Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach

Wolfram Burgard Andreas Derr Dieter Fox Armin B. Cremers
Institute of Computer Science III, University of Bonn, Römerstr. 164, D-53117 Bonn

Abstract

Localization is one of the fundamental problems of mobile robots. In order to efficiently perform useful tasks such as office delivery, mobile robots must know their position in their environment. Existing approaches can be distinguished according to the type of localization problem they are designed to solve. Tracking techniques aim at monitoring the robot's position. They assume that the position is initially known and cannot recover from situations in which they lost track of the robot's position. Global localization techniques, on the other hand, are able to estimate the robot's position under complete uncertainty. In this paper we present the dynamic Markov localization technique as a uniform approach to position estimation, which is able (1) to globally estimate the position of the robot, (2) to efficiently track its position whenever the robot's certainty is high, and (3) to detect and recover from localization failures. The approach has been implemented and intensively tested in real-world environments. We present several experiments illustrating the strength of our method.

1. Introduction

The estimation of the robot's position is one of the fundamental problems in mobile robotics. Knowledge of its position is necessary for it to use existing or previously learned maps of the environment as well as to perform useful tasks such as office delivery. Existing techniques for estimating the position of a mobile robot can be roughly divided into two classes [2]: methods for keeping track of the robot's position and methods for global position estimation. Most of the research carried out so far has concentrated on the first class which assumes that the initial position of the robot is known and aims to compensate the accumulated odometry error of the robot while it performs its tasks (see [2] for a comprehensive overview). Although several robust and even accurate methods have been developed, tracking techniques have two important weaknesses.

- Tracking techniques are not able to globally estimate the position of the robot, which is obviously highly important for truly autonomous robots. Instead, tracking systems require the starting position of the robot to be known in advance.
- Since tracking methods are not inherently able to detect and recover from localization failures, they require to be monitored by human operators and must be stopped whenever a localization failure occurs.

The second class contains methods for globally estimating a robot's position. These methods allow robots to be started even under complete uncertainty with respect to their initial position. This capability is a basic precondition for truly autonomous robots.

Tracking techniques and methods for global localization can be distinguished by the state space in which they operate. While tracking techniques maintain and update only a small state space generally centered around the estimated position of the robot, global position estimation methods deal with the complete configuration space. Consequently, since they reduce the search space, tracking techniques are more efficient than global localization methods but they lack the crucial ability to recover from localization failures.

In this paper we present the **D**ynamic **M**arkov **L**ocalization method (which we will denote as the DML-method) which combines the advantages of both approaches. The DML-method is a global localization approach that uses a probabilistic technique to reduce the state space during localization thus achieving the efficiency of position tracking methods. Moreover, it can determine when the robot lost track of its position. By using a dynamic representation technique for the state space, the memory requirements are adapted according to the robot's certainty in its position. Thus, the DML-method builds a uniform framework for global position estimation and efficient position tracking and simultaneously inherits the advantages of both methods.

This paper is organized as follows. After discussing related work in the following section, we describe the basic Markov localization technique together with some of its variants in Section 3. In Section 4 we describe the DML-approach. Finally, Section 5 contains experiments illustrating various aspects of the DML-method.

2. Related Work

The problem of mobile robot localization has been intensively studied in the past and a variety of systems and techniques have been developed (see [2] for a comprehensive overview). Recently there has been increasing interest in probabilistic methods. Several systems apply Kalman filters to keep track of the robot's position [10, 13, 7, 1]. Kalman filter techniques have been proven to be robust and accurate for keeping track of the robot's position. However, they cannot represent ambiguities and lack the ability to globally (re-)localize the robot in the case of localization failures.

To overcome these disadvantages, recently different variants of Markov localization have been developed and employed successfully [11, 14, 8, 5, 15]. The basic idea of Markov localization is to maintain a position probability density over the whole three-dimensional $\langle x, y, \theta\rangle$ state space of the robot in its environment. This density, which is not restricted to a Gaussian as in Kalman filter techniques, is updated whenever the robot moves or receives new information from its sensors. The different variants of this technique can be roughly distinguished by the type of discretization used for the representation of the state space. In [11, 14, 8, 15] Markov localization is used for landmark-based corridor navigation and the state space is organized according to the topological structure of the environment. Based on an orthogonality assumption [11, 14, 8] consider only four possible headings of the robot.

