Sequential Importance Sampling for Visual Tracking Reconsidered

Péter Torma
Mindmaker, Ltd.
Budapest 1121 HU
Konkoly Th. M. u. 29-33
tyus@mindmaker.hu

Csaba Szepesvári
Mindmaker, Ltd.
Budapest 1121 HU
Konkoly Th. M. u. 29-33
szepes@mindmaker.hu

Abstract

We consider the task of filtering dynamical systems observed in noise by means of sequential importance sampling when the proposal is restricted to the innovation components of the state. It is argued that the unmodified sequential importance sampling/resampling (SIR) algorithm may yield high variance estimates of the posterior in this case, resulting in poor performance when e.g. in visual tracking one tries to build a SIR algorithm on the top of the output of a color blob detector. A new method that associates the innovations sampled from the proposal and the particles in a separate computational step is proposed. The method is shown to outperform the unmodified SIR algorithm in a series of vision based object tracking experiments, both in terms of accuracy and robustness.

1 Introduction

Let us consider a stochastic dynamical system observed in noise. Assume that the state of the system ($x_t \in \mathbf{R}^n$) evolves in time according to a Markovian dynamics given by the transition kernel $p(x_t|x_{t-1})$, whilst the observations ($y_0, y_1, \dots \mathbf{R}^m$) are generated from an observation density of the form $p(y_t|x_t)$.[1] Further, let us assume that x_0 is distributed according to the prior $p(x_0)$.

In this paper we shall consider sequential importance sampling algorithms for estimating the posterior $p(x_t|y_{0:t})$ corresponding to the observation sequence $y_{0:t} = (y_0, \dots, y_t)$, where $t = 0, 1, 2, \dots$.

In particular we shall condider sequential importance sampling/resampling (SIR) algorithms (e.g. [2]). SIR algorithms work by keeping track of a finite number of hypotheses $x_t^{(1)}, \dots, x_t^{(N)}$ of the state along with some weights $w_t^{(1)}, \dots, w_t^{(N)}$ associated with them and providing a discrete approximation to the posterior $p(x_t|y_{0:t})$. Often, $x_t^{(1)}, \dots, x_t^{(N)}$ are called particles. In each time step, when a new observation arrives the states associated with the particles are updated first. This is done by drawing a proposed next state $\hat{x}_{t+1}^{(i)}$ from a so-called *proposal density* $\pi_{t,i}$: $\hat{x}_{t+1}^{(i)} \sim \pi_{t,i}(\cdot)$, $i = 1, \dots, N$. The choice of the proposal $\pi_{t,i}$ is left to the user. Obviously, the choice of $\pi_{t,i}$ effects the efficiency of the algorithm in a fundamental way. Often $\pi_{t,i}$ takes the form $\pi(\cdot|x_t^{(i)}, y_{t+1})$ where π is a fixed density function, i.e., the proposal depends only on the particle's previous state and the most recent observation. Two simple and popular choices are $\pi(\hat{x}_{t+1}|x_t^{(i)}, y_{t+1}) = p(\hat{x}_{t+1}|x_t^{(i)})$ (the dynamics) and $\pi(\hat{x}_{t+1}|x_t^{(i)}, y_{t+1}) = \pi'(\hat{x}_{t+1}|y_{t+1})$. Once the proposed states $\hat{x}_{t+1}^{(i)}$ ($i = 1, \dots, N$) are all computed, the weights associated with them are calculated according to the formulae:

$$\hat{w}_{t+1}^{(i)} \propto w_t^{(i)} \frac{p(y_{t+1}|\hat{x}_{t+1}^{(i)}) p(\hat{x}_{t+1}^{(i)}|x_t^{(i)})}{\pi_{t,i}(\hat{x}_{t+1}^{(i)})}, \qquad (1)$$

where the weights $\hat{w}_{t+1}^{(i)}$ are normalized such that $\sum_{i=1}^N \hat{w}_{t+1}^{(i)} = 1$. In the final step, the particle states and weights are either updated by equating them with the proposed states and weights, one-by-one (i.e., $x_{t+1}^{(i)} = \hat{x}_{t+1}^{(i)}$ and $w_{t+1}^{(i)} = \hat{w}_{t+1}^{(i)}$), or they are resampled from the weighted multi-set

[1] The function symbol p is overloaded as is usual in the literature: the particular density corresponding to a particular occurance of p is determined uniquely by the types of symbols that appear as the arguments of p in that instance of p. Therefore, in the case of $p(x_t|x_{t-1})$, p denotes the density that describes the time-evolution of states, whilst in the case of $p(y_t|x_t)$, p denotes the density that describes the dependence of observations on the states. The same applies to the symbol π to be introduced below.

$[(\hat{x}_{t+1}^{(1)}, \hat{w}_{t+1}^{(1)}), \ldots, (\hat{x}_{t+1}^{(N)}, \hat{w}_{t+1}^{(N)})]$ as follows:

$$j_{t+1}^{(i)} \sim [w_{t+1}^{(1)}, \ldots, w_{t+1}^{(N)}],\ x_{t+1}^{(i)} = \hat{x}_{t+1}^{(j_{t+1}^{(i)})}, w_{t+1}^{(i)} = \frac{1}{N}.$$

Afterwards, the whole process is started again. For the sake of simplicity in what follows we shall assume that resampling happens in each iteration.

The problem studied in this article arises when the proposal density and the dynamics have a special form and when the cost of evaluating the likelihood function is high. Such problems commonly arise in vision based tracking as we shall see it later. For now, let us consider our assumptions in more details.

