CONDENSATION — conditional density propagation for visual tracking

Michael Isard and Andrew Blake,
Department of Engineering Science,
University of Oxford,
Oxford OX1 3PJ, UK.

Int. J. Computer Vision, in press (1998).

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.

Contents

1 Tracking curves in clutter **2**

2 Discrete-time propagation of state density **3**

3 Factored sampling **6**

4 The CONDENSATION algorithm **8**

5 Stochastic dynamical models for curve motion **10**

6 Observation model **13**

7 Applying the CONDENSATION algorithm to video-streams **17**

8 Conclusions **26**

A Non-linear filtering **31**

B Derivation of the sampling rule **33**

C Asymptotic correctness of the CONDENSATION Algorithm **34**

1 Tracking curves in clutter

The purpose of this paper[1] is to establish a stochastic framework for tracking curves in visual clutter, using a sampling algorithm. The approach is rooted in ideas from statistics, control theory and computer vision. The problem is to track outlines and features of foreground objects, modelled as curves, as they move in *substantial* clutter, and to do it at, or close to, video frame-rate. This is challenging because elements in the background clutter may mimic parts of foreground features. In the most severe case of camouflage, the background may consist of objects similar to the foreground object, for instance when a person is moving past a crowd. Our approach aims to dissolve the resulting ambiguity by applying probabilistic models of object shape and motion to analyse the video-stream. The degree of generality of these models is pitched carefully: sufficiently specific for effective disambiguation but sufficiently general to be broadly applicable over entire classes of foreground objects.

1.1 Modelling shape and motion

Effective methods have arisen in computer vision for modelling shape and motion. When suitable geometric models of a moving object are available, they can be matched effectively to image data, though usually at considerable computational cost (Hogg, 1983; Lowe, 1991; Sullivan, 1992; Huttenlocher et al., 1993). Once an object has been located approximately, tracking it in subsequent images becomes more efficient computationally (Lowe, 1992), especially if motion is modelled as well as shape (Gennery, 1992; Harris, 1992). One important facility is the modelling of curve segments which interact with images (Fischler and Elschlager, 1973; Yuille and Hallinan, 1992) or image sequences (Kass et al., 1987; Dickmanns and Graefe, 1988). This is more general than modelling entire objects but more clutter-resistant than applying signal-processing to low-level corners or edges. The methods to be discussed here have been applied at this level, to segments of parametric B-spline curves (Bartels et al., 1987) tracking over image sequences (Menet et al., 1990; Cipolla and Blake, 1990). The B-spline curves could, in theory, be parameterised by their control points. In practice this allows too many degrees of freedom for stable tracking and it is necessary to restrict the curve to a low-dimensional parameter $\mathbf{x}$, for example over an affine space (Koenderink and Van Doorn, 1991; Ullman and Basri, 1991; Blake et al., 1993), or more generally allowing a "shape-space" of non-rigid motion (Cootes et al., 1993).

Finally, *prior* probability densities can be defined over the curves (Cootes et al., 1993) represented by appropriate parameter vectors $\mathbf{x}$, and also over their motions (Terzopoulos and Metaxas, 1991; Blake et al., 1993), and this constitutes a powerful facility for tracking. Reasonable defaults can be chosen for those densities. However, it is obviously more satisfactory to measure or estimate them from data-sequences $(\mathbf{x}_1, \mathbf{x}_2, \ldots)$. Algorithms to do this, assuming Gaussian densities, are known in the control-theory literature (Goodwin and Sin, 1984) and have been applied in computer vision (Blake and Isard, 1994; Baumberg and Hogg, 1995). Given the prior, and an *observation density* that characterises the statistical variability of image data $\mathbf{z}$ given a curve state $\mathbf{x}$, a posterior distribution can, in principle, be estimated for $\mathbf{x}_t$ given $\mathbf{z}_t$ at successive times t.

[1] This paper has appeared in short form (Isard and Blake, 1996) as joint winner of the prize of the European Conference on Computer Vision, 1996.

1.2 Kalman filters and data-association

Spatio-temporal estimation, the tracking of shape and position over time, has been dealt with thoroughly by Kalman filtering, in the relatively clutter-free case in which $p(\mathbf{x}_t)$ can satisfactorily be modelled as Gaussian (Dickmanns and Graefe, 1988; Harris, 1992; Gennery, 1992; Rehg and Kanade, 1994; Matthies et al., 1989) and can be applied to curves (Terzopoulos and Szeliski, 1992; Blake et al., 1993). These solutions work relatively poorly in clutter which causes the density for $\mathbf{x}_t$ to be multi-modal and therefore non-Gaussian. With simple, discrete features such as points or corners combinatorial data-association methods can be effective with clutter but combinatorial methods to do not apply naturally to curves. There remains a need for an appropriately general probabilistic mechanism to handle multi-modal density functions.

1.3 Temporal propagation of conditional densities

The Kalman filter as a recursive linear estimator is a special case, applying only to Gaussian densities, of a more general probability density propagation process. In continuous time this can be described in terms of diffusion, governed by a "Fokker-Planck" equation (Astrom, 1970), in which the density for $\mathbf{x}_t$ drifts and spreads under the action of a stochastic model of its dynamics. In the simple Gaussian case, the diffusion is purely linear and the density function evolves as a Gaussian pulse that translates, spreads and is reinforced, remaining Gaussian throughout, as in figure 1, a process that is described analytically and exactly by the Kalman filter. The random component of the dynamical model leads to spreading — increasing uncertainty — while the deterministic component causes the density function to drift bodily. The effect of an external observation $\mathbf{z}_t$ is to superimpose a reactive effect on the diffusion in which the density tends to peak in the vicinity of observations. In clutter, there are typically several competing observations and these tend to encourage a non-Gaussian state-density (figure 2).

The CONDENSATION algorithm is designed to address this more general situation. It has the striking property that, generality notwithstanding, it is a considerably simpler algorithm than the Kalman filter. Moreover, despite its use of random sampling which is often thought to be computationally inefficient, the CONDENSATION algorithm runs in near real-time. This is because tracking over time maintains relatively tight distributions for shape at successive time-steps, and particularly so given the availability of accurate, learned models of shape and motion.

2 Discrete-time propagation of state density

For computational purposes, the propagation process must be set out in terms of discrete time t. The state of the modelled object at time t is denoted $\mathbf{x}_t$ and its history is $\mathcal{X}_t = \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$. Similarly the set of image features at time t is $\mathbf{z}_t$ with history $\mathcal{Z}_t = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$. Note that no functional assumptions (linearity, Gaussianity, unimodality) are made about densities in the general treatment, though particular choices will be made in due course in order to demonstrate the approach.

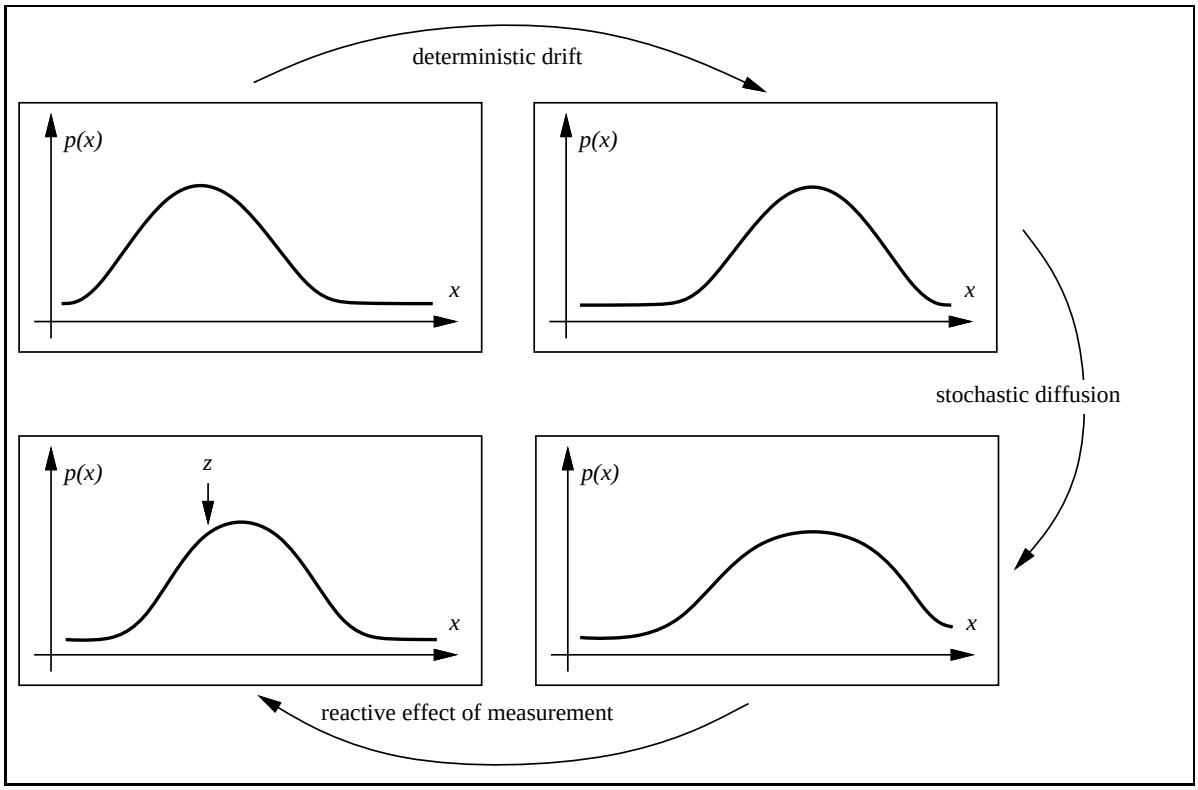


Figure 1: **Kalman filter as density propagation.** *In the case of Gaussian prior, process and observation densities, and assuming linear dynamics, the propagation process of figure 2 reduces to a diffusing Gaussian state density, represented completely by its evolving (multivariate) mean and variance — precisely what a Kalman filter computes.*

2.1 Stochastic dynamics

A somewhat general assumption is made for the probabilistic framework that the object dynamics form a temporal Markov chain so that

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

— the new state is conditioned directly only on the immediately preceding state, independent of the earlier history. This still allows quite general dynamics, including stochastic difference equations of arbitrary order; we use second order models and details are given later. The dynamics are entirely determined therefore by the form of the conditional density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. For instance,

$$p(x_t|x_{t-1}) \propto \exp -\frac{1}{2}(x_t - x_{t-1} - 1)^2$$

represents a one-dimensional random walk (discrete diffusion) whose step length is a standard normal variate, superimposed on a rightward drift at unit speed. Of course, for realistic problems, the state $\mathbf{x}$ is multi-dimensional and the density is more complex (and, in the applications presented later, learned from training sequences).