In [5] we proposed a fine-grained grid-based discretization of the state space. The advantage of this approach is that it provides accurate position estimates and that it can be applied in arbitrary unstructured and even densely populated environments [3, 6]. The disadvantage of this approach, however, is the huge state space which has to be maintained. The DML-approach presented in this paper overcomes this problem because it dynamically adopts the size of the state space according to the robot's certainty in its position. It is able to globally localize the robot whenever necessary and to efficiently keep track of the robot's position in normal situations in which the robot has almost certain knowledge about its location. Moreover, and in contrast to all Markov localization techniques presented so far, the DML-method is able to decrease the estimation error by dynamically enhancing the resolution of the state space. As a consequence, DML is a uniform approach to localization, which inherits the benefits and the advantages of both global localization and tracking techniques.

3. Markov Localization

The key idea of Markov localization is to compute and update a probability distribution over all possible locations in the environment. Let $l = \langle x, y, \theta\rangle$ denote a location in the state space of the robot. The distribution, denoted by $P(L_t = l)$, expresses the robot's subjective belief for being at position l at time t. Initially, $P(L_{t_0})$ reflects the initial state of knowledge: if the robot knows its initial position, $P(L_{t_0})$ is centered on the correct location; if the robot does not know its initial location, $P(L_{t_0})$ is uniformly distributed to reflect the global uncertainty of the robot. $P(L)$ is updated whenever the robot receives new sensor readings. These sensor readings generally come from the odometry or from sensors measuring features of the environment such as proximity sensors or cameras.

Robot motion is modelled by a conditional probability $p(l \mid l', a) = P(L_{t+1} = l \mid L_t = l', A_t = a)$ specifying the probability that a measured movement action a, when executed at l', carries the robot to l. $P(L_{t+1} = l)$ is then computed according to the following general formula coming from the domain of Markov chains:

$$P(L_{t+1} = l) \quad = \quad \sum_{l'} p(l \mid l', a) P(L_t = l') \qquad (1)$$

The term $p(l \mid l', a)$ represents a model of the robot's kinematics, which in our current implementation is a bounded Gaussian centered around l.

Sensor readings are integrated according to the well-known Bayesian update formula. Let s denote a sensor reading and $p(s \mid l) = P(S_t = s \mid L_t = l)$ be the likelihood of perceiving s given that the robot is at position l. Thus, after receiving s, $P(L_{t+1})$ is computed according to the following rule:

$$P(L_{t+1} = l) \quad = \quad \frac{p(s \mid l)\, P(L_t = l)}{P(S_t = s)} \qquad (2)$$

The term $P(S_t = s)$ is a normalizer which ensures that $P(L_{t+1} = l)$ sums up to 1 over all l.

Our method uses a fine-grained discretization of the state space. While the resolution of robot orientation is typically of the order of 1° to 5°, the longitudinal resolution varies from 4 to 64cm. To provide accurate position estimates with a finer resolution than the cell size we compute the maximum likelihood estimate l^* by integrating over all neighbor cells in a small region $R(l_m)$ around the global maximum l_m of the state space:

$$l^* \quad = \quad \frac{\sum_{l \in R(l_m)} l * p(L_t = l)}{\sum_{l \in R(l_m)} p(L_t = l)} \qquad (3)$$

In contrast to the topological approaches described [11, 14, 8], which use predefined landmarks to compute $p(s \mid l)$, we obtain this likelihood directly from a metric model of the environment and a model of proximity sensors [4]. The advantage of this approach is that it can operate based on the raw data of the proximity sensors and thus permits the exploitation of arbitrary geometric features of the environment such as the width of a corridor or the size of a cupboard. However, it can easily be extended to incorporate the abstract features or landmarks used in [11, 14, 8].

