

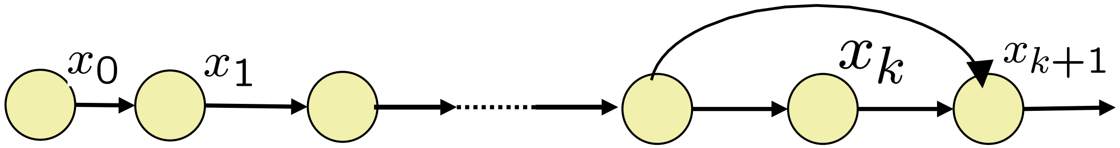
$$(\mathbf{x}_k - \bar{\mathbf{x}}) = a(\mathbf{x}_{k-1} - \bar{\mathbf{x}}) + Bw_k$$



    - steady-state value (e.g. 1 for scale)

  => Constrained Brownian motion

Dynamical model: 2nd order AR model



mass

$$m\ddot{\mathbf{x}} = -\nu\dot{\mathbf{x}} - k\mathbf{x} + f$$

external forces

Friction component

potential energy component

- Example: ballistic model of falling ball

  constant acceleration model (x=height)

$$\ddot{\mathbf{x}}_k = a + Bw_k$$

$$\mathbf{x}_k = 2\mathbf{x}_{k-1} - \mathbf{x}_{k-2} + a + Bw_k$$

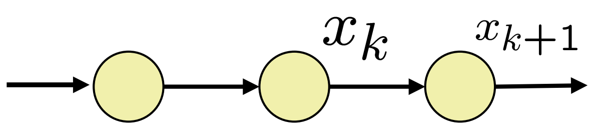
$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k|k-1}, \Gamma)$$

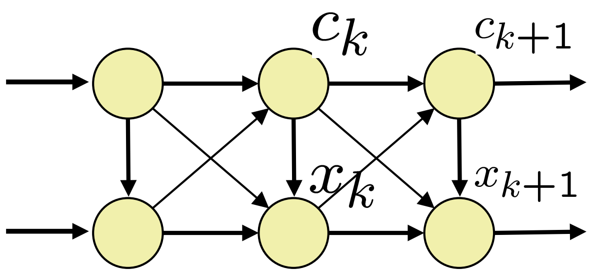
$$\mathbf{x}_{k|k-1} = 2\mathbf{x}_{k-1} - \mathbf{x}_{k-2} + a$$

- Can be set as order 1:

$$\begin{cases} h_k = h_{k-1} + v_{k-1} + a + w_t \\ v_k = h_k - h_{k-1} = v_{k-1} + a + w_t \end{cases} \qquad \mathbf{x}_k = [h_k \;\; v_k]^T$$

Switching dynamics model





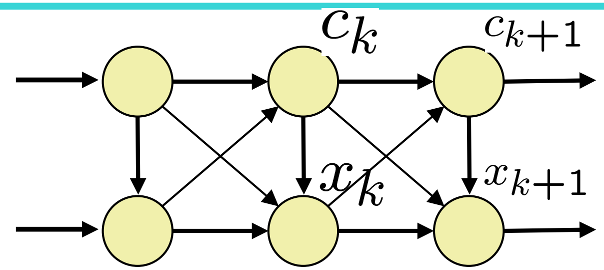
- AR model: modelling of one continuous activity
- However, in general
  - dynamics present discontinuities
  - sequence of different activities

  $\Rightarrow$ discrete variables to model these effects

- Mixed state approach   state = $(\mathbf{x}_k, c_k)$

Switching dynamics model

- Model



$$p(\mathbf{x}_k, c_k | \mathbf{x}_{k-1}, c_{k-1}) = \boxed{}$$

- two main distributions

$$p_{ij}(\mathbf{x}_k|\mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, c_k = j, c_{k-1} = i)$$

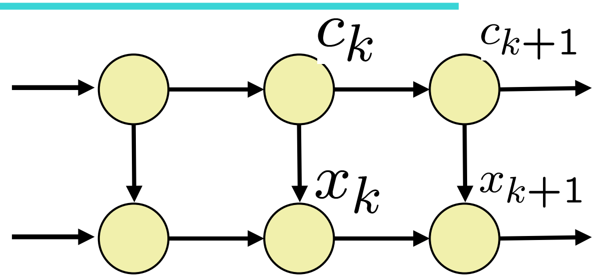
  - specific continuous dynamics
    - on transitions (i≠j)
    - for a given activity (i=j)

$$T_{ij}(\mathbf{x}_k) = p(c_k = j|\mathbf{x}_{k-1}, c_{k-1} = i)$$

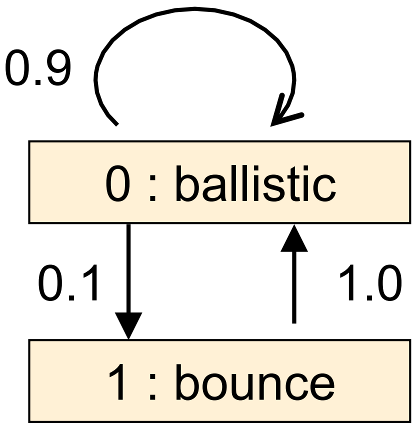
  - discrete transition matrices of activity
    - can depend on the state value

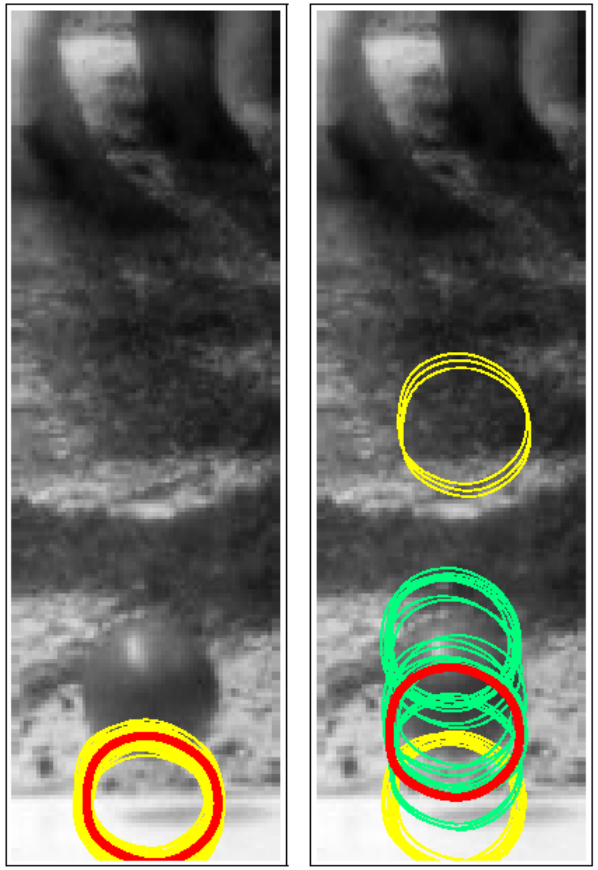
      e.g. activity changes occur on specific image regions

Switching dynamics: bouncing ball example



- Two distinctive activites c
  - 0 : ballistic (constant acceleration)
  - 1 : bouncing instant
- State transitions
  - bounce last one instant





  - dynamics

Switching dynamics: bouncing ball example

Constant acceleration model

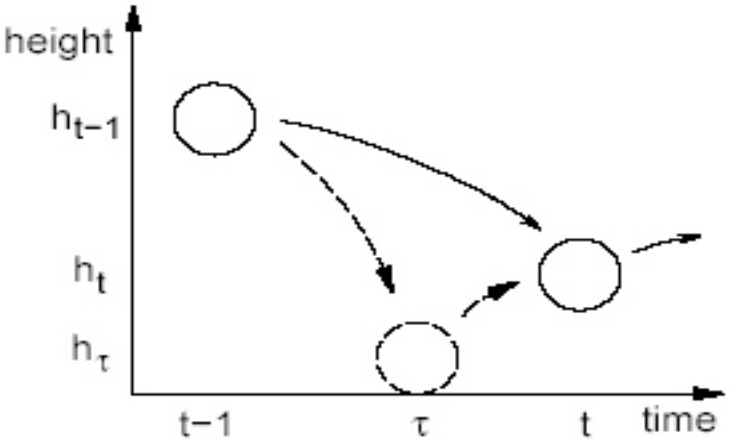
$$\begin{aligned} h_t &= h_{t-1} + v_{t-1} + a + \omega_t \\ v_t &= h_t - h_{t-1} \end{aligned}$$

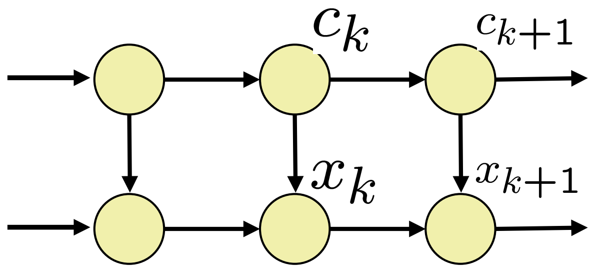
$$\omega_t \in N(0, \sigma_h)$$

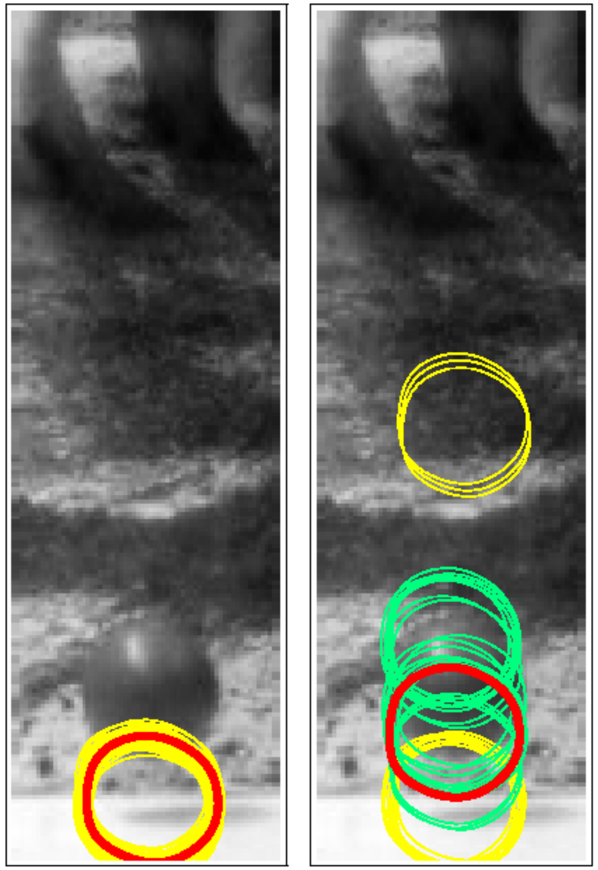
Bounce model

$$\begin{aligned} h_\tau &= h_{t-1} + \tau h_{t-1} \\ v_\tau &= v_{t-1} + \tau a \\ h_t &= h_\tau - e(1-\tau)v_\tau + \gamma_t \\ v_t &= -e v_\tau + (1-\tau)a + \nu_t \end{aligned}$$

$$\tau \in U[0,1), \ \nu_t \in N(0, \sigma_v), \ \gamma_t \in N(0, \sigma_B)$$







Dynamical models: conclusion

- Dynamical state models
  - AR representation
  - defined from physical principle
  - Learning can be done through Maximum Likelihood (AR models)
  - Switching models: indicators of different activities/situations

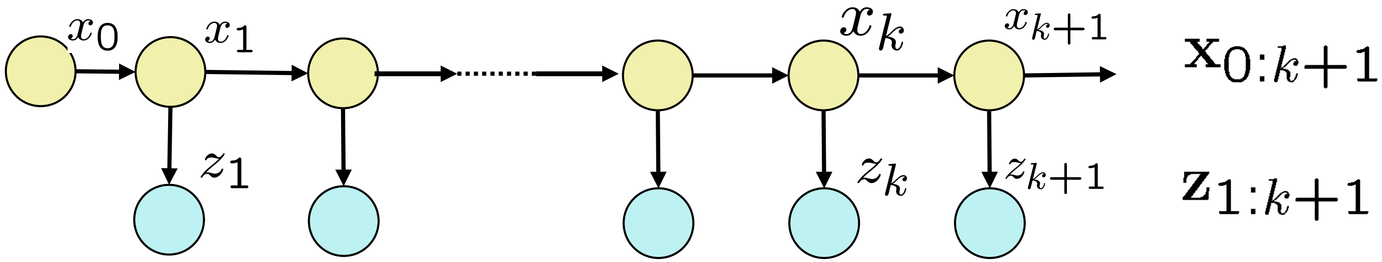
- Issues:
  - availability of training data
  - exploitation in tracking
    - not always easy: test data has to matched well the training data
    - often, parameters set by hands
    - unpredictable motions => simpler models are better

Outline

- Bayesian tracking

  - Sequential estimation
    - Kalman filter
    - Particle filter

Sequential estimation



- First (simplified) approach
  - Succession of instantaneous estimation problems

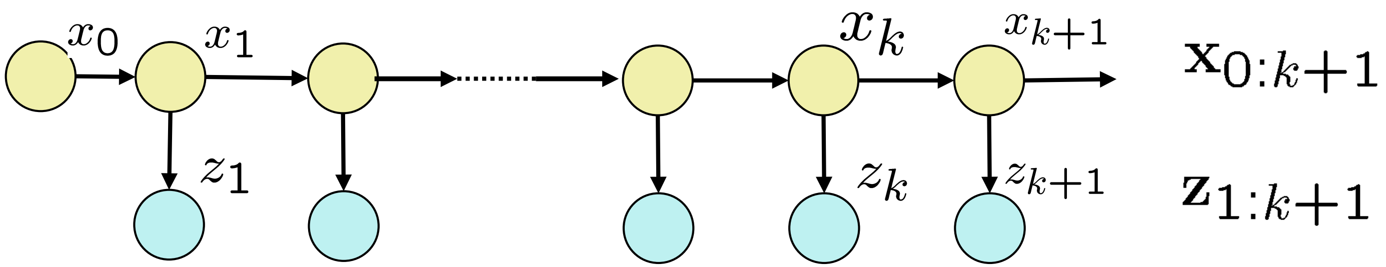
    => finding the best estimate at each time step

$$
\begin{aligned}
\hat{\mathbf{x}}_k &= \arg\max p(\mathbf{x}_k|\mathbf{z}_k, \hat{\mathbf{x}}_{k-1}) \\
&= \arg\max p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1})
\end{aligned}
$$

  - used especially for complex state-spaces (e.g. free form shapes)
  - efficient, but sensitive to temporarily tracking loss (e.g. during occlusion)

  => Bayesian filtering

Bayesian tracking



- Goal: recursive estimation of state probability given the sequence of observations
  - posterior distribution : $p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k})$
  - filtering distribution : $p(\mathbf{x}_k|\mathbf{z}_{1:k})$

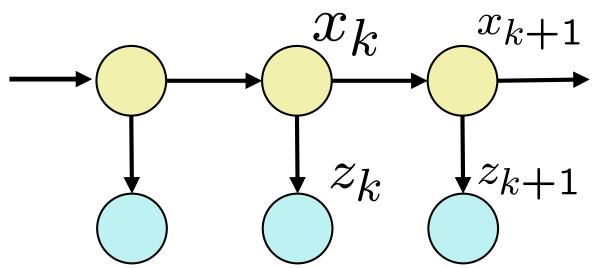
- Allows to compute quantities of interest
  - e.g. mean (expected value) of state at time k

$$\bar{\mathbf{x}}_k = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k$$

- Recursive estimation of filtering distribution using Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_i) \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}$$

Chapman-Kolmogorov equation



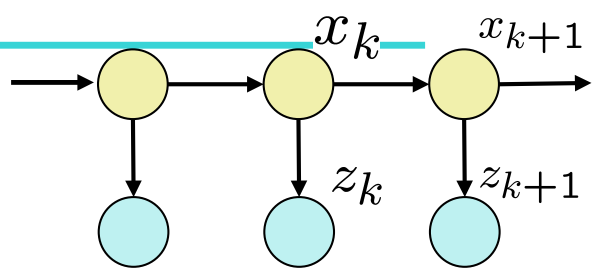
$$
\begin{aligned}
p(\mathbf{x}_k|\mathbf{z}_{1:k}) &= p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{z}_{1:k-1}) \\
&= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\
&= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}
\end{aligned}
$$

- Note: comparison with general formula

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p_0(\mathbf{x})}{p(\mathbf{z})}$$

- $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ plays the role of the prior on the current state learned from the previous observations

Recursive Bayesian filtering



- At each time step, two steps
  - **prediction step**

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

  - **update step:** new observation available
    - apply Chapman-Kolmogorov equation

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

    - predicted likelihood

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \boxed{}$$

- At each time step:
  - two integrals (or summation, given the nature of the state space)

Recursive Bayesian filtering

- Discret state case: integrals => summations

  cf HMM: forward pass of Baum-Welsh algorithm

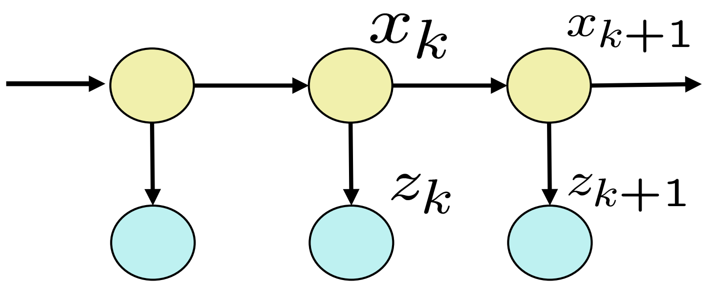
- Continous state case
  - Linear and Gaussian case: analytical integration tractable

    => Kalman filter
  - Monomodal distributions: gaussian approximation

    => Extended Kalman filter (EKF), or «unscented» (UKF)
  - Discretized state (Grid based filters): cf HMM approach
  - Muti-modal general case : normalizations unfeasible) *Monte Carlo approximations*

