Self Adaptive Particle Filter

Alvaro Soto
Pontificia Universidad Catolica de Chile
Department of Computer Science
Vicuna Mackenna 4860 (143), Santiago 22, Chile
asoto@ing.puc.cl

Abstract

The particle filter has emerged as a useful tool for problems requiring dynamic state estimation. The efficiency and accuracy of the filter depend mostly on the number of particles used in the estimation and on the propagation function used to re-allocate these particles at each iteration. Both features are specified beforehand and are kept fixed in the regular implementation of the filter. In practice this may be highly inappropriate since it ignores errors in the models and the varying dynamics of the processes. This work presents a self adaptive version of the particle filter that uses statistical methods to adapt the number of particles and the propagation function at each iteration. Furthermore, our method presents similar computational load than the standard particle filter. We show the advantages of the self adaptive filter by applying it to a synthetic example and to the visual tracking of targets in a real video sequence.

1 Introduction

The particle filter is a useful tool to perform dynamic state estimation via Bayesian inference. It provides great efficiency and extreme flexibility to approximate any functional nonlinearity. The key idea is to use samples, also called particles, to represent the posterior distribution of the state given a sequence of sensor measurements. As new information arrives, these particles are constantly re-allocated to update the estimation of the state of the system.

The efficiency and accuracy of the particle filter depend mainly on two key factors: the number of particles used to estimate the posterior distribution and the propagation function used to re-allocate these particles at each iteration. The standard implementation of the filter specifies both factors beforehand and keeps them fixed during the entire operation of the filter. In this paper, we present a self adaptive particle filter that uses statistical methods to select an appropriate number of particles and a suitable propagation function at each iteration.

This paper is organized as follows. Section 2 provides background information about the standard particle filter. Section 3 presents our method to adaptively estimate the number of particles. Section 4 presents our method to adaptively improve the propagation function. Section 5 shows the results of applying the self adaptive filter to a visual tracking task. Finally, Section 6 presents the main conclusions of this work.

2 Particle Filter

In Bayesian terms, the posterior distribution of the state can be expressed as:

$$p(x_t/\vec{y}_t) = \beta\, p(y_t/x_t)\, p(x_t/\vec{y}_{t-1}) \tag{1}$$

where β is a normalization factor; x_t represents the state of the system at time t; and $\vec{y}_t$ represents all the information collected until time t. Equation (1) assumes that x_t totally explains the current observation y_t.

The particle filter provides an estimation of the posterior in Equation (1) in 3 main steps: sampling, weighting, and re-sampling. The sampling step consists of taking samples (particles) from the so-called dynamic prior distribution, $p(x_t/\vec{y}_{t-1})$. Next, in the weighting step, the resulting particles are weighted by the likelihood term $p(y_t/x_t)$. Finally, a re-sampling step is usually applied to avoid the degeneracy of the particle set. The key point that explains the efficiency of the filter comes from using a Markovian assumption and expressing the dynamic prior by:

$$p(x_t/\vec{y}_{t-1}) = \int p(x_t/x_{t-1})\, p(x_{t-1}/\vec{y}_{t-1}) dx_{t-1}. \tag{2}$$

This expression provides a recursive implementation of the filter that allows it to use the last estimation $p(x_{t-1}/\vec{y}_{t-1})$ to select the particles x^i_{t-1} in the next iteration. These particles are then propagated by the dynamics of the process $p(x_t/x^i_{t-1})$ to complete the sampling step.

At each iteration the operation of the particle filter can be seen as an importance sampling process [Tanner, 1996]. Importance sampling provides an efficient way to obtain samples from a density $p(x)$, in cases where this function can be evaluated, but it is not affordable or possible to sample from it directly. The basic idea in importance sampling is to use a proposal distribution $q(x)$, also called importance function, to obtain the samples x_i, and then weigh each sample using a compensatory term given by $p(x_i)/q(x_i)$. It is possible to show [Tanner, 1996] that under mild assumptions the set of weighted-samples resembles the target distribution $p(x)$.

