Visual Motion Analysis and Tracking
Part II

David J Fleet and Allan D Jepson

CIAR NCAP Summer School

July 12-16, 2005

Outline

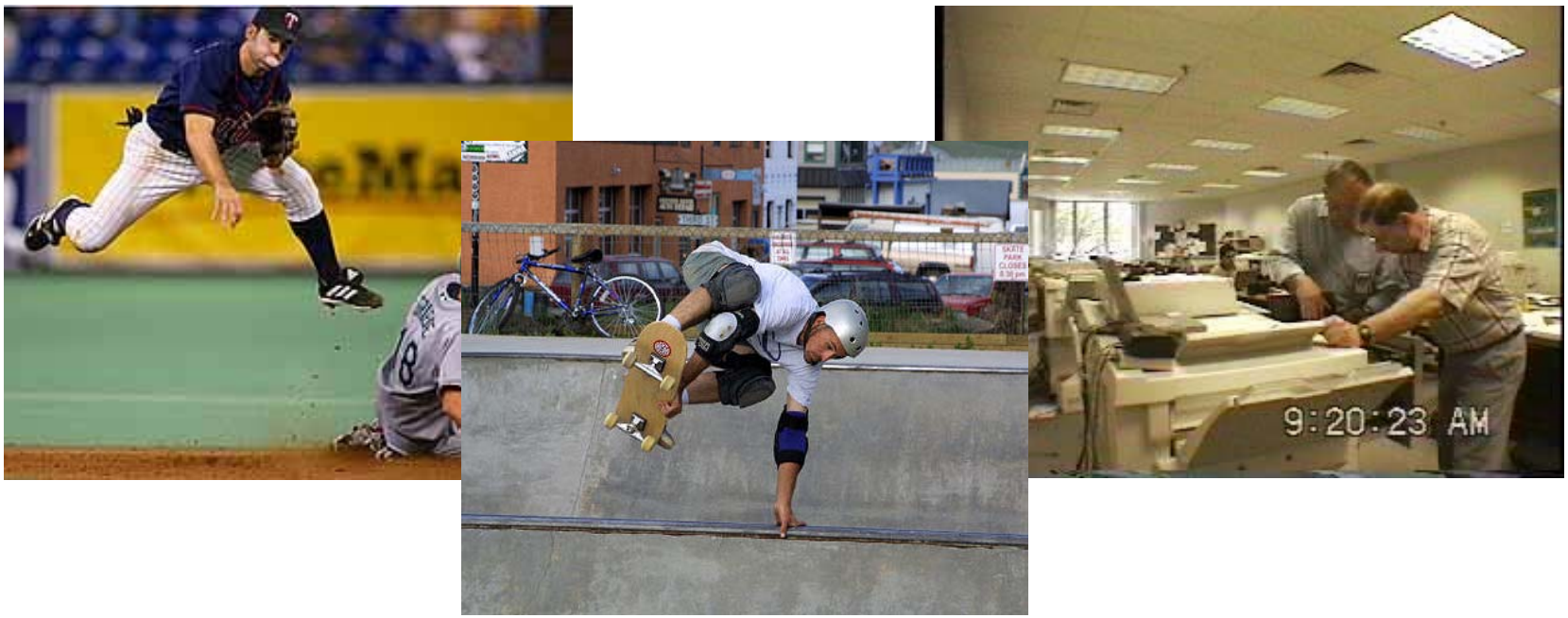
Optical Flow and Tracking:

- Optical flow estimation
  (robust, iterative refinement, coarse-fine)
- Motion-based tracking
  (EigenTracking, WSL, Features)

Model-Based Tracking:

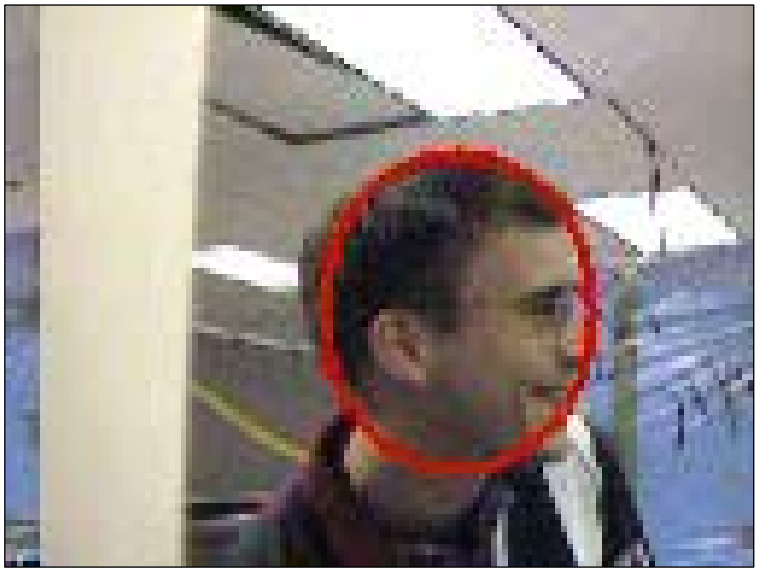
- Bayesian filtering / smoothing
- Kalman Filter
- Particle filters
- Lessoned learned
- ...

Why is tracking sometimes hard?



- complex nonlinear dynamics, with high dimensional object models
- complex appearance, and temporal appearance variation (deformable objects, shadows & lighting variations, clothing, …)
- impoverished information due to occlusion or scale
- multiple objects and background clutter

Ambiguity: Clutter, Occlusion & Multiple Objects

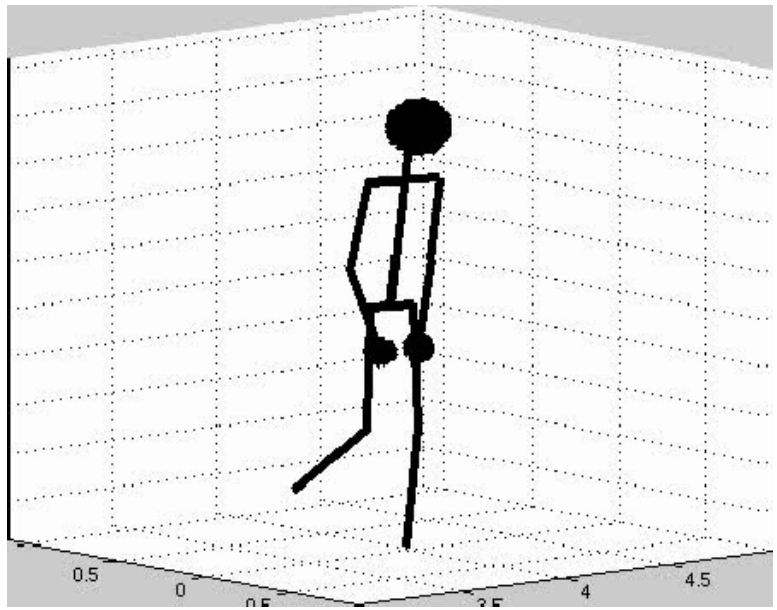


[Birchfield, "Elliptical head tracking using intensity gradients and color histograms." Proc CVPR, 1998]

[Khan, Balch, Dellaert, "A Rao-Blackwellized Particle Filter for EigenTracking" Proc CVPR, 2004]

(remember Bill's class)

Ambiguity: Poorly Constrained Models



[Sidenbladh, Black & Fleet, "3D people tracking using particle filtering." Proc ECCV, 2000]

Probabilistic Formulation

- State: n-vector containing variables to be estimated: $\vec{\mathbf{x}}_t$
  - continuous variables [eg., position, velocity, shape, size, …]
  - discrete state variables [eg., # objects, gender, activity, …]
  - state history: $\vec{\mathbf{x}}_{1:t} = (\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_t)$
- Observations: data from which we estimate state: $\vec{\mathbf{z}}_t = f(\vec{\mathbf{x}}_t)$
  - observation history: $\vec{\mathbf{z}}_{1:t} = (\vec{\mathbf{z}}_1, \ldots, \vec{\mathbf{z}}_t)$
- Posterior distribution over states conditioned on observations

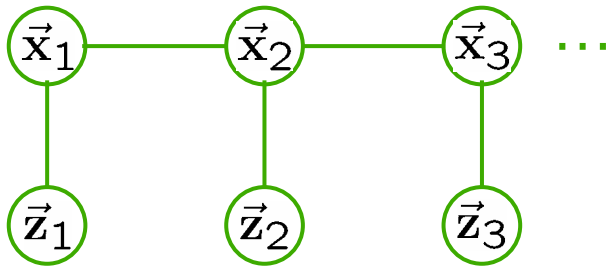
$$p(\vec{\mathbf{x}}_{1:t} \mid \vec{\mathbf{z}}_{1:t})$$

- Filtering distribution: marginal posterior at current time

$$p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t}) = \int_{\vec{\mathbf{x}}_1} \cdots \int_{\vec{\mathbf{x}}_{t-1}} p(\vec{\mathbf{x}}_{1:t} \mid \vec{\mathbf{z}}_{1:t})$$

Model Assumptions

- Graphical model:



- 1st -order Markov model for state dynamics:

$$p(\vec{x}_t \mid \vec{x}_{1:t-1}) = p(\vec{x}_t \mid \vec{x}_{t-1})$$

so

$$p(\vec{x}_{1:t}) = \left(\prod_{j=2}^{t} p(\vec{x}_j \mid \vec{x}_{j-1}) \right) p(\vec{x}_1)$$

- Conditional independence of observations

$$p(\vec{z}_{1:t} \mid \vec{x}_{1:t}) = p(\vec{z}_t \mid \vec{x}_t)\, p(\vec{z}_{1:t-1} \mid \vec{x}_{1:t-1})$$
$$= \prod_{\tau=1}^{t} p(\vec{z}_\tau \mid \vec{x}_\tau)$$

Filtering Equations

Filtering distribution:

$$p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t}) = \int_{\vec{\mathbf{x}}_1} \cdots \int_{\vec{\mathbf{x}}_{t-1}} p(\vec{\mathbf{x}}_{1:t} \mid \vec{\mathbf{z}}_{1:t})$$

