Monte Carlo Localization With Mixture Proposal Distribution

Sebastian Thrun **Dieter Fox**
School of Computer Science
Carnegie Mellon University
http://www.cs.cmu.edu/$\sim${thrun,dfox}

Wolfram Burgard
Computer Science Department
University of Freiburg, Germany
http://www.informatik.uni-freiburg.de/$\sim$burgard

Abstract

Monte Carlo localization (MCL) is a Bayesian algorithm for mobile robot localization based on particle filters, which has enjoyed great practical success. This paper points out a limitation of MCL which is counter-intuitive, namely that *better sensors can yield worse results*. An analysis of this problem leads to the formulation of a new proposal distribution for the Monte Carlo sampling step. Extensive experimental results with physical robots suggest that the new algorithm is significantly more robust and accurate than plain MCL. Obviously, these results transcend beyond mobile robot localization and apply to a range of particle filter applications.

Introduction

Monte Carlo Localization (MCL) is a probabilistic algorithm for mobile robot localization that uses samples (particles) for representing probability densities. MCL is a version of *particle filters* [4, 10, 12, 15]. In computer vision, particle filters are known under the name *condensation algorithm* [9]. They have been applied with great practical success to visual tracking problems [9, 2] and mobile robot localization [3, 6, 11].

The basic idea of MCL is to approximate probability distributions by sets of samples. When applied to the problem of state estimation in a partially observable dynamical system, MCL successively calculates weighted sets of samples that approximate the posterior probability over the current state. Its practical success stems from the fact that it is non-parametric, hence can represent a wide range of probability distributions. It is also computationally efficient, and it is easily implemented as an *any-time* algorithm, which adapts the computational load by varying the number of samples in the estimation process [6].

This paper proposes a modified version of MCL, which uses a different sampling mechanism. Our study begins with the characterization of a key limitation of MCL (and particle filters in general). While MCL works well with *noisy* sensors, they fail catastrophically when the sensors are too accurate. This effect is undesirable: Ideally, the accuracy of any sound statistical estimator should *increase* with the accuracy of the sensors.

An analysis of this effect leads to the formulation of a new sampling mechanism (i.e., the proposal distribution), which changes the way samples are generated in MCL. We propose three different ways of computing the importance factors for this new proposal distribution. Our approach, which can be viewed as the natural dual to MCL, works well in cases where conventional MCL fails (and vice versa). To gain the best of both worlds, the conventional and our new proposal distribution are mixed together, leading to a new MCL algorithm with a mixture proposal distribution that is extremely robust.

Empirical results illustrate that the new mixture proposal distribution does not suffer the same limitation as MCL, and yields uniformly superior results. For example, our new approach with 50 samples consistently outperforms standard MCL with 1,000 samples. Additional experiments illustrate that our approach yields much better solutions in challenging variants of the localization problem, such as the *kidnapped robot problem* [5]. These experiments have been carried out both in simulation and with data collected from physical robots, using both laser range data and camera images for localization.

Our approach generalizes a range of previous extensions of MCL that have been proposed to alleviate these problems. Existing methods include the addition of *random* samples into the posterior [6], the generation of samples at locations that are consistent with the sensor readings [11], or the use of sensor models that assume an artificially high noise level [6]. While these approaches have shown superior performance over strict MCL in certain settings, they all lack mathematical rigor. In particular, neither of them approximates the true posterior, and over time they may diverge arbitrarily. Viewed differently, our approach can be seen as a theory that leads to an algorithm related to the ones above (with important differences), but also establishes a mathematical framework that is guaranteed to work in the limit.

The paper first reviews Bayes filters, the basic mathematical framework, followed by a derivation of MCL. Based on experiments characterizing the problems with plain MCL, we then derive dual MCL. Finally, the mixture proposal distribution is obtained by combining MCL and its dual. Empirical results are provided that illustrate the superior performance of our new extension of MCL.

Bayes Filtering

Bayes filters address the problem of estimating the state x of a dynamical system (partially observable Markov chain) from sensor measurements. For example, in mobile robot localization, the dynamical system is a mobile robot and its environment, the state is the robot's pose therein (often specified by a position in a Cartesian x-y space and the robot's heading direction θ). Measurements may include range measurements, camera images, and odometry readings. Bayes filters assume that the environment is *Markov*, that is, past

Figure 1: Global localization of a mobile robot using MCL (10,000 samples).

and future data are (conditionally) independent if one knows the current state.

The key idea of Bayes filtering is to estimate a probability density over the state space conditioned on the data. This posterior is typically called the *belief* and is denoted

$$Bel(x^{(t)}) \quad = \quad p(x^{(t)}|d^{(0\ldots t)})$$

Here x denotes the state, $x^{(t)}$ is the state at time t, and $d^{(0\ldots t)}$ denotes the data starting at time 0 up to time t. For mobile robots, we distinguish two types of data: *perceptual data* such as laser range measurements, and *odometry data* or *controls*, which carries information about robot motion. Denoting the former by o (for *observation*) and the latter by a (for *action*), we have

$$Bel(x^{(t)}) = p(x^{(t)}|o^{(t)}, a^{(t-1)}, o^{(t-1)}, a^{(t-2)} \ldots, o^{(0)}) \quad (1)$$

Without loss of generality, we assume that observations and actions arrive in an alternating sequence.