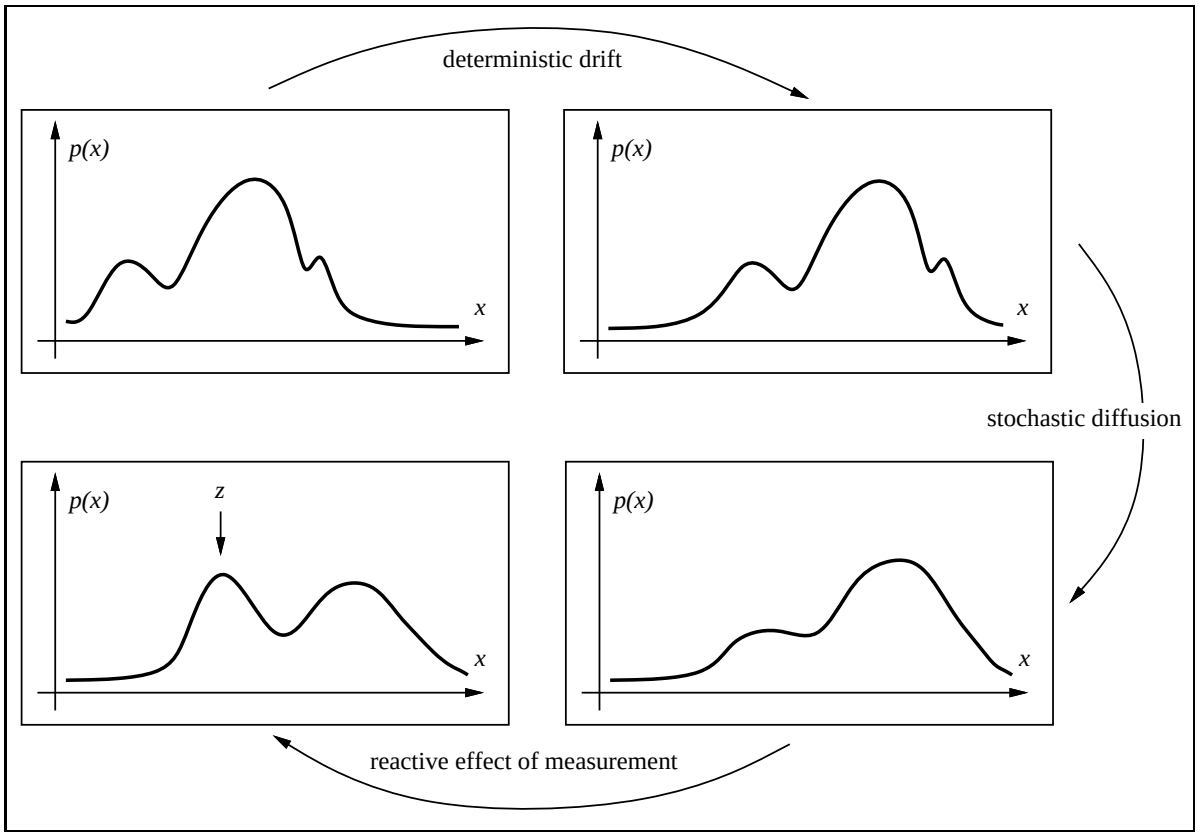


Figure 2: **Probability density propagation.** *Propagation is depicted here as it occurs over a discrete time-step. There are three phases: drift due to the deterministic component of object dynamics; diffusion due to the random component; reactive reinforcement due to observations.*

2.2 Measurement

Observations $\mathbf{z}_t$ are assumed to be independent, both mutually and with respect to the dynamical process. This is expressed probabilistically as follows:

$$p(\mathcal{Z}_{t-1}, \mathbf{x}_t | \mathcal{X}_{t-1}) = p(\mathbf{x}_t | \mathcal{X}_{t-1}) \prod_{i=1}^{t-1} p(\mathbf{z}_i | \mathbf{x}_i). \tag{2}$$

Note that integrating over $\mathbf{x}_t$ implies the mutual conditional independence of observations:

$$p(\mathcal{Z}_t | \mathcal{X}_t) = \prod_{i=1}^{t} p(\mathbf{z}_i | \mathbf{x}_i). \tag{3}$$

The observation process is therefore defined by specifying the conditional density $p(\mathbf{z}_t | \mathbf{x}_t)$ at each time t, and later, in computational examples, we take this to be a time-independent function $p(\mathbf{z}|\mathbf{x})$. Suffice it to say for now that, in clutter, the observation density is multi-modal. Details will be given in section 6

2.3 Propagation

Given a continuous-valued Markov chain with independent observations, the conditional state-density p_t at time t is defined by

$$p_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t|\mathcal{Z}_t).$$

This represents all information about the state at time t that is deducible from the entire data-stream up to that time. The rule for propagation of state density over time is:

$$p(\mathbf{x}_t|\mathcal{Z}_t) = k_t \; p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{Z}_{t-1}), \tag{4}$$

where

$$p(\mathbf{x}_t|\mathcal{Z}_{t-1}) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}) \tag{5}$$

and k_t is a normalisation constant that does not depend on $\mathbf{x}_t$. The validity of the rule is proved in the appendix.

The propagation rule (4) should be interpreted simply as the equivalent of the Bayes' rule (6) for inferring posterior state density from data, for the time-varying case. The effective prior $p(\mathbf{x}_t|\mathcal{Z}_{t-1})$ is actually a prediction taken from the posterior $p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})$ from the previous time-step, onto which is superimposed one time-step from the dynamical model (Fokker-Planck drift plus diffusion as in figure 2), which is expressed in (5). Multiplication in (4) by the observation density $p(\mathbf{z}_t|\mathbf{x}_t)$ in the Bayesian manner then applies the reactive effect expected from observations. Because the observation density is non-Gaussian, the evolving state density $p(\mathbf{x}_t|\mathcal{Z}_t)$ is also generally non-Gaussian. The problem now is how to apply a *nonlinear filter* to evaluate the state density over time, without incurring excessive computational load. Inevitably this means approximating. Numerous approaches, including "multiple hypothesis tracking", have been proposed but prove unsuitable for use with curves as opposed to discrete features — details are given in the appendix. In this paper we propose a sampling approach which is described in the following two sections.

3 Factored sampling

This section describes first the factored sampling algorithm dealing with non-Gaussian observations in single images. Then factored sampling is extended in the following section to deal with temporal image sequences.

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

$$p(\mathbf{x}|\mathbf{z}) = kp(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \tag{6}$$

where k is a normalisation constant that is independent of $\mathbf{x}$. In cases where $p(\mathbf{z}|\mathbf{x})$ is sufficiently complex that $p(\mathbf{x}|\mathbf{z})$ cannot be evaluated simply in closed form, 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 an index $i \in \{1, \ldots, N\}$ is chosen with probability π_i, where

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

and

$$p_z(\mathbf{x}) = p(\mathbf{z}|\mathbf{x}),$$

the conditional observation density. The value $\mathbf{x}' = \mathbf{x}_i$ chosen in this fashion has a distribution which approximates the posterior $p(\mathbf{x}|\mathbf{z})$ increasingly accurately as N increases (figure 3).

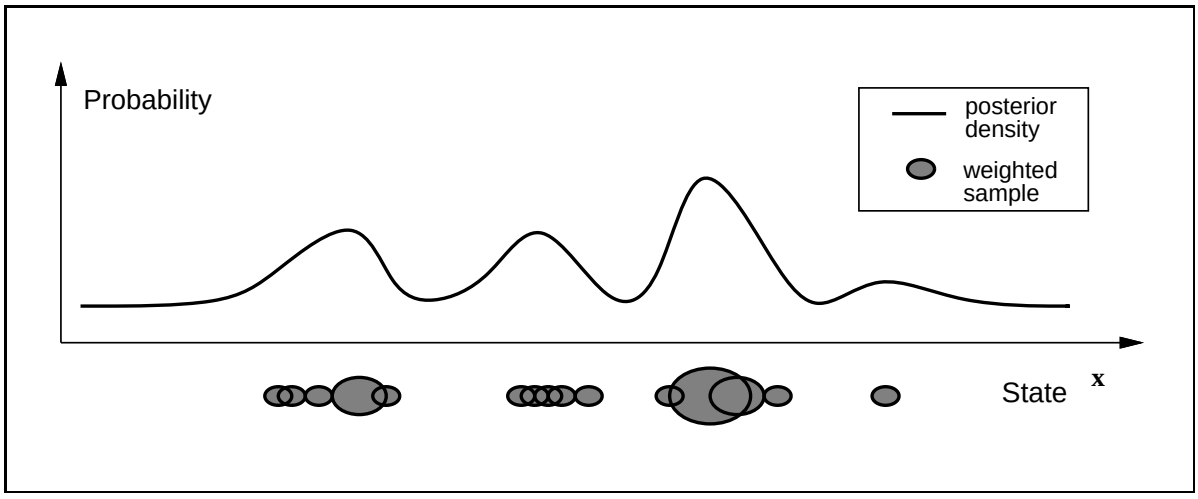


Figure 3: **Factored sampling.** *A set of points $\mathbf{s}^{(i)}$, the centres of the blobs in the figure, is sampled randomly from a prior density $p(\mathbf{x})$. Each sample is assigned a weight π_i (depicted by blob area) in proportion to the value of the observation density $p(\mathbf{z}|\mathbf{x} = \mathbf{s}^{(i)})$. The weighted point-set then serves as a representation of the posterior density $p(\mathbf{x}|\mathbf{z})$, suitable for sampling. The one-dimensional case illustrated here extends naturally to the practical case that the density is defined over several position and shape variables.*

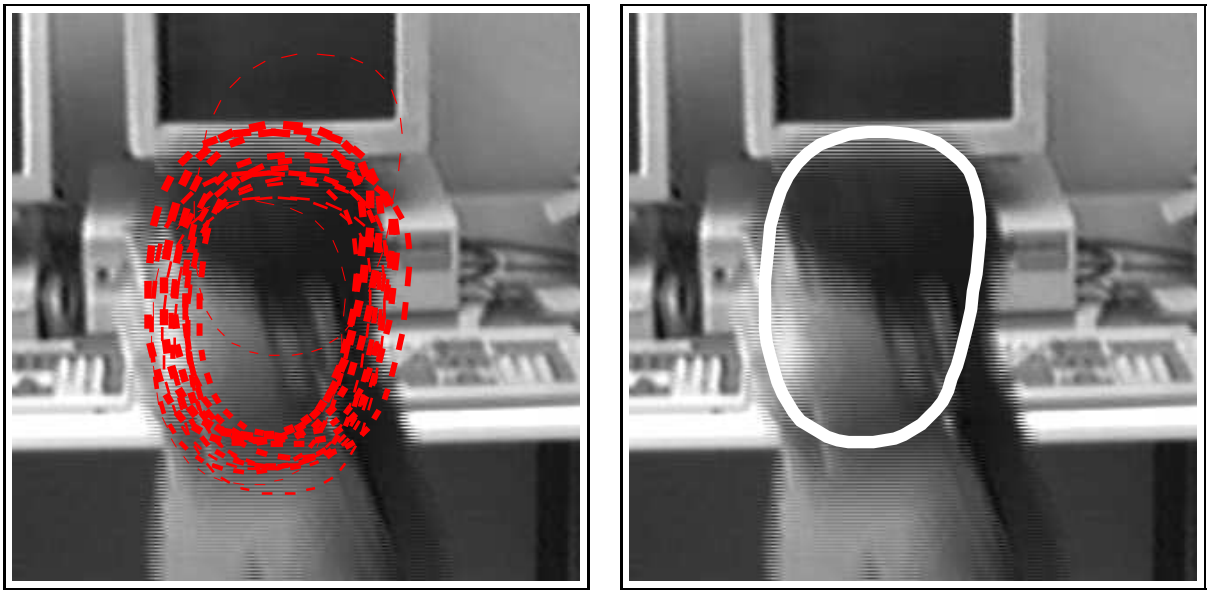
Note that posterior mean properties $\mathcal{E}[g(\mathbf{x})|\mathbf{z}]$ can be generated directly from the samples $\{\mathbf{s}^{(n)}\}$ by weighting with $p_z(\mathbf{x})$ to give:

$$\mathcal{E}[g(\mathbf{x})|\mathbf{z}] \approx \frac{\sum_{n=1}^{N} g(\mathbf{s}^{(n)}) p_z(\mathbf{s}^{(n)})}{\sum_{n=1}^{N} p_z(\mathbf{s}^{(n)})}. \tag{7}$$

For example, the mean can be estimated using $g(\mathbf{x}) = \mathbf{x}$ (illustrated in figure 4) and the variance using $g(\mathbf{x}) = \mathbf{x}\mathbf{x}^T$. In the case that $p(\mathbf{x})$ is a spatial Gauss-Markov process, Gibbs sampling from $p(\mathbf{x})$ has been used to generate the random variates $\{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(N)}\}$. Otherwise, for low-dimensional parameterisations as in this paper, standard, direct methods can be used for Gaussians[2] (Press et al., 1988). Note that, in the case that the density $p(\mathbf{z}|\mathbf{x})$ is normal, the mean obtained by factored sampling is consistent with an estimate obtained more conventionally, and efficiently, from linear least squares estimation. For multi-modal distributions which

[2]Note: the presence of clutter causes $p(\mathbf{z}|\mathbf{x})$ to be non-Gaussian, but the prior $p(\mathbf{x})$ may still happily be Gaussian, and that is what will be assumed in our experiments.

cannot be approximated as normal, so that linear estimators are unusable, estimates of mean $\mathbf{x}$ by factored sampling continue to apply.



a) b)

Figure 4: **Sample-set representation of shape distributions** *The sample-set representation of probability distributions, illustrated in one dimension in figure 3, is illustrated here (a) as it applies to the distribution of a multi-dimensional curve parameter* $\mathbf{x}$. *Each sample* $\mathbf{s}^{(n)}$ *is shown as a curve (of varying position and shape) with a thickness proportional to the weight* π_n. *The weighted mean of the sample set (b) serves as an estimator of the distribution mean.*

4 The CONDENSATION algorithm

The CONDENSATION algorithm is based on factored sampling but extended to apply iteratively to successive images in a sequence. The same sampling strategy has been developed elsewhere (Gordon et al., 1993; Kitagawa, 1996), presented as developments of Monte-Carlo methods. Jump-diffusion tracking (Miller et al., 1995) may also be related to the approach described here.

