Partitioned sampling, articulated objects, and interface-quality hand tracking

John MacCormick[1] and Michael Isard[2]

[1] Dept. of Eng. Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK
jmac@robots.ox.ac.uk, http://www.robots.ox.ac.uk/~jmac
[2] Compaq Systems Research Center, 130 Lytton Ave, Palo Alto, CA 94301, USA
misard@pa.dec.com

1 Introduction

Partitioned sampling is a technique which was introduced in [17] for avoiding the high cost of particle filters when tracking more than one object. In fact this technique can reduce the curse of dimensionality in other situations too. This paper describes how to use partitioned sampling on articulated objects, obtaining results that would be impossible with standard sampling methods. Because partitioned sampling is the statistical analogue of a hierarchical search, it makes sense to use it on articulated objects, since links at the base of the object can be localised before moving on to search for subsequent links.

A new concept relating to particle filters, termed the *survival rate* is introduced, which sheds light on the efficacy of partitioned sampling. The domain of articulated objects also highlights two important features of partitioned sampling which are discussed here for the first time: firstly, that the number of particles allocated to each partition can be varied to obtain the maximum benefit from a fixed computational resource; and secondly, that the number of likelihood evaluations (the most expensive operation in vision-based particle filters) required can be halved by taking advantage of the way the likelihood function factorises for an articulated object.

Another important contribution of the paper is the presentation of a vision-based "interface-quality" hand tracker: a self-initialising, real-time, robust and accurate system of sufficient quality to be used for complex interactive tasks such as drawing packages. The tracker models the hand as an articulated object and partitioned sampling is the crucial component in achieving these favourable properties. The system tracks a user's hand on an arbitrary background using a standard colour camera, in such a way that the hand can be employed as a 4-dimensional mouse (planar translation and the orientations of the thumb and index finger).

Hand gesture recognition is the subject of much research, for a wide variety of applications and by a plethora of methods. Kohler and Schröter [13] give a comprehensive survey. We are not aware of any hand tracking system which combines the speed, robustness, accuracy and simple hardware requirements of the system described here. Among the more successful systems which recover

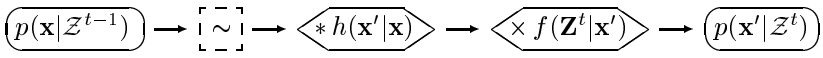
continuous parameters (rather than recognising gestures from a discrete "vocabulary"), some use a stereo rig (e.g. [5, 20]), some are not real time (e.g. [3, 9]), while others do not appear to have sufficient accuracy for the applications envisaged here (e.g. [1, 2, 8, 11, 12]). Of these, [1, 11] are the closest to our system in terms of the method used. In both cases, the tracking is good enough to permit navigation through a virtual environment, but not for the fine adjustment of interactive visual tools (e.g. drawing at pixel accuracy).

2 Partitioned sampling and the efficiency of particle filters

Partitioned sampling is a way of applying particle filters (also known as the Condensation algorithm e.g. [11]) to tracking problems with high-dimensional configuration spaces, without incurring the large computational cost that would normally be expected in such problems. In this section we first review particle filters, then explain why the large computational cost arises, and finally describe the basic idea behind partitioned sampling.

2.1 Particle filters

Consider a tracking problem with configuration space $\mathcal{X} \subset \mathbb{R}^d$. Recall that Condensation expresses its belief about the system at time t by approximating the posterior probability distribution $p(\mathbf{x}|\mathcal{Z}^t)$, where $\mathcal{Z}^t$ is the history of observations $\mathbf{Z}^1, \ldots \mathbf{Z}^t$ made at each time step, and $\mathbf{x} \in \mathcal{X}$. The distribution $p(\mathbf{x}|\mathcal{Z}^t)$ is approximated using a *weighted particle set* $(\mathbf{x}_i, \pi_i)_{i=1}^n$, which can be interpreted as a sum of δ-functions centred on the $\mathbf{x}_i$ with real, non-negative weights π_i (one requires that $\sum_i \pi_i = 1$). Each time step of the Condensation algorithm is just an update according to Bayes' formula, implemented using operations on particle sets which can be shown to have the desired effects (as $n \to \infty$) on the underlying probability distributions. One step of Condensation can be conveniently represented on a diagram as follows:



(1)

where the $\sim$ symbol denotes resampling, $*$ denotes convolving with dynamics, and $\times$ denotes multiplication by the observation density. Specifically, the resampling operation $\sim$ maps $(\mathbf{x}_i, \pi_i)_{i=1}^n$ to $(\mathbf{x}'_i, 1/n)_{i=1}^n$, where each $\mathbf{x}'_i$ is selected independently from the the $\{\mathbf{x}_1, \ldots \mathbf{x}_n\}$ with probability proportional to π_i. This operation has no effect on the distribution represented by the particle set, but often helps to improve the efficiency with which it is represented. The dynamical convolution operation $*$ maps $(\mathbf{x}_i, \pi_i)_{i=1}^n$ to $(\mathbf{x}'_i, \pi_i)_{i=1}^n$, where $\mathbf{x}'_i$ is a random draw from the conditional distribution $h(\mathbf{x}'|\mathbf{x}_i)$. Its effect on the distribution represented by the particle set is to transform a distribution $p(\mathbf{x})$ into $\int h(\mathbf{x}'|\mathbf{x})p(\mathbf{x})d\mathbf{x}$. Finally, the multiplication operation $\times$ maps $(\mathbf{x}_i, \pi_i)_{i=1}^n$ to $(\mathbf{x}_i, \pi'_i)_{i=1}^n$, where $\pi'_i \propto \pi_i f(\mathbf{Z}^t|\mathbf{x}_i)$. Its probabilistic effect is to transform a

distribution $p(\mathbf{x})$ into the distribution proportional to $p(\mathbf{x})f(\mathbf{Z}^t|\mathbf{x})$. Hence, the overall effect of diagram (1) on the distribution $p(\mathbf{x}|\mathcal{Z}^{t-1})$ is to transform it into the distribution proportional to $f(\mathbf{Z}^t|\mathbf{x}')\int h(\mathbf{x}'|\mathbf{x})p(\mathbf{x}|\mathcal{Z}^{t-1})d\mathbf{x}$ — precisely the Bayes update rule for dynamical diffusion governed by $h(\mathbf{x}'|\mathbf{x})$ and likelihood function $f(\mathbf{Z}^t|\mathbf{x}')$.