Bayes filters estimate the belief *recursively*. The *initial belief* characterizes the initial knowledge about the system state. In the absence of such, it is typically initialized by a *uniform distribution* over the state space. In mobile robotics, the state estimation without initial knowledge is called the *global localization problem*—which will be the focus throughout much of this paper.

To derive a recursive update equation, we observe that Expression (1) can be transformed by Bayes rule to

$$\frac{p(o^{(t)}|x^{(t)}, a^{(t-1)}, \ldots, o^{(0)})\ p(x^{(t)}|a^{(t-1)}, \ldots, o^{(0)})}{p(o^{(t)}|a^{(t-1)}, \ldots, o^{(0)})}$$

Under our Markov assumption, $p(o^{(t)}|x^{(t)}, a^{(t-1)}, \ldots, o^{(0)})$ can be simplified to $p(o^{(t)}|x^{(t)})$, hence we have

$$\frac{p(o^{(t)}|x^{(t)})\ p(x^{(t)}|a^{(t-1)}, \ldots, o^{(0)})}{p(o^{(t)}|a^{(t-1)}, \ldots, o^{(0)})}$$

We will now expand the rightmost term in the denominator by integrating over the state at time $t-1$

$$\frac{p(o^{(t)}|x^{(t)})}{p(o^{(t)}|a^{(t-1)}, \ldots, o^{(0)})} \int p(x^{(t)}|x^{(t-1)}, a^{(t-1)}, \ldots, o^{(0)}) \; p(x^{(t-1)}|a^{(t-1)}, \ldots, o^{(0)})\ dx^{(t-1)}$$

Again, we can exploit the Markov assumption to simplify $p(x^{(t)}|x^{(t-1)}, a^{(t-1)}, \ldots, o^{(0)})$ to $p(x^{(t)}|x^{(t-1)}, a^{(t-1)})$. Using the definition of the belief Bel, we obtain the important recursive equation

$$\begin{aligned} Bel(x^{(t)}) \quad &= \quad \frac{p(o^{(t)}|x^{(t)})}{p(o^{(t)}|a^{(t-1)}, \ldots, o^{(0)})} \qquad (2) \\ &\int p(x^{(t)}|x^{(t-1)}, a^{(t-1)})\ Bel(x^{(t-1)})\ dx^{(t-1)} \\ &= \eta p(o^{(t)}|x^{(t)}) \int p(x^{(t)}|x^{(t-1)}, a^{(t-1)}) Bel(x^{(t-1)}) dx^{(t-1)} \end{aligned}$$

where η is a normalization constant. This equation is of central importance, as it is the basis for various MCL algorithms studied here.

We notice that to implement (2), one needs to know three distributions: the initial belief $Bel(x^{(0)})$ (e.g., uniform), the next state probabilities $p(x^{(t)}|x^{(t-1)}, a^{(t-1)})$, and the perceptual likelihood $p(o^{(t)}|x^{(t)})$. MCL employs specific next state probabilities $p(x^{(t)}|x^{(t-1)}, a^{(t-1)})$ and perceptual likelihood models $p(o^{(t)}|x^{(t)})$ that describe robot motion and perception probabilistically. Such models are described in detail elsewhere [7].

Monte Carlo Localization

The idea of MCL (and other particle filter algorithms) is to represent the belief $Bel(x)$ by a set of m weighted samples distributed according to $Bel(x)$:

$$Bel(x) \quad = \quad \{x_i, w_i\}_{i=1,\ldots,m}$$

Here each x_i is a sample (a state), and w_i is a non-negative numerical factor (weight) called *importance factors*, which sums up to one over all i.

In global mobile robot localization, the *initial* belief is a set of poses drawn according to a uniform distribution over the robot's universe, and annotated by the uniform importance factor $\frac{1}{m}$. The recursive update is realized in three steps.

1. Sample $x_i^{(t-1)} \sim Bel(x^{(t-1)})$ using importance sampling from the (weighted) sample set representing $Bel(x^{(t-1)})$.
2. Sample $x_i^{(t)} \sim p(x^{(t)}|x_i^{(t-1)}, a^{(t-1)})$. Obviously, the pair $\langle x_i^{(t)}, x_i^{(t-1)} \rangle$ is distributed according to the product distribution

$$q^{(t)} \quad := \quad p(x^{(t)}|x^{(t-1)}, a^{(t-1)}) \times Bel(x^{(t-1)}) \quad (3)$$

which is commonly called *proposal distribution*.

