International Journal of Computer Vision 39(1), 57–71, 2000

A Probabilistic Exclusion Principle for Tracking Multiple Objects

JOHN MACCORMICK
Department of Engineering Science, University of Oxford[*]*, Oxford OX1 3PJ, UK*

ANDREW BLAKE
Microsoft Research Limited, St George House, 1 Guildhall Street, Cambridge CB2 3NH, UK

Abstract. Tracking multiple targets is a challenging problem, especially when the targets are "identical", in the sense that the same model is used to describe each target. In this case, simply instantiating several independent 1-body trackers is not an adequate solution, because the independent trackers tend to coalesce onto the best-fitting target. This paper presents an observation density for tracking which solves this problem by exhibiting a *probabilistic exclusion principle*. Exclusion arises naturally from a systematic derivation of the observation density, without relying on heuristics. Another important contribution of the paper is the presentation of *partitioned sampling*, a new sampling method for multiple object tracking. Partitioned sampling avoids the high computational load associated with fully coupled trackers, while retaining the desirable properties of coupling.

Keywords: partitioned sampling, Monte Carlo, particle filter, tracking, multiple objects

1. Introduction

This paper proposes a mathematically rigorous methodology for tracking multiple objects. The fundamental problem to be addressed is demonstrated in Fig. 1. Two instantiations of the same tracking algorithm, with different initial conditions, are used to track two targets simultaneously. When one target passes close to the other, both tracking algorithms are attracted to the single target which best fits the head-and-shoulders model being used. One might think of avoiding this problem in a number of ways: interpreting the targets as "blobs" which merge and split again (Haritaoglu et al., 1998; Intille et al., 1997), enforcing a minimum separation between targets (Rasmussen and Hager, 1998), or incorporating enough 3D geometrical information to distinguish the targets (Koller et al., 1994). However, each of these solutions can be unattractive.

A blob interpretation does not maintain the identity of the targets, and is difficult to implement for moving backgrounds and for targets which are not easily segmented. A minimum separation relies on heuristics and fails if the targets overlap. Incorporating 3D information is impossible without detailed scene modelling.

So it seems we must instead address the fundamental problem: that the observation model used to interpret image measurements permits two targets to occupy the same point in configuration space too easily. More specifically, a single piece of image data (such as an edgel, or a colour blob), must not simultaneously reinforce mutually exclusive hypotheses. What is needed is a "probabilistic exclusion principle", and an observation model exhibiting this behaviour is described in this paper. The formal model will initially be derived for "wire frame" targets—objects which have detectable boundaries but which do not occlude each other. We then describe how occlusion reasoning about solid objects can be incorporated naturally into the same framework. The most interesting feature of this approach is that it works even when the targets are *indistinguishable given the available information*. This is of both theoretical and practical interest.

Many visual tracking systems for multiple objects have been developed. One standard technique

[*]http://www.robots.ox.ac.uk/∼vdg

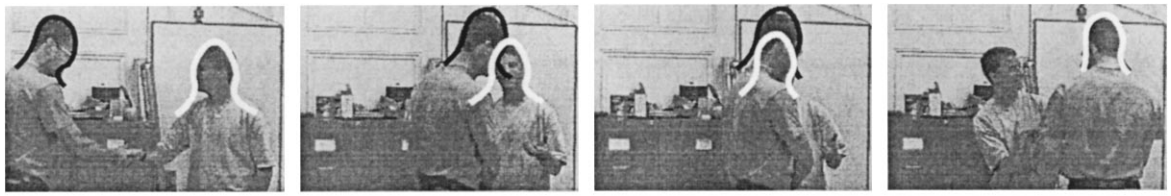


Figure 1. With an observation model designed for one target, two trackers initialised in distinct configurations eventually lock on to the one target which bests fits the model. The objective is to derive an observation model which does not permit the presence of two targets to be inferred from measurements of only one.

is the probabilistic data association filter (PDAF) (Bar-Shalom and Fortmann, 1988), and other successful examples include (Haritaoglu et al., 1998; Intille et al., 1997; Paragios and Deriche, 1998; Rasmussen and Hager, 1998). These generally employ a combination of blob identification and background subtraction; both techniques are complementary to the method proposed here. In particular, our exclusion principle does not allow two targets to merge when their configurations become similar; instead, the model continues to interpret the data in terms of two targets. As will be seen, it is a natural consequence of the methodology that the probability distribution for an obscured target diffuses until it is reinforced by further data. Furthermore, the method works for unknown and constantly changing backgrounds. Rasmussen and Hager (1998) proposed a promising method for combining colour blob and edge information, and incorporated an exclusion principle by using a joint PDAF. However, their algorithm for fusing edgel information enforced an arbitrary minimum separation between targets. Gordon (1997) employs a similar multi-target tracking methodology to this paper but with a rather different observation model and no explicit exclusion principle.

One of the difficulties with tracking multiple objects is the high dimensionality of the joint configuration space. Section 5 introduces a method known as *partitioned sampling* which diminishes the computational burden associated with the increased dimensionality of multi-target spaces.

2. The Observation Model

The target objects in this paper are described by their outlines, which are modelled as B-splines. We will call any such outline a *contour*. The space of contours which can correspond to a target or set of targets is called the *shape space* (Blake and Isard, 1998), and is parameterised as a low-dimensional vector space $\mathcal{X}$. The space $\mathcal{X}$ generally has 5–50 dimensions. This framework is based on standard concepts from the theory of snakes and deformable templates (e.g. Kass et al., 1987; Szeliski and Terzopoulos, 1991) and is summarised concisely in Blake and Isard (1998).

A configuration $\mathbf{x} \in \mathcal{X}$ is measured by the method of Fig. 2, obtaining a list of image coordinates $\mathbf{Z} = (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)})$. A component of $\mathbf{Z}$ is itself a vector $\mathbf{z}^{(m)}$ consisting of the measurements made along fixed *measurement lines* (see the figure) of the configuration $\mathbf{x}$. An advantage of this measurement line approach is that we have reduced the problem of analysing a 2D image to that of analysing several 1D measurement lines. The statistical processes generating features on different measurement lines are treated as independent (the merits of this approximation are discussed in

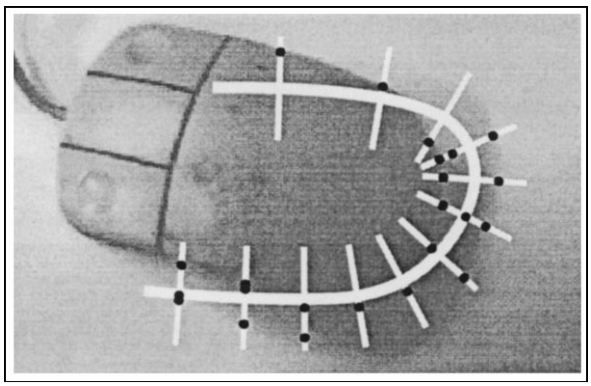


Figure 2. Measurement methodology. The thick white line is $\mathbf{x}$—a mouse-shaped contour in some hypothesised configuration. The thin lines are *measurement lines*, along which a one-dimensional feature detector is applied. Black dots show the output of the feature detector, which in this case responds to rapid changes in intensity—one-dimensional edges. Note that many spurious edges are generated by shadows, or more generally by clutter in the image.

Section 2.2), so we need only specify this process on 1D subsets of the image.

So consider just one fixed measurement line, of length L, positioned in an image known to contain two target objects. A one-dimensional edge detector is applied to this line, and some features are detected at image coordinates $\mathbf{z} = (z_1, z_2, \ldots, z_n)$. Some of the z_i might correspond to the target objects' boundaries, while the others are due to clutter in the image. So we must develop a *generative model* for both the target and clutter features—this is analogous to the models adopted in some pattern recognition tasks, such as the generation of printed matter as "character + ink spatter" (Hinton et al., 1992). For a given target configuration $\mathbf{x}$, there are three possibilities to consider: the measurement line may intersect $c = 0$, 1 or 2 of the targets. The probability densities for each case are denoted $p_c(n; \mathbf{z})$. To calculate the p_c, several concrete assumptions about the generative model for $\mathbf{z}$ are adopted:

- $c = 0$ ("random background clutter"): The probability of obtaining n features is $b(n)$, learnt from randomly placed measurement lines in typical images. The positions of the n features $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ are drawn from the uniform distribution on the measurement line. These assumptions are discussed in Section 2.1.
- $c = 1$ ("single target"): One of the n features corresponds to the target boundary, whose hypothesised position on the measurement line is denoted ν. If the boundary feature is z_i, then z_i is assumed to be drawn from a fixed probability distribution $\mathcal{G}(z_i \mid \nu)$, termed the "boundary feature distribution". In this paper $\mathcal{G}(z_i \mid \nu)$ is a Gaussian centred on ν with variance σ^2 (we take $\sigma = 7$ pixels in the examples later; see Table 1 for the justification of this value). The remaining $n - 1$ features are assumed to be drawn from the random background clutter distribution described above.
- $c = 2$ ("two targets"): Two of the n features, say z_i, z_j, correspond to target boundaries at hypothesised positions ν_1, ν_2. They are drawn from $\mathcal{G}(z_i \mid \nu_1), \mathcal{G}(z_j \mid \nu_2)$ respectively with, importantly, $i \neq j$. In other words, any edge feature can correspond to at most one target boundary. It is this assumption which leads to the enforcement of a probabilistic exclusion principle described later on. (The same assumption is made in (Rasmussen and Hager, 1998) to enforce exclusion in the context of a joint PDAF). Again the remaining $n - 2$ features are drawn from the background distribution.