Assumption A "Partitioned dynamical model." We shall assume that the state x_t is partitioned into two parts:

$$x_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix}, \qquad (2)$$

where $x_{1,t} \in \mathbf{R}^{n_1}$ and $x_{2,t} \in \mathbf{R}^{n_2}$ ($n_1 + n_2 = n$, $n_1, n_2 \geq 1$) and its evolution is given by

$$\begin{aligned} x_{1,t+1} &= f_1(x_t) + s_t, \\ x_{2,t+1} &= f_2(x_t), \end{aligned} \qquad (3)$$

where s_t is a martingal series: $E[s_t | s_{t-1}, s_{t-2}, \ldots] = 0$. According to Equation (3) the component $x_{1,t}$ evolves stochastically, while $x_{2,t}$ evolves in a deterministic way.

Systems of this type arise when a system's configuration at the next time step depends on a finite number of previous configurations. Hence, we shall call $x_{2,t}$ the *history* part of the state, whilst we call $x_{1,t}$ the *innovation* part of it. As a particular example of processes of this kind let us consider autoregressive (AR) processes. Remember that in the case of a k-dimensional order-p AR process the dynamics is given as follows: The state x_t is partitioned into p parts: $x_t = ((x_t)_0^T, \ldots, (x_t)_{p-1}^T)^T$, where $x_t \in \mathbf{R}^{kp}$ and $(x_t)_j \in \mathbf{R}^k$. Now, x_{t+1} is given by

$$\begin{aligned} (x_{t+1})_0 &= \sum_{j=0}^{p-1} A_j \cdot (x_t)_j + e_{t+1}, \quad \text{and} \\ (x_{t+1})_{j+1} &= (x_t)_j, \quad j = 0, \ldots, p-2. \end{aligned} \qquad (4)$$

Here $A_0, \ldots, A_{p-1} \in \mathbf{R}^{k \times k}$ are parameters of the process, and $e_0, e_1, \ldots$ is a series of independent, identically distributed zero-mean k-dimensional Gaussian random variables. The dynamics can be transformed into the form (3) by defining $n_1 = k$, $n_2 = k(p-1)$, $x_{1,t} = (x_t)_0$ and $x_{2,t} = ((x_t)_1^T, \ldots, (x_t)_{p-1}^T)^T$.

Assumption B "Restricted proposal." According to this assumption, the proposal π depends only on y_{t+1} and is defined only for the innovation component of the state. Therefore in what follows we shall write π in the form $\pi(x_{1,t+1}|y_{t+1})$.

Assumption C "The cost associated with evaluating the observation density is high."

In order to simplify the exposition we shall further assume the following:

Assumption D "The observation density depends only on $x_{1,t}$, the innovation part of the state." According to this assumption one can write $p(y_{t+1}|x_{t+1}) = p(y_{t+1}|x_{1,t+1})$.

Assumptions A, B, C, and D are often satisfied when particle filters are used in visual tracking. First, the dynamics of the object to be tracked is often represented by some AR process (satisfying Assumption A). It is also quite common that Assumptions B and D are satisfied. One example is given when a color blob detector is used to produce the proposal density (e.g. see [3]). Note that in the case of visual tracking, according to Assumption B the proposed states will depend only on information derived using the images (a "bottom up" approach). Finally, it is also reasonable to presume Assumption C since in visual tracking the evaluation of the observation density involves image processing steps and these are generally computationally very expensive operations.

Under Assumptions A, B, C, and D algorithm SIR takes the form presented in Table 1. In what follows we shall call this algorithm "Basic-SIR". In order to spare some space when descibing the algorithms we will always omit their initialization phase: we assume that initialization is always done by drawing samples from the true prior $p(x_0)$. Under this condition, the particle set $x_t^{(i)}$ updated using Basic-SIR can be shown to represent an unbiased estimate of the posterior, i.e., for any measurable function h defined over the state space, $E\left[\frac{1}{N}\sum_{i=1}^N h(x_t^{(i)}) \,\middle|\, y_{0:t}\right] = E\left[h(x_t) \,|\, y_{0:t}\right]$.[2]

Unfortunately, Basic-SIR can be very inefficient and may require a large number of particles to achieve even a modest precision. This is because in Step 2 of the algorithm, many weights can get pretty small at the same time, since in Step 1 the innovation $u_{t+1}^{(i)}$ that will be associated with particle i at time $t+1$ is sampled *independently* of the state ($x_t^{(i)}$) associated with that particle. Therefore, with high probability, the value of $p(\hat{x}_{t+1}^{(i)}|x_t^{(i)})$ will be small when e.g. the density $p(x_{t+1}^{(i)}|x_t^{(i)})$ is concentrated to a small portion of the state space. This happens e.g. when the variance

[2] Under certain ergodicity assumptions, one can also show that the estimated posterior decouples from the prior at a geometric rate [5]. However, these results are outside of the scope of the present article.

of the system noise s_t (cf. Equation (3)) is small. The problem is illustrated in Figure 1 below.

