

# Enhancing Particle Filters using Local Likelihood Sampling

Péter Torma<sup>1</sup> and Csaba Szepesvári<sup>2</sup>

<sup>1</sup> Tateyama Ltd., Zenta u. 1.  
1111 Budapest, Hungary  
tyus@tateyama.hu

<sup>2</sup> Computer Automation Institute of the Hungarian Academy of Sciences,  
Kende u. 13-17, 1111, Budapest, Hungary  
Csaba.Szepesvari@sztaki.hu; <http://www.sztaki.hu/szepesvari>

**Abstract.** Particle filters provide a means to track the state of an object even when the dynamics and the observations are non-linear/non-Gaussian. However, they can be very inefficient when the observation noise is low as compared to the system noise, as it is often the case in visual tracking applications. In this paper we propose a new two-stage sampling procedure to boost the performance of particle filters under this condition. We provide conditions under which the new procedure is proven to reduce the variance of the weights. Synthetic and real-world visual tracking experiments are used to confirm the validity of the theoretical analysis.

## 1 Introduction

In this paper we consider particle filters in the special case when the observation noise is low as compared to the noise in the system's dynamics (for brevity we call the latter noise the 'system noise'). This is a typical situation in computer vision where the discrimination power of the object model is typically high. Such models may e.g. use shape, contour, colour, intensity information or a combination of these and give rise to a highly peaked, low entropy observation likelihood function. Highly discriminative observation likelihoods are very desirable as they result in highly peaked posteriors and hence, in theory, the position of the object can then be estimated with high precision.

In practice the posterior cannot be obtained in a closed form, except in a few special cases. Thus in general one must revert to some approximate method to estimate the posterior. Particle filters represent a rich class of such approximate methods. They represent the posterior using a weighted particle set living in the state space of the process. Upon the receipt of a new observation the generic particle filter algorithm updates the position of the particles and recomputes the weights so that the new weighted sample becomes a good representation of the new posterior that takes into account the new observation, as well. The position of the particles are typically updated independently of each other by drawing them from a user-chosen proposal distribution. If the new particles are not in the

close vicinity of the modes of the likelihood function and the likelihood function is highly peaked then the representation of the posterior will degrade very fast. The peakier the observation likelihood function the more important is to sample the particles such that they will be close to the modes.

This paper introduces a method that draws the new positions of the particles using a two-stage sampling process that depends on the likelihood function and hence one expects the new method to have superior performance than that of those algorithms that do not use the likelihood function. At the core of the new filter is sampling method that is applicable when the density to sample from has a product form, with one of the terms being highly peaked, whilst the other having heavy tails. Under this conditions the new sampling method represents an alternative to importance sampling. The new method works by first drawing a particle from the broader density. In the second step this particle is perturbed such that on average it moves closer to one of the modes of the peaky term. Weights are calculated so that unbiasedness of the new sample is guaranteed. We compare the expected performance of the proposed scheme by means of a theoretical analysis to that of the basic importance sampling scheme and derive conditions under which the new scheme can be expected to perform better. The comparison is extended to the particle filtering setting, as well. The theoretical findings are confirmed in some computer experiments. In particular, the method is applied to the tracking of Japanese license plates where the new algorithm is shown to improve performance substantially both in terms of tracking accuracy and speed.

### 1.1 Related Work

The efficiency problem associated with low observation noise is well known in the literature and hence many approaches exist to resolving it. Among the many methods the Auxiliary variable Sampling Importance Resampling (ASIR) filter introduced by Pitt and Shephard [1] is one of the closest to our algorithm. ASIR approximates the proposal density using a mixture of the form  $\sum_{k=1}^N \beta_k \gamma_k(\cdot)$  where weight  $\beta_k$  approximates the normalized likelihood of the new observation ( $Y_t$ ) assuming that the state of the process at the  $t - 1$ th time step is  $X_{t-1}^{(k)}$ . The function  $\gamma_k(\cdot)$  approximates the density  $p(x_t | X_{t-1}^{(k)}, Y_1, \dots, Y_t)$ .<sup>3</sup> The biggest obstacle in applying ASIR is to obtain a good approximation of the conditional likelihood of the new observation given  $X_{t-1}^{(k)}$ , since this involves the evaluation of a potentially high dimensional integral and the efficiency of ASIR ultimately depends on the quality of this approximation. It turns out that except for some special cases it is not easy to come up with a sampling scheme that could sample from  $p(x_t | X_{t-1}^{(k)}, Y_1, \dots, Y_t) \propto p(Y_t | x_t) p(x_t | X_{t-1}^{(k)})$  in an efficient manner. This issue is the problem we are addressing in this paper.

