Monte Carlo Localization: Efficient Position Estimation for Mobile Robots

Dieter Fox, Wolfram Burgard[†], Frank Dellaert, Sebastian Thrun

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

[†]Computer Science Department III
University of Bonn
Bonn, Germany

Abstract

This paper presents a new algorithm for mobile robot localization, called Monte Carlo Localization (MCL). MCL is a version of Markov localization, a family of probabilistic approaches that have recently been applied with great practical success. However, previous approaches were either computationally cumbersome (such as grid-based approaches that represent the state space by high-resolution 3D grids), or had to resort to extremely coarse-grained resolutions. Our approach is computationally efficient while retaining the ability to represent (almost) arbitrary distributions. MCL applies sampling-based methods for approximating probability distributions, in a way that places computation "where needed." The number of samples is adapted on-line, thereby invoking large sample sets only when necessary. Empirical results illustrate that MCL yields improved accuracy while requiring an order of magnitude less computation when compared to previous approaches. It is also much easier to implement.

Introduction

Throughout the last decade, sensor-based localization has been recognized as a key problem in mobile robotics (Cox 1991; Borenstein, Everett, & Feng 1996). Localization is a version of on-line temporal state estimation, where a mobile robot seeks to estimate its position in a global coordinate frame. The localization problem comes in two flavors: *global localization* and *position tracking*. The second is by far the most-studied problem; here a robot knows its initial position and "only" has to accommodate small errors in its odometry as it moves. The global localization problem involves a robot which is not told its initial position; hence, it has to solve a much more difficult localization problem, that of estimating its position from scratch (this is sometimes referred to as the *hijacked robot problem* (Engelson 1994)). The ability to localize itself—both locally and globally—played an important role in a collection of recent mobile robot applications (Burgard *et al.* 1998a; Endres, Feiten, & Lawitzky 1998; Kortenkamp, Bonasso, & Murphy 1997).

Copyright ©1999, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

While the majority of early work focused on the tracking problem, recently several researchers have developed what is now a highly successful family of approaches capable of solving both localization problems: *Markov localization* (Nourbakhsh, Powers, & Birchfield 1995; Simmons & Koenig 1995; Kaelbling, Cassandra, & Kurien 1996; Burgard *et al.* 1996). The central idea of Markov localization is to represent the robot's belief by a probability distribution over possible positions, and use Bayes rule and convolution to update the belief whenever the robot senses or moves. The idea of probabilistic state estimation goes back to Kalman filters (Gelb 1974; Smith, Self, & Cheeseman 1990), which use multivariate Gaussians to represent the robot's belief. Because of the restrictive nature of Gaussians (they can basically represent one hypothesis only annotated by its uncertainty) Kalman-filters usually are only applied to position tracking. Markov localization employs discrete, but *multi-modal* representations for representing the robot's belief, hence can solve the global localization problem. Because of the real-valued and multi-dimensional nature of kinematic state spaces these approaches can only *approximate* the belief, and accurate approximation usually requires prohibitive amounts of computation and memory.

In particular, *grid-based* methods have been developed that approximate the kinematic state space by fine-grained piecewise constant functions (Burgard *et al.* 1996). For reasonably-sized environments, these approaches often require memory in the excess of 100MB, and high-performance computing. At the other extreme, various researchers have resorted to coarse-grained *topological* representations, whose granularity is often an order of magnitude lower than that of the grid-based approach. When high resolution is needed (see e.g., (Fox *et al.* 1998), who uses localization to avoid collisions with static obstacles that cannot be detected by sensors), such approaches are inapplicable.

In this paper we present **M**onte **C**arlo **L**ocalization (in short: MCL). Monte Carlo methods were introduced in the Seventies (Handschin 1970), and recently rediscovered independently in the target-tracking (Gordon, Salmond, & Smith 1993), statistical (Kitagawa 1996) and computer vision literature (Isard & Blake 1998), and they have also be applied in dynamic probabilistic networks (Kanazawa, Koller, & Rus-

sell 1995). MCL uses fast sampling techniques to represent the robot's belief. When the robot moves or senses, importance re-sampling (Rubin 1988) is applied to estimate the posterior distribution. An adaptive sampling scheme (Koller & Fratkina 1998), which determines the number of samples on-the-fly, is employed to trade-off computation and accuracy. As a result, MCL uses many samples during global localization when they are most needed, whereas the sample set size is small during tracking, when the position of the robot is approximately known.