1. Sample $u_{t+1}^{(i)} \sim \pi(u|y_{t+1})$

   and let $\hat{x}_{t+1}^{(i)} = \begin{pmatrix} u_{t+1}^{(i)} \\ f_2(x_t^{(i)}) \end{pmatrix}$.
2. Let $w_{t+1}^{(i)} \propto \frac{p(y_{t+1}|\hat{x}_{t+1}^{(i)})p(\hat{x}_{t+1}^{(i)}|x_t^{(i)})}{\pi(u_{t+1}^{(i)}|y_{t+1})}$.
3. Sample $j_{t+1}^{(i)} \sim [w_{t+1}^{(1)}, \dots, w_{t+1}^{(N)}]$.
4. Let $x_{t+1}^{(i)} = \hat{x}_{t+1}^{(j_{t+1}^{(i)})}$.

Table 1: *Main loop of Basic-SIR. N is the number of particles and $i = 1, 2, \dots, N$ is a particle index. Note that $u_{t+1}^{(i)}$ corresponds exactly to the innovation component of the state.*

The inefficiency of particle filter methods have been observed by many authors. The typical sugestion is to replace the proposal distribution by a "better" one. For example, Pitt and Shephard proposed the so-called auxiliary variable method whereas they suggest to use a proposal distribution that is an appropriately defined mixture that depends both on the past state and the most recent observation [6]. Although their method overcomes the problem of weight degeneracy in most of the cases, it involves $R \gg N$ evaluation steps of the observation density and thus, under Assumption C, the computational burden of this algorithm can be pretty high. More recently, van der Merwe et. al proposed to use a bank of unscented Kalman filters to define the proposal distribution [9]. This algorithm, called the unscented particle filter, is similar in essence to the auxiliary variable method but avoids the expensive likelihood calculations.

Note that the computational example used by Isard and Blake to illustrate their ICondensation algorithm [3] satisfies both Assumptions A, B, C, and D. We will discuss ICondensation in the last section.

2 Algorithms

The idea of the algorithms we consider is to ensure that for each particle the history component of the particle will match the innovation component sampled from the proposal. We achieve this by drawing an appropriate history for each innovation component.

The main loop of our first algorithm called HS-SIR (SIR with History Sampling) is shown in Table 2.

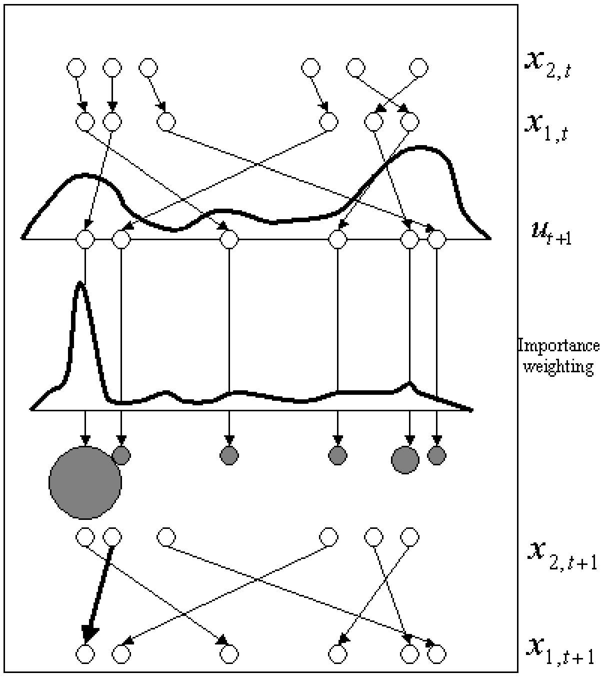

Figure 1: *Illustration of the behavior of Basic-SIR.* Consider a system where the state $x_t = (x_{1,t}, x_{2,t})$ evolves according to a one-dimensional first-order AR-model. The first two rows of the figure represent the particle set at time t, where the individual particles are identified by the arrows connecting $x_{2,t}$ and $x_{1,t}$. The next row shows the proposal density and the innovations $(u_{t+1}^{(i)})$ drawn from it. The arrows from $x_{1,t}$ to u_{t+1} show the association of the randomly sampled innovations and the particles. In the lower part of the figure the new particle set is depicted just before the resampling step. Weights of the individual particles are represented by the strength of the respective arrows.

In order to understand this algorithm, let us introduce the auxiliary variables $(x_t^{(i,j)}, w_t^{(i,j)})$ such that $(x_t^{(i,j)}, w_t^{(i,j)}) = (x_t^{(i)}, w_t^{(i)})$ and let a particle set at time $t+1$ be defined by the equations

$$\begin{aligned} x_{t+1}^{(i,j)} &= \begin{pmatrix} u_{t+1}^{(j)} \\ f_2(x_t^{(i)}) \end{pmatrix} \quad \text{and} \\ w_{t+1}^{(i,j)} &= w_t^{(i,j)} \frac{p(y_{t+1}|x_{t+1}^{(i,j)})p(x_{t+1}^{(i,j)}|x_t^{(i,j)})}{\pi(u_{t+1}^{(j)}|y_{t+1})} \\ &= w_t^{(i)} \frac{p(y_{t+1}|u_{t+1}^{(j)})p(u_{t+1}^{(j)}|x_t^{(i)})}{\pi(u_{t+1}^{(j)}|y_{t+1})}. \end{aligned}$$

