Adapting the Sample Size in Particle Filters Through KLD-Sampling

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

12th May 2003

Abstract

Over the last years, particle filters have been applied with great success to a variety of state estimation problems. In this paper we present a statistical approach to increasing the efficiency of particle filters by adapting the size of sample sets during the estimation process. The key idea of the KLD-sampling method is to bound the approximation error introduced by the sample-based representation of the particle filter. The name KLD-sampling is due to the fact that we measure the approximation error by the Kullback-Leibler distance. Our adaptation approach chooses a small number of samples if the density is focused on a small part 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. Over the last years, particle filters have been applied with great success to a variety of state estimation problems including visual tracking [41], speech recognition [77], mobile robot localization [28, 54, 43], map building [59], people tracking [69, 60], and fault detection [76, 16]. A recent book provides an excellent overview of the state of the art [22]. Particle filters estimate the posterior probability density over the state space of a dynamic system [23, 66, 3]. The key idea of this technique is to represent probability densities by sets of samples. 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 with high likelihood, where good approximations are most important.

Since the complexity of particle filters depends on the number of samples used for estimation, several attempts have been made to make more efficient use of the available samples [34, 66, 35]. 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*. While sample sizes have been discussed in the context of genetic algorithms [65] and interacting particle filters [18], most existing approaches to particle filters use a fixed number of samples during the entire state estimation process. This can be highly inefficient, since the complexity of the probability densities can vary drastically over time. Previously, an adaptive approach for particle filters has been applied by [49] and [28]. This approach adjusts the number of samples based on the likelihood of observations, which has some important shortcomings, as we will show. In this paper we introduce a novel approach to adapting the number of samples over time. Our technique determines the number of samples based on statistical bounds on the sample-based approximation quality. Extensive experiments indicate that our approach yields significant improvements over particle filters with fixed sample set sizes and over a previously introduced adaptation technique.

In this paper, we investigate the utility of adaptive particle filters in the context of mobile robot localization, which is the problem of estimating a robot's pose relative to a map of its environment. The localization problem is occasionally referred to as " the most fundamental problem to providing a mobile robot with autonomous capabilities" [13]. The mobile robot localization problem comes in different flavors. The most simple localization problem is *position tracking*. Here the initial robot pose is known, and localization seeks to identify 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 position estimate cannot be represented adequately by unimodal probability densities [9]. Only recently, several approaches have been introduced that can solve the global localization problem, among them grid-based approaches [9], topological approaches [46, 70], particle filters [28], and multi-hypothesis tracking [4, 68]. The most challenging problem in mobile robot localization is the *kidnapped robot problem* [25], in which a well-localized robot is teleported to some other position without being told. This problem differs from the global localization problem in that the robot might firmly believe to be somewhere else at the time of the kidnapping. The kidnapped robot problem, thus, is often used to test a robot's ability to recover autonomously from catastrophic localization failures. Virtually all approaches capable of solving the global localization problem can be modified such that they can also solve the kidnapped robot problem. Therefore, we will focus on the tracking and global localization problem in this paper.

The remainder of this paper is organized as follows: In the next section we will outline the basics of Bayes filters and discuss different representations of posterior densities. Then we will introduce 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 review the basics of Bayes filters and alternative representations for the probability densities, or beliefs, underlying Bayes filters. Then we introduce particle filters as a sample-based implementation of Bayes filters, followed by a discussion of their application to mobile robot localization.

2.1 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 z_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

$$Bel(x_t) \quad = \quad p(x_t \mid z_t, u_{t-1}, z_{t-1}, u_{t-2} \ldots, u_0, z_0)$$

Bayes filters make the assumption that the dynamic system is Markov, *i.e.* observations z_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 z_t is made, the state estimate is *corrected* according to

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

Here, α is a normalizing constant which ensures that the belief over the entire state space sums up to one. The term $p(x_t \mid x_{t-1}, u_{t-1})$ describes the *system dynamics*, *i.e.* how the state of the system changes given control information u_{t-1}. $p(z_t \mid x_t)$, the *perceptual model*, describes the likelihood of making observation z_t given that the current state is x_t. Note that the random variables x, z and u can be high-dimensional vectors. The belief immediately after the prediction and before the observation is called the *predictive* belief $Bel^-(x_t)$, as given in (1). At the beginning, the belief $Bel(x_0)$ is initialized with the prior knowledge about the state of the system.

Bayes filters are an abstract concept in that they only provide a probabilistic framework for recursive state estimation. To implement Bayes filters, one has to specify the perceptual model $p(z_t|x_t)$, the dynamics $p(x_t \mid x_{t-1}, u_{t-1})$, and the representation of the belief $Bel(x_t)$.

2.2 Belief representations

The properties of the different implementations of Bayes filters strongly differ in the way they represent densities over the state x_t. We will now discuss different belief representations and

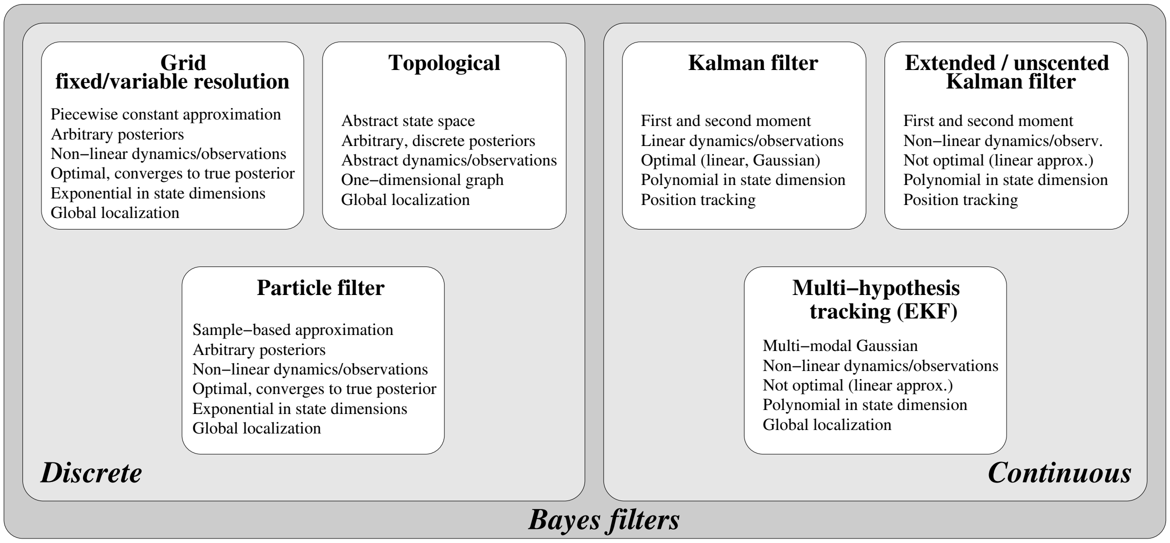


Fig. 1: Properties of the most common implementations of Bayes filters. All approaches are based on the Markov assumption underlying all Bayes filters.

their properties in the context of mobile robot localization. An overview of the different algorithms is given in Figure 1.

Kalman filters are the most widely used variant of Bayes filters [47, 32, 72, 80]. They approximate beliefs by their first and second moments, *i.e.* mean and covariance. Kalman filters are optimal under the assumptions that the initial state uncertainty is unimodal Gaussian and that the observation model and the system dynamics are linear in the state with Gaussian noise. Non-linearities are typically approximated by linearization at the current state, resulting in the extended Kalman filter (EKF). Recently, the unscented Kalman filter (UF) has been introduced [45, 79]. This approach deterministically generates samples (sigma-points) taken from the Gaussian state and passes these samples through the non-linear dynamics, followed by a Gaussian approximation of the predicted samples. The unscented filter has been shown to yield better approximations both in theory and in practice. Due to the linear approximations, both extended and unscented Kalman filters are not optimal.