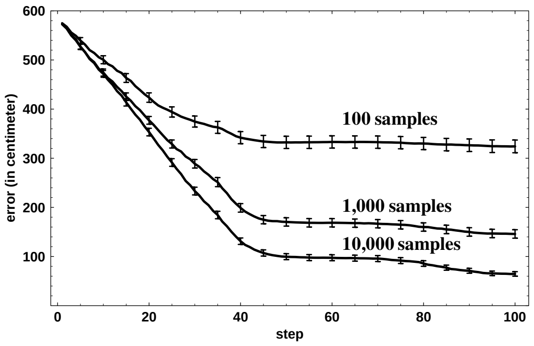

Figure 2: Average error of MCL as a function of the number of robot steps/measurements.

3. To offset the difference between the proposal distribution and the desired distribution (c.f., Equation (2))

$$\eta\, p(o^{(t)}|x^{(t)})p(x^{(t)}|x^{(t-1)}, a^{(t-1)})Bel(x^{(t-1)}) \tag{4}$$

   the sample is weighted by the quotient

$$\frac{\eta\, p(o^{(t)}|x_i^{(t)})p(x_i^{(t)}|x_i^{(t-1)}, a^{(t-1)})Bel(x_i^{(t-1)})}{Bel(x_i^{(t-1)})\; p(x_i^{(t)}|x_i^{(t-1)}, a^{(t-1)})}$$
$$\propto \quad p(o^{(t)}|x_i^{(t)}) \quad = \quad w_i \tag{5}$$

   This is exactly the new (non-normalized) importance factor w_i.

After the generation of m samples, the new importance factors are normalized so that they sum up to 1 (hence define a probability distribution). It is known [17] that under mild assumptions (which hold in our work), the sample set converges to the true posterior $Bel(x^{(t)})$ as m goes to infinity, with a convergence speed in $O(\frac{1}{\sqrt{m}})$. The speed may vary by a constant factor, which can vary drastically depending on the proposal distribution.

Examples

Figure 1 shows an example of MCL in the context of localizing a mobile robot globally in an office environment. This robot is equipped with sonar range finders, and it is also given a map of the environment. In Figure 1a, the robot is globally uncertain; hence the samples are spread uniformly trough the free-space (projected into 2D). Figure 1b shows the sample set after approximately 1 meter of robot motion, at which point MCL has disambiguated the robot's position up to a single symmetry. Finally, after another 2 meters of robot motion the ambiguity is resolved, and the robot knows where it is. The majority of samples is now centered tightly around the correct position, as shown in Figure 1c.

Unfortunately, data collected from a physical robot makes it impossible to freely vary the level of noise in sensing. Figure 2 shows results obtained from a robot simulation, modeling a B21 robot localizing an object in 3D with a mono camera while moving around. The noise simulation includes a simulation of measurement noise, false positives (phantoms) and false negatives (failures to detect the target object). MCL is directly applicable; with the added advantage that we can vary the level of noise arbitrarily. Figure 2 shows systematic error curves for MCL in global localization for different sample set sizes m, averaged over 1,000 individual experiments. The bars in this figure are confidence intervals at the 95% level. With 10,000 samples, the computation load on a Pentium III (500 MhZ) is only 14%, indicating that MCL is well-suited for real-time applications. The results also indicate good performance as the number of samples is large. The reader should notice that these results have been obtained for perceptual noise level of 20% (for both false-negative and false-positive) and an additional position noise that is Gaussian-distributed with a variance of 10 degrees. For our existing robot system, the errors are in fact much lower.

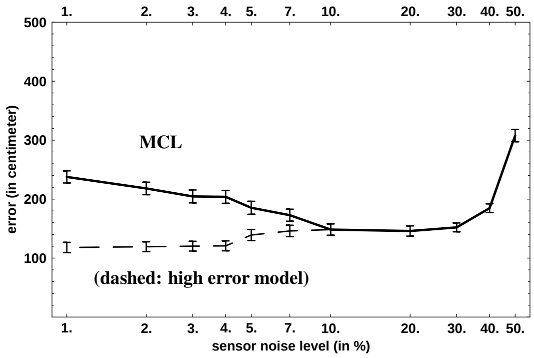

Figure 3: Solid curve: error of MCL after 100 steps, as a function of the sensor noise. 95% confidence intervals are indicated by the bars. Notice that this function is *not* monotonic, as one might expect. Dashed curve: Same experiment with high-error model.

A Problem with MCL

As noticed by several authors [4, 11, 12, 15], the basic particle filter performs poorly if the proposal distribution, which is used to generate samples, places too little samples in regions where the desired posterior $Bel(x_t)$ is large.