By using a sampling-based representation, MCL has several key advantages over earlier work in the field:

1. In contrast to existing Kalman filtering based techniques, it is able to represent multi-modal distributions and thus can *globally* localize a robot.
2. It drastically reduces the amount of memory required compared to grid-based Markov localization and can integrate measurements at a considerably higher frequency.
3. It is more *accurate* than Markov localization with a fixed cell size, as the state represented in the samples is not discretized.
4. It is much easier to implement.

Markov Localization

This section briefly outlines the basic Markov localization algorithm upon which our approach is based. The key idea of Markov localization—which has recently been applied with great success at various sites (Nourbakhsh, Powers, & Birchfield 1995; Simmons & Koenig 1995; Kaelbling, Cassandra, & Kurien 1996; Burgard *et al.* 1996; Fox 1998)—is to compute a probability distribution over all possible positions in the environment. Let $l = \langle x, y, \theta \rangle$ denote a position in the state space of the robot, where x and y are the robot's coordinates in a world-centered Cartesian reference frame, and θ is the robot's orientation. The distribution $Bel(l)$ expresses the robot's belief for being at position l. Initially, $Bel(l)$ reflects the initial state of knowledge: if the robot knows its initial position, $Bel(l)$ is centered on the correct position; if the robot does not know its initial position, $Bel(l)$ is uniformly distributed to reflect the global uncertainty of the robot. As the robot operates, $Bel(l)$ is incrementally refined.

Markov localization applies two different probabilistic models to update $Bel(l)$, an action model to incorporate movements of the robot into $Bel(l)$ and a perception model to update the belief upon sensory input:

Robot motion is modeled by a conditional probability $P(l \mid l', a)$ (a kernel), specifying the probability that a measured movement action a, when executed at l', carries the robot to l. $Bel(l)$ is then updated according to the following general formula, commonly used in Markov chains (Chung 1960):

$$Bel(l) \quad \longleftarrow \quad \int P(l \mid l', a)\ Bel(l')\ dl' \tag{1}$$

The term $P(l \mid l', a)$ represents a model of the robot's kinematics, whose probabilistic component accounts for errors in odometry. Following (Burgard *et al.* 1996), we assume odometry errors to be distributed normally.

Sensor readings are integrated with Bayes rule. Let s denote a sensor reading and $P(s \mid l)$ the likelihood of perceiving s given that the robot is at position l, then $Bel(l)$ is updated according to the following rule:

$$Bel(l) \quad \longleftarrow \quad \alpha\ P(s \mid l)\ Bel(l) \tag{2}$$

Here α is a normalizer, which ensures that $Bel(l)$ integrates to 1.

Strictly speaking, both update steps are only applicable if the environment is *Markovian*, that is, if past sensor readings are conditionally independent of future readings given the true position of the robot. Recent extensions to non-Markovian environments (Fox *et al.* 1998) can easily be stipulated to the MCL approach; hence, throughout this paper will assume that the environment is Markovian and will not pay further attention to this issue.

Prior Work

Existing approaches to mobile robot localization can be distinguished by the way they represent the state space of the robot.

Kalman filter-based techniques. Most of the earlier approaches to robot localization apply Kalman filters (Kalman 1960). The vast majority of these approaches is based on the assumption that the uncertainty in the robot's position can be represented by a unimodal Gaussian distribution. Sensor readings, too, are assumed to map to Gaussian-shaped distributions over the robot's position. For these assumptions, Kalman filters provide extremely efficient update rules that can be shown to be optimal (relative to the assumptions) (Maybeck 1979). Kalman filter-based techniques (Leonard & Durrant-Whyte 1992; Schiele & Crowley 1994; Gutmann & Schlegel 1996) have proven to be robust and accurate for keeping track of the robot's position. However, since these techniques do not represent multi-modal probability distributions, which frequently occur during global localization. In practice, localization approaches using Kalman filters typically require that the starting position of the robot is known. In addition, Kalman filters rely on sensor models that generate estimates with Gaussian uncertainty—which is often unrealistic.