Despite their restrictive assumptions, Kalman filters have been applied with great success to mobile robot localization, where they yield very efficient and accurate position estimates even for highly non-linear systems [55, 58, 2, 39]. One of the key advantages of Kalman filters is their efficiency. The complexity is polynomial in the dimensionality of the state space and the observations. Due to this graceful increase in complexity, Kalman filters can be applied to the simultaneous localization and map building problem (SLAM), where they estimate full posteriors over both robot positions and landmark positions (typically consisting of hundreds of dimensions) [15, 20, 56, 10, 17].

However, due to the assumption of unimodal posteriors, Kalman filters applied to robot localization solely aim at tracking a robot's location. They are not designed to globally localize a robot from scratch in arbitrarily large environments. Some of the limitations of Kalman filters in the context of robot localization have been shown experimentally in [37].

Multi-hypothesis approaches represent the belief state by mixtures of Gaussians [4, 68, 42, 1]. Each hypothesis, or Gaussian, is typically tracked by an extended Kalman filter. Due to their ability to represent multi-modal beliefs, these approaches are able to solve the global localization problem. Since each hypothesis is tracked using a Kalman filter, these methods still rely on the assumptions underlying the Kalman filters (apart from uni-modal posteriors). In practice, however, multi-hypothesis approaches have been shown to be very robust to violations of these assumptions. So far it is not clear how these methods can be applied to extremely non-linear observations, such as those used in [19]. In addition to pure Kalman filtering, multi-hypothesis approaches require sophisticated heuristics to solve the data association problem and to determine when to add or delete hypotheses [6, 14].

Topological approaches are based on symbolic, graph structured representations of the environment. The state space of the robot consists of a set of discrete, locally distinctive locations such as corners or crossings of hallways [46, 70, 52, 40, 11, 53]. The advantage of these approaches lies in their efficiency and in the fact that they can represent arbitrary distributions over the discrete state space. Therefore they can solve the global localization problem. Additionally, these approaches may scale well towards high dimensional state spaces because the complexity of the topological structure does not directly dependent on the dimensionality of the underlying state space. A key disadvantage, however, lies in the coarseness of the representation, due to which position estimates provide only rough information about the robot location. Furthermore, only sensor information related to the symbolic representation of the environment can be used, and adequate features might not be available in arbitrary environments.

Grid-based, metric approaches rely on discrete, piecewise constant representations of the belief [9, 8, 51, 63]. For indoor localization, the spatial resolution of these grids is usually between 10 and 40 cm and the angular resolution is usually 5 degrees. As the topological approaches, these methods can represent arbitrary distributions over the discrete state space and can solve the global localization problem [1]. In contrast to topological approaches, the metric approximations provide accurate position estimates in combination with high robustness to sensor noise. A grid-based method has been applied successfully for the position estimation of the museum tour-guide robots Rhino and Minerva [7, 73, 30]. A disadvantage of grid-based approaches lies in their computational complexity, based on the requirement to keep the typically three-dimensional position probability grid in memory and to update it for each new observation. Efficient sensor models [30, 51], selective update schemes [30], and adaptive, tree-based representations [8] greatly increase the efficiency of these methods, making them applicable to online robot localization. Since the complexity of these methods grows exponentially with the number of dimensions, it is doubtful whether they can be applied to higher-dimensional state spaces.

Sample-based approaches represent beliefs by sets of samples, or particles [28, 19, 31, 54, 43, 36]. A key advantage of particle filters is their ability to represent arbitrary probability densities, which is why they can solve the global localization problem. Furthermore,

[1] Topological and in particular grid-based implementations of Bayes filters for robot localization are often referred to as Markov localization [26].

particle filters can be shown to converge to the true posterior even in non-Gaussian, non-linear dynamic systems [18]. Compared to grid-based approaches, particle filters are very efficient since they focus their resources (particles) on regions in state space with high likelihood. Since the efficiency of particle filters strongly depends on the number of samples used for the filtering process, several attempts have been made to make more efficient use of the available samples [34, 66, 35]. Since the worst-case complexity of these methods grows exponentially in the dimensions of the state space, it is not clear how particle filters can be applied to arbitrary, high-dimensional estimation problems. If, however, the posterior focuses on small, lower-dimensional regions of the state space, particle filters can focus the samples on such regions, making them applicable to even high-dimensional state spaces. We will discuss the details of particle filters in the next section.

Mixed approaches use independences in the structure of the state space to break the state into lower-dimensional sub-spaces (random variables). Such structured representations are known under the name of dynamic Bayesian networks [33]. The individual sub-spaces can be represented using the most adequate representation, such as continuous densities, samples, or discrete values [50]. Recently, under the name of Rao-Blackwellised particle filters [21, 16, 36], the combination of particle filters with Kalman filters yielded extremely robust and efficient approaches to higher dimensional state estimation including full posteriors over robot positions and maps [62, 59].

2.3 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 $Bel(x_t)$:

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

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 [57, 23, 22]). A time update of the basic particle filter algorithm is outlined in Table 1.

At each iteration, the algorithm receives a sample set S_{t-1} representing the previous belief of the robot, a control measurement u_{t-1}, and an observation z_t. Steps 3–8 generate n samples representing the posterior belief: Step 4 determines which sample to draw from the previous set. In this resampling step, a sample index is drawn with probability proportional to the sample weight [2]. Once a sample index is drawn, the corresponding sample and the control information u_{t-1} are used to predict the next state $x_t^{(i)}$. This is done by sampling from the density $p(x_t \mid x_{t-1}, u_{t-1})$, which represents the system dynamics. Each $x_t^{(i)}$ corresponds to a sample drawn from the predictive belief $Bel^-(x_t)$ in (1). In order to generate samples according to the posterior belief $Bel(x_t)$, importance sampling is applied, with $Bel(x_t)$ as target distribution

[2]Resampling with minimal variance can be implemented efficiently (constant time per sample) using a procedure known under the name deterministic selection [48, 3] or stochastic universal sampling [5].

1. **Inputs:** $S_{t-1} = \{\langle x_{t-1}^{(i)}, w_{t-1}^{(i)} \rangle \mid i = 1, \ldots, n\}$ representing belief $Bel(x_{t-1})$, control measurement u_{t-1}, observation z_t
2. $S_t := \emptyset,\ \alpha := 0$ *// Initialize*
3. **for** $i := 1, \ldots, n$ **do**   *// Generate n samples*

   // Resampling: Draw state from previous belief
4. Sample an index j from the discrete distribution given by the weights in S_{t-1}

   // Sampling: Predict next state
5. Sample $x_t^{(i)}$ from $p(x_t \mid x_{t-1}, u_{t-1})$ conditioned on $x_{t-1}^{(j)}$ and u_{t-1}
6. $w_t^{(i)} := p(z_t \mid x_t^{(i)})$;   *// Compute importance weight*
7. $\alpha := \alpha + w_t^{(i)}$ *// Update normalization factor*
8. $S_t := S_t \cup \{\langle x_t^{(i)}, w_t^{(i)} \rangle\}$ *// Insert sample into sample set*
9. **for** $i := 1, \ldots, n$ **do**   *// Normalize importance weights*
10. $w_t^{(i)} := w_t^{(i)} / \alpha$
11. **return** S_t

Table 1: The basic particle filter algorithm.

and $p(x_t \mid x_{t-1}, u_{t-1})Bel(x_{t-1})$ as proposal distribution [71, 31]. By dividing these two distributions, we get $p(z_t \mid x_t^{(i)})$ as the importance weight for each sample (see Eq.(2)). Step 7 keeps track of the normalization factor, and Step 8 inserts the new sample into the sample set. After generating n samples, the weights are normalized so that they sum up to one (Steps 9-10). It can be shown that this procedure in fact implements the Bayes filter, using an (approximate) sample-based representation [23, 22]. Furthermore, the sample-based posterior converges to the true posterior at a rate of $1/\sqrt{n}$ as the number n of samples goes to infinity [18].