Given that the 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. 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 (5) must then be applied.

The iterative process as applied to sample-sets, depicted in figure 5, mirrors the continuous diffusion process in figure 2. 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 replacement)

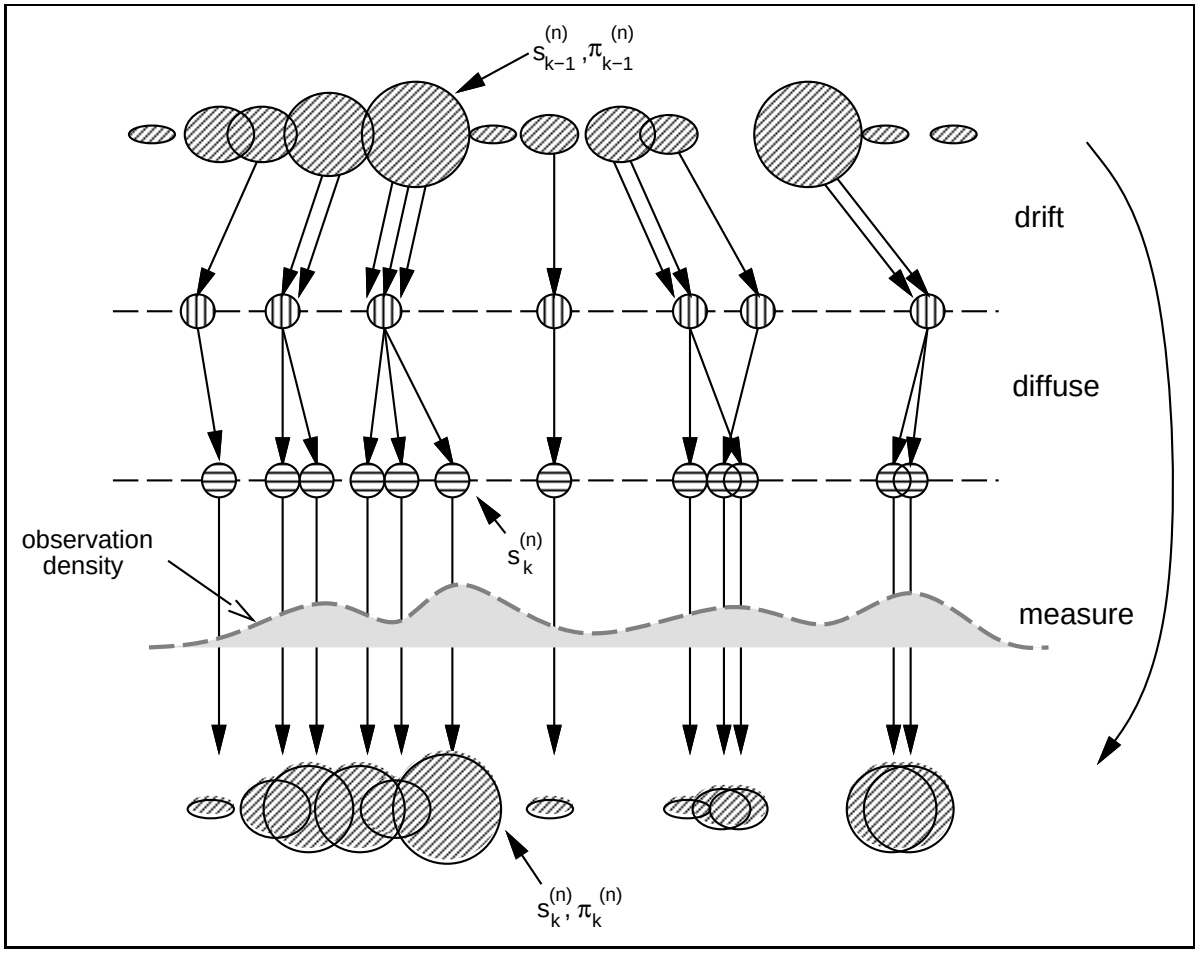


Figure 5: **One time-step in the** CONDENSATION **algorithm.** *Each of the three steps — drift-diffuse-measure — of the probabilistic propagation process of figure 2 is represented by steps in the* CONDENSATION *algorithm.*

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 the predictive steps. First, an element undergoes drift and, since this is deterministic, identical elements in the new set undergo the same drift. This is apparent in the figure. The second predictive step, diffusion, is random and identical elements now split because each undergoes its own independent Brownian 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.

Figure 6 gives a synopsis of the algorithm. Note the use of *cumulative* weights $c_{t-1}^{(j)}$ (constructed in step 3) to achieve efficient sampling in step 1. After any time-step, it is possible to "report" on the current state, for example by evaluating some moment of the state density as shown.

One of the striking properties of the CONDENSATION algorithm is its simplicity, compared with the Kalman filter, despite its generality. Largely this is due to the absence of the Riccati equation which appears in the Kalman filter for the propagation of covariance. The Riccati equation is relatively complex computationally but is not required in the CONDENSATION algorithm which instead deals with variability by sampling, involving the repeated computation of a relatively simple propagation formula.

5 Stochastic dynamical models for curve motion

In order to apply the CONDENSATION algorithm, which is general, to tracking curves in image-streams, specific probability densities must be established both for the dynamics of the object and for the observation process. In the examples described here, x is a linear parameterisation of the curve and allowed transformations of the curve are represented by linear transformations of x. The CONDENSATION algorithm itself does not demand necessarily a *linear* parameterisation though linearity is an attraction for another reason — the availability of algorithms to learn object dynamics. The algorithm could also be used, in principle, with non-linear parameterised kinematics — for instance representing an articulated hand in terms of joint angles (Rehg and Kanade, 1994).

5.1 Linear parameterisations of splines for tracking

We represent the state of a tracked object following methods established for tracking using a Kalman filter (Blake et al., 1995). Objects are modelled as a curve (or set of curves), typically though not necessarily the occluding contour, and represented at time t by a parameterised image curve $\mathbf{r}(s,t)$. The parameterisation is in terms of B-splines, so

$$\mathbf{r}(s,t) = (\mathbf{B}(s)\cdot\mathbf{Q}^x(t),\ \mathbf{B}(s)\cdot\mathbf{Q}^y(t)) \quad \text{for} \quad 0 \le s \le L \tag{8}$$

where $\mathbf{B}(s)$ is a vector $(B_1(s),\ldots,B_{N_B}(s))^T$ of B-spline basis functions, $\mathbf{Q}^x$ and $\mathbf{Q}^y$ are vectors of B-spline control point coordinates and L is the number of spans. It usually desirable (Blake

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
$$p(\mathbf{x}_t | \mathbf{x}_{t-1} = \mathbf{s}_t'^{(n)})$$
to choose each $\mathbf{s}_t^{(n)}$. For instance, in the case that the dynamics are governed by a linear stochastic differential equation, the new sample value may be generated as: $\mathbf{s}_t^{(n)} = \mathbf{A}\mathbf{s}_t'^{(n)} + B\mathbf{w}_t^{(n)}$ where $\mathbf{w}_t^{(n)}$ is a vector of standard normal random variates, and BB^T is the process noise covariance — see section 5.
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
$$\begin{aligned} c_t^{(0)} &= 0, \\ c_t^{(n)} &= c_t^{(n-1)} + \pi_t^{(n)} \quad (n = 1, \ldots, N). \end{aligned}$$

Once the N samples have been constructed: **estimate**, if desired, moments of the tracked position at time-step t as
$$\mathcal{E}[f(\mathbf{x}_t)] = \sum_{n=1}^{N} \pi_t^{(n)} f\left(\mathbf{s}_t^{(n)}\right)$$
obtaining, for instance, a mean position using $f(\mathbf{x}) = \mathbf{x}$.

Figure 6: **The** CONDENSATION **algorithm.**

et al., 1993) to restrict the configuration of the spline to a shape-space of vectors $\mathbf{X}$ defined by

$$\begin{pmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{pmatrix} = W\mathbf{X} + \begin{pmatrix} \overline{\mathbf{Q}}^x \\ \overline{\mathbf{Q}}^y \end{pmatrix}, \tag{9}$$

where the matrix W is a matrix of rank N_X considerably lower than the $2N_B$ degrees of freedom of the unconstrained spline. Typically the shape-space may allow affine deformations of the template shape $\overline{\mathbf{Q}}$, or more generally a space of rigid and non-rigid deformations. The space is constructed by applying an appropriate combination of three methods to build a W-matrix:

1. determining analytically combinations of contours derived from one or more views (Ullman and Basri, 1991; Koenderink and Van Doorn, 1991; Blake et al., 1993), a method that is usable both for affine spaces and for certain classes of articulated object;

2. capturing sequences of key frames of the object in different poses (Blake et al., 1995);

3. performing principal components analysis on a set of outlines of the deforming object (Cootes et al., 1993; Baumberg and Hogg, 1994) to derive a small set of representative contours.

5.2 Dynamical model

Exploiting earlier work on dynamical modelling (Blake et al., 1993; Blake et al., 1995), object dynamics are modelled as a 2nd order process, conveniently represented in discrete time t as a 2nd order linear difference equation:

$$\mathbf{x}_t - \bar{\mathbf{x}} = A(\mathbf{x}_{t-1} - \bar{\mathbf{x}}) + B\mathbf{w}_t \tag{10}$$

where $\mathbf{w}_t$ are independent vectors of independent standard normal variables, the state-vector

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{X}_{t-1} \\ \mathbf{X}_t \end{pmatrix}, \tag{11}$$

and where $\bar{\mathbf{x}}$ is the mean value of the state and A, B are matrices representing the deterministic and stochastic components of the dynamical model respectively. The system is a set of damped oscillators, whose modes, natural frequencies and damping constants are determined by A, driven by random accelerations coupled into the dynamics via B from the noise term $B\mathbf{w}$. While it is possible to set sensible defaults for A, $\bar{\mathbf{x}}$ and B, it is more satisfactory and effective to estimate them from input data taken while the object performs typical motions. Methods for doing this via Maximum Likelihood Estimation are essential to the work described here and are described fully elsewhere (Blake et al., 1995; Reynard et al., 1996).