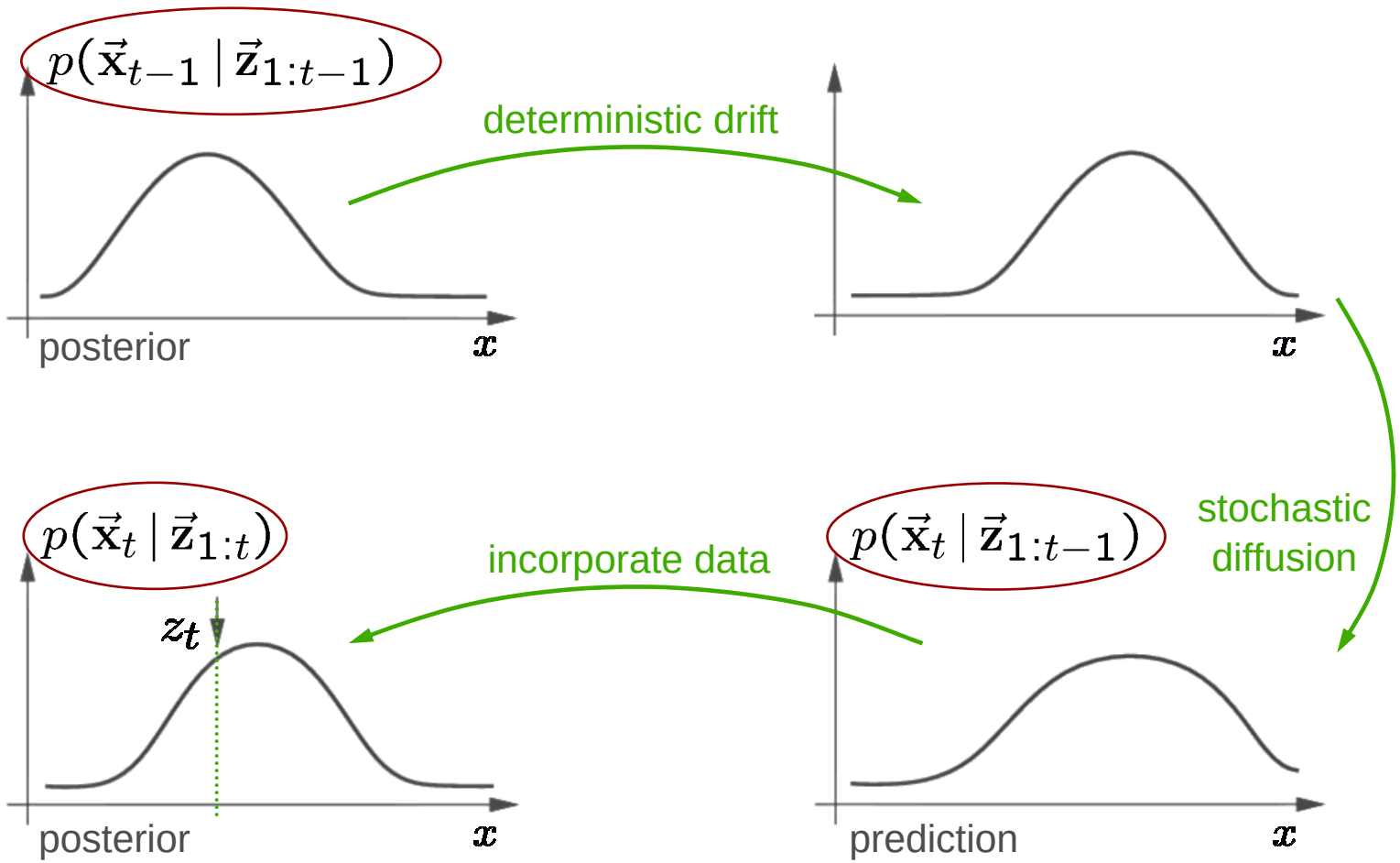
$$= c\, p(\vec{\mathbf{z}}_t \mid \vec{\mathbf{x}}_t)\; p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t-1})$$

likelihood prediction

Prediction distribution (temporal prior):

$$p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t-1}) = \int_{\vec{\mathbf{x}}_{t-1}} p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{x}}_{t-1})\, p(\vec{\mathbf{x}}_{t-1} \mid \vec{\mathbf{z}}_{1:t-1})$$

Recursive Filtering



Filtering and Smoothing

Similar recursive computation backwards in time:

$$p(\vec{\mathbf{x}}_\tau \mid \vec{\mathbf{z}}_{\tau:t}) = c\, p(\vec{\mathbf{z}}_\tau \mid \vec{\mathbf{x}}_\tau) \int_{\vec{\mathbf{x}}_{\tau+1}} p(\vec{\mathbf{x}}_\tau \mid \vec{\mathbf{x}}_{\tau+1})\, p(\vec{\mathbf{x}}_{\tau+1} \mid \vec{\mathbf{z}}_{\tau+1:t})$$

$$= c\, p(\vec{\mathbf{z}}_\tau \mid \vec{\mathbf{x}}_\tau)\, p(\vec{\mathbf{x}}_\tau \mid \vec{\mathbf{z}}_{\tau+1:t})$$

Smoothing: optimal computation given entire sequence

$$p(\vec{\mathbf{x}}_\tau \mid \vec{\mathbf{z}}_{1:t}) = c\, p(\vec{\mathbf{z}}_\tau \mid \vec{\mathbf{x}}_\tau)\, p(\vec{\mathbf{x}}_\tau \mid \vec{\mathbf{z}}_{1:\tau-1})\, p(\vec{\mathbf{x}}_\tau \mid \vec{\mathbf{z}}_{\tau+1:t})$$

current evidence

prediction from past data

prediction from future data

[Belief Propagation: combining predictions (or messages) from neighbors with local evidence to compute local estimates of state]

Where do dynamics & observation eqns come from?

Make ‘em up

Derive ‘em (from models you made up)

Learn ‘em (with models you made up)

Kalman Filter

Assume linearity & Gaussianity for observation and dynamics eqns:

$$\vec{\mathbf{x}}_t = A\,\vec{\mathbf{x}}_{t-1} + \vec{\eta}_d \qquad \vec{\eta}_d \sim \mathcal{N}(0, C_d)$$

$$\vec{\mathbf{z}}_t = M\,\vec{\mathbf{x}}_t + \vec{\eta}_m \qquad \vec{\eta}_m \sim \mathcal{N}(0, C_m)$$

The conditional Markov and likelihood distributions become:

$$p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{x}}_{t-1}) = G(\vec{\mathbf{x}}_t - A\vec{\mathbf{x}}_{t-1}, C_d)$$

$$p(\vec{\mathbf{z}}_t \,|\, \vec{\mathbf{x}}_t) = G(\vec{\mathbf{z}}_t - M\vec{\mathbf{x}}_t, C_m)$$

Key Result: Prediction and filtering distributions are Gaussian, so they may be represented by sufficient statistics:

$$p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{z}}_{1:t-1}) = \int_{\vec{\mathbf{x}}_{t-1}} p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{x}}_{t-1})\, p(\vec{\mathbf{x}}_{t-1} \,|\, \vec{\mathbf{z}}_{1:t-1}) \sim \mathcal{N}(\vec{\mathbf{x}}_t^-, C_t^-)$$

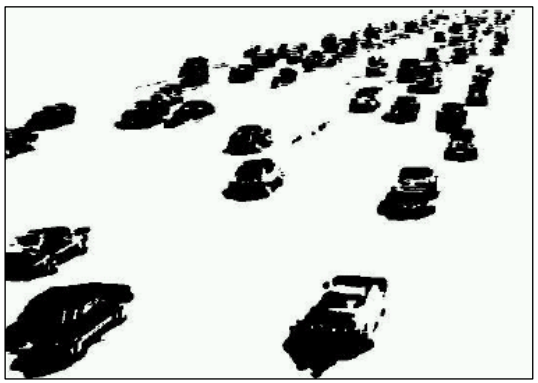
$$p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{z}}_{1:t}) = c\, p(\vec{\mathbf{z}}_t \,|\, \vec{\mathbf{x}}_t)\, p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{z}}_{1:t-1}) \sim \mathcal{N}(\vec{\mathbf{x}}_t^+, C_t^+)$$

Kalman Filter

First well-known uses in computer vision:

- Road following by tracking lane markers
  [Dickmanns & Graefe, "Dynamic monocular machine vision." Machine Vision and Applications, 1988]
- Rigid structure from feature tracks under perspective projection
  [Broida et al., "Recursive estimation of 3D motion from monocular image sequence. IEEE Trans. Aerosp. & Elec. Sys., 1990]

E.g., Vehicle Tracking

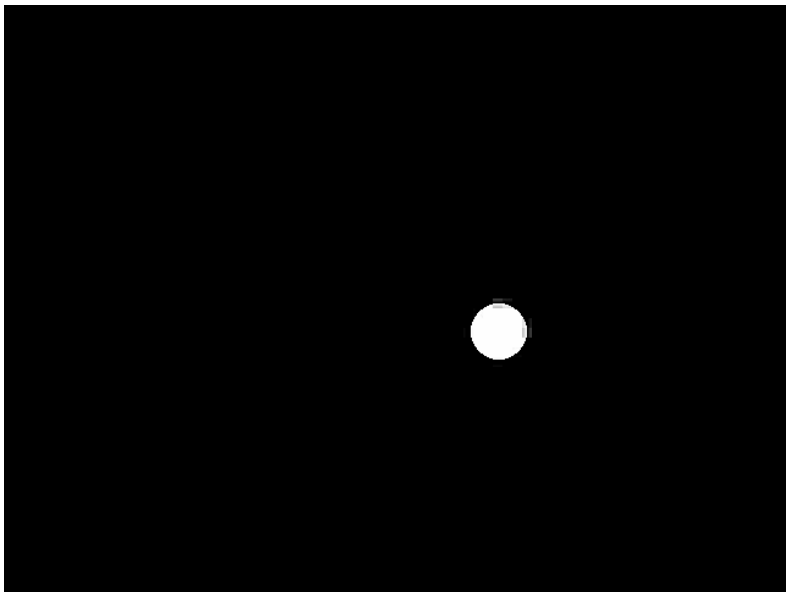


[Koller, Weber & Malik, "Robust multiple car tracking with occlusion reasoning." Proc ECCV,1994]

Problems Remain: Multi-Modal Likelihoods

Measurement clutter and occlusion in natural images often cause likelihood functions to have multiple, local maxima.

Problems Remain: Non-Linear Dynamics



[WSL Tracker]

- Animate objects and interactions between object often produce complex nonlinear dynamics
- Non-linear dynamics does not preserve simple distributions

Short Cuts?

Hill Climbing: works great when you are always close enough to the optimal (ML or MAP) state. This may be suitable for many apps, but tracking diagnostics and restarts may be required.

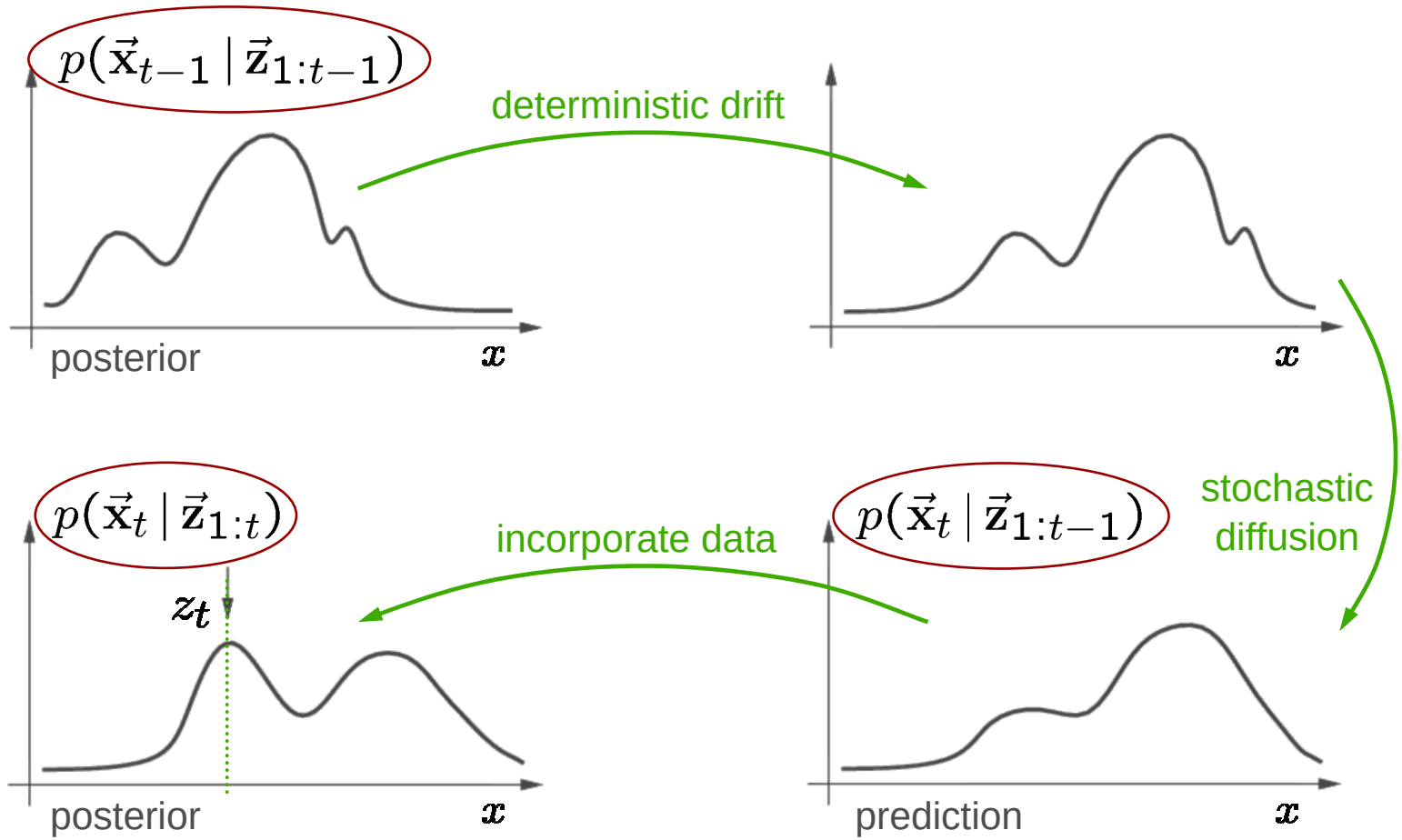
Data Association: Select a subset of "measurements" from the entire collection of observations, i.e., those

- that are related to the object being tracked,
- for which simpler inference applies (ie Kalman filter)

May work when it's easy to separate foreground from background, otherwise both foreground and background should be modeled.

