The CONDENSATION algorithm — conditional density propagation and applications to visual tracking

A. Blake and M. Isard*
Department of Engineering Science,
University of Oxford,
Oxford OX1 3PJ, UK.

Abstract

The power of sampling methods in Bayesian reconstruction of noisy signals is well known. The extension of sampling to temporal problems is discussed. Efficacy of sampling over time is demonstrated with visual tracking.

1 INTRODUCTION

The problem of tracking curves in dense visual clutter is a challenging one. Trackers based on Kalman filters are of limited power; because they are based on Gaussian densities which are unimodal they cannot represent simultaneous alternative hypotheses. Extensions to the Kalman filter to handle multiple data associations (Bar-Shalom and Fortmann, 1988) work satisfactorily in the simple case of point targets but do not extend naturally to continuous curves.

Tracking is the propagation of shape and motion estimates over time, driven by a temporal stream of observations. The noisy observations that arise in realistic problems demand a robust approach involving propagation of probability distributions over time. Modest levels of noise may be treated satisfactorily using Gaussian densities, and this is achieved effectively by Kalman filtering (Gelb, 1974). More pervasive noise distributions, as commonly arise in visual background clutter, demand a more powerful, non-Gaussian approach.

One very effective approach is to use random sampling. The CONDENSATION algorithm, described here, combines random sampling with learned dynamical models to propagate an entire probability distribution for object position and shape, over time. The result is accurate tracking of agile motion in clutter, decidedly more

*Web: http://www.robots.ox.ac.uk/~ab/

robust than what has previously been attainable by Kalman filtering . Despite the use of random sampling, the algorithm is efficient, running in near real-time when applied to visual tracking.

2 SAMPLING METHODS

A standard problem in statistical pattern recognition is to find an object parameterised as $\mathbf{x}$ with prior $p(\mathbf{x})$, using data $\mathbf{z}$ from a single image. The posterior density $p(\mathbf{x}|\mathbf{z})$ represents all the knowledge about $\mathbf{x}$ that is deducible from the data. It can be evaluated in principle by applying Bayes' rule (Papoulis, 1990) to obtain

$$p(\mathbf{x}|\mathbf{z}) = kp(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \tag{1}$$

where k is a normalisation constant that is independent of $\mathbf{x}$. However $p(\mathbf{z}|\mathbf{x})$ may become sufficiently complex that $p(\mathbf{x}|\mathbf{z})$ cannot be evaluated simply in closed form. Such complexity arises typically in visual clutter, when the superfluity of observable features tends to suggest multiple, competing hypotheses for $\mathbf{x}$. A one-dimensional illustration of the problem is illustrated in figure 1 in which multiple features give

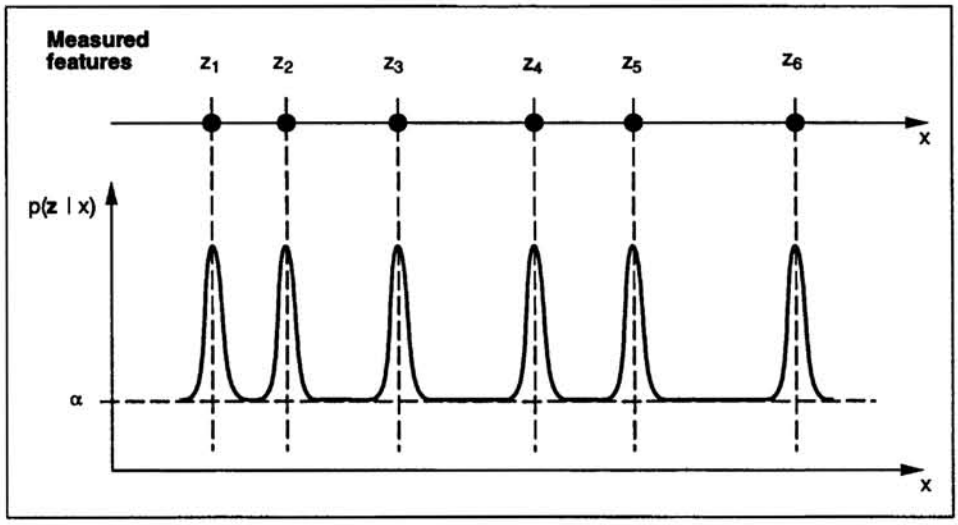

Figure 1: **One-dimensional observation model.** *A probabilistic observation model allowing for clutter and the possibility of missing the target altogether is specified here as a conditional density* $p(\mathbf{z}|x)$.

rise to a multimodal observation density function $p(\mathbf{z}|\mathbf{x})$.

