Tracking
with focus on the particle filter
(part II)

Michael Rubinstein

IDC

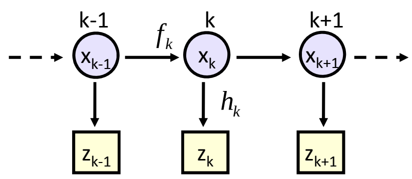
Last time…

- Background
  - State space
  - Dynamic systems
  - Recursive Bayesian filters
  - Restrictive cases
    - Kalman filter
    - Grid-based filter
  - Suboptimal approximations
    - Extended Kalman filter

This talk

- Particle filtering
  - MC integration
  - Sequential Importance Sampling (SIS)
  - Resampling
  - PF variants
- Multiple-target tracking
  - BraMBLe: A Bayesian Multiple-Blob Tracker/ Isard, MacCormick

Dynamic System



Stochastic diffusion

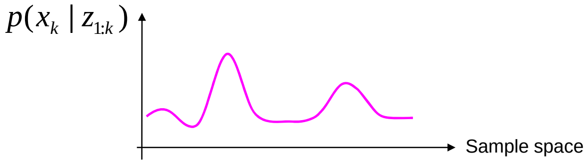
State equation: $x_k = f_k(x_{k-1}, \boxed{v_k})$

x_k state vector at time instant k
f_k state transition function, $f_k : R^{N_x} \times R^{N_v} \to R^{N_x}$
v_k i.i.d process noise

Observation equation: $z_k = h_k(x_k, \boxed{w_k})$

z_k observations at time instant k
h_k observation function, $h_k : R^{N_x} \times R^{N_w} \to R^{N_z}$
w_k i.i.d measurement noise

Recursive Bayes filter



- Prediction:

$$\boxed{p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{1:k-1})\, dx_{k-1}} \qquad (1)$$

- Update:

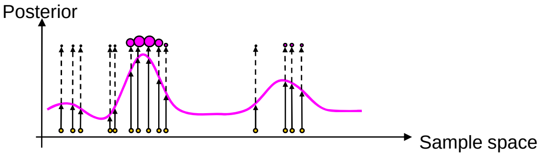
$$\boxed{p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}} \qquad (2)$$

$$p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})\, dx_k$$

Particle filtering

- Many variations, one general concept:

Represent the posterior pdf by a set of randomly chosen weighted samples (particles)



- Randomly Chosen = Monte Carlo (MC)
- As the number of samples become very large – the characterization becomes an equivalent representation of the true pdf

Particle filtering

- Compared to methods we've mentioned last time
  - Can represent any arbitrary distribution
    - multimodal support
  - Keep track of many hypotheses as there are particles
  - **Approximate representation of complex model rather than exact representation of simplified model**

- The basic building-block: *Importance Sampling*

Monte Carlo integration

- Evaluate complex integrals using probabilistic techniques
- Assume we are trying to estimate a complicated integral of a function *f* over some domain *D*:

$$F = \int_D f(\vec{x})d\vec{x}$$

- Also assume there exists some PDF *p* defined over *D*

Monte Carlo integration

- Then

$$F = \int_D f(\vec{x})d\vec{x} = \int_D \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x})d\vec{x}$$

- But

$$\int_D \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x})d\vec{x} = E\left[\frac{f(\vec{x})}{p(\vec{x})}\right], x \sim p$$

- This is true for <u>any</u> PDF *p* over D!

Monte Carlo integration

- Now, if we have i.i.d random samples $\vec{x}_1,\ldots,\vec{x}_N$ sampled from *p*, then we can approximate $E\left[\frac{f(\vec{x})}{p(\vec{x})}\right]$ by

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\vec{x}_i)}{p(\vec{x}_i)}$$

- Guaranteed by law of large numbers:

$$N \to \infty, F_N \xrightarrow{a.s} E\left[\frac{f(\vec{x})}{p(\vec{x})}\right] = F$$

Importance Sampling (IS)

- What about $p(\vec{x}) = 0$?
- If *p* is very small, f / p can be arbitrarily large, 'damaging' the average **Importance weights**
  - Design *p* such that f / p is bounded
  - Rule of thumb: take *p* similar to *f* as possible