The dynamical model can be re-expressed in such a way as to make quite clear that it is a temporal Markov chain:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) \propto \exp -\frac{1}{2}\|B^{-1}((\mathbf{x}_t - \overline{\mathbf{x}}) - A(\mathbf{x}_{t-1} - \overline{\mathbf{x}}))\|^2 \tag{12}$$

where $\|\ldots\|$ is the Euclidean norm. It is therefore clear that the learned dynamical models are appropriate for use in the CONDENSATION algorithm.

5.3 Initial conditions

Initial conditions for tracking can be determined by specifying the prior density $p(\mathbf{x}_0)$, and if this is Gaussian, direct sampling can be used to initialise the CONDENSATION algorithm. Alternatively it is possible simply to allow the density $p(\mathbf{x}_t)$ to settle to a steady state $p(\mathbf{x}_\infty)$, in the absence of object measurements. Provided the learned dynamics are stable (free of undamped oscillations) a unique steady state exists. Furthermore, if $p(\mathbf{x}_0)$ is Gaussian, $p(\mathbf{x}_\infty)$ is Gaussian with parameters that can be computed by iterating the Riccati equation (Gelb, 1974). At this point the density function represents an envelope of possible configurations of the object, as learned during the training phase. (Background clutter, if present, will modify and bias this envelope to some extent.) Then, as soon as the foreground object arrives and is measured, the density $p(\mathbf{x}_t)$ begins to evolve appropriately.

6 Observation model

The observation process defined by $p(\mathbf{z}_t|\mathbf{x}_t)$ is assumed here to be stationary in time (though the CONDENSATION algorithm does not necessarily demand this) so a static function $p(\mathbf{z}|\mathbf{x})$ needs to be specified. As yet we have no capability to estimate it from data, though that would be ideal, so some reasonable assumptions must be made. First a measurement model for one-dimensional data with clutter is suggested. Then an extension is proposed for two-dimensional observations that is also used later in computational experiments.

6.1 One-dimensional observations in clutter

In one dimension, observations reduce to a set of scalar positions $\{\mathbf{z} = (z_1, z_2, \ldots, z_M)\}$ and the observation density has the form $p(\mathbf{z}|x)$ where x is one-dimensional position. The multiplicity of measurements reflects the presence of clutter so either one of the events

$$\phi_m = \{\text{true measurement is } z_m\},\ m = 1, \ldots, M$$

occurs, or else the target object is not visible with probability $q = 1 - \sum_m P(\phi_m)$. Such reasoning about clutter and false alarms is commonly used in target tracking (Bar-Shalom and Fortmann, 1988). Now the observation density can be expressed as

$$p(\mathbf{z}|x) = qp(\mathbf{z}|\text{clutter}) + \sum_{m=1}^{M} p(\mathbf{z}|x, \phi_m)P(\phi_m).$$

A reasonable functional form for this can be obtained by making some specific assumptions: that[3] $P(\phi_m) = p,\ \forall\ m$, that the clutter is a Poisson process along the line with spatial density λ and that any true target measurement is unbiased and normally distributed with standard deviation σ. This leads to

$$p(\mathbf{z}|x) \propto 1 + \frac{1}{\sqrt{2\pi}\sigma\alpha} \sum_m \exp -\frac{\nu_m^2}{2\sigma^2} \tag{13}$$

where $\alpha = q\lambda$ and $\nu_m = z_m - x$, and is illustrated in figure 7. Peaks in the density function

[3] There could be some benefit in allowing the $P(\phi_m)$ to vary with m to reflect varying degrees of feature-affinity, based on contrast, colour or orientation.

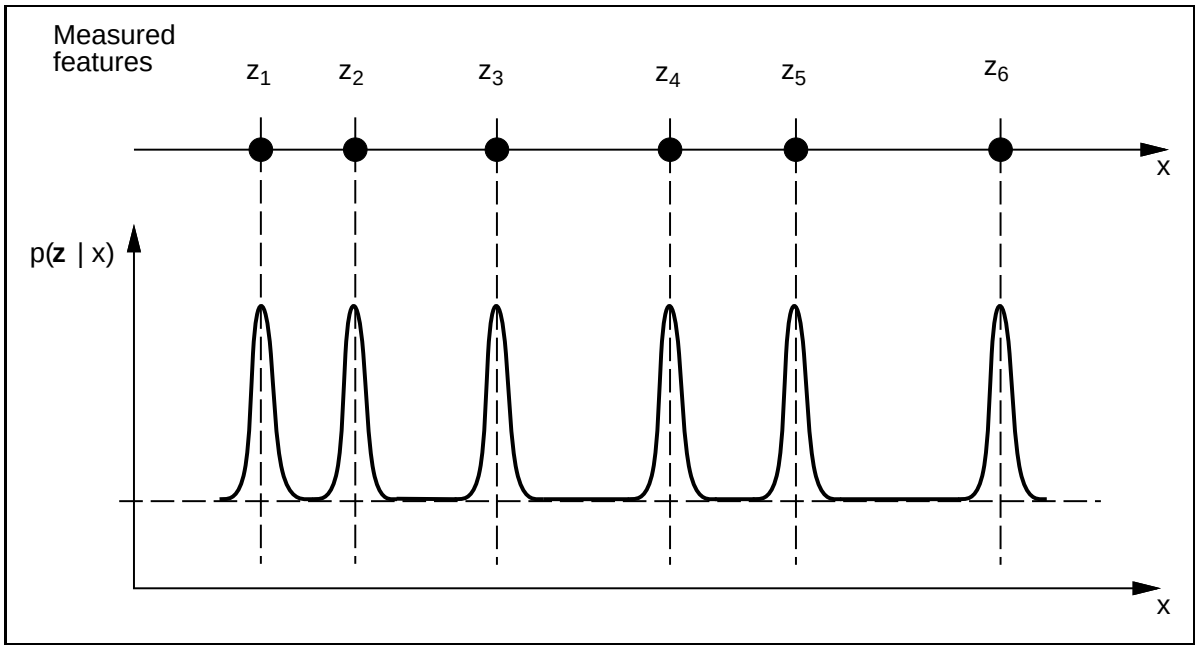


Figure 7: **One-dimensional observation model.** *A probabilistic observation model allowing for clutter and the possibility of missing the target altogether is specified here as a conditional density* $p(\mathbf{z}|x)$.

correspond to measured features and the state density will tend to be reinforced in the CONDENSATION algorithm at such points. The background level reflects the possibility that the true target has not been detected at all. The effect on tracking behaviour is to provide for the possibility of "tunneling": a good hypothesis should survive a transitory failure of observations due, for example, to occlusion of the tracked object. The parameters σ (units of distance) and α (units of inverse distance) must be chosen, though in principle they could be estimated from data by observing measurement error σ and both the density of clutter λ and probability of non-detection q.

Considerable economy can be applied, in practice, in the evaluation of the observation density. Given a hypothesised position x in the "observation" step (figure 6) it is not necessary to attend to all features $z_1, \ldots, z_M$. Any ν_m for which

$$\frac{1}{\sqrt{2\pi}\sigma\alpha} \exp -\frac{\nu_m^2}{2\sigma^2} \ll 1$$

can be neglected and this sets a search window around the position x outside which measurements can be ignored. For practical values of the constants the search window will have a width of a few σ. In practice the clutter is sufficiently sparse and σ is sufficiently small that the search window rarely contains more than one feature.

Note that the density $p(\mathbf{z}|x)$ represents the information about x given a fixed number M of measurements. Potentially, the *event* ψ_M that there are M measurements, regardless of the actual *values* of those measurements, also constitutes information about x. However, we can reasonably assume here that

$$P(\psi_M|x) = P(\psi_M),$$

for instance because x is assumed to lie always within the image window. In that case, by Bayes' theorem,

$$p(x|\psi_M) = p(x)$$

— the event ψ_M provides no additional information about the position x. (If x is allowed also to fall outside the image window then the event ψ_M *is* informative: a value of M well above the mean value for the background clutter enhances the probability that x lies within the window.)

6.2 Two-dimensional observations

In a two-dimensional image, the set of observations $\mathbf{z}$ is, in principle, the entire set of features visible in the image. However, an important aspect of earlier systems in achieving real-time performance (Lowe, 1992; Harris, 1992; Blake et al., 1993) has been the restriction of measurement to a sparse set of lines normal to the tracked curve. These two apparently conflicting ideas can be resolved as follows.

The observation density $p(\mathbf{z}|\mathbf{x})$ in two dimensions describes the distribution of a (linearly) parameterised image curve $\mathbf{z}(s)$, given a hypothetical shape in the form of a curve $\mathbf{r}(s)$, $0 \leq s \leq 1$, represented by a shape parameter $\mathbf{x}$. The two-dimensional density be derived as an extension of the one-dimensional case. It is assumed that a mapping $g(s)$ is known that associates each point $\mathbf{z}(s)$ on the image curve with a point $\mathbf{r}(g(s))$ on the shape. In practice this mapping is set up by tracing normals from the curve $\mathbf{r}$. Note that $g(s)$ is not necessarily injective because $\mathbf{z}(s)$ includes clutter as well as foreground features. Next the one-dimensional density (13) is approximated in a more amenable form that neglects the possibility of more than one feature lying inside the search interval:

$$p(\mathbf{z}|x) \propto \exp -\frac{1}{2\sigma^2} f(\nu_1; \mu) \quad \text{where} \quad f(\nu; \mu) = \min(\nu^2, \mu^2), \tag{14}$$

$\mu = \sqrt{2}\sigma \log(1/\sqrt{2\pi}\alpha\sigma)$ is a spatial scale constant, and ν_1 is the ν_m with smallest magnitude, representing the feature lying closest to the hypothesised position x. A natural extension to two dimensions is then

$$p(\mathbf{z}|\mathbf{x}) = Z \exp -\frac{1}{2r} \int_0^L f(\mathbf{z}_1(s) - \mathbf{r}(s); \mu)\, \mathrm{d}s \tag{15}$$

in which r is a variance constant and $\mathbf{z}_1(s)$ is the closest associated feature to $\mathbf{r}(s)$:

$$\mathbf{z}_1(s) = \mathbf{z}(s') \quad \text{where} \quad s' = \arg \min_{s' \in g^{-1}(s)} |\mathbf{r}(s) - \mathbf{z}(s')|.$$

Note that the constant of proportionality ("partition function") $Z(\mathbf{x})$ is an unknown function. We make the assumption that the variation of Z with $\mathbf{x}$ is slow compared with the other term in (15) so that Z can be treated as constant. It remains to establish whether this assumption is justified.

The observation density (15) can be computed via a discrete approximation, the simplest being:

$$p(\mathbf{z}|\mathbf{x}) \propto \exp \left\{ - \sum_{m=1}^{M} \frac{1}{2rM} f(\mathbf{z}_1(s_m) - \mathbf{r}(s_m); \mu) \right\}, \tag{16}$$

where $s_m = m/M$. This is simply the product of one-dimensional densities (14) with $\sigma = \sqrt{rM}$, evaluated independently along M curve normals as in figure 8.