This problem has indeed great practical importance in the context of MCL, as the following example illustrates. The solid curve in Figure 3 shows the accuracy MCL achieves after 100 steps, using $m = 1,000$ samples. These results were obtained in simulation, enabling us to vary the amount of perceptual noise from 50% (on the right) to 1% (on the left); in particular, we simulated a mobile robot localizing an object in 3D space from mono-camera imagery. It appears that MCL works best for 10% to 20% perceptual noise. The degradation of performance towards the right, when there is a lot of noise, barely surprises. The less accurate a sensor, the larger an error one should expect. However, MCL also performs poorly when the noise level is too small. In other words, MCL with accurate sensors may perform *worse* than MCL with inaccurate sensors. This finding is a bit counter-intuitive in that it suggests that MCL only works well in specific situations, namely those where the sensors possess the "right" amount of noise.

At first glance, one might attempt to fix the problem by using a perceptual likelihood $p(o^{(t)}|x^{(t)})$ that overestimates the sensor noise. In fact, such a strategy partially alleviates the problem: The dashed curve in Figure 3b shows the accuracy if the error model assumes a fixed 10% noise (shown there only for smaller "true" error rates). While the performance is better, this is barely a fix. The overly pessimistic sensor model is inaccurate, throwing away precious information in

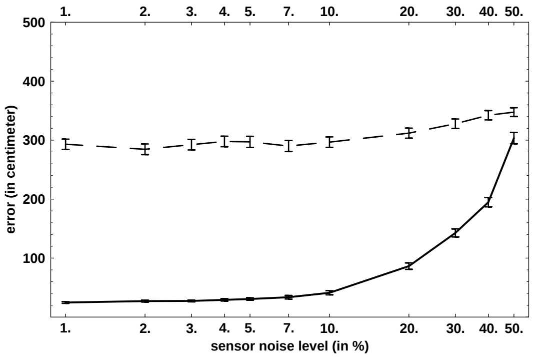

Figure 4: Error of MCL with the dual (dashed line) and the mixture (solid line) proposal distribution—the latter is the distribution advocated here. Compare the solid graph with dashed one, and the curves in Figure 3!

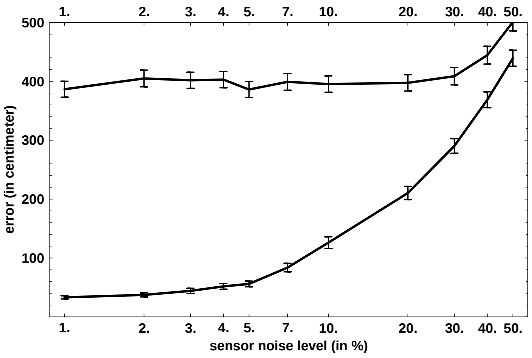

Figure 5: Error of plain MCL (top curve) and MCL with the mixture proposal distribution (bottom curve) with 50 samples (instead of 1,000) for each belief state.

the sensor readings. In fact, the resulting belief is not any longer a posterior, even if infinitely many samples were used. As we will see below, a mathematically sound method exists that produces much better results.

To analyze the problem more thoroughly, we first notice that the true goal of Bayes filtering is to calculate the product distribution specified in Equation (4). Thus, the optimal proposal distribution would be this product distribution. However, sampling from this distribution directly is too difficult. As noticed above, MCL samples instead from the proposal distribution $q^{(t)}$ defined in Equation (3), and uses the importance factors (5) to account for the difference. It is well-known from the statistical literature [4, 12, 15, 17] that the divergence between (3) and (4) determines the convergence speed. This difference is accounted by the perceptual density $p(o^{(t)}|x^{(t)})$: If the sensors are entirely uninformative, this distribution is flat and (3) is equivalent to (4). For low-noise sensors, however, $p(o^{(t)}|x^{(t)})$ is typically quite narrow, hence MCL converges slowly. Thus, the error in Figure 3 is in fact caused by two different types of errors: one arising from the limitation of the sensor data (=noise), and one that arises from the mismatch of (3) and (4) in MCL. As we will show in this paper, an alternative version of MCL exists that practically eliminates the second error source.

Alternative Proposal Distributions

An alternative proposal distribution, which alleviates this problem, can be obtained by sampling directly from

$$\bar{q}^{(t)} = \frac{p(o^{(t)}|x^{(t)})}{\pi(o^{(t)})} \quad \text{with} \quad \pi(o^{(t)}) = \int p(o^{(t)}|x^{(t)})\, dx^{(t)} \tag{6}$$

This proposal distribution leads to the *dual* of MCL. It can be viewed as the logical "inverse" of the sampling in regular MCL: Rather than forward-guessing and then using the importance factors to adjust the likelihood of a guess based on an observation, dual MCL guesses "backwards" from the observation and adjusts the importance factor based on the belief $Bel(x^{(t-1)})$. Consequently, the dual proposal distribution possesses complimentary strengths and weaknesses: while it is ideal for highly accurate sensors, its performance is negatively affected by measurement noise. The key advantage of dual MCL is that when the distribution of $p(o|x)$ is narrow—which is the case for low-noise sensors—dual sampling can be much more effective than conventional MCL.