***Importance* or *proposal* density**

- The effect: get more samples in 'important' areas of *f*, i.e. where *f* is large

Convergence of MC integration


Pafnuty Lvovich Chebyshev

- Chebyshev's inequality: let X be a random variable with expected value μ and std σ. For any real number k>0,

$$\Pr\{|X-\mu| \geq k\sigma\} \leq \frac{1}{k^2}$$

- For example, for $k=\sqrt{2}$, it shows that at least half the values lie in interval $(\mu-\sqrt{2}\sigma, \mu+\sqrt{2}\sigma)$
- Let $y_i = \frac{f(x_i)}{p(x_i)}$, then MC estimator is $F_N = \frac{1}{N}\sum_{i=1}^{N} y_i$

Convergence of MC integration

- By Chebyshev's,

$$\Pr\{|F_N - E[F_N]| \geq \left(\frac{V[F_N]}{\delta}\right)^{1/2}\} \leq \delta \qquad (k = 1/\sqrt{\delta})$$

$$V[F_N] = V\left[\frac{1}{N}\sum_{i=1}^{N} y_i\right] = \frac{1}{N^2} V\left[\sum_{i=1}^{N} y_i\right] = \frac{1}{N^2}\sum_{i=1}^{N} V[y_i] = \frac{1}{N} V[y]$$

$$\Rightarrow \quad \Pr\{|F_N - F| \geq \frac{1}{\sqrt{N}}\left(\frac{V[y]}{\delta}\right)^{1/2}\} \leq \delta$$

- Hence, for a fixed threshold, the error decreases at rate $1/\sqrt{N}$

Convergence of MC integration

- Meaning
  1. To cut the error in half, it is necessary to evaluate 4 times as many samples
  2. Convergence rate is independent of the integrand dimension!
     - On contrast, the convergence rate of grid-based approximations decreases as N_x increases

IS for Bayesian estimation

$$E(f(X)) = \int_X f(x_{0:k}) p(x_{0:k} \mid z_{1:k}) dx_{0:k}$$
$$= \int_X f(x_{0:k}) \frac{p(x_{0:k} \mid z_{1:k})}{q(x_{0:k} \mid z_{1:k})} q(x_{0:k} \mid z_{1:k}) dx_{0:k}$$

- We characterize the posterior pdf using a set of samples (particles) and their weights

$$\{x_{0:k}^i, w_k^i\}_{i=1}^N$$

- Then the joint posterior density at time k is approximated by

$$p(x_{0:k} \mid z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_{0:k} - x_{0:k}^i)$$

IS for Bayesian estimation

- We draw the samples from the importance density $q(x_{0:k} \mid z_{1:k})$ with importance weights

$$w_k^i \propto \frac{p(x_{0:k} \mid z_{1:k})}{q(x_{0:k} \mid z_{1:k})}$$

- Sequential update (after some calculation…)

Particle update

$$\boxed{x_k^i \sim q(x_k \mid x_{k-1}^i, z_k)}$$

Weight update

$$\boxed{w_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)}}$$

Sequential Importance Sampling (SIS)

$$\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{SIS}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\right]$$

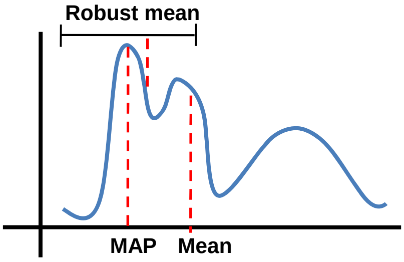
- FOR i=1:N
  - Draw $x_k^i \sim q(x_k \mid x_{k-1}^i, z_k)$
  - Update weights $w_k^i = w_{k-1}^i \dfrac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)}$
- END
- Normalize weights