Note: In principle, the likelihood $p(\vec{\mathbf{z}}_t \mid \vec{\mathbf{x}}_t)$ should account for the same observations $\vec{\mathbf{z}}_t$ when comparing different states $\vec{\mathbf{x}}_t$.

Bayesian Filtering



Non-Parametric Approximate Inference

Approximate the filtering distribution using point samples:

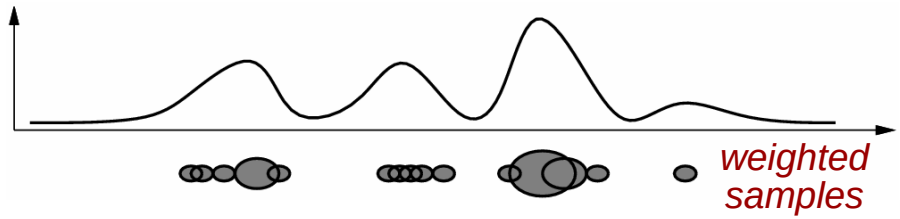
- By drawing a set of random samples from the filtering distribution, we could use samples statistics to approximate expectations

  Let $\mathcal{S} = \{\vec{\mathbf{x}}^{(j)}\}$ be a set of N fair samples from distribution $\mathcal{P}(\vec{\mathbf{x}})$, then for functions $f(\vec{\mathbf{x}})$

$$E_{\mathcal{S}}\left[f(\vec{\mathbf{x}})\right] \equiv \frac{1}{N}\sum_{j=1}^{N} f(\vec{\mathbf{x}}^{(j)}) \overset{N\to\infty}{\longrightarrow} E_{\mathcal{P}}\left[f(\vec{\mathbf{x}})\right]$$

  Problem: we don't know how to draw samples from $p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t})$

Importance Sampling



Weighted sample set $\mathcal{S} = \{\vec{\mathbf{x}}^{(j)}, w^{(j)}\}$

- draw samples $\vec{\mathbf{x}}^{(j)}$ from a *proposal distribution* $\mathcal{Q}(\vec{\mathbf{x}})$, with weights $w^{(j)} = w(\mathbf{x}^{(j)})$, then

$$E_{\mathcal{S}}\left[f(\vec{\mathbf{x}})\right] \equiv \sum_{j=1}^{N} w^{(j)} f(\vec{\mathbf{x}}^{(j)}) \overset{N\to\infty}{\longrightarrow} E_{\mathcal{Q}}\left[w(\vec{\mathbf{x}})\, f(\vec{\mathbf{x}})\right]$$

- If $w(\vec{\mathbf{x}}) = \mathcal{P}(\vec{\mathbf{x}})/\mathcal{Q}(\vec{\mathbf{x}})$ then weighted sample statistics approximate expectations under $\mathcal{P}(\vec{\mathbf{x}})$, i.e.,

$$\begin{aligned} E_{\mathcal{Q}}\left[w(\vec{\mathbf{x}})\, f(\vec{\mathbf{x}})\right] &= \int w(\vec{\mathbf{x}})\, f(\vec{\mathbf{x}})\, \mathcal{Q}(\vec{\mathbf{x}})\, d\vec{\mathbf{x}} \\ &= \int f(\vec{\mathbf{x}})\, \mathcal{P}(\vec{\mathbf{x}})\, d\vec{\mathbf{x}} \\ &= E_{\mathcal{P}}[f(\vec{\mathbf{x}})] \end{aligned}$$

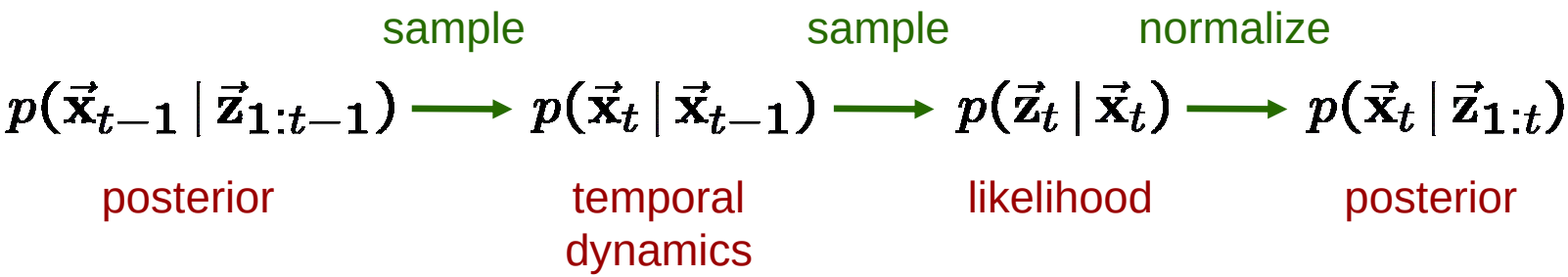
Particle Filters

Sequential Monte Carlo methods draw weighted samples to approximate the filtering distribution:

$$p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t}) \;=\; c\; p(\vec{\mathbf{z}}_t \mid \vec{\mathbf{x}}_t)\; p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t-1})$$

Simple particle filter (with resampling at each time step):

- draw samples from the prediction distribution $p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t-1})$
- weights are proportional to the ratio of posterior and prediction distributions, i.e. the normalized likelihood $c\, p(\vec{\mathbf{z}}_t \mid \vec{\mathbf{x}}_t)$



[Gordon et al '93; Isard & Blake '98; Liu & Chen '98, ...]

Sampling the Prediction Distribution

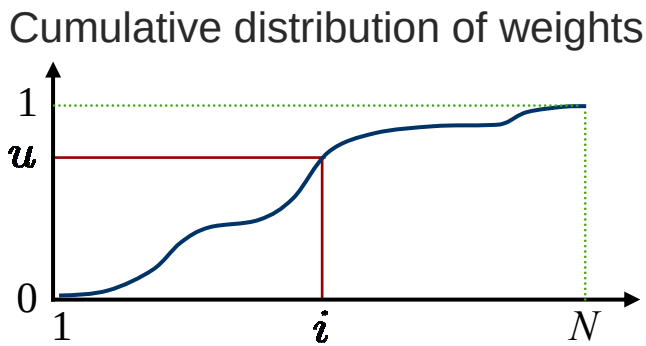
Given a weighted sample set $S_{t-1} = \{\vec{\mathbf{x}}_{t-1}^{(j)}, w_{t-1}^{(j)}\}$, the prediction distribution is a linear mixture model

$$p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{z}}_{1:t-1}) = \sum_{j=1}^{N} w^{(j)} p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{x}}_{t-1}^{(j)})$$

To draw a sample from it:

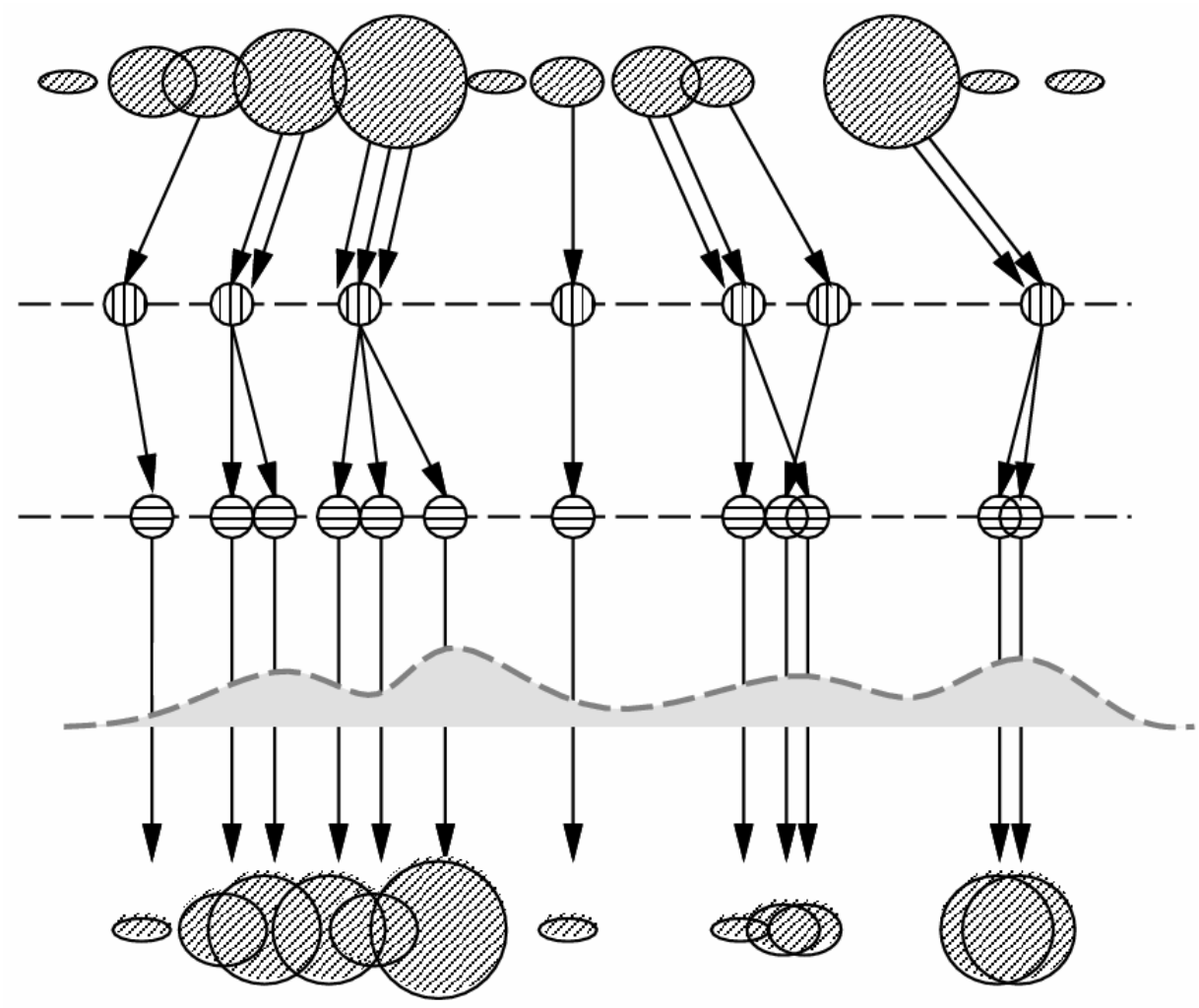
- sample a component of the mixture by the treating weights as mixing probabilities