Importance Factors

We will now provide three alternative ways to calculate the importance factors for $\bar{q}^{(t)}$.

Approach 1 (proposed by Arnaud Doucet, personal communication): Draw $x_i^{(t-1)} \sim Bel(x^{(t-1)})$. Hence, the pair $\langle x_i^{(t)}, x_i^{(t-1)} \rangle$ is distributed according to

$$\frac{p(o^{(t)}|x^{(t)})}{\pi(o^{(t)})} \times Bel(x^{(t-1)}) \tag{7}$$

and the importance factor is obtained as follows:

$$\begin{aligned} w_i &= \left[\frac{p(o^{(t)}|x_i^{(t)})}{\pi(o^{(t)})} \times Bel(x_i^{(t-1)}) \right]^{-1} \\ & \quad \frac{p(o^{(t)}|x_i^{(t)})\ p(x_i^{(t)}|x_i^{(t-1)}, a^{(t-1)})\ Bel(x_i^{(t-1)})}{p(o^{(t)}|a^{(t-1)}, \ldots, o^{(0)})} \\ &= \frac{p(x_i^{(t)}|x_i^{(t-1)}, a^{(t-1)})\ \pi(o^{(t)})}{p(o^{(t)}|a^{(t-1)}, \ldots, o^{(0)})} \\ &\propto p(x_i^{(t)}|x_i^{(t-1)}, a^{(t-1)}) \end{aligned} \tag{8}$$

This approach is mathematically more elegant than the two alternatives described below, in that it avoids the need to transform sample sets into densities (which will be the case below). We have not yet implemented this approach. However, in the context of global mobile robot localization, we suspect the importance factor $p(x_i^{(t)}|a^{(t-1)}, x_i^{(t-1)})$ will be zero for many pose pairs $\langle x_i^{(t)}, x_i^{(t-1)} \rangle$.

Approach 2 Alternatively, one may in an explicit forward phase sample $x_j^{(t-1)} \sim Bel(x^{(t-1)})$ and then $x_j^{(t)} \sim p(x^{(t)}|x_j^{(t-1)}, a^{(t-1)})$, which represents the robot's belief *before* incorporating the sensor measurement. The "trick" is then to transform the samples $x_j^{(t)}$ into a kd-tree [1, 14] that represents the density $p(x^{(t)}|a^{(t-1)}, d^{(0\ldots t)})$, which is again the pose belief just before incorporating the most recent observation $o^{(t)}$.

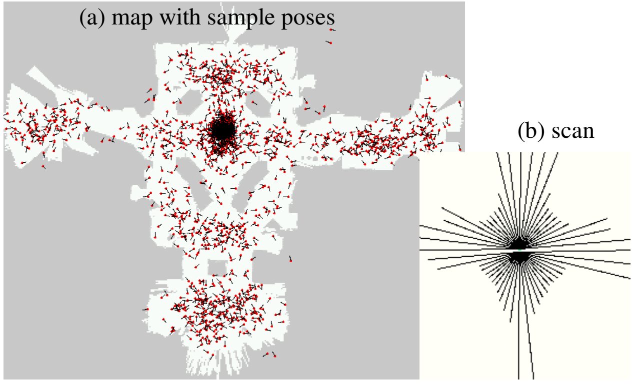

Figure 6: Robot poses sampled according to $\bar{q}$ for the scan shown on the right, using a pre-compiled version of the joint distribution $p(o, x)$ represented by kd-trees.

After this first phase, the importance weights of our samples $x_i^{(t)} \sim \bar{q}^{(t)}$ are then calculated as follows:

$$w_i = \left[\frac{p(o^{(t)}|x_i^{(t)})}{\pi(o^{(t)})} \right]^{-1} \frac{p(o^{(t)}|x_i^{(t)})\ p(x_i^{(t)}|a^{(t-1)}, d^{(0\ldots t-1)})}{p(o^{(t)}|d^{(0\ldots t-1)}, a^{(t-1)})}$$
$$\propto \quad p(x_i^{(t)}|a^{(t-1)}, d^{(0\ldots t-1)}) \tag{9}$$

This approach avoids the danger of generating pairs of poses $\langle x_i^{(t)}, x_i^{(t-1)} \rangle$ for which $w_i = 0$, but it involves an explicit forward sampling phase, which can be computationally expensive (below we will mix the forward samples with backward samples which partially overcomes this criticism).