Kalman filter

- Fundation: R.E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, 1960



- Assumptions

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + w_k \qquad p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k|A\mathbf{x}_{k-1}, Q)$$

Independent process and measurement noises

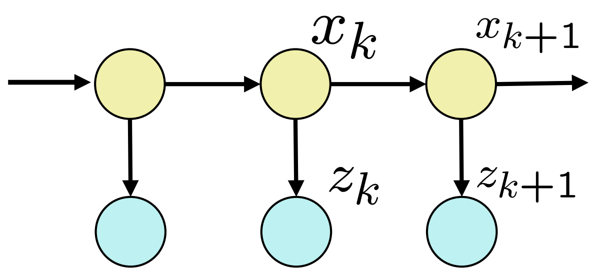
$$\mathbf{z}_k = H\mathbf{x}_k + v_k \qquad p(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k|H\mathbf{x}_k, R)$$

Kalman filter

- Result: direct graph, linear and Gaussian

  => joint Gaussian distribution over all variables

  => all marginals are Gaussian



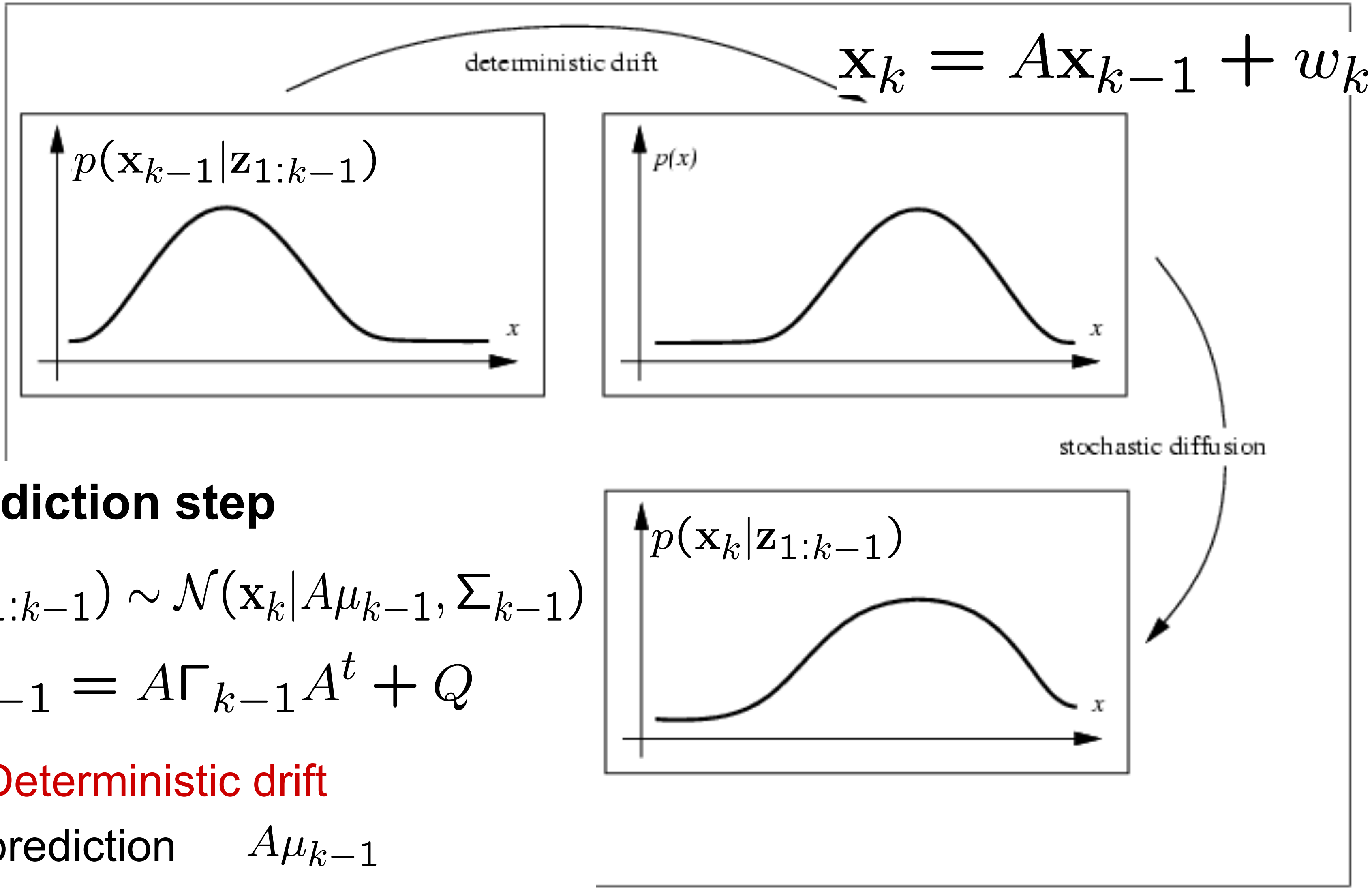
- In particular, the filtering distributions

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \sim \mathcal{N}(\mathbf{x}_{k-1}|\mu_{k-1}, \Gamma_{k-1})$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \sim \mathcal{N}(\mathbf{x}_k|\mu_k, \Gamma_k)$$

- Predictive and update steps can be solved using properties of Gaussian processes

Kalman filter



- **Prediction step**

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \sim \mathcal{N}(\mathbf{x}_k|A\mu_{k-1}, \Sigma_{k-1})$$

$$\Sigma_{k-1} = A\Gamma_{k-1}A^t + Q$$

  - Deterministic drift
    prediction $A\mu_{k-1}$
  - Stochastic diffusion: variance increase due to process noise

Kalman filter

Kalman gain

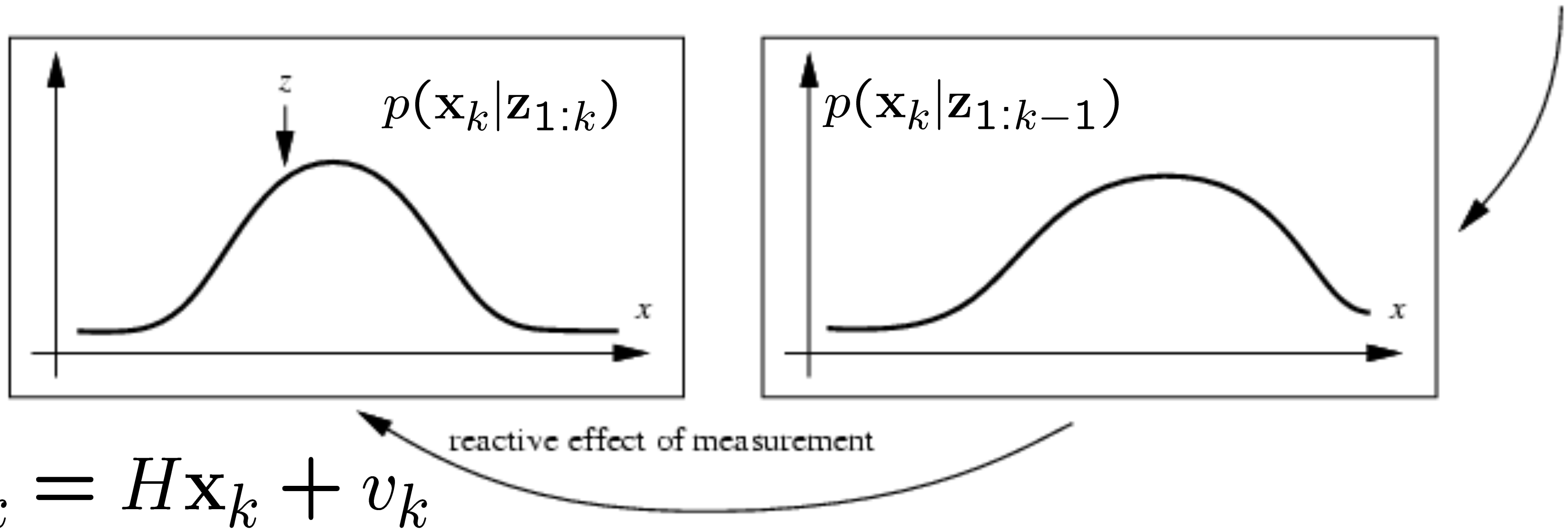
Innovation: difference between measure and prediction

- **Update step:**

$$\mu_k = A\mu_{k-1} + K_k(\mathbf{z}_k - HA\mu_{k-1})$$

$$\Gamma_k = (I - K_k H)\Sigma_{k-1} \qquad K_k = \Sigma_{k-1}H^t(H\Sigma_{k-1}H^t + R)^{-1}$$

- Reactive effect of measurement
  - Move prediction towards observation, depending on relative uncertainties (prediction vs observation), cf Kalman gain
  - Reduces the variance of the predicted estimation



$$\mathbf{z}_k = H\mathbf{x}_k + v_k$$

Kalman filter

Kalman gain

Innovation: difference between measure and prediction

- **Update step:**

$$\mu_k = A\mu_{k-1} + K_k(\underline{\mathbf{z}_k - HA\mu_{k-1}})$$

$$\Gamma_k = (I - K_k H)\Sigma_{k-1} \qquad K_k = \Sigma_{k-1}H^t(H\Sigma_{k-1}H^t + R)^{-1}$$

- Qualitatively
  - measurement noise R large w.r.t. process noise

    => Kalman gain close to 0

    => posterior mean/variances near predicted mean/variance

    => posterior mean close to average of measurements up to current instant

  - measurement is very precise

    (measurement noise small w.r.t. process noise)

    => Kalman gain close to H^{-1} (pseudo-inverse of H)

    => posterior mean close to the measurement

Kalman filter for visual tracking

- Applied as soon as the 80s, esp. for feature tracking (points, edges)

- Classical approaches
  - Use prediction to initialize optimization process at time k. Use result of optimization as measurement.

    e.g. Mean-Shift algorithm

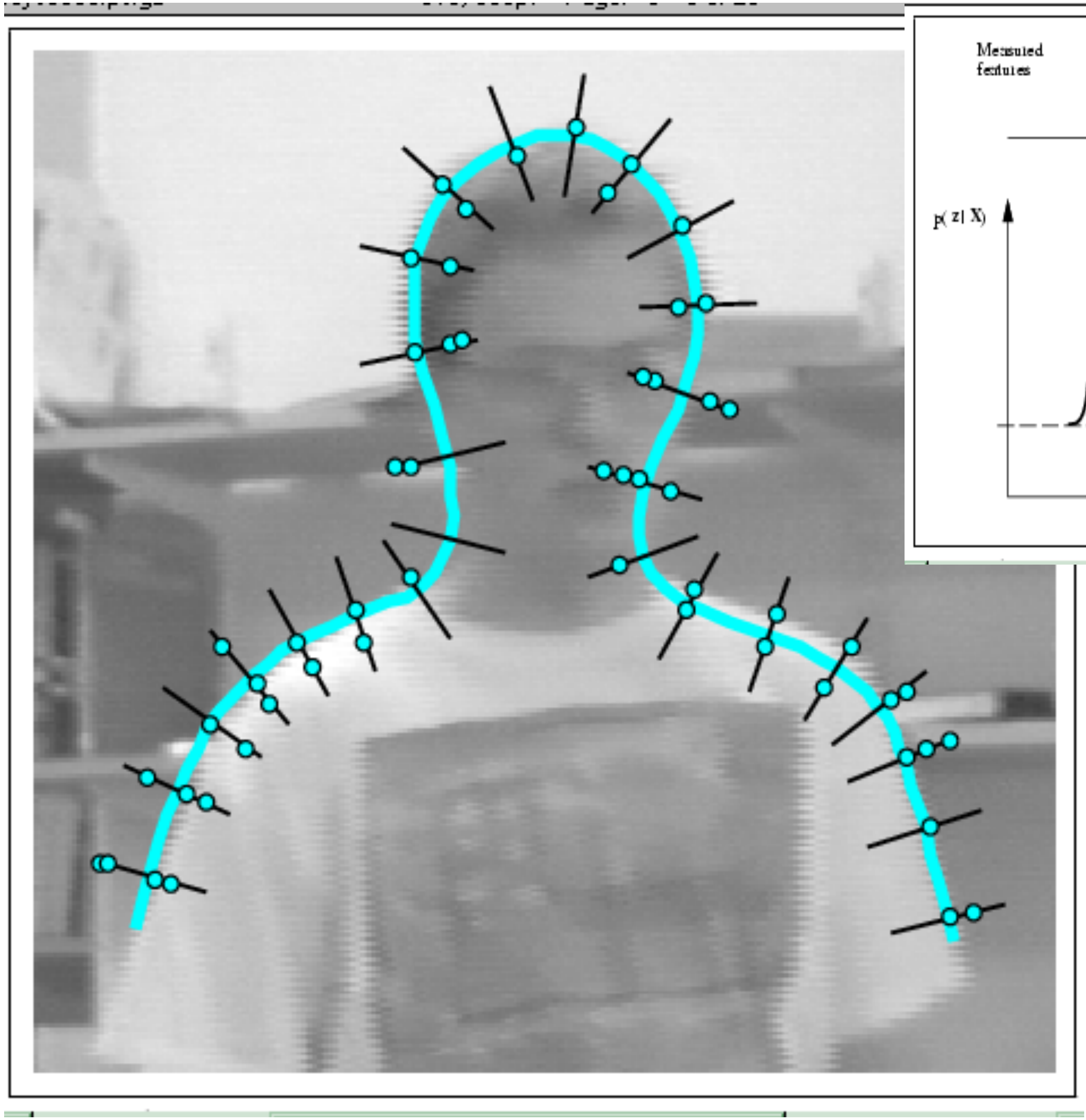
  - Local detector provides measurement location; measurement selected as the closest one to prediction (gating process)

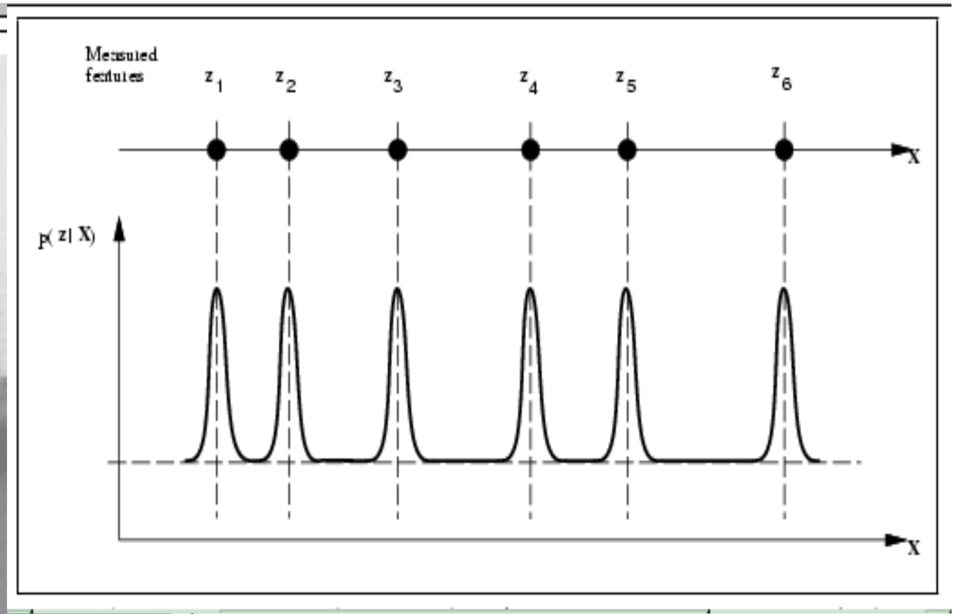
    e.g. point tracking

Kalman filter: example [Remagnino et al., 1997]

- Blob detection using foreground/background segmentation
- Blob extraction and matched with nearest entity (person/car) => measurement for Kalman filter

Visual clutter => observational non-linearities





[A: Blake]

Kalman filter issue

- **Issue:** in principle, extraction of **z** should be independent of previous measurements/states

=> often not the case in vision:

- gradient optimization starting from the prediction
- measurements selected near predictions

- In addition:
  - measurements depends on hypothesized state

  e.g. shape model shown

Kalman filter issue

- **Issue:** measurements need to be of the same nature as (part of) the state
  - $z = h(x) +$ noise
  - common cases: tracking of points and lines

    => **x** is a location/scale (+derivatives)

    => **z** has to be geometric parameters as well

  - **what if y is a template** ? a color histogram ? the image ?

    the likelihood models for shapes that we have seen ?

