KLD-Sampling: Adaptive Particle Filters and Mobile Robot Localization

Dieter Fox
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195
Email: fox@cs.washington.edu

Technical Report UW-CSE-01-08-02

September 19, 2001

Abstract

We present a statistical approach to adapting the sample set size of particle filters on-the-fly. The key idea of the KLD-sampling method is to bound the error introduced by the sample-based representation of the particle filter. Thereby, our approach chooses a small number of samples if the density is focused on a small subspace of the state space, and it chooses a large number of samples if the state uncertainty is high. Both the implementation and computation overhead of this approach are small. Extensive experiments using mobile robot localization as a test application show that our approach yields drastic improvements over particle filters with fixed sample set sizes and over a previously introduced adaptation technique.

1 Introduction

Estimating the state of a dynamic system based on noisy sensor measurements is extremely important in areas as different as speech recognition, target tracking, mobile robot navigation, and computer vision. Since many of these applications impose real-time constraints on the estimation process, *efficiency* is an important aspect of state estimation. Furthermore, since the underlying uncertainties can vary dramatically over time, another important aspect of state estimation is the need to deal with a *wide range of probability densities*.

Over the last years, particle filters have been applied with great success to a variety of state estimation problems [4, 11, 7, 16, 21, 25]. Particle filters estimate the posterior probability density over the complete state space of a dynamic system [5, 17, 19]. The key idea of this technique is to represent probability densities by sets of samples, or particles. It is due to this representation, that particle filters combine efficiency with the ability to represent a wide range of probability densities. The efficiency of particle filters lies in the way they place computational resources. By sampling

in proportion to likelihood, particle filters focus the computational resources on regions of the state space with high likelihood, where things really matter.

So far, however, an important source for increasing the efficiency of particle filters has only rarely been studied: *Adapting the number of samples over time*. Most existing approaches use a fixed number of samples during the whole state estimation process. However, this can be highly inefficient, since the complexity of the probability densities can vary drastically over time. An exception is the adaptive sampling approach applied by [14] and [7]. Both approaches adjust the number of samples based on the likelihood of observations. Unfortunately, this method has some important shortcomings, as we will show.

In this paper we introduce a novel approach to adapting the number of samples over time. In contrast to previous approaches, our technique determines the number of samples based on the complexity of the sampling distribution. Extensive experiments using a mobile robot indicate that our approach yields drastic improvements over particle filters with fixed sample set sizes and over the previously introduced adaptation technique.

The remainder of this paper is organized as follows: In the next section we will outline the basics of particle filters and their application to mobile robot localization. In Section 3, we will introduce our novel technique to adaptive particle filters. Experimental results are presented in Section 4 before we conclude in Section 5.

2 Particle filters for Bayesian filtering and robot localization

In this section we will review the basics of Bayes filters and particle filters, and their application to mobile robot localizaion (further details can be found in [9, 5]).

Bayes filters

Bayes filters address the problem of estimating the state x of a dynamical system from sensor measurements. The key idea of Bayes filtering is to recursively estimate the posterior probability density over the state space conditioned on the data collected so far. Without loss of generality, we assume that the data consists of an alternating sequence of time indexed observations o_t and control measurements u_t, which describe the dynamics of the system. The posterior at time t is called the belief $Bel(x_t)$, defined by

$$Bel(x_t) \quad = \quad p(x_t \mid y_t, u_{t-1}, y_{t-1}, u_{t-2} \ldots, u_0, y_0)$$

Bayes filters make the assumption that the dynamic system is Markov, i.e. observations y_t and control measurements u_t are conditionally independent of past measurements and control readings given knowledge of the state x_t. Under this assumption the posterior can be determined efficiently using the following two update rules: Whenever a new control measurement u_{t-1} is received, the state of the system is *predicted* according to

$$Bel^-(x_t) \quad \longleftarrow \quad \int p(x_t \mid x_{t-1}, u_{t-1})\; Bel(x_{t-1})\; dx_{t-1}\,, \tag{1}$$

and whenever an observation o_t is made, the state estimate is *corrected* according to

$$Bel(x_t) \quad \longleftarrow \quad \alpha\; p(y_t \mid x_t) Bel^-(x_t)\,. \tag{2}$$

Here, α is a normalizing constant which ensures that the belief over the entire state space sums up to one. The state right after the prediction and before the observation is called the *predictive* belief $Bel^{-}(x_t)$.