Another recent proposal is called ‘likelihood sampling’ (see e.g. [2]). In this approach it is the likelihood function  $p(Y_t | \cdot)$  that is used to generate the samples from and the prediction density is used to calculate the weights. The success

<sup>3</sup> Here and in what follows random variables are denoted by big capitals.

of this method depends on the amount of ‘state aliasing’ that comes from the limitations of the observation model. For multi-modal observation likelihoods a large number of particles can be generated that will have low weights when the posterior already concentrates on a small part of the state space. Our method overcomes this problem by first sampling from the prediction density. As a result, our method can suffer from inefficiency if the estimates posterior is considerably off from the target so we expect our method being competitive only when this is not the case.

Perhaps the most relevant to this work is the LS-N-IPS algorithm introduced in [3]. In LS-N-IPS the prediction density is used to derive the new particle set which is then locally modified by climbing the observation likelihood. Hence this algorithm introduces some bias and also needs an efficient method to climb the observation likelihood. The method proposed here resolves the inefficiency problem without introducing any additional bias or requiring a hill-climbing method.

## 2 Notation

The following notations will be used: for an integrable function  $f$ ,  $I(f)$  will denote the integral of  $f$  with respect to the Lebesgue measure.<sup>4</sup>  $\mathbb{R}^d$  denotes the  $d$ -dimensional Euclidean space.  $L^p$  ( $0 < p \leq +\infty$ ) denotes the set of functions with finite  $p$ -norm. The  $p$ -norm of a function is denoted by  $\|f\|_p$ . For a function  $f \in L^s(\mathbb{R}^d)$ ,  $s \in \{1, 2\}$ ,  $\hat{f}$  denotes its Fourier transform:  $\hat{f}(\omega) = \int e^{-i\omega^T x} f(x) dx$ . The inner product defined over  $L^2(\mathbb{R}^d)$  is defined by  $\langle f, g \rangle = \int f(x)\bar{g}(x) dx$ , where  $\bar{a}$  denotes the complex conjugate of  $a$ . Convolution is denoted by  $*$ :  $(f * g)(x) = \int f(y)g(x-y)dy$ . Expectation is denoted by  $E$  and variance by  $\text{Var}$ , as usual.

## 3 Random representation of functional products

Our main interest is to generate random samples that can be used to represent products with two terms  $f, p$ , where  $f$  is an integrable function ( $f$  plays the role of the observation likelihood) and  $p$  is a density (the prediction density). We begin with the definition of what we mean by a properly weighted set w.r.t.  $f$  and  $p$ . This definition is a slightly modified version of the definition given in [4]:

**Definition 1** *A random variable  $X$  is said to be properly weighted by the function  $w$  with respect to the density  $p$  and the integrable function  $f$  if for any integrable function  $h$ ,  $E[h(X)w(X)] = \int h(x)f(x)p(x) dx$ . Also, in this case  $(X, w(X))$  is said to form a properly weighted pair with respect to  $f, p$ .*

*A set of random draws and weight functions  $\{(X_j, w_j)\}_{j=1, \dots, N}$  is said to be properly weighted with respect to  $p$  and  $f$  if all components of the set are properly weighted with respect to  $p$  and  $f$ .*

<sup>4</sup> The underlying domain of the functions is not important at this point. It could be any Polish set, e.g. an Euclidean space.

It should be clear that if  $\{(X_j, w_j)\}_{j=1, \dots, N}$  is a properly weighted set with respect to  $p$  and  $f$  then, by the law of large numbers, the sample averages  $J_N(h, w) = (1/N) \sum_{j=1}^N h(X_j)w(X_j)$  will converge to  $I(hfp)$  under fairly mild conditions. Hence, in this sense, a properly weighted set with respect to  $f$  and  $p$  can thought of as representing the product density  $f(\cdot)p(\cdot)$ .

Let us now consider two constructions for properly weighted sets. It is clear from the definition that it is sufficient to deal with the case of a single random variable.

The most obvious way to obtain a properly weighted pair is to sample  $X$  from  $p$  and define  $w = f$ . Then, trivially  $E[h(X)w(X)] = E[h(X)f(X)] = I(hfp)$ . We shall call this the *canonical* or *basic sampling scheme*.