2.2 The survival diagnostic and survival rate

In assessing the efficacy of particle filters we have found two quantities to be of use: the *survival diagnostic* $\mathcal{D}$ and the *survival rate* α. The survival diagnostic[1] is defined for a particle set $(\mathbf{x}_i, \pi_i)_{i=1}^n$ as

$$\mathcal{D} = \left(\sum_{i=1}^{n} \pi_i^2\right)^{-1}. \tag{2}$$

Intuitively, it can be thought of as indicating the number of particles which would survive a resampling operation. Two extreme cases make this clear. If $\pi_1 = 1$ and all the other weights are zero, then $\mathcal{D} = 1$ — only one particle will survive the resampling. On the other extreme, if every weight is equal to $1/n$, then $\mathcal{D} = n$. In this case, every particle would be chosen exactly once by an ideal resampling operation, so all n particles would survive.[2] Any particle set lies somewhere between these two extremes. The survival diagnostic indicates whether tracking performance is reliable or not: a low value of $\mathcal{D}$ indicates that estimates (e.g. of the mean) based on the particle set may be unreliable, and that there is significant danger of the tracker losing lock on its target. The difficult problem of assessing the performance of particle filters is discussed in the statistical literature (e.g. [4, 6, 7, 14]) and no single approach has met with resounding success. In our experience, the survival diagnostic is as useful as any other indicator and has the significant advantage of having negligible computational cost.

Whereas the survival diagnostic is a property of a given particle set, the *survival rate* is a property of a given prior $p(\mathbf{x})$ and posterior $p'(\mathbf{x})$. Specifically, the survival rate is given by

$$\alpha = \left(\int p'(\mathbf{x})^2/p(\mathbf{x})\, d\mathbf{x}\right)^{-1}. \tag{3}$$

(See theorem 2 of [7] for another use of this quantity.) Again, a special case is instructive. Suppose p is a uniform distribution on a set $\mathcal{X}_p \subset \mathcal{X}$ of volume V_p, and that p' is also uniform, on a smaller subset $\mathcal{X}_{p'} \subset \mathcal{X}_p$ of volume $V_{p'}$. Then p'/p is equal to $V_p/V_{p'}$ everywhere on $\mathcal{X}_{p'}$, so that $\alpha = V_{p'}/V_p$. That is,

[1] Doucet [6] calls it the *estimated effective sample size*. See also [4].

[2] In fact, if truly random resampling is employed, a certain fraction of the particles would not survive even in this case. But in practice one uses a deterministic version of the resampling operation which selects every particle the appropriate number of times.

the survival rate is just the ratio of the volume of the posterior to the volume of the prior. It turns out that this interpretation is valid in more general cases too. Let $l(\mathbf{x}) = p'(\mathbf{x})/p(\mathbf{x})$ be the likelihood function and define a particle set $(\mathbf{x}_i, \pi_i)$ which represents p' by letting the $\mathbf{x}_i$ be i.i.d. draws from $p(\mathbf{x})$ and setting $\pi_i = l(\mathbf{x}_i)$. Then it can be shown (see appendix) that for large n,

$$\mathcal{D} \approx \alpha n. \tag{4}$$

This explains our terminology: α is called the survival rate because when multiplied by n it is approximately the number of particles expected to survive a resampling. Hence we expect the overall tracking performance to be related to the survival rate at each time step: if the survival rate is too low, the tracker will be in danger of producing inaccurate estimates or losing lock altogether.

Example Figure 1 shows an example of a survival rate α calculated for a contour likelihood in a real image. In this particular example, in which the configuration space is the one-dimensional interval $[-150, 150]$, the value of α was calculated numerically as 0.20. (In more realistic multi-dimensional examples, typical values of α are much lower than this.) Equation (4) can also be verified directly by simulations for this simple example.

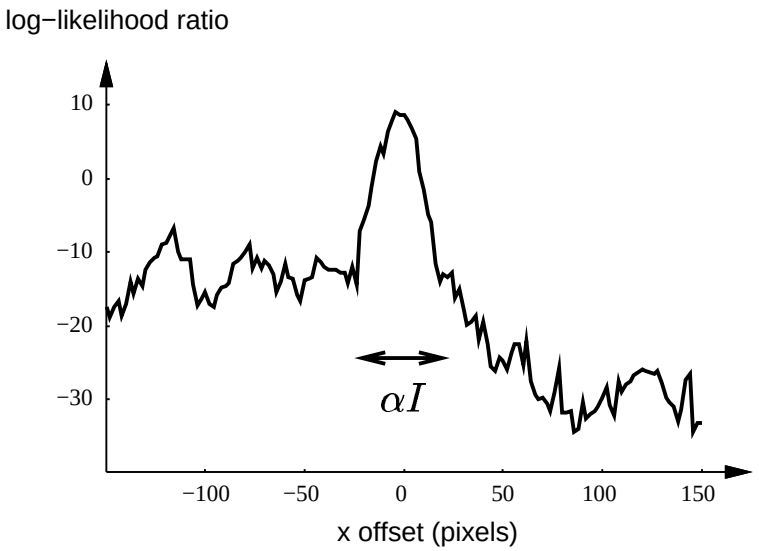


Fig. 1. Survival rate. A contour likelihood of the kind used in section 3.1 is graphed for a range of offsets in the x-direction from a template. Taking a uniform prior p on the interval $I = [-150, 150]$, the survival rate α for this particular likelihood function can be calculated numerically as 0.20. This corresponds to the "volume" αI indicated on the graph.