Particle filters for mobile robot localization

Figure 2 illustrates the particle filter algorithm using a one-dimensional robot localization example. For illustration purpose, the particle filter update is broken into two separate parts: one for robot motion (Steps 4-5), and one for observations (Steps 6-7). The robot is given a map of the hallway, but it does not know its position. Figure 2a shows the initial belief: a uniformly distributed sample set, which approximates a uniform distribution. Each sample has the same importance weight, as indicated by the equal heights of all bars in this figure. Now assume the robot detects the door to its left. The likelihood $p(z \mid x)$ for this observation is shown in the upper graph in Figure 2b. This likelihood is incorporated into the sample set by adjusting and then normalizing the importance factor of each sample, which leads to the sample set shown in the lower part of Figure 2b (Steps 6-10 in Table 1). These samples have the same states as before, but now their importance factors are proportional to $p(z \mid x)$. Next the robot moves to the right, receiving control information u. The particle filter algorithm now draws samples from the current, weighted sample set, and then randomly predicts the next location of the robot using the motion information u (Steps 4-5 in Table 1). The resulting sample set is shown in Figure 2c. Notice that this sample set differs from the original one in that the majority of sam-

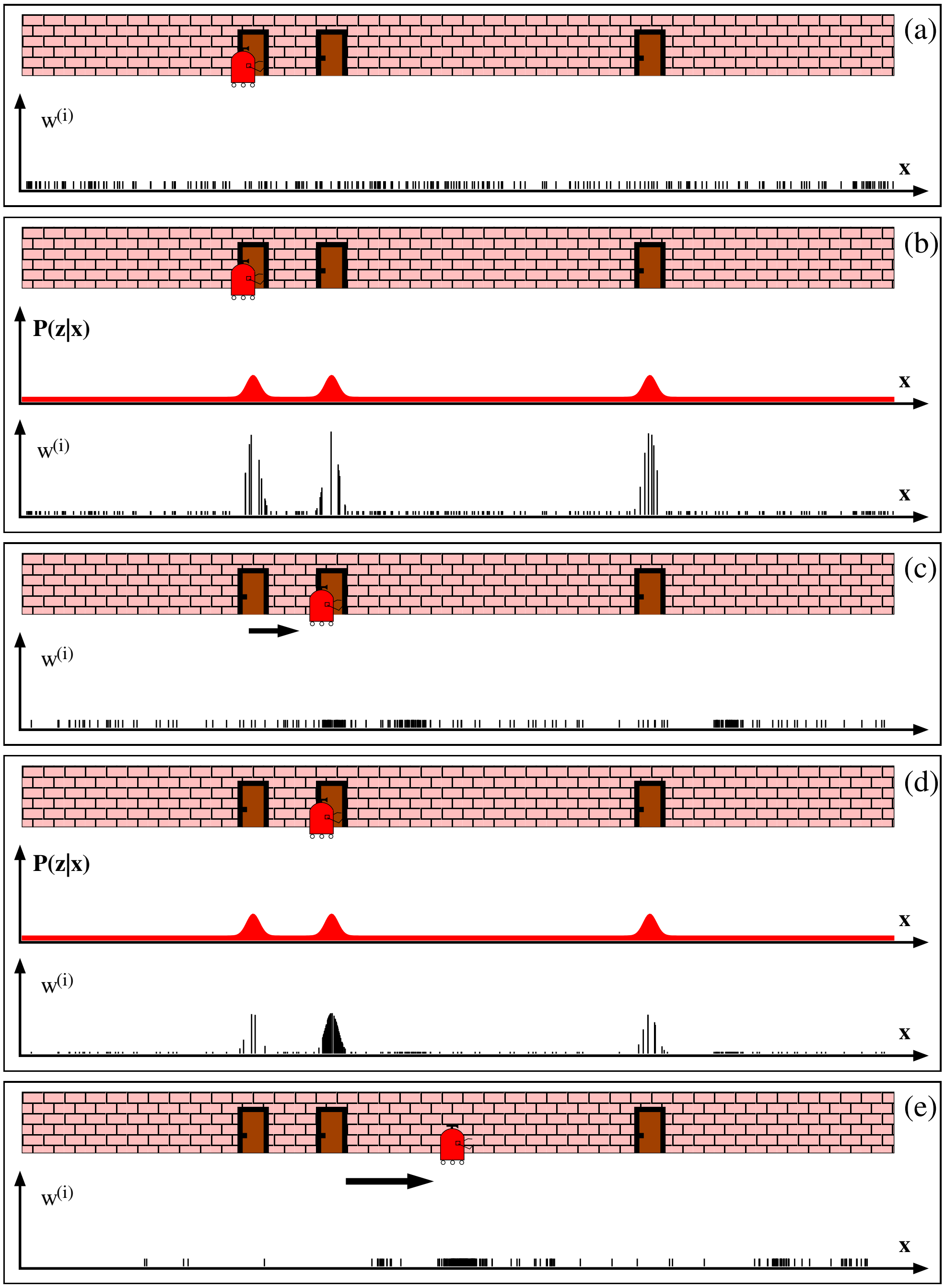


Figure 2: One-dimensional illustration of particle filters for mobile robot localization.

ples is centered around three locations. This concentration of the samples is achieved through resampling Step 4 (with a subsequent motion). The robot now senses a second door, leading to the probability $P(z \mid x)$ shown in the upper graph of Figure 2d. By weighting the importance factors in proportion to this probability, we obtain the sample set in Figure 2d. After the next robot motion, which includes a resampling step, most of the probability mass is consistent with the robot's true location.

In typical robot localization problems, the position of the robot is represented in the two-dimensional Cartesian space along with the robot's heading direction θ. Measurements z_t may

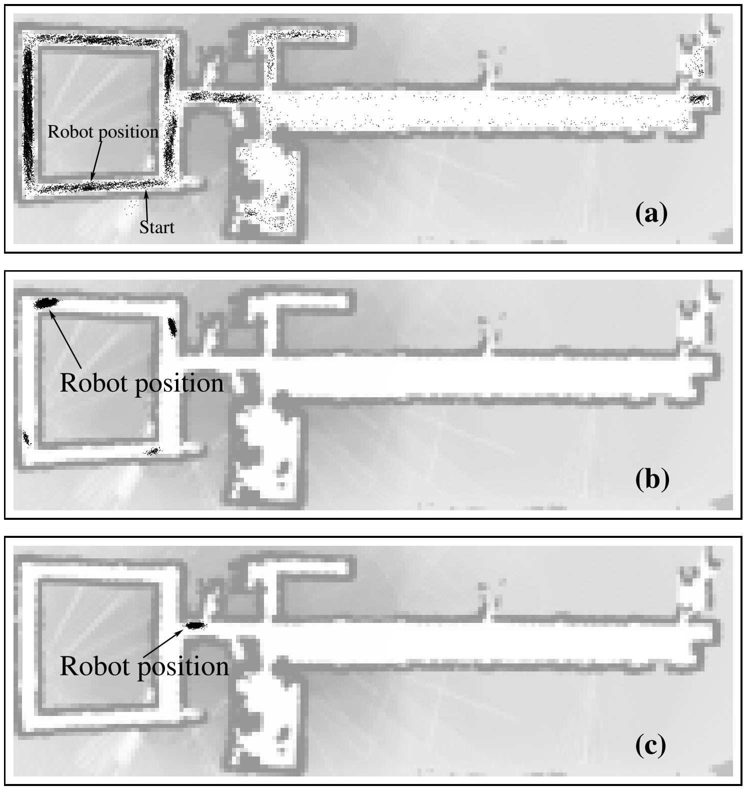


Fig. 3: Map of the UW CSE Department along with a series of sample sets representing the robot's belief during global localization using sonar sensors (samples are projected into 2D). The size of the environment is 54m $\times$ 18m. a) After moving 5m, the robot is still highly uncertain about its position and the samples are spread through 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.

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 white noise [28, 31]. The perceptual model $p(z_t \mid x_t)$ describes the likelihood of making the observation z_t given that the robot is at location x_t. Particle filters have been applied to a variety of robot platforms and sensors such as vision [19, 54, 24, 81, 78, 38] and proximity sensors [28, 31, 43].

Figure 3 illustrates the application of particle filters to mobile robot localization using sonar sensors. 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 might be necessary to accurately represent the belief during early stages of localization (cf. 3(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. 3(c)). Unfortunately, it is not straightforward how the number of samples can be adapted during the estimation process, and this problem has only rarely been addressed so far.