Implementations of Bayes filters mostly differ in the way they represent densities over the state x_t. For example, Kalman filters are Bayes filters which make the restrictive assumption that the posterior can be represented by Gaussian distributions [10, 22].

Particle filters

Particle filters are a variant of Bayes filters which represent the belief $Bel(x_t)$ by a set S_t of n weighted samples distributed according to $Bel(x_t)$:

$$S_t = \{\langle x_t^{(i)}, w_t^{(i)} \rangle \mid i = 1, \ldots, n\}$$

Here each $x_t^{(i)}$ is a state, and the $w_t^{(i)}$ are non-negative numerical factors called *importance weights*, which sum up to one. The basic form of the particle filter realizes the recursive Bayes filter according to a sampling procedure, often referred to as sequential importance sampling with resampling (SISR, see also [17, 5, 4]). A time update of one possible implementation of this algorithm is outlined in Table 1.

Inputs: $S_{t-1} = \{\langle x_{t-1}^{(i)}, w_{t-1}^{(i)} \rangle \mid i = 1, \ldots, n\}$ representing belief $Bel(x_{t-1})$

$S_t := \emptyset$, $\alpha = 0$

/ Generate n samples representing $Bel(x_t)$ */*
for $i := 1, \ldots, n$ **do**

/ Resampling */*
Sample an index $j(i)$ from the discrete distribution given by the weights in S_{t-1}

/ Sampling: Predict next state using the control information u_{t-1} */*
Sample $x_t^{(i)}$ from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{(j(i))}$ and u_{t-1}

/ Importance sampling: Compute importance weight based on likelihood of y_t */*
$w_t^{(i)} := p(y_t \mid x_t^{(i)})$
$\alpha := \alpha + w_t^{(i)}$

/ Insert sample into sample set */*
$S_t := S_t \cup \{\langle x_t^{(i)}, w_t^{(i)} \rangle\}$

/ Normalize importance weights */*
for $i := 1, \ldots, n$ **do**
$w_t^{(i)} := w_t^{(i)} / \alpha$

return S_t

Table 1: The basic particle filter algorithm.

In the resampling step, the algorithm basically generates samples drawn according to the prior belief $Bel(x_{t-1})$ represented by the weighted sample set S_{t-1}. The next state of these samples is predicted by sampling from the model of the systems dynamics, using the control information

u_{t-1}. After this sampling step, the samples are distributed according to the predictive density $p(x_t \mid x_{t-1}, u_{t-1})Bel(x_{t-1})$. In order to generate samples according to the next belief $Bel(x_t)$, importance sampling is applied, with $Bel(x_t)$ as target distribution and $p(x_t \mid x_{t-1}, u_{t-1})Bel(x_{t-1})$ as proposal distribution. By dividing these two distributions, we get $p(y_t \mid x_t^{(j)})$ as the importance weight for each sample [9]. After generating n samples, these weights are normalized so that they sum up to one. It can be shown that this procedure in fact implements the Bayes filter, using an (approximate) sample-based representation [5, 4].

Particle filters for mobile robot localization

In this paper, we use the problem of mobile robot localization to illustrate and test our novel approach to adaptive particle filters. Mobile robot localization is the problem of estimating a robot's pose relative to a map of its environment. Robot localization has been recognized as one of the most fundamental problems in mobile robotics [2, 1, 15]. The mobile robot localization problem comes in different flavors. The simplest localization problem—which has received by far the most attention in the literature—is *position tracking*. Here the initial robot pose is known, and localization seeks to correct small, incremental errors in a robot's odometry. More challenging is the *global localization problem*, where a robot is not told its initial pose, but instead has to determine it from scratch. The global localization problem is more difficult, since the robot's initial uncertainty can be arbitrarily large.