State estimates

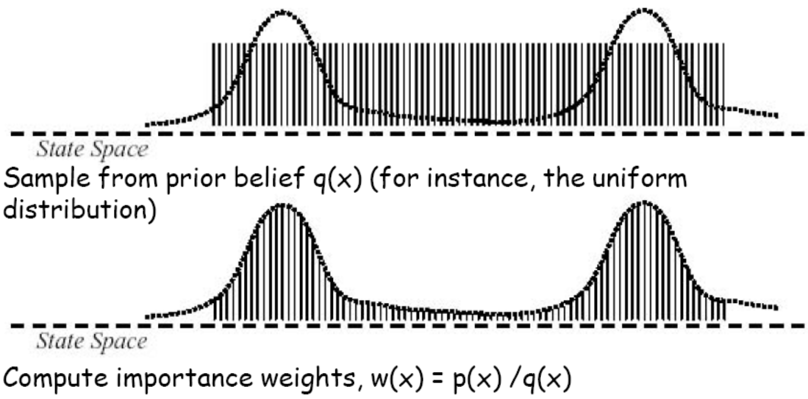
- Any function $f(x_k)$ can be calculated by discrete pdf approximation

$$E\left[f(x_k)\right] = \frac{1}{N}\sum_{i=1}^{N} w_k^i\, f(x_k^i)$$

- Example:
  - Mean (simple average)
  - MAP estimate: particle with largest weight
  - Robust mean: mean within window around MAP estimate



Choice of importance density



Hsiao et al.

Choice of importance density

- Most common (suboptimal): the transitional prior

$$q(x_k \mid x_{k-1}^i, z_k) = p(x_k \mid x_{k-1}^i)$$

$$\Rightarrow w_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} = w_{k-1}^i\, p(z_k \mid x_k^i)$$

Grid filter weight update: $$w_{k|k}^i = \frac{w_{k|k-1}^i\, p(z_k \mid x_k^i)}{\sum_{j=1}^{N_s} w_{k|k-1}^j\, p(z_k \mid x_k^j)}$$

The degeneracy phenomenon

- Unavoidable problem with SIS: after a few iterations most particles have negligible weights
  - Large computational effort for updating particles with very small contribution to $p(x_k \mid z_{1:k})$
- Measure of degeneracy - the effective sample size:

$$\boxed{N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}}$$

  - Uniform: $N_{eff} = N$, severe degeneracy: $N_{eff} = 1$

Resampling

- The idea: when degeneracy is above some threshold, eliminate particles with low importance weights and multiply particles with high importance weights

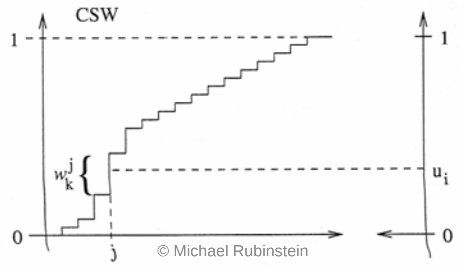
$$\{x_k^i, w_k^i\}_{i=1}^N \rightarrow \{x_k^{i*}, \tfrac{1}{N}\}_{i=1}^N$$

- The new set is generated by sampling with replacement from the discrete representation of $p(x_k \mid z_{1:k})$ such that $\Pr\{x_k^{i*} = x_k^j\} = w_k^j$

Resampling

$$\left[\{x_k^{i*}, w_k^i\}_{i=1}^N\right] = \mathrm{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^N\right]$$

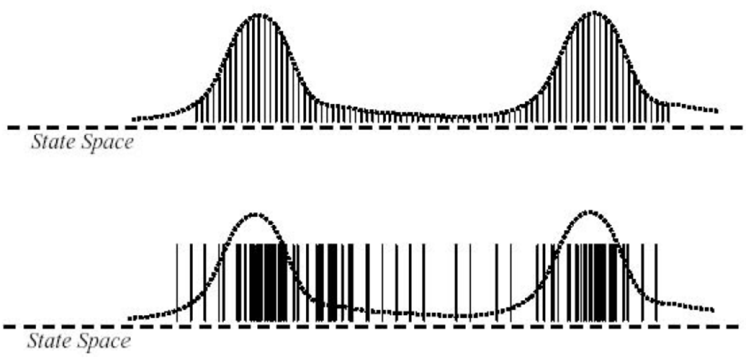
- Generate N i.i.d variables $u_i \sim U[0,1]$
- Sort them in ascending order
- Compare them with the cumulative sum of normalized weights



Ristic et al.

Resampling

- Complexity: O(NlogN)
  - O(N) sampling algorithms exist



Hsiao et al.