When direct evaluation of $p(\mathbf{x}|\mathbf{z})$ is infeasible, iterative sampling techniques can be used (Geman and Geman, 1984; Ripley and Sutherland, 1990; Grenander et al., 1991; Storvik, 1994). The *factored sampling* algorithm (Grenander et al., 1991). generates a random variate $\mathbf{x}$ from a distribution $\tilde{p}(\mathbf{x})$ that approximates the posterior $p(\mathbf{x}|\mathbf{z})$. First a sample-set $\{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(N)}\}$ is generated from the prior density $p(\mathbf{x})$ and then a sample $\mathbf{x} = \mathbf{x}_i$, $i \in \{1, \ldots, N\}$ is chosen with probability

$$\pi_i = \frac{p(\mathbf{z}|\mathbf{x} = \mathbf{s}^{(i)})}{\sum_{j=1}^{N} p(\mathbf{z}|\mathbf{x} = \mathbf{s}^{(j)})}.$$

Sampling methods have proved remarkably effective for recovering static objects from cluttered images. For such problems $\mathbf{x}$ is multi-dimensional, a set of parameters for curve position and shape. In that case the sample-set $\{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(N)}\}$ represents

a distribution of **x**-values which can be seen as a distribution of curves in the image plane, as in figure 2.

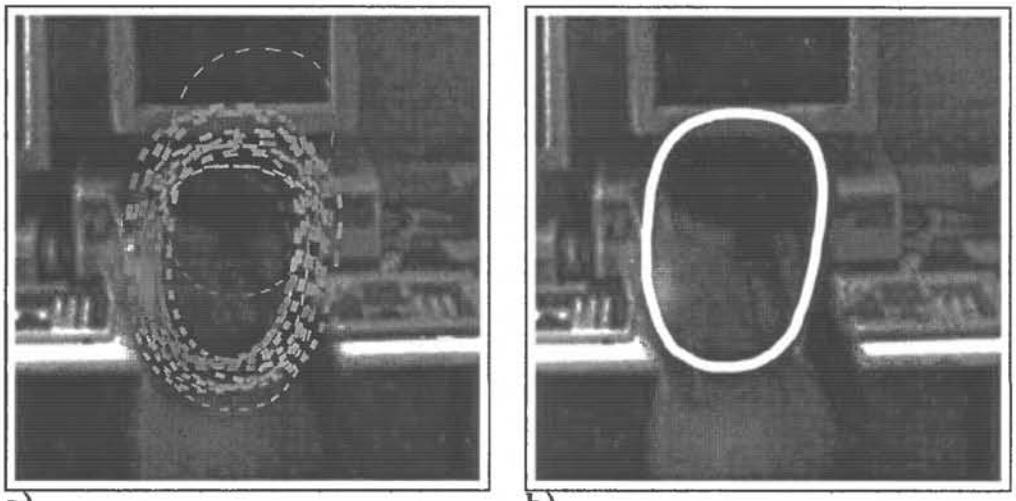

Figure 2: **Sample-set representation of shape distributions** *for a curve with parameters* $\mathbf{x}$, *modelling the outline (a) of the head of a dancing girl. Each sample* $\mathbf{s}^{(n)}$ *is shown as a curve (of varying position and shape) with a thickness proportional to the weight* $\pi^{(n)}$. *The weighted mean of the sample set (b) serves as an estimator of mean shape*

3 THE CONDENSATION ALGORITHM

The CONDENSATION algorithm is based on factored sampling but extended to apply iteratively to successive images in a sequence. Similar sampling strategies have appeared elsewhere (Gordon et al., 1993; Kitigawa, 1996), presented as developments of Monte-Carlo methods. The methods outlined here are described in detail elsewhere. Fuller descriptions and derivation of the CONDENSATION algorithm are in (Isard and Blake, 1996; Blake and Isard, 1997) and details of the learning of dynamical models, which is crucial to the effective operation of the algorithm are in (Blake et al., 1995).