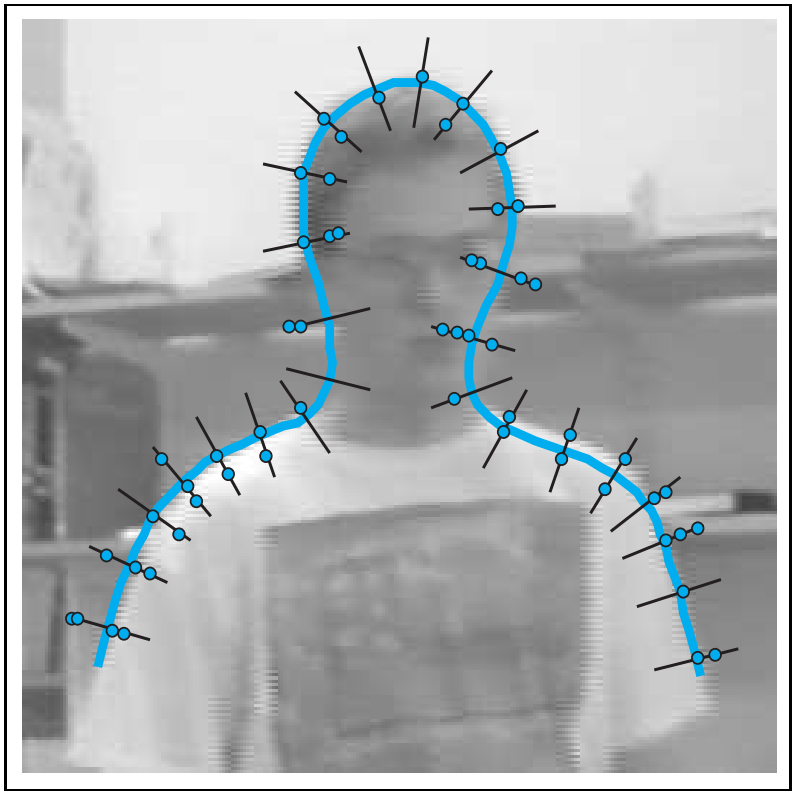


Figure 8: **Observation process.** *The thick line is a hypothesised shape, represented as a parametric spline curve. The spines are curve normals along which high-contrast features (white crosses) are sought.*

7 Applying the CONDENSATION algorithm to video-streams

Four examples are shown here of the practical efficacy of the CONDENSATION algorithm. Movie (MPEG) versions of some results are available on the web at `http://www.robots.ox.ac.uk/~ab/`.

7.1 Tracking a multi-modal distribution

The ability of the CONDENSATION algorithm to represent multi-modal distributions was tested using a 70 frame (2.8 second) sequence of a cluttered room containing three people each facing the camera (figure 9). One of the people moves from right to left, in front of the other two. The

Figure 9: **Tracking three people in a cluttered room.** *The first frame of a sequence in which one figure moves from right to left in front of two stationary figures.*

shape-space for tracking is built from a hand-drawn template of head and shoulders (figure 8) which is then allowed to deform via planar affine transformations . A Kalman filter contour-tracker (Blake et al., 1993) with default motion parameters is able to track a single moving person just well enough to obtain a sequence of outline curves that is usable as training data. Given the high level of clutter, adequate performance with the Kalman filter is obtained here by means of background modelling (Rowe and Blake, 1996), a statistical form of background subtraction, which effectively removes clutter from the image data before it is tracked. It transpires, for this particular training set, that the learned motions comprise primarily horizontal translation, with vertical translation and horizontal and vertical shear present to a lesser degree.

The learned shape and motion model can now be installed as $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ in the CONDENSATION algorithm which is now run on a test sequence but *without* the benefit of background modelling, so that the background clutter is now visible to the tracker. Figure 10 shows how the state-density evolves as tracking progresses. Initialisation is performed simply by iterating the stochastic model, in the absence of measurements, to its steady state and it can be seen that this corresponds, at time 0, to a roughly Gaussian distribution, as expected. The distribution rapidly

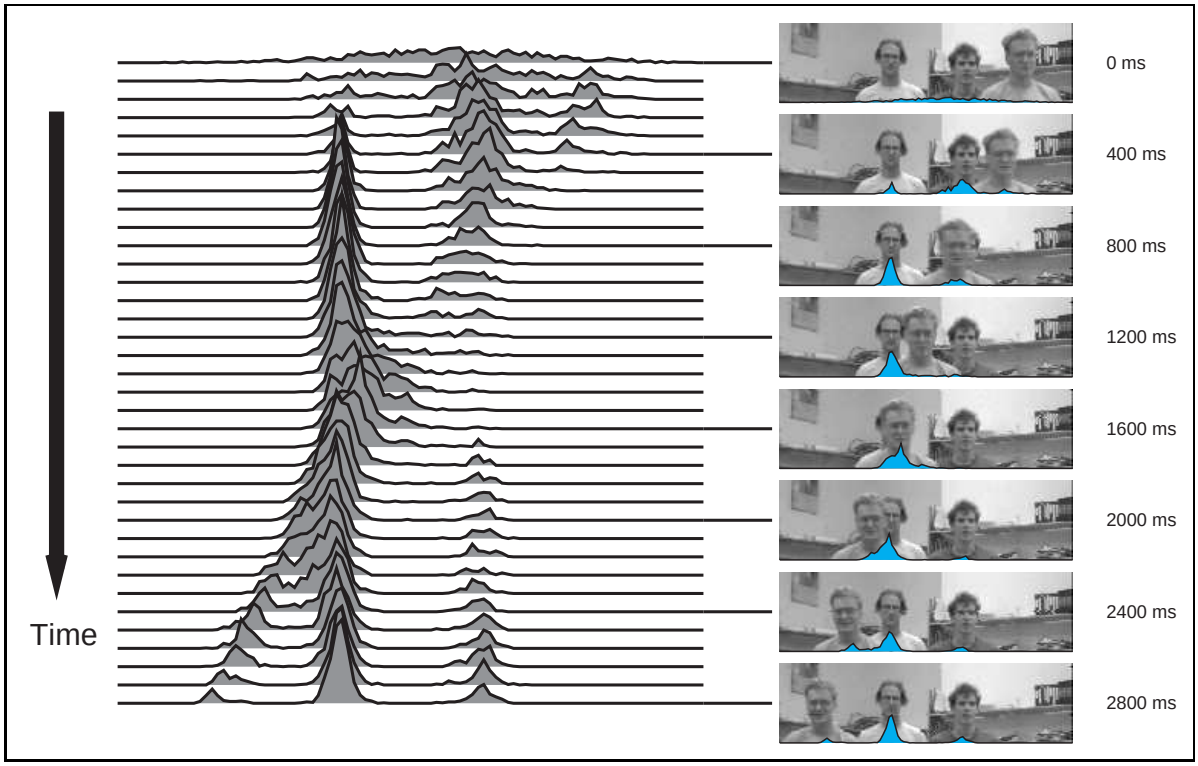


Figure 10: **Tracking with multi-modal state-density.** *An approximate depiction of the state-density is shown, computed by smoothing the distribution of point masses* $\mathbf{s}_t^{(1)}, \mathbf{s}_t^{(2)}, \ldots$ *in the* CONDENSATION *algorithm. The density is, of course, multi-dimensional; its projection onto the horizontal translation axis is shown here. The initial distribution is roughly Gaussian but this rapidly evolves to acquire peaks corresponding to each of the three people in the scene. The right-most peak drifts leftwards, following the moving person, coalescing with and separating from the other two peaks as it moves. Having specified a tracker for* one *person we effectively have, for free, a multi-person tracker, owing to the innate ability of the* CONDENSATION *algorithm to maintain multiple hypotheses.*

collapses down to three peaks which are then maintained appropriately even during temporary occlusion. Although the tracker was designed to track just one person, the CONDENSATION algorithm allows the tracking of all three, for free; the ability to represent multi-modal distributions effectively provides multiple hypothesis capability. Tracking is based on frame rate (40 ms) sampling in this experiment and distributions are plotted in the figure for alternate frames. The experiment was run using a distribution of $N = 1000$ samples per time-step.

7.2 Tracking rapid motions through clutter

The ability to track more agile motion, still against clutter, was tested using a 500 field (10 second) sequence of a girl dancing vigorously to a Scottish reel. The shape-space for tracking was planar affine, based on a hand-drawn template curve for the head outline. The training sequence consisted of dancing against a largely uncluttered background, tracked by a Kalman filter contour-tracker with default dynamics to record 140 fields (2.8 seconds) of tracked head positions, the most that could be tracked before losing lock. Those 140 fields were sufficient to learn a bootstrap motion model which then allowed the Kalman filter to track the training data for 800 fields (16 seconds) before loss of lock. The motion model obtained from these 800 fields was used in experiments with the CONDENSATION tracker and applied to the test data, now including clutter.

Figure 11 shows some stills from the test sequence, with a trail of preceding head positions to indicate motion. The motion is primarily translation, with some horizontal shear apparent as the dancer turns her head. Representing the state density with $N = 100$ samples at each time-step proves just sufficient for successful tracking. As in the previous example, a prior density can be computed as the steady state of the motion model and, in this case, that yields a prior for position that spreads across most of the image area, as might be expected given the range of the dance. Such a broad distribution cannot effectively be represented by just $N = 100$ samples. One alternative is to increase N in the early stages of tracking, and this is done in a later experiment. Alternatively, the prior can be based on a narrower distribution whose centre is positioned by hand over the object at time 0, and that is what was done here. Observation parameters were $\mu = 24,\ \sigma = 7$ with $M = 18$ normals.

Figure 12 shows the motion of the centroid of the estimated head position as tracked both by the CONDENSATION algorithm and by a Kalman filter using the same motion model. The CONDENSATION tracker correctly estimated head position throughout the sequence, but after about 40 fields (0.80 s), the Kalman filter was distracted by clutter, never to recover.

Given that there is only one moving person in this experiment, unlike the previous one in which there were three, it might seem that a unimodal representation of the state density should suffice. This is emphatically not the case. The facility to represent multiple modes is crucial to robustness as figure 13 illustrates. The figure shows how the distribution becomes misaligned (at 900ms), reacting to the distracting form of the computer screen. After a further 20ms the distribution splits into two distinct peaks, one corresponding to clutter (the screen), one to the dancer's head. At this point the clutter peak actually has the higher posterior probability — a unimodal tracker, for instance a Kalman filter, would almost certainly discard the lower peak, rendering it unable to recover. The CONDENSATION algorithm however, capable as it is of carrying several hypotheses simultaneously, does recover rapidly as the clutter peak decays for lack of confirmatory observation, leaving just one peak corresponding to the dancer at 960 ms.

Figure 11: **Tracking agile motion in clutter.** *The test sequence consists of 500 fields (10 seconds) of agile dance against a cluttered background. The dancer's head is tracked through the sequence. Several representative fields are shown here, each with a trail of successive mean tracked head positions at intervals of 40 ms. The* CONDENSATION *algorithm used* $N = 100$ *samples per time-step to obtain these results.*

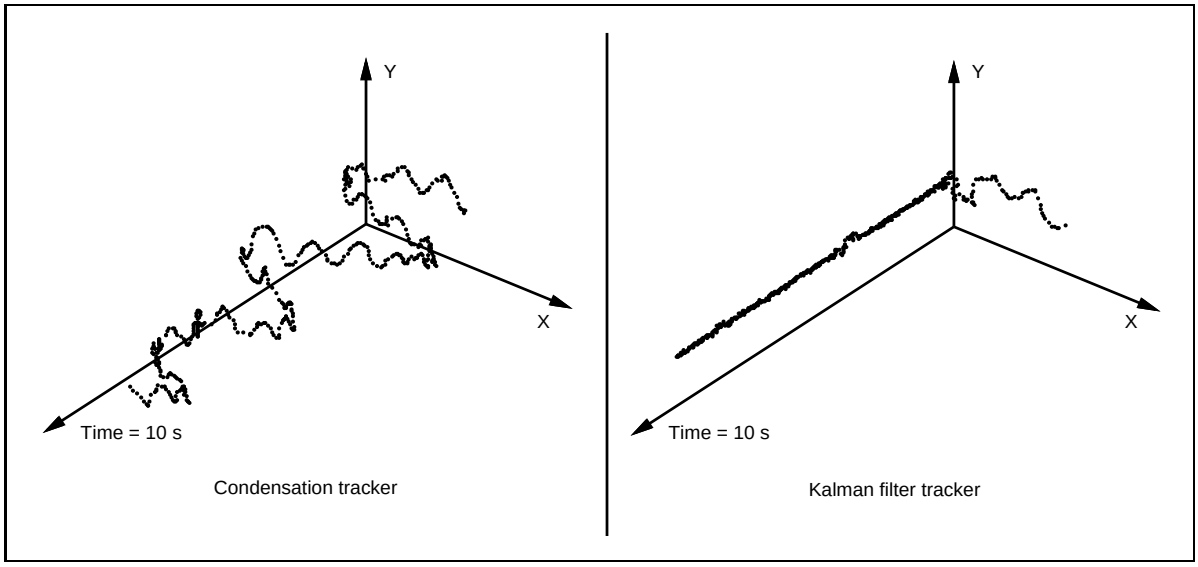


Figure 12: **The Condensation tracker succeeds where a Kalman filter fails.** *The estimated centroid for the sequence shown in figure 11 is plotted against time for the entire 500 field sequence, as tracked first by the* CONDENSATION *tracker, then by a comparable Kalman filter tracker. The* CONDENSATION *algorithm correctly estimates the head position throughout the sequence. The Kalman filter tracks briefly, but is soon distracted by clutter and never recovers.*