    => the definition of h becomes very tricky

Kalman filter: summary

- Advantages:
  - Exact computation
  - Optimal under hypothesis
  - Provides a mechanism to account for uncertainty in observation extraction
  - Parameters of model (A, H, Q, R) can be learned from training data
- Drawbacks
  - Strong limitations on observation model
    - Measurements need to be of the same nature as (part of) the state
    - Measurement of interest must be uniquely identified => data association issue
    - Clutter is frequent => posterior are not mono-modal

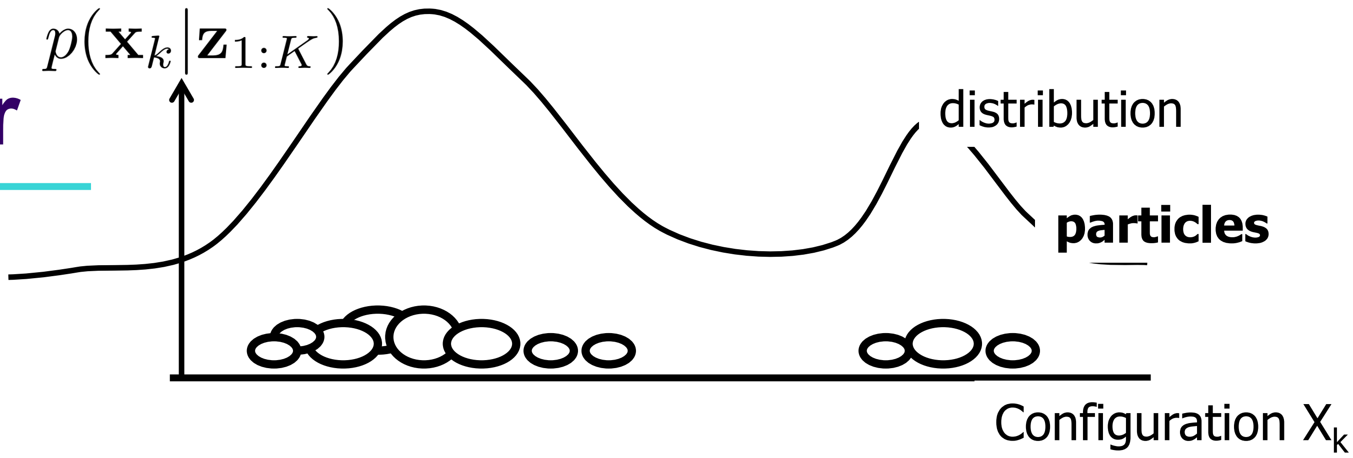
Bayesian filtering : “Particle filtering”

- Monte Carlo approximations
  - non-parametric representation of distribution through samples
  - different names: particle filter (PF), sequential Monte-Carlo (SMC), Sequential Importance sampling (SIS)

- Fundations
  - Gordon 1993, « *Novel approach to non-linear/non-Gaussian Bayesian state estimation* »
  - Isard et Blake 1996 « *CONDENSATION: CONditional DENSity propagATION for visual tracking* »

- Interest for visual tracking
  - Multiple hypothesis maintained => increased robustness to clutter, occlusions, short tracking failures
  - No restriction on model ingredients
  - Easy implementation

Particle filter



- **Intuition:**

  approximate at each time step the posterior distribution (the filtering distribution) of states using a set of M weighted samples

$$\{(\mathbf{x}_k^{(m)}, \pi_k^{(m)})\}_{m=1 \ldots M} \qquad \sum_{m=1}^{M} \pi_k^{(m)} = 1 \qquad p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{m=1}^{M} \pi_k^{(m)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(m)})$$

dirac distribution

- **Usage:** compute expectation of function f

$$\int f(\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k \approx \pi_k^{(m)} f(\mathbf{x}_k^{(m)})$$

  - In particular, mean expectation of state (f(x)=x)

$$\int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k \approx \pi_k^{(m)} \mathbf{x}_k^{(m)}$$

- How do we get these samples ?

Perfect sampling

- Target distribution $p(\mathbf{x})$
- Draw M samples $\mathbf{x}^{(m)} \sim p(\mathbf{x}),\ m = 1 \cdots M$
- Approximation $p(\mathbf{x}) \approx \sum_{m=1}^{M} \boxed{\frac{1}{M}} \delta(\mathbf{x} - \mathbf{x}^{(m)})$

weight of sample

- Expectation w.r.t. p

$$\mathbb{E}_p[f] = \int f(\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x} \longrightarrow I_M(f) = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{x}^{(m)})$$

- Approximation: unbiased, converges when M goes to infinity
- Usually: difficult to sample from p directly !

Importance sampling

- Use a ‘proposal’ auxiliary function **q**
  - q : as close as possible to p (and supp(p) included in supp(q)) (i.e. $q(\mathbf{x}) = 0 \Rightarrow p(\mathbf{x}) = 0$)
  - Draw the samples from q instead of p

$$\mathbf{x}^{(m)} \sim q(\mathbf{x}), \; m = 1 \cdots M$$

$$\Rightarrow \mathbb{E}_p[f] = \int f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) \mathrm{d}\mathbf{x} \approx \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{x}^{(m)}) \frac{p(\mathbf{x}^{(m)})}{q(\mathbf{x}^{(m)})}$$

- Importance weights

$$\boxed{\pi^{(m)} \propto \frac{p(\mathbf{x}^{(m)})}{q(\mathbf{x}^{(m)})} \qquad \sum_{m=1}^{M} \pi^{(m)} = 1}$$

$\Rightarrow$ correction factor: samples were drawn from q instead of p

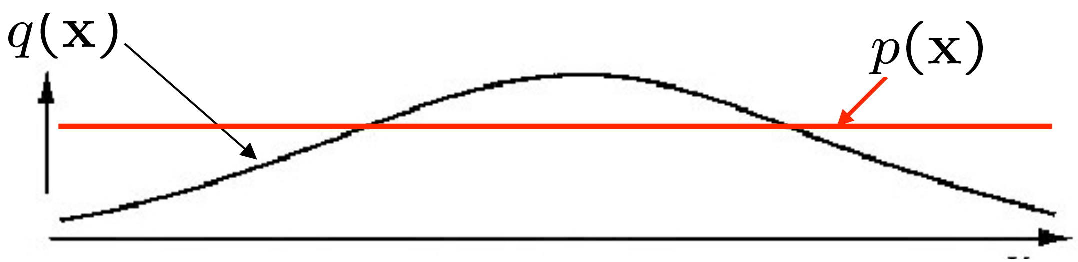
Importance sampling

- Importance weights

$$\boxed{\pi^{(m)} \propto \frac{p(\mathbf{x}^{(m)})}{q(\mathbf{x}^{(m)})} \qquad \sum_{m=1}^{M} \pi^{(m)} = 1}$$

⇒ large weight if q is smaller than p

⇒ larger weights where q will simulate less samples than p would



samples generated from q(x), and reweighted

- Approximation of p

$$p(\mathbf{x}) \approx \sum_{m=1}^{M} \pi^{(m)} \delta(\mathbf{x} - \mathbf{x}^{(m)})$$

Sequential Importance Sampling (SIS)
First example: Bootstrap filter

- Importance sampling: target distribution

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

- Proposal function: predictive distribution

$$\mathbf{x}_k^{(m)} \sim q(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$$

- Importance weight: Chapman-Kolmogorov equation

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

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

$$\Rightarrow \pi_k^{(m)} = \frac{p(\mathbf{x}_k^{(m)}|\mathbf{z}_{1:k})}{q(\mathbf{x}_k^{(m)})} \propto p(\mathbf{z}_k|\mathbf{x}_k^{(m)})$$

SIS: Bootstrap filter

- **How to simulate from q, the predictive distribution** ?

  assume that we have sample set from previous instant

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \approx \sum_{m=1}^{M} \pi_{k-1}^{(m)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(m)})$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} = \sum_{m=1}^{M} \pi_{k-1}^{(m)} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$$

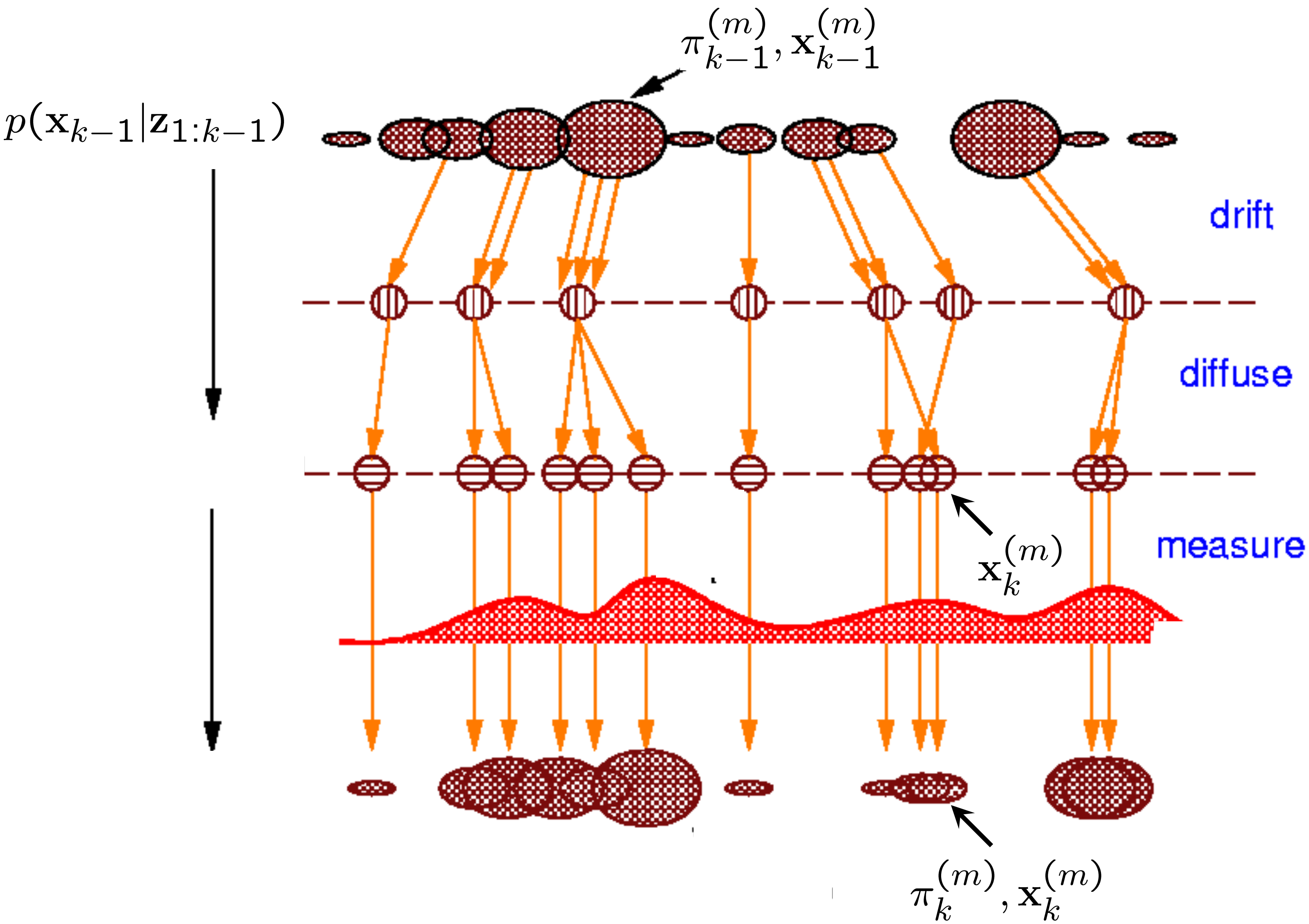
  - ⇒ mixture of distributions
  - ⇒ assumption: Gaussian dynamics $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k|A\mathbf{x}_{k-1}, Q)$
  - ⇒ **Mixture of Gaussians**

- Sampling
  - Sample the mixture weight $a_m \sim Multinomial(\pi_{k-1}^{(m)}, m = 1 \cdots M)$
  - Simulate noise from the Gaussian and apply the dynamical model to the selected sample

$$w^{(m)} \sim \mathcal{N}(w|0, Q)$$

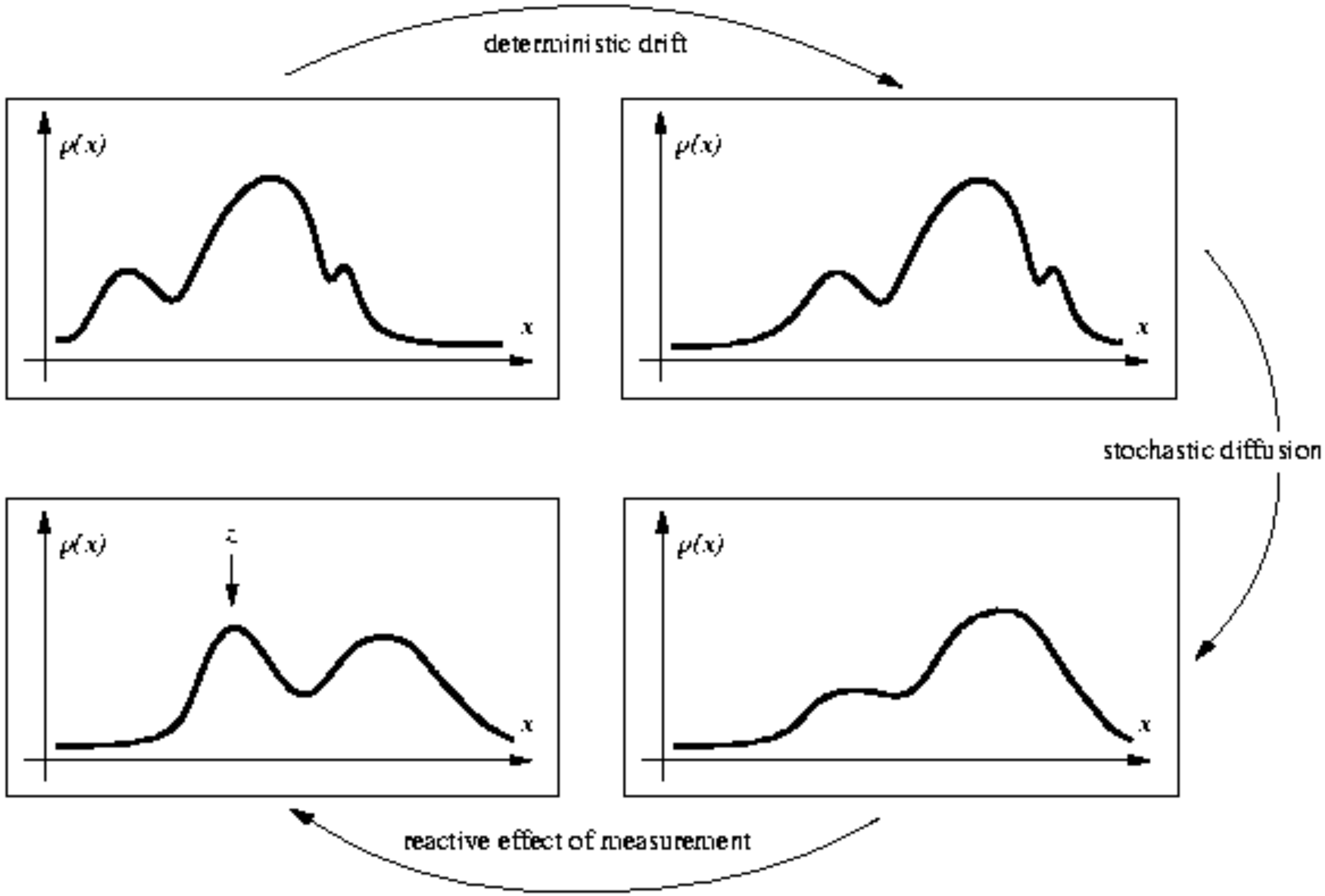
$$\mathbf{x}_k^{(m)} = A\mathbf{x}_{k-1}^{(a_m)} + w^{(m)}$$

SIS: Bootstrap filter



[A. Blake, 1998]

Non-Gaussian Bayesian Filter



SIS: general case

- Apply Importance sampling to posterior distribution

  => Target

$$p(\underline{\mathbf{x}_{1:k}}|\mathbf{z}_{1:k}) \propto p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{x}_i|\mathbf{x}_{i-1})p(\mathbf{z}_i|\mathbf{x}_i)$$

- Note: here, we want to simulate/sample **trajectories**

  => this will be done recursively (by extending trajectories over time)

  => the **importance weights** will be computed **recursively**

$$\underbrace{p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k})}_{\text{time k}} \propto \underbrace{p(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1})}_{\text{time k-1}}p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{z}_k|\mathbf{x}_k)$$

SIS: general case

- Target $p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}) \propto p(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{z}_k|\mathbf{x}_k)$

- Proposal function, factorized

$$q(\mathbf{x}_{1:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_0)\prod_{i=1}^{k} q(\mathbf{x}_i|\mathbf{x}_{i-1},\mathbf{z}_i) = q(\mathbf{x}_{1:k-1}|\mathbf{z}_{1:k-1})q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k)$$

- Importance sampling and weight: **recursion**
  - given weighted trajectories up to time k-1 $\{(\mathbf{x}_{0:k-1}^{(m)}, \pi_{k-1}^{(m)})\}$
  - extend trajectory with proposal

$$\mathbf{x}_i^{(m)} \sim q(\mathbf{x}_i|\mathbf{x}_{i-1}^{(m)},\mathbf{z}_i),\ m = 1 \cdots M$$

  - weight update

$$\pi_k^{(m)} \propto \pi_{k-1}^{(m)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})}{q(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)},\mathbf{z}_k)} \text{ with } \sum_{m=1}^{M} \pi_k^{(m)} = 1$$

Proposal densities

- What is the interest of this general approach ?

  introduction of an **explicit proposal density** we can play with
- Examples
  - Bootstrap filter (**proposal=dynamic**) first proposed, popular, simple

$$q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$$
$$\Rightarrow \pi_k^{(m)} \propto \pi_{k-1}^{(m)}\, p(\mathbf{z}_k|\mathbf{x}_k^{(m)}) \text{ with } \sum_{m=1}^{M} \pi_k^{(m)} = 1$$

=> same weights as before (assuming past weights at k-1 are equiprobable)
(notice however the difference in the way it was obtained)

  - Optimal proposal: takes into account **previous state** and **current observation** $q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k)$

Right hand term can not be computed in general for a given model

  - In between: use current data for a better efficiency

Resampling

- trajectory generation process independent of weight values
  - good (and esp. bad) trajectories are equally propagated
  - after some time steps, most of the weight is located in a few samples
  - ⇒ **many samples don't really contribute to the distribution approximation**
  - ⇒ **distribution degeneracy**

- solution: **resampling**
  - sample selection
    - **elimination** of sample with smaller weights
    - **duplication** of samples with larger weights
    - ⇒ keep the representation valid w.r.t. convergence properties