Robot localization can be phrased as a state estimation problem. In this context, the state x_t of the system is typically specified by the robot's position in a two-dimensional Cartesian space and the robot's heading direction θ. Measurements y_t may include range measurements and camera images, and control information u_t usually consists of the robot's odometry readings. The next state probability $p(x_t \mid x_{t-1}, u_{t-1})$ describes how the position of the robot changes based on information collected by the robot's wheel encoders. This conditional probability is typically a model of robot kinematics annotated with uncertainty. The perceptual model $p(y_t \mid x_t)$ describes the likelihood of making the observation y_t given that the robot is at location x_t. For proximity sensors such as sonar sensors, this probability can be computed from the map using ray-tracing and a model of the sensor uncertainty (see also [1, 8]).

Particle filters have been applied with great practical success to a variety of mobile robot systems [7, 3, 16, 6, 12]. Fig. 1 illustrates the application of particle filters to mobile robot localization. Shown there is a map of a hallway environment along with a sequence of sample sets during global localization. The pictures demonstrate the ability of particle filters to represent a wide variety of distributions, ranging from uniform to highly focused. Especially in symmetric environments, the ability to represent ambiguous situations is of utmost importance for the success of global localization. In this example, all sample sets contain 100,000 samples. While such a high number of samples is necessary to accurately represent the belief during early stages of localization (cf. 1(a)), it is obvious that only a small fraction of this number suffices to track the position of the robot once it knows where it is (cf. 1(c)). Unfortunately, it is not straightforward how the number of samples can be adapted on-the-fly, and this problem has only rarely been addressed so far.

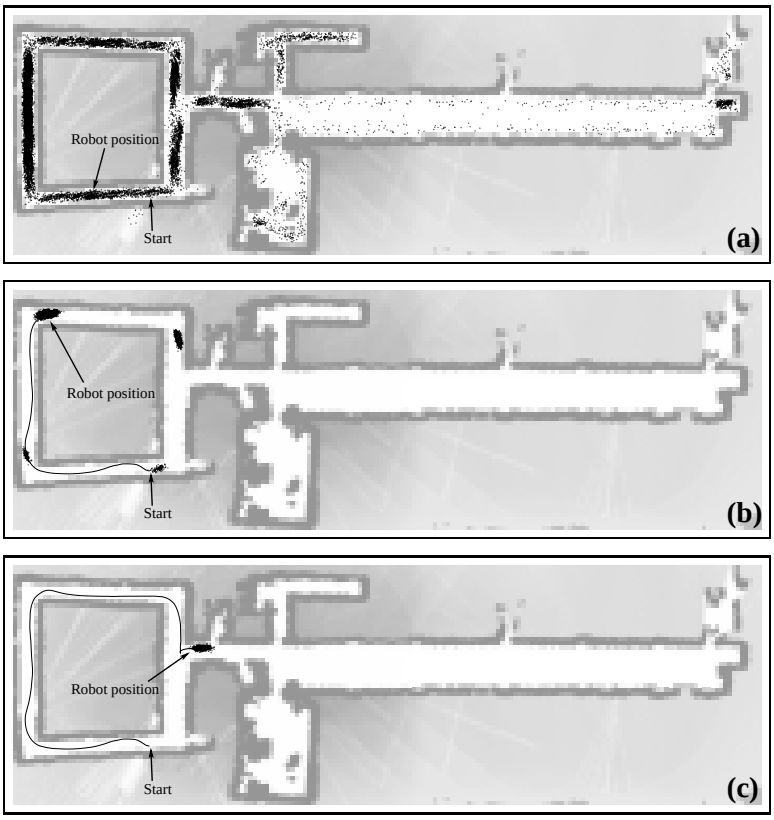

Fig. 1: Map of the UW CSE Department along with a series of sample sets representing the robot's belief during global localization using sonar sensors (samples are projected into 2D). The size of the environment is 54m $\times$ 18m. a) After moving 5m, the robot is still highly uncertain about its position and the samples are spread trough major parts of the free-space. b) Even as the robot reaches the upper left corner of the map, its belief is still concentrated around four possible locations. c) Finally, after moving approximately 55m, the ambiguity is resolved and the robot knows where it is. All computation is carried out in real-time on a low-end PC.