Topological Markov localization. To overcome these limitations, different approaches have used increasingly richer schemes to represent uncertainty, moving beyond the Gaussian density assumption inherent in the vanilla Kalman filter. These different methods can be roughly distinguished by the type of discretization used for the representation of the state space. In (Nourbakhsh, Powers, & Birchfield 1995; Simmons & Koenig 1995; Kaelbling, Cassandra, & Kurien 1996), Markov localization is used for landmark-based corridor navigation and the state space is organized according

to the coarse, topological structure of the environment. The coarse resolution of the state representation limits the accuracy of the position estimates. Topological approaches typically give only a rough sense as to where the robot is.

Grid-based Markov localization. To deal with multimodal and non-Gaussian densities at a fine resolution (as opposed to the coarser discretization in the above methods), grid-based approaches perform numerical integration over an evenly spaced grid of points (Burgard *et al.* 1996; 1998b; Fox 1998). This involves discretizing the interesting part of the state space, and use it as the basis for an approximation of the state space density, e.g. by a piece-wise constant function. Grid-based methods are powerful, but suffer from excessive computational overhead and *a priori* commitment to the size and resolution of the state space. In addition, the resolution and thereby also the precision at which they can represent the state has to be fixed beforehand. The computational requirements have an effect on accuracy as well, as not all measurements can be processed in real-time, and valuable information about the state is thereby discarded. Recent work (Burgard *et al.* 1998b) has begun to address some of these problems, using *oct-trees* to obtain a variable resolution representation of the state space. This has the advantage of concentrating the computation and memory usage where needed, and addresses the limitations arising from fixed resolutions.

Monte Carlo Localization

Sample-Based Density Approximation

MCL is a version of sampling/importance re-sampling (SIR) (Rubin 1988). It is known alternatively as the bootstrap filter (Gordon, Salmond, & Smith 1993), the Monte-Carlo filter (Kitagawa 1996), the Condensation algorithm (Isard & Blake 1998), or the survival of the fittest algorithm (Kanazawa, Koller, & Russell 1995). All these methods are generically known as *particle filters*, and a discussion of their properties can be found in (Doucet 1998).

The key idea underlying all this work is to represent the posterior belief $Bel(l)$ by a set of N weighted, random samples or *particles* $S = \{s_i \mid i = 1..N\}$. A sample set constitutes a discrete approximation of a probability distribution. Samples in MCL are of the type

$$\langle\langle x, y, \theta\rangle, p\rangle \tag{3}$$

where $\langle x, y, \theta\rangle$ denote a robot position, and $p \geq 0$ is a numerical weighting factor, analogous to a discrete probability. For consistency, we assume $\sum_{n=1}^{N} p_n = 1$.

In analogy with the general Markov localization approach outlined in the previous section, MCL proceeds in two phases:

Robot motion. When the robot moves, MCL generates N new samples that approximate the robot's position after the motion command. Each sample is generated by *randomly* drawing a sample from the previously computed sample set, with likelihood determined by their p-values. Let l' denote the position of this sample. The new sample's l is then generated by generating a single, random sample from $P(l \mid l', a)$, using the action a as observed. The p-value of the new sample is N^{-1}.

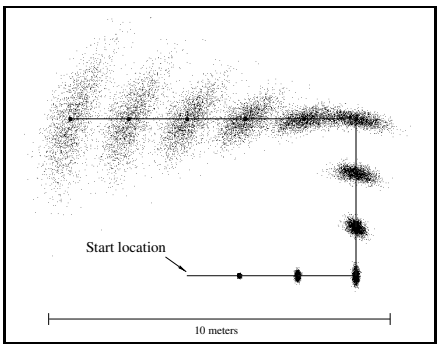

Fig. 1: Sampling-based approximation of the position belief for a non-sensing robot.

Figure 1 shows the effect of this sampling technique, starting at an initial known position (bottom center) and executing actions as indicated by the solid line. As can be seen there, the sample sets approximate distributions with increasing uncertainty, representing the gradual loss of position information due to slippage and drift.