Given that the estimation process at each time-step is a self-contained iteration of factored sampling, the output of an iteration will be a weighted, time-stamped sample-set, denoted $\mathbf{s}_t^{(n)}$, $n = 1, \ldots, N$ with weights $\pi_t^{(n)}$, representing approximately the conditional state-density $p(\mathbf{x}_t|\mathcal{Z}_t)$ at time t, where $\mathcal{Z}_t = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$. How is this sample-set obtained? Clearly the process must begin with a prior density and the effective prior for time-step t should be $p(\mathbf{x}_t|\mathcal{Z}_{t-1})$. This prior is of course multi-modal in general and no functional representation of it is available. It is derived from the sample set representation $(\mathbf{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)})$, $n = 1, \ldots, N$ of $p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})$, the output from the previous time-step, to which prediction must then be applied.

The iterative process applied to the sample-sets is depicted in figure 3. At the top of the diagram, the output from time-step $t-1$ is the weighted sample-set $\{(\mathbf{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)}),\ n = 1, \ldots, N\}$. The aim is to maintain, at successive time-steps, sample sets of fixed size N, so that the algorithm can be guaranteed to run within a given computational resource. The first operation therefore is to sample (with

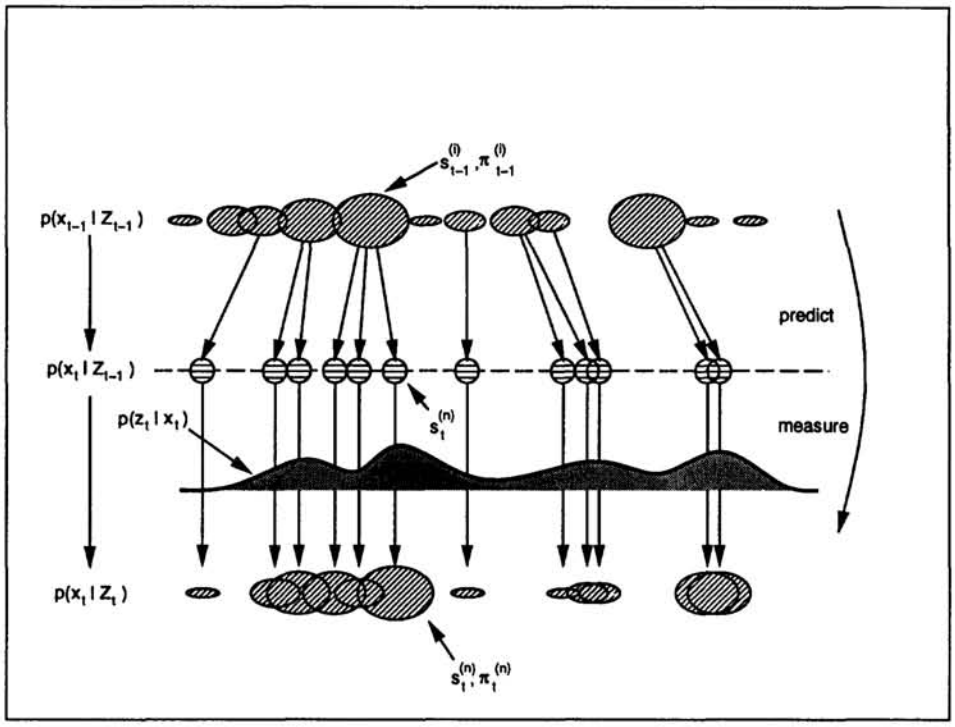

Figure 3: **One time-step in the** CONDENSATION **algorithm.** *Blob centres represent sample values and sizes depict sample weights.*

replacement) N times from the set $\{\mathbf{s}_{t-1}^{(n)}\}$, choosing a given element with probability $\pi_{t-1}^{(n)}$. Some elements, especially those with high weights, may be chosen several times, leading to identical copies of elements in the new set. Others with relatively low weights may not be chosen at all.