3 Adaptive sampling for particle filters

The localization example in the previous section illustrates that the efficiency of particle filters can be greatly increased by changing the number of samples over time: While a large number of samples is necessary to accurately represent the belief during early stages of localization (cf. 1(a)), it is obvious that only a small fraction of this number suffices to track the position of the robot once it knows where it is (cf. 1(c)). Before we introduce our novel approach to adaptive particle filters, let us first discuss and analyze an existing technique to adapting the number of samples.

3.1 Likelihood-based adaptation

We call this approach likelihood-based adaptation since it is based on the idea of determining the number of samples such that the non-normalized sum of likelihoods (importance weights) exceeds a pre-specified threshold. This approach has been applied to dynamic Bayesian networks [14] and mobile robot localization [7]. The intuition behind this approach can be illustrated in the robot localization context: If the sample set is well in tune with the sensor reading, each individual

importance weight is large and the sample set remains small. This is typically the case during position tracking (cf. 1(c)). If, however, the sensor reading carries a lot of surprise, as is the case when the robot is globally uncertain or when it lost track of its position, the individual sample weights are small and the sample set becomes large.

The likelihood-based adaptation directly relates to the property that the variance of the importance sampler is a function of the mismatch of the proposal distribution and the distribution that is being approximated with the weighted samples [23]. Unfortunately, this method does *not* consider the *complexity* of the underlying density. Consider, for example, the ambiguous belief state consisting of four distinctive sample clusters shown in Fig. 1(b). Due to the symmetry of the environment, the average likelihood of a sensor measurement observed in this situation is approximately the same as if the robot knew its position unambiguously (cf. 1(c)). Therefore, the likelihood-based approach would fail to distinguish between these two situations. Nevertheless, it is obvious that an accurate approximation of the belief shown in Fig. 1(b) requires a multiple of the samples needed to represent the belief in Fig. 1(c).

3.2 KLD-sampling

The key idea of our approach is to bound the error introduced by the sample-based representation of the particle filter. To derive this bound, we assume that the true posterior is given by a discrete, piecewise constant distribution such as a discrete density tree or a multi-dimensional histogram [14, 18, 24, 8]. For such a representation we can determine the number of samples so that the distance between the maximum likelihood estimate (MLE) based on the samples and the true posterior does not exceed a pre-specified threshold ε. We denote the resulting approach the KLD-sampling algorithm since the distance between the MLE and the true distribution is measured by the Kullback-Leibler distance. In what follows, we will first derive the equation for determining the number of samples needed to approximate a discrete probability distribution (see also [20, 13]). Then we will show how to modify the basic particle filter algorithm so that it realizes our adaptation approach.

To see, suppose that n samples are drawn from a discrete distribution with k different bins. Let the vector $\underline{X} = (X_1, \ldots, X_k)$ denote the number of samples drawn from each bin. $\underline{X}$ is distributed according to a multinomial distribution, i.e. $\underline{X} =\sim$ Multinomial$_k(n, \underline{p})$, where $\underline{p} = p_1 \ldots p_k$ specifies the probability of each bin. The maximum likelihood estimate of $\underline{p}$ is given by $\widehat{\underline{p}} = n^{-1}\underline{X}$. Furthermore, the likelihood ratio statistic λ_n for testing $\underline{p}$ is

$$\log \lambda_n = \sum_{j=1}^{k} X_j \log\left(\frac{\widehat{p}_j}{p_j}\right) = n \sum_{j=1}^{k} \widehat{p}_j \log\left(\frac{\widehat{p}_j}{p_j}\right). \tag{3}$$