Here the last equation follows by our assumptions on the observation and proposal densities. Now assume that at time t the particle set $(x_t^{(i)}, w_t^{(i)})_{i=1}^N$ represents an unbiased estimate of the posterior $p(x_t|y_{0:t})$.

```
1.  Sample $u_{t+1}^{(i)} \sim \pi(u_{t+1}|y_{t+1})$
2.  Sample $J_{t+1}^{(i)} \sim$
        $[\ldots, \frac{p(y_{t+1}|u_{t+1}^{(\cdot)})}{\pi(u_{t+1}^{(\cdot)}|y_{t+1})} \sum_{k=1}^{N} w_t^{(k)} p(u_{t+1}^{(\cdot)}|x_t^{(k)}), \ldots]$
3.  Sample $I_{t+1}^{(i)} \sim [\ldots, w_t^{(\cdot)} p(u_{t+1}^{(J_{t+1}^{(i)})}|x_t^{(\cdot)}), \ldots]$.
4.  Let $x_{t+1}^{(i)} = \left( \begin{array}{c} u_{t+1}^{(J_{t+1}^{(i)})} \\ f_2\left(x_t^{(I_{t+1}^{(i)})}\right) \end{array} \right)$.
```

Table 2: *Main loop of HS-SIR (SIR with history sampling).*

Clearly, by the unbiasedness of the basic importance sampling scheme, the particle set $(x_{t+1}^{(i,j)}, w_{t+1}^{(i,j)})_{i,j=1}^{N}$ will represent an unbiased estimate of the posterior $p(x_{t+1}|y_{0:t+1})$ (cf. Steps 1–2 of the basic SIR algorithm of Table 1). Now, if $I_{t+1}^{(i)}$ and $J_{t+1}^{(i)}$ are the random indexes drawn as in Step 2 and 3 of HS-SIR, then

$$P(I_{t+1}^{(i)} = k, J_{t+1}^{(i)} = l) = P(I_{t+1}^{(i)} = k|J_{t+1}^{(i)} = l) \cdot$$
$$P(J_{t+1}^{(i)} = l) = \frac{w_t^{(k,l)} p(u_{t+1}^{(l)}|x_t^{(k)})}{\sum_{l=1}^{N} w_t^{(k,l)} p(u_{t+1}^{(l)}|x_t^{(k)})} \cdot$$
$$\frac{\sum_{l=1}^{N} w_t^{(k,l)} p(u_{t+1}^{(l)}|x_t^{(k)})}{\sum_{k=1}^{N}\sum_{l=1}^{N} w_t^{(k,l)} p(u_{t+1}^{(l)}|x_t^{(k)})} \propto w_{t+1}^{(k,l)}.$$

Therefore Steps 2 and 3 of HS-SIR take the form of a standard resampling step for the particle set $(x_{t+1}^{(i,j)}, w_{t+1}^{(i,j)})$, and therefore the particle set $(x_{t+1}^{(i)}, 1/N)_{i=1}^{N}$ will represent an unbiased estimate of the posterior. Actually, Steps 2 and 3 of the above algorithm can be considered as sampling from $(\ldots, w_{t+1}^{(\cdot,\cdot)}, \ldots)$ by means of partitioned sampling [4]: Step 2 samples the innovation components, whilst Step 3 samples the appropriate histories to be associated with these components.

The advantage of HS-SIR over Basic-SIR should be clear by now: HS-SIR selects (by random sampling) pairs of innovations and histories that have high probability of co-occurring and thereby it will in general reduce the variance of the estimate of the posterior. Convergence theorems similar to those of [1] can be derived but are omitted due to lack of space.

Our next algorithm can be considered as a Rao-Blackwellised version of the previous one, whereas sampling of the innovation component indeces ($J_{t+1}^{(i)}$) is avoided - causing a further reduction in the variance of the estimate of the posterior. The algorithm, that we call RB-HS-SIR (Rao-Blackwellised SIR with history sampling) is shown in Table 3, whilst Figure 2 illustrates the algorithm's working principles. Again, one expects that during the course of the algorithm the effective sample size will stay high as the algorithm will prefer (on the average) highly probable history-innovation associations.

```
1.  Sample $u_{t+1}^{(i)} \sim \pi(u_{t+1}|y_{t+1})$
2.  Sample $I_{t+1}^{(i)} \sim [\ldots, w_t^{(\cdot)} p(u_{t+1}^{(i)}|x_t^{(\cdot)}), \ldots]$.
3.  Let $x_{t+1}^{(i)} = \left( \begin{array}{c} u_{t+1}^{(i)} \\ f_2\left(x_t^{(I_{t+1}^{(i)})}\right) \end{array} \right)$.
4.  Let $w_{t+1}^{(i)} = \frac{p(y_{t+1}|u_{t+1}^{(i)}) \sum_{l=1}^{N} w_t^{(l)} p(u_{t+1}^{(i)}|x_t^{(l)})}{\pi(u_{t+1}^{(i)}|y_{t+1})}$.
```

Table 3: *Main loop of RB-HS-SIR (SIR with Rao-Blackwellised history sampling).* Note that in this algorithm resampling occurs in Step 3 and not in the last step of the algorithm.

