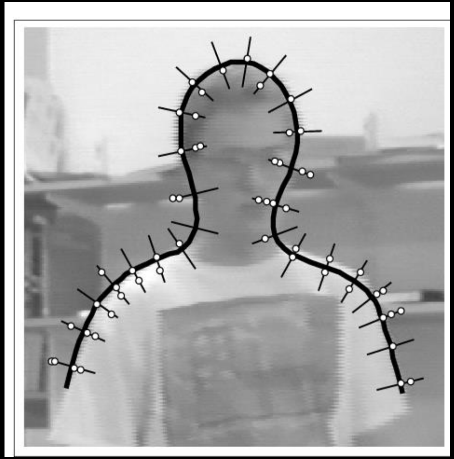
CS4495/6495
Introduction to Computer Vision

7C-L4 *Particle filters for real*



Bayes Filters: Framework

Given

1. Prior probability of the system state $p(x)$
2. Action (dynamical system) model:
$$p(x_t|u_{t-1}, x_{t-1})$$
3. Sensor model (likelihood) $p(z|x)$
4. Stream of observations z and action data u:
$$data_t = \{u_1, z_2 \ldots, u_{t-1}, z_t\}$$

To do real tracking…

$$p(\overbrace{x_t}^{\text{State}} \mid x_{t-1}, u_t)$$

$$\underbrace{p(x_t \mid x_{t-1}, u_t)}_{\text{State dynamics}}$$

$$p(x \mid z) = \frac{p(z \mid x)\, p(x)}{p(z)}$$

$$= \eta\, \underbrace{p(z \mid x)}_{\text{Sensor model}}\, p(x)$$

- ***x*** is the "state" – but of what? The object? Some representation of it?
- ***z*** is the "measurement" – but what measurement? And how does it relate to the state?
- Where do you get your dynamics from?

The source...

International Journal of Computer Vision 29(1), 5–28 (1998)
© 1998 Kluwer Academic Publishers. Manufactured in The Netherlands.

CONDENSATION—Conditional Density Propagation for Visual Tracking

MICHAEL ISARD AND ANDREW BLAKE
Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK

Received July 16, 1996; Accepted March 3, 1997

Abstract. The problem of tracking curves in dense visual clutter is challenging. Kalman filtering is inadequate because it is based on Gaussian densities which, being unimodal, cannot represent simultaneous alternative hypotheses. The CONDENSATION algorithm uses "factored sampling", previously applied to the interpretation of static images, in which the probability distribution of possible interpretations is represented by a randomly generated set. CONDENSATION uses learned dynamical models, together with visual observations, to propagate the random set over time. The result is highly robust tracking of agile motion. Notwithstanding the use of stochastic methods, the algorithm runs in near real-time.

Particle filter tracking – state

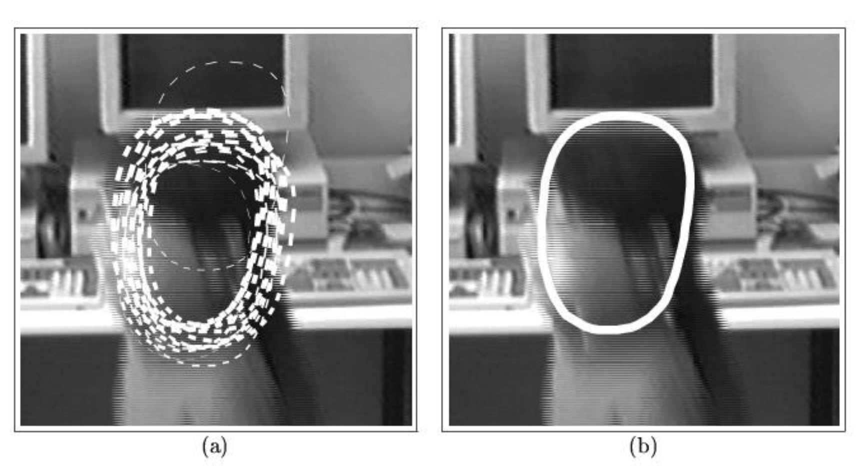
"Object" to be tracked here is a *hand-initialized* *contour*

The state is the countour *affine deformation*

How many parameters?