Generic PF

$$\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{PF}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\right]$$

- Apply SIS filtering $\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{SIS}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\right]$
- Calculate N_{eff}
- IF $N_{eff} < N_{thr}$
  - $\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^N\right]$
- END

Generic PF

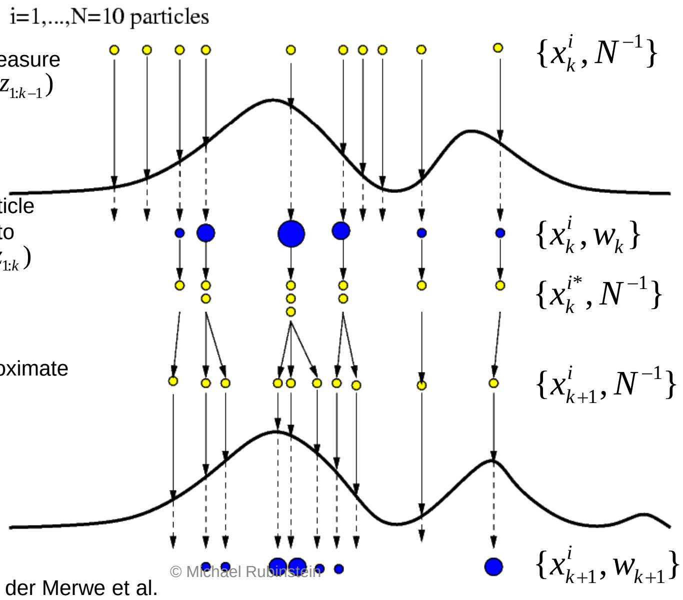
Uniformly weighted measure
Approximates $p(x_k \mid z_{1:k-1})$

Compute for each particle its importance weight to Approximate $p(x_k \mid z_{1:k})$

(Resample if needed)

Project ahead to approximate $p(x_{k+1} \mid z_{1:k})$

$p(x_{k+1} \mid z_{1:k+1})$



Van der Merwe et al.

PF variants

- Sampling Importance Resampling (SIR)
- Auxiliary Sampling Importance Resampling (ASIR)
- Regularized Particle Filter (RPF)
- Local-linearization particle filters
- Multiple models particle filters (maneuvering targets)
- …

Sampling Importance Resampling (SIR)

- A.K.A Bootstrap filter, Condensation

- **Initialize** $\{x_0^i, w_0^i\}_{i=1}^N$ from prior distribution X_0
- For k > 0 do
  - **Resample** $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ into $\{x_{k-1}^{i*}, \frac{1}{N}\}_{i=1}^N$
  - **Predict** $x_k^i \sim p(x_k \mid x_{k-1} = x_{k-1}^{i*})$
  - **Reweight** $w_k^i = p(z_k \mid x_k = x_k^i)$
  - **Normalize** weights
  - **Estimate** $\hat{x}_k$ (for display)

Intermission

Questions?

Multiple Targets (MTT/MOT)

- Previous challenges
  - Full/partial occlusions
  - Entering/leaving the scene
  - …
- And in addition
  - Estimating the number of objects
  - Computationally tractable for multiple simultaneous targets
  - Interaction between objects

  - Many works on multiple single-target filters

BraMBLe: A Bayesian Multiple-Blob Tracker

M. Isard and J. MacCormick

Compaq Systems Research Center

ICCV 2001

Some slides taken from Qi Zhao
Some images taken from Isard and MacCormick

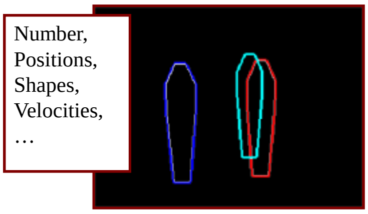
BraMBLE

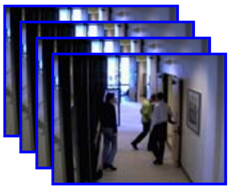
- First rigorous particle filter implementation with variable number of targets
- Posterior distribution is constructed over possible object configurations and number

- *Sensor*: single static camera
- *Tracking*: SIR particle filter
- *Performance*: real-time for 1-2 simultaneous objects