The model can be generalised to higher values of c, but for clarity only the cases $c = 0, 1, 2$ are considered here. The assumption for $c = 2$ that any one edge feature corresponds to at most one target is crucial, and requires further explanation. While it is true that wherever two targets cross, there *is* a single edge corresponding to two targets, such points form a very sparse set in the image. The possibility that such a point lies on one of the measurement lines is therefore disregarded. For an example, look ahead to Fig. 8.

The mathematical consequences of these assumptions are collected in the next proposition, which is proved in the appendix. Note that $p(n; \mathbf{z})$ is a probability distribution over both n and $\mathbf{z}$—this notation is explained in the appendix. Also note the density p follows the generative model in assuming that the measurements $(z_1, \ldots, z_n)$ might come in any order with equal likelihood; if it is assumed instead that the measurements are made in a prescribed order (e.g. $z_1 \leq z_2, \ldots, \leq z_n$) then each density should be multiplied by $n!$.

Table 1. Parameter values and other choices used for experiments. Suitable non-detection probabilities were determined by trial and error on simple examples. The discrete transition probability corresponds to a time constant of 2.0 seconds for a given discrete state. The standard deviation of the boundary feature distribution is estimated from the mean-square mismatch of templates fitted to the targets. The measurement lines extend approximately 3 of these standard deviations in each direction.

Non-detection probabilities, $c = 1$	(q_{01}, q_{11})	(0.1, 0.9)
Non-detection probabilities, $c = 2$	(q_{02}, q_{12}, q_{22})	(0.05, 0.2, 0.75)
Clutter feature probabilities	$b(n)$	MLE from first frame of sequence
Discrete transition probability	δ	0.01
Boundary feature distribution	$\mathcal{G}(z \mid \nu)$	Gaussian with std dev of 7 pixels
Length of measurement lines	L	40 pixels

Proposition 1. *The probability density functions resulting from the assumptions above are*

$$
\begin{aligned}
p_0(n;\mathbf{z}) &= b(n)/L^n \\
p_1(n;\mathbf{z}\mid\nu) &= b(n-1)\sum_{k=1}^{n}\mathcal{G}(z_k\mid\nu)/nL^{n-1} \qquad (1)\\
p_2(n;\mathbf{z}\mid\nu_1,\nu_2) &= b(n-2)\sum_{i\neq j}\frac{\mathcal{G}(z_i\mid\nu_1)\mathcal{G}(z_j\mid\nu_2)}{L^{n-2}n(n-1)}
\end{aligned}
$$

As described so far, the generative model assumes that if a target boundary is present, then the edge detector will detect it. This is unrealistic: occasionally the target object's boundary is not detected, because the background and target happen to have similar grey-scale values. Hence a final step is added to the generative model. It is assumed that when $c = 1$ there is a small fixed probability q_{01} of the edge detector failing to detect the target boundary, and $q_{11} = 1 - q_{01}$ that it will succeed. This is precisely analogous to the non-detection probabilities used in PDAFs (Bar-Shalom and Fortmann, 1988). Similarly, when $c = 2$, there are fixed probabilities q_{02}, q_{12}, q_{22} that $0, 1, 2$ target boundaries are detected successfully. Thus we can define pdfs $\tilde{p}$ for the final generative model as follows, for the cases $c = 0, 1, 2$:

$$
\begin{aligned}
\tilde{p}_0(\cdot) &= p_0(\cdot)\\
\tilde{p}_1(\cdot\mid\nu) &= q_{01}p_0(\cdot) + q_{11}p_1(\cdot\mid\nu) \qquad (2)\\
\tilde{p}_2(\cdot\mid\nu_1,\nu_2) &= q_{02}p_0(\cdot) + q_{12}(p_1(\cdot\mid\nu_1)\\
&\quad + p_1(\cdot\mid\nu_2))/2 + q_{22}p_2(\cdot\mid\nu_1,\nu_2)
\end{aligned}
$$

Typical graphs of the last two functions are shown in Figs. 3 and 4.

The above discussion was framed in terms of a single measurement line, but for any given hypothesised

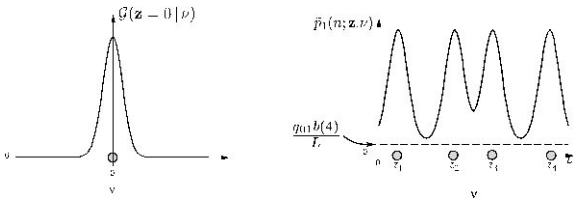


Figure 3. 1-target likelihood function for a single measurement line. Left: The boundary feature distribution $\mathcal{G}(z = 0\mid\nu)$. Right: The 1-target likelihood function $\tilde{p}_1(n;\mathbf{z}\mid\nu)$ graphed with respect to ν. The likelihood is a linear combination of shifted copies of $\mathcal{G}(z\mid\cdot)$ and of the constant p_0. It peaks near the 4 measurements z_i (shown as shaded circles).

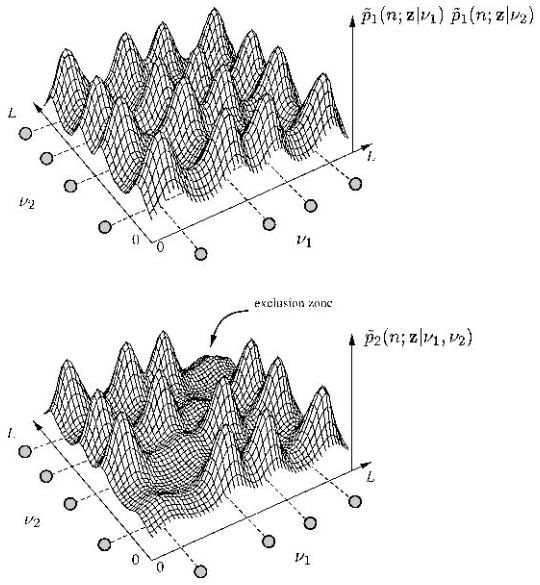


Figure 4. 2-target likelihood functions for a single measurement line. Top: A naïve 2-target likelihood $\tilde{p}_1(n;\mathbf{z}\mid\nu_1)\tilde{p}_1(n;\mathbf{z}\mid\nu_2)$ formed by taking the product of two 1-target densities (Fig. 3). The likelihood peaks near pairs of measurements z_i, z_j (shaded circles and dotted lines). Bottom: The 2-target likelihood $\tilde{p}_2(n;\mathbf{z}\mid\nu_1,\nu_2)$ derived from the generative model. Again, the likelihood peaks near pairs of measurements z_i, z_j (shaded circles and dotted lines), but now a probabilistic exclusion principle operates: because the sum in the definition of p_2 excludes $i = j$, the probability peaks are much smaller on the line $\nu_1 = \nu_2$.

configuration $\mathbf{x}$, the measurements $\mathbf{Z}$ will arise from say M distinct measurement lines. Let $c(i)$ be the number of target boundaries intersecting the ith measurement line for a given configuration $\mathbf{x}$, and let $\boldsymbol{\nu}^{(i)}$ be the coordinates of these intersections. By making the assumption that outputs on distinct measurement lines are statistically independent (Section 2.2), we define the *exclusive likelihood function* as

$$
\mathcal{P}(\mathbf{Z}\mid\mathbf{x}) = \prod_{i=1}^{M}\tilde{p}_{c(i)}\left(\mathbf{z}^{(i)}\;\middle|\;\boldsymbol{\nu}^{(i)}\right). \qquad (3)
$$

We call $c(i)$ the *intersection number* of the ith measurement line.

2.1. Discussion of the Background Model

Recall that the numbers $b(n), n \in \mathbb{N}$ specify the probability of obtaining n features on a measurement line positioned randomly on the background, and that these probabilities are learnt from typical training images. Of course this innocuous statement conceals

a perennial problem in computer vision: how does one characterise a "typical" image, and even worse, how does one specify a prior for such images? Even when an image is reduced to the simple level of one-dimensional features, there is no straightforward answer to this question. However, it turns out the tracking system described later is extremely robust to the choices of $b(n)$. Indeed, we routinely set $b(0) = b(1) = \cdots = b(n_{\max}) = 1/(1 + n_{\max})$ for some $n_{\max}$, with $b(n) = 0$ when $n > n_{\max}$. For measurement lines of 40 pixels, and an edge convolution operator with weights $(-0.375, -0.625, 0, 0.625, 0.375)$, one can take $n_{\max} \approx 10$ and obtain results indistinguishable from when the $b(n)$ are learnt from the entire sequence to be tracked. Another simple approach which gives equally good results in all our experiments is to learn the $b(n)$ from the first image of the sequence.