The sampling and weighting steps of the particle filter correspond to the basic steps of an importance sampling process. In this case, given that the true posterior $p(x_t/\vec{y}_t)$ is not known, the samples are drawn from an importance function that corresponds to the dynamic prior $p(x_t/\vec{y}_{t-1})$. Using this importance function, the compensatory terms are exactly the non normalized weights used in the weighting step of the particle filter. The methods presented in this paper use results from the theory of importance sampling to provide a self adaptive version of the particle filter.

3 Adaptive Selection of the Number of Particles

The selection of the number of particles is a key factor in the efficiency and accuracy of the particle filter. The computational load and the convergence of the filter depend on this number. Most applications select a fixed number of particles in advance, using ad hoc criteria, or statistical methods such as Monte Carlo simulations or some standard statistical bound [Boers, 1999]. Unfortunately, the use of a fixed number of particles is often inefficient. The dynamics of most processes usually produces great variability in the complexity of the posterior distribution[1]. As a consequence, the initial estimation of the number of particles can be much larger than the real number of particles needed to perform a good estimation of the posterior distribution or, worse, at some point, the selected number of particles can be too small causing the filter to diverge.

The effect of the number of particles on the accuracy of the filter is determined by two factors: the complexity of the true density and how closely the proposal density mimics the true density. Both factors are very intuitive; the estimation of a more complex pdf requires a greater number of samples to correctly represent the less predictable shape of the function. Also, a greater mismatch between the proposal and the true densities produces many wasted samples located in irrelevant parts of the true distribution. Previous works to adaptively determine an adequate number of particles have failed to consider these two factors together [Fox *et al.*, 1999; Koeller and Fratkina, 1998; Fox, 2001].

Here, we propose two methods based in the theory of Statistics that can be used to adaptively estimate the number of particles to represent the target posterior distribution without adding a significant load to the normal operation of the filter. At each cycle of the particle filter, these techniques estimate the number of particles that, with a certain level of confidence, limits the maximum error in the approximation.

3.1 KLD-Sampling Revised

The KLD-Sampling algorithm [Fox, 2001] is a method to adaptively estimate the number of samples needed to bound the error of the particle filter. The error is measured by the Kullback-Leibler divergence (KL-divergence) between the true posterior distribution and the empirical distribution, which is a well known nonparametric maximum likelihood estimate. KLD-Sampling is based on the assumption that the true posterior can be represented by a discrete piecewise constant distribution consisting of a set of multidimensional bins. This assumption allows the use of the χ^2 asymptotic convergence of the likelihood ratio statistic to find a bound for the number of particles N:

$$N > \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta} \tag{3}$$

where ϵ is the upper bound for the error given by the KL-divergence, $(1-\delta)$ is the quantile of the χ^2 distribution with $k-1$ degrees of freedom, and k is given by the number of bins with support.

The problem with KLD-Sampling is the derivation of the bound using the empirical distribution, which has the implicit assumption that the samples comes from the true distribution. This is not the case for particle filters where the samples come from an importance function. Furthermore, the quality of the match between this function and the true distribution is one of the main elements that determines the accuracy of the filter, hence the suitable number of particles. The bound given by KLD-Sampling only uses information about the complexity of the true posterior, but it ignores any mismatch between the true and the proposal distribution.

To fix the problem of KLD-Sampling we need a way to quantify the degradation in the estimation using samples from the importance function. The goal is to find the equivalent number of samples from the importance and the true densities that capture the same amount of information about the latter.

In the context of Monte Carlo (MC) integration, [Geweke, 1989] introduced the concept of relative numerical efficiency (RNE), which provides an index to quantify the influence of sampling from an importance function. The idea behind RNE is to compare the relative accuracy of solving an integral using samples coming from both the true and the proposal density. Accuracy is measured according to the variance of the estimator of the integral.

If we use MC integration to estimate the mean value of the state ($E_{MC}(x)$), the variance of the estimator is given by [Doucet *et al.*, 2001] [2]:

$$Var[E^N_{MC}(x)] = Var_p(x)/N \tag{4}$$

where N is the number of samples coming from the true distribution $p(x)$, and the subscript p expresses that the variance involved is computed using the target distribution.