- one approach: sample with replacement

$$\{\mathbf{x}_k^{(a_m)}, \frac{1}{M}, m = 1 \ldots M)\} \Leftarrow \{\mathbf{x}_k^{(m)}, \pi_k^{(m)}, m = 1 \ldots M)\}$$

$$a_m \sim Multinomial(\pi_k^{(m)}, m = 1 \cdots M)$$

An algorithm

Given the particle distribution at the previous time step

$$\{(\mathbf{x}_{0:k-1}^{(m)}, \pi_{k-1}^{(m)})\}_{m=1\cdots M}$$

- sample from proposal

$$\tilde{\mathbf{x}}_k^{(m)} \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{z}_k),\ m = 1\cdots M$$

- weight update

$$\tilde{\pi}_k^{(m)} \propto \pi_{k-1}^{(m)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})}{q(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{z}_k)} \text{ with } \sum_{m=1}^{M} \tilde{\pi}_k^{(m)} = 1$$

- resampling

$$\forall m,\ a_m \sim Multi(\tilde{\pi}_k^{(m)}), \mathbf{x}_{1:k}^{(m)} = (\mathbf{x}_{1:k-1}^{(a_m)}, \tilde{\mathbf{x}}_k^{(a_m)}) \text{ and } \pi_k^{(m)} = \frac{1}{M}$$

- Monte Carlo approximation

$$\mathbb{E}[f(\mathbf{x}_k)|\mathbf{z}_{1:k}] \approx \sum_{m=1}^{M} \pi_k^{(m)} f(\mathbf{x}_k^{(m)})$$

  - in particular, we can compute the mean value

CONDENSATION [Isard and Blake 96]

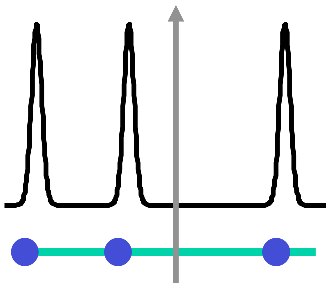
Object model: parametric contour on clutter

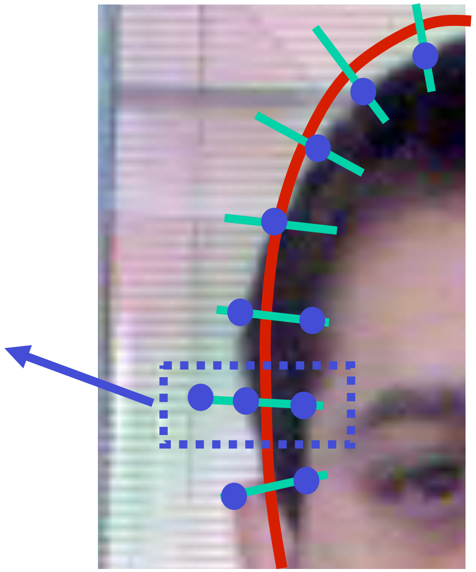
- **State:** affine parameters (mainly translation, scale in x and y, rotation)
- **Dynamics:** AR-2 model (on individual parameters)
- **Observations**: contours on lines perpendicular to shape model

$$z_t^l = \{v_m^l\}$$

- **Likelihood:** statistical hypothesis : independance of measures
- **Proposal:** dynamics => boostrap filter

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





$$p_{obj}(z_k \mid x_k) \propto \prod_{l=1}^{L} p(z_k^l \mid x_k)$$

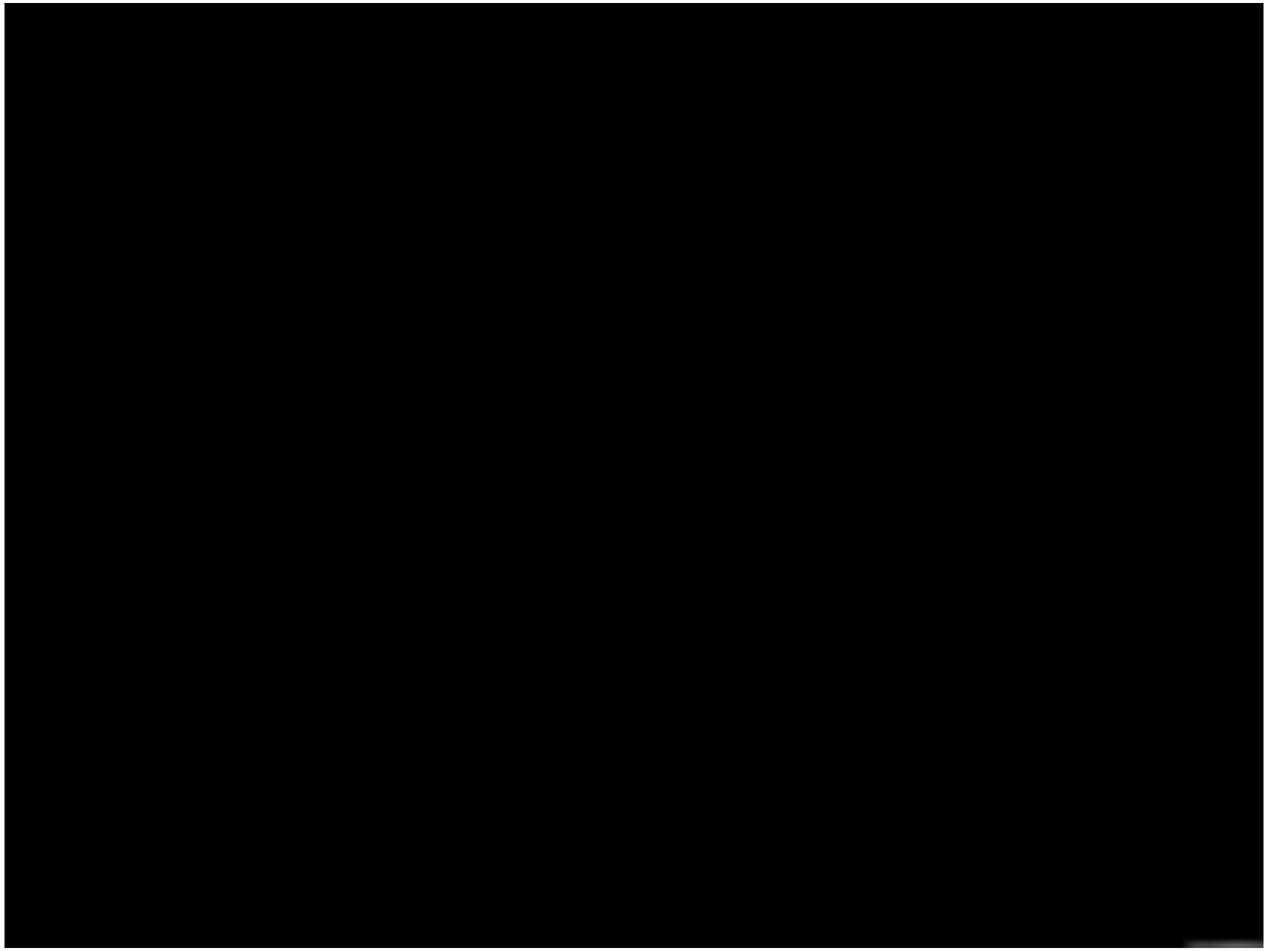
$$\propto \prod_{l=1}^{L} \max(K, \exp(-\frac{\|\hat{v}_m^l - v_0^l\|^2}{2\sigma^2}))$$

Nearest detection

Position on the shape model

CONDENSATION: Examples [Isard and Blake 98]

CONDENSATION: Examples [Isard and Blake 98]



sequence on white background: no clutter

⇒ allows to gather training data for learning dynamics

(without learned dynamics, model usually fails on clutter)

CONDENSATION: Example for head tracking

© Kodak

- **Red curve:** mean state
- **Yellow ellipses**: particles with larger weight (weight> 0.7 max weight)

Color Tracking [Perez et al, eccv 2002]

Object model: box with color measured in multiple regions

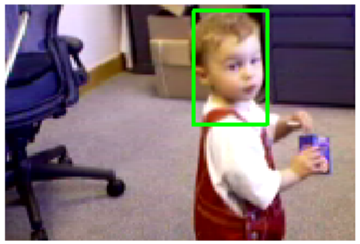
- **State:** translation, scale in x and y
- **Dynamics:** AR-2 model (manual parameter setting)
- **Proposal:** dynamics => boostrap filter

- **Observations**: multi-dimensional histogram (color histograms gathered in different regions of the objects => allows better localization)
- **Likelihood:** based on color similarity with a reference model

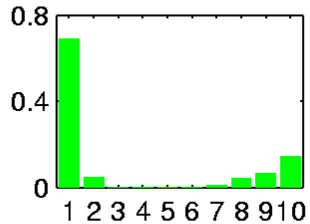
  => cf mean-shift tracker

$$p(\mathbf{z}_i | \mathbf{x}_i) \propto \exp -\lambda D^2[\mathbf{q}^\star, \mathbf{q}_i(\mathbf{x}_i)]$$

$$D[\mathbf{q}^\star, \mathbf{q}_i(\mathbf{x})] = \left[1 - \sum_{b=1}^{B} \sqrt{q^\star(b) q_i(b; \mathbf{x})}\right]^{\frac{1}{2}}$$

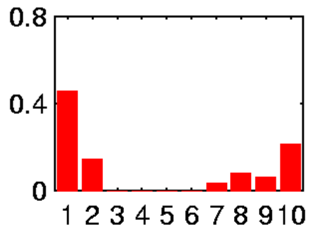


instant 0



référence

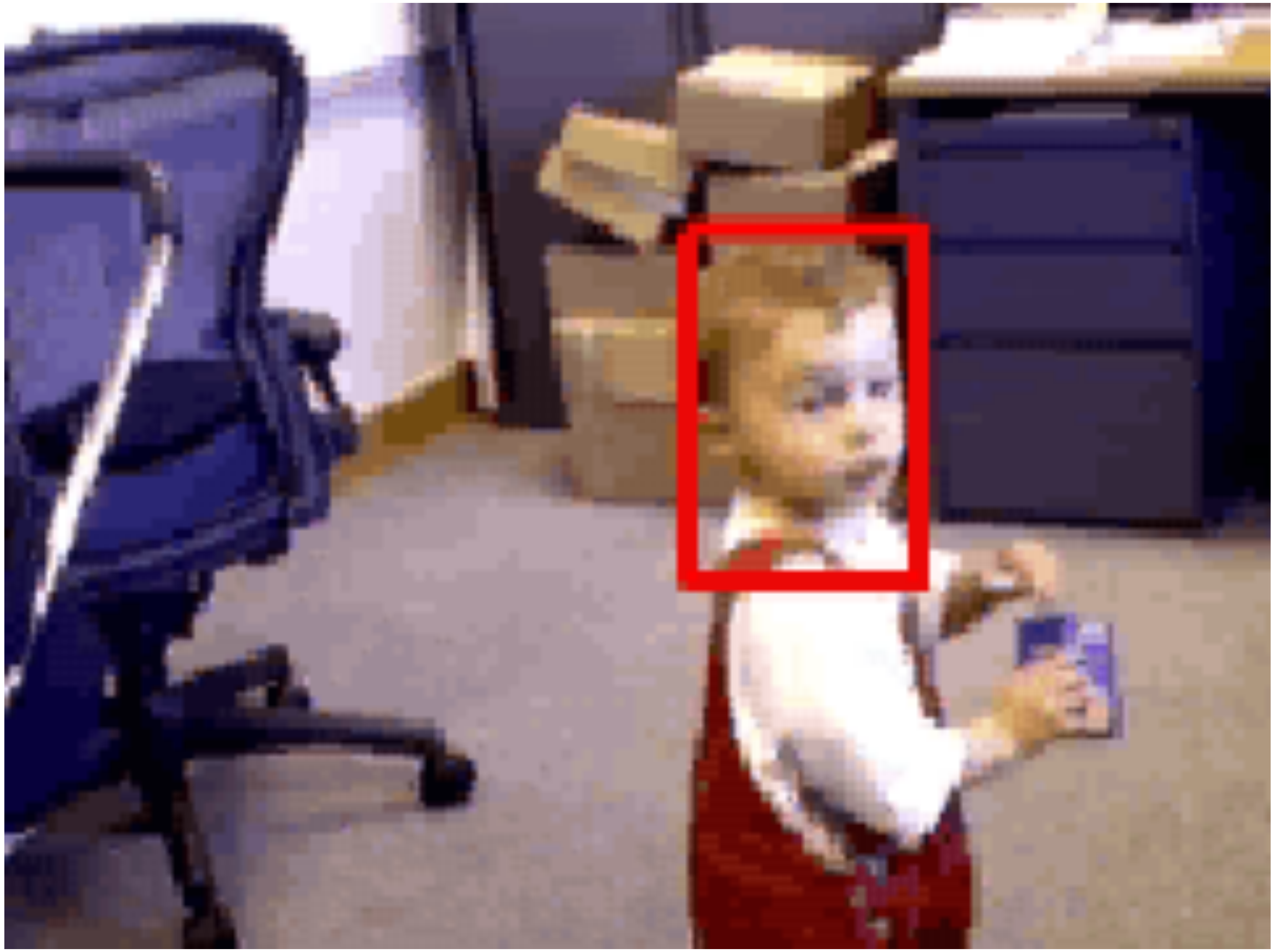
instant i



candidat

Color Tracking: examples [Perez et al, eccv 2002]

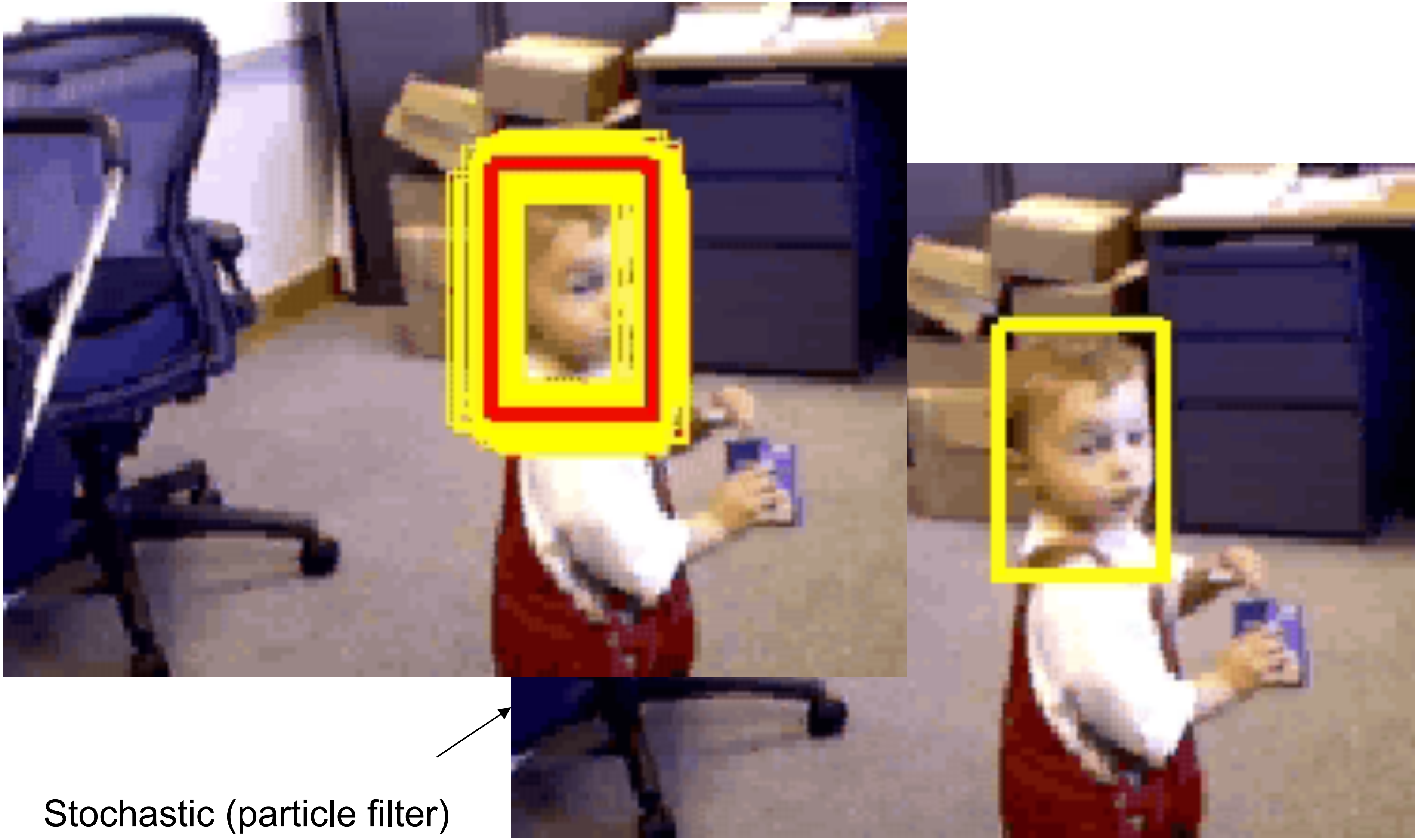
tracker exhibits robustness to color clutter



Deterministic (mean shift)