2.3 More dimensions means more particles

The survival rate concept makes it easy to see why particle filters require so many extra particles to achieve the same level of performance as the dimension of the configuration space increases. An informal argument runs as follows. Fix (by trial and error, if necessary) a value $\mathcal{D}_{\min}$ which represents the minimum acceptable survival diagnostic for successful tracking of a given object with a given steady-state prior $p(\mathbf{x})$ on a configuration space $\mathcal{X}$. Then according to (4), we should take $n \geq \mathcal{D}_{\min}/\alpha$ to achieve $\mathcal{D} \geq \mathcal{D}_{\min}$, where α is the survival rate for this particular problem. Now consider tracking two such objects. By the definition of the survival rate (3), it is easy to see the survival rate for the two-object problem is α^2, so that to achieve the same level of tracking performance (i.e. the same minimum survival diagnostic) we must take $n \geq \mathcal{D}_{\min}/\alpha^2$. Since typically $\alpha \ll 1$, this is a substantial additional requirement. Note this does not contradict the well-known result that the variance of standard Monte Carlo estimators is independent of the dimension of the configurations space. The general recipe of "sample from a prior, then weight by a likelihood" can be regarded as a type of importance sampling, and it is well-known that importance sampling scales badly with dimension. [18] gives a lucid explanation of these phenomena.

Partitioned sampling essentially eliminates the need for these additional particles. The intuition that α is the ratio of the posterior and prior volumes gives a hint as to how this problem could be solved. Take the simple case of tracking 2 objects A and B, whose configurations are described respectively by the one-dimensional variables $x_A, x_B \in [0, 1]$. Suppose the survival rate for the one-object problem is α: then as remarked above, we have a survival rate $\alpha' = \alpha^2$ for the two-object problem. Figure 2 shows a schematic representation of the situation. The intuition behind partitioned sampling is that instead of searching the entire unit square for the lightly shaded area α', we can divide the search into two stages: first, a search of the horizontal axis only, which will attempt to populate the dark shaded area α. This step will have survival rate α. Second, we try to populate the lightly shaded area. This second step will also have survival rate of approximately α, since the *relative* area of the dark shade to light shade is α'/α. This is the key idea behind partitioned sampling. It remains to show how we can "populate" certain parts of the configuration space with particles in the desired manner. This is done using an operation on particle sets called weighted resampling.

2.4 Weighted resampling

Let $g(\mathbf{x})$ be a strictly positive, continuous function on $\mathcal{X}$ called the *weighting function*. The weighted resampling function is analogous to the importance function used in standard importance sampling [19]. Weighted resampling with respect to g is an operation on a particle set which "populates" the peaks of g with particles, without altering the distribution actually represented by the particle set. Given a particle set $(\mathbf{x}_i, \pi_i)_{i=1}^n$, weighted resampling produces a new set $(\mathbf{x}'_i, \pi'_i)_{i=1}^n$ as follows. First define some "importance" weights $\rho_i =$

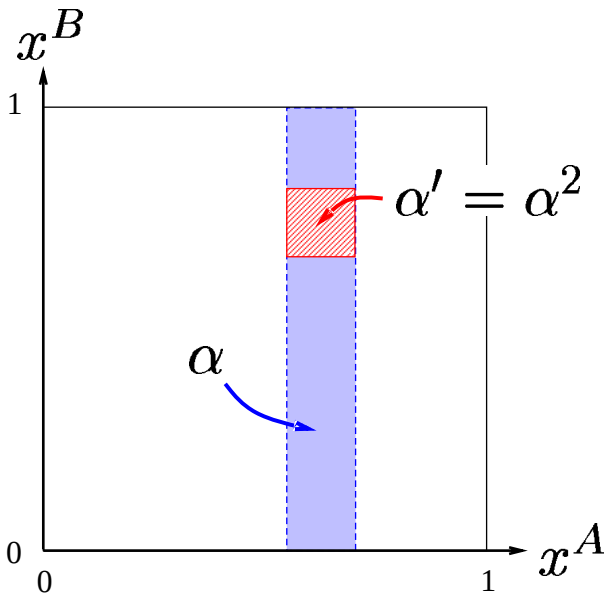
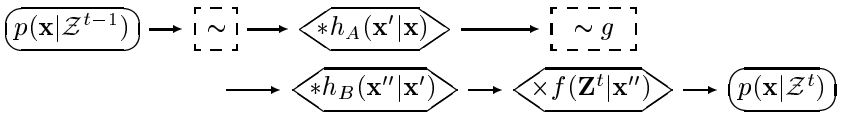


Fig. 2. Intuition behind partitioned sampling. To locate the peak of a 2D likelihood function, which has area $\alpha' = \alpha^2$, the search is split into two stages, each of which has survival rate α. The first stage populates the dark shaded area with particles, and the second stage populates the light shaded area.

$g(\mathbf{x}_i) / \sum_{j=1}^{n} g(\mathbf{x}_j)$. Next, select indices $k_1, k_2, \ldots k_n$ by setting $k_i = j$ with probability ρ_j, independently for $i = 1, \ldots n$. Finally, set $\mathbf{x}'_i = \mathbf{x}_{k_i}$ and $\pi'_i = \pi_{k_i} / \rho_{k_i}$. This last choice of weights has the effect of precisely counteracting the extent to which the particles were "biased" by the importance weights. A proof that the weighted resampling operation does not alter the underlying distribution can be found in [15]. On a Condensation diagram, the operation of weighted resampling with respect to g is denoted $\sim g$.

