Monte Carlo Localization for Mobile Robots

Frank Dellaert† Dieter Fox† Wolfram Burgard‡ Sebastian Thrun†
†Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213
‡Institute of Computer Science III, University of Bonn, D-53117 Bonn

Abstract

To navigate reliably in indoor environments, a mobile robot must know where it is. Thus, reliable position estimation is a key problem in mobile robotics. We believe that probabilistic approaches are among the most promising candidates to providing a comprehensive and real-time solution to the robot localization problem. However, current methods still face considerable hurdles. In particular, the problems encountered are closely related to the type of representation used to represent probability densities over the robot's state space. Recent work on Bayesian filtering with particle-based density representations opens up a new approach for mobile robot localization, based on these principles. In this paper we introduce the ***M****onte* ***C****arlo* ***L****ocalization method, where we represent the probability density involved by maintaining a set of* samples *that are randomly drawn from it. By using a sampling-based representation we obtain a localization method that can represent arbitrary distributions. We show experimentally that the resulting method is able to efficiently localize a mobile robot without knowledge of its starting location. It is faster, more accurate and less memory-intensive than earlier grid-based methods.*

1 Introduction

Two key problems in mobile robotics are global position estimation and local position tracking. We define global position estimation as the ability to determine the robot's position in an a priori or previously learned map, given no other information than that the robot is somewhere on the map. If no a priori map is available, many applications allow for such a map to be built over time as the robot explores its environment. Once a robot has been localized in the map, local tracking is the problem of keeping track of that position over time. Both these capabilities are necessary to enable a robot to execute useful tasks, such as office delivery or providing tours to museum visitors. By knowing its global position, the robot can make use of the existing maps, which allows it to plan and navigate reliably in complex environments. Accurate local tracking on the other hand, is useful for efficient navigation and local manipulation tasks. Both these sub-problems are of fundamental importance to building truly autonomous robots.

We believe that probabilistic approaches are among the most promising candidates to providing a comprehensive and real-time solution to the robot localization problem, but current methods still face considerable hurdles. Kalman-filter based techniques have proven to be robust and accurate for keeping track of the robot's position. However, a Kalman filter cannot represent ambiguities and lacks the ability to globally (re-)localize the robot in the case of localization failures. Although the Kalman filter can be amended in various ways to cope with some of these difficulties, recent approaches [1, 2, 3, 4, 5] have used richer schemes to represent uncertainty, moving away from the restricted Gaussian density assumption inherent in the Kalman filter. In previous work [5] we introduced the grid-based Markov localization approach which can represent arbitrarily complex probability densities at fine resolutions. However, the computational burden and memory requirements of this approach are considerable. In addition, the grid-size and thereby also the precision at which it can represent the state has to be fixed beforehand.

In this paper we present the **M**onte **C**arlo **L**ocalization method (which we will denote as the MCL-method) where we take a different approach to representing uncertainty: instead of describing the probability density function itself, we represent it by maintaining a set of *samples* that are randomly drawn from it. To update this density representation over time, we make use of Monte Carlo methods that were invented in the seventies [6], and recently rediscovered independently in the target-tracking [7], statistical [8] and computer vision literature [9, 10].

By using a sampling-based representation we obtain a localization method that has several key advantages with respect to earlier work:

1. In contrast to 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 it

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 easy to implement.

The remainder of this paper is organized as follows: in the next section (Section 2) we introduce the problem of localization as an instance of the Bayesian filtering problem. Then, in Section 3, we discuss existing approaches to position estimation, focusing on the type of density representation that is used. In Section 4, we describe the Monte Carlo localization method in detail. Finally, Section 5 contains experimental results illustrating the various properties of the MCL-method.