An alternative approach to modelling the occurrence of background features is the careful use of a Kalman filter framework to disregard spurious features (e.g. Peterfreund, 1998), but in order for this to work in cluttered backgrounds, one needs much more accurate dynamical models than those available in the type of problems considered here. Other researchers explicitly adopt a uniform distribution on the $b(n)$ (e.g. Lowe, 1992), as suggested above.

Our second assumption about random background clutter features is that their *positions* are drawn from a uniform distribution. What is the corresponding assumption about 2D image features that would make this true? It would certainly hold provided the positions of all edgels of a given orientation were also distributed uniformly. We find this is sufficiently true over the small regions (scale around 40 pixels) occupied by the measurement lines, but it is clear that this approximation is unsatisfactory for larger regions. Further work is needed here: perhaps the recent ideas on filters and scale-invariance (Mumford and Gidas, 1999; Zhu et al., 1998) can be applied to obtain a more coherent theory.

2.2. Independence of Measurement Lines

The exclusive likelihood function (3) was derived assuming that feature occurrences on distinct measurement lines are statistically independent. Of course this is an approximation, since there are generally continuous edges in the background as well as on the boundary of the target object. There have been some attempts to allow explicitly for this type of dependence—for example, the Markov discriminant of (MacCormick and Blake, 1998b), or MRFs in general (Chellappa and Jain, 1993; Kent et al., 1996; Winkler, 1995). However, these are too computationally expensive for tracking tasks, so instead we adopt the assumption of independence between measurement lines. One might hope this approximation is acceptable if the measurement lines used for inferences are sufficiently far apart. Figure 5 investigates the meaning of "sufficiently far" in this context. This figure shows the autocorrelation of a random process $x(d)$ defined as follows (see also Fig. 6). First, randomly position a measurement line, uniformly in position and orientation, on a typical background image (in this case the first frame of the leaf sequence—see Fig. 16). Apply a feature detector, select the closest feature to the centre of the measurement line, and define $x(0)$ to be the offset of this feature. The value of $x(d)$ is defined by first displacing the original measurement line a distance of d pixels in the direction of its normal, then applying the feature detector and setting $x(d)$ to be the offset of the most central feature. Of course Fig. 5 does not establish the joint independence of the feature occurrences on all measurement lines which are sufficiently far apart. The autocorrelation function involves only 2nd-order moments, whereas independence requires that moments of all orders vanish. In addition, even if pairwise independence of the measurement lines was established, it would still not follow that they were *jointly* independent. Nevertheless, Fig. 5 does imply that the outputs of measurement lines separated by less than 10–20 pixels are rather strongly correlated, but that

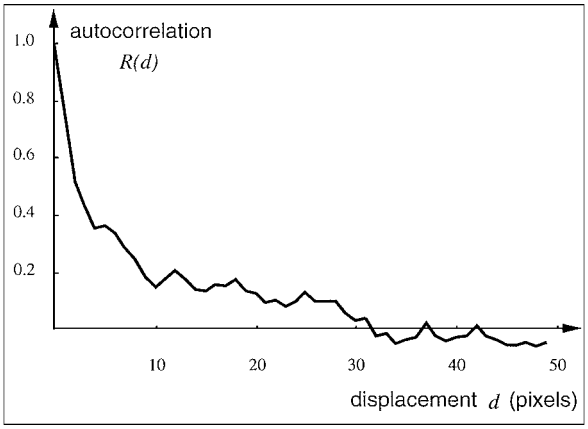


Figure 5. Feature autocorrelation is low for displacements of more than 30 pixels. This is our justification for treating distinct measurement lines as statistically independent. The random process $x(d)$ whose autocorrelation is graphed here is described in the text and Fig. 6, and the autocorrelation function is defined as usual by $R(d) = (E[x(d)x(0)] - E[x(0)]^2)/(E[x(0)^2] - E[x(0)]^2)$.

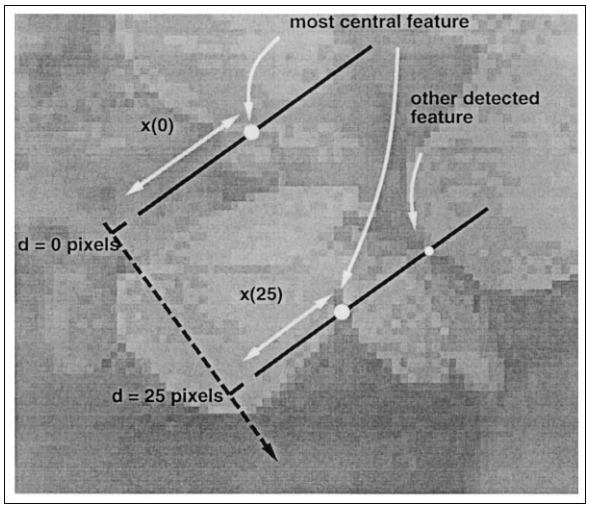


Figure 6. Investigating feature correlation. The top solid black line is a measurement line positioned randomly on a typical background image. The value of the random process $x(d)$ is the offset of the most central detected feature after the initial measurement line has been displaced by d pixels in the direction of its normal.

this correlation is much weaker for separations of 30 or more pixels. The likelihoods in this paper employed a separation between measurement lines of approximately 30 pixels.

2.3. A Separate Interior Model

Features detected in the *interior* of an opaque target object are not generated by random background clutter. This contradicts the simple generative model above, and it was shown in (MacCormick and Blake, 1998a) that a more complex model explicitly accounting for the interior of the target can improve the resulting inferences. However, even simple interior models lead to intractable pdfs involving numerical integration. Hence, for simplicity, the results in this paper assume that features detected in the interior of an opaque target are drawn from the same distribution as the background.

2.4. Selection of Measurement Lines

Often we need to perform Bayesian inference on the image, based on the measurements $\mathbf{Z}$ of several hypothesised configurations $\mathbf{x}_1, \ldots, \mathbf{x}_n$. For Bayes' Theorem to be valid, the set of measurement lines must be fixed in advance. However, it is sometimes convenient to allow the precise choice of measurement lines to depend on the configuration $\mathbf{x}$, as in Fig. 2. When the $\mathbf{x}_i$ are tightly clustered, this is a minor approximation which was adopted in this paper for ease of implementation. Our experiments on other tracking tasks with measurement lines fixed in advance produce indistinguishable results.

3. Tracking Multiple Wire Frames

Tracking is performed in this paper by the Condensation algorithm (Isard and Blake, 1998a), which is capable of dealing with complex likelihood functions such as (3). Condensation is a filtering algorithm which performs a Bayesian estimation of the posterior for the state of a system at each time step. Because of the complex likelihood function, there is no closed form of the Bayesian update at each time step. Condensation circumvents this problem by *approximating* the distribution to be estimated using "weighted particle sets". To be specific, a Condensation tracker represents the state of a system at time t by a weighted set of *samples* or *particles* $s_1^t, \ldots, s_N^t$ whose weights are $\pi_1^t, \ldots, \pi_N^t$. This set is intended to be an approximate representation of some probability distribution function $p(\mathbf{x})$, in the sense that selecting one of the s_i with probability proportional to π_i is approximately the same as making a random draw from $p(\mathbf{x})$. This concept is formalised in Section 5.1.