When $\underline{p}$ is the true distribution, the likelihood ratio converges to a chi-square distribution:

$$2 \log \lambda_n \rightarrow_d \chi^2_{k-1} \qquad \text{as} \qquad n \rightarrow \infty \tag{4}$$

Please note that the sum in the rightmost term of Eq. (3) specifies the K-L distance $K(\widehat{\underline{p}}, \underline{p})$ between the MLE and the true distribution. Now we can determine the probability that this distance is smaller than ε, given that n samples are drawn from the true distribution:

$$P_{\underline{p}}(K(\widehat{\underline{p}}, \underline{p}) \leq \epsilon) \;\; = \;\; P_{\underline{p}}(2nK(\widehat{\underline{p}}, \underline{p}) \leq 2n\epsilon) \;\; \doteq \;\; P(\chi^2_{k-1} \leq 2n\epsilon) \tag{5}$$

The second step in Eq. (5) follows by replacing $nK(\hat{\underline{p}}, \underline{p})$ with the likelihood ratio statistic, and by using the convergence result stated in Eq. (4). The quantiles of the chi-square distribution are given by

$$P(\chi^2_{k-1} \leq \chi^2_{k-1,1-\delta}) = 1 - \delta \,. \tag{6}$$

Now if we choose n such that $2n\epsilon$ is equal to $\chi^2_{k-1,1-\delta}$, we can combine Eq. (5) and Eq. (6) to get

$$P_{\underline{p}}(K(\hat{\underline{p}}, \underline{p}) \leq \epsilon) \quad \doteq \quad 1 - \delta \,. \tag{7}$$

This derivation can be summarized as follows: If we choose the number of samples n as

$$n = \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta}, \tag{8}$$

then we can guarantee that with probability $1 - \delta$, the K-L distance between the MLE and the true distribution is less than ε. In order to determine n according to Eq. (8), we need to compute the quantiles of the chi-square distribution. A good approximation is given by the Wilson-Hilferty transformation [13], which yields

$$n \;=\; \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta} \;\doteq\; \frac{k-1}{2\epsilon}\left\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}}z_{1-\delta}\right\}^3 , \tag{9}$$

where $z_{1-\delta}$ is the upper $1 - \delta$ quantile of the standard normal $N(0, 1)$ distribution.

This concludes the derivation of the sample size needed to approximate a discrete distribution with an upper bound ε on the K-L distance. From Eq. (9) we see that the required number of samples is proportional to the inverse of the ε bound, and to first order linear in the number k of bins with non-zero probability (support).

It remains to be shown how to apply this result to particle filters. The problem is that we do not know the true posterior distribution (note that the estimation of this posterior is the main goal of the particle filter). Fortunately, Eq. (9) shows that we do not need the complete discrete distribution but that it suffices to determine the number k of bins with non-zero probability. We estimate this quantity by counting the number of bins with support during sampling. To be more specific, we estimate k for the *proposal distribution* $p(x_t \mid x_{t-1}, u_{t-1})Bel(x_{t-1})$ resulting from the first two steps of the particle filter update. The determination of k can be done efficiently by checking for each generated sample whether it falls into an empty bin or not. Sampling is stopped as soon as the number of samples exceeds the threshold specified in Eq. (9). An update step of the resulting KLD-sampling particle filter is given in Table 2.

As can be seen there, the implementation of this modified particle filter is trivial. The only difference to the original algorithm is that we have to keep track of the number k of supported bins. The bins can be implemented either as a fixed, multi-dimensional grid [8], or more efficiently as a tree structure [14, 18]. Please note that the sampling process is guaranteed to stop, since for a given bin size Δ, the number k of possible bins is limited.