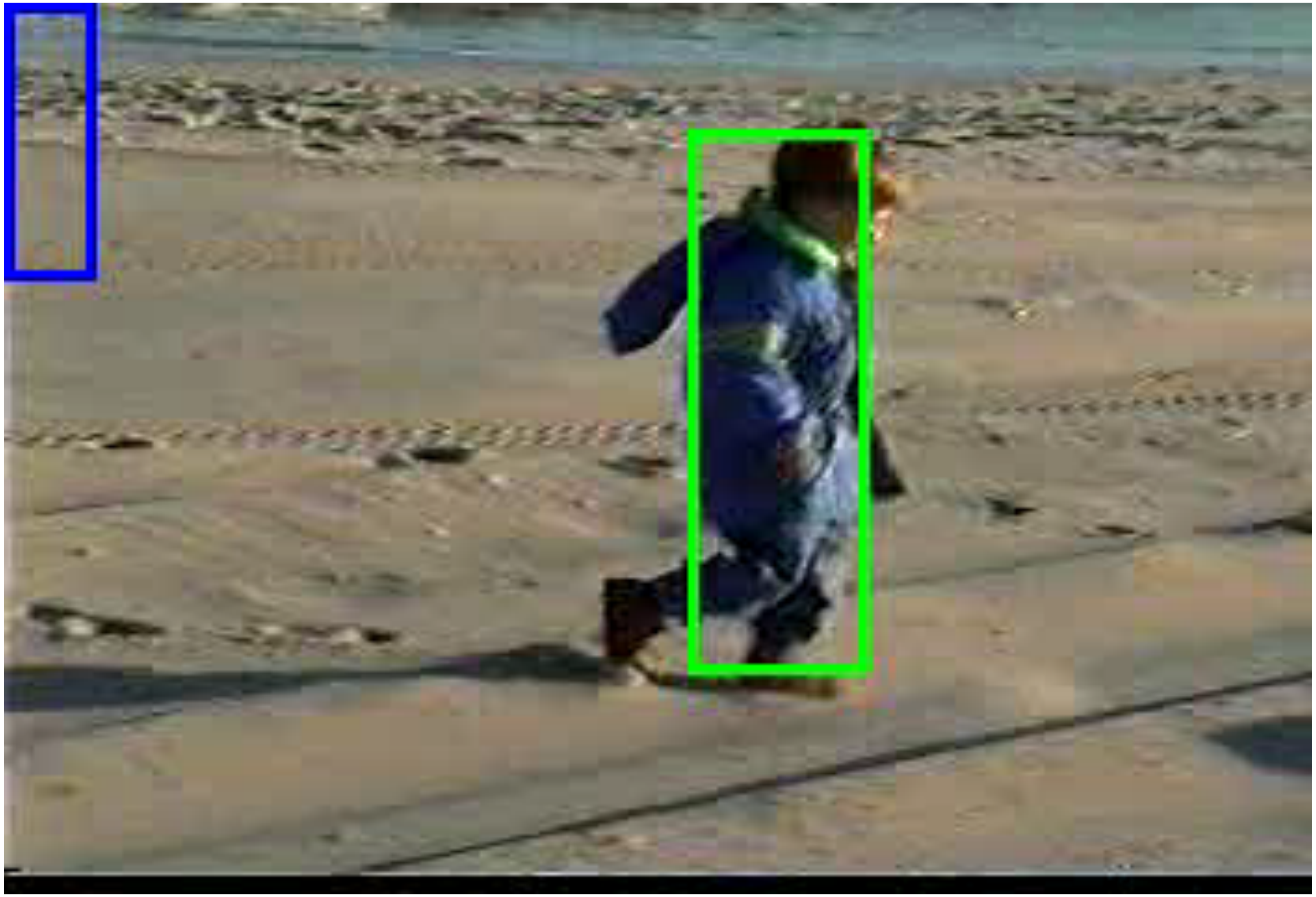
Color Tracking: examples [Perez et al, eccv 2002]

tracker exhibits robustness to color clutter



Stochastic (particle filter)

Color Tracking: examples [Perez et al, eccv 2002]



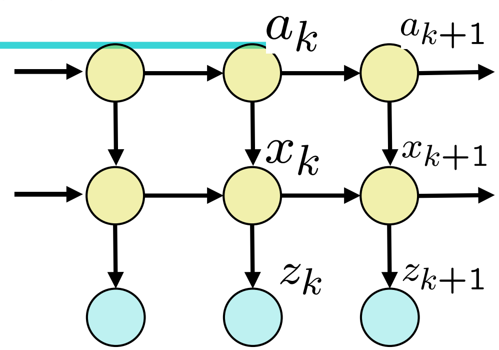
© Kodak

(tracker exhibits robustness to change of scale, object orientation, motion, illumination changes etc

Color Tracking: examples [Perez et al, eccv 2002]

tracker exhibits robustness to occlusions

Tracking with switching dynamics



Object model: shape

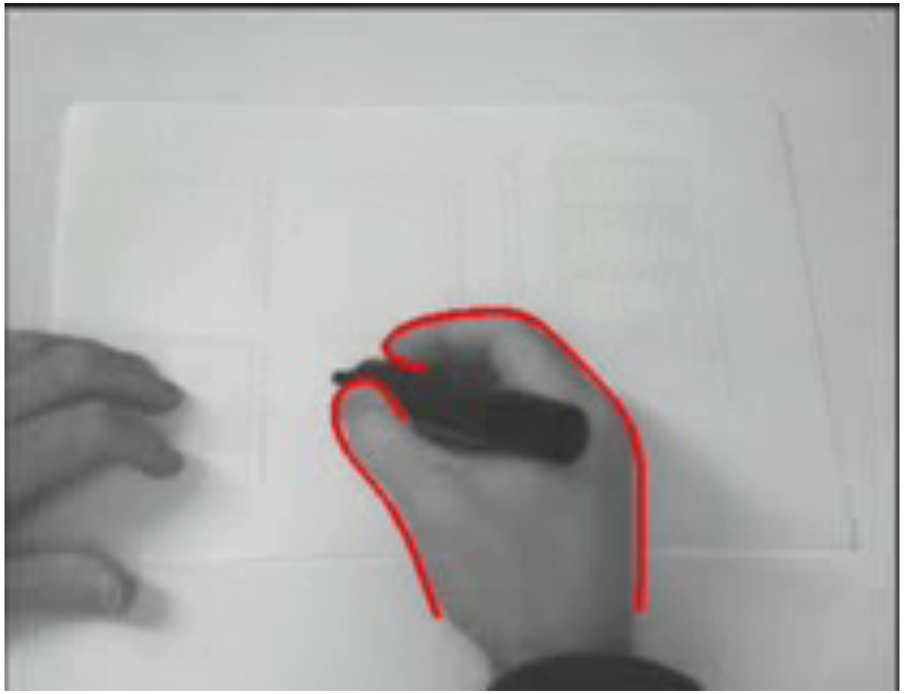
- **State:** translation and scale+ activity index
- **Dynamics:**
  - Markov transition (activity indices) => probability of changing activity

$$T_{ij} = p(a_k = j | a_{k-1} = i)$$

  - AR-2 models on location (depending on activity variable)

$$p_i(\mathbf{x}_k|\mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, a_k = i)$$

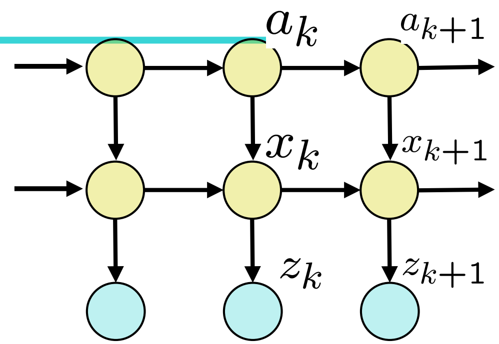
- **Proposal:** dynamics => boostrap filter
  - first sample activity
  - then sample location depending on activity



Activities: **Red** line drawing, **Blue**: pause, **Green**: scribbling

[Isard Blake, 2001]

Tracking with switching dynamics

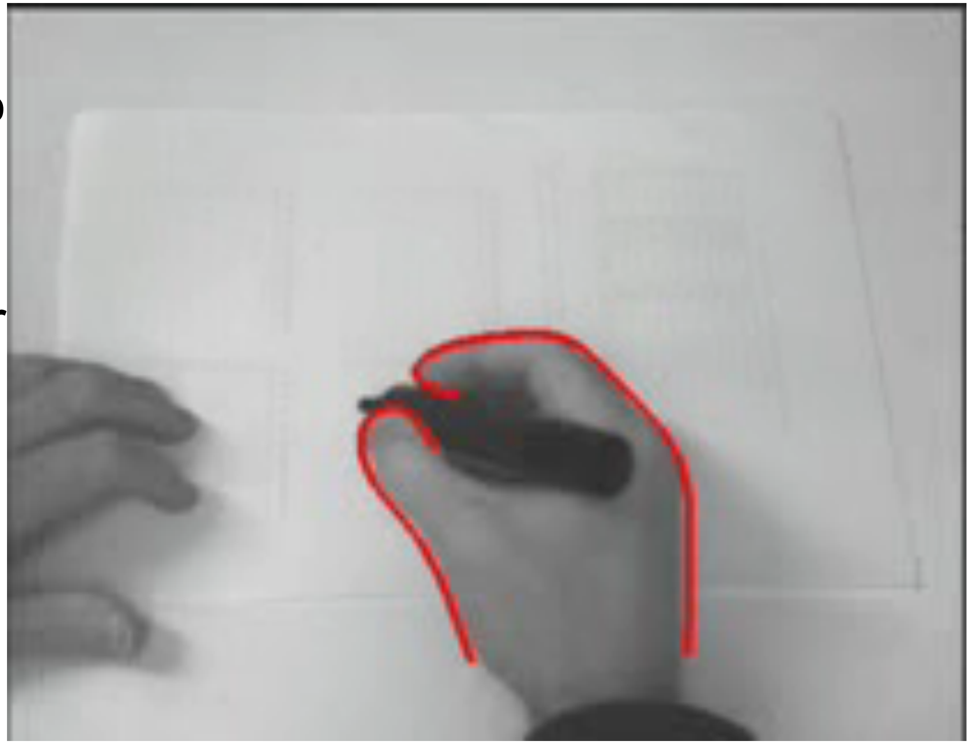


Object model: shape

- **State:** translation/scale+ activity index
- **Dynamics:**
  - Markov transition (activity indices) => probability of changing activity
  - AR-2 models on location (depending o activity variable)
- **Proposal:** dynamics => boostrap filter
- **Observations**

  grey scale values on point lying on perpendicular contours to the shape
- **Likelihood**

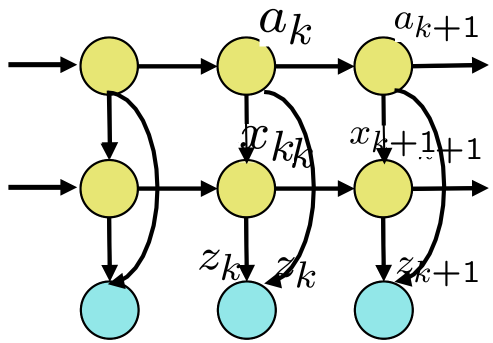
  gaussians with mean depending on whether the point is inside/outside of the object



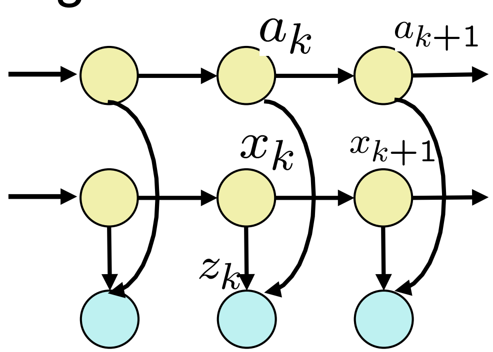
Activities: **Red** line drawing, **Blue**: pause, **Green**: scribbling

[Isard Blake, 2001]

Tracking with auxiliary variables



- Discrete auxiliary processes
  - not only for switching between dynamics
  - ⇒ also influences likelihood
  - => E.g. a_k
    - switching between reference appearence (cf examplars)
    - existence (a_k=1) or not (a_k=0) of object in the image



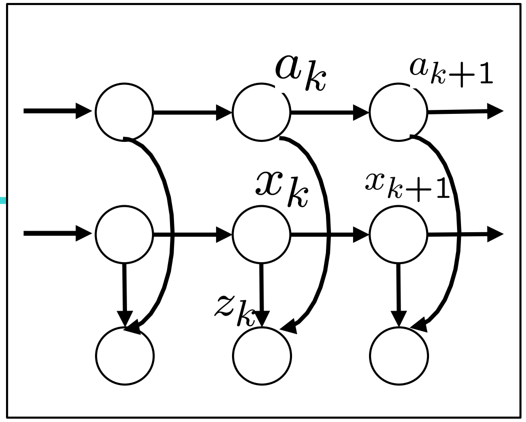
- Model example
  - dynamics: independence of state variables

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

  - likelihood $$p(\mathbf{z}_k|\mathbf{x}_k, a_k = i) = p_i(\mathbf{z}_k|\mathbf{x}_k)$$

(note: state variables **dependent** given observations, due to the explaining away principle)

Tracking with examplars

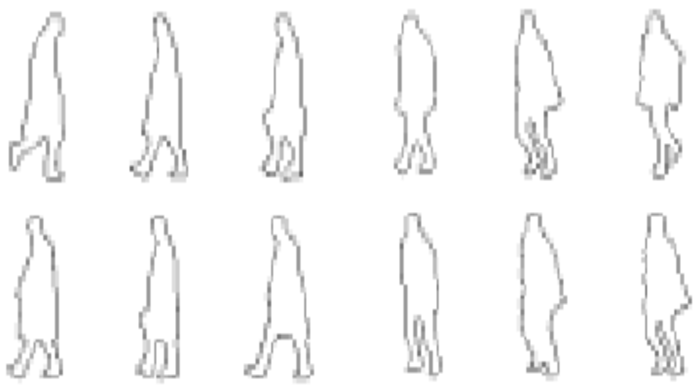


Object model: catalogue of shape/appearance templates

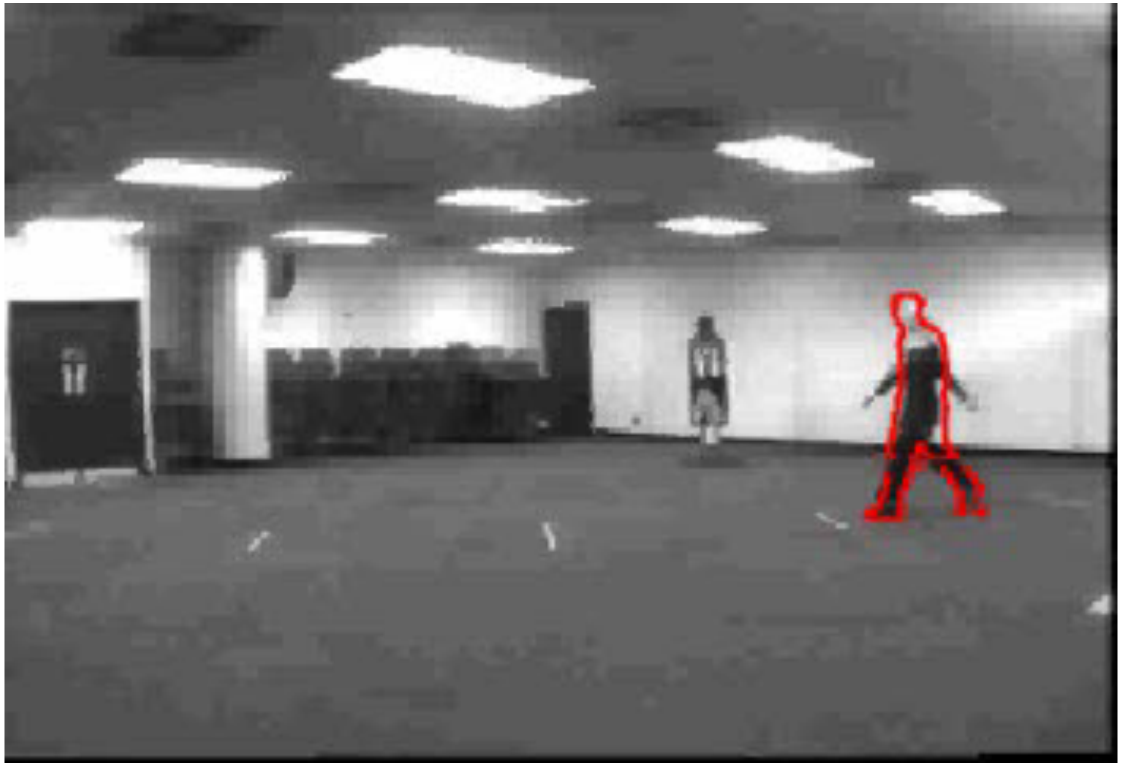
- **State:** translation, scale in x and y
  + examplar index (discrete)

=> **joint** estimation of the **location and shape** that best fit the data

- **Dynamics:** AR-2 model (on location) Markov transition (on examplar indices)
- **Proposal:** dynamics => boostrap filter
- **Observations and likelihood**:
  based on chamfer distance
  (distance **from examplar a_k** edges to nearest image edge)



Set of examplars

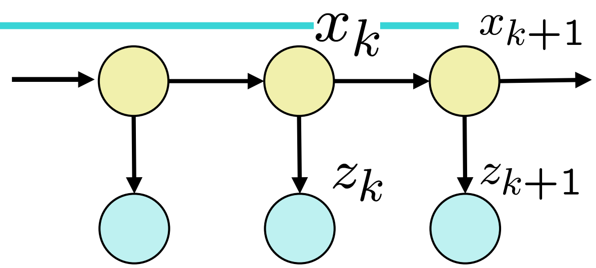


[Toyama Blake, 2001]

Particle filters

- Several issues
  - Proposal
  - Data fusion
  - Multi mode handling
  - (Frequency of resampling – in appendix)

About proposals: bootstrap filter



- Data likelihood
  - contour measures, color distribution
  - might be unspecific → $p(\mathbf{z}_k|\mathbf{x}_k)$ **multimodal** → ambiguities

$$p(\mathbf{x}_k|\mathbf{x}_{k-1})$$
$$\mathbf{x}_k = A\mathbf{x}_{k-1} + \eta_k$$

- Dynamics: 2 contradictory roles
  1. **as prior:** small variance (to increase prior level in case of smooth motion => less sensitivity to ambiguities)
  2. **as proposal:** noise variance large enough to handle sudden/fast motion and configuration changes

     => propose particles in a larger region than where they are expected using smooth motion

=> tuning of dynamical parameters difficult to obtain good results

Using better proposals

- dynamic prior might be unsufficient to extract good samples near likelihood modes (e.g. if tracker is lost/distracted)

  => use data at current instant to sample from

- there exist some technics to approach the optimal proposal (unscented filter, auxiliary PF, hybrid..)