Given a particle set $(s_i^t, \pi_i^t)_{i=1}^n$ which represents the posterior at time t, the Condensation algorithm generates a particle set representing the posterior at time $t+1$ in three steps: (i) resampling: sample N times with replacement from the set of particles s_i^t, according to the weights π_i^t—this produces a set $s_1^{t'}, \ldots, s_N^{t'}$; (ii) dynamical propagation: sample from $p(\mathbf{x}^{t+1} \mid \mathbf{x}^t = s_i^{t'})$ to choose each s_i^{t+1}; and (iii) measurement: examine the image to obtain the features $\mathbf{Z}^{t+1}$, then assign each of the new particles a weight $\pi_i^{t+1} \propto p(\mathbf{Z}^{t+1} \mid \mathbf{x}^{t+1} = s_i^{t+1})$. The three transformations of the particle set in any time step can be conveniently summarised diagrammatically:

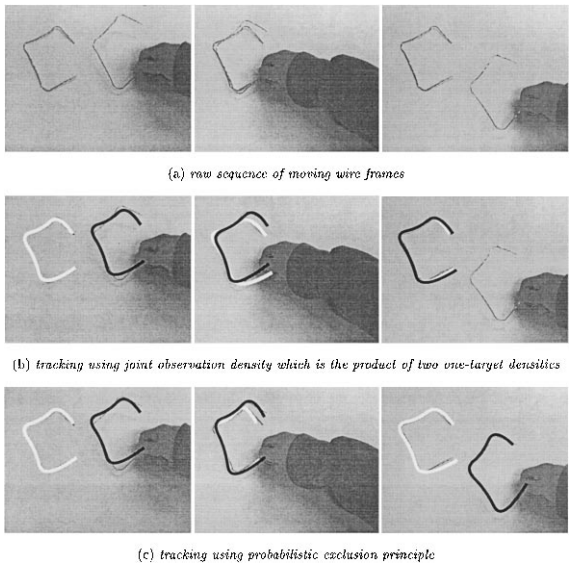
$$\boxed{\text{prior}} \longrightarrow \boxed{\sim} \longrightarrow \boxed{* p(\mathbf{x}'|\mathbf{x})} \longrightarrow \boxed{\times p(\mathbf{Z}|\mathbf{x}')} \longrightarrow \boxed{\text{posterior}} \quad (4)$$

$$(a) \qquad (b) \qquad (c)$$

The $\sim$ symbol represents resampling as described above, the $*$ is application of a stochastic dynamical step, and the $\times$ represents multiplication (i.e. reweighting) by the measurement density. The labels (a)–(c) refer to an example given later (Fig. 12), and can be disregarded for the moment. Of course, to demonstrate the exclusion principle we use the exclusive likelihood function $\mathcal{P}(\mathbf{Z} \mid \mathbf{x})$ as the measurement density. Note that $\mathcal{P}$ as defined in (3) is not valid for opaque objects, since the model expects to observe all boundaries, even those which are occluded. However, it *is* valid for wire frame objects, so an experiment on wire frames was performed. As a control for the experiment, we need a likelihood $\mathcal{P}'$, similar to $\mathcal{P}$, but which does not incorporate an exclusion principle. Naming the two targets A and B, and writing $c_A(i)$ for the number of intersections of A with line i, let $\nu_A^{(i)}$ be the coordinates of these intersections and define the *1-body likelihood*

$$\mathcal{P}_A(\mathbf{Z} \mid \mathbf{x}) = \prod_{i=1}^{M} \tilde{p}_{c_A(i)}\left(\mathbf{z}^{(i)} \,\middle|\, \boldsymbol{\nu}_A^{(i)}\right), \tag{5}$$

and similarly for $\mathcal{P}_B$. We take $\mathcal{P}' = \mathcal{P}_A \mathcal{P}_B$, so the posteriors for A and B given $\mathbf{Z}$ are treated as independent. A typical graph of $\mathcal{P}'$ for just one measurement line is shown at the top of Fig. 4—note that in contrast to the graph of $\mathcal{P}$ below it, $\mathcal{P}'$ has four additional peaks down the line $\nu_1 = \nu_2$. Figure 7 shows the results of the wire frame experiment: as expected, $\mathcal{P}$ successfully maintains exclusion between the targets whereas $\mathcal{P}'$ does not.



(a) *raw sequence of moving wire frames*

(b) *tracking using joint observation density which is the product of two one-target densities*

(c) *tracking using probabilistic exclusion principle*

Figure 7. The exclusion principle operating on a wire-frame example. (a) Three stills from a sequence of two pieces of wire with similar shapes. Note that for several frames in the middle of the sequence, the two wires have very similar configurations. (b) Results using the likelihood $\mathcal{P}'$, which does not incorporate an exclusion principle. When the configurations become similar, both targets settle on the best-fitting wire. (c) Successful tracking using the exclusion principle likelihood $\mathcal{P}$.

4. Tracking Multiple Opaque Objects

The wire-frame model can be adapted for use with solid objects. The method uses the mixed state Condensation tracker of (Isard and Blake, 1998c), combined with a "2.1D" (Mumford and Nitzberg, 1990) or "layered" (Irani and Anandan, 1998) model of the targets. The basic idea of a mixed state Condensation tracker is that each particle carries a discrete label in addition to the continuous parameters describing its configuration. Let y be a discrete variable labelling the current model, and let $\mathbf{x}$ be a shape space vector of continuous parameters specifying the configuration of the targets. The extended state is defined to be

$$\mathbf{X} = (\mathbf{x}, y), \mathbf{x} \in \mathbb{R}^d, y \in \{1, \ldots, N_s\}. \tag{6}$$

In the two-object case, $\mathbf{x} = (\mathbf{x}^A, \mathbf{x}^B)$ and y can take one of two values: $y = 1$ if A is nearer the camera than B, and $y = 2$ if B is nearer than A. This is what we mean by a 2.1D model: the only 3D geometric aspect to be inferred is whether target A can occlude target B or vice versa.

We assume the dynamics of the continuous parameters do not depend on the discrete state, so that $p(\mathbf{x}_t \mid \mathbf{X}_{t-1}) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. Then the process density can be decomposed as follows:

$$p(\mathbf{X}_t \mid \mathbf{X}_{t-1}) = P(y_t \mid \mathbf{x}_t, \mathbf{X}_{t-1})\, p(\mathbf{x}_t \mid \mathbf{x}_{t-1}),$$

and if $y_{t-1} = j$ and $y_t = i$ this can be written more explicitly as

$$p(\mathbf{X}_t \mid \mathbf{X}_{t-1}) = T_{ij}(\mathbf{x}_t, \mathbf{x}_{t-1}) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}),$$

where T_{ij} is a transition matrix and p is a density specifying the continuous dynamics for a particle. Here it is appropriate for $T_{ij}(\mathbf{x}_t, \mathbf{x}_{t-1})$ to be independent of $\mathbf{x}_{t-1}$. If $\mathbf{x}_t^A$ and $\mathbf{x}_t^B$ overlap then the occlusion relationship cannot change in the the current time-step and so we take $T_{ij}(\mathbf{x}_t)$ to be the identity matrix. If $\mathbf{x}_t^A$ and $\mathbf{x}_t^B$ do not overlap then we assume there is a small, fixed probability that y will change, represented by taking $T_{ij}(\mathbf{x}_t) = \begin{pmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{pmatrix}$ with $0 < \delta \ll 1$.

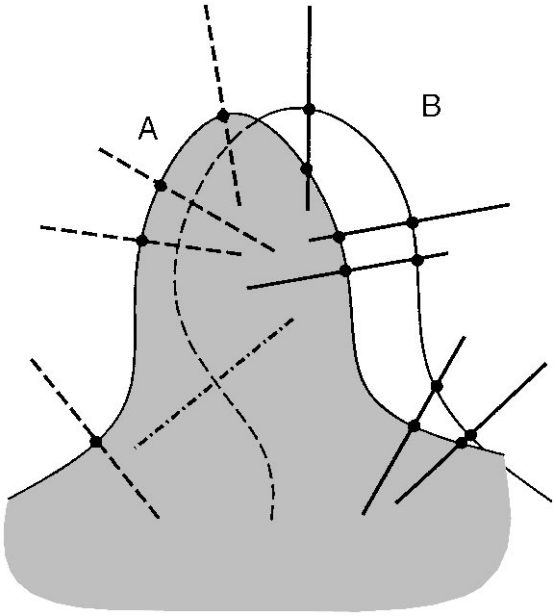


Figure 8. Intersection numbers calculated from 2.1D geometry. In this diagram, $y = 1$, meaning the shaded area is occluded by target A. Visible intersections of measurement lines and target boundaries are shown as solid circles. The solid lines have intersection number $c = 2$, dashed have $c = 1$ and dot-dashed $c = 0$. These are c-values used in (7).

The mixed state Condensation tracker presented here incorporates a significant difference to that of (Isard and Blake, 1998c)—the observation density $p(\mathbf{Z}_t \mid \mathbf{X}_t)$ depends not only on $\mathbf{x}_t$ but also on the discrete state y_t. The multi-target exclusive likelihood function (3) is used, but now the intersection counts $c(i)$ are calculated using the discrete variable y and the 2.1D geometry to determine if a given boundary feature should be visible or not, as in Fig. 8. To emphasise this we can write $c(i, y)$ for the number of *visible* target boundaries intersecting the ith measurement line of a configuration $(\mathbf{x}, y)$; the coordinates of the visible boundaries on the ith line are written $\boldsymbol{\nu}^{(i,y)}$. Then the likelihood in the occluded case is

$$\mathcal{P}_{\text{occl}}(\mathbf{Z} \mid \mathbf{x}) = \prod_{i=1}^{M} \tilde{p}_{c(i,y)}\left(\mathbf{z}^{(i)} \mid \boldsymbol{\nu}^{(i,y)}\right). \qquad (7)$$

To understand this, compare with Eq. (3). The functions $\tilde{p}_c, c = 0, 1, 2$ are still as defined in (2). The only change is that the intersection numbers c and target boundary positions ν now depend on the discrete state y which specifies which target is in front of the other.