3 Adaptive Particle Filters with Variable Sample Set Sizes

The time complexity of one update of the particle filter algorithm is linear in the number of samples needed for the estimation. Therefore, several attempts have been made to make more effective use of the available samples, thereby allowing sample sets of reasonable size. One such method incorporates Markov chain Monte Carlo (MCMC) steps to improve the quality of the sample-based posterior approximation [34]. Another approach, auxiliary particle filters, applies a one-step lookahead to minimize the mismatch between the proposal and the target distribution, thereby minimizing the variability of the importance weights, which in turn determines the efficiency of the importance sampler [66]. Auxiliary particle filters have been applied recently to robot localization [78]. Along a similar line of reasoning, the injection of observation samples into the posterior can be very advantageous [54, 74, 31]. However, this approach requires the availability of a sensor model from which it is possible to efficiently generate samples.

In this paper we introduce an approach to increasing the efficiency of particle filters by *adapting the number of samples* to the underlying state uncertainty [27]. The localization example in Figure 3 illustrates when such an approach can be very beneficial. In the beginning of global localization, the robot is highly uncertain and a large number of samples is needed to accurately represent its belief (c.f. 3(a)). On the other extreme, once the robot knows where it is, only a small number of samples suffices to accurately track its position (c.f. 3(c)). However, with a fixed number of samples one has to choose large sample sets so as to allow a mobile robot to address both the global localization and the position tracking problem. Our approach, in contrast, adapts the number of samples during the localization process, thereby choosing large sample sets during global localization and small sample sets for position tracking. Before we introduce our method for adaptive particle filters, let us first discuss an existing technique to changing the number of samples during the filtering process [49, 28].

3.1 Likelihood-based adaptation

We call this approach likelihood-based adaptation since it determines the number of samples based on the likelihood of observations. More specifically, the approach generates samples until the sum of the non-normalized likelihoods exceeds a pre-specified threshold. This sum is equivalent to the normalization factor α updated in Step 7 of the particle filter algorithm (see Table 1). Likelihood-based adaptation has been applied to dynamic Bayesian networks [49] and mobile robot localization [28]. The intuition behind this approach is as follows: 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. 3(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 between the proposal distribution and the target distribution. Unfortunately, this mismatch is not always an accurate indicator for the necessary number of samples. Consider, for example, the ambiguous belief state consisting of four distinctive sample clusters shown in Fig. 3(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. 3(c)). Likelihood-based adaptation would hence use the same number of samples in both situations. Nevertheless, it is obvious that an accurate approximation of the belief shown in Fig. 3(b) requires a multiple of the samples needed to represent the belief in Fig. 3(c).

The likelihood-based approach has been shown to be superior to particle filters with fixed sample set sizes [49, 28]. However, the previous discussion makes clear that this approach does not fully exploit the potential of adapting the size of sample sets.

3.2 KLD-sampling

The key idea of our approach to adaptive particle filters can be stated as follows:

> *At each iteration of the particle filter, determine the number of samples such that, with probability $1-\delta$, the error between the true posterior and the sample-based approximation is less than ε.*

3.2.1 The KL-distance

To derive a bound on the approximation error, 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 [49, 61, 64, 75]. For such a representation we show how to determine the number of samples so that the distance between the sample-based maximum likelihood estimate (MLE) and the true posterior does not exceed a pre-specified threshold ε. We denote the resulting approach as the KLD-sampling algorithm since the distance between the MLE and the true distribution is measured by the Kullback-Leibler distance (KL-distance) [12]. The KL-distance is a measure of the difference between two probability distributions p and q:

$$K(p,q) = \sum_x p(x) log \frac{p(x)}{q(x)} \tag{3}$$

KL-distance is never negative and it is zero if and only if the two distributions are identical. It is not a metric, since it is not symmetric and does not obey the triangle property. Despite this fact, it is accepted as a standard measure for the difference between probability distributions (or densities).

In what follows, we will first determine the number of samples needed to achieve, with high probability, a good approximation of an arbitrary, discrete probability distribution (see also [67, 44]). 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.* $\underline{X} \sim \text{Multinomial}_k(n, \underline{p})$, where $\underline{p} = p_1 \ldots p_k$ specifies the true probability of each bin.

The maximum likelihood estimate of $\underline{p}$ using the n samples is given by $\underline{\widehat{p}} = n^{-1}\underline{X}$. Furthermore, the likelihood ratio statistic λ_n for testing $\underline{p}$ is

$$\log \lambda_n \;=\; \sum_{j=1}^{k} X_j \log\left(\frac{\widehat{p}_j}{p_j}\right). \tag{4}$$

Since X_j is identical to $n\widehat{p}_j$ we get

$$\log \lambda_n \;=\; n \sum_{j=1}^{k} \widehat{p}_j \log\left(\frac{\widehat{p}_j}{p_j}\right). \tag{5}$$

From (3) and (5) we can see that the likelihood ratio statistic is n times the KL-distance between the MLE and the true distribution:

$$\log \lambda_n \;=\; nK(\underline{\widehat{p}}, \underline{p}). \tag{6}$$

It can be shown that the likelihood ratio converges to a chi-square distribution with $k-1$ degrees of freedom [67]:

$$2 \log \lambda_n \rightarrow_d \chi^2_{k-1} \qquad \text{as} \qquad n \rightarrow \infty. \tag{7}$$

Now let $P_{\underline{p}}(K(\underline{\widehat{p}}, \underline{p}) \leq \varepsilon)$ denote the probability that the KL-distance between the true distribution and the sample-based MLE is less than or equal to ε (under the assumption that $\underline{p}$ is the true distribution). The relationship between this probability and the number of samples can be derived as follows:

$$\begin{aligned} P_{\underline{p}}(K(\underline{\widehat{p}}, \underline{p}) \leq \varepsilon) &= P_{\underline{p}}(2nK(\underline{\widehat{p}}, \underline{p}) \leq 2n\varepsilon) && (8)\\ &= P_{\underline{p}}(2 \log \lambda_n \leq 2n\varepsilon) && (9)\\ &\doteq P(\chi^2_{k-1} \leq 2n\varepsilon) && (10) \end{aligned}$$

(10) follows from (6) and the convergence result stated in (7). The quantiles of the chi-square distribution are given by

$$P(\chi^2_{k-1} \leq \chi^2_{k-1,1-\delta}) = 1 - \delta\,. \tag{11}$$

If we choose n such that $2n\varepsilon$ is equal to $\chi^2_{k-1,1-\delta}$, we can combine (10) and (11) and get

$$P_{\underline{p}}(K(\underline{\widehat{p}}, \underline{p}) \leq \varepsilon) \;\doteq\; 1 - \delta\,. \tag{12}$$

Now we have a clear relationship between the number of samples and the resulting approximation quality. To summarize, if we choose the number of samples n as

$$n = \frac{1}{2\varepsilon}\chi^2_{k-1,1-\delta}, \tag{13}$$

then we can guarantee that with probability $1-\delta$, the KL-distance between the MLE and the true distribution is less than ε (see (12)). In order to determine n according to (13), we need to compute the quantiles of the chi-square distribution. A good approximation is given by the Wilson-Hilferty transformation [44], which yields

$$n \;=\; \frac{1}{2\varepsilon}\chi^2_{k-1,1-\delta} \;\doteq\; \frac{k-1}{2\varepsilon}\left\{1-\frac{2}{9(k-1)}+\sqrt{\frac{2}{9(k-1)}}z_{1-\delta}\right\}^3 , \tag{14}$$

where $z_{1-\delta}$ is the upper $1-\delta$ quantile of the standard normal distribution. The values of $z_{1-\delta}$ for typical values of δ are readily available in standard statistical tables.

This concludes the derivation of the sample size needed to approximate a discrete distribution with an upper bound ε on the KL-distance. From (14) we see that the required number of samples is proportional to the inverse of the error bound ε, and to the first order linear in the number k of bins with support. Here we assume that a bin of the multinomial distribution has support if its probability is above a certain threshold (*i.e.* if it contains at least one particle) [3].