In order to show the unbiasedness of this algorithm we evaluate $R = E[\sum_{i=1}^{N} w_{t+1}^{(i)} h(x_{t+1}^{(i)})|y_{0:t+1}]$ directly:

$$R = E\left[\sum_{i=1}^{N} w_{t+1}^{(i)} h(x_{t+1}^{(I_{t+1}^{(i)},i)}) \,\Big|\, y_{0:t+1}\right]$$
$$= \sum_{i=1,l=1}^{N} E\Bigg[P(I_{t+1}^{(i)} = l|y_{0:t+1}, w_t^{(\cdot)}, x_t^{(\cdot)}, u_{t+1}^{(\cdot)}) \cdot$$
$$E\big[w_{t+1}^{(i)} h(x_{t+1}^{(I_{t+1}^{(i)},i)}) \,|\, y_{0:t+1},$$
$$I_{t+1}^{(i)} = l, w_t^{(\cdot)}, x_t^{(\cdot)}, u_{t+1}^{(\cdot)}\big] \,|\, y_{0:t+1} \Bigg]$$

Now, since

$$P(I_{t+1}^{(i)} = l|y_{0:t+1}, w_t^{(\cdot)}, x_t^{(\cdot)}, u_{t+1}^{(\cdot)}) =$$
$$= \frac{w_t^{(l)} p(u_{t+1}^{(i)}|x_t^{(l)})}{\sum_{r=1}^{N} w_t^{(r)} p(u_{t+1}^{(i)}|x_t^{(r)})},$$

by the definition of $w_{t+1}^{(i)}$ one gets that

$$R \quad = \sum_{l=1,i=1}^{N} E\Bigg[\frac{w_t^{(l)} p(u_{t+1}^{(i)}|x_t^{(l)})}{\sum_{r=1}^{N} w_t^{(r)} p(u_{t+1}^{(i)}|x_t^{(r)})} \cdot$$
$$\frac{p(y_{t+1}|u_{t+1}^{(i)}) \sum_{k=1}^{N} w_t^{(k)} p(u_{t+1}^{(i)}|x_t^{(k)})}{\pi(u_{t+1}^{(i)}|y_{t+1})} \,.$$

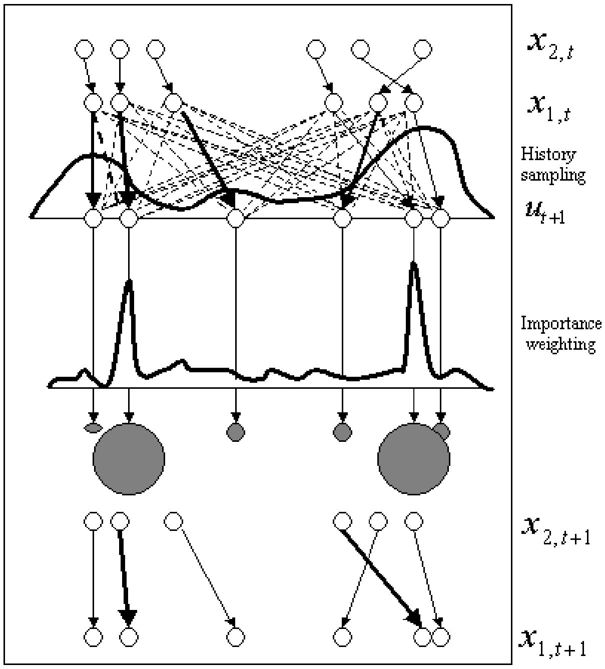

Figure 2: *Illustration of the behavior of the RB-HS-SIR.* The figure is almost identical to Figure 1, except that now the arrows from $x_{1,t}$ to u_{t+1} show all the possible associations of innovations and particles, and the strength of these arrows are proportional to the weights that are used in associating particles histories and innovation components.

$$h(x^{(l,i)}_{t+1}) \Big| y_{0:t+1} \Bigg] = \sum_{l=1,i=1}^{N} E\Bigg[w_t^{(l)} p(u_{t+1}^{(i)}|x_t^{(l)}) \cdot$$

$$\frac{p(y_{t+1}|u_{t+1}^{(i)})}{\pi(u_{t+1}^{(i)}|y_{t+1})} \cdot h(x^{(l,i)}_{t+1}) \Big| y_{0:t+1} \Bigg]$$

$$= \sum_{l=1,i=1}^{N} E\Bigg[w^{(l,i)}_{t+1} h(x^{(l,i)}_{t+1}) \Big| y_{0:t+1} \Bigg],$$

which finishes the proof of unbiasedness, since we have already seen that the particle set $(x^{(p,q)}_{t+1}, w^{(p,q)}_{t+1})$ gives an unbiased representation of the posterior.[3]

Other variants of the algorithm can also be given. As an example let us mention the variant when the particle's innovation components are resampled, whilst their history components are retained. This variant can be advantageous if the particle set bears more information about the posterior than the proposal function.

Finally, let us mention the practical variant when the proposal function is further restricted to a few selected components of the innovation component. For example in visual tracking, often the configuration is composed of translational and other components (e.g. rotation, scale) and the proposal depends only on the translational component (this is the case when color blob detection is used to define the proposal [3]). For this case we propose a variant of RB-HS-SIR that we shall call RB-SS-HS-SIR (Rao-Blackwellised subspace SIR with history sampling). The main loop of this algorithm is given in Table 4. We will still use the symbol u_{t+1} for the component sampled from the proposal, while v_{t+1} is used to denote the remaining components of the innovation. In order to see the unbiasedness of the algorithm note that

```
1. Sample $u_{t+1}^{(i)} \sim \pi(u_{t+1}|y_{t+1})$
2. Sample $I_{t+1}^{(i)} \sim [\ldots, w_t^{(\cdot)} p(u_{t+1}^{(i)}|x_t^{(\cdot)}), \ldots]$.
3. Draw $v_{t+1}^{(i)}$ from $p(v_{t+1}^{(i)}|u_{t+1}^{(i)}, x_t^{(I_{t+1}^{(i)})})$.
4. Let $x_{t+1}^{(i)} = \begin{pmatrix} u_{t+1}^{(i)} \\ v_{t+1}^{(i)} \\ f_2\left(x_t^{(I_{t+1}^{(i)})}\right) \end{pmatrix}$.
4. Let $w_{t+1}^{(i)} = \frac{p(y_{t+1}|u_{t+1}^{(i)}) \sum_{l=1}^{N} w_t^{(l)} p(u_{t+1}^{(i)}|x_t^{(l)})}{\pi(u_{t+1}^{(i)}|y_{t+1})}$.
```