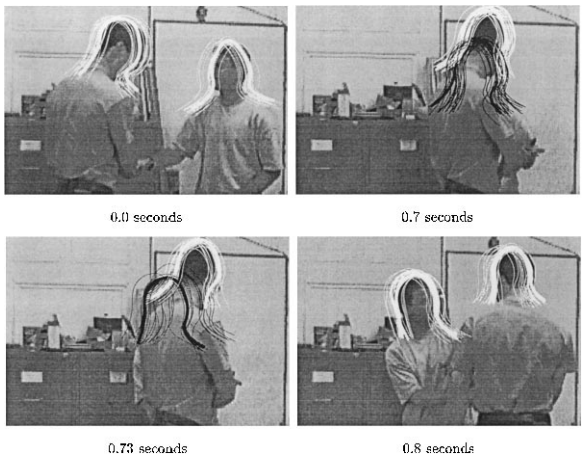


Figure 9. Successful tracking with a density incorporating occlusion reasoning (c.f. Fig. 1). 20 of the 2000 particles are shown in each frame, with widths proportional to their probabilities. Recall that a single "particle" in this context is a *joint* hypothesis for the configuration of both targets. Initially, each particle consists of two white contours: one initialised on each of the two targets. A contour is drawn in black if its value of y, as defined in (6), implies that it is partially occluded.

The derivation of (7) is otherwise identical to (3). A detailed example is given in Fig. 8.

The likelihood $\mathcal{P}_{\text{occl}}$ performs well in experiments. Figure 9 shows a typical sequence involving occlusion. The configuration space has 16 dimensions: 8 key-frames from principal components analysis of templates (Baumberg and Hogg, 1994; Cootes and Taylor, 1992), for each of 2 targets. Tracking is performed with $N = 2000$ particles, and predictive dynamics in the form of Brownian motion with an amplitude matched to the speed of a walking person. Note how the occluded contours diffuse at 0.7 seconds. Because of the exclusion principle they coalesce again only when some evidence from the correct target is observed. The undesirable tracking behaviour of Fig. 1 has been corrected.

As a canonical tracking challenge, the same multiple target methodology was applied to the "leaf sequence" used in (Isard and Blake, 1998a). Two leaves were tracked, using an affine shape space and $N = 4000$ samples with learnt dynamics. (The need for 4000 samples is reduced to 750 by the partitioned sampling method described in the next section.) Tracking is successful despite occlusions; some stills are shown in Fig. 10.

Table 1 gives details of the parameter values used for all the experiments.

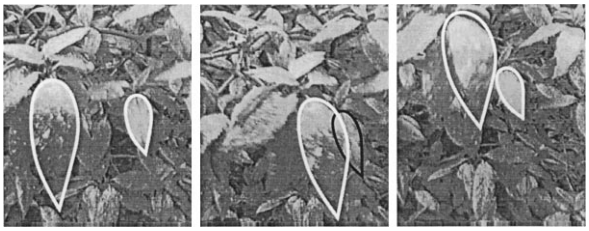


Figure 10. Tracking multiple leaves, in moving clutter and with occlusions. Three stills from a tracked sequence are shown. The black contour shows a correctly inferred occlusion.

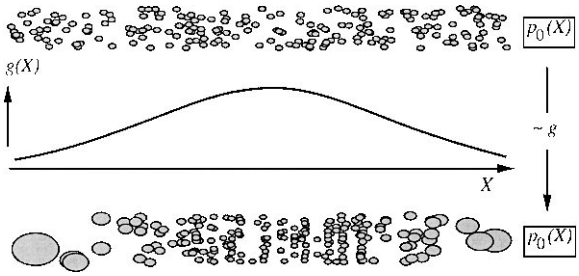


Figure 11. Weighted resampling. A uniform prior $p_0(X)$, represented as a particle set (top), is resampled via an importance resampling function g to give a new, re-weighted particle set representation of p_0. Note that these are one-dimensional distributions; the particles are spread in the y-direction only so they can be seen more easily.

5. Partitioned Sampling for Condensation

A potential limitation of the Condensation algorithm is that if the state space has many dimensions, then the number of particles required to model a distribution can be very large indeed. This is of particular concern when tracking multiple objects, since the number of dimensions in the state space is proportional to the number of objects. Fortunately, "partitioned sampling" significantly mitigates this curse of dimensionality. It is the statistical analogue of a hierarchical search: the intuition is that it should be more efficient to search *first* for whichever target is unoccluded, and only then to search for another target which may lie behind.

5.1. Weighted Resampling

The partitioned sampling algorithm requires an additional operation on particle sets, termed weighted resampling. This operation *does not alter the distribution represented by the particle set*. However, it can be used to reposition the locations of the particles so that the representation is more efficient for future operations.

Weighted resampling is usually carried out with respect to a strictly positive *importance function* $g(\mathbf{x})$. Given a particle set $s_1, \ldots, s_n$ with weights $\pi_1, \ldots, \pi_n$, the basic idea is to produce a new particle set by resampling, with replacement, from the s_i, using probabilities proportional to $g(s_i)$—this has the effect of selecting many particles in regions where g is peaked. The weights of the resampled particles are calculated in such a way that the overall distribution represented by the new particle set is the same as the old one. Intuitively, $g(\mathbf{x})$ is a function with high values in regions where we would like to have many particles. The objective of the weighted resampling is to populate such regions so that subsequent operations on the particle set will produce accurate representations of the desired probability distributions. Figure 11 shows a simple one-dimensional example of weighted resampling with respect to an importance function. A more formal discussion follows.

Definition (Weighted resampling). Let $s_1, \ldots, s_n$ be a particle set with weights $\pi_1, \ldots, \pi_n$, and let $\rho_1, \ldots, \rho_n$ be any list of strictly positive weights with $\sum \rho_i = 1$. The operation of *weighted resampling* with respect to the ρ_i produces a new particle set $s'_1, \ldots, s'_n$ with weights $\pi'_1, \ldots, \pi'_n$ by the following algorithm:

1. For $i = 1, \ldots, n$
   (a) Randomly select an index $k \in \{1, \ldots, n\}$ with probability ρ_k.
   (b) Set $s'_i = s_k$.
   (c) Set $\pi'_i = \pi_k / \rho_k$.
2. Normalise the π'_i so that $\sum \pi'_i = 1$.

Often, the ρ_i are determined from a strictly positive function $g(\mathbf{x})$, in the sense that $\rho_i \propto g(s_i)$. In this case, $g(\mathbf{x})$ is called the *importance function* and we refer to weighted resampling with respect to $g(\mathbf{x})$.

Before stating the key property of importance resampling, we must define precisely what it means for a particle set to represent a distribution.

Definition (Representation of a probability distribution by a particle set). Suppose we have a (possibly stochastic) algorithm which takes a positive integer n as input, and outputs a particle set $s_1, \ldots, s_n$ with

weights $\pi_1, \ldots, \pi_n$. This particle set can be regarded as a probability distribution $p_n(\mathbf{x}) = \sum_{i=1}^{n} \pi_i \delta(\mathbf{x} - s_i)$—a weighted sum of Dirac δ-functions centred on the s_i. The particle set is said to *represent* a probability distribution $p(\mathbf{x})$ if $p_n \to p$, weakly, as $n \to \infty$.

Remark (i). One Let $\mathcal{P}(\mathcal{X})$ be the space of all probability distributions on the configuration space $\mathcal{X}$, and let $\mathcal{P}(\mathcal{P}(\mathcal{X}))$ be the space of all probability distributions on $\mathcal{P}(\mathcal{X})$. Although we are used to considering weak convergence in the space $\mathcal{P}(\mathcal{X})$, the convergence referred to above is in the weak topology on $\mathcal{P}(\mathcal{P}(\mathcal{X}))$. Nevertheless, the definition of weak convergence remains the same (Billingsley, 1995). Specifically, we require that for all continuous, bounded, real-valued functions f on $\mathcal{P}(\mathcal{X})$, the expectation of $f(p_n)$ tends to $f(p)$ as $n \to \infty$. The expectation is over all possible random choices of the s_i and π_i. Interested readers are referred to (MacCormick, 2000; Del Moral, 1998).

Remark (ii). Strictly speaking, it is the *algorithm* for producing a particle set of arbitrary size which represents a given distribution. Nevertheless, it is convenient to speak of the set itself as representing a distribution when no confusion can arise.

Now it is possible to state accurately the fact that weighted resampling does not affect the distribution represented.