4 Experimental results

We evaluated our approach using data collected with one of our robots. The robot and the map used for localization are shown in Fig. 2. The data consists of a sequence of sonar scans and odometry

Inputs: $S_{t-1} = \{\langle x_{t-1}^{(i)}, w_{t-1}^{(i)} \rangle \mid i = 1, \ldots, n\}$ representing belief $Bel(x_{t-1})$
Control measurement u_{t-1}, observation y_t
bounds ε and δ, bin size Δ

$S_t := \emptyset,\ n = 0,\ k = 0,\ \alpha = 0$

/ Generate samples until K-L bound is reached */*
do
/ Predict next state using the control information u_{t-1} */*
Sample an index $j(n)$ from the discrete distribution given by the weights in S_{t-1}
Sample $x_t^{(n)}$ from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{(j(n))}$ and u_{t-1}

/ Compute importance weight and update normalization factor */*
$w_t^{(n)} := p(y_t \mid x_t^{(n)})$
$\alpha := \alpha + w_t^{(n)}$

/ Insert sample into sample set */*
$S_t := S_t \cup \{\langle x_t^{(n)}, w_t^{(n)} \rangle\}$

/ Update number of bins with support */*
if ($x_t^{(n)}$ falls into an empty bin b) **then**
$k := k + 1$
$b :=$ non-empty

$n := n + 1$
while $(n < \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta})$

/ Normalize importance weights */*
for $i := 1, \ldots, n$ **do**
$w_t^{(i)} := w_t^{(i)} / \alpha$

return S_t

Table 1: The KLD-sampling particle filter algorithm.

measurements annotated with time-stamps to allow systematic real-time evaluations. In all experiments we compared our KLD-sampling approach to the likelihood-based approach discussed in Section 3.1, and to particle filters with fixed sample set sizes. Throughout the experiments we used different parameters for the three approaches. For the fixed approach we varied the number of samples, for the likelihood-based approach we varied the weight threshold used to determine the number of samples, and for our approach we varied ε, the bound on the K-L distance. In all experiments, we used a value of 0.99 for δ and a fixed bin size Δ of 50cm $\times$ 50cm $\times$ 10deg. We furthermore limited the maximum number of samples for all approaches to 100,000.

Approximation of the true posterior

In the first set of experiments we evaluated how accurately our adaptive approach approximates the true posterior density. Since ground truth for these posteriors is not available, we compared the sample sets generated by the different approaches with reference sample sets. These reference

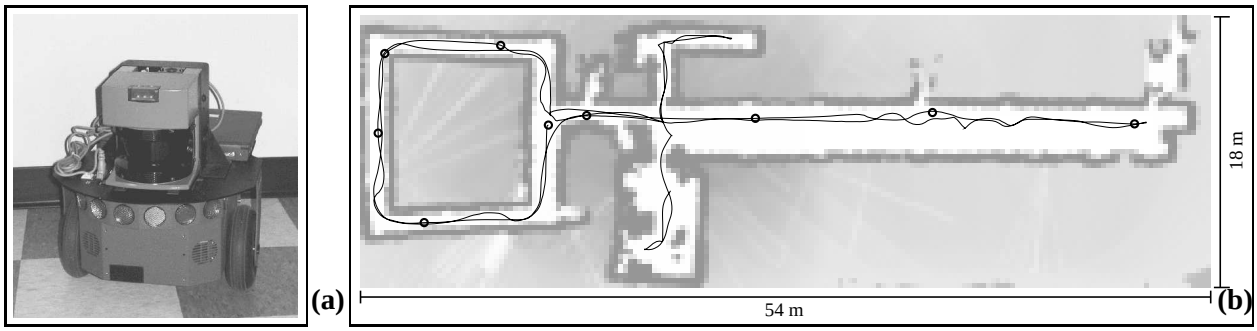

Fig. 2: a) Pioneer II robot used throughout the experiments. b) Map used for localization along with the path followed by the robot during data collection. The small circles mark the different start points for the global localization experiments.