4. Dynamic Markov localization

The disadvantage of our grid-based method lies in the huge state space which has to be maintained. For an environment of size $30 \times 30\text{m}^2$, a cell size of $15 \times 15\text{cm}^2$, and an angular resolution of 2° the state space consists of $7,200,000$ states. The DML-approach presented here is motivated by the observation that in typical situations, in which the system is almost certain about the robot's position, the major part of the probability density is centered on the true position of the robot while the remaining states have vanishingly small probabilities. Therefore, the DML-system uses an octree-based representation of the state space permitting dynamic expansion, refinement and shrinking according to the current degree of belief. It thus focuses on the relevant part of the state space. Simultaneously, it estimates the likelihood, that the robot's position is not contained in the currently considered states. This way, it is able to determine, that it lost track of the robot's position.

The first optimization of the DML-approach is the *selective update strategy*, which excludes unlikely states from being updated. For this purpose, we introduce a threshold θ, which is 10^{-10} times the average a priori position probability in our current implementation. States for which $P(L_t = l) > \theta$ are updated in a different way than states which have $P(L_t = l) \leq \theta$. The basic idea is to assign the same small probability to all unlikely states. This way we have to apply Formulas 1 and 2 *only once* to update the probabilities of *all* unlikely states.

4.1. Reducing the Computation Time

4.1.1. Integration of Sensory Input

To compute the new belief state given new sensory input, we approximate $p(s \mid l)$ for cells with $P(L = l) \leq \theta$ by $\tilde{p}(s)$, which is the average or a priori probability of measuring the feature s given a uniform distribution over all possible locations. Thus, the formula to integrate new sensory input changes to:

$$P(L_{t+1} = l) \quad = \quad \begin{cases} \frac{p(s|l)P(L_t=l)}{P(S_t=s)} & \text{if } P(L = l) > \theta \\ \frac{\tilde{p}(s)P(L_t=l)}{P(S_t=s)} & \text{otherwise} \end{cases} \tag{4}$$

Please note that $\tilde{p}(s)$ differs from $P(S_t = s)$, which is the probability of measuring s given the current belief state of the robot. Obviously, this update rule is equivalent to

$$P(L_{t+1} = l) = \begin{cases} \alpha_t \, \frac{p(s|l)}{\tilde{p}(s)} \, P(L_t = l) & \text{if } P(L_t = l) > \theta \\ \alpha_t \, P(L_t = l) & \text{otherwise} \end{cases} \tag{5}$$

where α_t is a normalizer, ensuring that $P(L_{t+1} = l)$ sums up to one over all l.

Instead of updating all unlikely cells we update a single variable u_t containing the probability, that the robot's position is not included in the active states. This cell u_t, which is initialized by 0, accumulates the probability of all unlikely positions. Whenever a cell is deactivated, u_t is increased by the likelihood of that cell. Measurements are incorporated by multiplying u_t by the normalization factor α_t:

$$u_{t+1} \quad = \quad \alpha_t \cdot u_t \tag{6}$$

In essence, u_t represents the likelihood, that the actual state of the robot is not contained in the cells with $P(L) > \theta$. Thus, whenever u_t exceeds a given threshold, which is 0.001 in our implementation, all unlikely states are again considered as possible states of the robot.

4.1.2. Integration of Odometric Information

While measurements generally increase the certainty in $P(L)$, movements generally decrease it because of the inherent uncertainty of the odometry. In Markov localization systems the motions are modelled by Equation 1. Since we assume that the odometric errors can be approximated by a bounded Gaussian, this equation is equivalent to shifting all cells according to the measured movement combined with an application of a Gaussian kernel. When applying this kernel to the likely cells of $P(L)$, only a certain number of unlikely states lies within the scope of the kernel and are thus re-activated after each movement. Since the likelihood $p(s \mid l)$ is generally small for unlikely cells these reactivated states become inactive again after incorporating the next sensor readings. Accordingly, robot motion generally decreases the certainty of the positioning system and increases the number of active states, whereas sensing generally increases the certainty and decreases the number of states.