7.3 Tracking an articulated object

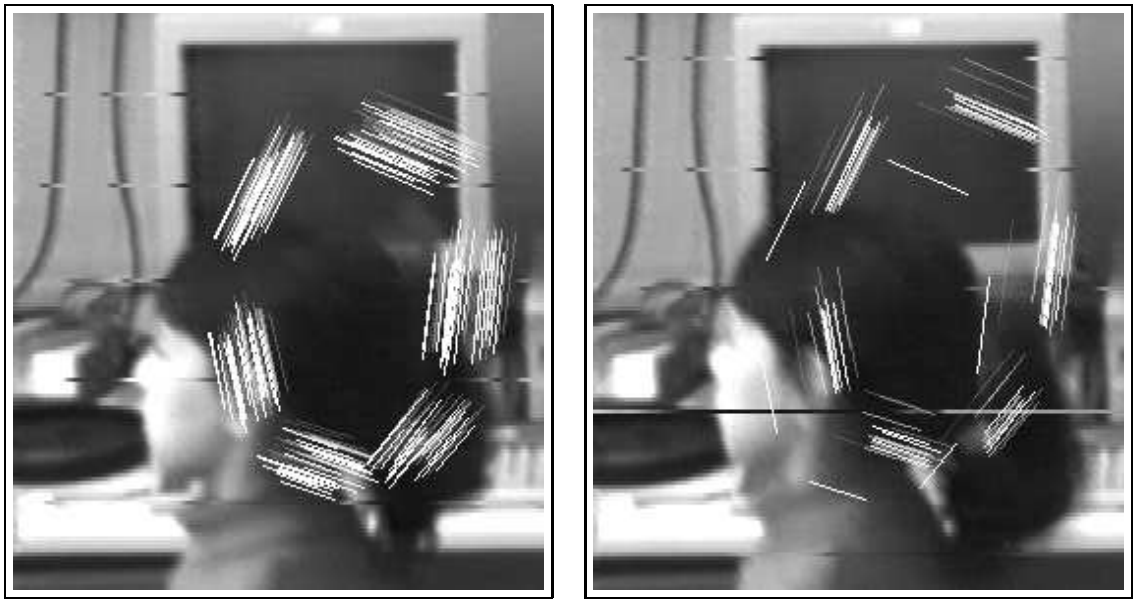
The preceding sequences show motion taking place in affine shape-spaces of just 6 dimensions. High dimensionality is one of the factors, in addition to agility and clutter, that makes tracking hard (Blake et al., 1993). In order to investigate tracking performance in higher dimensions, we used a 500 field (10 second) test sequence of a hand translating, rotating, and flexing its fingers independently, over a highly cluttered desk scene (figure 14). Figure 15 shows just how severe the clutter problem is — the hand is immersed in a dense field of edges.

A model of shape and motion model was learned from training sequences of hand motion against a plain background, tracked by Kalman filter (using signed edges to help to disambiguate finger boundaries). The procedure comprised several stages, creative assembly of methods from the available "toolkit" for learning (Blake et al., 1995).

1. **Shape-space** was constructed from 6 templates drawn around the hand with the palm in a fixed orientation and with the fingers and thumb in various configurations. The 6 templates combined linearly to form a 5-dimensional space of deformations which were then added to the space of translations to form a 7 dimensional shape-space.

2. **Default dynamics** in the shape-space above were adequate to track a clutter-free training sequence of 600 frames in which the palm of the hand maintained an approximately fixed attitude.

3. **Principal components analysis:** the sequence of 600 hand outlines was replicated with each hand contour rotated through 90 degrees and the sequences concatenated to give a



field 45 (900 ms) field 46 (920 ms)

field 47 (940 ms) field 48 (960 ms)

Figure 13: **Recovering from tracking failure.** *Detail from 4 consecutive fields of the sequence illustrated in figure 11. Each sample from the distribution is plotted on the image, with intensity scaled to indicate its posterior probability. (Most of the* $N = 100$ *samples have too low a probability to be visible in this display.) In field 45, the distribution is misaligned, and has begun to diverge. In fields 46 and 47 it has split into two distinct peaks, the larger attracted to background clutter, but converges back onto the dancer in field 48.*

Figure 14: **A hand moving over a cluttered desk.** *Field 0 of a 500 field (10 second) sequence in which the hand translates, rotates, and the fingers and thumb flex independently.*

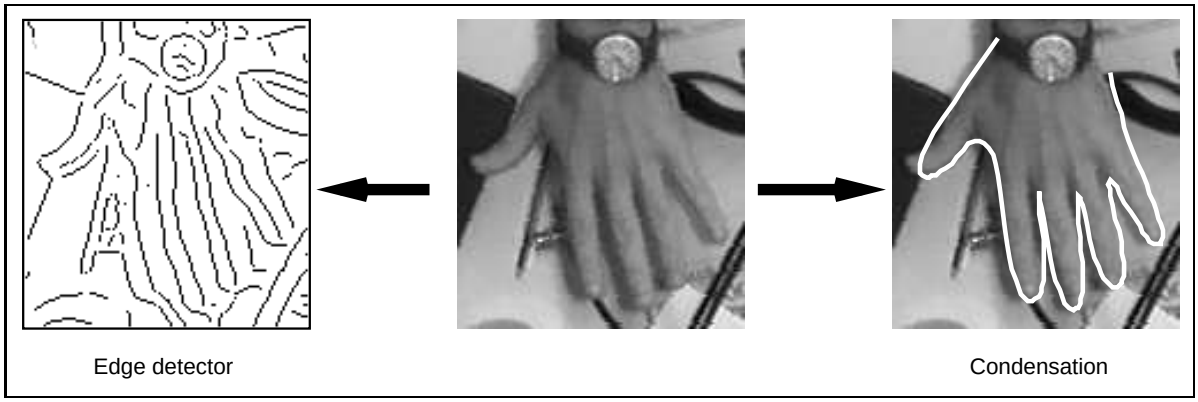


Figure 15: **Severe clutter.** *Detail of one field (figure 14) from the test-sequence shows the high level of potential ambiguity. Output from a directional Gaussian edge detector shows that there are many clutter edges present as potential distractors.*

sequence of 1200 deformations. Projecting out the translational component of motion, the application of Principal Component Analysis (PCA) to the sequence of residual deformations of the 1200 contours established a 10-dimensional space that accounted almost entirely for deformation. This was then combined with the translational space to form a 12-dimensional shape-space that accounted both for the flexing of fingers and thumb and also for rotations of the palm.

4. **Bootstrapping:** a Kalman filter with default dynamics in the 12-dimensional shape-space was sufficient to track a training sequence of 800 fields of the hand translating, rotating, and flexing fingers and thumb slowly. This was used to learn a model of motion.

5. **Re-learning:** that motion model was installed in a Kalman filter used to track another, faster training-sequence of 800 fields. This allowed a model for more agile motion to be learned, which was then used in experiments with the CONDENSATION tracker.

Figure 16: **Tracking a flexing hand across a cluttered desk.** *Representative stills from a 500 field (10 second) sequence show a hand moving over a highly cluttered desk scene. The fingers and thumb flex independently, and the hand translates and rotates. Here the* CONDENSATION *algorithm uses* $N = 1500$ *samples per time-step initially, dropping gradually over 4 fields to* $N = 500$ *for the tracking of the remainder of the sequence. The mean configuration of the contour is displayed.*

Figure 16 shows detail of a series of images from a tracked, 500 field test-sequence. The initial state density was simply the steady state of the motion model, obtained by allowing the filter to iterate in the absence of observations. Tracker initialisation was facilitated by using more samples per time-step ($N = 1500$) at time $t = 0$, falling gradually to 500 over the first 4 fields. The rest of the sequence was tracked using $N = 500$. As with the previous example of the dancer, clutter can distract the tracker but the ability to represent multi-modal state density means that tracking can recover.

7.4 Tracking a camouflaged object

Finally, we tested the ability of the algorithm to track rapid motion against background distraction in the extreme case that background objects actually mimiced the tracked object. A 12 second (600 field) sequence showed 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 was not available, the motion model was again learned by means of a bootstrap procedure. A tracker with default dynamics proved capable of tracking the first 150 fields of a training sequence before losing the leaf, and those tracked positions allowed a first approximation to the model to be learned. Installing that in a CONDENSATION tracker, the entire sequence could be tracked, though with occasional misalignments. Finally a third learned model was sufficient to track accurately the entire 12-second training sequence. Despite occasional violent gusts of wind and temporary obscuration by another leaf, the CONDENSATION algorithm successfully followed the object, as figure 17 shows. In fact, tracking is accurate enough using $N = 1200$ samples to

Figure 17: **Tracking with camouflage.** *The aim is to track a single camouflaged moving leaf in this 12-second sequence of a bush blowing in the wind. Despite the heavy clutter of distractors which actually mimic the foreground object, and occasional violent gusts of wind, the chosen foreground leaf is successfully tracked throughout the sequence. Representative stills depict mean contour configurations, with preceding tracked leaf positions plotted at 40 ms intervals to indicate motion.*

separate the foreground leaf from the background reliably, an effect which can otherwise only be achieved using "blue-screening". Having obtained the model iteratively as above, independent

test sequences could be tracked without further training. With $N = 1200$ samples per time-step the tracker runs at 6.5 Hz on a SGI Indy SC4400 200MHz workstation. Reducing this to $N = 200$ increases processing speed to video frame-rate (25 Hz), at the cost of occasional misalignments in the mean configuration of the contour. Observation parameters were $\mu = 8$, $\sigma = 3$ with $M = 21$ normals.

8 Conclusions

Tracking in clutter is hard because of the essential multi-modality of the conditional observation density $p(\mathbf{z}|\mathbf{x})$. In the case of curves multiple-hypothesis tracking is inapplicable and a new approach is needed. The CONDENSATION algorithm is a fusion of the statistical factored sampling algorithm for static, non-Gaussian problems with a stochastic model for object motion. The result is an algorithm for tracking rigid and non-rigid motion which has been demonstrated to be far more effective in clutter than comparable Kalman filters. Performance of the CONDENSATION algorithm improves as the sample size parameter N increases; formally computational complexity is $O(N \log N)$, although this can be made $O(N)$ with a minor modification to the sampling procedure. Impressive results have been demonstrated for models with between 6 and 12 degrees of freedom, even when N is as low as 100–200. Performance in several cases was improved still further with an increased value $N \approx 1000$. In a 6-dimensional shape-space, the system currently runs with $N = 100$ in real-time (50Hz) on a desk-top graphics workstation (SGI Indy R4400SC, 200 MHz).