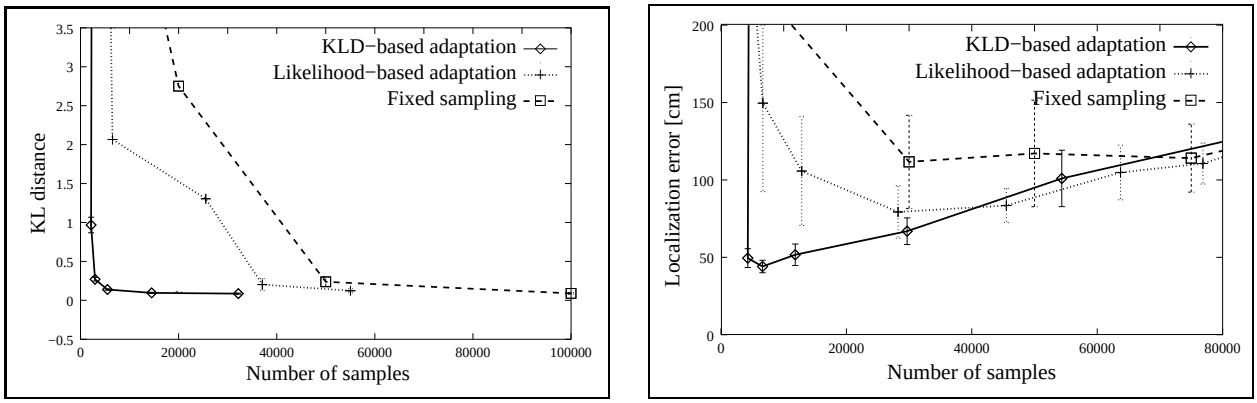

Fig. 3: The x-axis represents the average sample set size for different parameters of the three approaches. a) The y-axis plots the K-L distance between the reference densities and the sample sets generated by the different approaches. b) The y-axis represents the average localization error measured by the distance between estimated positions and reference positions.

sets were generated using a particle filter with a fixed number of 200,000 samples (far more than actually needed for position estimation). The comparison was done by computing the K-L distance between the sets and their corresponding reference sets, using histograms for both sets. Note that in these experiments the time-stamps were ignored and the algorithms was given as much time as needed to process the data. Fig. 3(a) plots the average K-L distance along with 95% confidence intervals against the number of samples for the different algorithms (for clarity, we omitted the large error bars for K-L distances above 1.0). Each data point represents the average of 16 global localization runs with different start positions of the robot (each run itself consists of approximately 150 sample set comparisons at the different points in time). As expected, the more samples are used, the better the approximation. The curves show the superior performance of our approach: While the fixed approach requires about 50,000 samples before it converges to a K-L distance below 0.25, our approach converges to the same level using only 3,000 samples on average. This is also an improvement by a factor of 12 compared to the approximately 36,000 samples needed by the likelihood-based approach. In essence, these experiments indicate that our approach, even though based on several approximations, is able to accurately track the true posterior using only a small number of samples.

Real-time performance

In these experiments we investigated the real-time localization performance of our approach. This time we performed multiple global localization experiments under real-time considerations and we estimated at each iteration the distance between the estimated robot position and the corresponding reference position [1]. The results are shown in Fig. 3(b). The U-shape of all three graphs nicely illustrates the trade-off involved in choosing the number of samples under real-time constraints: Choosing not enough samples results in a poor approximation of the underlying posterior and the robot frequently fails to localize itself. On the other hand, if we choose too many samples, each update of the algorithm takes several seconds and valuable sensor data has to be discarded. Fig. 3(b) also shows that our KLD-sampling approach yields drastic improvements over both fixed sampling and likelihood-based sampling. The smallest average localization error is 44cm in contrast to an average error of 79cm and 114cm for the likelihood-based and the fixed approach, respectively. Due to the smaller sample sets, our approach also needs significantly less processing power than any of the other approaches.