A central question of Monte-Carlo sampling is how to obtain a properly weighted set such that the variance of the estimate of  $I(hfp)$  provided by the sample average  $J_N(h, w)$  is minimized. Actually, we are more interested in studying the weight-normalized averages  $I_N(h, w) = \frac{\sum_{j=1}^N h(X_j)w(X_j)}{\sum_{j=1}^N w(X_j)}$  that converge to the normalized value  $I(hfp)/I(fp)$  as  $N \rightarrow \infty$  with probability one, under a broad range of conditions on  $(X_j, w_j)$ . Obviously, the optimal sampling construction depends on  $h$ . Since we are interested in the case when  $h$  is not fixed, it is sensible to use the ‘‘rule of thumb’’ presented in Liu [5] (based on [6]) to measure the efficiency of a sampling construction by a quantity inversely proportional to the variance of  $w(X)$ , where  $w$  is such that  $E[w(X)] = 1$ . By straightforward calculations one can show that Liu’s measure still applies to our case, because  $E[w(X)] = I(fp) \equiv \text{const}$ , independently of the choice of  $X$  and  $w$ . We make this rule as our starting point and will compare different sampling schemes by the variance of the weights. Now, since for any properly weighted pair  $(X, w)$ ,  $E[w(X)] = I(fp)$  and  $\text{Var}[w(X)] = E[w^2(X)] - E[w(X)]^2$  we find that minimizing  $\text{Var}[w(X)]$  is equivalent to minimizing  $E[w^2(X)]$ . Note that if  $X$  is drawn from  $p$  and  $w$  is set to be equal to  $f$ , then  $E[w^2(X)] = I(f^2p)$ .

The sampling scheme we propose works by locally perturbing the samples drawn from  $p$  to move them closer to the modes of  $f$ . Let  $g \in L^1$  be a compactly supported function with  $I(g) = 1$ .

#### *Locally Perturbed Sampling Procedure*

1. Draw  $N$  independent samples  $X_1, \dots, X_N$  from  $p$ .
2. For each  $1 \leq j \leq N$ , draw samples  $Z_j$  from  $\frac{f(z)g(X_j - z)}{(f * g)(X_j)} dz$ .
3. Calculate the weights  $w_j = (f * g)(X_j)p(Z_j)/p(X_j)$  and output  $\{(Z_j, w_j)\}$ .

The algorithm first draws samples from  $p$ , just like the canonical one. In the second step, the samples are ‘moved’ towards the modes of  $f$ , but stay in the close vicinity of the drawn samples thanks to the compact support of  $g$ .<sup>5</sup> Hence the name of the procedure. In the next step, weights are calculated so that  $(Z_j, w_j)$  becomes a properly weighted pair for  $f, p$ .

<sup>5</sup> A slight variant can be obtained by employing a two-variable kernel function  $G(x, z)$  in place of  $g(x - z)$ . Then in the algorithm  $(f * g)(X_j)$  is replaced by  $\int f(z)G(x, z)dz$ .

In the above algorithm the function  $g$  could be a truncated Gaussian, or the characteristic function of some convex set. In one extreme but practical case  $g$  equals the weighted sum of translated Dirac-delta functions:  $g_m(x) = \sum_{k=1}^m v_k \delta(x - t_k)$ . We shall call such a function a Dirac-comb. When  $g$  is a Dirac-comb, the random sample  $Z_j$  is drawn from the weighted discrete distribution  $\{(X_j + t_l, p_l)\}$ , where  $p_l \propto f(X_j + t_l)$ .

The following proposition shows that the proposed sampling scheme results in unbiased samples.<sup>6</sup>

**Proposition 1** *Assume that  $g \in L^1$  is a compactly supported function satisfying  $g \geq 0$  and  $I(g) = 1$ . Let  $f$  be a bounded, integrable function and let  $p$  be a density. Then, the above sampling procedure yields properly weighted pairs  $(Z_j, w_j)$  with respect to  $f, p$ .*

The efficiency of the scheme will obviously depend on the correlation of  $f$  and  $p$ : if the modes of  $p$  were far away from the modes of  $f$  then the scheme will be inefficient. Another source of inefficiency is when the support of  $g$  is too small to move the samples to the vicinity of the modes of  $f$  or when it is too large. In the limit when the support of  $g$  grows to  $\mathcal{X}$  the scheme reduces to likelihood sampling. The following proposition provides the basic ground for the analysis of the efficiency of this scheme.