When the samples come from an importance function $q(x)$, the variance of the estimator corresponds to the variance of Importance Sampling (IS), which is given by [Geweke, 1989]:

$$Var[E^N_{IS}(x)] = E_q((x - E_p(x))^2\, w(x)^2)/N_{IS} = \sigma^2_{IS}/N_{IS}, \tag{5}$$

where $w(x)$ corresponds to $p(x)/q(x)$ the weights of IS and N_{IS} is the number of samples coming from the importance function.

[1] Here complexity is understood in terms of the amount of information needed to code the distribution.

[2] In the rest of the section, we concentrate on one iteration of the filter and we drop the subscript t.

To achieve similar levels of accuracy, the variance of both estimators should be equal. This allow us to find a relation that quantifies the equivalence between samples from the true and the proposal density,

$$N = \frac{N_{IS}\, Var_p(x)}{\sigma^2_{IS}} \tag{6}$$

Replacing (6) in (3) allows us to correct the bound given by KLD-Sampling when the samples do not come from the true distribution but from an importance function:

$$N_{IS} > \frac{\sigma^2_{IS}}{Var_p(x)} \frac{1}{2\epsilon} \chi^2_{k-1,1-\delta}. \tag{7}$$

Using MC integration, $Var_p(x)$ and σ^2_{IS} can be estimated by:

$$Var_p(x) = E_p(x^2) - E_p(x)^2 \approx \frac{\sum_{i=1}^N x_i^2\, w_i}{\sum_{i=1}^N w_i} - E_p(x)^2$$

and

$$\sigma^2_{IS} \approx \frac{\sum_{i=1}^N x_i^2\, w_i^2}{\sum_{i=1}^N w_i} - \frac{2 \sum_{i=1}^N x_i\, w_i^2\, E_p(x)}{\sum_{i=1}^N w_i} + \frac{\sum_{i=1}^N w_i^2\, E_p(x)^2}{\sum_{i=1}^N w_i} \tag{8}$$

with $E_p(x) = \sum_{i=1}^n x_i\, w_i / \sum_{i=1}^n w_i$. Equation (8) shows that using appropriate accumulators, it is possible to calculate the bound incrementally keeping the $O(N)$ complexity of the filter.

3.2 Asymptotic Normal Approximation

Usually, the particle filter keeps track of the posterior density with the goal of estimating the mean or higher order moments of the state. This suggests an alternative error metric to determine the number of particles. Instead of checking the accuracy of the estimation of the posterior, it is possible to check the accuracy of a particle filter in the estimation of a moment of this density.

Under weak assumptions and using the strong law of large numbers, it is possible to show that at each iteration the estimation of the mean given by the particle filter is asymptotically unbiased [DeGroot, 1989]. Furthermore, if the variance of this estimator is finite, the central limit theorem justifies an asymptotic normal approximation for it [DeGroot, 1989], which is given by:

$$E_{PF}(x) \sim \mathcal{N}(Ep(x), \sigma^2_{IS}/N_{IS}) \tag{9}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and standard deviation σ.

Using this approximation, it is possible to build a one sided confidence interval for the number of particles that limits the error in the estimation of the mean:

$$P(\mid \frac{E_{PF}(x) - E_p(x)}{E_p(x)} \mid \leq \epsilon) \geq (1-\alpha) \tag{10}$$

where $\mid \cdot \mid$ denotes absolute value; $E_p(x)$ is the true mean value of the state; ϵ corresponds to the desired error; and $(1-\alpha)$ corresponds to the confidence level.

Following the usual derivation of confidence intervals, Equation (10) produces the following bound for the number of particles:

$$N_{IS} \geq \frac{Z^2_{1-\alpha/2}\, \sigma^2_{IS}}{\epsilon^2\, E_p(x_t)^2} \tag{11}$$

3.3 Testing the Bounds

Figure 1 shows the distributions used to test the bounds. The true distribution corresponds to $p(x) = 0.5\,\mathcal{N}(3,2) + 0.5\,\mathcal{N}(10,2)$ and the importance function to $q(x) = 0.5\,\mathcal{N}(2,4) + 0.5\,\mathcal{N}(7,4)$. In all the trials, the desired error was set to 0.01 and the confidence level to 95%.