Approach 3 The third approach combines the best of both worlds, in that it avoids the explicit forward-sampling phase of the second approach, but also tends to generate large importance factors. In particular, it transforms the initial belief $Bel(x^{(t-1)})$ into a kd-tree. For each sample $x_i^{(t)} \sim \bar{q}^{(t)}$, we now draw a sample $x_i^{(t-1)}$ from the distribution

$$\frac{p(x_i^{(t)}|a^{(t-1)}, x^{(t-1)})}{\pi(x_i^{(t)}|a^{(t-1)})} \tag{10}$$

where

$$\pi(x_i^{(t)}|a^{(t-1)}) \quad = \quad \int p(x_i^{(t)}|a^{(t-1)}, x^{(t-1)})\ dx^{(t-1)} \tag{11}$$

In other words, our approach projects $x_i^{(t)}$ back to a possible successor pose $x_i^{(t-1)}$. Consequently, the pair of poses $\langle x_i^{(t)}, x_i^{(t-1)} \rangle$ is distributed according to

$$\frac{p(o^{(t)}|x_i^{(t)})}{\pi(o^{(t)})} \times \frac{p(x_i^{(t)}|a^{(t-1)}, x_i^{(t-1)})}{\pi(x_i^{(t)}|a^{(t-1)})} \tag{12}$$

which gives rise to the following importance factor:

$$w_i \quad = \quad \left[\frac{p(o^{(t)}|x_i^{(t)})}{\pi(o^{(t)})} \times \frac{p(x_i^{(t)}|a^{(t-1)}, x_i^{(t-1)})}{\pi(x_i^{(t)}|a^{(t-1)})} \right]^{-1}$$

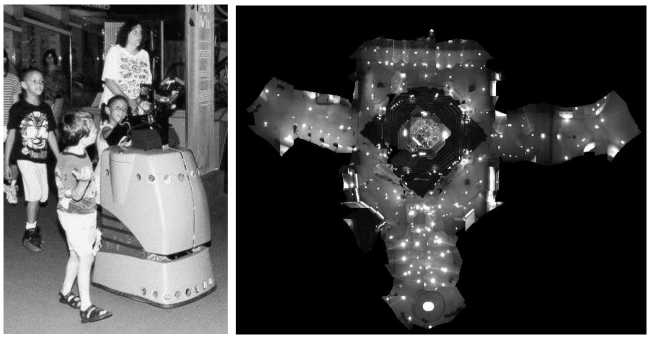

Figure 7: Left: The interactive tourguide robot Minerva. Right: Ceiling Map of the Smithsonian Museum.

$$\frac{p(o^{(t)}|x_i^{(t)}) p(x_i^{(t)}|x_i^{(t-1)}, a^{(t-1)})\ Bel(x_i^{(t-1)})}{p(o^{(t)}|d^{(0\ldots t-1)})}$$
$$= \quad \frac{\pi(o^{(t)})\ \pi(x_i^{(t)}|a^{(t-1)})\ Bel(x_i^{(t-1)})}{p(o^{(t)}|d^{(0\ldots t-1)})}$$
$$\propto \quad \pi(x_i^{(t)}|a^{(t-1)})\ Bel(x_i^{(t-1)}) \tag{13}$$

where $Bel(x_i^{(t-1)})$ is calculated using the kd-tree representing this belief density. The only complication arises from the need to calculate $\pi(x_i^{(t)}|a^{(t-1)})$, which depends on both $x_i^{(t)}$ and $a^{(t-1)}$. Luckily, in mobile robot localization, $\pi(x_i^{(t)}|a^{(t-1)})$ can safely be assumed to be a constant, although this assumption may not be valid in general.

The reader should notice that all three approaches require a method for sampling poses from observations according to $\bar{q}^{(t)}$—which can be non-trivial in mobile robot applications. The first approach is the easiest to implement and mathematically most straightforward. However, as noted above, we suspect that it will be inefficient for mobile robot localization. The two other approaches rely on a density estimation method (such as kd-trees). The third also requires a method for sampling poses backwards in time, which further complicates its implementation. However, the superior results given below may well make this additional work worthwhile.

The Mixture Proposal Distribution

Obviously, neither proposal distribution is sufficient, as they both fail in certain cases. To illustrate this, the dashed line in Figure 4 shows the performance for the dual. As in the previous figure, the horizontal axis depicts the amount of noise in perception, and the vertical axis depicts the error in centimeters, averaged over 1,000 independent runs. Two things are remarkable in these experimental results: First, the accuracy if now *monotonic* in perceptual noise: More accurate sensors give better results. Second, however, the overall performance is much poorer than that of conventional MCL. The poor performance of the dual is due to the fact that *erroneous* sensor measurements have a devastating effect on the estimated belief, since almost all samples are generated at the "wrong" place.