Each particle represents those six parameters

Impact on the number of particles?



Picture of states represented by top weighted particles

The mean state

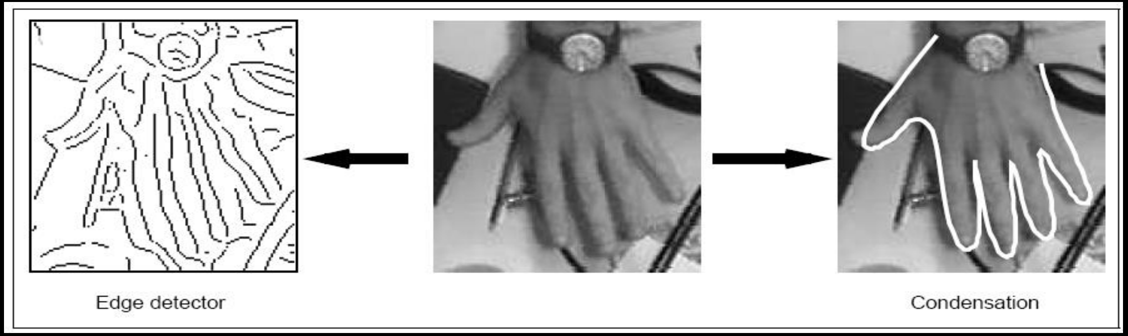
[Isard 1998]

Getting the dynamics – cheating?

More complex state

Tracking of a hand movement using an edge detector

State: Translation, rotation of hand + “shape” of hand => 12 DoF

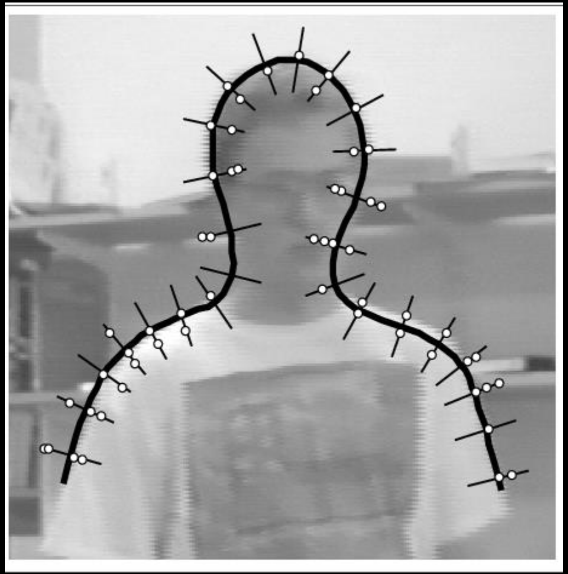


Particle filter tracking - measurement

Suppose x is a contour

- What is z?

$$p(\mathbf{z} \mid x) \propto \exp\left(-\frac{(dist\ to\ edge)^2}{2\sigma^2}\right)$$