Each element chosen from the new set is now subjected to a predictive step. (The dynamical model we generally use for prediction is a linear stochastic differential equation (s.d.e.) learned from training sets of sample object motion (Blake et al., 1995).) The predictive step includes a random component, so identical elements may now split as each undergoes its own independent random motion step. At this stage, the sample set $\{\mathbf{s}_t^{(n)}\}$ for the new time-step has been generated but, as yet, without its weights; it is approximately a fair random sample from the effective prior density $p(\mathbf{x}_t|\mathcal{Z}_{t-1})$ for time-step t. Finally, the observation step from factored sampling is applied, generating weights from the observation density $p(\mathbf{z}_t|\mathbf{x}_t)$ to obtain the sample-set representation $\{(\mathbf{s}_t^{(n)}, \pi_t^{(n)})\}$ of state-density for time t.

The algorithm is specified in detail in figure 4. The process for a single time-step consists of N iterations to generate the N elements of the new sample set. Each iteration has three steps, detailed in the figure, and we comment below on each.

1. **Select** nth new sample $\mathbf{s}_t'^{(n)}$ to be some $\mathbf{s}_{t-1}^{(j)}$ from the old sample set, sampled with replacement with probability $\pi_{t-1}^{(j)}$. This is achieved efficiently by using *cumulative* weights $c_{t-1}^{(j)}$ (constructed in step 3).
2. **Predict** by sampling randomly from the conditional density for the dynamical model to generate a sample for the new sample-set.
3. **Measure** in order to generate weights $\pi_t^{(n)}$ for the new sample. Each weight

is evaluated from the observation density function which, being multimodal in general, "infuses" multimodality into the state density.

Iterate

From the "old" sample-set $\{\mathbf{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}, n = 1, \ldots, N\}$ at time-step $t-1$, construct a "new" sample-set $\{\mathbf{s}_t^{(n)}, \pi_t^{(n)}, c_t^{(n)}\}, n = 1, \ldots, N$ for time t.

Construct the n^{th} of N new samples as follows:

1. **Select** a sample $\mathbf{s}_t'^{(n)}$ as follows:
   (a) generate a random number $r \in [0, 1]$, uniformly distributed.
   (b) find, by binary subdivision, the smallest j for which $c_{t-1}^{(j)} \geq r$
   (c) set $\mathbf{s}_t'^{(n)} = \mathbf{s}_{t-1}^{(j)}$
2. **Predict** by sampling from
$$p(\mathbf{x}_t | \mathbf{x}_{t-1} = \mathbf{s}'^{(n)}_{t-1})$$
to choose each $\mathbf{s}_t^{(n)}$.
3. **Measure** and weight the new position in terms of the measured features $\mathbf{z}_t$:
$$\pi_t^{(n)} = p(\mathbf{z}_t | \mathbf{x}_t = \mathbf{s}_t^{(n)})$$
then normalise so that $\sum_n \pi_t^{(n)} = 1$ and store together with cumulative probability as $(\mathbf{s}_t^{(n)}, \pi_t^{(n)}, c_t^{(n)})$ where
$$\begin{aligned} c_t^{(0)} &= 0, \\ c_t^{(n)} &= c_t^{(n-1)} + \pi_t^{(n)} \quad (n = 1 \ldots N). \end{aligned}$$

Figure 4: **The CONDENSATION algorithm.**

At any time-step, it is possible to "report" on the current state, for example by evaluating some moment of the state density as

$$\mathcal{E}[f(\mathbf{x}_t)] = \sum_{n=1}^{N} \pi_t^{(n)} f\left(\mathbf{s}_t^{(n)}\right). \tag{2}$$

4 RESULTS

A good deal of experimentation has been performed in applying the CONDENSATION algorithm to the tracking of visual motion, including moving hands and dancing figures. Perhaps one of the most stringent tests was the tracking of a leaf on a bush, in which the foreground leaf is effectively camouflaged against the background.

A 12 second (600 field) sequence shows a bush blowing in the wind, the task being to track one particular leaf. A template was drawn by hand around a still of one chosen leaf and allowed to undergo affine deformations during tracking. Given that a clutter-free training sequence is not available, the motion model was learned by means of a bootstrap procedure (Blake et al., 1995). A tracker with default dynamics proved capable of tracking the first 150 fields of a training sequence before losing