Sensor readings are incorporated by re-weighting the sample set, in a way that implements Bayes rule in Markov localization. More specifically, let $\langle l, p\rangle$ be a sample. Then

$$p \quad \longleftarrow \quad \alpha\, P(s \mid l) \tag{4}$$

where s is the sensor measurement, and α is a normalization constant that enforces $\sum_{n=1}^{N} p_n = 1$. The incorporation of sensor readings is typically performed in two phases, one in which p is multiplied by $P(s \mid l)$, and one in which the various p-values are normalized. An algorithm to perform this re-sampling process efficiently in O(N) time is given in (Carpenter, Clifford, & Fernhead 1997).

In practice, we have found it useful to add a small number of uniformly distributed, random samples after each estimation step. Formally, this is legitimate because the SIR methodology (Rubin 1988) can accommodate arbitrary distributions for sampling as long as samples are weighted appropriately (using the factor p), and as long as the distribution from which samples are generated is non-zero at places where the distribution that is being approximated is non-zero—which is actually the case for MCL. The added samples are essential for relocalization in the rare event that the robot loses track of its position. Since MCL uses finite sample sets, it may happen that no sample is generated close to the correct robot position. In such cases, MCL would be unable to re-localize the robot. By adding a small number of random samples, however, MCL can effectively re-localize the robot, as documented in the experimental results section of this paper.

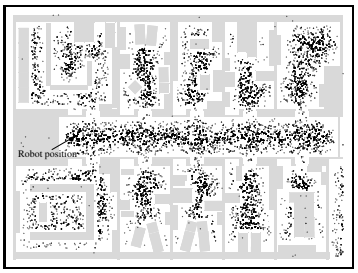

Fig. 2: Global localization: Initialization.

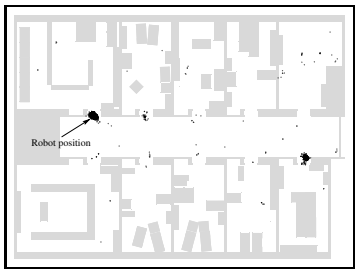

Fig. 3: Ambiguity due to symmetry.

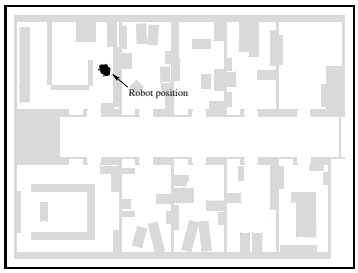

Fig. 4: Successful localization.

Properties of MCL

A nice property of the MCL algorithm is that it can universally approximate arbitrary probability distributions. As shown in (Tanner 1993), the variance of the importance sampler converges to zero at a rate of $1/\sqrt{N}$ (under conditions that are true for MCL). The sample set size naturally trades off accuracy and computational load. The true advantage, however, lies in the way MCL places computational resources. By sampling in proportion to likelihood, MCL focuses its computational resources on regions with high likelihood, where things really matter.

MCL is an online algorithm. It lends itself nicely to an any-time implementation (Dean & Boddy 1988; Zilberstein & Russell 1995). Any-time algorithms can generate an answer at *any* time; however, the quality of the solution increases over time. The sampling step in MCL can be terminated at any time. Thus, when a sensor reading arrives, or an action is executed, sampling is terminated and the resulting sample set is used for the next operation.

Adaptive Sample Set Sizes

In practice, the number of samples required to achieve a certain level of accuracy varies drastically. During global localization, the robot is completely ignorant as to where it is; hence, it's belief uniformly covers its full three-dimensional state space. During position tracking, on the other hand, the uncertainty is typically small and often focused on lower-dimensional manifolds. Thus, many more samples are needed during global localization to accurately approximate the true density, than are needed for position tracking.

MCL determines the sample set size on-the-fly. As in (Koller & Fratkina 1998), the idea is to use the divergence of $P(l)$ and $P(l \mid s)$, the belief *before* and *after* sensing, to determine the sample sets. More specifically, both motion data and sensor data is incorporated in a single step, and sampling is stopped whenever the sum of weights p (before normalization!) exceeds a threshold η. If the position predicted by odometry is well in tune with the sensor reading, each individual p is large and the sample set remains small. If, however, the sensor reading carries a lot of surprise, as is typically the case when the robot is globally uncertain or when it lost track of its position, the individual p-values are small and the sample set is large.