The BraMBLe posterior

$$p(\underbrace{x_k}_{\text{State at frame } k} \mid \underbrace{z_{1:k}}_{\text{Image Sequence}})$$





State space

- Hypothesis configuration:

$$X_k = (m_k, x_k^1, x_k^2, \ldots, x_k^m)$$

- Object configuration:

$$N_x = 1 + 13M_{\max}$$

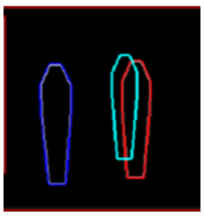
$$x_k^i = (\phi_k^i, \mathbf{X}_k^i, \mathbf{V}_k^i, \mathbf{S}_k^i)$$

identifier

shape
$\mathbf{S} = (w_f, w_w, w_s, w_h, h, \theta, \alpha_w, \alpha_s)$

velocity
$\mathbf{V} = (v_x, v_z)$

position
$\mathbf{X} = (x, z)$

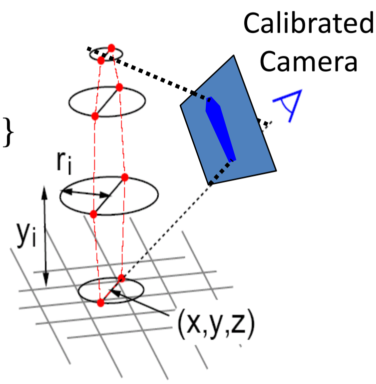


Object model

- A person is modeled as a *generalized-cylinder* with vertical axis in the world coords

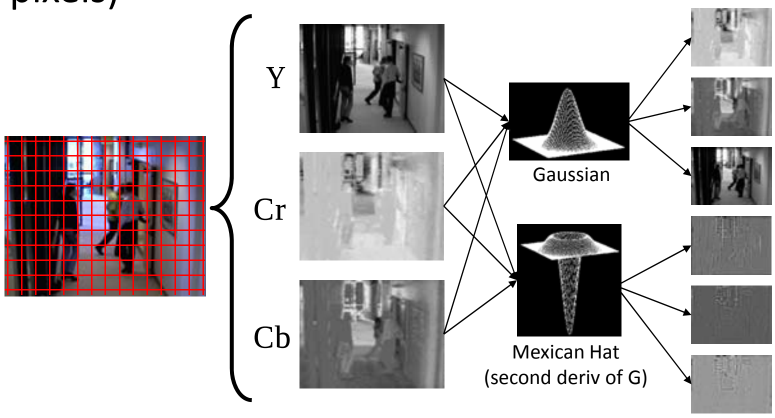
$$\mathbf{S}=(w_f, w_w, w_s, w_h, h, \theta, \alpha_w, \alpha_s)$$

$$(r_i, y_i)=\{(w_f,0),(w_w\theta,\alpha_w h),(w_s\theta,\alpha_s h),(w_h,h)\}$$



Observation likelihood $p(\mathbf{Z}_t \mid \mathbf{X}_t)$

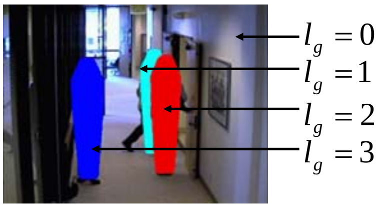
- Image overlaid with rectangular Grid (e.g. 5 pixels)



Observation likelihood $p(\mathbf{Z}_t \mid \mathbf{X}_t)$

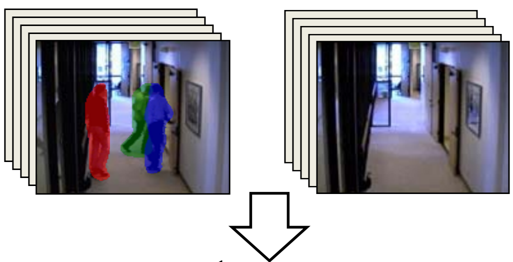
- The response values are assumed conditionally independent given X

$$\mathrm{p}(\mathbf{Z} \mid \mathbf{X}) = \prod_g p(z_g \mid \mathbf{X}) = \prod_g p(z_g \mid l_g)$$