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



- then sample from the associated dynamics pdf $p(\vec{\mathbf{x}}_t \mid \vec{\mathbf{x}}_{t-1}^{(i)})$

Particle Filters



weighted sample set

re-sample & drift

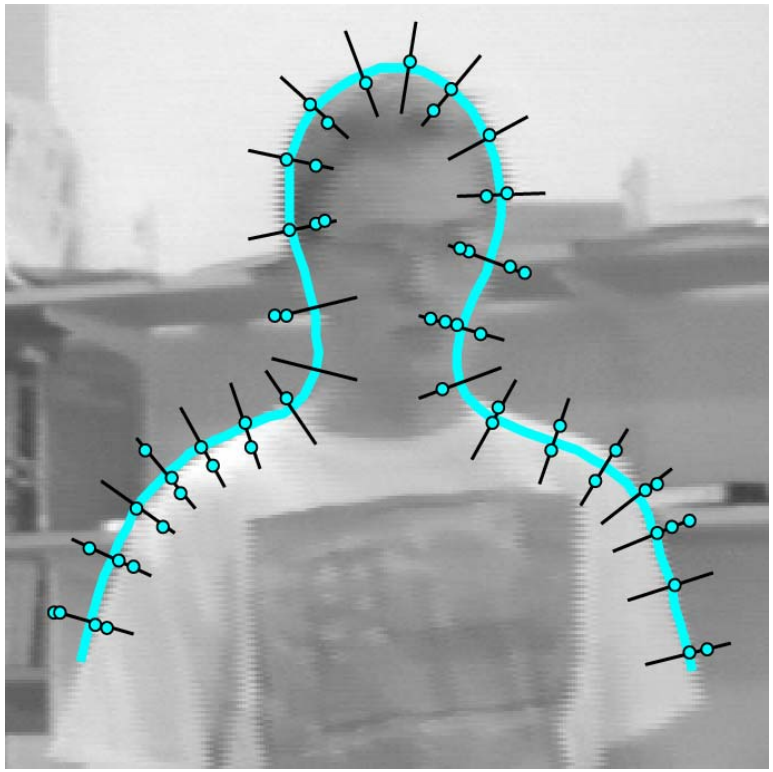
diffuse & re-sample

compute likelihoods

weighted sample set

[Isard and Blake, '98]

2D Contour Tracking



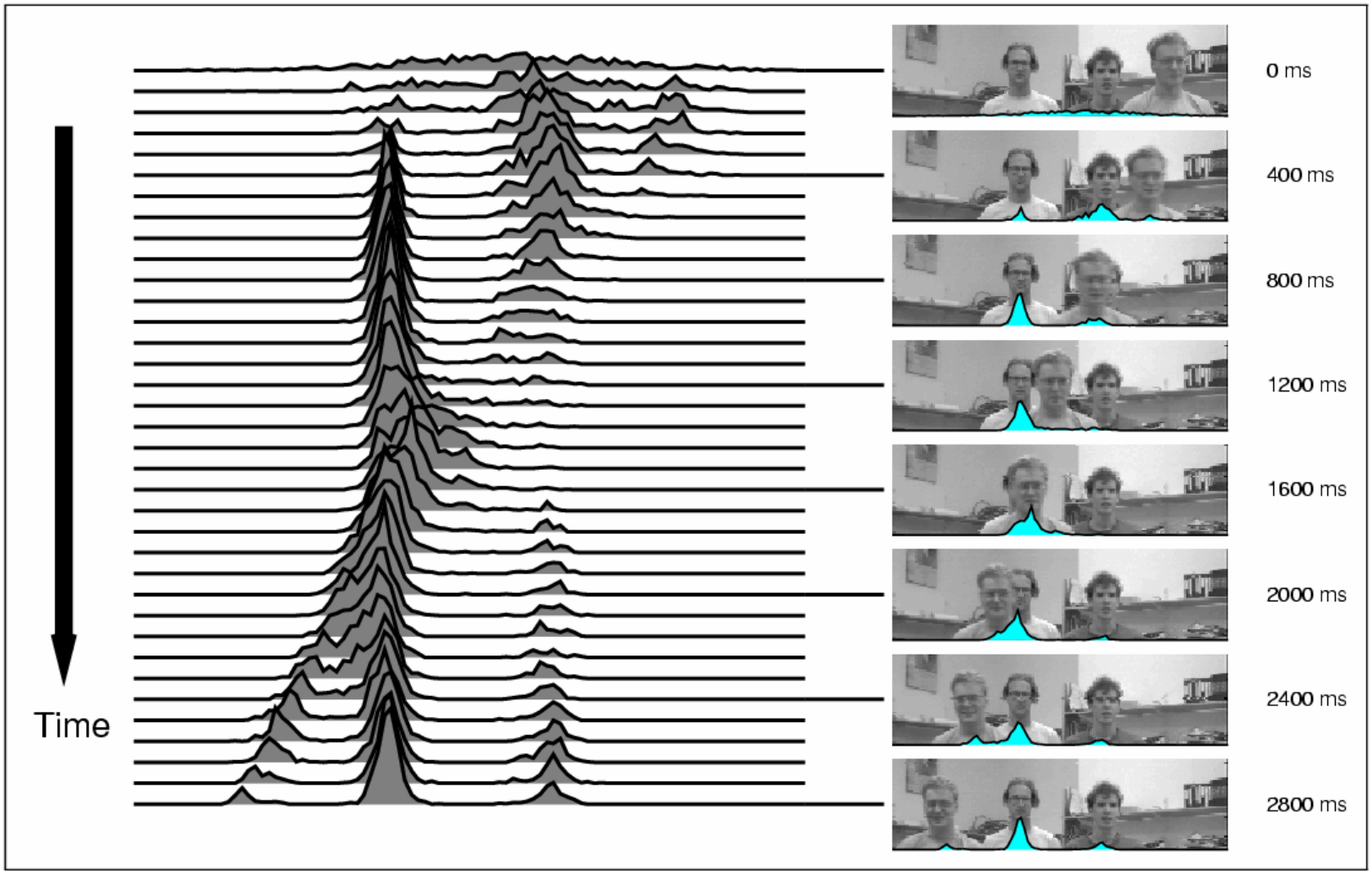
State: 6 parameters of affine deformation.

Measurements: edge strength perpendicular to contour

Dynamics: 2nd-order Markov model (often learned)

[Isard & Blake, "Condensation - conditional density propagation for visual tracking." IJCV, 1998]

2D Contour Tracking



(6 DOF state space, 1000 particles)

2D Contour Tracking

(6D affine state, 100 particles)

(6D affine state, 1200 particles)

[Isard & Blake, "Condensation - conditional density propagation for visual tracking." IJCV, 1998]

2.1D Blob Tracking

State: number of people, their positions/velocities on ground plane, and simple shape models (10 dimensions / person)

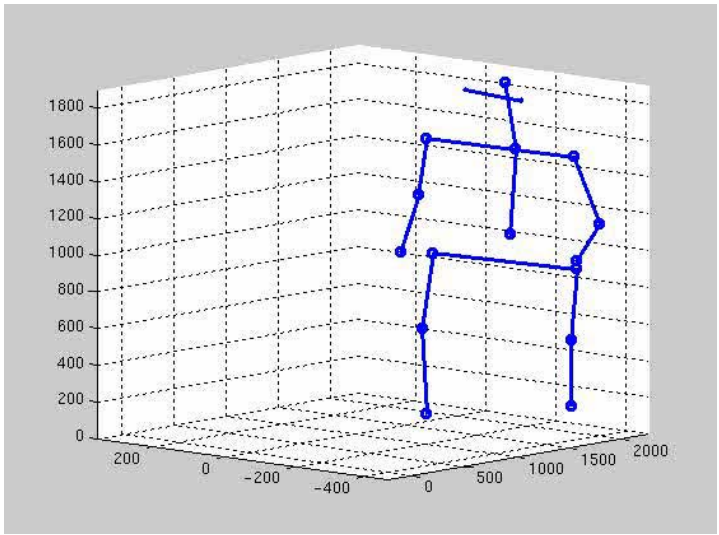
Appearance: filter response histograms for background, and for people

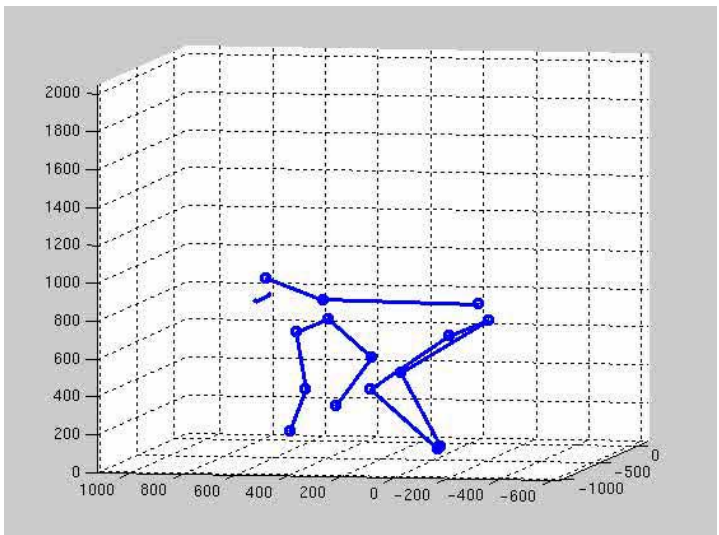
Dynamics: damped 2nd-order model for position/velocity, 1st-order for shape model

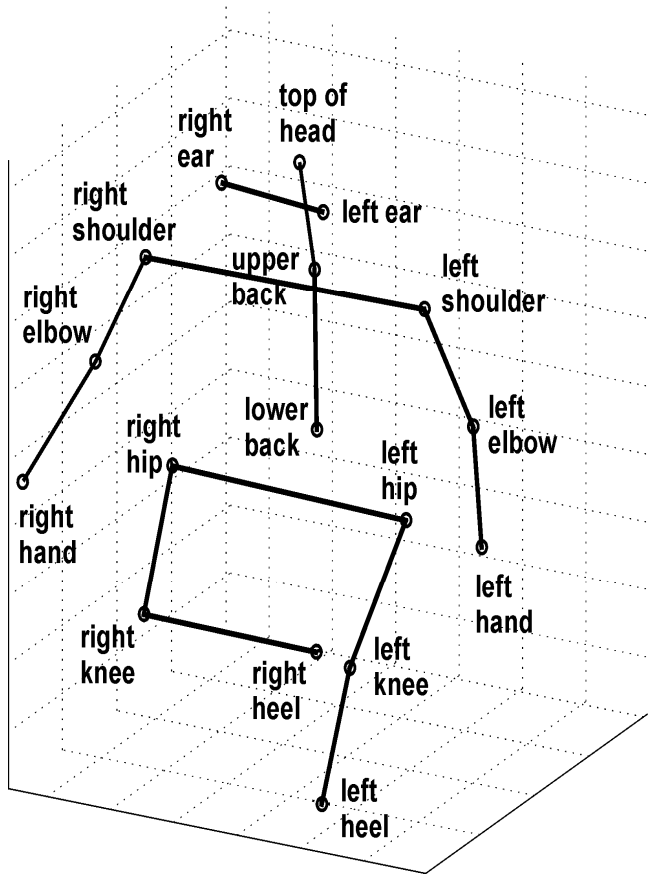
(1 person required ~500 particles, 2-3 people required >10,000 particles)

[Isard and MacCormick, "Bramble: A Multiple Blob Bayesian Tracker." Proc ICCV, 2001]

Monocular 3D People Tracking







3D Kinematic Model

(28D state, with 22 joint angles, 6 global DOFs)