Table 4: *Main loop of RB-SS-HS-SIR (Rao-Blackwellised subspace SIR with history sampling).*

$$p(x_{1,t+1}^{(i)}|x_t^{(I_{t+1}^{(i)})}) = p(u_{t+1}^{(i)}, v_{t+1}^{(i)}|x_t^{(I_{t+1}^{(i)})}) =$$

$$= p(v_{t+1}^{(i)}|u_{t+1}^{(i)}, x_t^{(I_{t+1}^{(i)})}) p(u_{t+1}^{(i)}|x_t^{(I_{t+1}^{(i)})}).$$

Here the first term of the last line, $p(v_{t+1}^{(i)}|u_{t+1}^{(i)}, x_t^{(I_{t+1}^{(i)})})$ gives the posterior of $v_{t+1}^{(i)}$ given $u_{t+1}^{(i)}$ and $x_t^{(I_{t+1}^{(i)})}$. Since this is the term used to sample $v_{t+1}^{(i)}$ in Step 3, it cancels out when computing the importance weights. Hence follows the unbiasedness of the RB-SS-HS-SIR.

Sampling from $p(v_{t+1}|u_{t+1}, x_t)$ is not necessarily straightforward. Two exceptions are when the system noise is Gaussian (in this case $p(v_{t+1}|u_{t+1}, x_t)$ will still be Gaussian) and when v_{t+1} and u_{t+1} are independent given x_t and $p(v_{t+1}|x_t)$ assumes a form that is easy to sample from (in this case $p(v_{t+1}|u_{t+1}, x_t) = p(v_{t+1}|x_t)$ and hence sampling from $p(v_{t+1}|u_{t+1}, x_t)$ reduces to sampling from $p(v_{t+1}|x_t)$).

[3] Again, under appropriate conditions, bounds on the variance of the estimate of the posterior can be derived along the lines of [1].

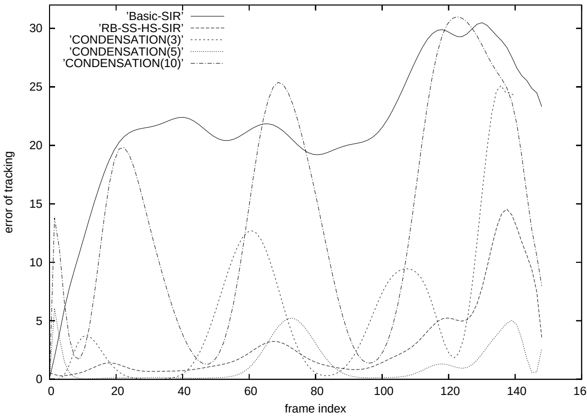

Figure 3: *Tracking error as a function of frame number.* The individual curves were smoothed by plotting them using gnuplot's "bezier" style.

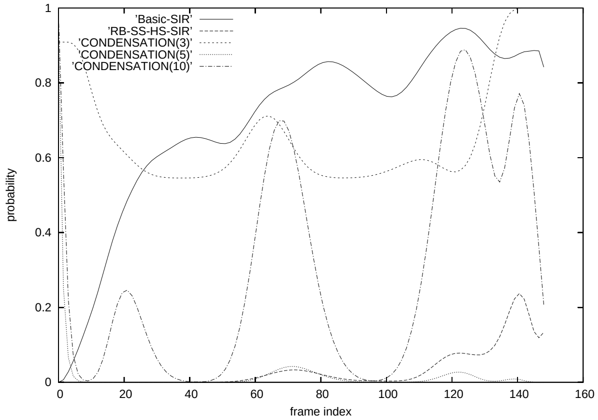

Figure 4: *Probability of losing the object.* The individual curves were smoothed by plotting them using gnuplot's "bezier" style.

3 Experiments

In order to study the efficiency of the new algorithms we carried out some vision based tracking experiments. The results of these experiments are presented here.

We considered tracking an artificial object moving in front of a camera in a normal office room environment (see Figure 5). The object is represented by its contour (using a spline based representation) and the task was to estimate the object's configuration (translation, rotation, scale) on each frame. The resolution of the images was set to 240 × 180. The dynamical model was a second-order four-dimensional AR process (cf. Equation (4)). The observation density is computed by matching the contour to the image. Details of these computations can be found in [7].

The output of a Gaussian color blob detector working on the original frames was used as the basis of the proposal, just like in [3]. First, the output of the blob detector was down-sampled to a resolution of 24 × 18 pixels. Then spatial coordinates were drawn from the appropriately re-scaled output of the blob detector. These coordinates were then mapped back to the original coordinate system of the images. The final coordinates were obtained by applying a random perturbation to the coordinates calculated so far, by adding a random "fine-scale" random displacement vector drawn uniformly from the set $\{-5, -4, \ldots, 5\} \times \{-5, -4, \ldots, 5\}$. Note that another object with color and shape identical to that of the object to be tracked was lying on the table. As a consequence, the proposal keeps to draw "fake" positions that need to be "filtered out", making the job of the blob-detector based trackers more difficult.