  => involved, not always efficient

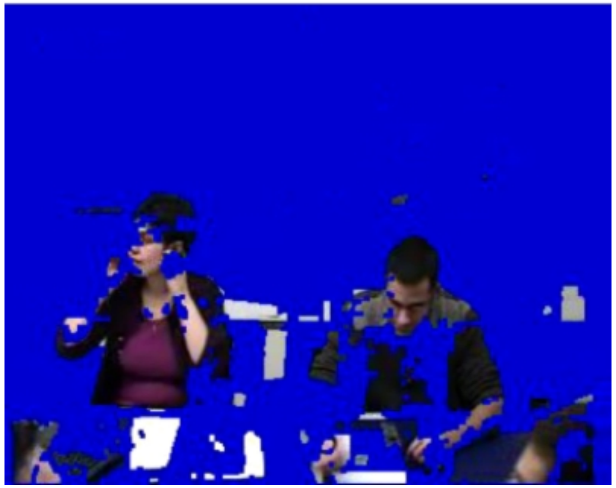
- finding the modes of the likelihood target is usually not possible
  => **use detection based** on
  - other cues (e.g. color, motion, audio etc) to do sampling
  - part or approximation of the cues

Using better proposals

- example: head tracking
- proposal goal : sample new particles in **high likelihood** regions

=> proposal defined as a mixture

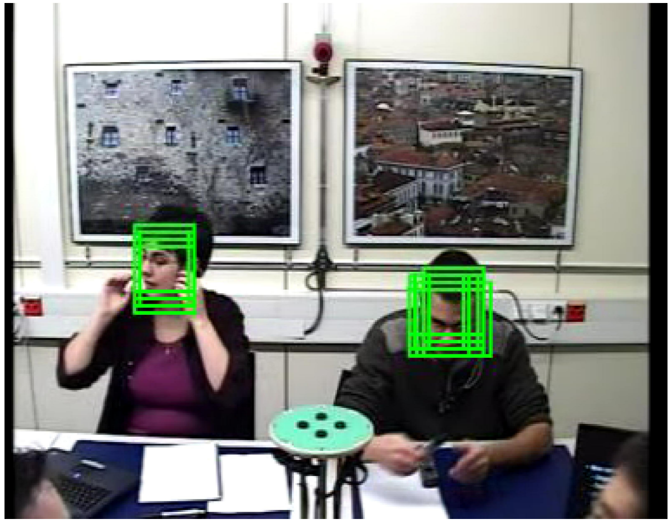
$$q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k) = \beta p(\mathbf{x}_k|\mathbf{x}_{k-1}) + \frac{1-\beta}{D_i}\sum_{d=1}^{D_i}\mathcal{N}(\mathbf{x}_k;\mu_d,\Gamma_d)$$



- state dynamics

  => preserves temporal continuity
- output of a head detector: Di detections

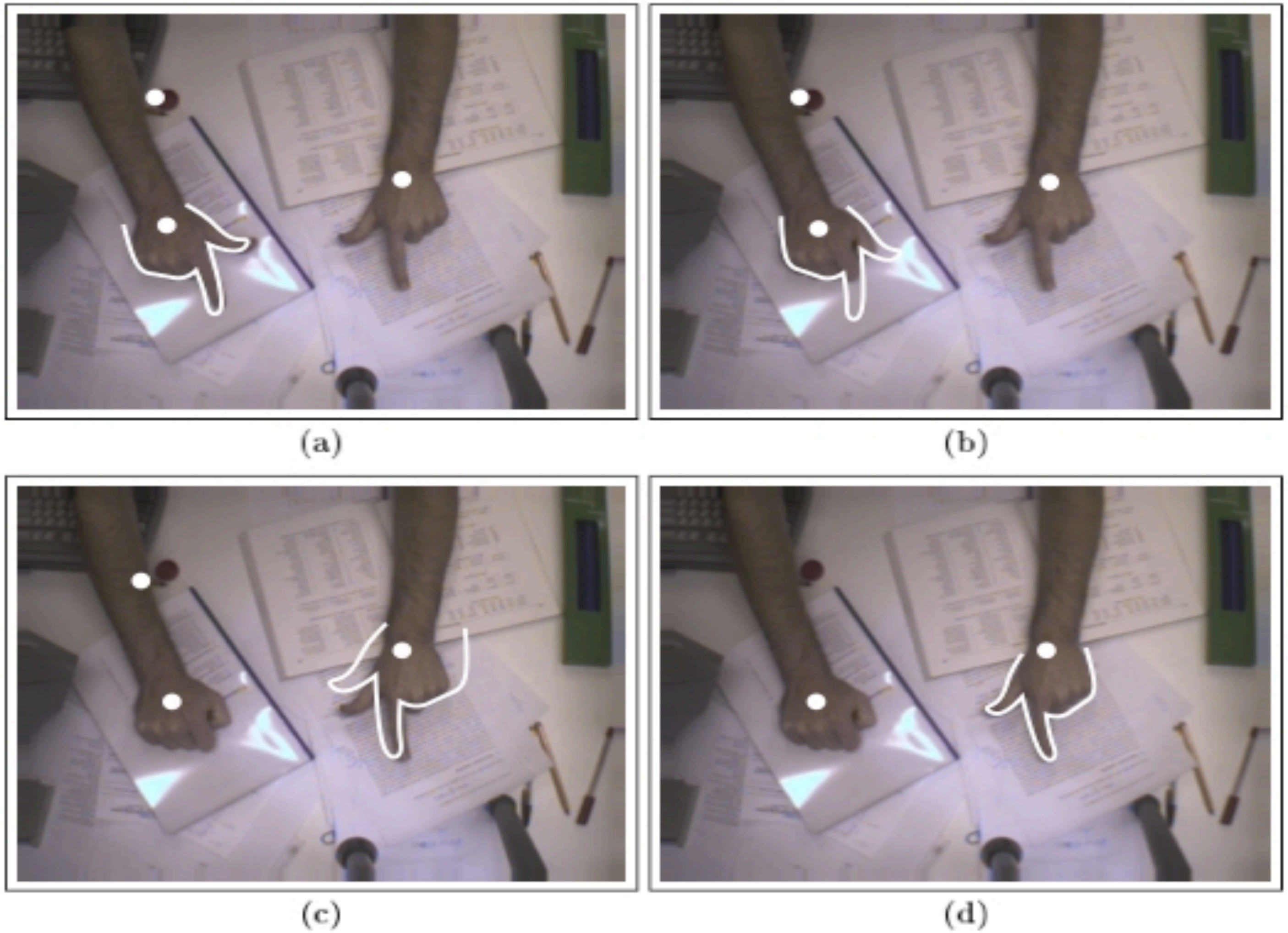
  => automatic (re)initialization and failure recovery



Example: I-Condensation [Isard & Blake 1998]

- **Object:** shape space
- **State:** location/scale/rotation/handness (left/right)
- **Likelihood:** shape measures
- **Proposal: mixture**
  - Dynamics (AR-2)
  - Detections
    - skin blobs (only used to sample location)
    - other parameters sampled from prior distribution

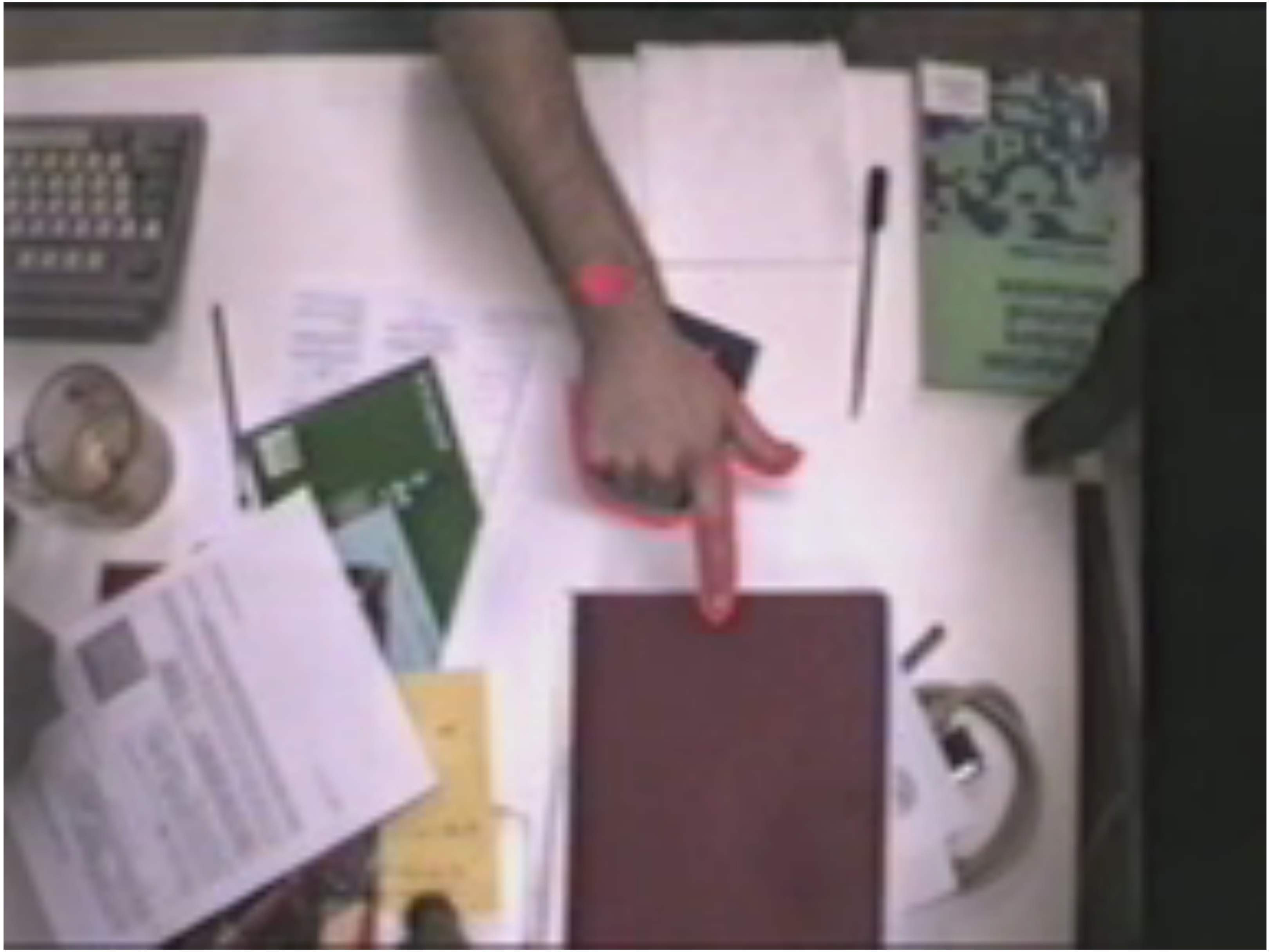
Example: I-Condensation [Isard & Blake 1998]



(a) (b) (c) (d)

- sequence of results: when the current dominant state model (right hand in a) does not fit well anymore (index finger of right hand unstretched in b), the left hand model super-seeds and takes over after several frames

Example: I-Condensation [Isard & Blake 1998]



Data fusion

- Data provide complementary information
  - constantly observed but ambiguous (e.g. shape, color)
  - intermittent, but potentially precise (e.g motion, audio)
  - sensitive to different clutter, invariant to different perturbations (e.g. global color histogram, local intensity, contours)

- Usual assumption: **conditional independence**

$$p(\mathbf{z}_k^1 \dots \mathbf{z}_k^A | \mathbf{x}_k) = \prod_{a=1}^{A} p(\mathbf{z}_k^a | \mathbf{x}_k)$$

Example: contours/color [odobez et al, 2005]

- Object model : element of shape space (ellipse)
- State space : subspace of affine transform
- Proposal: dynamics
- Dynamics : AR model, order 2 (independent on each parameter)

Obs/Likelihood : CONTOURS (CONDENSATION) Obs/Likelihood : COLOR HISTOGRAM

- **mean** configuration in **red** - **highly likely** particles in **yellow**

Example: contours/color [odobez et al, 2005]

- Object model : element of shape space (ellipse)
- State space : subspace of affine transform
- Proposal: dynamics
- Dynamics : AR model, order 2 (independent on each parameter)

Obs/Likelihood : Product of CONTOURS and COLOR likelihoods

- **mean** configuration in **red** - **highly likely** particles in **yellow**

Example: contours/color/motion [odobez et al, 2005]

- Object model : element of shape space (ellipse)
- State space : subspace of affine transform

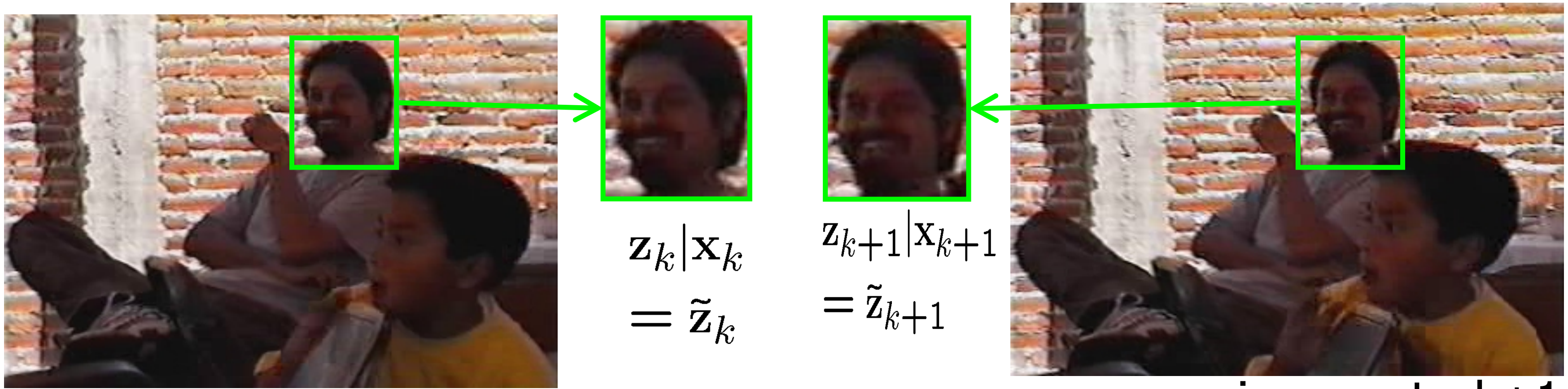
- Proposal:
  - dynamics

    (also, particle drawn from

    motion estimated between frames)

- Dynamics :
  - AR model, order 2 (independent on each parameter)

Data likelihood: discussion on temporal conditional independence



$\mathbf{z}_k | \mathbf{x}_k = \tilde{\mathbf{z}}_k$

$\mathbf{z}_{k+1} | \mathbf{x}_{k+1} = \tilde{\mathbf{z}}_{k+1}$

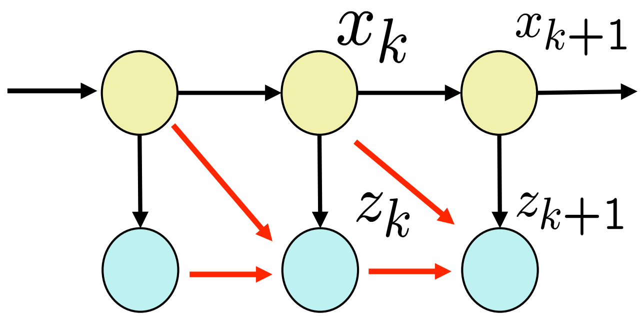
image at k

image at k+1

$$\boxed{\tilde{\mathbf{z}}_{k+1} = \tilde{\mathbf{z}}_k + noise}$$

- Given states, **high correlation** between observations
  => hypothesis not valid

$$p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k}, \mathbf{x}_{0:k+1}) \neq p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$$



- Solution: change the model accordingly

$$p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k}, \mathbf{x}_{0:k+1}) = p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}, \mathbf{z}_k, \mathbf{x}_k)$$

Data likelihood

$$p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \mathbf{z}_k, \mathbf{x}_k) = p_{obj}(\mathbf{z}^o_{k+1}|\mathbf{x}_{k+1}) \times p_{corr}(\mathbf{z}^c_{k+1}|\mathbf{x}_{k+1}, \mathbf{z}^c_k, \mathbf{x}_k)$$

- Hypothesis: independence between observations from
  - object model: where is the object in the current image
  - temporal correlation: object motion follows optical flow

- Object model: shape on clutter

- Temporal term:

$$p_{corr}(\mathbf{z}^c_{k+1}|\mathbf{x}_{k+1}, \mathbf{z}^c_k, \mathbf{x}_k) \propto \exp\left(-\lambda_c d_c(\tilde{\mathbf{z}}_k, \tilde{\mathbf{z}}_{k+1})\right)$$