Proposition 2. *Suppose* $(s_i, \pi_i)_{i=1}^n$ *is a particle set representing a probability distribution* $p(\mathbf{x})$, *and* $(s'_i, \pi'_i)_{i=1}^n$ *is the result of weighted resampling with respect to an importance function* $g(\mathbf{x})$. *Suppose further that*

- *the support of* p *is a closed and bounded subset of* $\mathbb{R}^d$
- *the* π_i *in the particle set are proportional to some continuous function* f, *i.e.*

$$\pi_i = \frac{f(\mathbf{x}_i)}{\sum_{j=1}^{n} f(\mathbf{x}_j)}$$

- g *is continuous and strivtly positive on the support of* p

Then $(s'_i, \pi'_i)_{i=1}^n$ *represents* $p(\mathbf{x})$.

A sketch of the proof is given in the appendix.

Note that weighted resampling has a similar objective and effect to the "importance resampling" introduced in (Isard and Blake, 1998b), but that the algorithms for the two types of resampling are completely different. Importance resampling draws particles randomly from the importance distribution, then attaches weights to these particles by calculating transition probabilities from each of the old particles to each of the new ones. A crucial advantage of weighted resampling is that its number of operations is $O(n)$, whereas the calculation of transition probabilities in importance resampling is $O(n^2)$. Weighted resampling is a generalisation of both tempered weights (Carpenter et al., 1999) and the auxiliary particle filter (Pitt and Shepherd, 1997).

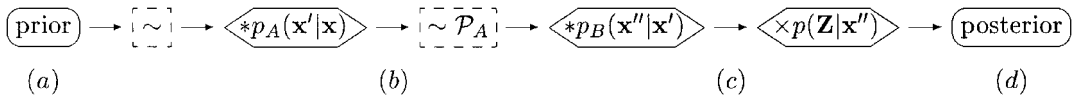
5.2. *Basic Partitioned Sampling*

Let us return to the problem of tracking two targets, A and B. If each target deforms and moves in a space of M dimensions, there are $2M$ dimensions to be inferred at each time step. By employing partitioned sampling, this problem will be reduced to the more feasible task of performing 2 inferences of M dimensions each. To be more concrete, suppose it is known that target A partially occludes target B. Then we can localise the two targets efficiently by first inferring the configuration of target A, and then using this knowledge to localise B. To infer the configuration of A, we will use the 1-body likelihood $\mathcal{P}_A$ defined by (5).

The basic algorithm is as follows. Suppose we can decompose the joint dynamics as

$$p(\mathbf{x}'' \mid \mathbf{x}) = \int_{\mathbf{x}'} p_B(\mathbf{x}'' \mid \mathbf{x}')\, p_A(\mathbf{x}' \mid \mathbf{x})\, d\mathbf{x}'$$

where p_A are the dynamics for target A and similarly for B. (This assumption would hold if, as is often the case, the dynamics of the targets were independent of each other.) One time step of the partitioned sampling algorithm consists of five steps: given a particle set $(s_i^t, \pi_i^t)_{i=1}^n$ which represents the posterior at time t, (i) resampling: just as in standard Condensation, sample the s_i with replacement, using probabilities proportional to the π_i, and set all weights in the new particle set to $1/n$; (ii) first partition of the dynamics: apply dynamics for target A only to all particles; (iii) weighted resampling: perform weighted resampling with respect to the importance function $\mathcal{P}_A$; (iv) second partition of dynamics: apply dynamics for target b only to all particles; (v) multiply by likelihood: multiply the weight π_i^{t+1} of each particle by the likelihood $p(\mathbf{Z} \mid s_i^{t+1})$. These steps are summarised by the following diagram:



(8)

The symbol $\sim \mathcal{P}_A$ means "perform weighted resampling with respect to the importance function $\mathcal{P}_A$", and the labels (a)–(d) refer to the example given later in Fig. 13. The validity of this algorithm is guaranteed by the following

Proposition 3. *If* $p(\mathbf{x}'' \mid \mathbf{x}) = \int_{\mathbf{x}'} p_B(\mathbf{x}'' \mid \mathbf{x}') p_A(\mathbf{x}' \mid \mathbf{x})$, *the posterior generated by diagram* (8) *is the same as that generated by diagram* (4).

Proof: It is easy to check the conditions of Proposition 2 are satisfied here: in tracking problems we can always restrict the configuration space to be closed and bounded; the weights before the weighted resampling operation are all equal so are certainly derived from a continuous function; and the importance function $\mathcal{P}_A$ is positive and continuous. So by Proposition 2, the reweighting operation $\sim \mathcal{P}_A$ has no effect (asymptotically, as the number of particles $N \to \infty$) on the distribution represented. Hence we may delete this step from the diagram without affecting the posterior. The step $* p_A(\mathbf{x}' \mid \mathbf{x})$ is now followed immediately by $* p_B(\mathbf{x}'' \mid \mathbf{x}')$ and by assumption the consecutive application of these steps is equivalent to $* p(\mathbf{x}'' \mid \mathbf{x})$. Making this substitution on the diagram, we obtain (4), as desired. □

Remark. It is clear from the proof that instead of $\mathcal{P}_A$ in diagram (8), one could use any strictly positive function without affecting the posterior. However the objective of partitioned sampling is to obtain an accurate representation of the posterior with a moderate number of particles. Hence we would like the weighted resampling step to position as many particles as possible near peaks in the posterior. Because we assumed target A partially occludes target B, the one-body density $\mathcal{P}_A$ is a good choice as importance reweighting function. Particles surviving the weighted resampling step lie in peaks of $\mathcal{P}_A$, and this function has peaks in the "right" place because target A is completely visible.

Example. Consider a simple example with a 2-dimensional configuration space; then each particle in a particle set can be schematically represented on a plane, with area proportional to its weight. Figure 12 uses this convention to illustrate one iteration of the conventional (non-partitioned) Condensation algorithm. Box (a) shows the prior—a Gaussian centred on the centre of the image. The black cross shows the actual position of the target, which of course is not known to the algorithm at this stage. Box (b) shows the distribution after the prior has been resampled and the dynamics (which in this case are isotropic additive Gaussian) have been applied. Note that at this point each particle has equal weight. In (c), the particles have the same configurations as in (b), but their weights are now proportional to the observation density. This is the particle representation of the posterior distribution.

Figure 13 shows the application of partitioned sampling in the same scenario. The dynamics and observations are partitioned into x and y components. Box (a) shows the same prior as in Fig. 12. In (b), the prior has been resampled and the $\mathbf{x}^A$-component of the dynamics has been applied. To produce (c), we first perform weighted resampling on these particles, with respect to an importance function centred on an observation of the $\mathbf{x}^A$-coordinate of the target. Recall that this has no effect on the distribution represented, but of course it selects many particles whose $\mathbf{x}^A$-coordinate is close to the target's—this will be beneficial later. Next the $\mathbf{x}^B$-component of the dynamics is applied, producing the particle set shown in (c). Finally, this set is multiplied

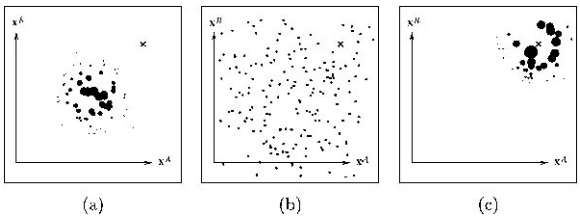


Figure 12. Conventional (i.e. non-partitioned) Condensation. The true position of the target in this 2-dimensional configuration space is shown as a cross; particles representing a probability distribution are shown as circles whose areas are proportional to their weights. Each step shown is one stage in the condensation diagram (4). (a) The prior. (b) After the dynamics have been applied. (c) After reweighting by the posterior. The posterior is centred at approximately the correct position, but this representation of the posterior is not very accurate because relatively few particles have significant weights. In technical terms, the estimated effective sample size (10) is low. Superior results are achieved using partitioned sampling (Figs. 13 and 14).

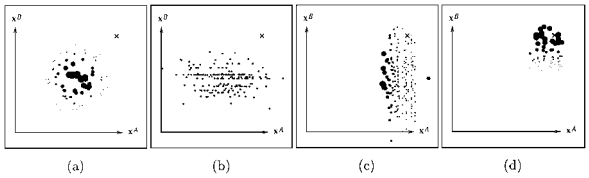


Figure 13. Partitioned sampling. A simple example implementing the condensation diagram (8). The 2-dimensional configuration space is partitioned as the cross product of the $\mathbf{x}^A$ and $\mathbf{x}^B$ dimensions, and the true position of the target is shown as a cross. (a) The prior. (b) The particles in (a) have been resampled, and dynamics have been applied in the $\mathbf{x}^A$-direction. (c) The weighted resampling operation has been performed, and the remaining dynamics (i.e. in the $\mathbf{x}_B$ direction) applied. (d) The particles in (c) are re-weighted by the posterior. Note how fine-grained the sample set for the posterior is, compared with the final set from conventional sampling in Fig. 12. In other words, this representation of the posterior has a higher estimated effective sample size (10) than that in Fig. 12.