2 Robot Localization

In robot localization, we are interested in estimating the state of the robot at the current time-step k, given knowledge about the initial state and all measurements $Z^k = \{\mathbf{z}_k, i = 1..k\}$ up to the current time. Typically, we will work with a three-dimensional state vector $\mathbf{x} = [x, y, \theta]^T$, i.e. the position and orientation of the robot. This estimation problem is an instance of the Bayesian filtering problem, where we are interested in constructing the posterior density $p(\mathbf{x}_k|Z^k)$ of the current state conditioned on all measurements. In the Bayesian approach, this probability density function (PDF) is taken to represent all the knowledge we possess about the state $\mathbf{x}_k$, and from it we can estimate the current position. Often used estimators are the mode (the maximum a posteriori or MAP estimate) or the mean, when the density is unimodal. However, particularly during the global localization phase, this density will be multi-modal and calculating a single position estimate is not appropriate.

Summarizing, to localize the robot we need to recursively compute the density $p(\mathbf{x}_k|Z^k)$ at each time-step. This is done in two phases:

Prediction Phase In the first phase we use a *motion model* to predict the current position of the robot in the form of a predictive PDF $p(\mathbf{x}_k|Z^{k-1})$, taking only motion into account. We assume that the current state $\mathbf{x}_k$ is only dependent on the previous state $\mathbf{x}_{k-1}$ (Markov) and a known control input $\mathbf{u}_{k-1}$, and that the motion model is specified as a conditional density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$. The predictive density over $\mathbf{x}_k$ is then obtained by integration:

$$p(\mathbf{x}_k|Z^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})\, p(\mathbf{x}_{k-1}|Z^{k-1})\, d\mathbf{x}_{k-1} \tag{1}$$

Update Phase In the second phase we use a *measurement model* to incorporate information from the sensors to obtain the posterior PDF $p(\mathbf{x}_k|Z^k)$. We assume that the measurement $\mathbf{z}_k$ is conditionally independent of earlier measurements Z^{k-1} given $\mathbf{x}_k$, and that the measurement model is given in terms of a likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$. This term expresses the likelihood that the robot is at location $\mathbf{x}_k$ given that $\mathbf{z}_k$ was observed. The posterior density over $\mathbf{x}_k$ is obtained using Bayes theorem:

$$p(\mathbf{x}_k|Z^k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|Z^{k-1})}{p(\mathbf{z}_k|Z^{k-1})} \tag{2}$$

After the update phase, the process is repeated recursively. At time t_0 the knowledge about the initial state $\mathbf{x}_0$ is assumed to be available in the form of a density $p(\mathbf{x}_0)$. In the case of global localization, this density might be a uniform density over all allowable positions. In tracking work, the initial position is often given as the mean and covariance of a Gaussian centered around $\mathbf{x}_0$. In our work, as in [11], the transition from global localization to tracking is automatic and seamless, and the PDF evolves from spanning the whole state space to a well-localized peak.

3 Existing Approaches: A Tale of Density Representations

The solution to the robot localization problem is obtained by recursively solving the two equations (1) and (2). Depending on how one chooses to represent the density $p(\mathbf{x}_k|Z^k)$, one obtains various algorithms with vastly different properties:

The Kalman filter If both the motion and the measurement model can be described using a Gaussian density, and the initial state is also specified as a Gaussian, then the density $p(\mathbf{x}_k|Z^k)$ will remain Gaussian at all times. In this case, equations (1) and (2) can be evaluated in closed form, yielding the classical Kalman filter [12]. Kalman-filter based techniques [13, 14, 15] have proven to be robust and accurate for keeping track of the robot's position. Because of its concise representation (the mean and covariance matrix suffice to describe the entire density) it is also a particularly efficient algorithm. However, in its pure form, the Kalman filter does not correctly handle non-linear or non-Gaussian motion and measurement models, is unable to recover from tracking failures, and can not deal with multi-modal densities as encountered during global localization. Whereas non-linearities, tracking failure and even multi-modal densities can be accomodated using non-optimal extensions of the Kalman filter, most of these difficulties stem from the the restricted Gaussian density assumption inherent in the Kalman filter.

Topological Markov Localization To overcome these disadvantages, different approaches have used increasingly richer schemes to represent uncertainty. These different methods can be roughly distinguished by the type of discretization used for the representation of the state space. In [1, 2, 3, 4], Markov localization is used for landmark-based corridor navigation and the state space is organized according to the topological structure of the environment.