5 Conclusions

We presented a statistical approach to adapting the sample set size of particle filters on-the-fly. The key idea of the KLD-sampling approach is to bound the error introduced by the sample-based belief representation of the particle filter. At each iteration, our approach generates samples until their number is large enough to guarantee that the K-L distance between the maximum likelihood estimate and the underlying posterior does not exceed a pre-specified bound. Thereby, our approach chooses a small number of samples if the density is focused on a small subspace of the state space, and chooses a large number of samples if the samples have to cover a major part of the state space.

Both the implementational and computational overhead of this approach are small. Extensive experiments using mobile robot localization as a test application show that our approach yields drastic improvements over particle filters with fixed sample sets and over a previously introduced adaptation approach [14, 7]. For example, our KLD-sampling method yields better approximations using less than 6% of the samples required by particle filters with fixed sample set sizes, and using less than 9% of the samples required by the adaptation approach based on observation likelihoods. Our algorithm typically uses all available samples during early stages of localization and reduces the sample set size to several hundred samples once the position of the robot is determined. Animations illustrating the effectiveness of the approach can be found under `http://www.cs.washington.edu/homes/fox/animations.html`.

Despite these encouraging results, several open questions need to be addressed in future research. In our current implementation we use a discrete distribution with a *fixed* bin size to estimate the number of samples. It needs to be evaluated whether the performance of the filter can be further improved by changing the discretization over time and space (e.g. one might be interested in more accurate estimates in certain areas of the state space). So far, the KLD-sampling approach has been tested using robot localization only. However, we conjecture that many other applications of particle filters can benefit from this method. Investigating this technique in the context of other

[1] The position estimate is generated from a sample set using histograming and local averaging, and the reference positions were determined by evaluating the robot's laser range-finder information.

applications is part of future work.

Acknowledgements

The author wishes to thank Jon A. Wellner and Vladimir Koltchinskii for their help in deriving the statistical background of this work. Additional thanks go to Wolfram Burgard and Sebastian Thrun for giving valuable feedback on early versions of the technique.

References

[1] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[2] I. J. Cox and G. T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[3] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[4] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.

[5] A. Doucet, S.J. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 2000.

[6] S. Enderle, M. Ritter, D. Fox, S. Sablatng, G Kraetzschmar, and G. Palm. Soccer-robot locatization using sporadic visual features. In *Proc. of the International Conference on Intelligent Autonomous Systems (IAS)*, 2000.

[7] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999.

[8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11:391–427, 1999.

[9] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In Doucet et al. [4].

[10] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.

[11] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[12] P. Jensfelt, O. Wijk, D. Austin, and M. Andersson. Feature based condensation for mobile robot localization. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.

[13] N. Johnson, S. Kotz, and N. Balakrishnan. *Continuous univariate distributions*, volume 1. John Wiley & Sons, New York, 1994.

[14] D. Koller and R. Fratkina. Using learning for approximation in stochastic processes. In *Proc. of the International Conference on Machine Learning (ICML)*, 1998.

[15] D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors. MIT/AAAI Press, Cambridge, MA, 1998.

[16] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.

[17] J. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443), 1998.

[18] A. W. Moore, J. Schneider, and K. Deng. Efficient locally weighted polynomial regression predictions. In *Proc. of the International Conference on Machine Learning (ICML)*, 1997.

[19] M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446), 1999.

[20] J.A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, second edition, 1995.

[21] D. Schulz, W. Burgard, D. Fox, and A. B Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[22] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[23] M. A. Tanner. *Tools for Statistical Inference*. Springer Verlag, New York, 1993. 2nd edition.

[24] S. Thrun, J. Langford, and D. Fox. Monte Carlo hidden Markov models: Learning non-parametric models of partially observable stochastic processes. In *Proc. of the International Conference on Machine Learning (ICML)*, 1999.

[25] J. Vermaak, C. Andrieu, A. Doucet, and S.J. Godsill. Particle methods for bayesian modelling and enhancement of speech signals. Technical Report CUED/F-INFENG/TR.361, Cambridge University Engineering Department, 2001.