Appearance models

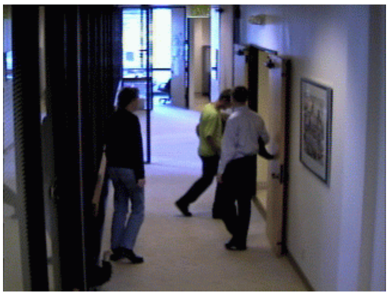
- GMMs for background and foreground are trained using kmeans

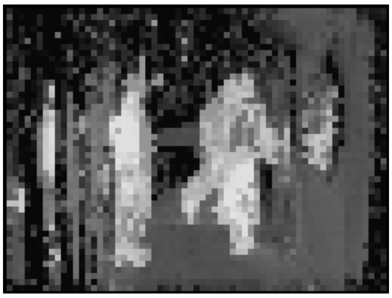


$$\mathrm{p}(z_g \mid l_g = 0) = \frac{1}{K} \sum_k N(\mu_g^k, \Sigma_g^k + \Delta_B) + \tau_B \qquad K = 4$$

$$\mathrm{p}(z_g \mid l_g \neq 0) = \frac{1}{K} \sum_k N(\mu_F^k, \Sigma_F^k) + \tau_F \qquad K = 16$$

Observation likelihood





$$\log\left(\frac{\mathrm{p}(z_g \mid l_g \neq 0)}{\mathrm{p}(z_g \mid l_g = 0)}\right)$$

System (prediction) model $p(X_t \mid X_{t-1})$

- The number of objects can change:
  - Each object has a constant probability λ_r to remain in the scene.
  - At each time step, there is constant probability λ_i that a new object will enter the scene.
- $X_{t-1}^{n'} = (m_{t-1}^{n'}, \widetilde{x}_{t-1}^{n',1}, \ldots) \rightarrow X_t^{n} = (m_t^{n}, \widetilde{x}_t^{n,1}, \ldots)$

Prediction function

Initialization function

1. **set** $m_t^n := 0$.
2. **for** $i = 1$ **to** $m_{t-1}^{n'}$:
   (a) **generate** r **distributed as** $U[0,1)$.
   (b) **if** $r < \lambda_r$ **set** $m_t^n := m_t^n + 1$ **and** $\tilde{x}^{n,m_t^n} := f(\tilde{x}_{t-1}^{n',i})$
3. **generate** r **distributed as** $U[0,1)$.
4. **if** $r < \lambda_i$ **set** $m_t^n := m_t^n + 1$ **and set** $\tilde{x}^{n,m_t^n} := g(t)$

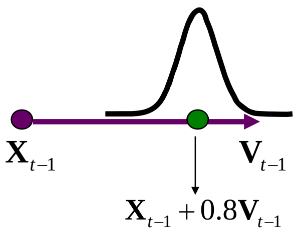
Figure 6: **The multi-object prediction algorithm**

Prediction function

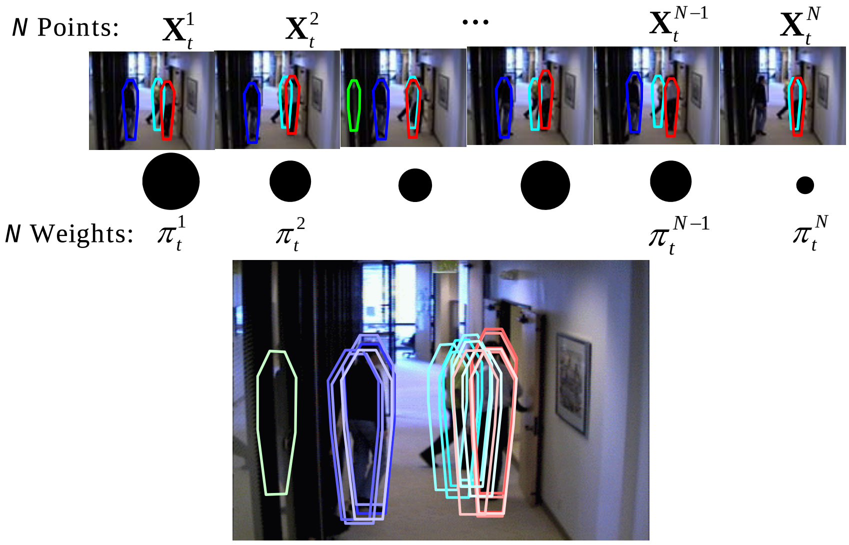
- Motion evolution: damped constant velocity
- Shape evolution: 1st order auto-regressive process model (ARP)