2.5 Partitioned sampling

Partitioned sampling is a generic term for the strategy which consists of dividing the state space into two or more "partitions", and sequentially applying the dynamics for each partition followed by an appropriate weighted resampling operation. For example, the two-object problem described above could be implemented as the following condensation diagram:



(5)

where we have assumed the dynamics can be decomposed as

$$h(\mathbf{x}''|\mathbf{x}) = \int_{\mathbf{x}'} h_B(\mathbf{x}''|\mathbf{x}')h_A(\mathbf{x}'|\mathbf{x})d\mathbf{x}'. \tag{6}$$

The algorithm is formally valid for any choice of h_A, h_B satisfying (6), and for any g; the objective of partitioned sampling is to use one's intuition about the problem to choose a decomposition of the dynamics, and a weighting function g, which are beneficial. In the example shown in figure 2, the overall strategy is to populate the dark shaded region first, so h_A should be such that some particles are diffused into dark region, g should be peaked in the dark region, and h_B should be such that particles already in the dark region are not diffused out of it. A natural choice, therefore, is to take h_A to be the dynamics for object A, g to be a likelihood function for the location of object A only, and h_B to be the dynamics for object B. This was the approach taken by the authors of [17].

3 Partitioned sampling for articulated objects

Although the preceding discussion was phrased for clarity in terms of multiple objects, partitioned sampling is not restricted to improving the efficiency of multiple object tracking. In fact, it can be used whenever the following conditions hold.

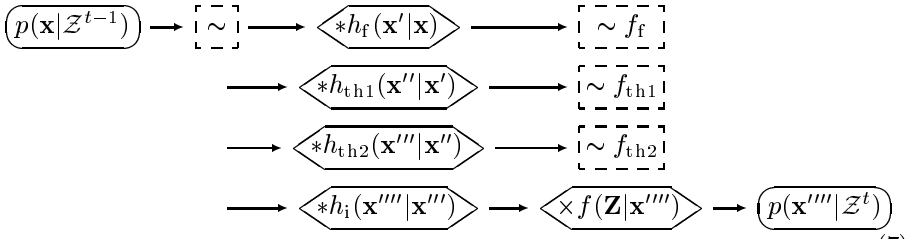
- The configuration space $\mathcal{X}$ can be partitioned as a Cartesian product $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$.
- The dynamics h can be decomposed as $h = h_1 * h_2$, where h_2 acts on $\mathcal{X}_2$. This means that if $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ and $\mathbf{x}' = (\mathbf{x}'_1, \mathbf{x}'_2)$ with $\mathbf{x}_i, \mathbf{x}'_i \in \mathcal{X}_i$, and $\mathbf{x}'$ is a random draw from $h_2(\cdot|\mathbf{x})$, then $\mathbf{x}'_1 = \mathbf{x}_1$. Informally, the second partition of the dynamics does not change the value of the projection of any particle into the first partition of the configuration space.[3] We refer to this later as property $(\star)$.
- A weighting function g_1 defined on $\mathcal{X}_1$ is available, which is peaked in the same region as the posterior restricted to $\mathcal{X}_1$.

There is also an obvious generalisation to $k > 2$ partitions: the configuration space is partitioned as $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_k$, the dynamics as $h = h_1 * \ldots * h_k$ with each h_j acting on $\mathcal{X}_j \times \ldots \times \mathcal{X}_k$, and we have weighting functions $g_1, g_2, \ldots g_{k-1}$ with each g_j peaked in the same region as the posterior restricted to $\mathcal{X}_j$.

One example of such a system is an articulated object. The example given in this paper is of a hand tracker which models the fist, index finger and thumb as an articulated rigid object with three joints. The partitioned sampling algorithm

[3] This condition is stronger than necessary, but a more general discussion would obscure the important idea.

used for this application is shown in the following Condensation diagram:



(7)

The subscript 'f' stands for "fist", 'th1' for "first thumb joint", 'th2' for "second thumb joint", and 'i' for "index finger". So the configuration space is partitioned into 4 parts:

- $\mathcal{X}_{\mathrm{f}} \equiv$ scale, orientation, and x and y translation of the fist
- $\mathcal{X}_{\mathrm{th1}} \equiv$ joint angle of base of thumb
- $\mathcal{X}_{\mathrm{th2}} \equiv$ joint angle of tip of thumb
- $\mathcal{X}_{\mathrm{i}} \equiv$ joint angle of index finger

The dynamics are decomposed as $h = h_{\mathrm{f}} * h_{\mathrm{th1}} * h_{\mathrm{th2}} * h_{\mathrm{i}}$ with the last three operations consisting of a deterministic shift plus Gaussian diffusion within the appropriate partition only. Note that although $\mathcal{X}$ is a shape space of splines, it is not described by the linear parameterisation normally used for such shape spaces. Instead it is parameterised by the 7 physical variables listed above (scale, orientation, x and y translation, and the 3 joint angles), so that any $\mathbf{x}$ is an element of $\mathbb{R}^7$.

3.1 Likelihood function and weighting functions

It remains to specify the measurement likelihood $f(\mathbf{Z}|\mathbf{x})$. Recall that the parameters $\mathbf{x}$ correspond to a B-spline in the image. A one-dimensional grey-scale edge operator is applied to the normal lines to this B-spline at 28 points (8 on the main hand, 6 on each of the thumb joints and 8 on the index finger). Each of the 28 resulting "edges" (actually points which are the nearest above-threshold responses of a 1D operator) has a normal distance ν_i from the B-spline, which would be zero if the model fitted the image edges perfectly. By assuming (i) the deviations of the model from the template shape are Gaussian, (ii) that such deviations are independent on different normal lines, and (iii) there is a fixed probability of finding no edge, it is easy to see that the form of $f(\mathbf{Z}|\mathbf{x})$ should be

$$\log f(\mathbf{Z}|\mathbf{x}) \propto \mathrm{const} + \sum_m {\nu_m}^2, \tag{8}$$

where the constant was set by hand for this application. We can also exploit the fact that the portion of a normal line on the *interior* of the B-spline should be skin-coloured. This is reflected by adding to (8) the output of correlating the

(colour) normal line pixel values with a colour template. Full details on densities of the form (8) can be found in [15, 16], for example.

