Tracking using CONDENSATION: Conditional Density Propagation

M. Isard and A. Blake, CONDENSATION – Conditional density propagation for visual tracking, *Int. J. Computer Vision* **29**(1), 1998, pp. 4-28.

Goal

- Model-based visual tracking in dense clutter at near video frame rates

Example of CONDENSATION Algorithm

Approach

- Probabilistic framework for tracking objects such as curves in clutter using an iterative sampling algorithm
- Model motion and shape of target
- Top-down approach
- Simulation instead of analytic solution

Probabilistic Framework

- Object dynamics form a temporal Markov chain $p(x_t \mid X_{t-1}) = p(x_t \mid x_{t-1})$
- Observations, z_t, are independent (mutually and w.r.t process)

$$p(Z_{t-1}, x_t \mid X_{t-1}) = p(x_t \mid X_{t-1}) \prod_{i=1}^{t-1} p(z_i \mid x_i)$$

- Use Bayes' rule

Notation

X State vector, e.g., curve's position and orientation

Z Measurement vector, e.g., image edge locations

p(**X**) Prior probability of state vector; summarizes prior domain knowledge, e.g., by independent measurements

p(**Z**) Probability of measuring **Z**; fixed for any given image

p(**Z** | **X**) Probability of measuring **Z** given that the state is **X**; compares image to expectation based on state

p(**X** | **Z**) Probability of **X** given that measurement **Z** has occurred; called state posterior

Tracking as Estimation

- Compute state posterior, *p*(**X|Z**), and select next state to be the one that maximizes this (Maximum a Posteriori (MAP) estimate)
- Measurements are complex and noisy, so posterior cannot be evaluated in closed form
- Particle filter (iterative sampling) idea: Stochastically approximate the state posterior with a set of *N* weighted particles, (*s*, π), where *s* is a sample state and π is its weight
- Use Bayes' rule to compute *p*(**X|Z**)

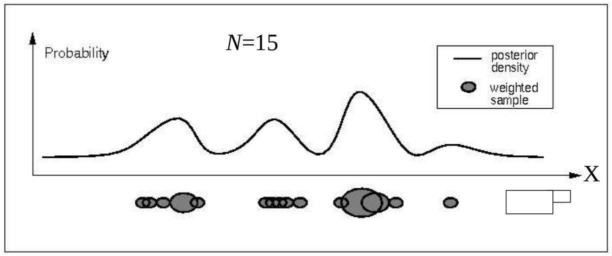
Factored Sampling

- Generate a set of samples that *approximates* the posterior *p*(**X|Z**)
- Sample set $\mathbf{s} = \{s^{(1)}, \ldots, s^{(N)}\}$ generated from *p*(**X**); each sample has a weight ("probability")

$$\pi_i = \frac{p_z(s^{(i)})}{\sum_{j=1}^{N} p_z(s^{(j)})}$$

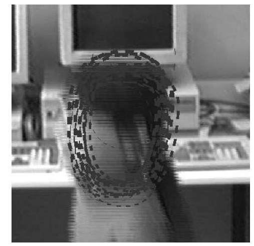
$$p_z(x) = p(z \mid x)$$

Factored Sampling

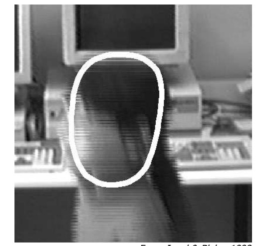


• CONDENSATION for one image

Estimating Target State



State samples



From *Isard & Blake, 1998*

Mean of weighted state samples

Bayes' Rule

This is what you can evaluate

This is what you may know a priori, or what you can ***predict***

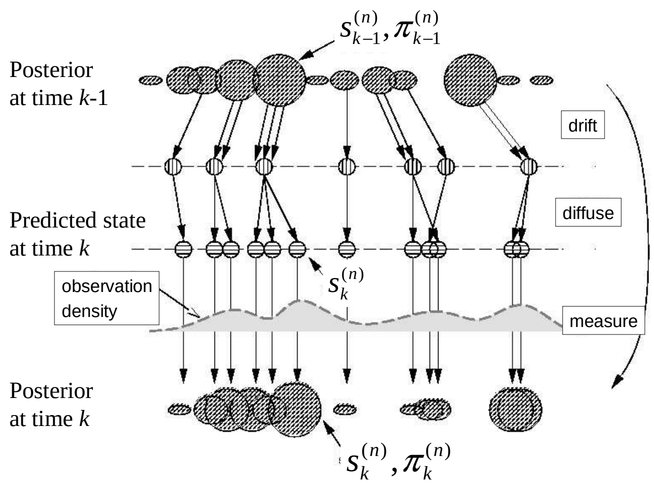
$$p(\mathbf{X} \mid \mathbf{Z}) = \frac{p(\mathbf{Z} \mid \mathbf{X}) \; p(\mathbf{X})}{p(\mathbf{Z})}$$

This is what you want. Knowing $p(\mathbf{X}|\mathbf{Z})$ will tell us what is the most likely state **X**.