The new approach raises a number of questions. First, alternative observation models could be explored in order to make greater use of image intensity variations, though without sacrificing too much in the way of photometric invariance. It is to be hoped in the interests of efficiency that, as happens with the search window in the edge-based case, computational attention could be concentrated in a band around the hypothesised curve without significant loss of accuracy in the model. Such a model would have echoes of correlation matching but of course without the exhaustive search characteristic of correlation matchers which is quite infeasible in more than two or three dimensions.

Secondly, the availability of general state densities suggests the need for more general representations of those densities. When the density is approximately unimodal, first and second moments may be adequate to convey the likely states, but in the multi-modal case, as for example when several people are tracked simultaneously, the mean configuration is not a particularly useful statistic — it meaninglessly combines the configurations of the three people. An alternative is to attempt to develop a mode finder capable of pin-pointing several modes when present. More generally there is a need for "operators" to interrogate densities: for instance, an operator to find a person moving to the right, or to find the tallest person. Perhaps such operators could be formulated as hypothesis tests applied to sample sets.

A third question concerns the random sampling scheme and its efficiency. Factored sampling can be inefficient as the modes of $p(\mathbf{z}|\mathbf{x})$ become narrow. One approach is "importance sampling" (Ripley, 1987) in which a heuristically chosen distribution, approximating $p(\mathbf{z}|\mathbf{x})$, is used to concentrate random sampling around modes. However, this has the drawback that the prior $p(\mathbf{x})$ must be repeatedly evaluated whereas, in temporal propagation, the prior (prediction) $p(\mathbf{x}_t|\mathbf{z}_{t-1})$ cannot be evaluated pointwise, only sampled.

Fourthly and finally, it is striking that the density propagation equation (4) in the CONDENSATION algorithm is a continuous form of the propagation rule of the "forward algorithm"

for Hidden Markov Models (HMMs) (Rabiner and Bing-Hwang, 1993). The integral over continuous states in (5) becomes a summation over discrete states in the HMM, with $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ represented by a transition matrix. This suggests a natural opportunity to combine the two so that mixed discrete/continuous states could be propagated over time. This would allow switching between multiple models, for instance walk-trot-canter-gallop, each model represented by a stochastic differential equation, with transitions governed by a discrete conditional probability matrix. It seems likely that such a system could be executed as a CONDENSATION tracker. A further challenge is to develop a learning algorithm for mixed dynamical models.

Acknowledgements

The authors would like to acknowledge the support of the EPSRC. They are also grateful for discussions with Roger Brockett, John Kent, Ian Reid, David Mumford, Brian Ripley, David Reynard, Simon Rowe, Andrew Wildenberg and Andrew Zisserman, and for experimental assistance from Sarah Blake.

References

Anderson and Moore (1979). *Optimal filtering.* Prentice Hall.

Astrom, K. J. (1970). *Introduction to stochastic control theory.* Academic Press.

Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and Data Association.* Academic Press.

Bartels, R., Beatty, J., and Barsky, B. (1987). *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann.

Baumberg, A. and Hogg, D. (1994). Learning flexible models from image sequences. In Eklundh, J.-O., editor, *Proc. 3rd European Conference on Computer Vision*, 299–308. Springer–Verlag.

Baumberg, A. and Hogg, D. (1995). Generating spatiotemporal models from examples. In *Proc. of the British Machine Vision Conference*, 2, 413–422.

Blake, A., Curwen, R., and Zisserman, A. (1993). A framework for spatio-temporal control in the tracking of visual contours. *Int. Journal of Computer Vision*, 11, 2, 127–145.

Blake, A. and Isard, M. (1994). 3D position, attitude and shape input using video tracking of hands and lips. In *Proc. Siggraph*, 185–192. ACM.

Blake, A., Isard, M., and Reynard, D. (1995). Learning to track the visual motion of contours. *Artificial Intelligence*, 78, 101–134.

Bucy, R. (1969). Bayes theorem and digital realizations for non-linear filters. *J. Astronautical Sciences*, 17, 2, 80–94.

Cipolla, R. and Blake, A. (1990). The dynamic analysis of apparent contours. In *Proc. 3rd Int. Conf. on Computer Vision*, 616–625.

Cootes, T., Taylor, C., Lanitis, A., Cooper, D., and Graham, J. (1993). Building and using flexible models incorporating grey-level information. In *Proc. 4th Int. Conf. on Computer Vision*, 242–246.

Dickmanns, E. and Graefe, V. (1988). Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1, 241–261.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24, 381–95.

Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE. Trans. Computers*, C-22, 1.

Gelb, A., editor (1974). *Applied Optimal Estimation.* MIT Press, Cambridge, MA.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6, 6, 721–741.

Gennery, D. (1992). Visual tracking of known three-dimensional objects. *Int. Journal of Computer Vision*, 7:3, 243–270.

Goodwin, C. and Sin, K. (1984). *Adaptive filtering prediction and control.* Prentice-Hall.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proc. F*, 140, 2, 107–113.

Grenander, U., Chow, Y., and Keenan, D. M. (1991). *HANDS. A Pattern Theoretical Study of Biological Shapes.* Springer-Verlag. New York.

Hager, G. (1990). *Sensor fusion and planning: a computational approach.* Kluwer Academic Publishers.

Harris, C. (1992). Tracking with rigid models. In Blake, A. and Yuille, A., editors, *Active Vision*, 59–74. MIT.

Hogg, D. (1983). Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1, 1, 5–20.

Huttenlocher, D., Noh, J., and Rucklidge, W. (1993). Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, 93–101.

Isard, M. and Blake, A. (1996). Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. on Computer Vision*, 343–356, Cambridge, England.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, 259–268.

Kitagawa, G. (1996). Monte-carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Computational and Graphical Statistics*, 5, 1, 1–25.

Koenderink, J. and Van Doorn, A. (1991). Affine structure from motion. *J. Optical Soc. of America A.*, 8, 2, 337–385.

Lowe, D. (1991). Fitting parameterised 3D models to images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:5, 441–450.

Lowe, D. (1992). Robust model-based motion tracking through the integration of search and estimation. *Int. Journal of Computer Vision*, 8, 2, 113–122.

Matthies, L. H., Kanade, T., and Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *Int. Journal of Computer Vision*, 3, 209–236.

Menet, S., Saint-Marc, P., and Medioni, G. (1990). B-snakes: implementation and application to stereo. In *Proceedings DARPA*, 720–726.

Miller, M., Srivasta, A., and Grenander, U. (1995). Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition. *IEEE Transactions on Signal Processing*, 43, 11, 2678–2690.

Papoulis, A. (1990). *Probability and Statistics.* Prentice-Hall.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1988). *Numerical Recipes in C.* Cambridge University Press.

Rabiner, L. and Bing-Hwang, J. (1993). *Fundamentals of speech recognition.* Prentice-Hall.

Rao, B. (1992). Data association methods for tracking systems. In Blake, A. and Yuille, A., editors, *Active Vision*, 91–105. MIT.

Rao, C. (1973). *Linear Statistical Inference and Its Applications.* John Wiley and Sons, New York.

Rehg, J. and Kanade, T. (1994). Visual tracking of high dof articulated structures: an application to human hand tracking. In Eklundh, J.-O., editor, *Proc. 3rd European Conference on Computer Vision*, 35–46. Springer–Verlag.

Reynard, D., Wildenberg, A., Blake, A., and Marchant, J. (1996). Learning dynamics of complex motions from image sequences. In *Proc. 4th European Conf. on Computer Vision*, 357–368, Cambridge, England.

Ripley, B. (1987). *Stochastic simulation.* New York: Wiley.

Ripley, B. and Sutherland, A. (1990). Finding spiral structures in images of galaxies. *Phil. Trans. R. Soc. Lond. A.*, 332, 1627, 477–485.

Rowe, S. and Blake, A. (1996). Statistical feature modelling for active contours. In *Proc. 4th European Conf. on Computer Vision*, 560–569, Cambridge, England.

Sorenson, H. W. and Alspach, D. L. (1971). Recursive bayesian estimation using gaussian sums. *Automatica*, 7, 465–479.

Storvik, G. (1994). A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16, 10, 976–986.

Sullivan, G. (1992). Visual interpretation of known objects in constrained scenes. *Phil. Trans. R. Soc. Lond. B.*, 337, 361–370.

Terzopoulos, D. and Metaxas, D. (1991). Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, 7.

Terzopoulos, D. and Szeliski, R. (1992). Tracking with Kalman snakes. In Blake, A. and Yuille, A., editors, *Active Vision*, 3–20. MIT.

Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, 10, 992–1006.

Yuille, A. and Hallinan, P. (1992). Deformable templates. In Blake, A. and Yuille, A., editors, *Active Vision*, 20–38. MIT.

A Non-linear filtering

There are four distinct probability distributions represented in a non-linear Bayesian filter. Three of them form part of the problem specification and the fourth constitutes the solution. The three specified distributions are:

1. the prior density $p(\mathbf{x})$ for the state $\mathbf{x}$
2. the process density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ that describes the stochastic dynamics
3. the observation density $p(\mathbf{z}|\mathbf{x})$

and the filter evolves over time to generate, as the solution at each time-step, the state-density $p_t(\mathbf{x})$ where $p_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t|\mathcal{Z}_t)$. Only when all of the three specified distributions are Gaussian is the state-density p_t also Gaussian. Otherwise, for non-Gaussian p_t, it is possible to use one of a number of approximate filters, depending on which of the specified densities it is that is non-Gaussian.

A.1 Non-Gaussian prior density

The case that the prior density is non-Gaussian is the simplest to deal with provided only that it can adequately be represented (or approximated) as an additive Gaussian mixture:

$$p_0(\mathbf{x}) = \sum_{m=1}^{M} w^{(m)} G(\mathbf{x}; \mu^{(m)}, P_0^{(m)}).$$

In that case, provided that other specified densities are Gaussian, the state density can also be represented as a corresponding mixture

$$p_t(\mathbf{x}) = \sum_{m=1}^{M} w^{(m)} G(\mathbf{x}; \mu_t^{(m)}, P_t^{(m)})$$

in which the means $\mu_t^{(m)}$ and variances $P_t^{(m)}$ vary over time but the weights $w^{(m)}$ are fixed. Each of the M mixture components evolves as an independent Gaussian so that, in fact, the state density is just a sum of densities from M independent linear Kalman filters.

A.2 Non-Gaussian process density

Non-Gaussian state densities can arise from the nature of the process either because the dynamics are driven by non-Gaussian process noise, or, more generally, because the deterministic dynamics are non-linear. One approach to filtering is then to approximate the dynamics by Taylor expansion as a linear process with time-varying coefficients and proceed as for linear Kalman filters. This generates a Gaussian representation of the evolving state-density which may be a good approximation depending on the nature of the non-linearity. This is the basis of the "Extended Kalman Filter" (EKF) (Gelb, 1974; Bar-Shalom and Fortmann, 1988). Alternatively, one can attempt a mixture representation, as earlier, but now allowing the weights $w^{(m)}$ also to vary over time. Unfortunately, even allowing dynamic re-weighting (Sorenson and Alspach, 1971) does not produce exact solutions for $p_t(\mathbf{x})$, because the individual Gaussian components do not remain Gaussian over time. For example, consider the case in which the process

density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is itself an additive mixture of $k > 1$ Gaussian components. According to the Bayesian propagation equation (5) each component of p_t splits into k separate components in the transition from time n to time $n+1$; the total number of components in p_t grows exponentially as k^t. Clearly p_t must be approximated at each time-step to prune back the number of components (Anderson and Moore, 1979) within some resource-limited bound M. Effectively there are Mk full Kalman filters running at each time-step, each bringing the computational expense of its own Riccati equation step. Clearly the success of this approach depends on how well the densities p_t and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ can be approximated with a modest number Mk of components.