Recall there are 28 measurement lines on the hand template: 8 on the fist, 6 on each of the thumb joints and 8 on the index finger. Since the likelihood factorises as a product of likelihoods for individual measurement lines, this gives us a convenient way to re-express the likelihood:

$$f(\mathbf{Z}|\mathbf{x}) = f_{\mathrm{f}}(\mathbf{Z}_{\mathrm{f}}|\mathbf{x}_{\mathrm{f}})\, f_{\mathrm{th1}}(\mathbf{Z}_{\mathrm{th1}}|\mathbf{x}_{\mathrm{f}}, \mathbf{x}_{\mathrm{th1}})\, f_{\mathrm{th2}}(\mathbf{Z}_{\mathrm{th2}}|\mathbf{x}_{\mathrm{f}}, \mathbf{x}_{\mathrm{th1}}, \mathbf{x}_{\mathrm{th2}})\, f_{\mathrm{i}}(\mathbf{Z}_{\mathrm{i}}|\mathbf{x}_{\mathrm{f}}, \mathbf{x}_{\mathrm{i}}) \quad (9)$$

where, for example, $\mathbf{Z}_{\mathrm{f}}$ are the measurements on the 8 fist locations, $\mathbf{x}_{\mathrm{f}}$ are the components of $\mathbf{x}$ which specify the configuration of the fist, and similarly for the other subscripts.

The factorisation (9) immediately suggests the use of f_{f}, f_{th1} and f_{th2} as weighting functions, since they should be peaked at the correct locations of the fist and thumb joints respectively. This is precisely what the implementation does; hence the presence of f_{f}, f_{th1} and f_{th2} on diagram (7).

3.2 Dividing effort between the partitions

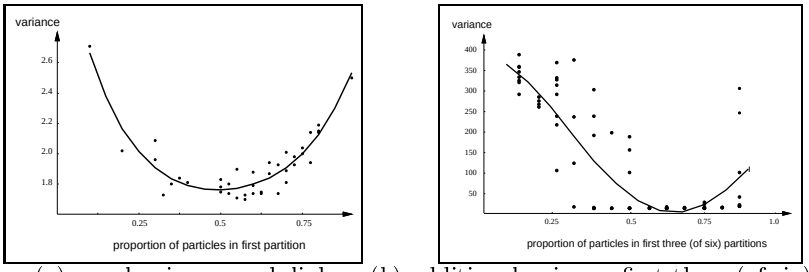
An important advantage of partitioned sampling is that the number of particles devoted to each partition can be varied. Partitions which require a large number of particles for acceptable performance can be satisfied without incurring additional effort in the other partitions. For instance, in the hand tracking application, the fist often moves rapidly and unpredictably whereas the joint angles of finger and thumb tend to change more slowly. Hence we use $n_1 = 700$ particles for the fist partition, but only $n_2 = n_3 = 100$ particles for the two thumb partitions and $n_4 = 90$ for the index finger partition. A glance at diagram (7) shows this produces a substantial saving, since at every time-step we avoid calculating $f_{\mathrm{th1}}(\mathbf{Z}_{\mathrm{th1}}|\mathbf{x})$, $f_{\mathrm{th2}}(\mathbf{Z}_{\mathrm{th2}}|\mathbf{x})$ and $f_{\mathrm{i}}(\mathbf{Z}_{\mathrm{i}}|\mathbf{x})$ for over 600 values of $\mathbf{x}$ that would otherwise have been required.

Note that the analysis of section 2.3, in terms of survival rates, cannot necessarily be used to determine the optimum allocation of particles between the partitions. If the dynamics and observations in each partition are completely independent, and inaccuracies in the estimated parameters for each partition are equally costly, then one can show that the number of particles in each partition should be inversely proportional to the survival rate for that partition. However, these conditions are never satisfied for an articulated object. Indeed, almost the opposite is true in the hand-tracking case. For one thing, since the intended application is a drawing tool based on the position of the finger tip, inaccuracies of many pixels are acceptable in the fist position, provided only that lock is not lost on the finger tip. However, even small errors in the finger tip position will degrade the performance of the drawing tool greatly. Thus one might think that the majority of particles should be devoted to the finger tip partition.

Two factors militate against this conclusion, however. One is that the precise location of the finger tip is in fact determined by an auxiliary least-squares fitting operation mentioned later; hence the imperative for accuracy in this partition is

not so great. Second, it is of overwhelming importance that lock is not lost on the fist, because the search lines for locating finger and thumb are placed relative to the fist. Experiment showed that this factor is the most crucial, which is why the majority of particles are devoted to the fist partition.

So far we have not been able to develop a coherent theory for choosing how to allocate the particles between partitions in such cases, and can only recommend careful experimentation. Some insight can be gained by studying simulated data, however. Figure 3 shows the results of tracking a simulated articulated chain with several links. The state space is divided into one partition for each link, and a fixed number of particles was divided between these in various ways. The graphs show the variance (in pixels^2) of the end-point of the articulated object, as estimated by partitioned sampling averaged over 200 frames. Several different runs were made for each set of parameter values; the curves shown are the best-fit (least-squares) quartics through all data points. Figure 3(a) is for a 3-link object whose dynamics have equal variance at each link. A total of 300 particles were available; 100 were allocated to the final partition and the remaining particles divided between the first two partitions. Because the dynamics and likelihood function are the same for each partition, the survival rates are similar for each partition, and as we might expect, the minimum variance is achieved by equally dividing these particles between the first two partitions.