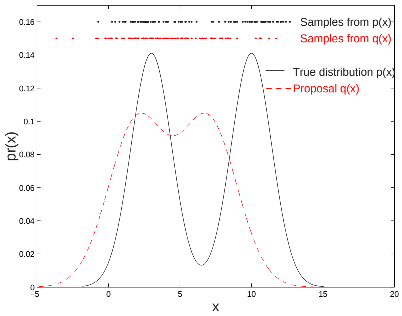

Figure 1: Inefficient allocation of samples due to a mismatch between $p(x)$ and $q(x)$.

Figure 2a shows the number of particles predicted by the different bounds. The predicted number of particles is highly consistent. Also, the revised versions of KLD-Sampling consistently require a larger number of particles than the original algorithm. This is expected because of the clear mismatch between the proposal and the true densities.

Figure 2b shows the resulting KL-divergence for each case. It is interesting to note how the practical results match closely the theoretical ones. Using the original version of KLD-Sampling, the error in the estimation is significantly greater than the one specified. However, when the same predicted number of particles is sampled from the true distribution (solid-line), the resulting error matches closely the one specified. This shows clearly that constraining the sampling to the right assumption, the original bound predicted by KLD-Sampling is correct. In the case of the revised versions of KLD-Sampling the resulting error using Equation (7) matches closely the one specified. In the same way, the error provided by the bound in Equation (11) also matches closely the level specified.

4 Adaptive Propagation of the Particles

The regular implementation of the particle filter propagates the particles using the dynamic prior $p(x_t / \vec{y}_{t-1})$. This strategy has the limitation of propagating the samples without considering the most recently available observation, y_t. Importance sampling suggests that one can use alternative propagation functions that can provide a better allocation of the samples, for example, a suitable functional of y_t. Unfortunately, the use of an arbitrary importance function significantly increases the computational load of the particle filter. In this case, as opposed to the standard particle filter, the estimation of each weight requires the evaluation of the dynamic prior. This section shows a method to build an importance function that takes into account the most recent observation y_t without increasing the computational complexity of the filter.

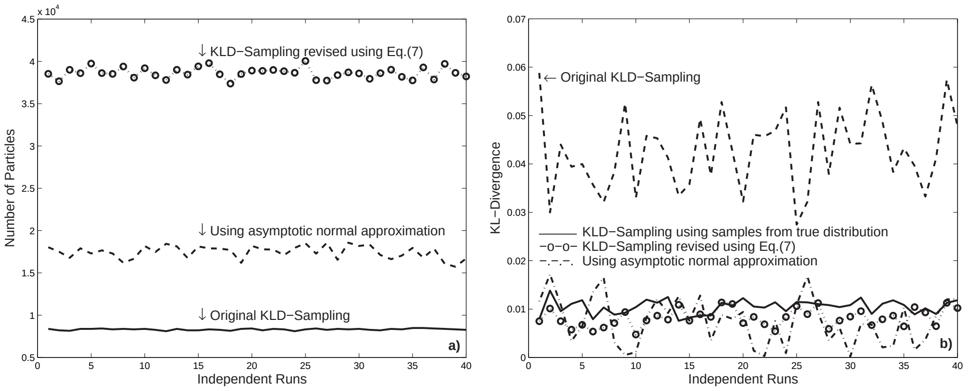

Figure 2: a) Number of particles given by each bound. b) Resulting KL-divergence.

4.1 Previous Work

In the literature about particle filters and importance sampling, it is possible to find several techniques that help to allocate the samples in areas of high likelihood under the target distribution. The most basic technique is rejection sampling [Tanner, 1996]. The idea of rejection sampling is to accept only the samples with an importance weight above a suitable value. The drawback is efficiency: there is a high rejection rate in cases where the proposal density does not match closely the target distribution. In [West, 1993], West presents a kernel-based approximation to build a suitable importance function. However, the computational complexity of the method is unaffordable. In the context of mobile robot localization, Thrun et al. [Thrun *et al.*, 2000] propose to sample directly from the likelihood function, but in many applications this is not feasible or prohibitive.