Our approach directly relates to the well-known property that the variance of the importance sampler is a function of the mismatch of the sampling distribution (in our case $P(l)$) and the distribution that is being approximated with the weighted sample (in our case $P(l \mid s)$) (Tanner 1993). The less these distributions agree, the larger the variance (approximation error). The idea is here to compensate such error by larger sample set sizes, to obtain approximately uniform error.

A Graphical Example

Figures 2 to 4 illustrate MCL in practice. Shown there is a series of sample sets (projected into 2D) generated during global localization of our robot RHINO (Figure 5), as it operates in an office building. In Figure 2, the robot is globally uncertain; hence the samples are spread uniformly through the free-space. Figure 3 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, the robot knows where it is. The majority of samples is now centered tightly around the correct position, as shown in Figure 4.

Experimental Results

To evaluate the utility of sampling in localization, we thoroughly tested MCL in a range of real-world environments, applying it to three different types of sensors (cameras, sonar, and laser proximity data). The two primary results are:

1. MCL yields significantly more accurate localization results than the most accurate previous Markov localization algorithm, while consuming an order of magnitude less memory and computational resources. In some cases, MCL reliably localizes the robot whereas previous methods fail.
2. By and large, adaptive sampling performs equally well as MCL with fixed sample sets. In scenarios involving a large range of different uncertainties (global vs. local), however, adaptive sampling is superior to fixed sample sizes.

Our experiments have been carried out using several B21, B18, and Pioneer robots manufactured by ISR/RWI, shown in Figure 5. These robots are equipped with arrays of sonar sensors (from 7 to 24), one or two laser range finders, and

Fig. 5: Four of the robots used for testing: Rhino, Minerva, Robin, and Marian.

in the case of Minerva, shown in Figure 5, a B/W camera pointed at the ceiling. Even though all experimental results discussed here use pre-recorded data sets (to facilitate the analysis), all evaluations have been performed strictly under run-time conditions (unless explicitly noted). In fact, we have routinely ran cooperative teams of mobile robots using MCL for localization (Fox *et al.* 1999).

Comparison to Grid-Based Localization

The first series of experiments illustrates different capabilities of MCL and compares it to grid-based Markov localization, which presumably is the most accurate Markov localization technique to date (Burgard *et al.* 1996; 1998b; Fox 1998).

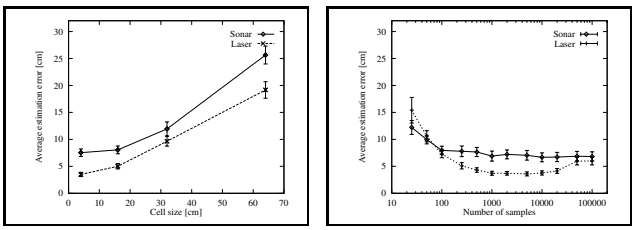

Fig. 6: Accuracy of (a) grid-based Markov localization using different spatial resolutions and (b) MCL for different numbers of samples (log scale).

Figure 6 (a) plots the localization accuracy for grid-based localization as a function of the grid resolution. These results were obtained using data recorded in the environment shown in Figure 2. They are nicely suited for our experiments because the exact same data has already been used to compare different localization approaches, including grid-based Markov localization (which was the only one that solved the global localization problem) (Gutmann *et al.* 1998). *Notice that the results for grid-based localization shown in Figure 6 were not generated in real-time.* As shown there, the accuracy increases with the resolution of the grid, both for sonar (solid line) and for laser data (dashed line). However, grid sizes below 8 cm do not permit updating in real-time, even when highly efficient, selective update schemes are used (Fox, Burgard, & Thrun 1999). Results for MCL with fixed sample set sizes are shown in Figure 6 (b). These results have been generated using real-time conditions. Here very small sample sets are disadvantageous, since they infer too large an error in the approximation. Large sample sets are also disadvantageous, since processing them requires too much time and fewer sensor items can be processed in real-time. The "optimal" sample set size, according to Figure 6 (b), is somewhere between 1,000 and 5,000 samples. Grid-based localization, to reach the same level of accuracy, has to use grids with 4cm resolution—which is infeasible given even our best computers.