A.3 Non-Gaussian observation density

In the case of visual tracking in clutter, non-linearity of the tracking filter arises, as we have seen, because the observation density $p(\mathbf{z}|\mathbf{x})$ is non-Gaussian and, furthermore, is multi-modal so that it cannot be well approximated by a single Gaussian. Each of the methods just mentioned for handling non-Gaussian process density, the EKF and Gaussian mixtures, are relevant also to non-Gaussian observation density but continue to have the same drawbacks. Note that, in the case of Gaussian mixtures, the number of mixture components again proliferates at each time-step of (4), albeit via a different mechanism involving products of Gaussians rather than convolutions. Even this assumes that the observation density can be approximated as a mixture but in clutter this becomes rather inefficient, requiring at least one component per visible feature.

There is an additional class of techniques which applies to this case when the non-Gaussian state density arises from clutter of a particular sort. In the simplest case, one of a finite set of measurements $\mathbf{z}_t = \{z_{t,1}, \ldots, z_{t,k}\}$ at time n is to be associated with the state $\mathbf{x}_t$ at time t. Heuristic mechanisms such as the validation gate and the probabilistic data-association filter (PDAF) (Bar-Shalom and Fortmann, 1988) attempt to deal with the ambiguity of association. Alternatively it can, in principle, be dealt with exactly by "multiple hypothesis filtering" but with computational cost that grows exponentially over time and which is therefore ruled out in practice. The "RANSAC" algorithm (Fischler and Bolles, 1981) deals probabilistically with multiple observations but the observations have to be discrete, and there is no mechanism for temporal propagation. More complex methods including the Joint PDAF (JPDAF) (Bar-Shalom and Fortmann, 1988; Rao, 1992) address the more difficult problem of associating not simply single features but subsequences of $\mathcal{Z}_t$ with the state. However, these methods rely on the existence of discrete features. In contour tracking the features are continuous curves and so are not naturally amenable to discrete association.

A.4 Direct integration

Finally, one very general approach to nonlinear filtering must be mentioned. This is simply to integrate (5) directly, using a suitable numerical representation of the state density such as finite elements. This in essence is what (Bucy, 1969) proposed and more recently (Hager, 1990) investigated with respect to robotics applications. It is usable in one or two dimensions but, complexity being exponential in the dimension, is altogether infeasible for problems of dimension around 6–20, typical of the tracking problems dealt with here. The CONDENSATION algorithm is designed to offer a viable alternative.

B Derivation of the sampling rule

The correctness of the sampling rule (4) on page 6 is proved by first deriving two lemmas from the independence assumption (2). (This is similar to the derivation found in (Bar-Shalom and Fortmann, 1988), except that our independence assumptions are explicitly specified.)

Lemma 1

$$p(\mathbf{z}_t|\mathcal{X}_t, \mathcal{Z}_{t-1}) = p(\mathbf{z}_t|\mathbf{x}_t).$$

Proof:

$$\begin{aligned} p(\mathcal{Z}_t|\mathcal{X}_t) &= p(\mathbf{z}_t, \mathcal{Z}_{t-1}|\mathcal{X}_t) \\ &= p(\mathbf{z}_t|\mathcal{Z}_{t-1}, \mathcal{X}_t)p(\mathcal{Z}_{t-1}|\mathcal{X}_t) \\ &= p(\mathbf{z}_t|\mathcal{Z}_{t-1}, \mathcal{X}_t) \prod_{i=1}^{t-1} p(\mathbf{z}_i|\mathbf{x}_i). \end{aligned}$$

(Taking (3) at time t and integrating w.r.t. $\mathbf{z}_t$ yields the reduction of the second term in line 2.) Now, using (3) again gives the result.

Lemma 2

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}, \mathcal{Z}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

Proof:

$$p(\mathbf{x}_t, \mathcal{Z}_{t-1}|\mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathcal{X}_{t-1})p(\mathcal{Z}_{t-1}|\mathcal{X}_{t-1})$$

from (2) so

$$p(\mathbf{x}_t|\mathcal{Z}_{t-1}, \mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}),$$

using the Markov assumption (1).

Derivation of the propagation formula: consider

$$\begin{aligned} p(\mathcal{X}_t|\mathcal{Z}_t) &= \frac{p(\mathbf{z}_t|\mathcal{X}_t, \mathcal{Z}_{t-1})p(\mathcal{X}_t|\mathcal{Z}_{t-1})}{p(\mathbf{z}_t|\mathcal{Z}_{t-1})} \\ &= k_t\ p(\mathbf{z}_t|\mathcal{X}_t, \mathcal{Z}_{t-1})p(\mathcal{X}_t|\mathcal{Z}_{t-1}) \\ &= k_t\ p(\mathbf{z}_t|\mathbf{x}_t)p(\mathcal{X}_t|\mathcal{Z}_{t-1}) \quad \text{using lemma 1.} \end{aligned}$$

Now integrating w.r.t. $\mathcal{X}_{t-1}$ gives

$$p(\mathbf{x}_t|\mathcal{Z}_t) = k_t\ p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{Z}_{t-1}).$$

The last term can be expanded:

$$\begin{aligned} p(\mathbf{x}_t|\mathcal{Z}_{t-1}) &= \int_{\mathcal{X}_{t-1}} p(\mathbf{x}_t|\mathcal{X}_{t-1}, \mathcal{Z}_{t-1})p(\mathcal{X}_{t-1}|\mathcal{Z}_{t-1}) \\ &= \int_{\mathbf{x}_{t-1}} \int_{\mathcal{X}_{t-2}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathcal{X}_{t-1}|\mathcal{Z}_{t-1}) \quad \text{using lemma 2} \\ &= \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}) \end{aligned}$$

which is the required result.

C Asymptotic correctness of the CONDENSATION Algorithm

The CONDENSATION algorithm is validated here by a probabilistic argument showing that the sample-set representation of conditional density is correct, asymptotically, as the size N of the sample set at each time-step gets large. The argument is based on the one by Grenander *et al.* to justify their factored sampling algorithm for interpretation of static images. They use the standard probabilistic tool of "weak convergence" (Rao, 1973) and the "weak law of large numbers" to show that a posterior distribution inferred by factored sampling can be made arbitrarily accurate by choosing N sufficiently large. No formal indication is given as to how large N should be for a given level of accuracy, something which is determined in practice by experimentation.

In the proof that follows, the correctness proof for factored sampling of a static image is made inductive so that it can be applied to successive images in a sequence. This would be sufficient to apply several independent images to the estimation of a static underlying object. A further generalisation takes account of the predictive step (step 2 of the CONDENSATION algorithm) that deals with the dynamics of an object in motion.

C.1 Factored sampling

The asymptotic correctness of the factored sampling algorithm (section 3) is expressed in a theorem of Grenander et al (1991):

Theorem 3 (Factored sampling) *If $\alpha p_0 p_z$ is an (absolutely continuous) density function (with α a suitable normalisation constant) then for any given value* $\mathbf{x}$

$$\tilde{p}(\mathbf{x}) \to \alpha p_0(\mathbf{x}) p_z(\mathbf{x}), \text{ weakly, as } N \to \infty$$

— pointwise, weak convergence of the density function to the required posterior.

(Recall $\tilde{p}$ is the density function of the random variate $\mathbf{x}$ generated by factored sampling, as defined in section 3.) The proof of the theorem was given by Grenander *et al.*.

C.2 Dynamic extension of factored sampling

The first step in the extension for dynamic problems is to state a corollary of the theorem above that generalises it slightly to the case where the prior is not known exactly but has itself been simulated approximately.

Cor. 4 (Weak factored sampling) *The sequence $\mathbf{s}_1, \ldots, \mathbf{s}_N$ is now generated by sampling from a density p_s chosen such that*

$$p_s(\mathbf{x}) \to p_0(\mathbf{x}), \text{ weakly, as } N \to \infty,$$

where convergence is uniform with respect to $\mathbf{x}$. *Provided p_z is bounded, the random variate $\mathbf{x}'$ generated from the $\mathbf{s}_n$ as before has a density function $\tilde{p}$ for which*

$$\tilde{p}(\mathbf{x}) \to \alpha p_0(\mathbf{x}) p_z(\mathbf{x}) \text{ weakly, as } N \to \infty$$

and convergence is uniform with respect to $\mathbf{x}$.

The proof of this corollary is straightforward.

C.3 Propagation of approximated state density

First note that the samples $\mathbf{s}_t^{(n)}$ generated by the algorithm can themselves be regarded as random variables. Using the corollary it is possible to establish that asymptotically the probability density of any given $\mathbf{s}_t^{(n)}$ converges to the desired probability density $p(\mathbf{x}_t|\mathcal{Z}_{t-1})$. From now on the limit symbol $\rightarrow$ is used to denote weak, uniform convergence of density functions as $N \rightarrow \infty$. The correctness result is expressed in the theorem below. We first require a normalisation assumption for the process density, that

$$\int p(\mathbf{x}_t|\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1} \quad \text{is bounded}^4. \tag{17}$$

Theorem 5 (Weak propagation) *Each sample* $\mathbf{s}_t^{(n)}$, $n = 1, \ldots, N$ *at time* t *is drawn from a distribution with density* $\tilde{p}_t$ *such that*

$$\tilde{p}_t(\mathbf{x}_t) \rightarrow p(\mathbf{x}_t|\mathcal{Z}_{t-1}).$$

Proof

The proof is inductive. Suppose the result holds for $\tilde{p}_{t-1}$; then after step 1 of the algorithm in figure 6, by the corollary, and observing that the sampling probabilities are

$$\pi_{t-1}^{(n)} \propto\ p(\mathbf{z}_{t-1}|\mathbf{x}_{t-1} = \mathbf{s}_{t-1}^{(n)}),$$

each $\mathbf{s}'^{(n)}_{t-1}$ has a density p'_{t-1} such that

$$p'_{t-1} \rightarrow \alpha_{t-1} p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-2}) p(\mathbf{z}_{t-1}|\mathbf{x}_{t-1})$$

where α_{t-1} is a normalisation constant so that

$$p'_{t-1} \rightarrow p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1}).$$

In step 2 of the algorithm the random dynamical step is applied to $\mathbf{s}'^{(n)}_t$ to give $\mathbf{s}_t^{(n)}$ with density p'' such that

$$\begin{aligned}
p''(\mathbf{x}_t) &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1} = \mathbf{s}'^{(n)}_t)\ p(\mathbf{s}'^{(n)}_t)\, d\mathbf{s}'^{(n)}_t \\
&= \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p'(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1} \\
&\rightarrow \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})\, d\mathbf{x}_{t-1} \quad \text{(making use of (17))} \\
&= p(\mathbf{x}_t|\mathcal{Z}_{t-1})
\end{aligned}$$

and this is the required density function for $\mathbf{s}_t^{(n)}$, establishing the inductive step as required.

Finally the ground instance is straightforward. Initial samples $\mathbf{s}'^{(n)}_1$ are drawn in step 1 from the prior p_0 so that, after step 2 of the algorithm, the $\mathbf{s}_1^{(n)}$ are sampled predictions for time $t = 1$ from a density $\tilde{p}_1$ such that

$$\tilde{p}_1(\mathbf{x}_1) = p(\mathbf{x}_1) \equiv p(\mathbf{x}_1|\mathcal{Z}_0).$$

[4] This assumption is not restrictive in practice but is a little inelegant and perhaps there is a way to do without it.

($\mathcal{Z}_0$ is an empty set) so certainly

$$\tilde{p}_1(\mathbf{x}_1) \rightarrow p(\mathbf{x}_1|\mathcal{Z}_0)$$

as required.

Note that convergence has not been proved to be uniform in t. For a given fixed t, there is convergence as $N \rightarrow \infty$ but nothing is said about the limit $t \rightarrow \infty$. In practice this could mean that at later times t larger values of N may be required, though that could depend also on other factors such as the nature of the dynamical model.