3.2.2 Using KL-distance in particle filters

It remains to be shown how to incorporate this result into the particle filter algorithm. The problem is that we do not know the true posterior distribution for which we can estimate the number of samples needed for a good approximation (the efficient estimation of this posterior is the main goal of the particle filter). Our solution to this problem is to rely on the sample-based representation of the *predictive belief* as an estimate for the posterior (samples from this belief are generated in step 5 of the basic particle filter algorithm shown in Table 1). Furthermore, (14) shows that it is not necessary to determine the complete discrete distribution, but that it suffices to determine the number k of bins with support (for given ε and δ). Even though we do not know this quantity before we actually generated all samples from the predictive distribution, we can estimate k by counting the number of bins with support *during sampling*.

An update step of the KLD-sampling particle filter is summarized in Table 2. As can be seen, we update the number of supported bins k for the predictive distribution after each sample generated in step 5. The determination of k can be done incrementally by checking for each generated sample whether it falls into an empty bin or not (steps 9-11). After each sample, we use (14) to update the number n_χ of samples required for our current estimate of k (step 12). This concurrent increase in number n of already generated samples and in the desired number n_χ of samples works as follows: In the early stages of sampling, k increases with almost every new sample since virtually all bins are empty. This increase in k results in an increase in the number of desired samples n_χ. However, over time, more and more bins are non-empty and n_χ increases only occasionally. Since n increases with each new sample, n will finally reach n_χ and sampling can be stopped (condition in step 14).

The implementation of this modified particle filter is straightforward. The difference to the original algorithm is that we have to keep track of the number k of supported bins and the number n_χ of desired samples (steps 9-12; for clarity we omitted the fact that n_χ can

[3] This way of determining the number of degrees of freedom of a multinomial distribution is a common statistical tool. In our approach, the key advantage of this approximation is that it results in an efficient implementation that does not even depend on a threshold itself (see next paragraph). We also implemented a version of the algorithm using the complexity of the state space to determine the number of samples. Complexity is measured by 2^H, where H is the entropy of the distribution. This approach does not depend on thresholding at all, but it does not have guaranteed approximation bounds and cannot be implemented as efficiently as the method described here. Furthermore, it does not yield noticeably different results.

Inputs: $S_{t-1} = \{\langle x_{t-1}^{(i)}, w_{t-1}^{(i)} \rangle \mid i = 1, \dots, n\}$ representing belief $Bel(x_{t-1})$,
control measurement u_{t-1}, observation z_t,
bounds ε and δ, bin size Δ, minimum number of samples $n_{\chi_{min}}$

$S_t := \emptyset$, $n = 0$, $n_\chi = 0$, $k = 0$, $\alpha = 0$ *// Initialize*

do *// Generate samples . . .*

// Resampling: Draw state from previous belief
Sample an index j from the discrete distribution given by the weights in S_{t-1}

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

$w_t^{(n)} := p(z_t \mid x_t^{(n)})$; *// Compute importance weight*
$\alpha := \alpha + w_t^{(n)}$ *// Update normalization factor*
$S_t := S_t \cup \{\langle x_t^{(n)}, w_t^{(n)} \rangle\}$ *// Insert sample into sample set*

if $(x_t^{(n)}$ falls into empty bin $b)$ **then**
$k := k + 1$ *// Update number of bins with support*
$b :=$ non-empty *// Mark bin*
$n_\chi := \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta} \right\}^3$ *// Update number of desired samples*

$n := n + 1$ *// Update number of generated samples*

while $(n < n_\chi)$ *// . . . until KL-bound is reached*

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

return S_t

Table 2: KLD-sampling algorithm.

be determined only when $k > 1$). The bins can be implemented either as a fixed, multi-dimensional grid, or more compactly as tree structures [49, 61, 64, 75, 29]. Note that the sampling process is guaranteed to terminate, since for a given bin size Δ, the maximum number k of bins is limited, which also limits the maximum number n_χ of desired samples.

To summarize, our approach adapts the number of samples based on the approximation error introduced by the sample-based representation. It uses the predictive belief state as an estimate of the underlying posterior. Therefore, it is not guaranteed that our approach does not diverge from the true (unknown) belief. However, as our experiments show, divergence only occurs when the error bounds are too loose. KLD-sampling is easy to implement and the determination of the sample set size can be done without noticeable loss in processing speed (at least in our low-dimensional robot localization context). Furthermore, our approach can be used in combination with any scheme for improving the approximation of the posterior [34, 66, 35, 74].

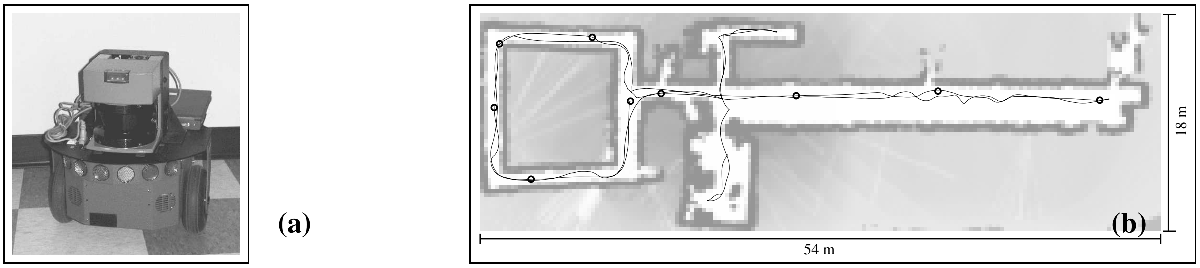


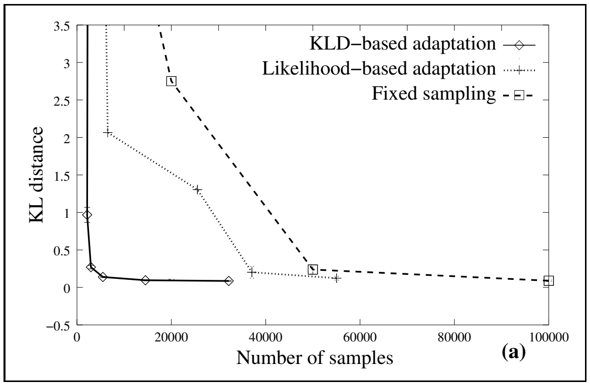
Fig. 4: a) Pioneer II robot used throughout the experiments. b) Map used for localization along with the path followed by the robot during data collection. The small circles mark the different start points for the global localization experiments.

4 Experimental Results

We evaluated KLD-sampling in the context of indoor mobile robot localization using data collected with a Pioneer robot (see Figure 4). The data consists of a sequence of sonar and laser range-finder scans along with odometry measurements annotated with time-stamps to allow systematic real-time evaluations. We used a beam-based sensor model to compute the likelihood of the sensor scans [30, 31]. In the first 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 threshold used to determine the number of samples, and for our approach we varied ε, the bound on the KL-distance. In all experiments, we used a value of 0.99 for $(1-\delta)$ and a fixed bin size Δ of 50cm $\times$ 50cm $\times$ 10deg. We limited the maximum number of samples for all approaches to 100,000. The influence of different parameter settings on the performance of KLD-sampling is discussed in more detail in Section 4.4.