Likelihood and Dynamics

Given the state, $\vec{\mathbf{s}}$, and the articulated model, the 3D marker positions $\vec{\mathbf{X}}_j$ onto the 2D image plane:

$$\mathbf{d}_j(\vec{\mathbf{s}}) = T_j(\vec{\mathbf{X}}_j; \vec{\mathbf{s}})$$

Observation model:

$$\hat{\mathbf{d}}_j \;=\; \mathbf{d}_j + \eta_j \;, \quad \eta_j \sim \mathcal{N}(0; \sigma_m^2 \mathbf{I}_2)$$

Likelihood of observed 2D locations, $\mathbf{D} = \{\hat{\mathbf{d}}_\mathbf{j}\}$:

$$p(\mathbf{D} \,|\, \mathbf{s}) \;\propto\; \exp\left(-\frac{1}{2\sigma_\mathbf{m}^2} \sum_\mathbf{j} || \,\hat{\mathbf{d}}_\mathbf{j} - \mathbf{d}_\mathbf{j}(\mathbf{s}) \,||^2 \right)$$

Smooth dynamics:

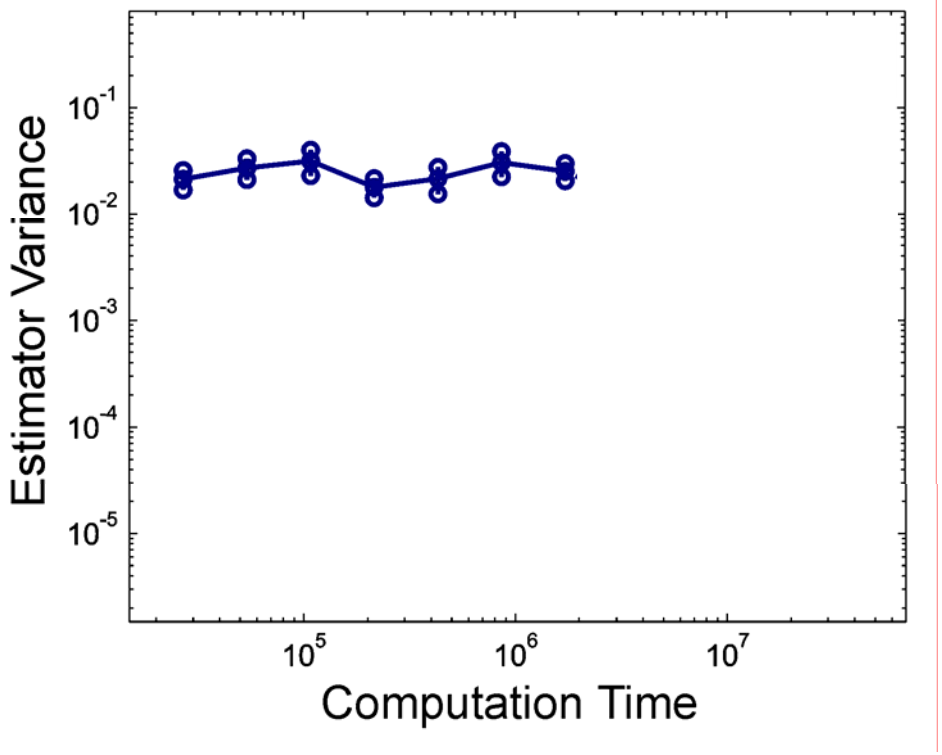
$$\vec{\mathbf{s}}_t \;=\; \vec{\mathbf{s}}_{t-1} + \epsilon_t$$

where ϵ_t is isotropic Gaussian for translational & angular variables

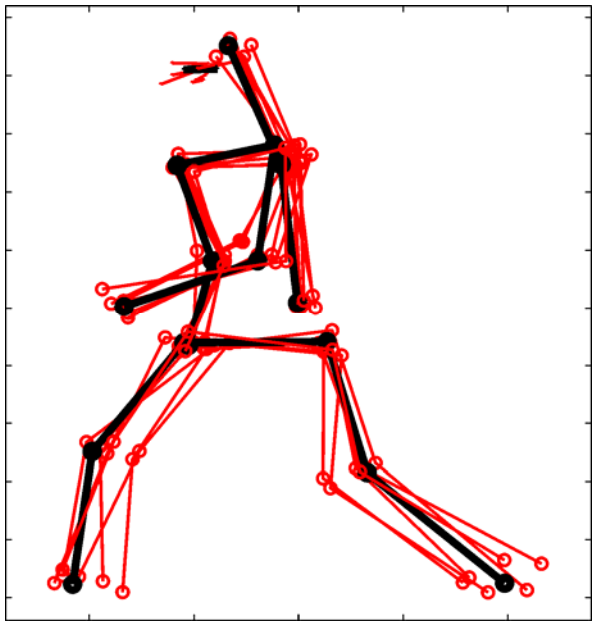
Experimental Evaluation

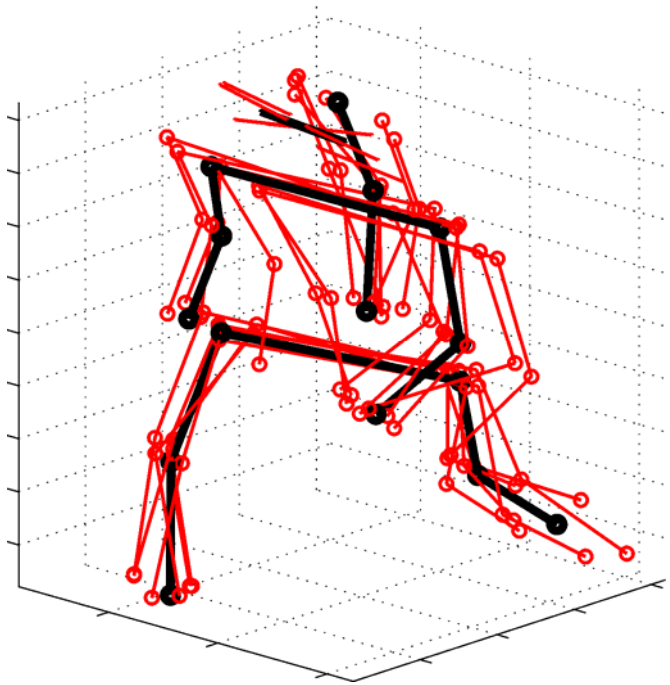
Estimator Variance:

- multiple runs with independent noise & sampling
- variance measured as MSE from ground truth (from MCMC)



Experimental Evaluation





Black: ground truth (at frame 10)

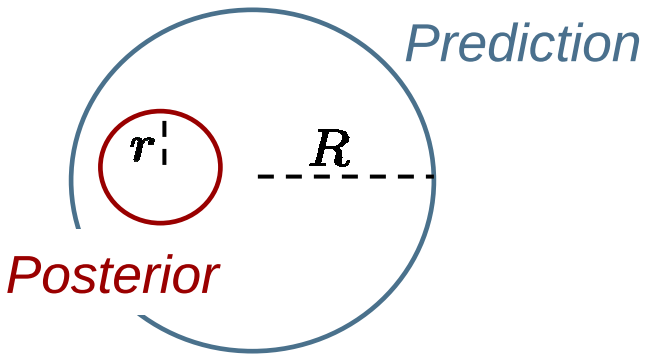
Red: mean states from 6 random trials

Problem: Exponential Numbers of Samples?

Number of samples needed depends on the effective volumes (entropies) of the prediction and posterior distributions.

- with ramdom sampling from the prediction density, the number of particles must grow exponentially in state dimension for samples to fall states with high posterior

  E.g., for D-dim spheres, with radii R and r,



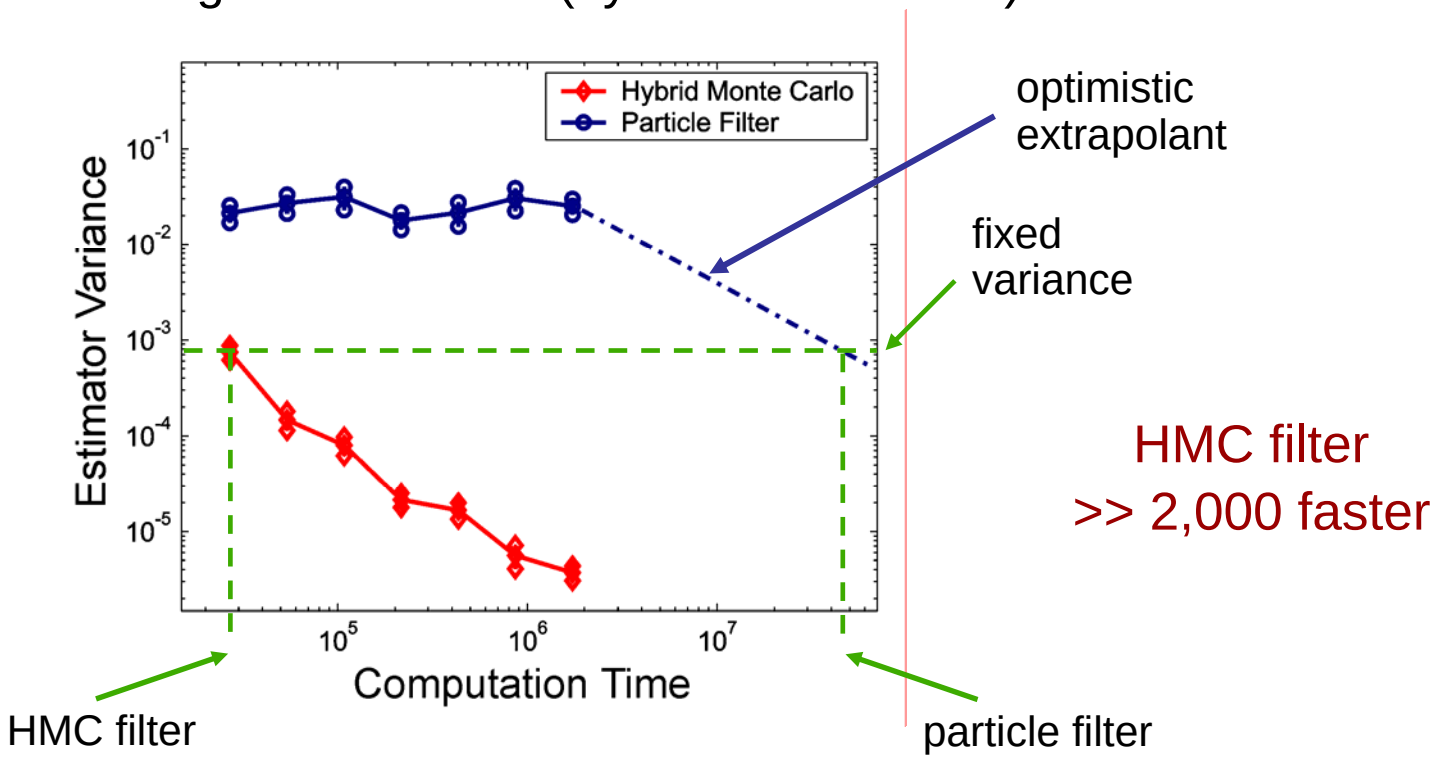
$$N > \left(\frac{R}{r}\right)^D$$

- effective number of 'independent' samples: $N_e = 1/\sum_j (w^{(j)})^2$

Hybrid Monte Carlo Filter

Particles can exploit information obtained by existing particles

- initialize a set of particles from a particle filter
- select a subset from which to begin MCMC simulation with stochastic gradient ascent (hybrid Monte Carlo)