Pitt and Shephard propose the auxiliary particle filter [Pitt and Shephard, 1999]. They augment the state representation with an auxiliary variable. To sample from the resulting joint density, they describe a generic scheme with computational complexity of $O(N)$. The disadvantage is the additional complexity of finding a convenient importance function. Pitt and Sheppard provide just general intuitions about the form of a this function. In this paper we improve on this point by presenting a method to find a suitable importance function.

4.2 Adaptive Propagation of the Samples

Sampling from the dynamic prior in Equation (2) is equivalent to sample from the following mixture distribution:

$$p(x_t/\vec{y}_{t-1}) \approx \sum_{k=1}^{n} \beta_k \, p(x_t/x_{t-1}^k) \qquad (12)$$

where the mixture coefficients β_k are proportional to $p(x_{t-1}/\vec{y}_{t-1})$. The key observation is that under this scheme the selection of each propagation density depends on the mixture coefficients β_k's, which do not incorporate the most recent observation y_t. From an MC perspective, it is possible to achieve a more efficient allocation of the samples by including y_t in the generation of the coefficients. The intuition is that the incorporation of y_t increases the number of samples drawn from mixture components $p(x_t/x_{t-1}^k)$ associated with areas of high probability under the likelihood function.

Under the importance sampling approach, it is possible to generate a new set of coefficients β_k^* that takes into account y_t by sampling from the importance function $p(x_{t-1}/\vec{y}_t)$. In this way, the set of samples x_t^i from the dynamic prior $p(x_t/\vec{y}_{t-1})$ is generated by sampling from the mixture,

$$\sum_{k=1}^{n} \beta_k^* \, p(x_t/x_{t-1}^k) \qquad (13)$$

and then adding to each particle x_t^i a compensatory weight given by,

$$w_t^i = \frac{p(x_{t-1}^k/\vec{y}_{t-1})}{p(x_{t-1}^k/\vec{y}_t)}, \quad \text{with} \quad x_t^i \sim p(x_t/x_{t-1}^k) \qquad (14)$$

The resulting set of weighted samples $\{x_t^i, w_t^i\}_{i=1}^n$ still comes from the dynamic prior, so the computational complexity of the resulting filter is still $O(N)$. The extra complexity of this operation comes from the need to evaluate and to draw samples from the importance function $p(x_{t-1}^i/\vec{y}_t)$. Fortunately, the calculation of this function can be obtained directly from the operation of the regular particle filter. To see this clearly, consider the following:

$$\begin{aligned} p(x_t, x_{t-1}/\vec{y}_t) &\propto p(y_t/x_t, x_{t-1}, \vec{y}_{t-1}) \, p(x_t, x_{t-1}/\vec{y}_{t-1}) \\ &\propto p(y_t/x_t) \, p(x_t/x_{t-1}, \vec{y}_{t-1}) p(x_{t-1}/\vec{y}_{t-1}) \\ &\propto p(y_t/x_t) p(x_t/x_{t-1}) p(x_{t-1}/\vec{y}_{t-1}) \quad (15) \end{aligned}$$

Equation (15) shows that, indeed, the regular steps of the particle filter generate an approximation of the joint density $p(x_t, x_{t-1}/\vec{y}_t)$. After re-sampling from $p(x_{t-1}/\vec{y}_{t-1})$, propagating these samples with $p(x_t/x_{t-1})$, and calculating the weights $p(y_t/x_t)$, the set of resulting sample pairs (x_t^i, x_{t-1}^i) with correcting weights $p(y_t/x_t^i)$ forms a valid set of samples from the joint density $p(x_t, x_{t-1}/\vec{y}_t)$. Considering that $p(x_{t-1}/\vec{y}_t)$ is just a marginal of this joint distribution, the set of weighted-samples x_{t-1}^i are valid samples from it.

The previous description provides an adaptive algorithm that allows the particle filter to use y_t in the allocation of the samples. First, N particles are used to generate the importance function $p(x_{t-1}/\vec{y}_t)$. Then, starting from this importance function, another N particles are used to generate the desired posterior $p(x_t/\vec{y}_t)$. The relevant compensatory weights are calculated according to Equation (14) and the likelihood term $P(y_t/x_t)$. The resulting filter has a computational complexity of $O(2N)$.