Grid-based Markov Localization To deal with multi-modal and non-Gaussian densities at a fine resolution (as opposed to the coarser discretization in the above methods), one can perform numerical integration over a grid of points. This involves discretizing the interesting part of the state space, and use it as the basis for an approximation of the density $p(\mathbf{x}_k|Z^k)$, e.g. by a piece-wise constant function [16]. This idea forms the basis of our previously introduced grid-based Markov localization approach (see [5, 11]). Methods that use this type of representation are powerful, but suffer from the disadvantages of computational overhead and *a priori* commitment to the size 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 [11] has begun to address some of these problems, using octrees 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 to some extent the limitation of fixed accuracy.

Sampling-based Methods Finally, one can represent the density by a set of samples that are randomly drawn from it. This is the representation we will use, and it forms the topic of the next section.

4 Monte Carlo Localization

In sampling-based methods one represents the density $p(\mathbf{x}_k|Z^k)$ by a set of N random samples or *particles* $S_k = \{s_k^i; i = 1..N\}$ drawn from it. We are able to do this because of the essential duality between the samples and the density from which they are generated [17]. From the samples we can always approximately reconstruct the density, e.g. using a histogram or a kernel based density estimation technique.

The goal is then to recursively compute at each time-step k the set of samples S_k that is drawn from $p(\mathbf{x}_k|Z^k)$. A particularly elegant algorithm to accomplish this has recently been suggested independently by various authors. It is known alternatively as the bootstrap filter [7], the Monte-Carlo filter [8] or the Condensation algorithm [9, 10]. These methods are generically known as *particle filters*, and an overview and discussion of their properties can be found in [18].

In analogy with the formal filtering problem outlined in Section 2, the algorithm proceeds in two phases:

Prediction Phase In the first phase we start from the set of particles S_{k-1} computed in the previous iteration, and apply the motion model to each particle s_{k-1}^i by sampling from the density $p(\mathbf{x}_k|s_{k-1}^i, \mathbf{u}_{k-1})$:

(i) for each particle s_{k-1}^i:
    draw one sample ${s'}_k^i$ from $p(\mathbf{x}_k|s_{k-1}^i, \mathbf{u}_{k-1})$

In doing so a new set S'_k is obtained that approximates a random sample from the predictive density $p(\mathbf{x}_k|Z^{k-1})$. The prime in S'_k indicates that we have not yet incorporated any sensor measurement at time k.

Update Phase In the second phase we take into account the measurement $\mathbf{z}_k$, and weight each of the samples in S'_k by the weight $m_k^i = p(\mathbf{z}_k|{s'}_k^i)$, i.e. the likelihood of ${s'}_k^i$ given $\mathbf{z}_k$. We then obtain S_k by resampling from this *weighted* set:

(ii) for j=1..N:
    draw one S_k sample s_k^j from $\{{s'}_k^i, m_k^i\}$

The resampling selects with higher probability samples ${s'}_k^i$ that have a high likelihood associated with them, and in doing so a new set S_k is obtained that approximates a random sample from $p(\mathbf{x}_k|Z^k)$. An algorithm to perform this resampling process efficiently in O(N) time is given in [19].

After the update phase, the steps (i) and (ii) are repeated recursively. To initialize the filter, we start at time $k = 0$ with a random sample $S_0 = \{s_0^i\}$ from the prior $p(\mathbf{x}_0)$.

4.1 A Graphical Example

One iteration of the algorithm is illustrated in Figure 1. In the figure each panel in the top row shows the exact density, whereas the panel below shows the particle-based representation of that density. In panel A, we start out with a cloud of particles S_{k-1} representing our uncertainty about the robot position. In the example, the robot is fairly localized, but its orientation is unknown. Panel B shows what happens to our belief state when we are told the robot has moved exactly one meter since the last time-step: we now know the robot to be somewhere on a circle of 1 meter radius around the previous location. Panel C shows what happens when we observe a landmark, half a meter away, somewhere in the top-right corner: the top panel shows the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$, and the bottom panel illustrates how each sample ${s'}_k^i$ is weighted according to this likelihood.