4.2. Reducing the Space Requirements

In all approaches to Markov localization, which have been developed so far, the discretization of the state space is fixed and is left unchanged during operation. Increasing the resolution, however, is advantageously if fine-motor manipulations have to be carried out. On the other hand, in order to speed-up the global self-localization process a coarser partitioning of the state space is recommended. Our technique for representing the state space is motivated by octrees, which are a hierarchical data structure “based on the principle of recursive decomposition of space” [12]. The root node of the tree corresponds to the entire state space of the robot. Each son of the a node represents one of eight cubes obtained by subdividing the node in octants. Our tree structure distinguishes two kinds of nodes: leaf nodes and nonleaf nodes. A leaf node corresponds to a single state of the robot and contains the probability of being at the corresponding position. A nonleaf node only contains pointers to its eight sons. The pointer to a son is *nil* if that son contains no state with $P(L_t = l) > \theta$.

The advantage of this data structure is (1) that the space requirement is adopted automatically according to the number of likely positions and (2) that the resolution of the state space can be easily increased or even decreased. An important aspect of this tree structure is that all necessary operations, which are the integration of new sensory input, the normalization, and the convolution, can be performed by tree traversals and thus require $O(n)$ steps where n is the number of leaf nodes [12].

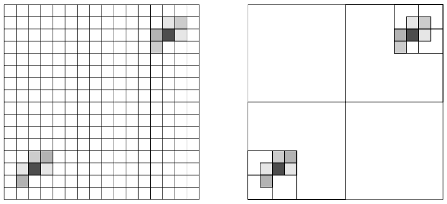

Fig. 1: Memory requirements of the grid- and tree-based representations in an ambiguous situation

Figure 1 illustrates the advantage of our tree-based representation over a grid-based representation. For the sake of clarity, this figure only shows a two-dimensional grid and the corresponding two-dimensional variant of our tree structure. The grey-shaded fields are the likely states of the robot (more likely positions are darker). White fields in the tree correspond to inner nodes which are not sub-divided because all their sons represent unlikely states. In this situation the state space contains two local maxima.

5. Experimental Results

The DML-approach has been implemented and intensively tested using our mobile robot RHINO [16], which is an RWI B21 robot equipped with a ring of 24 ultrasonic sensors and two laser-range finders. The experiments described below focus on the following aspects of the DML-approach: (1) global (re-)localization ability, (2) dynamic state space representation and representation of ambiguities, and (3) the relationship between the localization accuracy and the cell size.

5.1. Global Localization

The first experiment is designed to demonstrate the ability of the DML-approach to globally (re-)localize the robot within its environment. This ability is important after localization failures which for example occur after bumping into an obstacle. Throughout the experiment, which was carried out in a $27m \times 20m$ large section of our department, only the ultrasound sensors were used.

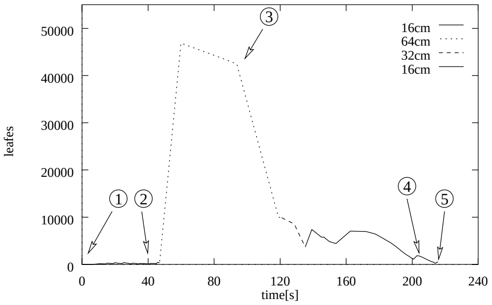

Fig. 2: Number of active states during the experiment

Figure 3 contains an outline of the environment and the trajectory of the robot and Figure 2 shows a plot of the number of active cells in the tree during this experiment. The robot started at position 1 and $P(L)$ was initialized by the corresponding Dirac distribution. At position 2 we manually rotated the robot by 140 degrees thus introducing a large rotational error. Between positions 2 and 3 the DML-method detected that it lost track of the robot’s position. Accordingly, all states were activated and the resolution of the cells was reduced to $64 \times 64cm^2$. Before reaching position 3 two ultrasound scans could be integrated so that the number of active cells started to decrease. The corresponding belief state is shown in Figure 4. It contains a logarithmic density plot of $P(L)$ when the robot was at position 3. Between positions 3 and 4 further ultrasound scans were integrated into $P(L)$ so that the certainty of the robot increased. Simultaneously, the resolution of the state space was increased to $16 \times 16cm^2$. At position 4, the state space contained only 3,600 cells corresponding to two local maxima coming from the symmetry of the corridor (see Figure 5). Finally, at position 5 and after entering room A the robot was able to resolve the ambiguous situation because of the differences between room A and B. At this point, the state space shown in Figure 6 contained only 400 cells.