(a) equal noise on each link (b) additional noise on first three (of six) links

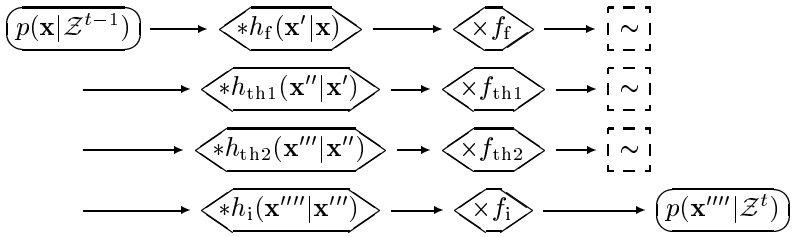
Fig. 3. Allocating resources to different partitions. *(a) Because the variance of the dynamics for each link is equal, the survival rate for each partition is approximately the same and the best allocation of particles is to distribute them evenly between partitions. (b) Now the variance of the dynamics on early partitions is 9 times higher than the later ones, so the survival rates on early partitions are lower and it is best to devote a higher proportion of the particles to these partitions.*

Figure 3(b) is a more extreme example. Now there are 6 links, and the first three links have dynamics which are much "noisier" than the last three links. Specifically, the first three links have the same dynamics $h_{123}(\cdot|\mathbf{x})$ and the last three share a different conditional density $h_{456}(\cdot|\mathbf{x})$ for their dynamics. The densities h_{123}, h_{456} were Gaussian with $\mathrm{var}(h_{123}) = 9\mathrm{var}(h_{456})$. Because of the higher variance of the dynamics, the survival rate for particles in the first three parti-

tions is lower than those in the last three; hence we expect that it will be most efficient to devote the majority of particles to the first three partitions. This is indeed the case; from the graph it appears that the best results are achieved when 60-70% of the particles are devoted to the first three partitions. Notice the extremely high variances for many data points outside this range: these are caused by the tracker losing lock on the early partitions.

3.3 Articulated objects can be evaluated twice as fast

In the particular case in which the overall likelihood $f(\mathbf{Z}|\mathbf{x})$ can be expressed as a product (9) of the weighting functions and another easily calculated function (in this case, f_{i}), the diagram (7) can be given a simpler form which uses standard resampling rather than weighted resampling:



(10)

One can check the equivalence by just writing out in detail the algorithm described by each diagram. The key is property ($\star$) mentioned in section 3: e.g. the "fist" component $\mathbf{x}_{\mathrm{f}}$ of a particle does not change after the fist partition, so the value of f_{f} for the particle does not change either. In other words, the evaluation of any given importance function commutes with the dynamics from subsequent partitions.

The reformulation of (7) as (10) is important because the computational expense of the hand tracking largely resides in evaluating the likelihood functions. Using diagram (7), the likelihood of each measurement line (except those on the index finger) is evaluated twice — once as part of a weighting function, and once as part of the final likelihood function. In diagram (10), each measurement line is examined only once.

3.4 Other details

Initialisation and re-initialisation are handled by the ICondensation mechanism of [11]. Various standard tools, such as background subtraction (which can be performed on an SGI Octane very cheaply using the alpha-blending hardware), and least-squares fitting of an auxiliary spline to the tip of the index finger, are used to refine the performance of the tracker. Details of these tools can be found in our technical report [10].

4 Results: a vision-based drawing package

The hand tracker described in the previous section was implemented on an SGI Octane with a single 175MHz R10000 CPU. Using 700 samples for the hand base, 100 samples for each of the thumb joints and 90 samples for the index finger, the tracker consumes approximately 75% of the machine cycles, which allows real-time operation at 25Hz with no dropped video frames even while other applications are running on the machine. The tracker is robust to clutter (figure 4), including skin-coloured objects (figure 5). The position of the index finger is located with considerable precision (figure 5) and the two articulations in the thumb are also recovered with reasonable accuracy (figure 6).

Fig. 4. Heavy clutter *does not hinder the hand tracker. Even moving the papers on the desk to invalidate the background subtraction does not prevent the Condensation tracker functioning. The fingertip localisation is less robust, however, and jitter increases in heavily cluttered areas.*

We have developed a simple drawing package to explore the utility of a vision-based hand tracker for user-interface tasks. The tracking achieved is sufficiently good that it can compete with a mouse for freehand drawing, though (currently) at the cost of absorbing most of the processing of a moderately powerful workstation. It is therefore instructive to consider what additional strengths of the vision system we can exploit to provide functionality which could not be reproduced using a mouse.

The current prototype drawing package provides only one primitive, the freehand line. When the thumb is extended, the pointer draws, and when the thumb is placed against the hand the virtual pen is lifted from the page. Immediately we can exploit one of the extra degrees of freedom estimated by the tracker, and use the orientation of the index finger to control the width of the line be-

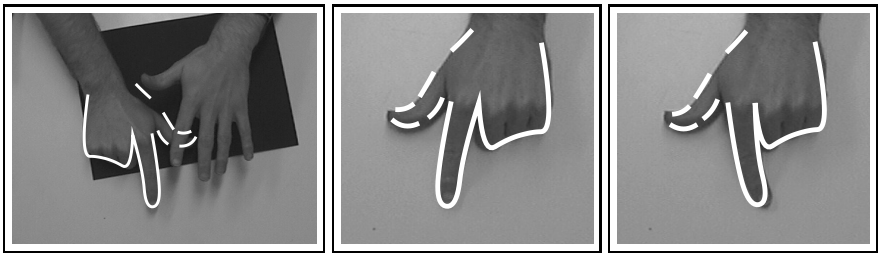


Fig. 5. Left: **Skin-coloured objects** *do not distract the tracker. Here two hands are present in the image but tracking remains fixed to the right hand. If the right hand were to leave the field of view the tracker would immediately reinitialise on the left hand.* Middle and right: **The index finger** *is tracked rotating relative to the hand body. The angle of the finger is estimated with considerable precision, and agile motions of the fingertip, such as scribbling gestures, can be accurately recorded.*

ing produced. When the finger points upwards on the image, the pen draws with a default width, and as the finger rotates the width varies from thinner (finger anti-clockwise) to thicker (finger clockwise) — see figure 7. The scarcity of variable-thickness lines in computer-generated artwork is a testament to the difficulty of producing this effect with a mouse.