In comparison, the grid-based approach, with a resolution of 20 cm, requires almost exactly ten times as much memory when compared to MCL with 5,000 samples. During global localization, integrating a single sensor scan requires up to 120 seconds using the grid-based approach, whereas MCL consumes consistently less than 3 seconds under otherwise equal conditions. Once the robot has been localized globally, however, grid-based localization updates grid-cells *selectively* as described in (Burgard *et al.* 1998b; Fox 1998), and both approaches are about equally fast.

Vision-based Localization

To test MCL in extreme situations, we evaluated it in a populated public place. During a two-week exhibition, our robot Minerva was employed as a tour-guide in the Smithsonian's Museum of Natural History (Thrun *et al.* 1999). To aid localization, Minerva is equipped with a camera pointed towards the ceiling. Figure 7 shows a mosaic of the museum's ceiling, constructed using a method described in (Thrun *et al.* 1999). The data used here is the most difficult data set in our possession, as the robot traveled with speeds of up to 163 cm/sec. Whenever it entered or left the carpeted area in the center of the museum, it crossed a 2cm bump which introduced significant errors in the robot's odometry. Figure 8 shows the path measured by Minerva's odometry.

When *only* using vision information, grid-based localization fails to track the robot accurately. This is because the computational overhead makes it impossible to incorporate sufficiently many images. MCL, however, succeeded in globally localizing the robot, and tracking the robot's position (see also (Dellaert *et al.* 1999a)). Figure 9 shows the path estimated by our MCL technique. Although the localization error is sometimes above 1 meter, the system is able to keep track of multiple hypotheses and thus to recover from localization errors. The grid-based Markov localization system, however, was not able to track the whole 700m long path of the trajectory. In all our experiments, which were carried out under real-time conditions, the grid-based technique quickly lost track of the robot's position (which, as was verified, would not be the case if the grid-based approach was given unlimited computational power). These results document that MCL is clearly superior to our previous grid-based approach.

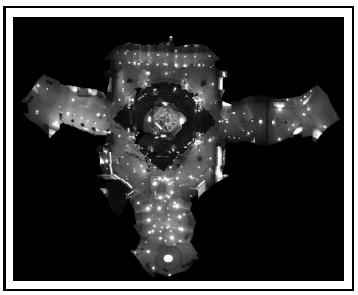

Fig. 7: Ceiling map of the NMAH

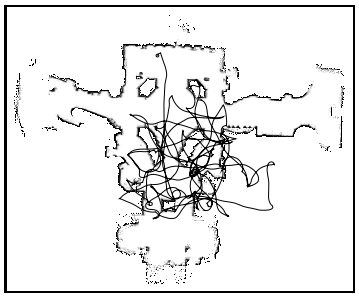

Fig. 8: Odometry information recorded by Minerva on a 700 m long trajectory

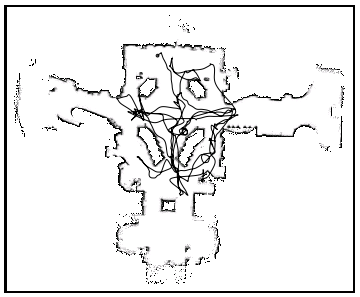

Fig. 9: Trajectory estimated given the ceiling map and the center pixels of on-line images.

Adaptive Sampling

Finally, we evaluated the utility of MCL's *adaptive* approach to sampling. In particular, were were interested in determining the relative merit of the adaptive sampling scheme, if any, over a fixed, static sample set (as used in some of the experiments above and in an earlier version of MCL (Dellaert *et al.* 1999b)). In a final series of experiments, we applied MCL with adaptive and fixed sample set sizes using data recorded with Minerva in the Smithsonian museum. Here we use the laser range data instead of the vision data, to illustrate that MCL also works well with laser range data in environments as challenging as the one studied here.

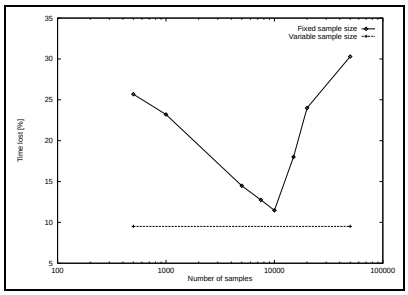

Fig. 10: Localization error for MCL with fixed sample set sizes (top figure) and adaptive sampling (bottom line)

In the first set of experiments we tested the ability of MCL to track the robot as it moved through the museum. In this case it turned out that adaptive sampling has no significant impact on the tracking ability of the Monte Carlo Localization. This result is not surprising since during tracking the position of the robot is concentrated on a small area.