$$f(\phi, (\mathcal{X}, \mathcal{V}, \mathcal{S})^T) = (\phi, (\mathcal{X}', \mathcal{V}', \mathcal{S}')^T)$$

$$\begin{aligned}
\mathcal{X}' &= \mathcal{X} + \lambda_v \mathcal{V} + b_X \omega_X \\
\mathcal{V}' &= \lambda_v \mathcal{V} + b_X \omega_X \\
\mathcal{S}' &= A_S(\mathcal{S} - \overline{\mathcal{S}}) + B_S \omega_S \\
\overline{\mathcal{S}} &= (\mu_1, \ldots, \mu_8) \\
B_S &= \mathrm{diag}(\rho_1, \ldots, \rho_8) \\
A_S &= \mathrm{diag}(a_1, \ldots, a_8)
\end{aligned}$$



Particles



Estimate $\hat{X}_t$

- Denote $\mathbf{M}_t = \{\Phi_1, \ldots, \Phi_M\}$ the set of existing unique identifiers

Total probability that object Φ_i exists

(particle,target)

for $i = 1$ to M_t

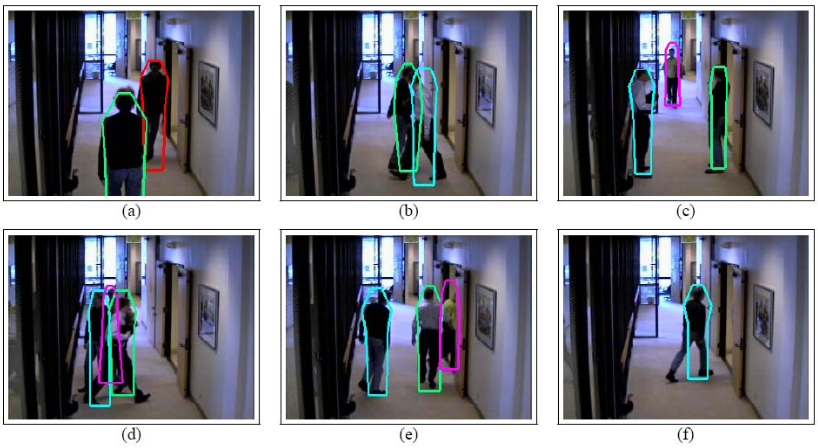
(a) **compute** $\mathcal{M}_t^{\Phi_i} = \{\boxed{(n,j)} : \phi_t^{n,j} = \Phi_i\}$.

(b) **compute** $\boxed{\Pi_t^{\Phi_i}} = \sum_{(n,j)\in\mathcal{M}_t^{\Phi_i}} \pi_t^n$.

(c) **if** $\Pi_t^{\Phi_i} > \lambda_A$ **estimate** $\hat{x}_t^{\Phi_i} = \sum_{(n,j)\in\mathcal{M}_t^{\Phi_i}} \pi_t^n s_t^{(n,j)} / \Pi_t^{\Phi_i}$.

Figure 7: **The estimation algorithm**

Results



- N=1000 particles
- initialization samples always generated

Results

- Single foreground model cannot distinguish between overlapping objects – causes id switches

Parameters

symbol	meaning	value
λ_r	object survival probability	0.99
λ_i	new object arrival probability	0.02
λ_d	object display threshold	0.8
δ_e	minimum physical separation between distinct objects (m)	0.5
δ_B	background likelihood additional covariance factor (grey-levels^2)	100
τ_B	background likelihood cutoff (grey-levels^{-6})	2.0×10^{-14}
τ_F	foreground likelihood cutoff (grey-levels^{-6})	3.0×10^{-13}
b_X	translation process noise (m)	0.11

	w_f	w_w	w_s	w_h	h	θ	α_w	α_s
mean μ_i	0.20m	0.22m	0.25m	0.08m	1.80m	0.75	0.60	0.83
steady-state standard deviation σ_i	0.03m	0.04m	0.04m	0.02m	0.05m	0.25	0.02	0.02
process noise ρ_i	0.003m	0.002m	0.002m	0.002m	0.003m	0.05	0.001	0.001