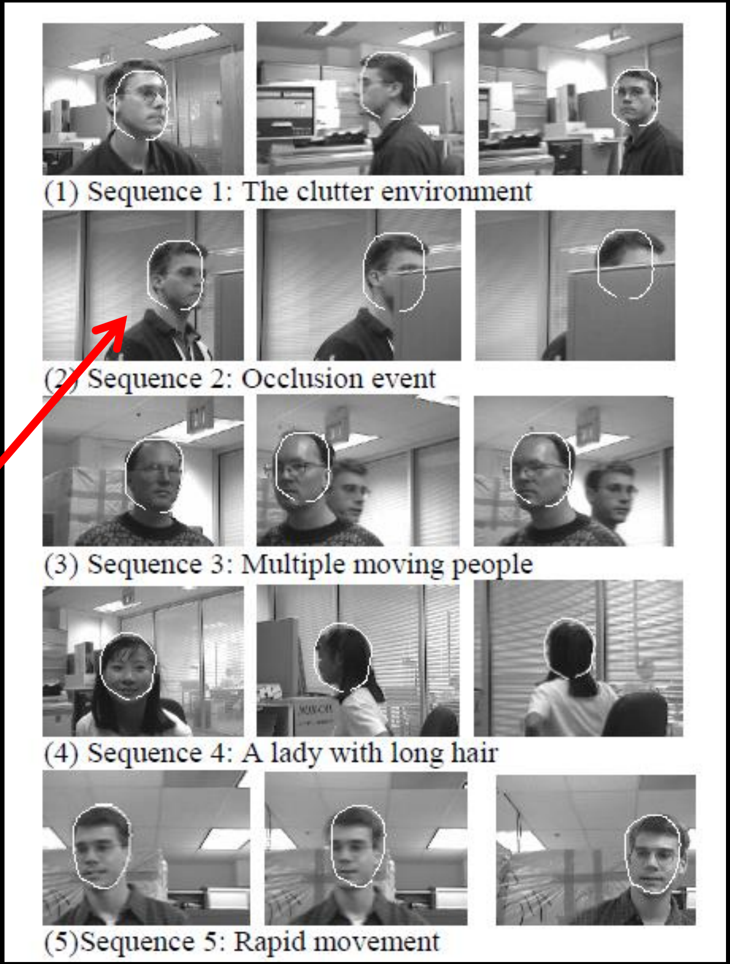
Gaussian of Distance to nearest high-contrast feature summed over the contour

[Isard 1998]

More tracking contours

- Head tracking with contour models (Zhihong Zeng et al. 2002)

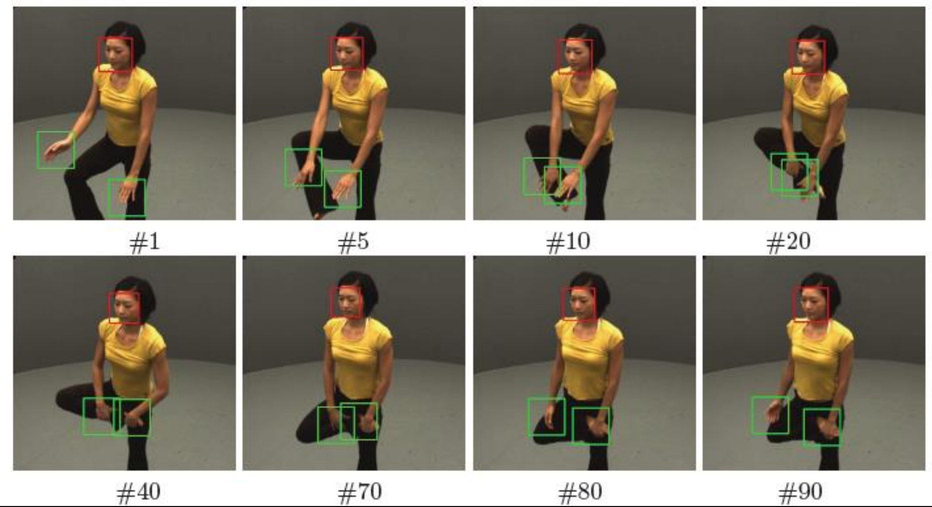
- How did it do occlusion?
  - With velocity?
  - Without velocity?



A different model

Hands and head movement tracking using color models and optical flow

- State: Location of colored blob (x, y)
- Prediction based upon flow
- Sensor model: Color match



[Tung et al. 2008]

An even better model

Suppose you want to track a region of colors…

What would be a good model/state?

- Location
- Region size?
- *Distribution of colors*

What would be a good sensor model?

- *Similarity of distributions*

How about a really, really simple model?

- State: Just location of an image patch (x, y)
- Dynamics: Random noise
- Sensor model: Mean squared difference of pixel intensities
  - Really a similarity model: More similar is more likely

- Oh, you need a patch…