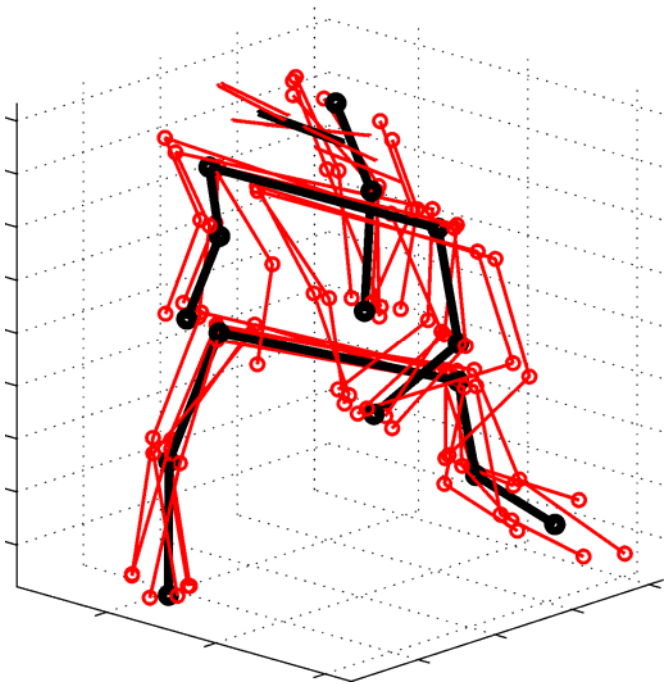


HMC filter
>> 2,000 faster

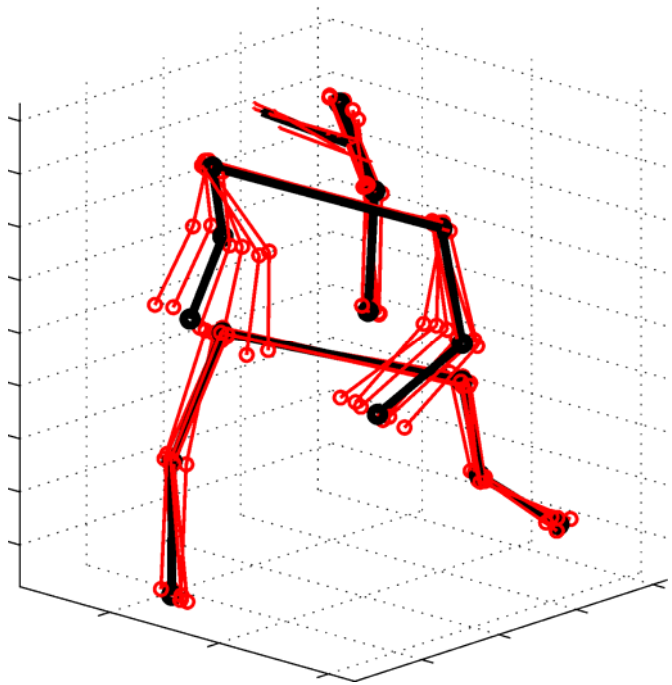
[Choo & Fleet, "People Tracking Using Hybrid Monte Carlo Filtering.", Proc IEEE ICCV, 2001]

Experimental Results

Particle Filter



Hybrid MC Filter



Black: Ground truth (at frame 10)

Red: Mean state from 6 random trials

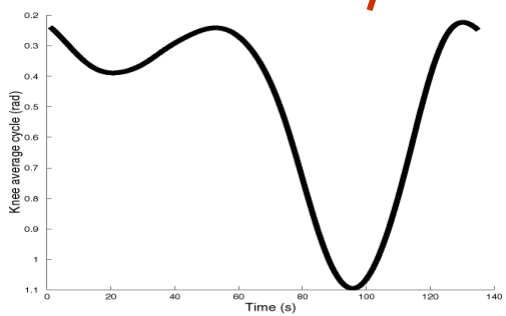
Activity-Specific Dynamics

Subspace Walking Model:

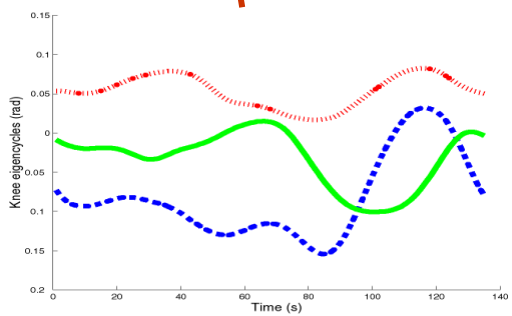
Joint angle curves are segmented and scaled to yield data curves $\{\theta_i(\psi)\}_{i=1}^{N}$ where $\psi \in [0, 1)$ is the phase the walking cycle.

PCA provides a linear basis for the joint angles at phase ψ_t :

$$\vec{\phi}(\vec{\mathbf{c}};\, \psi) \;=\; \vec{\mu}(\psi) \;+\; \sum_{k=1}^{B} c_k\, \vec{\mathbf{b}}_k(\psi)$$



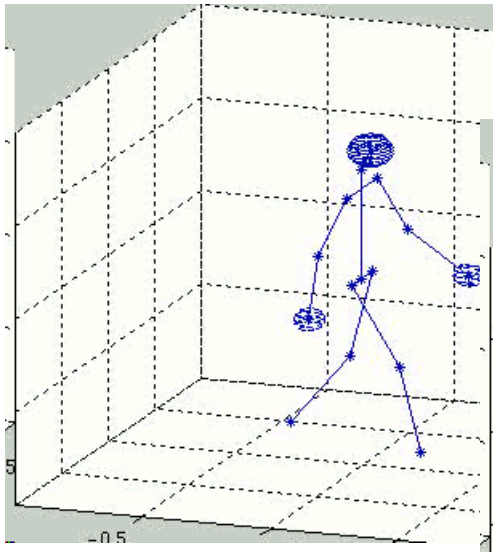
mean knee angle



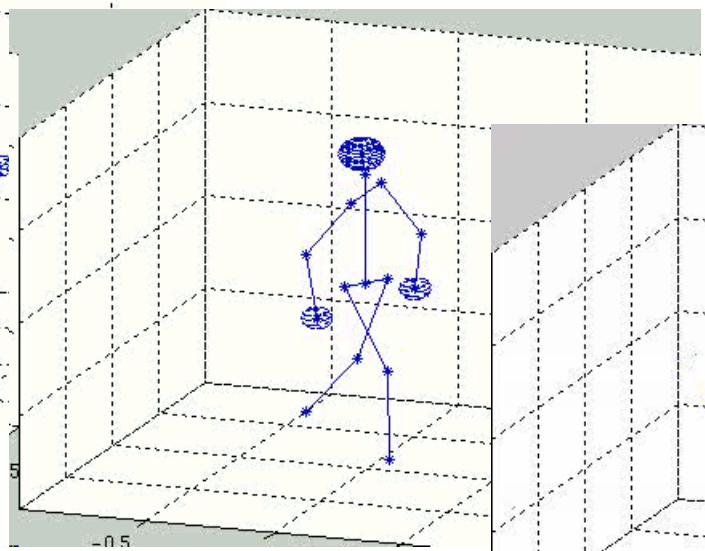
knee angle basis

[Sidenbladh, Black & Fleet, "Stochastic tracking of 3D human figures using 2D image motion." Proc ECCV, 2000]

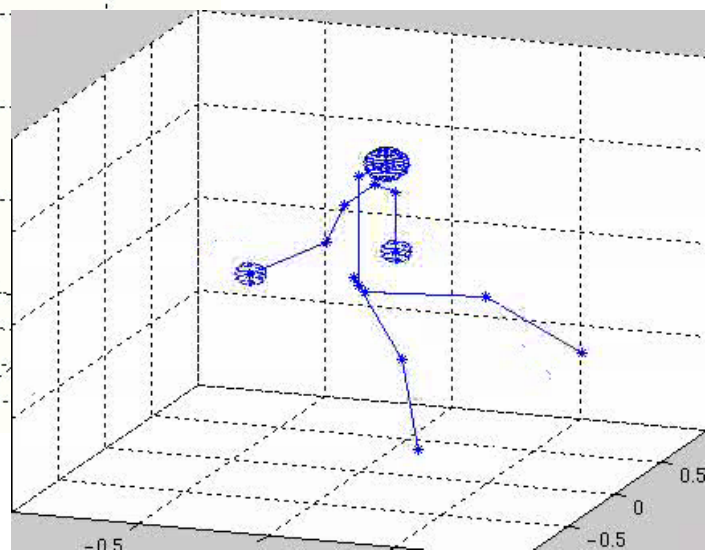
Temporal Dynamics: Walking Model



mean walking



mean walking plus
moderate noise



mean walking plus
large noise

Motion Likelihood

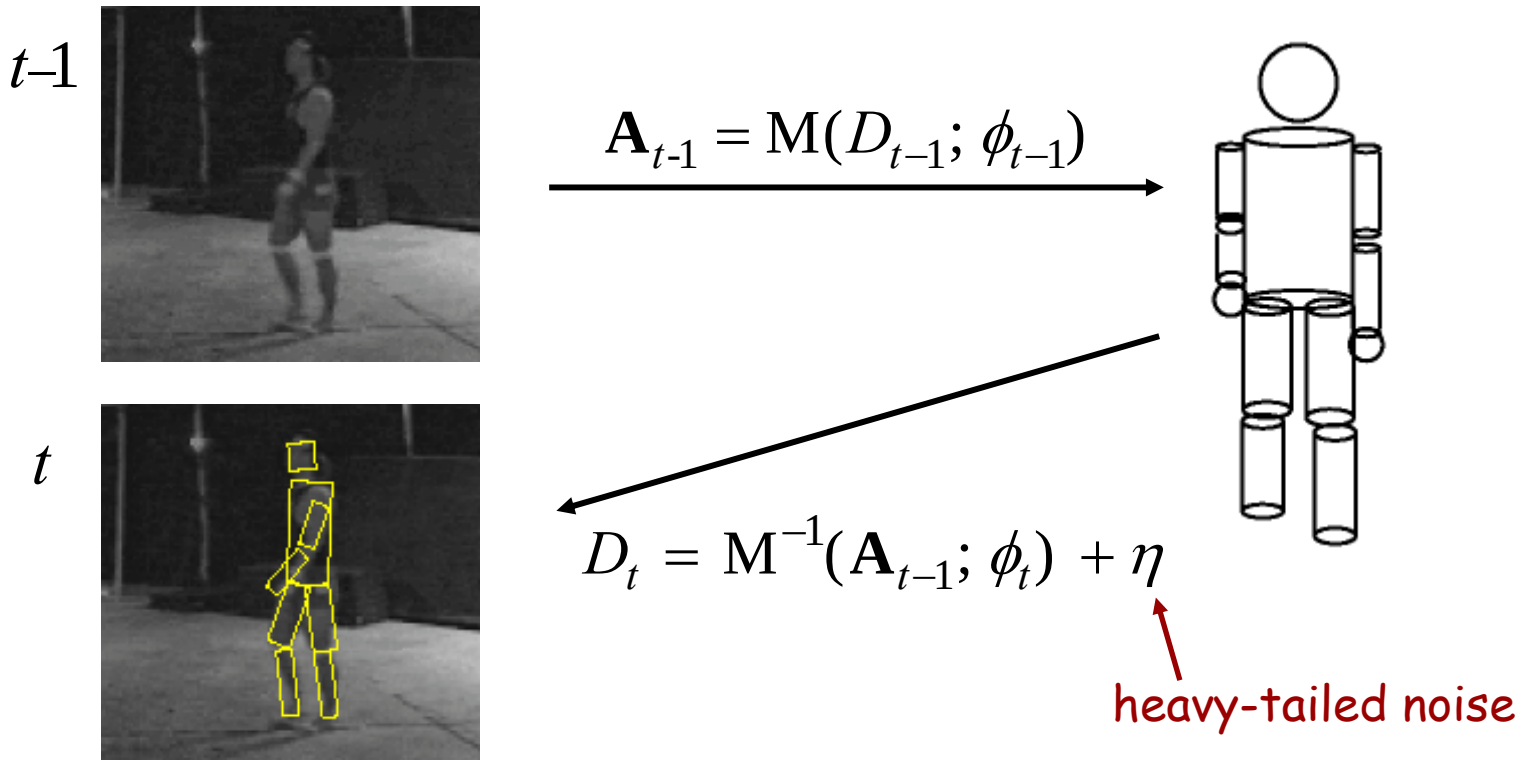


Image formation: perspective projection of texture-mapped 3D shape (assumes brightness constancy and additive noise)