This consideration leads us to the central algorithm proposed in this paper, which uses the following mixture

$$(1 - \phi)q^{(t)} \ + \ \phi\bar{q}^{(t)} \tag{14}$$

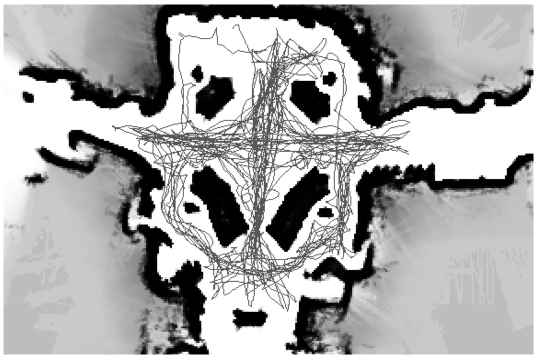

Figure 8: Part of the map of the Smithsonian's Museum of National History and path of the robot.

with $0 \leq \phi \leq 1$ as the proposal distribution. In our experiments, the mixing rate is set to ϕ throughout. Experiments with an adaptive mixing rate (using pursuit methods) did not improve the performance in a noticeable way.

Sampling Form $\bar{q}$

The remaining principle difficulty in applying our new approach to robotics is that it may not be easy to sample poses x_i based on sensor measurements o. In particular, the previous MCL algorithm "only" requires an algorithm for calculating $p(o|x)$; while there is obvious ways to extend this into a sampling algorithm, sampling *efficiently* from $\bar{q}$ may not a straightforward matter—this is specifically the case for the experiments with laser range finders described below.

Unfortunately, space limitations prohibit a detailed description of our solution. In our implementation, a kd-tree representing the joint distribution $p(o, x)$ is learned in a preprocessing phase, using real robot data (a log-file) as a "sample" of o, and $p(o|x)$ with randomly generated poses x to generate a weighted sample that represents $p(o, x)$. The nice aspect of the tree is that it permits efficient sampling of the desired conditional. Figure 6 shows a set of poses generated for a specific laser range measurement o.

Experimental Results

Our experiments were carried out both in simulation and for data collected with our tour-guide robot Minerva (shown in Figure 7), collected during a two-week period in which it gave tours to thousands of people in the Smithsonian's Museum of National History [18]. The simulation experiments were carried out using the third method for calculating importance factors outlined above. A comparative study showed no noticeable difference between this and the second method. All real-world results were carried out using the second approach, in part because it avoids backwards sampling of poses. As noted above, we did not yet implement the first method.

Simulation Figure 4 shows the performance of MCL with the mixture proposal distribution, under conditions that are otherwise identical to those in Figures 3. As these results suggest, our new MCL algorithm outperforms both MCL and its dual by a large margin. At every single noise level, our

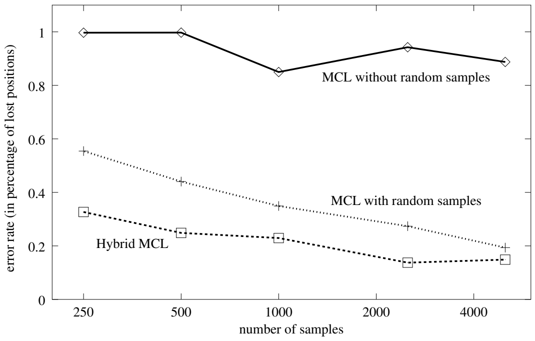

Figure 9: Performance of conventional (top curve), conventional with random samples (middle curve) and our new mixture (bottom curve) MCL for the kidnapped robot problem in the Smithsonian museum. The error rate is measured in percentage of time during which the robot lost track of its position.

new algorithm outperforms its alternatives by a factor that ranges from 1.07 (high noise level) to 9.7 (low noise level). For example, at a noise level of 1%, our new MCL algorithm exhibits an average error of 24.6cm, whereas MCL's error is 238cm and that of dual MCL is 293cm. In comparison, the average error with noise-free sensors and the optimal estimator is approximately 19.5cm (it's not zero since the robot has to face the object to see it).

Our approach also degrades nicely to very small sample sets. Figure 5 plots the error of conventional MCL (top curve) and MCL with mixture proposal distribution (bottom curve) for different error levels, using $m = 50$ samples only. With 50 samples, the computational load is 0.126% on a 500MHz Pentium Computer—meaning that the algorithm is approximately 800 faster than real-time. While plain MCL basically fails under this circumstances to track the robot's position, our new version of MCL performs excellently, and is only slightly inferior to $m = 1,000$ samples.

Real-World Experiments with Lasers Our approach was tested using data recorded during a two-week deployment of the mobile robot Minerva as a museum tour-guide in the Smithsonian's Museum of National History [18]. The data contains logs of odometry measurements and sensor scans taken by Minerva's two laser range-finders. Figure 8 shows part of the map of the museum and the path of the robot used for this experiment.

As reported in [2, 3, 6], conventional MCL reliably succeeds in localizing the robot. To test our new approach under even harder conditions, we repeatedly introduced errors into the odometry information. These errors made the robot lose track of its position with probability of 0.01 when advancing one meter. The resulting localization problem is known as the *kidnapped robot problem* [5], which is generally acknowledged as the most challenging localization problem. As argued in [7], this problem tests the ability to recover from extreme failures of the localization algorithm.