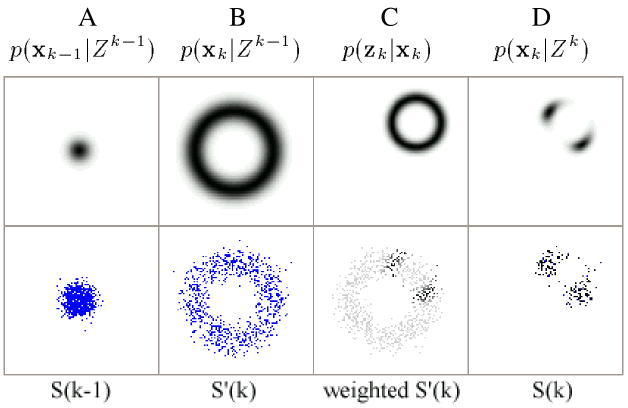

Fig. 1: The probability densities and particle sets for one iteration of the algorithm. See text for detail.

Finally, panel D shows the effect of resampling from this weighted set, and this forms the starting point for the next iteration.

4.2 Theoretical Justification

Good explanations of the mechanism underlying the elegant and simple algorithm sketched above are given in [19, 20]. We largely follow their exposition below:

Prediction Phase To draw an approximately random sample from the exact predictive PDF $p(\mathbf{x}_k|Z^{k-1})$, we use the motion model and the set of particles S_{k-1} to construct the *empirical predictive density* [20]:

$$\hat{p}(\mathbf{x}_k|Z^{k-1}) = \sum_{i=1}^{N} p(\mathbf{x}_k|s_{k-1}^i, \mathbf{u}_{k-1}) \tag{3}$$

Equation (3) describes a mixture density approximation to $p(\mathbf{x}_k|Z^{k-1})$, consisting of one equally weighted mixture component $p(\mathbf{x}_k|s_{k-1}^i, \mathbf{u}_{k-1})$ per sample s_{k-1}^i. To sample from this mixture density, we use *stratified sampling* and draw exactly one sample ${s'}_k^i$ from each of the N mixture components to obtain S'_k.

Update Phase In the second phase we would like to use the measurement model to obtain a sample S'_k from the posterior $p(\mathbf{x}_k|Z^k)$. Instead we will use Eq. (3) and sample from the *empirical posterior density*:

$$\hat{p}(\mathbf{x}_k|Z^k) \propto p(\mathbf{z}_k|\mathbf{x}_k)\hat{p}(\mathbf{x}_k|Z^{k-1}) \tag{4}$$

This is accomplished using a technique from statistics called *importance sampling*. It is used to obtain a sample from a difficult to sample density $p(x)$ by instead sampling from an easier density $f(x)$. In a corrective action, each sample is then re-weighted by attaching the *importance weight* $w = p(x)/f(x)$ to it. In the context of the particle filter, we would like to sample from $p(x) = \hat{p}(\mathbf{x}_k|Z^k)$, and we use as importance function $f(x) = \hat{p}(\mathbf{x}_k|Z^{k-1})$, as we have already obtained a random sample S'_k from it in the prediction step. We then reweight each sample by:

$$m_k^i = \frac{g(x)}{f(x)} = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\hat{p}(\mathbf{x}_k|Z^{k-1})}{\hat{p}(\mathbf{x}_k|Z^{k-1})} = p(\mathbf{z}_k|\mathbf{x}_k)$$

The subsequent resampling is needed to convert the set of weighted or non-random samples back into a set of equally weighted samples $S_k = \{s_k^i\}$.

The entire procedure of sampling, reweighting and subsequently resampling to sample from the posterior is called *Sampling/Importance Resampling* (SIR), and is discussed in more depth in [17].

5 Experimental Results

Fig. 2: The robots RHINO (left) and MINERVA (right) used for the experiments.