**Proposition 2** *Assume that  $g \in L^1$  is an even, compactly supported function satisfying  $g \geq 0$  and  $I(g) = 1$  and let  $f$  be a bounded, integrable, nonnegative function and let  $p$  be a density. Define the operator  $A : L^1 \rightarrow L^\infty$  by*

$$(Ah)(u) = \begin{cases} \int h(t)p(t)g(t-u) \left( \frac{p(t)}{p(u)} - 1 \right) dt, & \text{when } p(u) > 0; \\ 0, & \text{otherwise.} \end{cases}$$

*Assume that for some  $s \in [1, \infty]$ ,  $\epsilon = \sup_{h \in L^1, h \geq 0} \sup_u (Ah)(u) / \|h\|_s < +\infty$ . Let  $(Z, w)$  be a random sample as defined in the above algorithm. Then  $E[w^2] \leq \langle f * g, fp * g \rangle + \epsilon I(f) \|f\|_s$ .*

From this proposition it follows immediately that the proposed scheme is more efficient than the canonical algorithm whenever  $\epsilon I(f) \|f\|_s \leq \langle f, fp \rangle - \langle f * g, (fp) * g \rangle$ . Clearly, this formula agrees well with our earlier intuition: the right hand side is maximized, when the cross-correlation of  $f$  and  $fp$  is high and the cross-correlation of  $f * g$  and  $(fp) * g$  is small. In some cases convolution with  $g$  can thought of as a low-pass filtering operation (e.g. think about when  $g$  is the characteristic function of the unit interval) and hence  $g$  cuts some of the high frequency of  $f$  and  $fp$ . As a result, the cross-correlation of  $f * g$  and  $(fp) * g$  can be expected to be smaller than the cross-correlation of  $f$  and  $fp$ .

It remains to see, however, whether  $A$  is bounded and  $\epsilon$  is well defined. For this define  $p_d(x) = \inf_{\|y\| < d} p(x + y)$ . Then, one can show that for  $s = 1$ ,

<sup>6</sup> Here and in what follows the proofs are omitted due to a lack of space. An extended version of the paper available on the website of the authors contains all the missing proofs. The proof of this proposition uses Fubini's theorem and  $I(f * g) = I(f)$ .

$\epsilon \leq \|p\|_\infty \|p/p_d\|_\infty < +\infty$ . Of course, this bound is not particularly tight and much tighter bounds can be derived in special cases. For example when  $g$  is equal to the Dirac-comb defined earlier and if  $\max_k \|t_k\| \leq d$  then  $(Af)(u) \leq \|f\|_\infty \|p\|_\infty \|p/p_d\|_\infty$ . Therefore, in this case Proposition 2 holds with  $s = +\infty$ . By means of some convexity arguments one may also derive bounds for mixture densities. This can be useful to derive sharper bounds on  $\epsilon$ .

## 4 Particle Filters Enhanced by Local Likelihood Sampling

Let us now consider the problem of filtering a non-linear system of the form  $X_t = a(X_{t-1}) + W_t$ ,  $Y_t = b(X_t) + V_t$ , where  $X_t \in \mathcal{X}$  is the state of the system<sup>7</sup> at time  $t$  and  $Y_t \in \mathbb{R}^p$  is the observation at time  $t$ . We assume that  $X_0 \sim p_0$ . Here  $W_1, V_1, W_2, V_2, \dots$  are independent,  $W_1, W_2, \dots$  are identically distributed, just like  $V_1, V_2, \dots$ . For the sake of simplicity, we further assume that the densities  $K(x|x') = p(X_t = x | X_{t-1} = x')$  and  $f(y|x) = p(Y_t = y | X_t = x)$  exist. The problem we consider is the estimation of the posterior  $p(X_t | Y_{1:t})$ , where  $Y_{1:t}$  denotes the sequence of past observations:  $Y_{1:t} = (Y_1, \dots, Y_t)$ .

Particle filters approximate the posterior by a random measure  $\pi_t(x) = \left( \sum_{k=1}^N w_t^{(k)} \delta(x - X_t^{(k)}) \right) / \sum_{k=1}^N w_t^{(k)}$ , where  $X_t^{(k)}$  are called the particles, and  $w_t^{(k)}$  is the weight of the  $i$ th particle.  $X_t^{(k)}, w_t^{(k)}$  are random quantities and depend on the sequence of past observation  $Y_{1:t}$ . The best known particle filter is probably the SIR<sup>8</sup> filter [7], also known as CONDENSATION [8]:

*SIR filter*

1. Draw  $N$  independent samples  $X_0^{(1)}, \dots, X_0^{(N)}$  from  $p_0$ .
2. Repeat for  $t = 1, 2, \dots$ :
  - (a) Draw  $\hat{X}_t^{(k)} \sim q(x_t | X_{t-1}^{(k)}, Y_t)$ ,  $k = 1, \dots, N$  independently of each other.
  - (b) Calculate the weights  $w_t^{(k)} = \left( f(Y_t | \hat{X}_t^{(k)}) K(\hat{X}_t^{(k)} | X_{t-1}^{(k)}) \right) / q(\hat{X}_t^{(k)} | X_{t-1}^{(k)}, Y_t)$ .
  - (c) Draw a sequence of independent indexes  $j_1, \dots, j_N$  such that  $p(j_i = k) \propto w_t^{(k)}$  and set  $X_t^{(k)} = \hat{X}_t^{(j_k)}$ . The corresponding weights are set to  $1/N$ .

SIR uses importance sampling to sample from  $\frac{f(Y_t | \cdot) K(\cdot | X_{t-1}^{(k)})}{p(Y_t | Y_{1:t})}$ , hence its efficiency will depend on how well the proposal density  $q$  matches the shape of this function. A particularly popular choice for the proposal  $q$  is the *prediction density*  $K$ :  $q(x_t | X_{t-1}^{(k)}, Y_t) = K(x_t | X_{t-1}^{(k)})$ . In this case drawing from  $q$  is equivalent to simulating the dynamics of the system for a single time-step starting from  $X_{t-1}^{(k)}$ . This is typically simple to implement, hence the popularity of this choice. Also, in this case the weights become particularly simple to compute:  $w_t^{(k)} = f(Y_t | X_t^{(k)})$ . We shall call the SIR algorithm with this choice the “basic” or “canonical” SIR algorithm.

<sup>7</sup> Typically we will have  $\mathcal{X} = \mathbb{R}^d$ .

<sup>8</sup> Sampling Importance Resampling

Following the “rule of thumb” discussed in the previous section we shall measure the efficiency of sampling at time step  $t$  by  $\text{Var}[w_t^{(k)} | X_{t-1}^{(k)}, Y_{1:t}]$ .

Our proposed new particle filter uses the method of the previous section to boost the performance of this algorithm in the case of low observation noise:

*Local Likelihood Sampling SIR (LLS-SIR)*

1. Draw  $N$  independent samples  $X_0^{(1)}, \dots, X_0^{(N)}$  from  $p_0$ .
2. Repeat for  $t = 1, 2, \dots$ :
  - (a) Draw  $\hat{X}_t^{(k)} \sim K(\cdot | X_{t-1}^{(k)})$ ,  $k = 1, \dots, N$ , independently of each other.
  - (b) Draw  $Z_t^{(k)} \sim \frac{1}{\alpha_t^{(k)}} f(Y_t | \cdot) g(\hat{X}_t^{(k)} - \cdot)$ ,<sup>9</sup> independently of each other, where  $\alpha_t^{(k)} = \int f(Y_t | x) g(\hat{X}_t^{(k)} - x) dx$ .
  - (c) Calculate the weights  $w_t^{(k)} = \alpha_t^{(k)} K(Z_t^{(k)} | X_{t-1}^{(k)}) / K(\hat{X}_t^{(k)} | X_{t-1}^{(k)})$ .
  - (d) Resample from  $\{(Z_t^{(k)}, p_t^k)\}$  with  $p_t^k \propto w_t^{(k)}$  just like it was done in SIR to get the particles  $X_t^{(k)}$ . Set the weights uniformly to  $1/N$ .

The algorithm is identical to SIR except that the new particle positions are determined using the two-stage sampling procedure introduced in the previous section.

The following proposition shows the 1-step unbiasedness of the algorithm:

**Proposition 3** *Assume that  $g$  is a non-negative, compactly supported, integrable function satisfying  $I(g) = 1$ . Then LLS-SIR does not introduce any more bias than the SIR algorithm in the sense that for any integrable function  $h$  one has*

$$E[w_t^{(k)} h(Z_t^{(k)}) | Y_{1:t}] = E[h(X_t) | Y_{1:t}] p(Y_t | Y_{1:t-1}).$$

As a consequence of this proposition, convergence results analogous to those that are known for the SIR algorithm can be derived.

Now, we compare the efficiency of the proposed algorithm with that of the basic SIR algorithm. We shall focus on the case when  $g$  is a Dirac-comb since this choice allows one to implement the filter for continuous state spaces which is the case that we are particularly interested in.