This is a constant for a given image

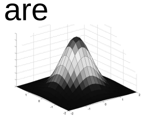
CONDENSATION Algorithm

1. **Select**: Randomly select *N* particles from $\{s_{t-1}^{(n)}\}$ based on weights $\pi_{t-1}^{(n)}$; same particle may be picked multiple times (*factored sampling*)
2. **Predict**: Move particles according to deterministic dynamics (*drift*), then perturb individually (*diffuse*)
3. **Measure**: Get a likelihood for each new sample by comparing it with the image's local appearance, i.e., based on $p(z_t|x_t)$; then update weight accordingly to obtain $\{(s_t^{(n)}, \pi_t^{(n)})\}$

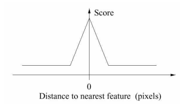


Notes on Updating

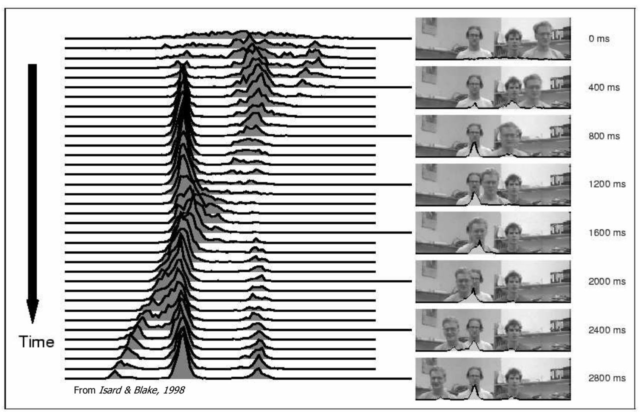
- Enforcing plausibility: Particles that represent impossible configurations are discarded
- Diffusion modeled with a Gaussian



- Likelihood function: Convert "goodness of prediction" score to pseudo-probability
  - More markings closer to predicted markings → higher likelihood



State Posterior



From *Isard & Blake, 1998*

State Posterior Animation

Object Motion Model

- For video tracking we need a way to propagate probability densities, so we need a "motion model" such as
  - $\mathbf{X}_{t+1} = \mathbf{A}\,\mathbf{X}_t + \mathbf{B}\,\mathbf{W}_t$ where **W** is a noise term and **A** and **B** are state transition matrices that can be learned from training sequences
- The state, **X**, of an object, e.g., a B-spline curve, can be represented as a point in a 6D state space of possible 2D affine transformations of the object

Evaluating $p(\mathbf{Z} \mid \mathbf{X})$

$$p(z \mid x) = q p(z \mid clutter) + \sum_{m=1}^{M} p(z \mid x, \phi_m)\, p(\phi_m)$$

where ϕ_m = {true measurement is z_m} for $m = 1,\ldots,M$, and $q = 1 - \Sigma_m p(\phi_m)$ is the probability that the target is not visible

$$\phi_m = \begin{cases} |x_m - z_m|^2 & if \quad |x_m - z_m| < \delta \\ \rho & otherwise \end{cases}$$

Dancing Example

Hand Example

Pointing Hand Example

Glasses Example

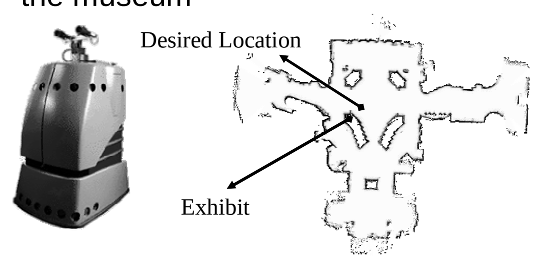
- 6D state space of affine transformations of a spline curve
- Edge detector applied along normals to the spline
- Autoregressive motion model

3D Model-based Example

- 3D state space: image position + angle
- Polyhedral model of object

Minerva

- Museum tour guide robot that used CONDENSATION to track its position in the museum



Advantages of Particle Filtering

- Nonlinear dynamics, measurement model easily incorporated
- Copes with lots of false positives
- Multi-modal posterior okay (unlike Kalman filter)
- Multiple samples provides multiple hypotheses
- Fast and simple to implement