The Monte Carlo localization technique has been tested extensively in our office environment using different robotic platforms. In all these applications our approach has shown to be both efficient and robust, running comfortably in real-time. In order to test our technique under more challenging circumstances, the experiments described here are based on data recorded from RHINO, an RWI B21 robot, and MINERVA, an RWI B18 robot (see Figure 2). While the data collected by RHINO was taken in a typical office environment, MINERVA's datasets consist of logs recorded during a deployment of the robot as a museum tour-guide in the Smithsonian's National Museum of American History. Although the data was collected at an earlier time the time-stamps in the logs were used to recreate the real-time datastream coming from the sensors, so that the results do not differ from results obtained on the real robots.

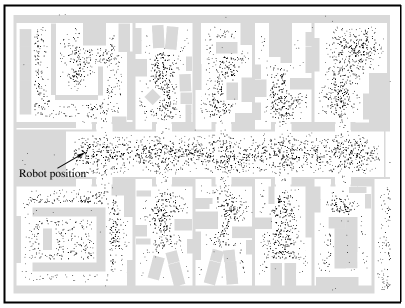

Fig. 3: Global localization: Initialization.

Fig. 4: Ambiguity due to symmetry.

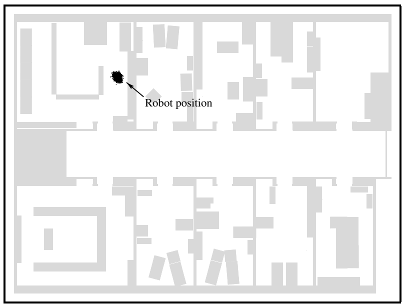

Fig. 5: Achieved localization.

5.1 Global Localization

One of the key advantages of the MCL-method over Kalman-filter based approaches is its ability to represent multi-modal probability distributions. This ability is a precondition for localizing a mobile robot from scratch, i.e. without knowledge of its starting location. The global localization capability of the MCL-method is illustrated in Figs. 3 to 5. In this particular experiment, we used the sonar readings recorded from RHINO in a department of the University of Bonn, Germany. In the first iteration, the algorithm is initialized by drawing 20,000 samples from a uniform probability density save where there are known to be (static) obstacles. The robot started in the left corner of the corridor and the distribution of the samples after the first scan of sonar measurements is observed, is shown in Figure 3. As the robot enters the upper left room (see Figure 4), the samples are already concentrated around two positions. One is the true location of the robot and the other occurs due to the symmetry of the corridor (imagine the robot moving into the lower right room). In addition, a few scattered samples survive here and there. It should be noted that in this early stage of localization, the ability to represent ambiguous probability distributions is vital for successful position estimation. Finally, in the last figure (Figure 5), the robot has been able to uniquely determine its position because the upper left room looks (to the sonars) different from the symmetrically opposed room.

5.2 Accuracy of Position Tracking

To compare the accuracy of the Monte Carlo method with our earlier grid-based approach, we again used data recorded from RHINO. Figure 6 shows the test environment with the path taken by the robot. The figure also depicts 22 reference points for which we determined the accurate positions of the robot on its path (this data has also been used for accuracy tests in [11, 21]). We conducted four evaluations of the laser and the sonar measurements

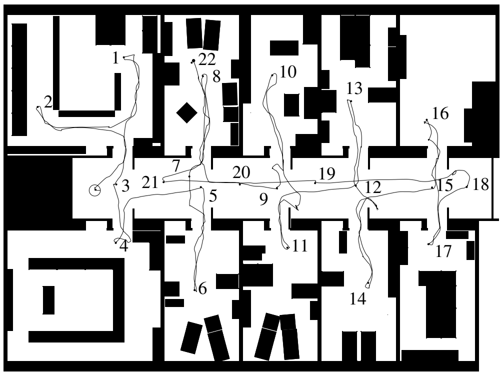

Fig. 6: Path of the robot and reference positions

with small corruptions on the odometry data to get statistically significant results. The average distance between the estimated positions and the reference positions using the grid-based localization approach is shown in Figure 7, as a function of cell size (the error-bars provide 95% confidence intervals). As is to be expected, the error increases with increasing cell size (see [11] for a detailed discussion).