## 5 Variance Analysis: the Case of the Dirac-Comb

In this section for the sake of simplicity we consider one-dimensional systems only. Note that these results extend to multi-dimensional systems without any problems.<sup>10</sup> We shall consider the choice  $g_m(x) = \frac{1}{m} \sum_{l=-(m-1)/2}^{(m-1)/2} \delta(x-l\lambda)$ , where  $m, \lambda > 0$ .

The following theorem expresses the difference between the appropriately normalized variance of basic SIR and that of LLS-SIR. The normalization is

<sup>9</sup> Here  $g$  is a compactly supported, integrable, nonnegative function satisfying  $I(g) = 1$  as before.

<sup>10</sup> Note, however, that for high-dimensional state spaces more efficient schemes are needed. One such scheme is given in Section 6.

intended to compensate for the  $m$ -times larger number of likelihood calculations required by LLS-SIR. The estimate developed here shows that LLS-SIR is more advantageous than SIR when started from the same particle set.

**Theorem 1** *Assume that both the basic SIR and the proposed Local Likelihood Sampling algorithm each draw  $X_t^{(k)}$  from  $K(\cdot|X_{t-1}^{(k)})$ , where  $X_{t-1}^{(k)}$  is a common sample from a density  $\hat{p}(\cdot) \approx p(\cdot|Y_{1:t-1})$ . Let  $w_t^{(k)}(\text{SIR})$  and  $w_t^{(k)}(P)$  denote the (unnormalized) weights of the basic SIR and the Local Likelihood Sampling algorithm, respectively. Let  $\Delta = \frac{1}{Nm} \text{Var}[w_t^{(k)}(\text{SIR})|X_{t-1}^{(k)}, Y_{1:t}] - \frac{1}{N} \text{Var}[w_t^{(k)}(P)|X_{t-1}^{(k)}, Y_{1:t}]$ . Let  $\epsilon, s > 0$  be defined as in Proposition 2, where in the definition of the operator  $A$  one uses  $p(\cdot) = K(\cdot|X_{t-1}^{(k)})$ . Let  $f(\cdot) = f(Y_t|\cdot)$ . Then  $N\Delta \geq (\langle f, fp \rangle - \langle f * g, (fp) * g \rangle) - (\frac{m-1}{m} \text{Var}_p[f] + \epsilon I(f) \|f\|_s)$ . Hence, the proposed sampling scheme is more efficient than the one used by SIR provided that*

$$\langle f, p \rangle^2 \geq \langle f * g, (fp) * g \rangle + \epsilon I(f) \|f\|_s. \quad (1)$$

Let us now specialize (1) to the case when  $g$  equals to the equidistant Dirac-comb defined earlier. By using harmonic analysis arguments, one gets  $m/2 \langle f * g_m, fp * g_m \rangle \rightarrow 1/(2\pi) I(f) I(fp)$ , as  $m \rightarrow \infty$ . Hence,  $\langle f * g_m, fp * g_m \rangle \sim \frac{1}{m\pi} I(f) I(fp)$ . Hence, condition 1 can be approximated by  $\langle f, p \rangle^2 \geq \frac{1}{m\pi} I(f) I(fp) + \epsilon_m I(f) \|f\|_\infty$ . Here, we have used  $\epsilon_m$  instead of  $\epsilon$  in order to emphasize the dependency of  $\epsilon$  on  $m$ . In general  $\epsilon_m$  may (and often will) diverge to infinity as  $m \rightarrow \infty$ .<sup>11</sup> As a result we get a tradeoff as a function of  $\epsilon_m$ . In general one expects that when condition (1) is satisfied then it will be satisfied for an interval of values of  $m$ .

## 6 Experiments and Results

### 6.1 Simulation

We have simulated the system  $x_t = x_{t-1}/2 + 25x_{t-1}/(1 + x_t^2) + 8\cos(1.2t) + W_t$ ,  $y_t = |x_t|/20 + V_t$ , where  $W_t \sim N(0, 10)$  and  $V_t \sim N(0, 2)$ .<sup>12</sup> LLS-SIR was implemented by using a Gaussian kernel  $G_\sigma(x, z) \propto \exp(-(\text{sgn}(x)x^2/20 - z^2/20)^2/(2\sigma^2))$ . In this case  $f(Y_t|z)G(X_t, z)$  becomes a Gaussian in  $z^2$  and hence one can use importance sampling to sample  $Z$  from the corresponding density. The system was simulated for  $T = 60$  time steps and we have measured the performance in terms of the average RMSE of  $|x_t|$ . The number of particles was  $N = 50$ . The RMSE results obtained are 2.12 and 0.94 for SIR and LLS-SIR, respectively, whilst LLS-SIR took 1.1 times more time. Thus, in this case LLS-SIR is slightly more expensive than SIR, but this may well pay off in its increased accuracy.