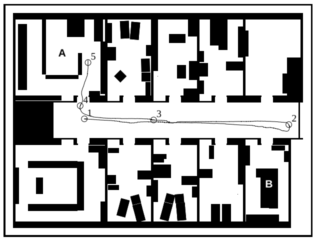

Fig. 3: Outline of the office and path of the robot

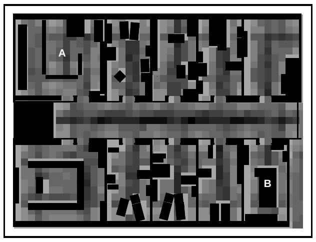

Fig. 4: Belief state position 3 shortly after starting the re-localization

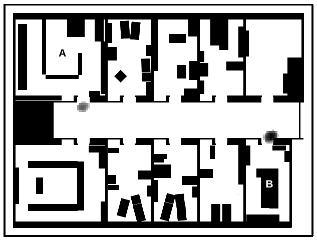

Fig. 5: Ambiguous situation at position 4

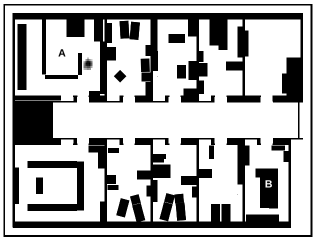

Fig. 6: The position of the robot has been determined uniquely at position 5

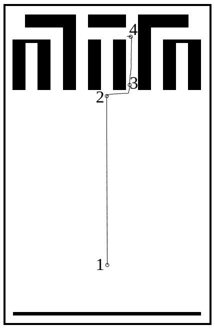

Fig. 7: Trajectory of the robot

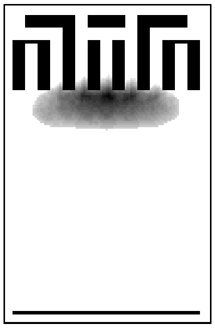

Fig. 8: Belief state position 2

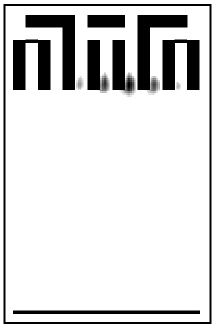

Fig. 9: Belief state at position 3

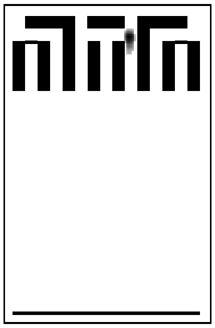

Fig. 10: Final belief state

5.2. Representation of Ambiguities

This experiment is designed to give a deeper insight in how the DML-method represents the state space in ambiguous situations. Due to the lack of an appropriate environment, this experiment has been carried out with the RHINO system simulator. The environment with a size of $48 \times 30 m^2$ and the path of the robot are illustrated in Figure 7. In this experiment we used a fixed cell size of 30cm. At the beginning of the experiment, the robot was absolutely certain about being at position 1. We assumed that the robot has large normally distributed odometric errors, so that there was a large region of possible positions when the robot arrived at position 2 after passing the hallway (see Figure 8).

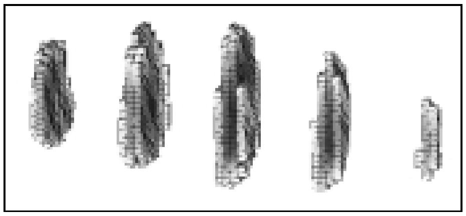

Fig. 11: Active cells in the tree structure at position 3

The active cells were grouped in a single but large cluster. After entering one of the corridors (at position 3), the set of active cells was partitioned in 5 different clusters each representing similar positions in the neighbouring corridors (see Figures 9 and 11). Finally, after reaching position 4, the robot again was absolutely certain about its position due to the different shapes of the corridors (see Figure 10).