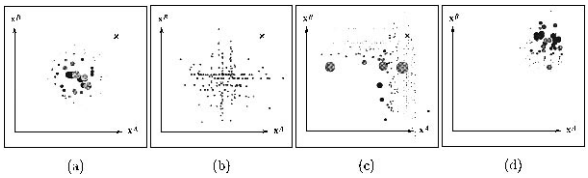


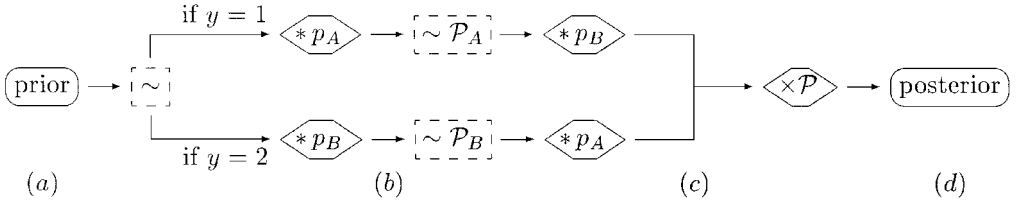
Figure 14. Branched partitioned sampling. Each step shows a stage from the Condensation diagram (9). The 2-dimensional configuration has been augmented with a binary variable y, shown as black ($y = 1$) or grey ($y = 2$), and the value of this variable determines which branch is taken in (9). (a) The prior. (b) Dynamics have been applied in the $\mathbf{x}^A$-direction for black particles and the $\mathbf{x}^B$-direction for grey particles. (c) The weighted resampling operation has been performed, and the remaining dynamics applied. (d) The particles from (c) are re-weighted by the posterior. The estimated effective sample size of the posterior is greater than for the unpartitioned method (Fig. 12) but in this simple example is no better than the non-branched, partitioned method (Fig. 13). However, that is because this example is symmetric in A and B: the branched method *would* be superior if the 2 importance functions $\mathcal{P}_A, \mathcal{P}_B$ used to produce (c) were not equally good predictors of particle position.

by the joint observation density for $\mathbf{x}^A$ and $\mathbf{x}^B$ coordinates. The result is shown in (d). Notice how dense this representation is, compared to the final outcome of non-partitioned sampling in Fig. 12.

5.3. Branched Partitioned Sampling

Branching is a refinement of partitioned sampling which is needed in our application to a mixed state Condensation tracker. In the discussion above, it was assumed target A partially occluded target B. This enabled us to select the one-body density $\mathcal{P}_A$ as a suitable importance function for the reweighting step in (8). However at any given time step, there are some particles for which $y = 1$ (i.e. A is unoccluded) and some for which $y = 2$ (i.e. B is unoccluded). It would be preferable to select a *different* importance function for each y value.

This is achieved by the *branched* partitioned sampling algorithm summarised on the following diagram:



(9)

Particles for which $y = 1$ follow the top path, which positions the $\mathbf{x}^A$-components first (near peaks in $\mathcal{P}_A$), since these particles believe A is unoccluded. Particles for which $y = 2$ follow the bottom path, since they believe B is unoccluded. The final result is that many more particles survive the resampling process, compared to the non-partitioned process, and the posterior is represented more accurately.

One technical point: the sum of weights π_i in any one branch need not be unity. Hence when performing weighted resampling, the new weights must be normalised to have the same sum as before the resampling.

Example. In Fig. 14, the 2-dimensional example has been augmented to include a binary discrete label, indicated by the colour of each particle (grey or black). The prior, (a), gives an equal weighting to the two discrete states. Box (b) shows the particle set one step after the branching: black particles have had the $\mathbf{x}^A$-component of the dynamics applied to them, whereas grey particles have received the $\mathbf{x}^B$-component. Box (c) shows the particle set after the branches merge again. The black

particles receive weighted resampling with respect to an observation of the target's $\mathbf{x}^A$-coordinate, while the grey particles receive weighted resampling with respect to an observation of the target's $\mathbf{x}^B$-coordinate. Then the remaining dynamics are applied: the $\mathbf{x}^A$ component to the grey particles, and the $\mathbf{x}^B$ component to the black particles. This results in (c). Finally, the weights are multiplied by the joint observation density for $\mathbf{x}^A$ and $\mathbf{x}^B$, producing the posterior shown in (d).

5.4. *Performance of Partitioned Sampling*

Evaluating the performance of particle filters such as Condensation is a difficult problem (Carpenter et al., 1999; Doucet, 1998; Kong et al., 1994; Liu and Chen, 1995, 1998). To compare the two schemes (9) and (4) we use Doucet's (Doucet, 1998) *estimated effective sample size* $\hat{N}$ defined for a set of particles with weights $\pi_1, \ldots, \pi_N$ as

$$\hat{N} = \left(\sum_{i=1}^{N} \pi_i^2 \right)^{-1} \tag{10}$$

Intuitively, this corresponds to the number of "useful" particles: if all have the same weight $1/N$ then $\hat{N} = N$, whereas if all but one of the weights are negligible we have $\hat{N} = 1$. Any other distribution of the weights falls between these two extremes. Figure 15 compares $\hat{N}$ for the conventional ("unpartitioned") and partitioned methods. It is clear that partitioned sampling achieves much higher values of $\hat{N}$ than unpartitioned sampling

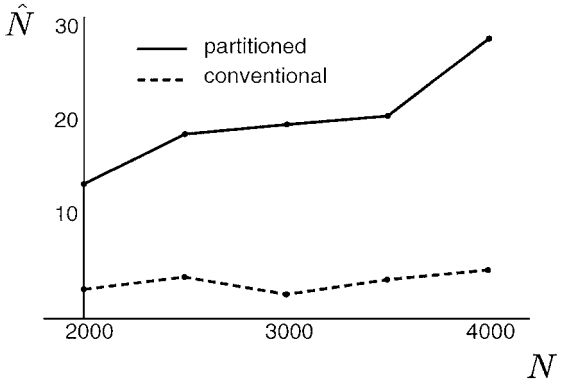


Figure 15. Estimated effective sample size $\hat{N}$ for partitioned and conventional (unpartitioned) sampling methods. The graph shows the average value of $\hat{N}$ following a 10-frame sequence tracking two leaves. Note the superior performance of the partitioned sampling method.

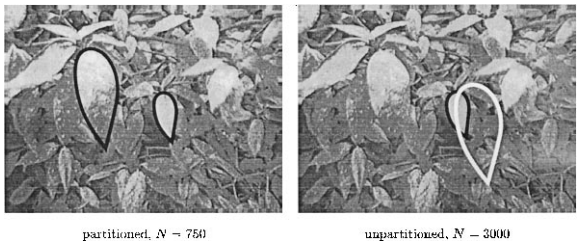


Figure 16. Unpartitioned sampling can fail when partitioned sampling does not, even if more particles are used. The final frame from a tracked sequence is shown: with unpartitioned sampling, the tracking fails despite using 4 times as many particles as the partitioned method.

and that we can therefore expect much better tracking performance for the same computational expense. We can show this is indeed the case in a practical example: Fig. 16 shows stills from a certain sequence tracked by each method. With partitioned sampling, and $N = 750$ particles, the tracking succeeds. However, despite using 4 times as many particles, unpartitioned sampling fails to track on the same sequence.

6. Conclusion

An exclusion principle for tracking multiple, indistinguishable targets has been introduced, which prevents a single piece of image data independently contributing to similar hypotheses for different targets. In its raw form, the model is valid only for wire-frame objects. However, by extending the tracking methodology to permit discrete states for describing the world in 2.1 dimensions, the same type of model can be used to track solid objects. Moreover, the approach requires only a simple model of the targets and no knowledge whatsoever of the background, which may itself be moving non-rigidly. A second contribution of the paper is to introduce partitioned sampling: a method of using particle filters with multiple objects, without incurring excessive additional computational cost for the extra dimensions.

The exclusion principle and the partitioned sampling algorithm were described and demonstrated for 2 targets. In principle, there are obvious generalisations to an arbitrary number of targets, but it remains to be seen whether these suffer from implementation difficulties.

So far the probabilistic exclusion principle has been developed for only the specific type of edge-based measurements described here. However, the fundamental idea is that any single measurement should reinforce

multiple hypotheses coherently; it is hoped this can be used to guide the implementation of exclusion principles for more general observation processes.

Acknowledgments

We are grateful for financial support from the EU (JM) and the Royal Society (AB).

Appendix A: Proof of Proposition 1

Remarks. Because of the discrete parameter n which indicates how many arguments z_i follow, the functions p_c are not quite probability density functions in the standard sense. However, this is a technical detail which can be avoided by explaining the notation more clearly. For example, $p_0(n; z_1, \ldots, z_n)$ is just shorthand for $p_0(z_1, \ldots, z_n \mid n, \nu)\text{Prob}(n)$, so that $p_0(n; z_1, \ldots, z_n)dz_1, \ldots, dz_n$ is just the probability of obtaining n features *and* that these features lie in the volume $dz_1, \ldots, dz_n$ centred on $\mathbf{z} = (z_1, \ldots, z_n)$.