Temporal Dynamics: Walking Model

Parameters of the generative model at time t :

$$\left(\vec{\mathbf{c}}_t,\ \psi_t,\ \nu_t,\ \tau_t^g,\ \theta_t^g\right)$$

5 basis coefficients — phase — speed — global pose

Smooth temporal dynamics (Gaussian process noise):

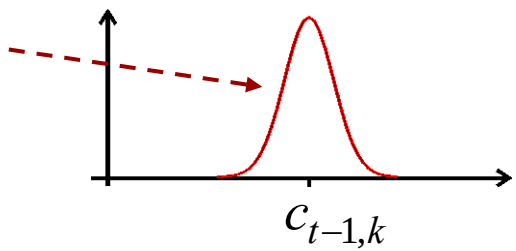
$$p(c_{t,k} \mid c_{t-1,k}) = G(c_{t,k} - c_{t-1,k}, \sigma_k^c)$$

$$p(v_t \mid v_{t-1}) = G(v_t - v_{t-1}, \sigma^v)$$

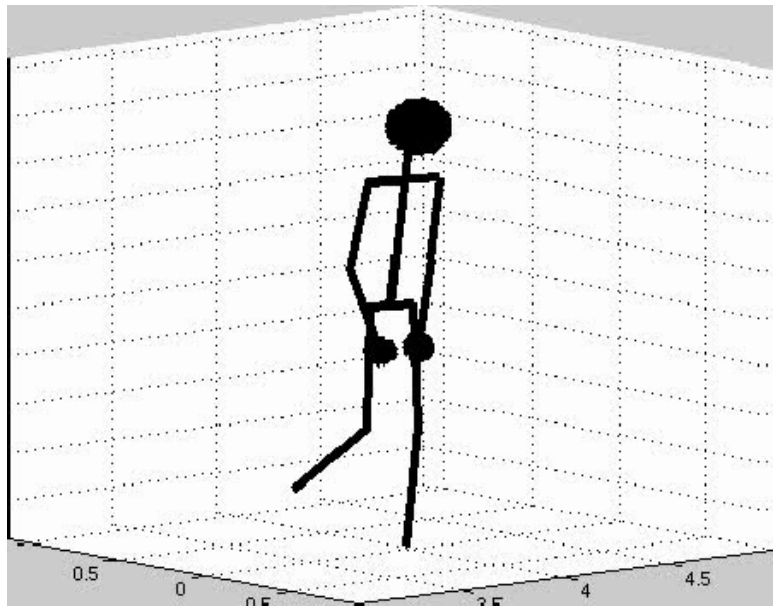
$$p(\psi_t \mid \psi_{t-1}) = G(\psi_t - \psi_{t-1} - v_{t-1}, \sigma^\psi)$$

$$p(\tau_t^g \mid \tau_{t-1}^g, v_{t-1}^g) = G(\tau_t^g - \tau_{t-1}^g - v_{t-1}^g,\ \sigma^\tau)$$

$$p(\theta_t^g \mid \theta_{t-1}^g) = G(\theta_t^g - \theta_{t-1}^g,\ \sigma^\theta)$$



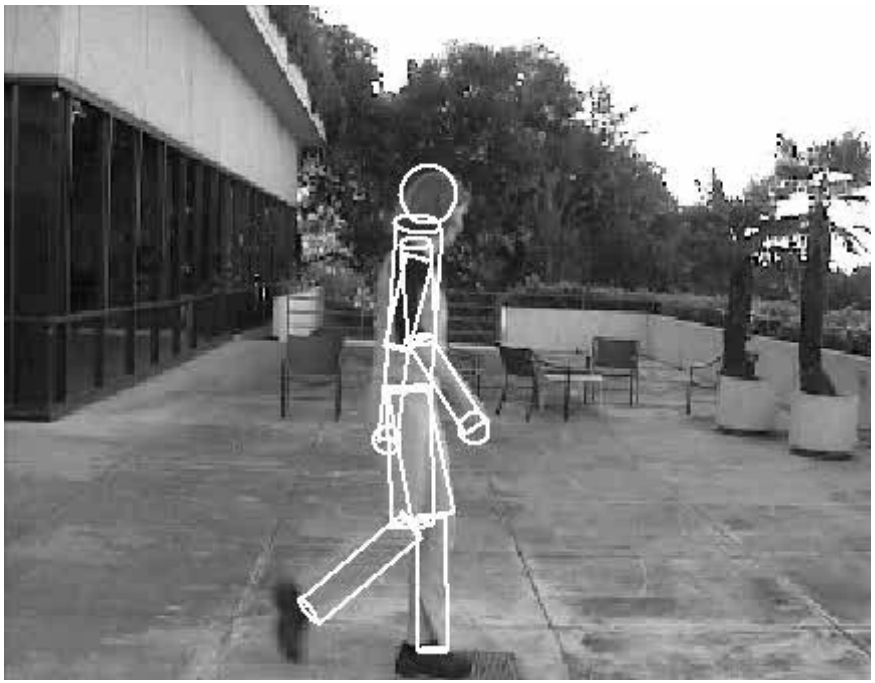
3D People Tracking



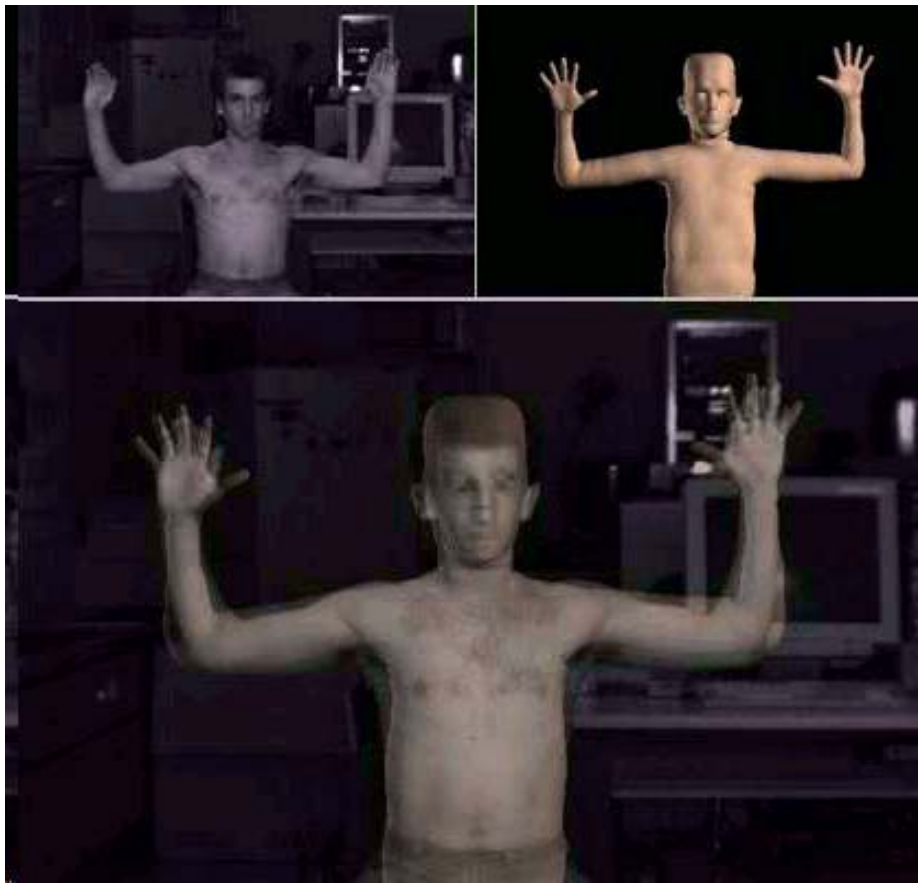
Mean posterior state shown from two viewpoints.
(15000 particles, manual initialization)

[Sidenbladh, Black & Fleet, "Stochastic tracking of 3D human figures using 2D image motion." Proc ECCV, 2000]

3D People Tracking



Multiple Cameras Help



(1 particle)

[Plankers & Fua, "Articulated soft objects for multiview shape and motion capture." IEEE Trans PAMI, 2003]

Proposals

The prediction distribution make a very poor proposal distribution unless the dynamical model is very strong

$$p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{z}}_{1:t}) \;=\; c\; p(\vec{\mathbf{z}}_t \,|\, \vec{\mathbf{x}}_t)\; p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{z}}_{1:t-1})$$

Instead, allow proposals exploit current observations:

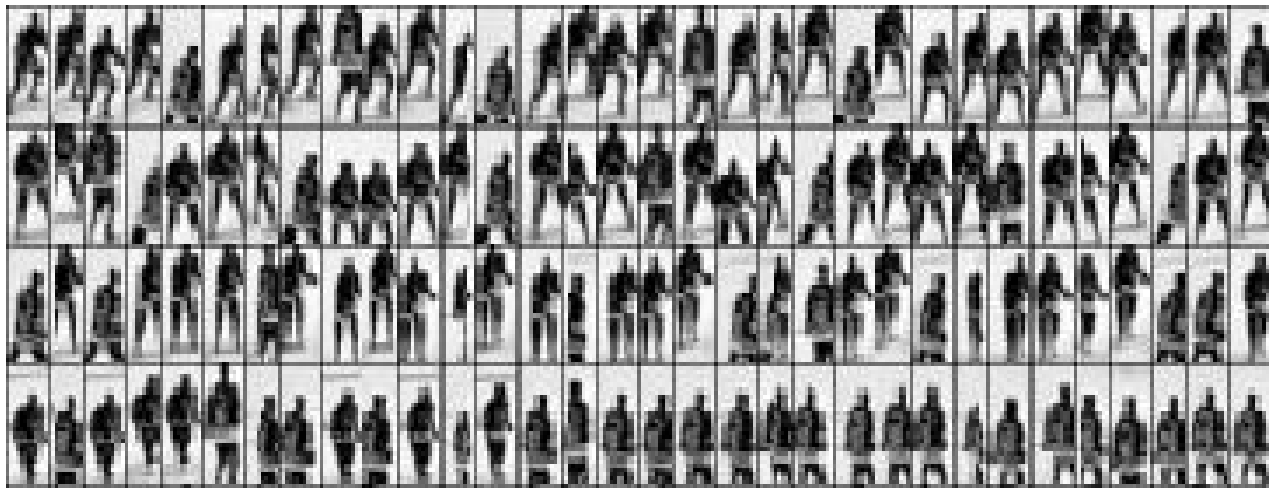
$$\mathcal{Q}(\vec{\mathbf{x}}_t) \;=\; \mathcal{D}(\vec{\mathbf{x}}_t)\; p(\vec{\mathbf{x}}_t \,|\, \vec{\mathbf{z}}_{1:t-1}) \quad , \quad w(\vec{\mathbf{x}}_t) \;=\; \frac{c\, p(\vec{\mathbf{z}}_t \,|\, \vec{\mathbf{x}}_t)}{\mathcal{D}(\vec{\mathbf{z}}_t \,|\, \vec{\mathbf{x}}_t)}$$

where $\mathcal{D}(\vec{\mathbf{x}}_t)$ is some continuous distribution obtained from some low-level detector (eg, Gaussian modes at locations of classifier hits)