For the sake of comparisons CONDENSATION (also known as N-IPS[4]), Basic-SIR, and RB-SS-HS-SIR were implemented and tried on a number of image sequences. A typical tracking scenario using RB-SS-HS-SIR is presented in Figure 5. In one sequence of 5 seconds of video sampled at 30Hz the configurations of the object to be tracked were determined by careful visual inspection (this is the sequence shown on Figure 5). Each algorithm was then tested on this sequence with 100 different random seeds. Note that since the accuracy of CONDENSATION depends heavily on the variance of the system noise e_t (cf. Equation (4)), CONDENSATION was tested with a number of dynamics, where the variance of e_t was varied. The three values used as the variance of e_t were $3, 5$ and 10. Accordingly, we write CONDENSATION(3), CONDENSATION(5) and CONDENSATION(10) to denote the respective algorithms. Tracking error and the probability of losing the object to be tracked were measured as a function of frame number. Equivalent running time experiments were considered on an Intel Pentium IV 1.4GHz computer with 128MB RAM, i.e., the particle sizes were set so that the running time of the various algorithms were the same (for CONDENSATION we used CONDENSATION(5) to set the number of particles - it was CONDENSATION(5) that gave the best results in the experiments). No attempt was made to do any serious optimizations on any of the algorithms. Respective particle set sizes are given in Table 5. Tracking errors of the various algorithms averaged over the 100 runs are shown in Figure 3. The error is computed as the distance (in pixels) in between the estimated position and the true position of the object to be tracked. It should be clear from

[4] This algorithm is obtained as a special case of the SIR algorithm described in Section 1 when the proposal is chosen to be the dynamics of the system.

Algorithm	**# Particles**
CONDENSATION	2000
Basic-SIR	3000
RB-SS-HS-SIR	400

Table 5: *Particle sizes used in the experiments.*

the figure that in terms of the errors: RB-SS-HS-SIR $\approx$ CONDENSATION(5) $\ll$ CONDENSATION(3) $\ll$ CONDENSATION(10) $\ll$ Basic-SIR, so that RB-SS-HS-SIR and CONDENSATION(5) are the best. The probability of losing the object as a function of frame indeces is shown in Figure 4. This probability is estimated by computing the fraction of cases (of the 100 runs) when the output of the tracker is outside of a certain large neighborhood of the object to be tracked (50 pixels).[5] Again, the ordering of the algorithms remains the same: RB-SS-HS-SIR and CONDENSATION(5) perform the best, whilst Basic-SIR performs the worst.[6]

4 Discussion

These experiments indicate that under a wide range of conditions the new algorithms do indeed overcome the inefficiency of Basic-SIR. Although the results are encouraging, one should bear in mind that the new algorithms are computationally more expensive than the original Basic-SIR algorithm: now one iteration requires $O(N^2)$ evaluations of the density $p(x_{t+1}|x_t)$.[7] Fortunately, however, the number of times the observation density needs to be evaluated still scales linearly with the number of particles. Therefore the new algorithms can be cheaper than Basic-SIR or CONDENSATION when the cost of evaluating the observation density for a larger number of particles is higher than the cost of evaluating the density $p(x_{t+1}|x_t)$ $O(N^2)$ times.[8] Note also that the experiments also revealed that performance of CONDENSATION depends heavily on the accuracy of the estimate of the variance of the dynamics. When the estimated variance is too low or too high, the performance of CONDENSATION degrades rapidly. Additional experiments (not shown here) have shown that the performance degredation of RB-SS-HS-SIR is much less pronounced. Another advantagous property of the new algorithms, not shared by CONDENSATION, is the ability of these algorithms to recover from gross tracking errors (loss of object lock) - thanks to the independence of the proposal from the process state.

What remains is the discussion of the relation of the algorithms to ICondensation, the algorithm introduced by Isard and Blake in [3]. At a first glance ICondensation looks very similar to Basic-SIR. However, let us take a closer look at this algorithm. In Figure 1 of [3] in Step 2(a) the next state is sampled from the proposal as normal. However, importance weights are calculated with the formula used in RB-HS-SIR (see Step 2(b) and 3 of [3])[9] - possibly causing serious performance deterioration.[10] Note that if all the particles are concentrated into a relatively small portion of the state space then the importance weights calculated as in RB-HS-SIR will be close to the “correct” ones (cf. Equation (1)). The same applies when the dynamics is close to the uniform distribution. Also, note that ICondensation as described in [3] mixes several algorithms: re-initialization, CONDENSATION and Basic-SIR with the modification described above. This “mix” can make ICondensation work under a wide range of conditions. However, this is exactly the reason why we have chosen to compare our new algorithms with the building blocks of ICondensation (Basic-SIR and CONDENSATION) instead of comparing them with it directly.

5 Conclusions

Motivated by problems that arise when particle filters are applied for visual tracking, we considered sequential importance sampling algorithms under the conditions that the proposal density is defined only for the innovation part of the state space and depends only on the last observation. We have argued that the unmodified SIR algorithm can be very inefficient in this case. Several new algorithms associating particles and innovations in a separate computational step were proposed. The new algorithms were shown to yield unbiased estimates of the posterior and, by means of some computer experiments a member of this family was shown to yield performance superior than that of

[5] In order to separate the effect of losing the object from problems with accuracy when the object is tracked, when the object is lost at a certain point in time, the corresponding distances are not included in the computation of the average error.