### 6.2 Visual Tracking

The proposed algorithm was tested and compared with basic SIR on the problem of tracking Japanese license plates. In these experiments, the outline of a license-plate was taken as a parallelogram with two vertical lines.<sup>13</sup>

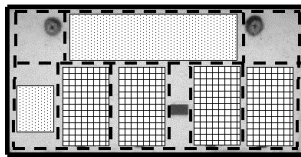
<sup>11</sup> Note that when the state space  $\mathcal{X}$  is compact then  $\epsilon_m$  stays bounded.

<sup>12</sup> This system is a slight variant of one system that has been used earlier in a number of papers, in particular in the observation function we used  $|x_t|$  instead of  $x_t^2$ .

<sup>13</sup> Thus the configuration of a license plate on the image can be defined with 5 parameters (assuming a fixed aspect ratio).



Japanese license plates enjoy a very specific geometrical structure, see Figure 1. This gives the basic idea of the observation model. The observation model is scale-free and the likelihood is expressed as the product of the likelihoods of the parts of the license plate where the parts are looked for at the precise locations as dictated by the geometry of the license plates: For each designated area of a candidate plate we compute the likelihood of the observed pattern assuming that the area is of the “right type”. The calculus of these likelihoods is implemented in an ad hoc manner using simple image processing operations that rely on measuring frequency content in the spatial domain. Based on a larger sample of images we have found that the likelihood is sufficiently specific to these kind of license plates. An example image is included in Figure 3.



**Fig. 1.** The model of a Japanese license plate. Checked means “thick line” area, dotted means “thin line” area, dashed line means “clear” area, solid line means “edge”.

The object dynamics is a mixture of an initial distribution  $p_0$  and a simple AR(2) product-process: in each time step with a constant probability we assume that the license plate reappears at a random position unrelated to the previous position. The probability of this event is set to a small value (0.1 in the experiments described here). Separate AR(2) models were used to evolve the position, scale, orientation and aspect ratio of the plate, each one independently of the others. The parameters of the AR(2) model were tuned by hand by conducting short preliminary experiments.

LLS was implemented only for the position of the plates. Denoting positions by  $(x_1, x_2)$  and by  $p(x_1, x_2)$  the corresponding weights used in the LLS step, we used  $p(x_1, x_2) = p(x_1|x_2)p(x_2)$  in the sampling step as follows. We sample first  $x_2$  from  $p(x_2)$ , and then  $x_1$  from  $p(\cdot|x_2)$ . The distribution  $p(x_2) = \sum_{x_1} p(x_2, x_1)$  was approximated by making the corresponding observation likelihoods insensitive to the precise horizontal location. The search length was half of the size of the predicted plate size, in both directions.

**Results** The performance of the local sampling algorithm with  $N = 100$  samples was compared to the performance of the basic SIR filter using  $N = 750$  particles. The number of particles for LLS-SIR was preestimated so that we expected the two algorithms to have roughly equivalent running times. It turned out that both algorithms were capable of running faster than real-time. In particular, on our 1.7GHz Intel test machine we have measured a processing speed of approximately 48 frames per seconds for the basic SIR algorithm, whilst for LLS-SIR the measured processing speed was approximately 65 frames per seconds, i.e., we have slightly overestimated the resource requirements of LLS-SIR.

Performance evaluation was done as follows: We selected a test video sequence that consisted of 298 frames. In each time step particle locations were averaged to get the final guess of the license plate position. This position was compared to the “ground truth” obtained by running basic SIR for the test video sequence with  $N = 10,000$  of particles and then correcting the results manually. Some frames of this sequence are shown in Figure 2. To be able to judge the difficulty of the tracking task Figure 3 shows the observation likelihood function of a selected frame. On this image the intensity of a pixel is proportional to the logarithm of the maximum of the observation likelihood where the maximum is taken for plate configurations with the center of gravity of plates matching the pixel’s position, the orientation matching the best orientation, but keeping the scale of the plates free.