Another subtle point is that each z_i is a point in the image, which would normally be described by an x and y coordinate. However in this context the features are constrained to lie on the measurement line, which is a one-dimensional subset of the image. So the notation dz_i refers to a small one-dimensional subset of the image.

Proof: The formula for p_0 follows almost immediately from the assumptions. By definition there is a probability $b(n)$ of obtaining n features, and these are distributed uniformly on the length L of the measurement line. Hence $p_0(n; z_1, \ldots, z_n) = b(n)/L^n$.

The formula for p_1 relies on a simple combinatoric argument. First note the generative model described above is equivalent to the following: (i) The number of background features, say m, is selected with probability $b(m)$. (ii) The positions of the background features are drawn from the uniform distribution on the measurement line, obtaining say $b_1, \ldots, b_m$. (iii) The position a of the boundary feature is selected by a random draw from $\mathcal{G}(a \mid \nu)$. (iv) The total number of features n is set to $m + 1$, and the vector $(a, b_1, \ldots, b_m)$ is randomly permuted and reported as $(z_1, \ldots, z_n)$. In mathematical terms, we can say that a permutation ρ is selected uniformly at random from the symmetric group S_n, and applied to the vector $(a, b_1, \ldots, b_m)$.

After stage (iii), the pdf $p(m; a, b_1, \ldots, b_m \mid \nu)$ of the unpermuted vector is just $b(m)\mathcal{G}(a \mid \nu)/L^m$, and since each of the $n!$ permutations has an equal probability we calculate

$$p_1(n; z_1, \ldots, z_n \mid \nu) = b(m) \sum_{\rho \in S_n} \frac{\mathcal{G}\big(z_{\rho(1)} \mid \nu\big)}{L^m} \times \frac{1}{n!}$$
$$= b(n-1) \sum_{k=1}^{n} \frac{\mathcal{G}(z_k \mid \nu)}{nL^{n-1}}$$

where the last line follows by collecting together the $(n-1)!$ permutations which leave z_k fixed.

The same type of reasoning leads to the stated formula for p_2. □

Appendix B: Sketch Proof of Proposition 2

A rigorous proof of Proposition 2 is given in (MacCormick, 2000), and related results can be found in (Doucet, 1998; Kong et al., 1994; Liu and Chen, 1995). However, the following informal proof is more intuitive and illuminating.

Sketch of proof. Set $\rho_i = g(s_i)/\sum_j g(s_j)$. Run step 1 of the weighted resampling algorithm, obtaining the s'_i and the unnormalised π'_i. Set $K = \sum_{i=1}^{n} \pi'_i$. We need the following lemma.

Lemma 1. *As $n \to \infty$, $K/n \to 1$ weakly.*

Informal proof of lemma. When n is large, each index $k \in \{1, \ldots, n\}$ is selected approximately $n\rho_k$ times. By collecting these together we can write

$$K = \sum_{i=1}^{n} \frac{\pi_i}{\rho_i} \approx \sum_{k=1}^{n} \frac{\pi_k}{\rho_k} n\rho_k = n \sum_{k=1}^{n} \pi_k = n$$

which completes the informal proof of the lemma.

Define indices $k_1, k_2, \ldots,$ so that $s'_i = s_{k_i}$. Then by the lemma we know the *normalised* weight π'_i is approximately $\pi_{k_i}/n\rho_{k_i}$. To complete the proof of Proposition 2 it will be enough to show that the total weight assigned to a value s_i is the same (as $n \to \infty$) in the initial and final particle sets. But this is now immediate: there are approximately $n\rho_{k_i}$ values equal to s'_i, and each has final weight $\pi_{k_i}/n\rho_{k_i}$. Thus the total

weight assigned to s'_i is $n\rho_{k_i} \times \pi_{k_i} / n\rho_{k_i} = \pi_{k_i}$, just as in the initial particle set $(s_i, \pi_i)_{i=1}^n$. □

References

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

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

Billingsley, P. 1995. *Probability and Measure*. 3rd edition, Wiley.

Blake, A. and Isard, M. 1998. *Active Contours*. Springer.

Carpenter, J., Clifford, P., and Fearnhead, P. 1999. An improved particle filter for non-linear problems. *IEE Proceedings—Radar, Sonar and Navigation*, 146:2–7.

Chellappa, R. and Jain, A. 1993. *Markov Random Fields: Theory and Application*. Academic Press.

Cootes, T. and Taylor, C. 1992. Active shape models. In *Proc. British Machine Vision Conf.*, pp. 265–275.

Del Moral, P. 1998. Measure-valued processes and interacting particle systems: application to nonlinear filtering problems. *The Annals of Applied Probability*, 8(2):438–495.

Doucet, A. 1998. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR310, Dept. of Engineering, University of Cambridge.

Gordon, N. 1997. A hybrid bootstrap filter for target tracking in clutter. *IEEE Trans. Aero. Elec. Systems*, 33:353–358.

Haritaoglu, I., Harwood, D., and Davis, L. 1998. w^4s: A real-time system for detecting and tracking people in 2.5D. In *Proc. 5th European Conf. Computer Vision*, Freiburg, Germany. Springer Verlag, Vol. 1, pp. 877–892.

Hinton, G., Williams, C., and Revow, M. 1992. Adaptive elastic models for hand-printed character recognition. *Advances in Neural Information Processing Systems*, 4.

Intille, S., Davis, J., and Bobick, A. 1997. Real-time closed-world tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 697–703.

Irani, M. and Anandan, P. 1998. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6).

Isard, M. and Blake, A. 1998a. Condensation—conditional density propagation for visual tracking. *Int. J. Computer Vision*, 28(1): 5–28.

Isard, M. and Blake, A. 1998b. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. 5th European Conf. Computer Vision*, pp. 893–908.

Isard, M. and Blake, A. 1998c. A mixed-state Condensation tracker with automatic model switching. In *Proc. 6th Int. Conf. on Computer Vision*, pp. 107–112.

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

Kent, J., Mardia, K., and Walder, A. 1996. Conditional cyclic markov random fields. *Adv. Appl. Prob. (SGSA)*, 28:1–12.

Koller, D., Weber, J., and Malik, J. 1994. Robust multiple car tracking with occlusion reasoning. In *Proc. 3rd European Conf. Computer Vision*, Springer-Verlag, pp. 189–196.

Kong, A., Liu, S., and Wong, W. 1994. Sequential imputations and Bayesian missing data problems. *J. Am. Stat. Assoc.*, 89(425): 278–288.

Liu, J. and Chen, R. 1995. Blind deconvolution via sequential imputations. *J. Am. Stat. Assoc.*, 90(430):567–576.

Liu, J. and Chen, R. 1998. Sequential monte carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93:1032–1044.

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

MacCormick, J. 2000. *Probabilistic models and stochastic algorithms for visual tracking.* Ph.D. Thesis, University of Oxford.

MacCormick, J. and Blake, A. 1998a. A probabilistic contour discriminant for object localisation. In *Proc. 6th Int. Conf. on Computer Vision*, pp. 390–395.

MacCormick, J. and Blake, A. 1998b. Spatial dependence in the observation of visual contours. In *Proc. 5th European Conf. Computer Vision*, pp. 765–781.

Mumford, D. and Gidas, B. 1999. Stochastic models for generic images. Technical report, Divison of Applied Mathematics, Brown University.

Mumford, D. and Nitzberg, M. 1990. The 2.1d sketch. In *Proc. 3rd Int. Conf. on Computer Vision*, pp. 138–144.

Paragios, N. and Deriche, R. 1998. A PDE-based level-set approach for detection and tracking of moving objects. In *Proc. 6th International Conf. Computer Vision*, pp. 1139–1145.

Peterfreund, N. 1998. Robust tracking with spatio-velocity snakes: Kalman filtering approach. In *Proc. 6th International Conf. Computer Vision*, pp. 433–439.

Pitt, M. and Shepherd, N. 1997. Filtering via simulation and auxiliary particle filters. Technical report, Nuffield College, University of Oxford.

Rasmussen, C. and Hager, G. 1998. Joint probabilistic techniques for tracking multi-part objects. In *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 16–21.

Szeliski, R. and Terzopoulos, D. 1991. Physically-based and probabilistic models for computer vision. In *SPIE Procs. Geometric methods in computer vision*, Vol. 1570, Vemuri, B. (Ed.).

Winkler, G. 1995. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer.

Zhu, S., Wu, Y., and Mumford, D. 1998. Filters, random fields and maximum entropy (FRAME). *Int. J. Computer Vision*, 27(2):107–126.