We then evaluated the influence of adapting the sample size on the ability to *globally* localize the robot, and to recover from extreme localization failure. For the latter, we manually introduced severe errors into the data, to test the robustness of MCL in the extreme. In our experiments we "tele-ported" the robot at random points in time to other locations. Technically, this was done by changing the robot's orientation by 180 ± 90 degrees and shifting it by ± 200 cm, without letting the robot know. These perturbations were introduced randomly, with a probability of 0.01 per meter of robot motion. Obviously, such incidents make the robot lose its position, and therefore are well suited to test localization under extreme situations.

Here we found adaptive sampling to be superior to MCL with fixed sample sets. Figure 10 shows the comparison. The top curve depicts the frequency with which the error was larger than 1 meter (our tolerance threshold), for different sample set sizes. The bottom line gives the same result for the adaptive sampling approach. As is easy to be seen, adaptive sampling yields smaller error than the best MCL with fixed sample set sizes. Our results have been obtained by averaging data collected along 700 meters of high-speed robot motion.

Conclusion and Future Work

This paper presented Monte Carlo Localization (MCL), a sample-based algorithm for mobile robot localization. MCL differs from previous approaches in that it uses randomized samples (particles) to represent the robot's belief. This leads to a variety of advantages over previous approaches: A significant reduction in computation and memory consumption, which leads to a higher frequency at which the robot can incorporate sensor data, which in turn implies much higher accuracy. MCL is also much easier to implement than previous Markov localization approaches. Instead of having to reason about entire probability distributions, MCL randomly *guesses* possible positions, in a way that favors likely positions over unlikely ones. An adaptive sampling scheme was proposed that enables MCL to adjust the number of samples in proportion to the amount of surprise in the sensor data. Consequently, MCL uses few samples when tracking the robot's position, but increases the sample set size when the robot loses track of its position, or otherwise is forced to globally localize the robot.

MCL has been tested thoroughly in practice. As our empirical results suggest, MCL beats previous Markov localization methods by an order of magnitude in memory and computation requirements, while yielding significantly more accurate results. In some cases, MCL succeeds where grid-based Markov localization fails.

In future work, the increased efficiency of our sample-based localization will be applied to multi robot scenarios, where the sample sets of the different robots can be synchronized whenever one robot detects another. First experiments conducted with two robots show that the robots are able to

localize themselves much faster when combining their sample sets (Fox *et al.* 1999). Here, the robots were equipped with laser range-finders and cameras to detect each other. We also plan to apply Monte Carlo methods to the problem of map acquisition, where recent work has led to new statistical frameworks that have been successfull applied to large, cyclic environments using grid representations (Thrun, Fox, & Burgard 1998).

Acknowledgment

This research is sponsored in part by NSF, DARPA via TACOM (contract number DAAE07-98-C-L032) and Rome Labs (contract number F30602-98-2-0137), and also by the EC (contract number ERBFMRX-CT96-0049) under the TMR programme.

References

Borenstein, J.; Everett, B.; and Feng, L. 1996. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd..

Burgard, W.; Fox, D.; Hennig, D.; and Schmidt, T. 1996. Estimating the absolute position of a mobile robot using position probability grids. Proc. of AAAI-96.

Burgard, W.; Cremers, A.; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; and Thrun, S. 1998a. The Interactive Museum Tour-Guide Robot. Proc. of AAAI-98.

Burgard, W.; Derr, A.; Fox, D.; and Cremers, A. 1998b. Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach. Proc. of IROS-98.

Carpenter, J.; Clifford, P.; and Fernhead, P. 1997. An improved particle filter for non-linear problems. TR, Dept. of Statistics, Univ. of Oxford.

Chung, K. 1960. *Markov chains with stationary transition probabilities*. Springer.

Cox, I. 1991. Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation* 7(2).

Dean, T. L., and Boddy, M. 1988. An analysis of time-dependent planning. Proc. of AAAI-92.