5.3. Accuracy

The final experiment is designed to demonstrate that the estimation error can be reduced by increasing the resolution of the discretization. Figure 12 shows a path of the robot in our office environment including 22 reference positions where the true position of the robot was determined using the scan matching technique presented in [7, 9]. All data recorded during this run were split into four disjoint traces of the sensor data. These different traces were then used to localize the robot. Figure 13 shows the average localization error for ultrasound and laser-range finder based localization over all reference positions in all four runs. These results demonstrate (1) that the the average localization error for both sensors is generally below the cell size and (2) that the laser-range finders provide a significantly higher accuracy than the ultrasound sensors. When using the laser-range finders at a spatial resolution of $4cm$, the average positioning error can even be reduced to $3.5cm$.

6. Conclusions

In this paper we presented the DML-approach as a uniform approach for estimating the global position of a mobile

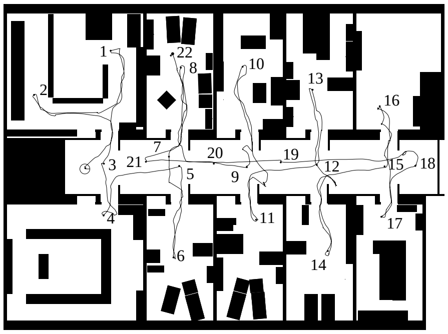

Fig. 12: Path of the robot and reference positions

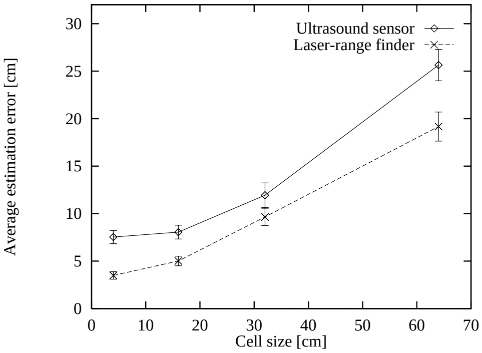

Fig. 13: Average estimation error for ultrasound sensors and laser-range finders depending on the cell size

robot as well as for fast and accurate position tracking. The DML-approach is a Markov localization technique which uses a fine-grained discretization of the state space. It applies a selective technique for updating only the likely states of the robot and dynamically adopts the resolution of the discretization according to its certainty. In the case of uncertainty it uses coarse resolutions to efficiently determine the position of the robot. On the other hand, it applies a fine-grained discretization to achieve accurate position estimates whenever the robot is quite certain about its position. An experiment carried out in a structured office environment demonstrates that average localization error can be reduced to less than $4cm$. Thus, the DML-technique combines the advantages of global localization techniques with the accuracy and efficiency of tracking methods.

Despite these exciting results, there are several warrants for future research. Currently we do not have any means for determining the optimal discretization with respect to the current belief state, the task to be performed, and the available computing resources. One way to deal with limited resources is to slow the robot down in order to be able to integrate a sufficient number of sensor readings. In this context the question has to be answered, how fast a robot may drive in order to be able to keep track of it's position.

References

[1] K. Arras and S. Vestli. Hybrid, high-precision localisation for the mail distributing mobile robot system MOPS. In *Proc. of the Int. Conf. on Robotics and Automation*, 1998.

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

[3] W. Burgard, A. Cremers, D. Fox, G. Lakemeyer, D. Hähnel, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc.of the Fifteenth National Conference on Artificial Intelligence*, 1998. To appear.

[4] W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In *Proceedings of the 21th German Conference on Artificial Intelligence (KI 97), Freiburg, Germany*. Springer Verlag, 1997.

[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*, pages 896–901, 1996.

[6] D. Fox, W. Burgard, S. Thrun, and A. Cremers. Position estimation for mobile robots in dynamic environments. In *Proc.of the Fifteenth National Conference on Artificial Intelligence*, 1998. To appear.

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

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

[9] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1994.

[10] P. S. Maybeck. The Kalman filter: An introduction to concepts. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer Verlag, 1990.

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

[12] H. Samet. *Applications of Spatial Data Structures*. Addison-Wesley Publishing Inc., 1989.

[13] B. Schiele and J. L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1628–1634, 1994.

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

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

[16] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case studies of successful robot systems*. MIT Press, Cambridge, MA, to appear.