In the previous algorithm the overlapping between a regular iteration of the regular particle filter and the process of generating the importance function provides a convenient way to perform an online evaluation of the benefits of updating the dynamic prior with information from the last observation. While in cases of a poor match between the dynamic prior and the posterior distribution the updating of the dynamic prior can be beneficial, in cases where these distributions agree, the updating does not offer a real advantage, and the extra processing should be avoided. To our current knowledge, this issue has not been addressed before.

The basic idea is to quantify at each iteration of the particle filter the trade-off between continuing drawing samples from a known but potentially inefficient importance function $p(x_{t-1}/\vec{y}_{t-1})$ versus incurring in the cost of building a new importance function $p(x_{t-1}/\vec{y}_t)$ that provides a better allocation of the samples under the likelihood function. The important observation is that, once the regular particle filter reaches an adequate estimate, it can be used to estimate both the posterior distribution $p(x_t/\vec{y}_t)$ and the updated importance function $p(x_{t-1}/\vec{y}_t)$.

The last step of the algorithm is to find a metric that provides a way to quantify the efficiency in the allocation of the samples. Considering that the efficiency in the allocation of the samples depends on how well the dynamic prior resembles the posterior distribution, an estimation of the distance between these two distributions is a suitable index to quantify the effectiveness of the propagation step. We found a convenient way to estimate the Kullback-Leibler divergence (KL-divergence) between these distributions, and in general between a target distribution $p(x)$ and an importance function $q(x)$:

$$KL(p(x), q(x)) \approx \log(N) - H(\hat{w_i}). \qquad (16)$$

Equation (16) states that for a large number of particles, the KL-divergence between the dynamic prior and the posterior distribution can be estimated by calculating how far the entropy of the distribution of the weights, $H(\hat{w_i})$, is from the entropy of a uniform distribution ($log(N)$). This is an intuitive result because in the ideal case of importance sampling, where $p(x) = q(x)$, all the weights are equal. In consequence the entropy of the weights is a suitable value to quantify the efficiency in the allocation of the samples.

5 Application

To illustrate the advantages of the self adaptive particle filter, we use a set of frames of a video sequence consisting of two children playing with a ball. The goal is to keep track of the positions of the ball and the left side child. Each hypothesis about the position of a target is given by a bounding box defined by its height, width, and the coordinates of its center. The motion model used for the implementation of the particle filter corresponds to a Gaussian function of zero mean and diagonal covariance matrix with standard deviations of 20 for the center of each hypothesis and 0.5 for the width and height.

Figure 3 shows the results of tracking the targets using the self adaptive particle filter. The bounding boxes correspond to the most probable hypotheses in the sample set used to estimate the posterior distributions of the states. In the estimation of the number of particles, we just consider the x and y coordinates of the center of the bounding boxes, assuming independence to facilitate the use of Equation (7). We set the desired error to 0.01 and the confidence level to 95%. A minimum number of 1000 samples is always used to ensure that convergence has been achieved. In the adaptation of the propagation function we set the threshold for the entropy of the weights in 2.

Figure 4-left shows the number of particles needed to estimate the posterior distribution of the ball at each frame without adapting the propagation function. Figure 4-right shows the number of particles in the case of adapting the importance function. The tracking engine decides to adapt the importance function at all the frames where the ball travels from one child to the other (Frames 3-7).

In the case of tracking the child, the result shows that there is not a major difference between the self adaptive particle filter and the regular filter. The self adaptive filter needs a roughly constant number of particles during the entire sequence without needing to adapt the importance function. This is expected because the child has only a small and slow motion around a center position during the entire sequence. Therefore the stationary Gaussian motion model is highly accurate and there is not a real advantage of adapting the number of particles or the propagation function.

In the case of the ball, the situation is different. During the period that the ball travels from one child to the other (Frames 3 to 7), it has a large and fast motion, therefore the Gaussian motion model is a poor approximation of the real motion. As a consequence there is a large mismatch between the dynamic prior and the posterior distribution. This produces an inefficient allocation of the samples and the estimate without adapting the importance function needs a larger set of samples to populate the relevant parts of the posterior. In contrast, when adapting the importance function during Frames 3 to 7 it is possible to observe a significant reduction in the number of samples due to a better allocation of them.