The fact that a camera is observing the desk also allows other intriguing features not directly related to hand-tracking. We have implemented a natural interface to translate and rotate the virtual workspace for the modest hardware investment of a piece of black paper (figure 8). The very strong white-to-black edges from the desk to the paper allow the paper to be tracked with great precision using a simple Kalman filter, at low computational cost. Translations and rotations of the paper are then reflected in the virtual workspace, a very satisfying interface paradigm. While one hand draws, the other hand can adjust the workspace to the most comfortable position. Figure 9 is a still from a movie which shows the system in action; this movie is available at [15]. In the future it should be possible to perform discrete operations such as switching between drawing tools using simple static gesture recognition on one of the hands. Tracking both hands would allow more complex selection tasks, for example continuous zooming, or colour picking.

5 Conclusion

It has been shown that the technique of partitioned sampling can be applied to articulated objects. A new concept termed the "survival rate" of particles in a particle filter was used to explain why partitioned sampling works, and some special features of the application to articulated objects were exploited for significant computational improvements. Although some progress has been made, the question of how to allocate a fixed number of particles between partitions has

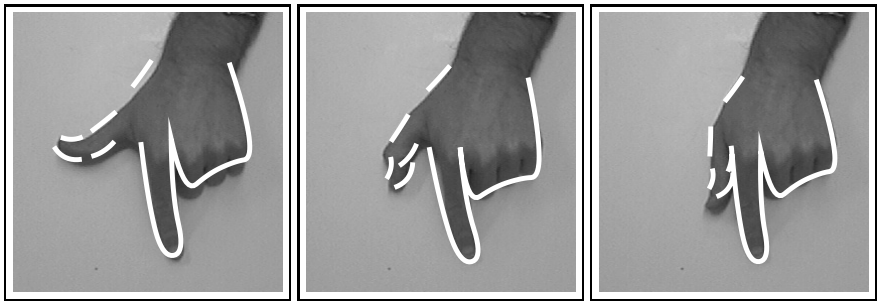


Fig. 6. The two degrees of freedom of the thumb *are tracked. The thumb angles are not very reliably estimated. This is probably partly because the joints are short, and so offer few edges to detect, and more importantly because the shape model gives a poor approximation to the thumb when it opposes. The gross position of the thumb can be extracted consistently enough to provide a stable switch which can be used analogously to a mouse button.*

not been answered coherently and this must be the subject of future work. Another open problem, not previously mentioned, is that our current "articulated partitioned" approach takes no account of the tree structure of the object: every link must be sampled as a chain even though the physical structure is a tree. Our present approach is valid mathematically, but it would be more appropriate, and possibly more efficient, to take account of the tree structure.

A hand-tracking system using partitioned sampling on articulated objects was described. It is of sufficient quality for very demanding interactive tasks. The main features of the system are robustness (from the Condensation algorithm), instantaneous initialisation and near-perfect responsiveness (from importance sampling based on colour segmentation) and inexpensive addition of extra degrees of freedom (from partitioned sampling). The system runs on a single-processor workstation with a standard colour camera and no additional hardware. Even in the simple drawing package described it is easily possible to produce figures which could not comfortably be produced with a mouse, and to do so using natural gestures and a natural, changing desk environment. We believe this system has significant implications for the everyday use of virtual environments with interactive computer vision.

A Appendix

An informal proof of (4) follows; more details can be found in [15]. Recall the scenario of section 2.2: a particle set $(\mathbf{x}_i, \pi_i)_{i=1}^n$ has been formed with prior (or "proposal density") $p(\mathbf{x})$ and weighted by likelihood $p'(\mathbf{x})/p(\mathbf{x})$, resulting in a

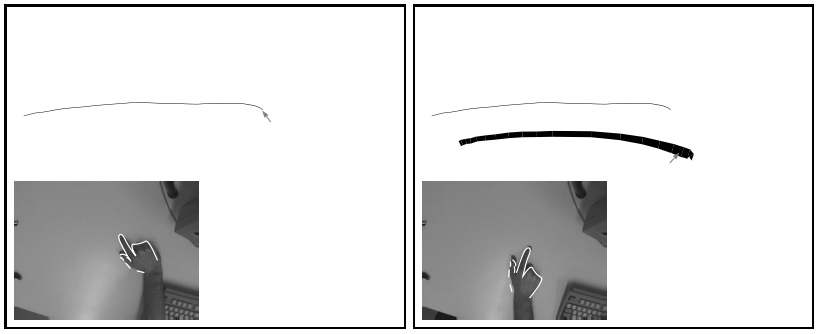


Fig. 7. Line thickness *is controlled using the orientation of the index finger. The top image shows a line drawn with the index finger pointing to the left, producing a thin trace. In the bottom image the finger pointed to the right and the line is fatter. Of course if the finger angle varies while the line is being drawn, a continuous variation of thickness is produced.*

posterior $p'(\mathbf{x})$. Some simple calculations give

$$\begin{aligned}
\mathcal{D} &= \left(\sum_{i=1}^{n} {\pi_i^{(n)}}^2\right)^{-1} && \text{by definition of } \mathcal{D}, \text{ equation (2)} \\
&\approx \left(\sum_{i=1}^{n} \tfrac{1}{n^2} p'(\mathbf{x}_i^{(n)})^2 / p(\mathbf{x}_i^{(n)})^2\right)^{-1} && \text{by defn of the } \pi_i, \text{ and comment below} \\
&\approx \left(\tfrac{1}{n} \int p'(\mathbf{x})^2 / p(\mathbf{x})\, d\mathbf{x}\right)^{-1} && \text{as the } \mathbf{x}_i^{(n)} \text{ are drawn from } p(\mathbf{x}) \\
&= \left(\int p'(\mathbf{x})^2 / p(\mathbf{x})\, d\mathbf{x}\right)^{-1} \times n \\
&= \alpha n
\end{aligned}$$

The second line uses the fact (see [15]) that for large n, the normalisation constant for the weights is approximately $1/n$ — so $\pi_i \approx p'(\mathbf{x}_i)/(np(\mathbf{x}_i))$.