Summary

- The particle filters were shown to produce good approximations under relatively weak assumptions
  - can deal with nonlinearities
  - can deal with non-Gaussian noise
  - Multiple hypotheses
  - can be implemented in O(N)
  - easy to implement
  - **Adaptive focus on more probable regions of the state-space**

In practice

1. State (object) model
2. System (evolution) model
3. Measurement (likelihood) model
4. Initial (prior) state
5. State estimate (given the pdf)

6. PF specifics
   1. Importance density
   2. Resampling method

- Configurations for specific problems can be found in literature

Thank you!

References

- *Beyond the Kalman filter/* Ristic,Arulamplam,Gordon
  - Online tutorial: *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking/* Arulampalam et al 2002
- *Stochastic models, estimation and control/* Peter S. Maybeck
- *An Introduction to the Kalman Filter/* Greg Welch, Gary Bishop
- *Particle filters an overview/* Matthias Muhlich

Sequential derivation 1

- Suppose at time k-1, $\{x^i_{0:k-1}, w^i_{k-1}\}_{i=1}^N$ characterize $p(x_{0:k-1} \mid z_{1:k-1})$
- We receive new measurement z_k and need to approximate $p(x_{0:k} \mid z_{1:k})$ using new set of samples
- We choose q such that

$$q(x_{0:k} \mid z_{1:k}) = q(x_k \mid x_{0:k-1}, z_{1:k})q(x_{0:k-1} \mid z_{1:k-1})$$

And we can generate new particles

$$x^i_k \sim q(x_k \mid x^i_{0:k-1}, z_{1:k})$$

Sequential derivation 2

- For the weight update equation, it can be shown that

$$p(x_{0:k} \mid z_{1:k}) = \frac{p(z_k \mid x_k)p(x_k \mid x_{k-1})}{p(z_k \mid z_{1:k-1})} p(x_{0:k-1} \mid z_{1:k-1})$$
$$\propto p(z_k \mid x_k)p(x_k \mid x_{k-1})p(x_{0:k-1} \mid z_{1:k-1})$$

And so

$$w^i_k = \frac{p(x_{0:k} \mid z_{1:k})}{q(x_{0:k} \mid z_{1:k})} = \frac{p(z_k \mid x_k)p(x_k \mid x_{k-1})p(x_{0:k-1} \mid z_{1:k-1})}{q(x_k \mid x_{0:k-1}, z_{1:k})q(x_{0:k-1} \mid z_{1:k-1})}$$
$$= w^i_{k-1} \frac{p(z_k \mid x^i_k)p(x^i_k \mid x^i_{k-1})}{q(x^i_k \mid x^i_{0:k-1}, z_{1:k})}$$

Sequential derivation 3

- Further, if $q(x_k \mid x_{0:k-1}, z_{1:k}) = q(x_k \mid x_{k-1}, z_k)$
- Then the weights update rule becomes

$$w_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \qquad (3)$$

  (and need not store entire particle paths and full history of observations)
- Finally, the (filtered) posterior density is approximated by $p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i)$

Choice of importance density

- Choose q to minimize variance of weights
- Optimal choice: $q(x_k \mid x_{k-1}^i, z_k)_{opt} = p(x_k \mid x_{k-1}^i, z_k)$
  $\Rightarrow w_k^i \propto w_{k-1}^i p(z_k \mid x_{k-1}^i)$
  – Usually cannot sample from q_{opt} or solve for w_k^i (in some specific cases is works)
- Most commonly used (suboptimal) alternative: $q(x_k \mid x_{k-1}^i, z_k)_{opt} = p(x_k \mid x_{k-1}^i)$
  $\Rightarrow w_k^i \propto w_{k-1}^i p(z_k \mid x_k^i)$
  – i.e. the transitional prior

Generic PF

- Resampling reduces degeneracy, but new problems arise…

1. Limits parallelization
2. *Sample impoverishment*: particles with high weights are selected many times which leads to loss of diversity
   – if process noise is small – all particles tend to collapse to single point within few interations
   – Methods exist to counter this as well…