  - patch distance dc
    - normalized correlation coefficient

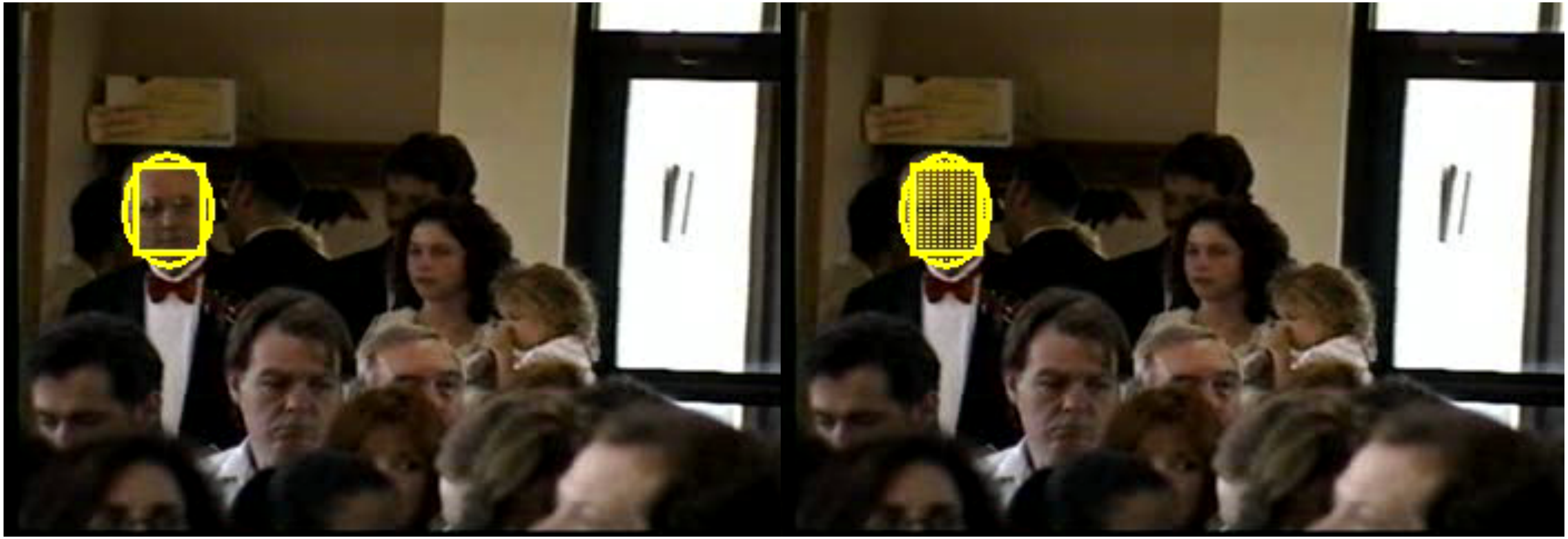
    => Implicit motion likelihood

$\mathbf{z}^c_k|\mathbf{x}_k = \tilde{\mathbf{z}}_k$

$\mathbf{z}^c_{k+1}|\mathbf{x}_{k+1} = \tilde{\mathbf{z}}_{k+1}$

Results

- First example: 500 particles, all parameters identical
  - Dynamic noise (a bit larger than normal) $\sigma_{trans} = 6 \quad \sigma_{scale} = 0.05$
- 2 models :
  - M1: CONDENSATION (using shape only, or color only)
  - M2: likeliood model : object likelihood x correlation likelihood



M1 : CONTOURS (CONDENSATION) M2 : CONTOUR + IMPLICIT MOTION

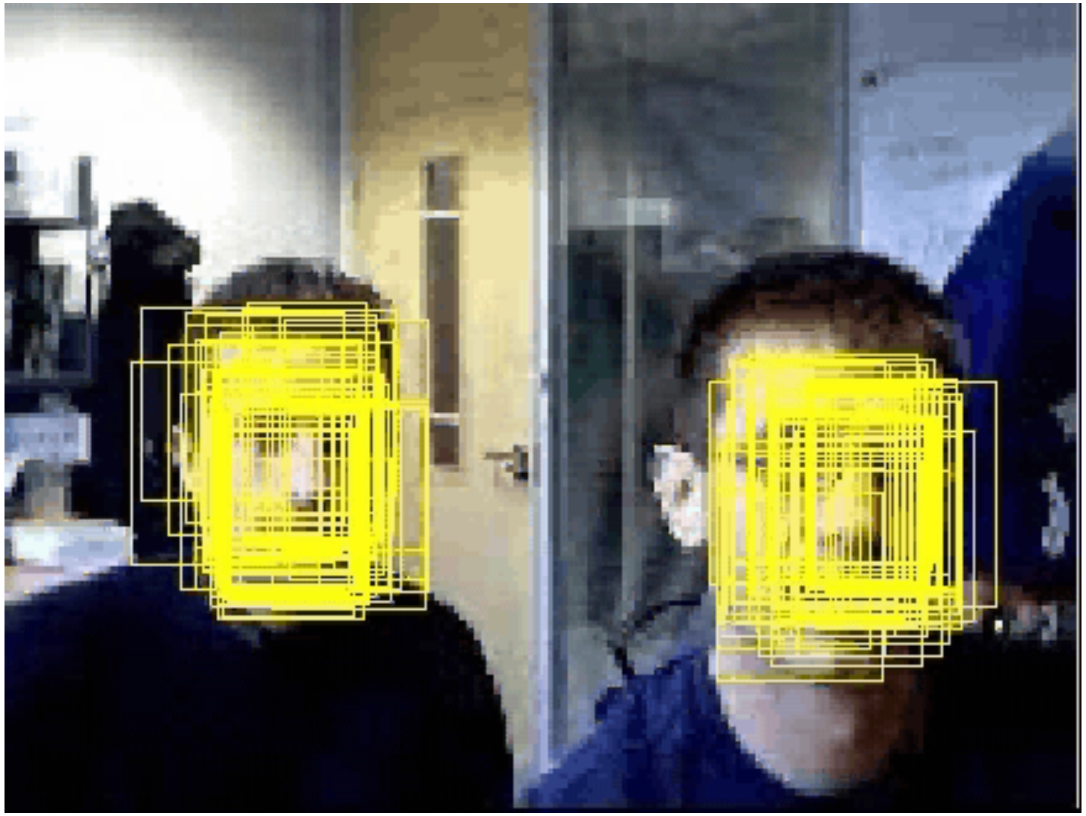
➢ **mean** configuration in **red** - **highly likely** particles in **yellow**

SMC and multimodality

- In theory

  Particle Filter (PF) approximates filtering distribution with *N* weighted samples

- In practice: because of *resampling,* multiple modes not jointly tracked for long
  - Even with large *N*
  - Even with peaks of similar weights

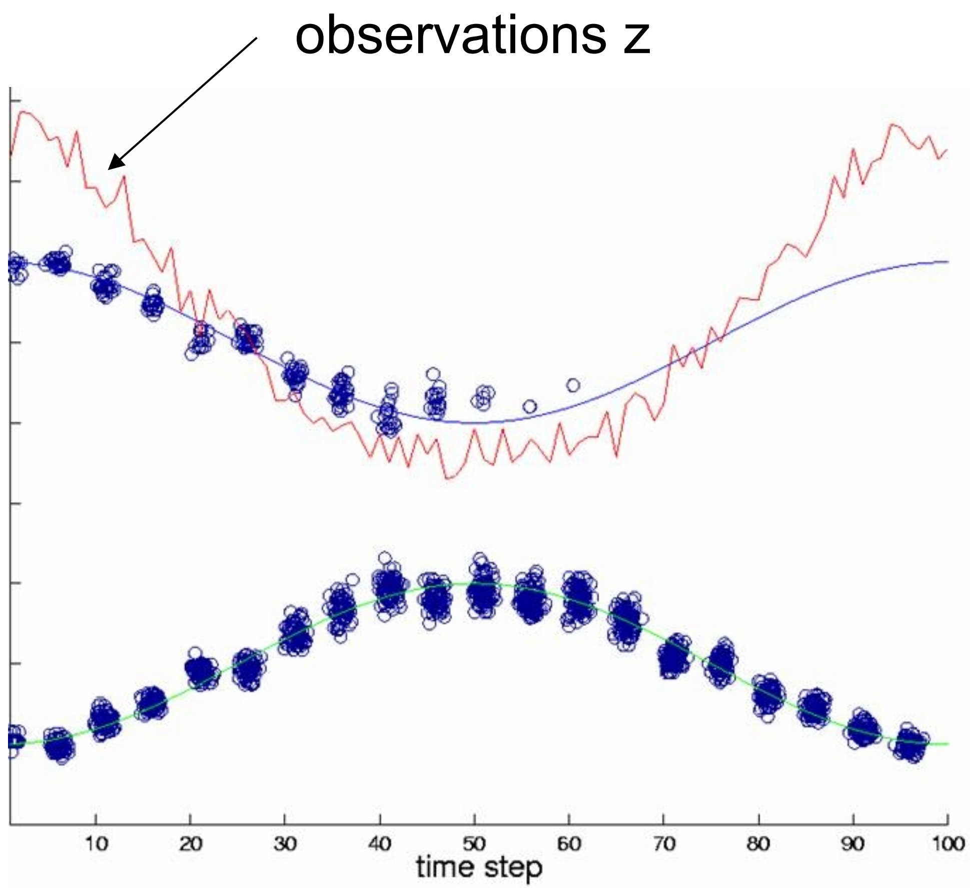


SMC and multimodality

- Example in one dimension

$$\mathbf{z}_k = \mathbf{x}_k^2 + noise$$

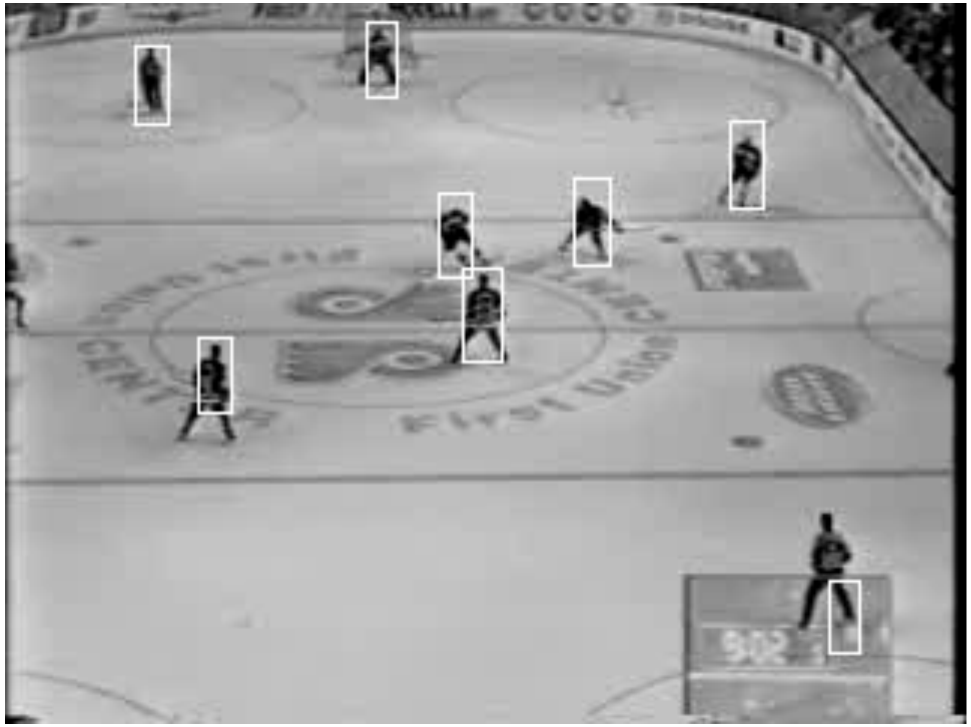
⇒ two modes of the same amplitude



- result **depends on the different noise levels** (process noise, observation noise, data noise)
- after some time, all particles tend to concentrate on one mode
  - more particle than needed to track one mode
  - less particles than needed to explore the second mode
- Consequences
  - sample-based approximation might be much poorer than expected
  - pruning occurs too early => no chance to resolve long standing ambiguities

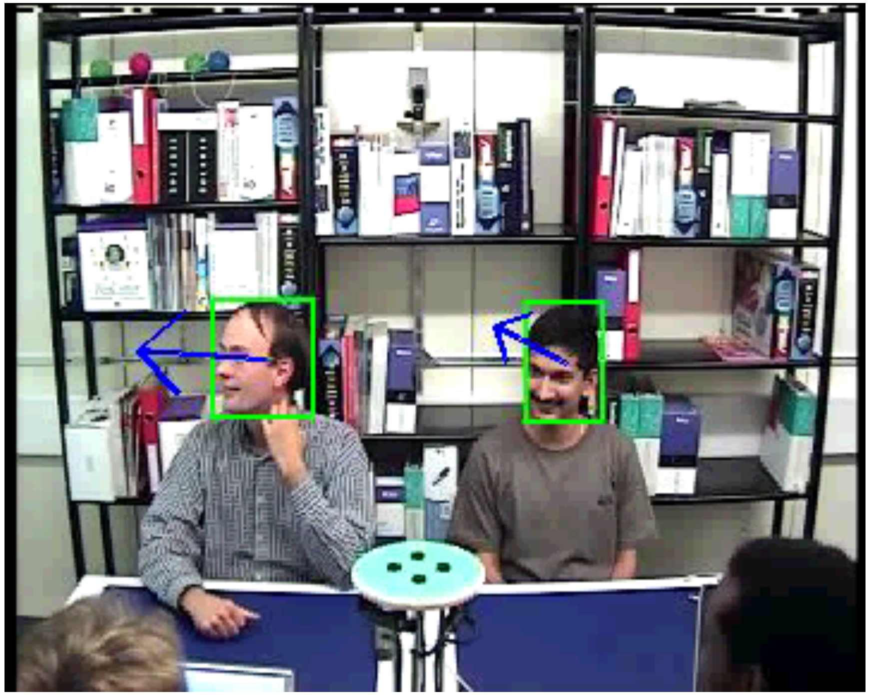
One solution: track mixtures of particles

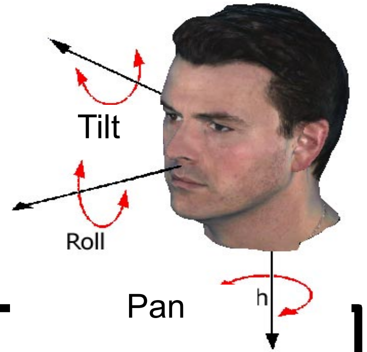
- Cluster the samples around different modes
- Each cluster/mixture can be identified as one ‘object’

Exampel here [Okuma, 2004]: uses detector trained for a given class
to initialize new mixtures/clusters

Final example
Joint Head Location and Pose Tracking [Ba 2005]





Head pose of each person
(pan/tilt)

- Joint optimization of location and pose (coupled problem)

  not head tracking then pose estimation

  - If we know the pose, we can do a better head localisation

  - If we know the head localisation, we can better infer the

  => Doing both simultaneously should help

- Approach
  - Mixes different ingredients we have seen

State model : exemplar approach

- Mixed-state approach

continuous (localization), discrete (appearance exemplar)

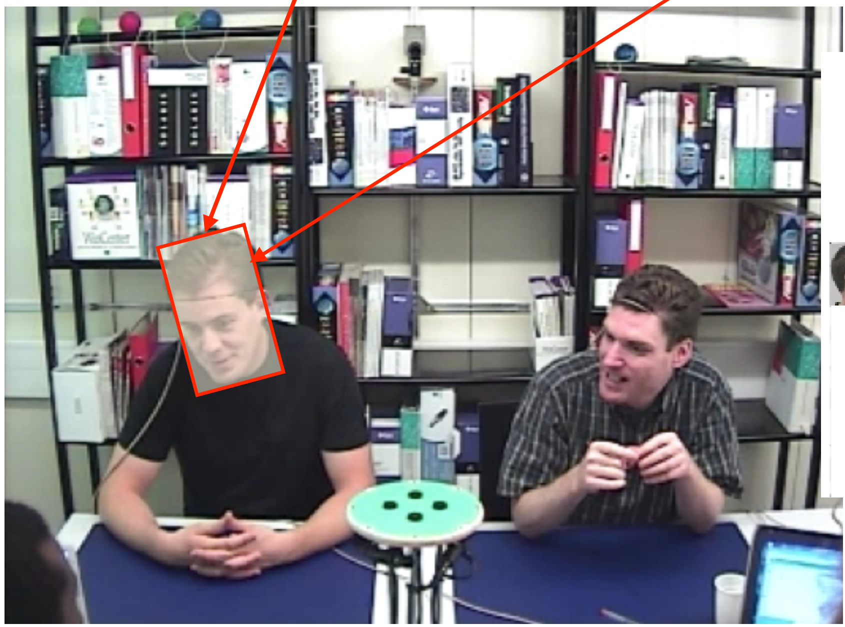
$$X_t = (S_t, r_t, k_t)$$

2D transform

Translation+scaling roll

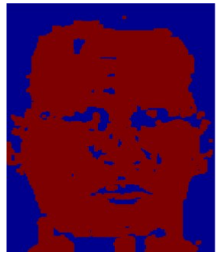
Out of plane head rotation

pose exemplar
(index)



Likelihood modeling p(Z|X)

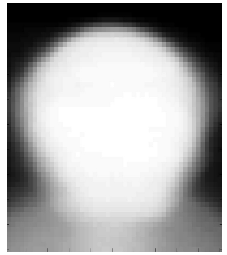
- Features
  - texture/skin features extracted at each position of reference grid
  - silhouette features extracted from a background subtraction image



- Generative head pose models => use of training data
  - texture/skin model (pose dependent, i.e. one for each pose value)

    => p (z | k)
  - silhouette model (pose independent) : p(z)