Figure 9 shows comparative results for three different approaches. The error is measured by the percentage of time, during which the estimated position deviates by more than 2 meters from the reference position. Obviously, the mixture proposal distribution yields significantly better results, even

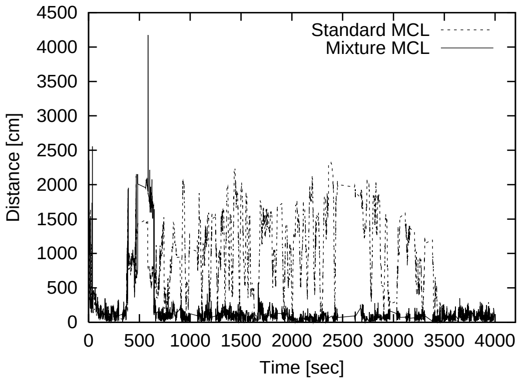

Figure 10: MCL with the standard proposal distribution (dashed curve) compared to MCL with the new mixture distribution (solid line). Shown here is the error for a 4,000-second episode of camera-based localization in the Smithsonian museum.

if the basic proposal distribution is mixed with 5% random samples (as suggested in [7] as a solution to the kidnapped robot problem). The mixture proposal distribution reduces the error rate of localization by as much as 70% more than MCL if the standard proposal distribution is employed; and 32% when compared to the case where the standard proposal distribution is mixed with a uniform distribution. These results are significant at the 95% confidence level, evaluated over actual robot data.

Real-World Experiments with Vision We also compared MCL with different proposal distributions in the context of visual localization, using only camera imagery obtained with the robot Minerva during public museum hours [2]. Figure 7 shows on the right a texture mosaic of the museum's ceiling. Since the ceiling height is unknown, only the center region in the camera image is used for localization.

The image sequence used for evaluation is of extremely poor quality, as people often intentionally covered the camera with their hand and placed dirt on the lens. Figure 10 shows the localization error obtained when using vision only (calculated using the localization results from the laser as ground truth). The data covers a period of approximately 4,000 seconds, during which MCL processes a total of 20,740 images. After approximately 630 seconds, a drastic error in the robot's odometry leads to a loss of the position (which is an instance of the kidnapped robot problem). As the two curves in Figure 10 illustrate, the regular MCL sampler (dashed curve) is unable to recover from this event, whereas MCL with mixture proposal distribution (solid curve) recovers quickly. These result are not statistically significant in that only a single run is considered, but they confirm our findings with laser range finders. Together, our result suggest that the mixture distribution drastically increases the robustness of the statistical estimator for mobile robot localization.

Conclusion

This paper introduced a new proposal distribution for Monte Carlo localization, a randomized Bayesian algorithm for mobile robot localization. Our approach combines two proposal distribution which sample from different factors of the desired posterior. By doing so, our approach overcomes a range of limitations that currently exist for different versions of MCL, such as the inability to estimate posteriors for highly accurate sensors, poor degradation to small sample sets, and the ability to recover from unexpected large state changes (robot kidnapping). Extensive experimental results suggest that our new approach consistently outperforms MCL by a large margin. The resulting algorithm is highly practical, and might improve the performance of particle filters in a range of applications.

Acknowledgments

The authors are indebted to Nando de Freitas and Arnaud Doucet for whose insightful comments on an earlier draft of a related paper. We also thank Frank Dellaert and the members of CMU's Robot Learning Lab for invaluable suggestions and comments.

References

[1] J.L. Bentley. Multidimensional divide and conquer. *Communications of the ACM*, 23(4), 1980.

[2] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. CVPR-99.

[3] J. Denzler, B. Heigl, and H. Niemann. Combining computer graphics and computer vision for probabilistic self-localization. Rochester University, Internal Report, 1999.

[4] A Doucet. On sequential simulation-based methods for Bayesian filtering. TR CUED/F-INFENG/TR 310, Cambridge Univ., 1998.

[5] S. Engelson and D. McDermott. Error correction in mobile robot map learning. ICRA-92.

[6] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. AAAI-99.

[7] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *JAIR* 11, 1999.

[8] R. Fung and B. Del Favero. Backward simulation in bayesian networks. UAI-94.

[9] M. Isard and A. Blake. Condensation: conditional density propagation for visual tracking. *IJCV*, In Press.

[10] K. Kanazawa, D. Koller, and S.J. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. UAI-95.

[11] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. ICRA-2000, to appear.

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

[13] P.S. Maybeck. The Kalman filter: An introduction to concepts. In *Autonomous Robot Vehicles*. Springer, 1990.

[14] A. W. Moore. *Efficient Memory-based Learning for Robot Control*. PhD thesis, Cambridge Univ., 1990.

[15] M. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filter. *Journal of the American Statistical Association*, 1999.

[16] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 1986.

[17] M.A. Tanner. *Tools for Statistical Inference*. Springer, 1993.

[18] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second generation mobile tour-guide robot. ICRA-99.