References

1. S. Ahmad. A usable real-time 3D hand tracker. In *28th Asilomar Conference on Signals, Systems and Computers*. IEEE Computer Society Press, 1995. Available from `www.icsi.berkeley.edu/~ahmad`.
2. A. Blake and M.A. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Proc. Siggraph*, pages 185–192. ACM, 1994.
3. L. Brettzner and T. Lindeberg. Use your hand as a 3D mouse. In *Proc. 5th European Conf. Computer Vision*, volume 1, pages 141–157, Freiburg, Germany, 1998. Springer Verlag.
4. J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. Technical report, Dept. of Statistics, University of Oxford, 1997. Available from `www.stats.ox.ac.uk/~clifford/index.html`.
5. R. Cipolla and N.J. Hollinghurst. Human-robot interface by pointing with uncalibrated stereo vision. *J. Image and Vision Computing*, 14(3):171–178, 1996.

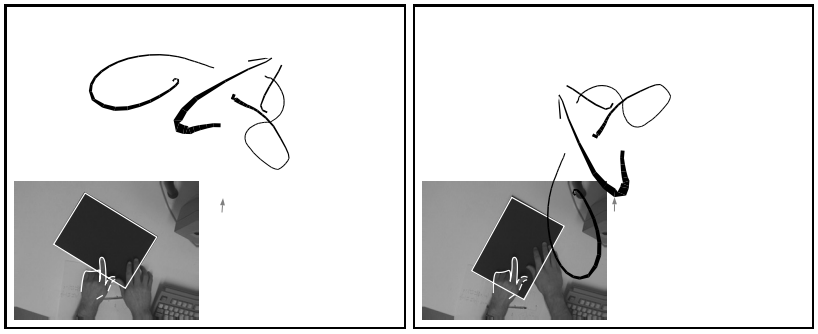


Fig. 8. Moving around the virtual workspace *is accomplished by following the tracked outline of a physical object. The piece of black paper can be tracked with a simple Kalman filter, and the virtual drawing follows the translations and rotations of the paper. The virtual workspace has been rotated anti-clockwise between the top and bottom frames. This is a very natural interface which can be used while drawing.*

6. A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR310, Dept. of Engineering, University of Cambridge, 1998. Available from `www.stats.bris.ac.uk:81/MCMC/pages/list.html`.
7. J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57:1317–1339, 1989.
8. H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. Multi-modal system for locating heads and faces. In *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 88–93, 1996.
9. T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. 6th Int. Conf. on Computer Vision*, 1998.
10. M. Isard and J. MacCormick. Hand tracking for vision-based drawing. Technical report, Visual Dynamics Group, Dept. Eng. Science, University of Oxford, 2000. Available from `www.robots.ox.ac.uk/~vdg`.
11. M.A. Isard and A. Blake. ICondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. 5th European Conf. Computer Vision*, pages 893–908, 1998.
12. R. Kjeldsen and J. Kender. Toward the use of gesture in traditional user interfaces. In *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 151–156, 1996.
13. Marcus Kohler and Sven Schröter. A Survey of Video-based Gesture Recognition: Stereo and Mono Systems. Technical Report 693, Informatik VII, University of Dortmund/Germany, August 1998. Available from `ls7-www.cs.uni-dortmund.de/~kohler`.
14. J. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93, 1998. In press. Available from `www-stat.stanford.edu/~jliu`.
15. J. MacCormick. *Probabilistic modelling and stochastic algorithms for visual localisation and tracking.* PhD thesis, University of Oxford, 2000. Available from `www.robots.ox.ac.uk/~jmac/research/thesis/`.
16. J. MacCormick and A. Blake. A probabilistic contour discriminant for object localisation. In *Proc. 6th Int. Conf. on Computer Vision*, pages 390–395, 1998.

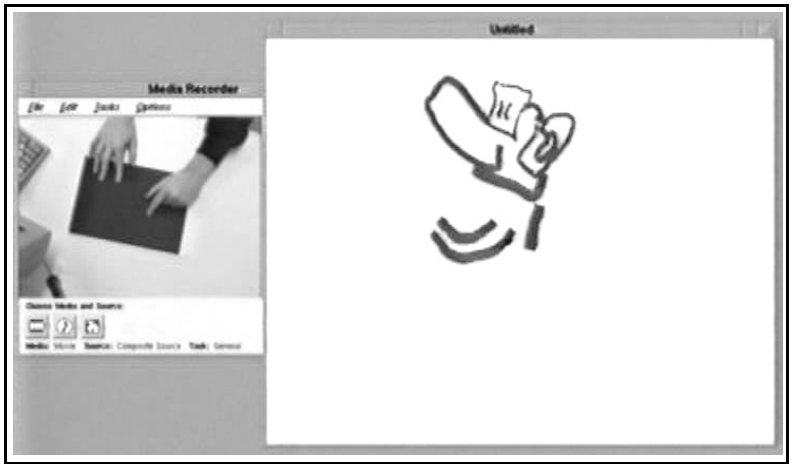


Fig. 9. The drawing package in action. *This screen shot shows a cartoon character drawn using the drawing package; note the variable-width lines. A movie clip showing this figure being created is available at [15]. The scene on the left is the camera's view; it is shown for information only and is not employed by the user.*

17. J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. 7th International Conf. Computer Vision*, pages 572–578, 1999.
18. D. Mackay. Introduction to Monte Carlo methods. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1999. Available as `ftp://wol.ra.phy.cam.ac.uk/pub/mackay/erice.ps.gz`.
19. B.D. Ripley. *Stochastic simulation*. New York: Wiley, 1987.
20. C. Wren and A. Pentland. Dynamic modeling of human motion. In *Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, April 1998.