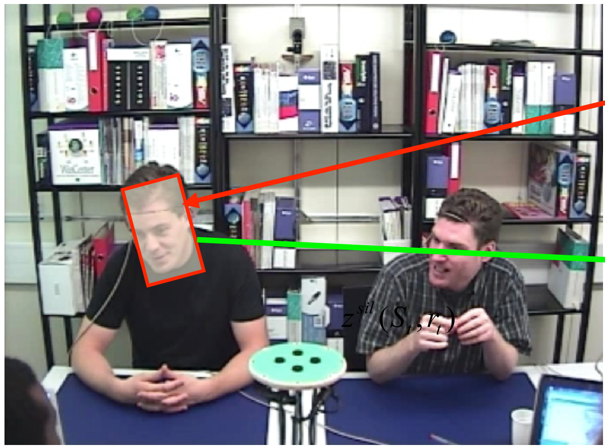
    => used to improve localization



- Observation Likelihood p (z | X)

assuming conditional independence => product of likelihoods

$$p(z|X) \propto p(z^{text}(S,r)|k)p(z^{skin}(S,r)|k)p(z^{sil}(S,r))$$



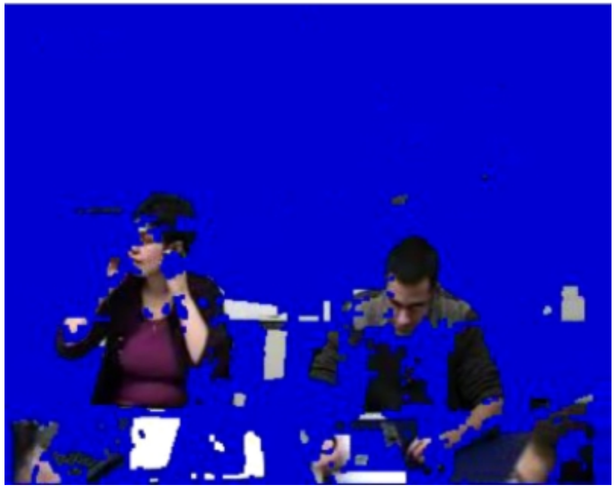
$$X_t = (S_t, r_t, k_t)$$

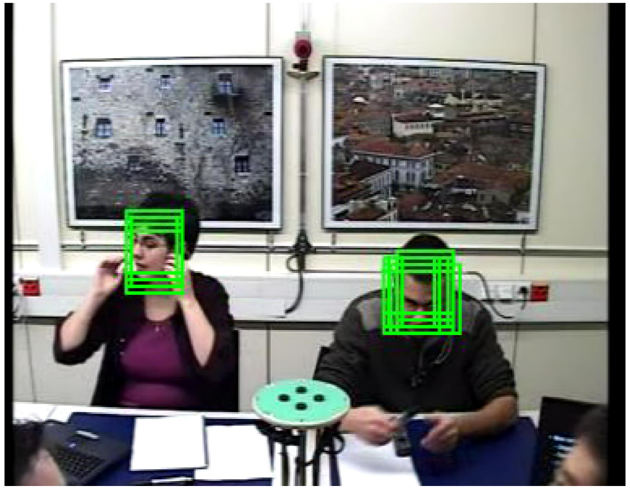
$$\left.\begin{array}{l} z^{text}(S_t, r_t) \\ z^{skin}(S_t, r_t) \\ z^{sil}(S_t, r_t) \end{array}\right\}$$

Proposal function

- Goal: sample new particles in
  high likelihood regions
  => proposal defined as mixture

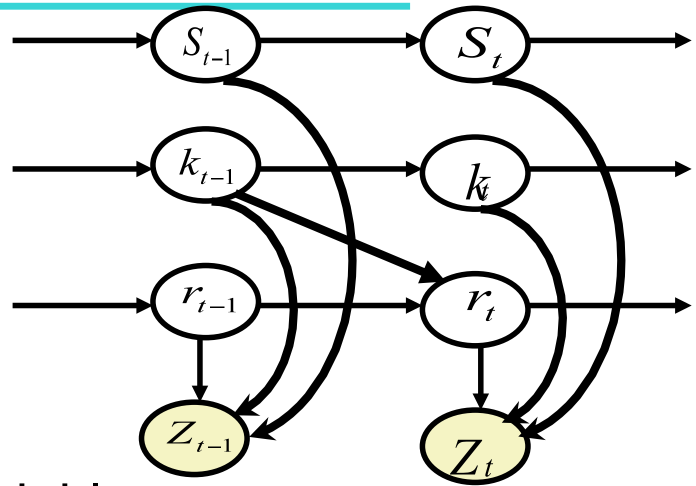
$$q(X_t \mid X_{t-1}^i, z_t) = (1-\varepsilon)\boxed{p(X_t \mid X_{t-1}^i)} + \varepsilon \boxed{\frac{1}{N_d}\sum_{n=1}^{N_d} p_{\text{det}}(X_t \mid X_t^{n,\text{det}}(z_t))}$$





- state dynamics
  => preserves temporal continuity
- output of a head detector
  => automatic (re)initialization and
  failure recovery

Sampling: Rao-Blackwellisation



- Importance sampling approach

=> particle set $\{S_t^i, r_t^i, k_t^i, w_t^i\}_{i=1..N}$

- Alternative : Rao-Blackwellisation
  - Importance sampling applied to continuous variables
    - position/scale/rotation S and r
  - compute exact posteriors for discrete one (here the exemplar index), given the sampled ones

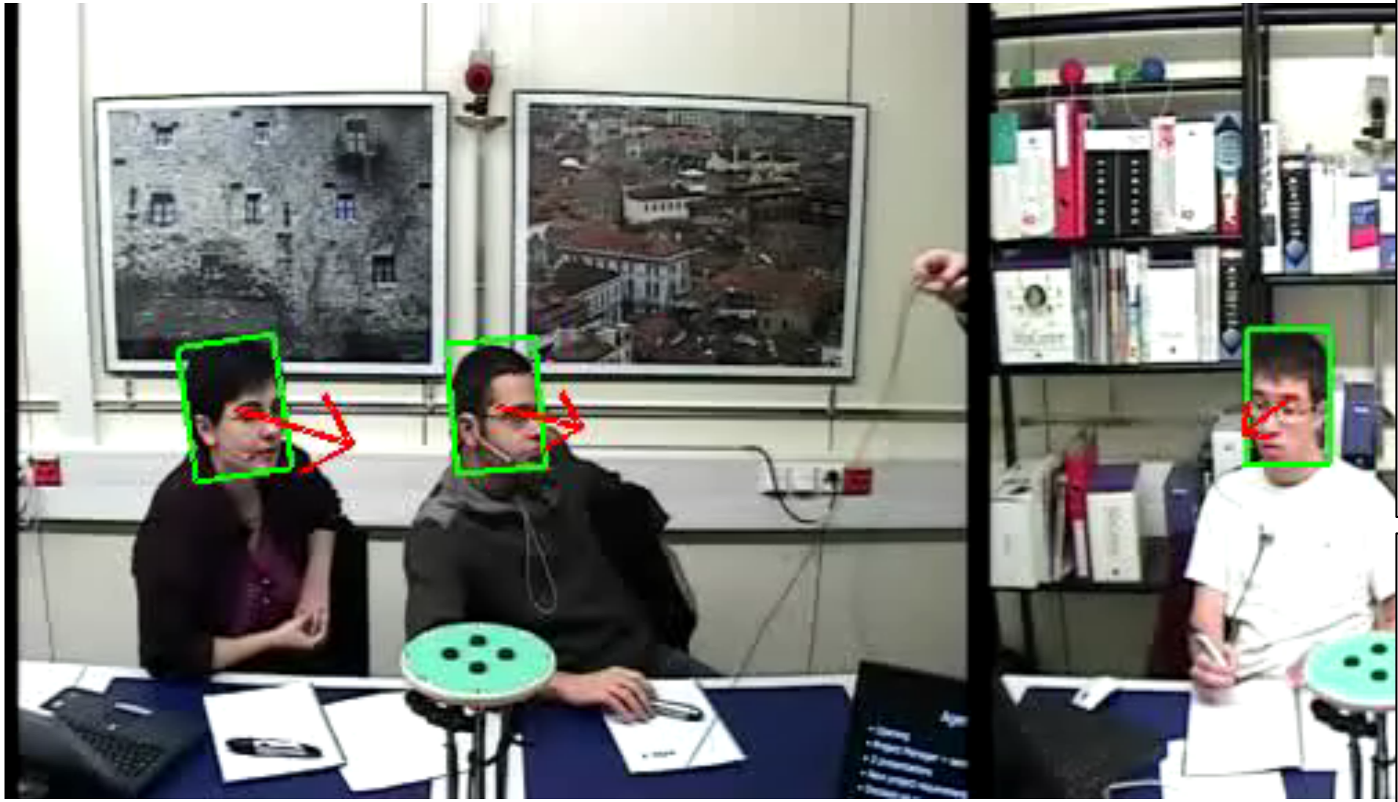
$$\{S_t^i, r_t^i, \pi^i(k_t), w_t^i\}_{i=1..N}$$

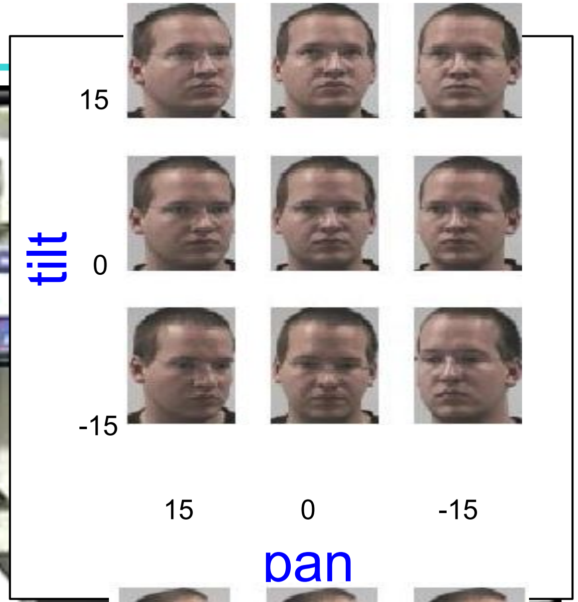
$$\pi^i(k_t) = p(k_t | Z_{1:t}, S_{1:t}^i, r_{1:t}^i)$$

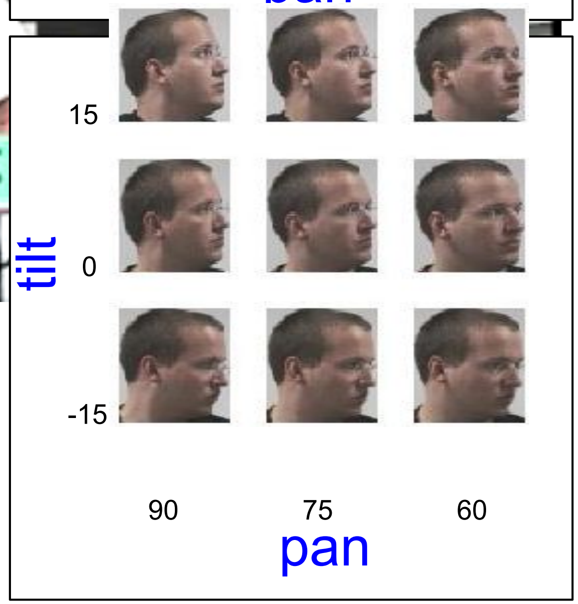
- Advantage
  - better parameter estimates
  - allows to evaluate, for the same image data $Z(S_t^i, r_t^i)$ which head pose is the best (i.e. allows to have comparable likelihood)

=> avoid being trapped in a wrong head pose estimate

Illustration of head pose tracking







- on 60 minute data: around **10-13 degree** error in pan
- tilt more difficult to estimate
- larger error near profile views
- large accuracy variation across people (depending on appearance)

Multi-view CHIL head pose data

- Dataset:
  - lecture room recording
  - smaller head/face resolution
  - 4 camera views, calibrations
- One approach:
  - tracking: head pose tracker independently applied to each of the camera
  - fuse the 4 measurements by combining the 2 more reliable
  - reliability factor
    - higher percentage of skin pixels in localized region
    (face is closer to frontal pose)

Results: CHIL data - demo

- Color squares indicates selected cameras for fusion
  (**green**: selected – **red**: unselected)

- Original views were zoomed in to allow better viewing

- **Blue arrow**: pointing vector

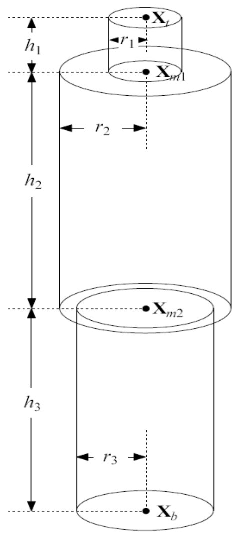
- Notice individual tracker errors

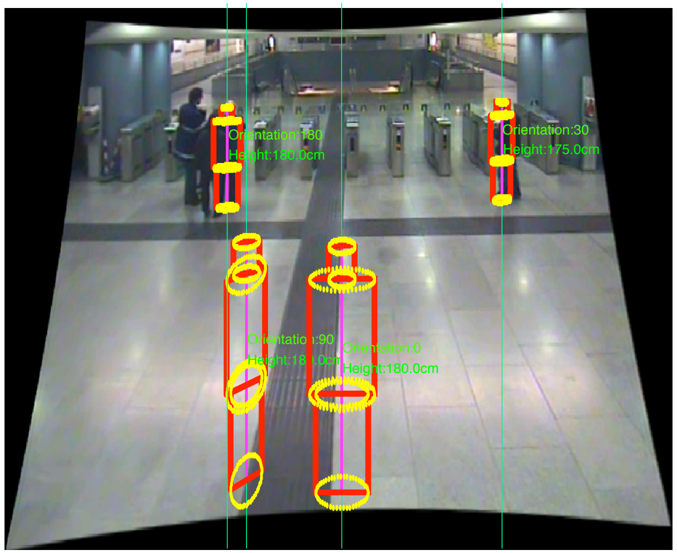
Multi-object tracking ?

- Single object tracking
  - element x in configuration space
    - e.g. 2D : x = (location, scale, activity) : 4 parameters
    - e.g. in 3D:
      - x = (position on ground-plane, speed, height, orientation)

    6 parameters

=> Multiple object tracking ?





3D human body model

Multi-Object Tracking

- Probabilistic approach

state

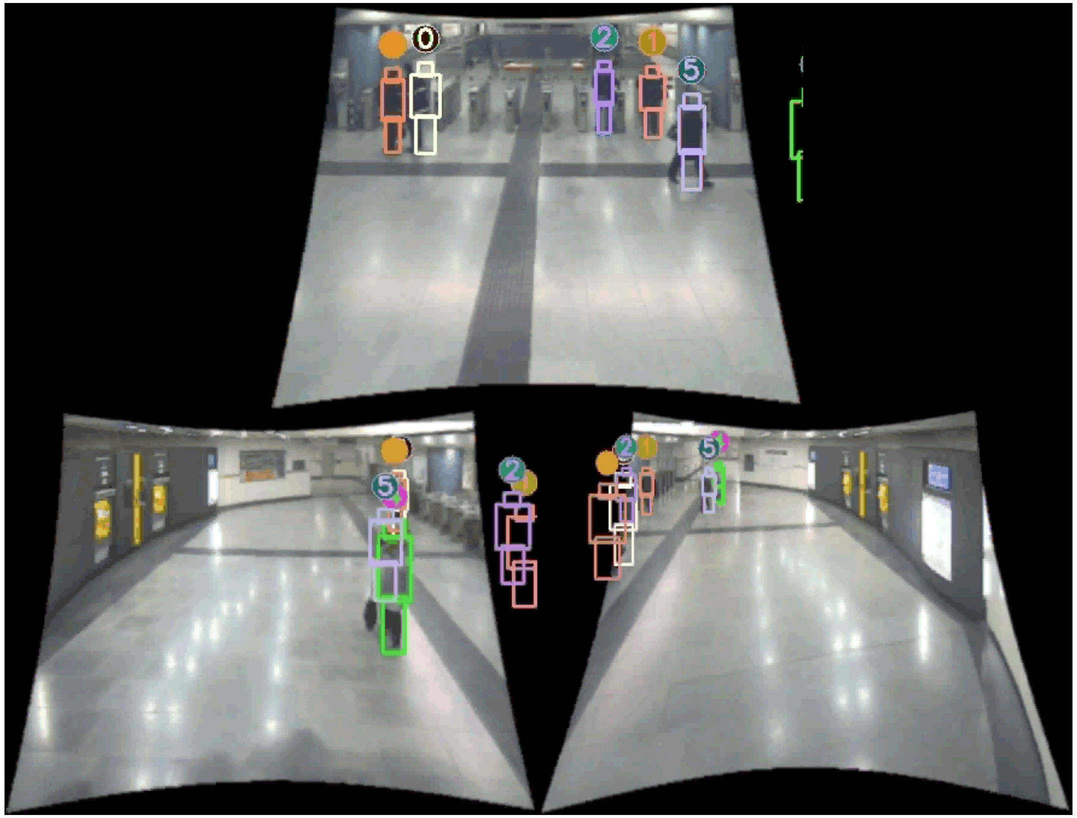
$$p(\tilde{\mathbf{X}}_t|\mathbf{Z}_{1:t}) \propto p(\mathbf{Z}_t|\tilde{\mathbf{X}}_t) \int p(\tilde{\mathbf{X}}_t|\tilde{\mathbf{X}}_{t-1})p(\tilde{\mathbf{X}}_{t-1}|\mathbf{Z}_{1:t-1})d\tilde{\mathbf{X}}_{t-1}$$

observations

observation likelihood model

dynamical model

- Issues
  - What is the state ?
  - Multi-object dynamic ?
  - Observation model ?
  - Optimization ?



Particle filters: conclusion

- Advantages
  - easy to implement and expand (addition of new variables, defining more precise likelihood, dynamics)
  - robust to clutter and brief occlusions
  - a lot of theoretical tools
  - **applicable to any filtering problem (not only visual tracking)**

- Problems
  - jitter of final estimate (mean ? mode of distribution ?)
  - computational load (on average, more samples –i.e. likelihood evaluations- than iteration in gradient descent algorithms)
  - only brief capture of multimodality

- Others
  - often, dynamics simply maintain temporal coherence
  - A discriminant and robust data model for the task at hand remains the challenge

readings and acknowledgement

- [Arulampalam et al.] *A tutorial on Particle Filters for on-line non-linear/ non-Gaussian Bayesian tracking,* IEEE Trans. Signal Processing 2001
- [North et al] *Learning and classification of complex dynamics*, B. North, A. Blake, M. Isard and J. Rittscher, PAMI 2000
- [Isard & Blake] *A mixed-state CONDENSATION tracker with automatic model-switching*, ECCV 1998
- [Blake 1998] *Active contours*, A. Blake and M. Isard, Springer 1998.
- [Odobez et al] *Embedding motion in model-based stochastic tracking,* IEEE Trans. On Image Processing, 2006