4.1 Approximation of the true posterior

In the first set of experiments we evaluated how accurately the different methods approximate the true posterior density. Since the ground truth for these posteriors is not available, we generated reference sample sets using a particle filter with a fixed number of 200,000 samples (far more than actually needed for position estimation). At each iteration of our test algorithms, we computed the KL-distance between the current sample sets and the corresponding reference sets, using histograms for both sets. We ignored the time-stamps in these experiments and gave each algorithm as much time as needed to process the data. Fig. 5(a) plots the average KL-distance along with 95% confidence intervals against the average number of samples for the different algorithms and parameter settings (for clarity, we omitted the large error bars for KL-distances above 1.0). The different data points for KLD-sampling were obtained by varying the error bound ε between 0.4 and 0.015. Each data point in the graph represents the average of 16 global localization runs with different start positions of the robot (each of these runs itself represents the average of approximately 150 sample set comparisons at the different points in time). As expected, the more samples are used by the different approaches, the better the approximation. The curves clearly illustrate the superior performance of our approach: While the fixed approach requires about 50,000 samples before it converges to a KL-distance below 0.25, our approach converges to the same level using only 3,000 samples on average. This is



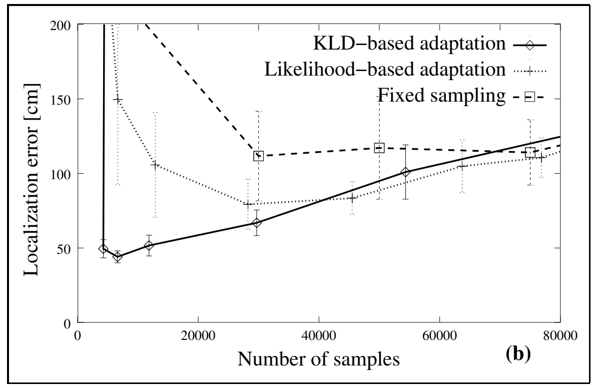


Fig. 5: The x-axis represents the average sample set size for different parameters of the three approaches. a) The y-axis plots the KL-distance between the reference densities and the sample sets generated by the different approaches (real-time constraints were not considered in this experiment). b) The y-axis represents the average localization error measured by the distance between estimated positions and reference positions. The U-shape in b) is due to the fact that under real-time conditions, an increasing number of samples results in higher update times and therefore loss of sensor data.

also an improvement by a factor of 12 compared to the approximately 36,000 samples needed by the likelihood-based approach. The graph also shows that our approach is not guaranteed to accurately track the true belief. This is due to the fact that the error bound is computed relative to the current estimate of the belief, not the true posterior. Therefore, our approach can diverge if the error bound is too loose (see leftmost data point in the KLD-sampling graph). However, these experiments indicate that our approach, even though based on several approximations, is able to accurately track the true posterior using far smaller sample sets on average than the other approaches.

4.2 Real-time performance

Due to the computational overhead for determining the number of samples, it is not clear that our approach yields better results under real-time conditions. To test the performance of our approach under realistic conditions, we performed multiple global localization experiments under real-time considerations using the timestamps in the data sets. As in the previous experiment, localization was based on the robot's sonar sensors. Again, the different average numbers of samples for KLD-sampling were obtained by varying the ε-bound. The minimum and maximum numbers of samples correspond to ε-bounds of 0.4 and 0.015, respectively. After each update of the particle filter we determined the distance between the estimated robot position and the corresponding reference position [4]. The results are shown in Fig. 5(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 can take several seconds and valuable sensor data has to be discarded, which results in less accurate position estimates. Fig. 5(b) also shows that even under real-time conditions, our KLD-sampling approach yields drastic improvements over both fixed sampling and likelihood-based sampling. The smallest

[4] Position estimates are extracted from sample sets using histograming and local averaging, and the reference positions were determined by evaluating the robot's highly accurate laser range-finder information.

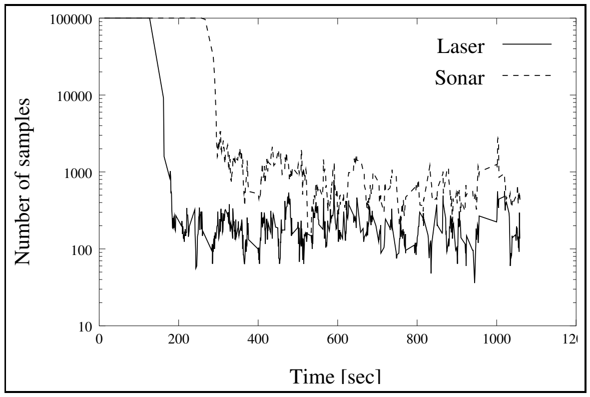


Fig. 6: Typical evolution of number of samples for a global localization run, plotted against time (number of samples is shown on a log scale). The solid line shows the number of samples when using the robot's laser range-finder, the dashed graph is based on sonar sensor data.

average localization error is 44cm in contrast to a minimal average error of 79cm and 114cm for the likelihood-based and the fixed approach, respectively. This result is due to the fact that our approach is able to determine the best mix between more samples during early stages of localization and less samples during position tracking. Due to the smaller sample sets, our approach also needs significantly less processing power than any of the other approaches. Note that the average errors are relatively high since they include the large errors occurring during the beginning of each localization run.

Figure 6 shows the sample set sizes during a typical global localization run using KLD-sampling. In addition to the sizes based on the sonar data used in the previous experiments (dashed line), the graph also shows the numbers of samples resulting from data collected by the robot's laser range-finder (solid line). As expected, the higher accuracy of the laser range-finder results in a faster drop in the number of samples and in smaller sample sets during the final tracking phase of the experiment. Note that the same parameters were used for both runs. Extensions 1 and 2 illustrate global localization using sonar and laser range-finders. Shown there are animations of sample sets during localization, annotated with the number of samples used at the different points in time. Both runs start with 40,000 samples. The robot is plotted at the position estimated from the sample sets. The timing of the animations is proportional to the approximate update times for the particle filter (real updates are more than two times faster).

4.3 Tracking performance

So far we only showed that KLD-sampling is beneficial if the state uncertainty is extremely high in the beginning and then constantly decreases to a certain level. In this experiment we test whether KLD-sampling can produce improvements even for tracking, when the initial state of the system is known. Again, we used the data collected by the Pioneer robot shown in Figure 4. This time we used the robot's laser range-finder data. In each run, the particle filter was initialized with a Gaussian distribution centered at the start location of the robot. Our initial experiments showed that in this context KLD-sampling has no significant advantage over fixed sample sets. This result is not surprising since the uncertainty does not vary much during position tracking. However, if we increase the difficulty of the tracking task, our approach can improve the tracking ability. Here, the difficulty was increased by adding random noise to the robot's odometry measurements (30%) and by modeling temporary loss of sensor data.

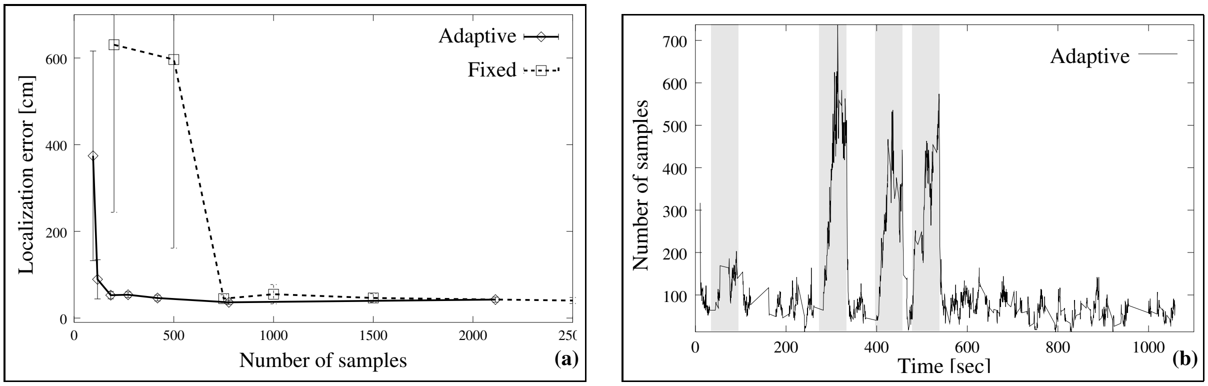


Fig. 7: (a) Localization error for different average sample set sizes. The solid line shows the error using KLD-sampling, and the dashed line illustrates the results using fixed sample set sizes. (b) Number of samples during one of the KLD-sampling runs. The grey shaded areas indicate time intervals of sensor loss, *i.e.*during which no laser data was available.