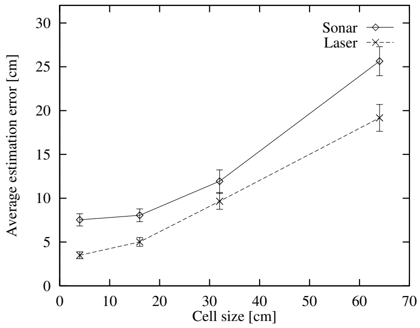

Fig. 7: Accuracy of grid-based Markov localization using different spatial resolutions.

We also ran our MCL-method on the recordings from the same run, while varying the number of samples used to represent the density. The result is shown in Figure 8. It can be seen that the accuracy of our MCL-method can

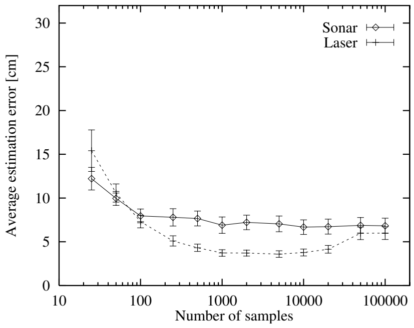

Fig. 8: Accuracy of MCL-method for different numbers of samples (log scale).

only be reached by the grid-based localization when using a cell size of 4cm. Another salient property of this graph is the trade-off between increased representational power and computational overhead. Initially, the accuracy of the method increases with the number of samples (shown on a log scale for clarity). However, as increased processing time per iteration causes available measurements to be discarded, less information is integrated into the posterior densities and the accuracy goes down.

5.3 National Museum of American History

The MCL-method is also able to track the position of a robot for long periods of time, even when using inaccurate occupancy grid maps and when the robot is moving at high speeds. In this experiment, we used recorded laser data from the robotic tour-guide MINERVA, as it was moving with speeds up to 1.6 m/sec through the Smithsonian's National Museum of American History. At the time of this run there were no visitors in the museum, and the robot was remotely controlled by users connected through the world wide web[1].

Tracking performance is illustrated in Figure 9, which shows the occupancy grid map of the museum used for localization along with the trajectory of the robot (the area shown is about 40 by 40 meters). This run lasted for 75 minutes with the robot traveling over 2200 meters, during which the algorithm never once lost track. For this particular tracking experiment, we used 5000 samples. In general, far fewer samples are needed for position tracking than for global localization, and an issue for future research is to adapt the number of samples appropriately.

[1] See also `http://www.cs.cmu.edu/~minerva`

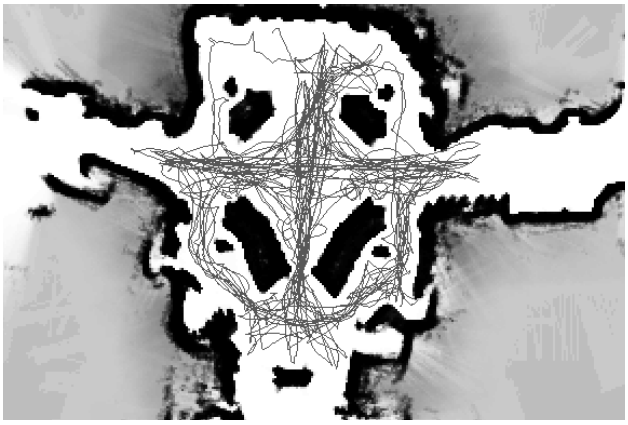

Fig. 9: A laser-based map of the Smithsonian museum with a succesful track of over 2 km.

Global localization in this environment behaved equally impressive: using MCL, the robot's location was uniquely determined in less than 10 seconds. This level of performance can only be obtained with the grid-based Markov localization when using very coarse grids, which are unsuitable for tracking purposes. To gain the same flexibility as the MCL-method, a variable resolution approach is needed, as proposed in [11] (see also discussion below).