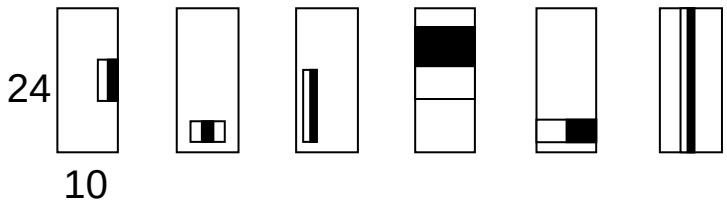
Tracking Hockey Players

Adaboost used to train a 23 layer classifier to detect hockey players:

- 2000 negative examples from locations on rink without players
- 200 positive examples



- Key (Haar) features:



[Okuma et al., "Boosted Particle Filter." Proc. ECCV 2004]

Tracking Hockey Players

State: number of players, plus positions / velocities (in rink coords)

Appearance: color histograms for top & bottom

Factored Posterior: independent filters applied to players (unless players in close proximity)

[Okuma et al., "Boosted Particle Filter." Proc. ECCV 2004]

Motion Boundary Analysis

Goal: Estimate flow, detect and track motion boundaries, and infer local depth ordering of adjacent surfaces

Model the optical flow in each local image neighborhood in terms of one of two types of motion model (i.e. a hybrid state space):

Smooth Motion

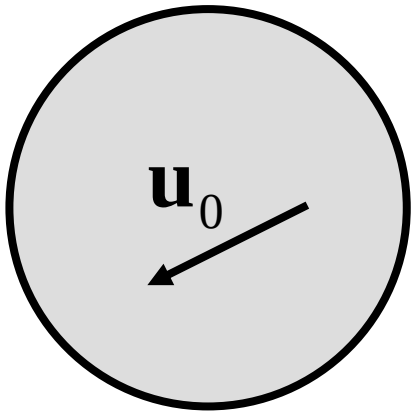
- translation

Motion Boundary

- location, orientation, velocities of 2 sides
- foreground/background assignment
- occluded / disoccluded pixels

[Fleet, Black and Nestares, 2002]

Generative Model: Smooth Motion



State Description: $\mathbf{s} = (\mu_0, \mathbf{u}_0)$

model type

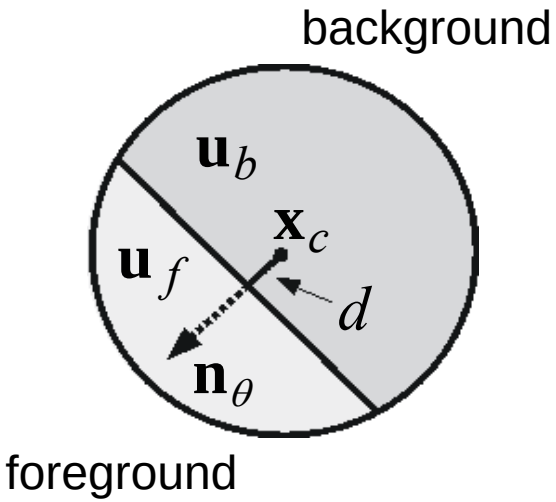
model parameters

Image Generation:

- state encodes deformation, $\mathbf{x}'(\mathbf{s}) = \mathbf{x} + \mathbf{u}_0$
- assume brightness constancy & Gaussian noise

$$I(\mathbf{x}, t) = I(\mathbf{x}'(\mathbf{s}), t-1) + \eta(\mathbf{x}, t)$$

Generative Model: Motion Boundary



State description:

$$\mathbf{s} = (\mu_1, \theta, \mathbf{u}_f, \mathbf{u}_b, d)$$

orientation

location

velocities

Image Generation:

$$\mathbf{x}'(\mathbf{s}) = \begin{cases} \mathbf{x} + \mathbf{u}_f & \text{if } \mathbf{x} \text{ on foreground} \\ \mathbf{x} + \mathbf{u}_b & \text{if } \mathbf{x} \text{ on background \& visible} \end{cases}$$

$$I(\mathbf{x}, t) = I(\mathbf{x}'(\mathbf{s}), t-1) + \eta(\mathbf{x}, t)$$

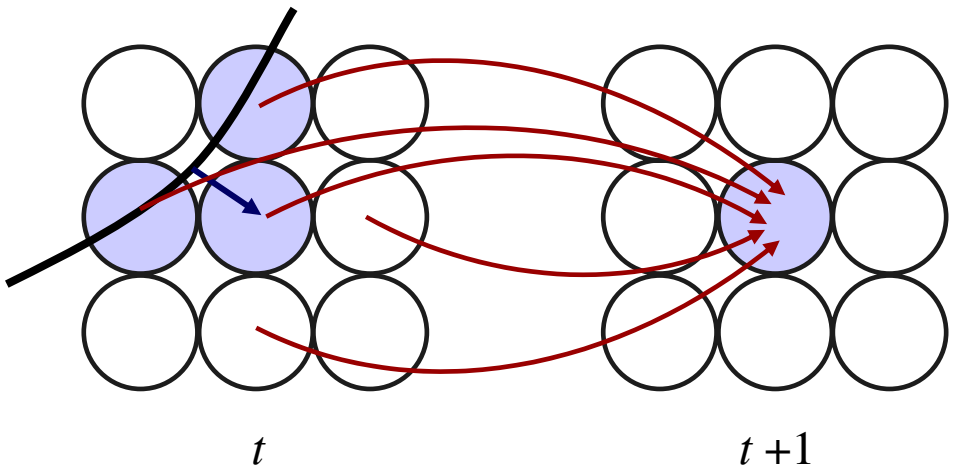
Hybrid MRF

State: Dense array of neighbourhoods with hybrid state

Likelihood: brightness constancy and translational flow

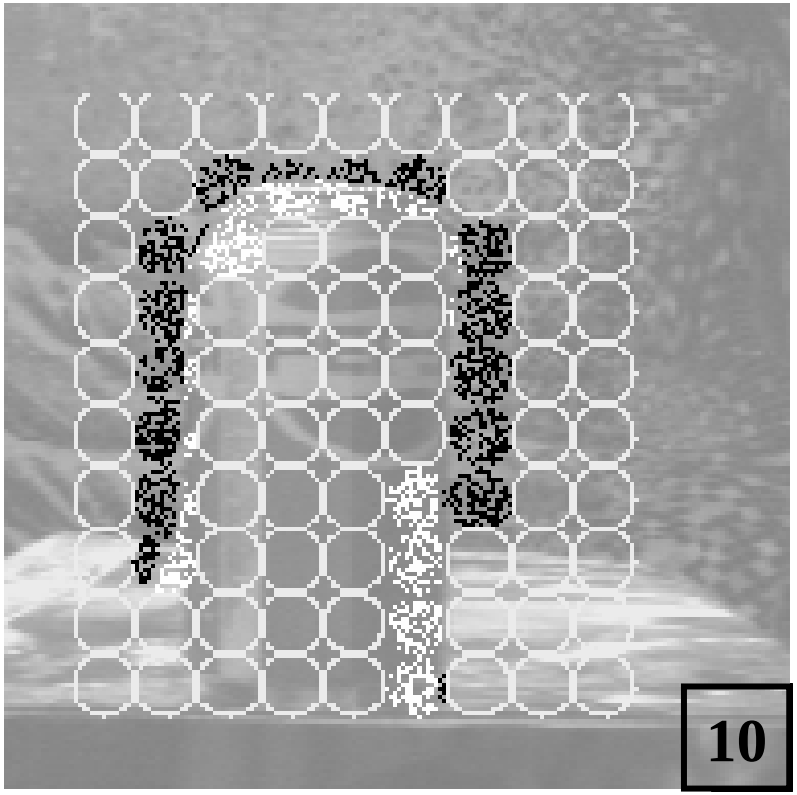
Dynamics: slow motion

Proposals: Low-level detectors for each neighbourhood, plus messages from neighbours from previous time (non-parametric belief propagation)

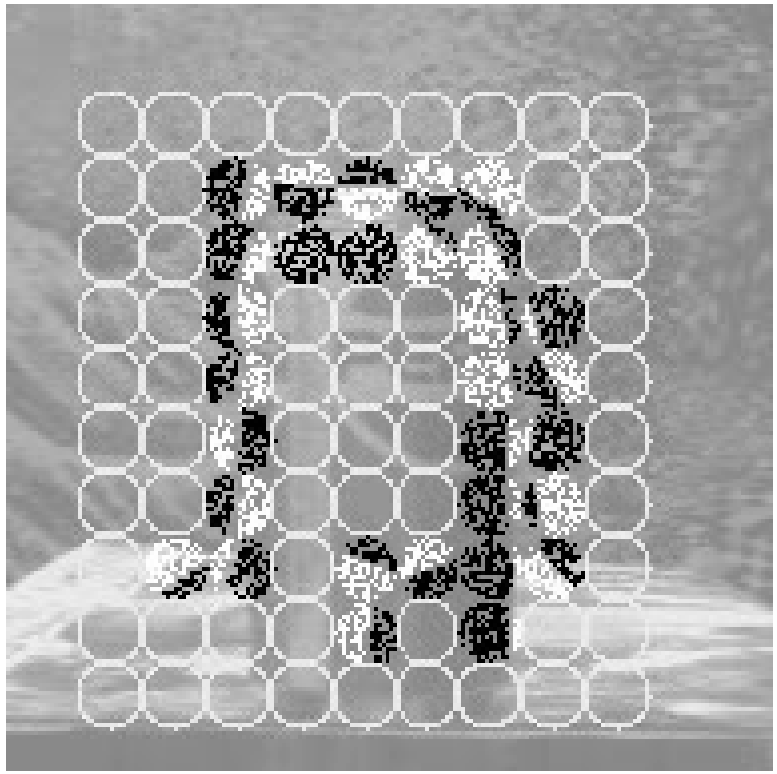


Pepsi Sequence

Pepsi Results



Pepsi Results



Factored State Spaces

- Factor posterior where possible (e.g., kinematic tree)
- Partitioned sampling (MacCormick et al, Proc. ECCV 2000)
- Handle portions of posterior analytically if possible (e.g., Rao-Blackwellization, Khan et al., CVPR 2004)

Discussion

Current trackers show real promise:

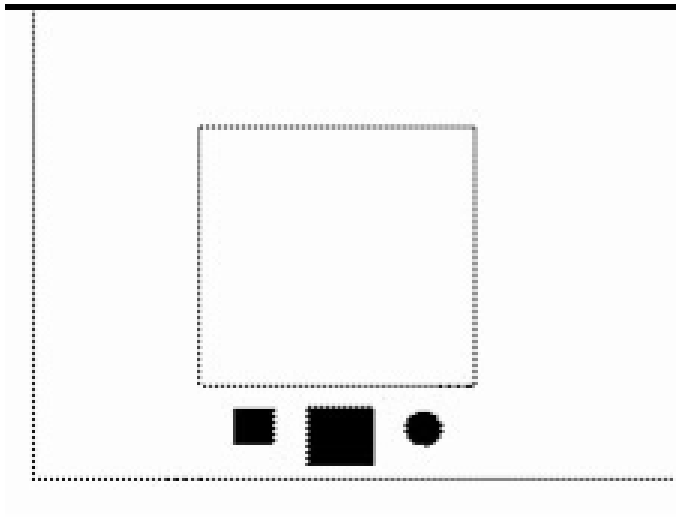
- handling non-Gaussian and multi-modal distributions with reasonably large state-space models of motion and appearance

Some hazards

- posteriors must be sufficient constrained with some combinations of posterior factorization, dynamics, measurement equations, …
- proposal distribution should be non-zero wherever the posterior distribution is non-zero (usually heavy-tailed)
- proposals should exploit current observations in addition to prediction distribution
- sampling variability can be a problem
  - must have enough samples in regions of high probability for normalization to be useful
  - too many samples needed for high dimensional problems (esp. when samples drawn independently from prediction dist)
  - samples tend to migrate to a single mode
  - sample deterministically where possible

Where do we go from here… long term?

Scene Dynamics



Interpretation of behaviour

(Heider and Simmel (1944))