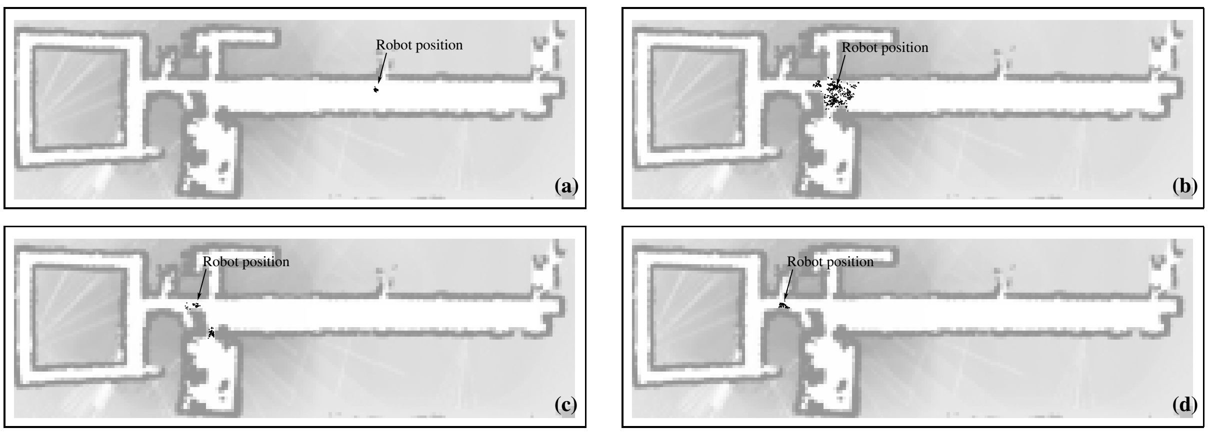


Fig. 8: Sample sets before, during and after an interval of sensor loss. (a) During normal tracking, the robot is highly certain and only 64 samples are used. (b) At the end of the sensor loss interval, the robot is uncertain and 561 samples are used to represent the belief. (c) Even after receiving sensor data again, the robot has to keep track of multiple hypotheses (129 samples). (d) The position uncertainty is low and the number of samples is reduced accordingly (66 samples).

This was done by randomly choosing intervals of 30 seconds during which all laser scans were deleted. Figure 7(a) illustrates the localization error for different sample set sizes for both KLD-sampling and fixed sample set sizes [5]. Again, our approach is highly superior to the fixed sampling scheme. With our approach, good tracking results (average error of 52.8 ± 8.5cm) can be achieved with only 184 samples on average, while the fixed approach requires 750 samples to achieve comparable accuracy. Note that the relatively large errors are due to the intervals of sensor loss.

Figure 7(b) illustrates why our approach performs better than particle filters with fixed sample sets. The figure plots the number of samples during one of the runs using KLD-sampling. The grey shaded areas indicate time intervals during which the laser data was removed from the script. The graph shows that our approach automatically adjusts the number of samples

[5]We did not compare KLD-sampling to the likelihood-based approach, since this method is not able to adequately adapt to situations of sensor loss. This is due to the fact that, without integrating sensor data, the likelihoods are always the same.

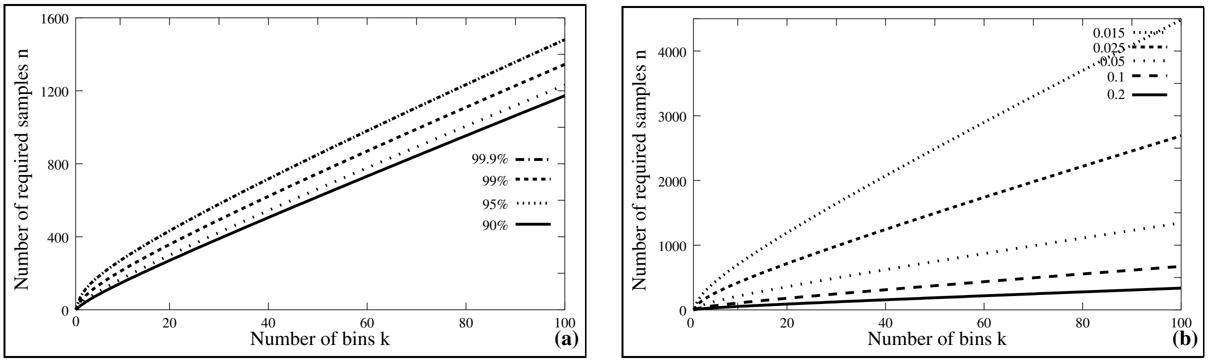


Fig. 9: Number of required samples n versus number of bins k for different settings of ε and δ. (a) ε was fixed to 0.05 and the graphs show Eq. (14) for different values of $(1-\delta)$. (b) $(1-\delta)$ was fixed to 95% and ε was varied.

to the availability of sensor data. While less than 100 samples are used during normal tracking, the number of samples automatically increases whenever no laser data is available. This increased number of samples is needed to keep track of the increased estimation uncertainty due to sensor loss. Figure 8 shows a sequence of sample sets (a) before, (b) during and (c,d) after one of the intervals without laser data. It is clear that distributions during the sensor loss interval (Figure 8(b)) require more samples than the distributions during normal tracking (Figure 8(a) and (d)). Furthermore, Figure 8(c) shows that it is essential to have enough samples to represent ambiguous situations occurring due to increased uncertainty.

4.4 Parameter settings

This section discusses the influence of the different parameters on the performance of KLD-sampling. Our approach is independent of the parameters used for the sensor and motion models, *i.e.* these values do not have to be changed when using KLD-sampling. The key parameters of the approach are the error bound ε, the probability bound δ, and the the bin size Δ. In the experiments so far, the bound δ and the bin size Δ were fixed, and only the error bound ε was changed. For a given number of non-empty bins k, the number of samples n is computed using (14). Figure 9 illustrates this function for different combinations of ε and δ. The graphs are shown for up to 100 bins only. For larger values the individual graphs continue in almost perfectly straight lines. As can be seen, the number of required samples changes less significantly with different settings of δ than with different values of ε (see also Eq. (14)). Hence it is reasonable to fix δ and adjust ε so as to achieve good performance.

Figure 10 shows how KLD-sampling changes with the bin size Δ. These results were obtained in the context of the global localization experiment described in Section 4.1. For each bin size, $(1-\delta)$ was fixed to 0.99 and ε was varied so as to achieve different sample set sizes. Figure 10(a) shows the average number of samples for bin sizes ranging from 25cm$\times$25cm$\times$5deg to 200cm$\times$200cm$\times$40deg. For fixed values of δ and ε, the number of samples increases with smaller bin sizes. The difference is most prominent for small values of the tolerated approximation error ε. This result is not surprising, since smaller bin sizes result in larger values of non-empty bins k, thereby increasing the number n of required samples.

Figure 10(b) shows the localization error plotted versus average number of samples for different bin sizes. Again, the number of samples was varied by changing the value of ε. In this experiment we found no significant difference between the performances when using

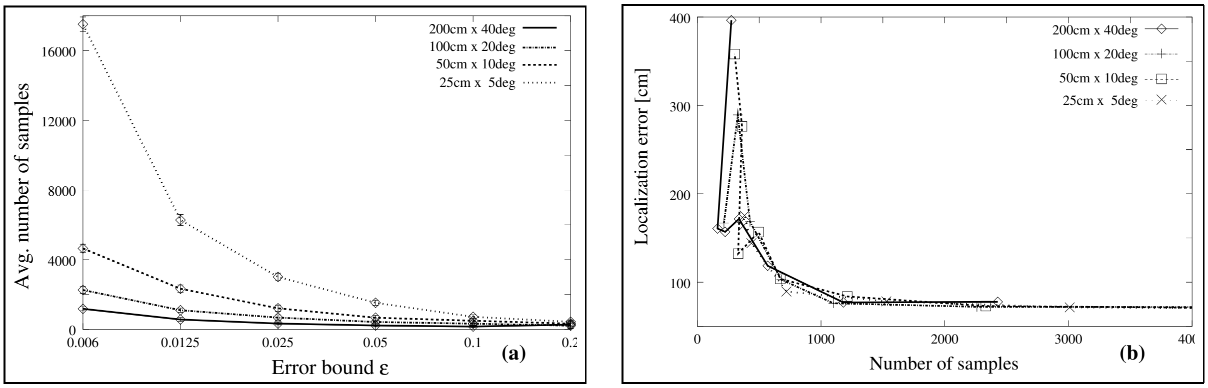


Fig. 10: Dependency on bin size. (a) The x-axis represents different values for the error bound ε. The y-axis plots the resulting average number of samples for different bin sizes. (b) The x-axis shows the average number of samples and the y-axis plots the localization error achieved with the corresponding number of samples and bin size. Obviously, all bin sizes achieve comparable results.

different bin sizes. When using approximately 1,200 samples on average, all bin sizes achieved the same, low level of localization error. The only difference lies in the value of ε required to achieve a certain average number of samples. More specifically, the same quality can be achieved when using large bin sizes in combination with small approximation bound ε, or when using small bin sizes in combination with larger approximation bound ε. This result was slightly surprising since we expected that smaller bin sizes result in better overall performance. However, the result is also very encouraging since it shows that the performance of KLD-sampling is very robust to changes in its parameters. In our experience, a good approach was to fix two parameters to reasonable values ($e.g. (1-\delta) = 0.99$ and $\Delta = 50cm \times 50cm \times 10deg$), and to adjust the remaining parameter so as to get the desired results.