[6] According to the graphs CONDENSATION(5) looks to perform slightly better then RB-SS-HS-SIR at the end of the tracking sequence. However, since the variance of these performance figures (not shown) is high, no significant difference can be concluded. Of course, these comparisions can only be indicative since the actual performance figures always depend on the actual implementations.

[7] Note that this is also the case for ICondensation [3].

[8] If the system noise e_t is Gaussian, a substantial speedup of the current implementation can be realized when e.g. a lookup-table is used to evaluate the underlying Gaussian function.

[9] The same problem appears when they describe the details of the algorithm in Section 4.2.

[10] Note that “incorrect” weights do not necessarily cause a problem, see e.g. Theorem 3.1 of [8].

Basic-SIR. Further, this algorithm was shown to perform at least as well as CONDENSATION, but with fewer number of particles. Further, we have argued that the new algorithms are generally more robust than CONDENSATION, i.e., these algorithms recover faster when the object is lost. One particularly interesting avenue for future research would be to combine CONDENSATION, LS-N-IPS (see [7]) and the algorithms proposed here. With a clever combination that adds up the advantages of these algorithms one hopes be able to create an algorithm that outperforms all the previous ones and under a wide range of conditions.

6 Acknowledgements

We are grateful to the referees for their useful comments and suggestions.

References

[1] D. Crisan and A. Doucet. Convergence of sequential monte carlo methods, 2000.

[2] Arnaud Doucet. On sequential simulation based methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 1998.

[3] Michael Isard and Andrew Blake. ICondensation: Unifying low-level and high-level tracking in a stochastic framework. *Proc 5th European Conf. Computer Vision*, 1998.

[4] John MacCormick. *Probabilistic Modelling and Stochastic Algorithms for Visual Localisation and Tracking.* PhD thesis, University of Oxford, 2000.

[5] Pierre Del Moral. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincar*, 37(2):155–194, 2001.

[6] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filter. *Journal of the American Statistical Association*, 94:590–9, 1999.

[7] Csaba Szepesvári and Péter Torma. Vision based object tracking using LS-N-IPS. *In Proc. ISAP'01*, pages 277–282, 2001.

[8] Péter Torma and Csaba Szepesvári. LS-N-IPS: an improvement of particle filters by means of local search. *In Proc. Non-Linear Control Systems(NOLCOS'01)*, 2001.

[9] Rudolph van der Merwe, Nando de Freitas, Arnaud Doucet, and Eric Wan. The unscented particel filter. In *Advances in Neural Information Processing Systems 13*, Nov 2001.

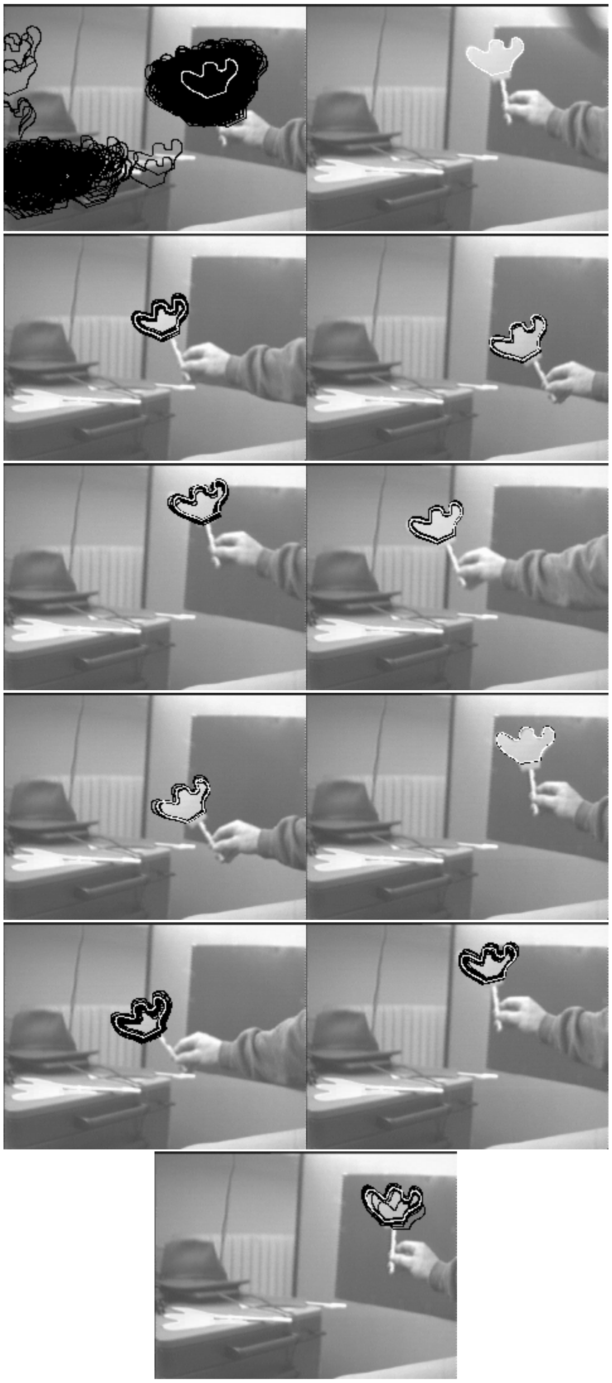

Figure 5: *A typical tracking sequence with RB-SS-HS-SIR and* 600*-particles.* Black contours show configurations with high probabilities, while the white contour represents the expected configuration. Note that there is an object lying on the table that has the same characteristics (e.g. is made of the same material of the same color) as the object to be tracked, making the tracking task more difficult.