6 Conclusion and Future Work

In this work we introduced a novel approach to mobile robot position estimation, the Monte Carlo localization method. As in our previous grid-based Markov localization work, we represent probability densities over the entire state space. Instead of directly approximating this density function we represent it by a set of samples randomly drawn from it. Recent research on propagating and maintaining this representation over time as new measurements arrive, made this technique applicable to the problem of mobile robot localization.

By using Monte Carlo type methods, we have combined the advantages of grid-based Markov localization with the efficiency and accuracy of Kalman filter based techniques. As with grid-based methods, we are able to represent arbitrary probability densities over the robot's state space. Therefore, the MCL-method is able to deal with ambiguities and thus can *globally* localize a robot. By concentrating the computational resources (samples) on the relevant parts of the state space, our method can *efficiently* and *accurately* estimate the position of the robot.

Compared to our previous grid-based method, this approach has significantly reduced memory requirements while at the same time incorporating sensor measurements at a considerably higher frequency. Grid-based Markov localization requires dedicated techniques for achieving the same efficiency or increasing the accuracy. Recent work, for example [11], uses octrees to reduce the space and time requirements of Markov localization. However, this technique has a significant overhead (in space, time, and programming complexity) based on the nature of the underlying data structures.

Even though we obtained promising results with our technique, there are still warrants for future work. One potential problem with the specific algorithm we used is that of sample impoverishment: in the resampling step, samples s_k^i with high weight will be selected multiple times, resulting in a loss of 'diversity' [18]. Several improvements to the basic algorithm have recently been suggested [19, 20, 18], and it makes sense to see whether they would also improve localization performance.

In future work, the reduced memory requirements of the algorithm will allow us to extend the robot's state with velocity information, possibly increasing the tracking performance. One can even extend the state with discrete variables indicating the mode of operation of the robot (e.g. cruising, avoiding people, standing still), enabling one to select a different motion model for each mode. This idea has been explored with great success in the visual tracking literature [22], and it might further improve localization performance. In addition, this would also allow us to generate symbolic descriptions of the robot's behavior.

Acknowledgments

The authors would like to thank Michael Isard for his helpful comments.

References

[1] I. Nourbakhsh, R. Powers, and S. Birchfield, "DERVISH an office-navigating robot," *AI Magazine* **16**, pp. 53–60, Summer 1995.

[2] R. Simmons and S. Koenig, "Probabilistic robot navigation in partially observable environments," in *Proc. International Joint Conference on Artificial Intelligence*, 1995.

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

[4] S. Thrun, "Bayesian landmark learning for mobile robot localization," *Machine Learning* , 1998. To appear.

[5] 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 Fourteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 896–901, 1996.

[6] J. E. Handschin, "Monte Carlo techniques for prediction and filtering of non-linear stochastic processes," *Automatica* **6**, pp. 555–563, 1970.

[7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Procedings F* **140**(2), pp. 107–113, 1993.

[8] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics* **5**(1), pp. 1–25, 1996.

[9] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *European Conference on Computer Vision*, pp. 343–356, 1996.

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

[11] 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[12] P. Maybeck, *Stochastic Models, Estimation and Control*, vol. 1, Academic Press, New York, 1979.

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

[14] B. Schiele and J. L. Crowley, "A comparison of position estimation techniques using occupancy grids," in *Proceedings of IEEE Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1628–1634, 1994.

[15] J.-S. Gutmann and C. Schlegel, "Amos: Comparison of scan matching approaches for self-localization in indoor environments," in *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*, IEEE Computer Society Press, 1996.

[16] R. S. Bucy, "Bayes theorem and digital realisation for nonlinear filters," *Journal of Astronautical Science* **17**, pp. 80–94, 1969.

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

[18] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge, 1998.

[19] J. Carpenter, P. Clifford, and P. Fernhead, "An improved particle filter for non-linear problems," tech. rep., Department of Statistics, University of Oxford, 1997.

[20] M. K. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filters," tech. rep., Department of Mathematics, Imperial College, London, October 1997.

[21] 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 (IROS'98)*, 1998.

[22] M. Isard and A. Blake, "A mixed-state Condensation tracker with automatic model-switching," in *European Conference on Computer Vision*, 1997.