5 Conclusions and Future Research

We presented a statistical approach to adapting the sample set sizes of particle filters during the estimation process. The key idea of the KLD-sampling approach is to bound the error introduced by the sample-based belief representation. At each iteration, our approach generates samples until their number is large enough to guarantee that the KL-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 numbers of samples if the samples have to cover a major part of the state space. KLD-sampling is most advantageous when the complexity of the posterior changes drastically over time, as is the case, for example, in global robot localization and position tracking with sensor loss.

Both the implementational and computational overhead of this approach are small. Extensive experiments in the context of mobile robot localization show that our approach yields drastic improvements over particle filters with fixed sample sets and over a previously introduced adaptation approach [49, 28]. In our experiments, KLD-sampling yields better approximations using only 6% of the samples required by the fixed approach, and using less than 9% of the samples required by the likelihood adaptation approach. In addition to the experiments presented in this paper, KLD-sampling has been tested in various indoor environments including a museum with wide open spaces. In all environments, the results were comparable to the ones presented here.

So far, KLD-sampling has been tested using robot localization only. We conjecture, however, that many other applications of particle filters can benefit from this method. In general, our approach achieves maximal improvements in lower-dimensional state spaces and when state uncertainties vary over time. In high-dimensional state spaces the bin counting can be implemented using tree structures [49, 61, 64, 75, 29]. In this case the additional cost of KLD-sampling is higher, since each tree lookup takes time logarithmic in the size of the state space (instead of the constant time for the grid we used in the experiments). Furthermore, our approach can be combined with any other method for improving the efficiency of particle filters [34, 66, 35, 74].

KLD-sampling opens several directions for future research. In our current implementation we use a discrete distribution with a *fixed* bin size to determine the number of samples. We assume that the performance of the filter can be further improved by changing the discretization over time, using coarse discretizations when the uncertainty is high, and fine discretizations when the uncertainty is low. Our approach can also be extended to the case where in certain parts of the state space highly accurate estimates are needed, while in other parts a rather crude approximation is sufficient. This problem can be addressed by locally adapting the discretization to the desired approximation quality using multi-resolution tree structures [49, 61] in combination with stratified sampling. As a result, more samples are used in "important" parts of the state space, while less samples are used in other parts.

5.0.1 Acknowledgments

This research is sponsored in part by the National Science Foundation (CAREER grant number 0093406) and by DARPA's MICA and SDR programme (contract numbers F30602-98-2-0137 and NBCHC020073). 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 valuable discussions.

References

[1] K.O. Arras, J.A. Castellanos, and R. Siegwart. Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2002.

[2] K.O. Arras and S.J. Vestli. Hybrid, high-precision localization for the mail distributing mobile robot system MOPS. In *Proc. of the IEEE International Conference on Robotics & Automation*, 1998.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear / non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, XX, 2001.

[4] D. Austin and P. Jensfelt. Using multiple Gaussian hypotheses to represent probability distributions for mobile robot localization. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2000.

[5] J.E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proc. of the Second International Conference on Genetic Algorithms.*, 1987.

[6] Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995.

[7] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.

[8] W. Burgard, A. Derr, D. Fox, and A. B. Cremers. Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[9] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the National Conference on Artificial Intelligence*, 1996.

[10] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós. The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5), 1999.

[11] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2), 2001.

[12] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley, New York, 1991.

[13] I. J. Cox. Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.

[14] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[15] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence*, 66:311–344, 1994.

[16] N. de Freitas. Rao-Blackwellised particle filtering for fault diagnosis. *IEEE Aerospace,*, 2002.

[17] M.C. Deans. Maximally informative statistics for localization and mapping. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2002.

[18] P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non linear filtering. In *Seminaire de Probabilites XXXIV*, number 1729 in Lecture Notes in Mathematics. Springer-Verlag, 2000.

[19] 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.

[20] M.W.M. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3), 2001.

[21] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

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

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

[24] 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.

[25] S. Engelson and D. McDermott. Error correction in mobile robot map learning. In *Proc. of the IEEE International Conference on Robotics & Automation*, 1992.

[26] D. Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Naviagation*. PhD thesis, Dept. of Computer Science, University of Bonn, Germany, December 1998.

[27] D. Fox. KLD-sampling: Adaptive particle filters. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS)*, Cambridge, MA, 2002. MIT Press.

[28] 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*, 1999.

[29] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.

[30] 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.

[31] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In Doucet et al. [22].

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

[33] Z. Ghahramani. An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), 2001.

[34] W.R. Gilks and C. Berzuini. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 61(1), 2001.

[35] S. Godsill and T. Clapp. Improvement strategies for Monte Carlo particle filters. In Doucet et al. [22].

[36] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 2002.

[37] J.S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[38] J.S. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[39] J.S. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.

[40] J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proc. of the First Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

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

[42] P. Jensfelt and S. Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760, October 2001.

[43] 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*, 2000.

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

[45] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[46] L. P. Kaelbling, A. R. Cassandra, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[47] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME, Journal of basic engineering*, 82:35–45, March 1960.

[48] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1), 1996.

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

[50] D. Koller and U. Lerner. Sampling in factored dynamic systems. In Doucet et al. [22].

[51] K. Konolige and K. Chou. Markov localization using correlation. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[52] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[53] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proc. of the National Conference on Artificial Intelligence*, 2002.

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

[55] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

[56] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proc. of the Ninth International Symposium on Robotics Research*, 1999.

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

[58] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18, 1997.

[59] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. of the National Conference on Artificial Intelligence*, 2002.

[60] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2002.

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

[62] K. Murphy and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In Doucet et al. [22].

[63] C.F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1), 2000.

[64] S. M. Omohundro. Bumptrees for efficient function, constraint, and classification learning. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*. Morgan Kaufmann, 1991.

[65] M. Pelikan, D.E. Goldberg, and E. Cant-Paz. Bayesian optimization algorithm, population size, and time to convergence. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, 2000.

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

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

[68] S.I. Roumeliotis and G.A. Bekey. Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2000.

[69] 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*, 2001.

[70] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[71] A. F. M Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *American Statistician*, 46(2):84–88, 1992.

[72] 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.

[73] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research (IJRR)*, 19(11), 2000.

[74] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2), 2001.

[75] 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*, 1999.

[76] V. Verma, J. Langford, and R. Simmons. Non-parametric fault identification for space rovers. In *International Symposium on Artificial Intelligence and Robotics in Space (iSAIRAS)*, 2001.

[77] J. Vermaak, C. Andrieu, A. Doucet, and S.J. Godsill. Particle methods for Bayesian modelling and enhancement of speech signals. *IEEE Transactions on Speech and Audio Processing*, 2002. Accepted for publication.

[78] N. Vlassis, B. Terwijn, and B. Kröse. Auxiliary particle filter robot localization from high-dimensional sensor observations. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2002.

[79] E.A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proc. of Symposium 2000 on Adaptive Systems for Signal Processing, Communications, and Control*, 2000.

[80] G. Welch and G. Bishop. An introduction to the Kalman filter. Notes of ACM SIGGRAPH tutorial on the Kalman filter, 2001.

[81] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2002.