Dellaert, F.; Burgard, W.; Fox, D.; and Thrun, S. 1999a. Using the condensation algorithm for robust, vision-based mobile robot localization. Proc. of CVPR-99.

Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1999b. Monte Carlo localization for mobile robots. Proc. of ICRA-99.

Doucet, A. 1998. On sequential simulation-based methods for Bayesian filtering. TR CUED/F-INFENG/TR.310, Dept. of Engineering, Univ. of Cambridge.

Endres, H.; Feiten, W.; and Lawitzky, G. 1998. Field test of a navigation system: Autonomous cleaning in supermarkets. Proc. of ICRA-98.

Engelson, S. 1994. *Passive Map Learning and Visual Place Recognition*. Ph.D. Diss., Dept. of Computer Science, Yale University.

Fox, D.; Burgard, W.; Thrun, S.; and Cremers, A. 1998. Position estimation for mobile robots in dynamic environments. Proc. of AAAI-98.

Fox, D.; Burgard, W.; Kruppa, H.; and Thrun, S. 1999. A monte carlo algorithm for multi-robot localization. TR CMU-CS-99-120, Carnegie Mellon University.

Fox, D.; Burgard, W.; and Thrun, S. 1998. Active markov localization for mobile robots. *Robotics and Autonomous Systems* 25:3-4.

Fox, D. 1998. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Naviagation*. Ph.D. Diss, University of Bonn, Germany.

Gelb, A. 1974. *Applied Optimal Estimation*. MIT Press.

Gordon, N.; Salmond, D.; and Smith, A. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F* 140(2).

Gutmann, J.-S., and Schlegel, C. 1996. AMOS: Comparison of scan matching approaches for self-localizati on in indoor environments. Proc. of Euromicro. IEEE Computer Society Press.

Gutmann, J.-S.; Burgard, W.; Fox, D.; and Konolige, K. 1998. An experimental comparison of localization methods. Proc. of IROS-98.

Handschin, J. 1970. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica* 6.

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

Kaelbling, L.; Cassandra, A.; and Kurien, J. 1996. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. Proc. of IROS-96.

Kalman, R. 1960. A new approach to linear filtering and prediction problems. *Tansaction of the ASME–Journal of basic engineering* 35–45.

Kanazawa, K.; Koller, D.; and Russell, S. 1995. Stochastic simulation algorithms for dynamic probabilistic networks. Proc. of UAI-95.

Kitagawa, G. 1996. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* 5(1).

Koller, D., and Fratkina, R. 1998. Using learning for approximation in stochastic processes. Proc. of ICML-98.

Kortenkamp, D.; Bonasso, R.; and Murphy, R., eds. 1997. *AI-based Mobile Robots: Case studies of successful robot systems*. MIT Press.

Leonard, J., and Durrant-Whyte, H. 1992. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic.

Maybeck, P. 1979. *Stochastic Models, Estimation and Control*, Vol. 1. Academic Press.

Nourbakhsh, I.; Powers, R.; and Birchfield, S. 1995. DERVISH an office-navigating robot. *AI Magazine* 16(2).

Rubin, D. 1988. Using the SIR algorithm to simulate posterior distributions. *Bayesian Statistics 3*. Oxford University Press.

Schiele, B., and Crowley, J. 1994. A comparison of position estimation techniques using occupancy grids. Proc. of ICRA-94.

Simmons, R., and Koenig, S. 1995. Probabilistic robot navigation in partially observable environments. Proc. of ICML-95.

Smith, R.; Self, M.; and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. Cox, I., and Wilfong, G., eds., *Autonomous Robot Vehicles*. Springer.

Tanner, M. 1993. *Tools for Statistical Inference*. Springer.

Thrun, S.; Bennewitz, M.; Burgard, W.; Cremers, A.; Dellaert, F.; Fox, D.; Hähnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; and Schulz, D. 1999. MINERVA: A second generation mobile tour-guide robot. Proc. of ICRA-99.

Thrun, S.; Fox, D.; and Burgard, W. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31.

Zilberstein, S., and Russell, S. 1995. Approximate reasoning using anytime algorithms. *Imprecise and Approximate Computation*. Kluwer.