Define the distance of two license plate configurations as the sum of distances of their corresponding vertex points. If this distance is larger than one third of the license plate height then the license plate is considered to be “lost”. The probability of this event was estimated for each frame by means of running  $n = 100$  Monte-Carlo experiments. Figure 4 shows the histogram of the probabilities of object loss on the test sequence. Note the log-scale of both axis. The percentage of frames when LLS-SIR tracks the plates (i.e it never loses the plate in any of the experiments) is over 94%. The corresponding number is 77% for SIR. Tracking error was measured for those frames when the object was tracked by the respective algorithms. The median tracking error is 1.23 pixels and 5.36 pixels for LLS-SIR and SIR, respectively. The corresponding means are 3.62 and 5.29, the standard deviations are 4.88 and 5.91. Thus, we conclude that in this case LLS-SIR is more efficient than the basic SIR algorithm both in terms of execution speed and tracking performance.

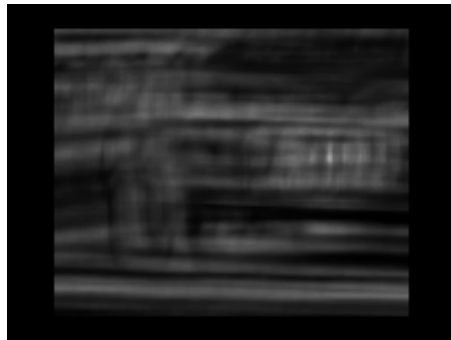
## 7 Conclusions

We have proposed a new algorithm, LLS-SIR to enhance particle filters in the low observation noise limit. The algorithm is a modification of the standard particle filter algorithm whereas after the prediction step the position of the particles are randomly resampled from the localized observation density. Theoretical analysis revealed that the scheme does not introduce any bias as compared to the basic SIR algorithm. It was also shown that the new algorithm achieves a higher effective sample size than the basic one when the observations are reliable. This results in a better tracking performance, as it was illustrated on a synthetic and a real world tracking problem.

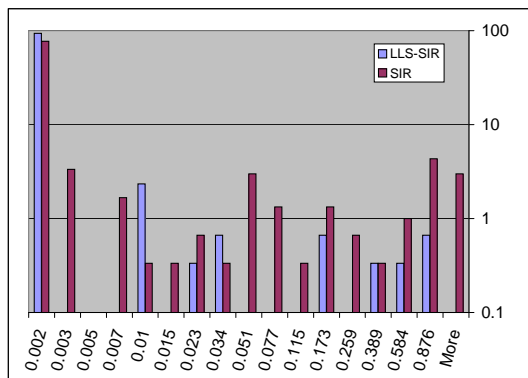
Further work shall include a more thorough evaluation of the proposed algorithm and more comparisons with competing algorithms. On the theoretical side, extending previous uniform convergence results to the new algorithm looks like an interesting challenge. Another important avenue of research is to extend the results of Section 5 so that one can compare the long term behaviour of the various algorithms. Derivation of lower bounds on the tracking accuracy could be another important next step, too.



**Fig. 2.** Sample images of the test video sequence. The video sequence is recorded by a commercial NTSC camera. The frame indexes of the images are 9,29,82,105,117 and 125. The plate positions predicted by LLS-SIR are projected back on the image.



**Fig. 3.** Log-likelihood of a selected frame of the test video sequence. Note that pixel intensities are taken for the maximum of the logarithm of the observation likelihood where scale is kept free. For more information see the text.



**Fig. 4.** Histogram of the probability of not tracking the object. Note the log-scales of the axis.

## References

1. Pitt, M.K., Shephard, N.: Filtering via simulation: Auxiliary particle filter. *Journal of the American Statistical Association* **94** (1999) 590–599
2. Fox, D., Thrun, S., Burgard, W., Dellaert, F.: Particle filters for mobile robot localization. In Doucet, A., de Freitas, N., Gordon, N., eds.: *Sequential Monte Carlo Methods in Practice*, New York, Springer (2001)
3. Torma, P., Szepesvári, C.: LS-N-IPS: an improvement of particle filters by means of local search. *Proc. Non-Linear Control Systems(NOLCOS'01)* St. Petersburg, Russia (2001)
4. Liu, J.S., Chen, R.: Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* **93** (1998) 1032–1044
5. Liu, J.S.: Metropolized independent sampling. *Statistics and Computing* **6** (1996) 113–119
6. Kong, A.: A note on importance sampling using renormalized weights. Technical report (1992)
7. Rubin, D.: Using the SIR algorithm to simulate posterior distributions. In J.M. Bernardo, M.H. DeGroot, D.L.e.A.S., ed.: *Bayesian Statistics 3*. Oxford University Press (1988) 395–402
8. Isard, M., Blake, A.: CONDENSATION – conditional density propagation for visual tracking. *International Journal Computer Vision* **29** (1998) 5–28