6 Conclusions

In this paper we present a self adaptive version of the particle filter that uses statistical techniques to estimate a suitable number of particles and to improve the propagation function. In terms of the estimation of the number of particles, the validation of the bounds using a synthetic example shows that the empirical results match closely the theoretical predictions. In particular, the results indicate that by considering the complexity of the true density and how closely the proposal density mimics the true density, the new bounds show a clear improvement over previous techniques such as KLD-Sampling.

The mechanisms used by the self adaptive filter to adapt the importance function and to identify when the adaptation of the importance function may be beneficial proved to be highly relevant. Using these mechanisms to track targets in a real video sequence, the self adaptive filter was able to ef-

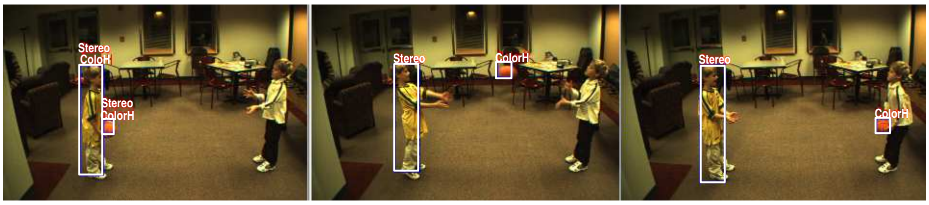

Figure 3: Tracking results for the ball and the left side child for frame 1, 5, and 14. The bounding boxes correspond to the most probable hypotheses in the sample set used to estimate the posterior distributions.

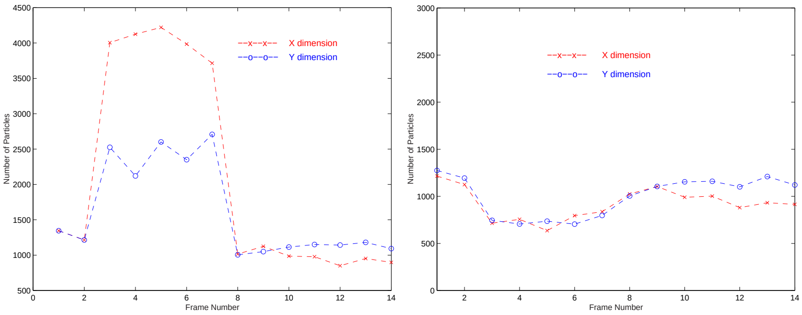

Figure 4: Number of particles used at each iteration to track the ball. Left: without adapting the importance function. Right: Adapting the importance function.

ficiently track targets with different motions using a general Gaussian motion model. Furthermore, by avoiding the need of overestimating the number of particles and by allocating these particles in areas of high likelihood, the self adaptive filter proved to operate with a similar computational complexity to the regular particle filter.

References

[Boers, 1999] Y. Boers. On the number of samples to be drawn in particle filtering. In *IEE Colloquium on Target Tracking: Algorithms and Applications*, pages 5/1 –5/6, 1999.

[DeGroot, 1989] M. DeGroot. *Probability and Statistics*. Addison Wesley, 2nd edition, 1989.

[Doucet *et al.*, 2001] A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*, pages 3–14. Springer, 2001.

[Fox *et al.*, 1999] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proc. of the nat. conf. on AI (AAAI)*, 1999.

[Fox, 2001] D. Fox. KLD-Sampling: Adaptive particle filters. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.

[Geweke, 1989] J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57:1317–1339, 1989.

[Koeller and Fratkina, 1998] D. Koeller and R. Fratkina. Using learning for approximation in stocastic processes. In *Proc. of the international conf. on machine learning (ICML)*, 1998.

[Pitt and Shephard, 1999] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.

[Tanner, 1996] M. Tanner. *Tools for Statistical Inference*. Springer-Verlag, 3nd edition, 1996.

[Thrun *et al.*, 2000] S. Thrun, D. Fox, and W. Burgard. Monte Carlo localization with mixture proposal distribution. In *AAAI National Conf. on Artificial Intelligence*, pages 859–865, Austin, Tx, 2000.

[West, 1993] M. West. Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society, Serie B*, 55(2):409–422, 1993.