Fast Grid-based Position Tracking for Mobile Robots

Wolfram Burgard, Dieter Fox, and Daniel Hennig

Universität Bonn, Institut für Informatik III, Römerstr. 164, D-53117 Bonn
{wolfram,fox,hennig}@uran.cs.uni-bonn.de

Abstract. One of the fundamental problems in the field of mobile robotics is the estimation of the robot's position in the environment. Position probability grids have been proven to be a robust technique for the estimation of the absolute position of a mobile robot. In this paper we describe an application of position probability grids to position tracking. Given a starting position our approach keeps track of the robot's current position by matching sensor readings against a metric model of the environment. The method is designed to work with noisy sensors and approximative models of the environment. Furthermore, it is able to integrate sensor readings of different types of sensors over time. By using raw sensor data, the method exploits arbitrary features of the environment and, in contrast to many other approaches, is not restricted to a fixed set of predefined features such as doors, openings or corridor junction types. An adaptable sensor model allows a fast integration of new sensings. The results described in this paper illustrate the robustness of our method in the presence of sensor noise and errors in the environmental model.

1 Introduction

In order to autonomously operate in their environments, mobile robots must know their position. The problem of estimating the robot's position can be divided into two sub-problems: the estimation of the *absolute* position in the environment and the tracking of the robot's position *relative* to a given starting point [1]. The task of the tracking techniques is the correction of accumulated dead reckoning errors caused by the inherent inaccuracy of the wheel encoders and other factors such as slipping. Position tracking in fact can be regarded as a special case of estimating the absolute position, because it uses a restricted search space generally centered around the robot's estimated position instead of considering each point in the environment as a possible position. We have the following requirements to such a method:

1. **It must able to deal with uncertain information.** This is important because
   - sensors are generally imperfect. This concerns wheel encoders as well as proximity sensors such as ultrasonic sensors or laser range-finders.
   - models of the environment are generally inaccurate. Possible reasons for deviations of the map from the real world come from imperfect sensors, measuring errors, simplifications, open or closed doors, or even moving objects such as humans or other mobile robots.

2. **It must allow the integration of sensor readings from different types of sensors over time.** Sensor fusion improves reliability while the integration over time compensates noise.
3. **It must be able to exploit arbitrary features of the environment which are visible by the sensors.** Many techniques for position tracking are based on landmarks such as doorways, openings or junction types in corridors. These approaches are therefore not able to exploit geometric features of the environment such as the width of corridors or even the size of rooms and objects.

Our fast grid-based position tracking technique presented in this paper meets all these requirements. The principle of the *position probability grid approach* is to accumulate in each cell of the grid the posterior probability that this cell refers to the current position of the robot. Since each possible state of the robot is defined by a tuple (x, y, θ) representing the position and orientation of the robot, position probability grids have three dimensions. Such a grid provides a discrete approximation of the probability function of the robot's current position. The approximation is adapted by integrating the likelihoods of sensor information over time. These likelihoods are computed by matching the measurements against a given environmental model.

In [3] we showed that this technique allows a mobile robot to determine its absolute position in typical office environments within a short time. In this paper we describe a specialization of the position probability grid approach to the tracking problem. Instead of considering all possible positions of the robot which leads to large state spaces even for environments with a reasonable size, we consider a small cubic state space containing only those positions centered around the currently estimated position of the robot. In addition, we apply a fast sensor model allowing frequent updates of the state space given new sensory input. In different examples we demonstrate the robustness of this tracking technique even if noisy sensors such as ultrasonic sensors are used and if approximative environmental models are given. After discussing related work, Section 3 shows how to build position probability grids for tracking the position of a mobile robot. In Section 4 we describe our fast and adaptable model for proximity sensors. Finally, Section 5 describes different experiments in typical office environments.

2 Related work

Different techniques for the tracking of the position of mobile vehicles by matching sensor readings with maps of the environment have been developed in the past (see [1] for a comprehensive overview). Recently, more and more probabilistic techniques are applied to position estimation problems. These approaches can be distinguished according to the type of maps they rely on.

Techniques based on metric or grid-based representations of the environment generally produce Gaussian distributions representing the estimation of the robot's position. Weiß et al. [18] store angle histograms constructed out of range-finder scans taken at different locations of the environment. The position and orientation of the robot is calculated by maximizing the correlation between histograms of new measurements with the stored histograms. Schiele and Crowley [14] compare different strategies to track the robot's position based on occupancy grid maps. They use two different maps: a local grid computed using the most recent sensor readings, and a global map built during a previous exploration of the environment or by an appropriate CAD-tool. The local map is matched with the global map to produce a position and orientation estimate. This estimate is combined with the previous estimate using a Kalman filter [9], where the uncertainty is represented by the width of the Gaussian distribution. Compared to the approach of Weiß et al., this technique allows the integration of different measurements over time rather than taking the optimum match of the most recent sensing as a guess for the current position.

Other techniques are designed to deal with topological maps. Nourbakhsh et al. [12] apply Markov Models to determine the node of the topological map which refers to the current position of the robot. Different nodes of the topological map are distinguished by walls, doors or hallway openings. Such objects are detected using ultrasonic sensors, and the position of the robot is determined by a "state-set progression technique", where each state represents a node in the topological map. This technique is augmented by certainty factors which are computed out of the likelihoods that the items mentioned above will be detected by the ultrasonic sensors. Hertzberg and Kirchner [6] apply a similar approach for mobile robot navigation in sewerage pipes. Simmons and Koenig [15] additionally utilize metric information coming from the wheel encoders to compute state transition probabilities. This metric information puts additional constraints on the robot's location and results in more reliable position estimates. Kortenkamp and Weymoth [8] combine information obtained from sonar sensors and cameras using a Bayesian network to detect gateways between nodes of the topological map. The integration of sonar and vision information results in a much better place recognition which reduces the number of necessary robot movements respectively transitions between different nodes of the topological map.

Due to the separation of the environment into different nodes the methods based on topological maps, in contrast to the methods based on metric maps, allow to deal with ambiguous situations. Such ambiguities are represented by different nodes having high position probabilities. However, the techniques based on topological maps provide a limited accuracy because of the low granularity of the discretization. This restricted precision is disadvantageously if the robot has to navigate fast through the environment or even perform manipulation tasks.

Position probability grids provide a metric discretization of the environment and thus provide metric estimates of the robot's position. A further advantage is the ability to exploit every sensing instead of only those taken at certain reference points or such sensor readings identifying pre-defined landmarks, such as doors, openings or junction types. By using raw sensory information arbitrary features that are seen by the sensors are exploited. Furthermore, the fast sensor model allows the direct integration of new readings without constructing local maps of the environment. Finally, the increased number of parameters in the discrete state space results in more accurate representations of the position probability density than obtained with Kalman filters.

3 Position tracking with position probability grids

The position probability grid approach [3] initially has been designed to estimate the *global* position of a mobile robot in its environment. The basic idea of this approach is to provide a discrete approximation of the position probability density function for the given environment. A position probability grid is a three-dimensional array containing in each field the probability that this field refers to the current position and orientation of the robot. For a grid field x this value is obtained by repeatedly firing the robot's sensors and accumulating in x the likelihoods of the sensed values supposed the center of x currently is the position of the robot in the environment.

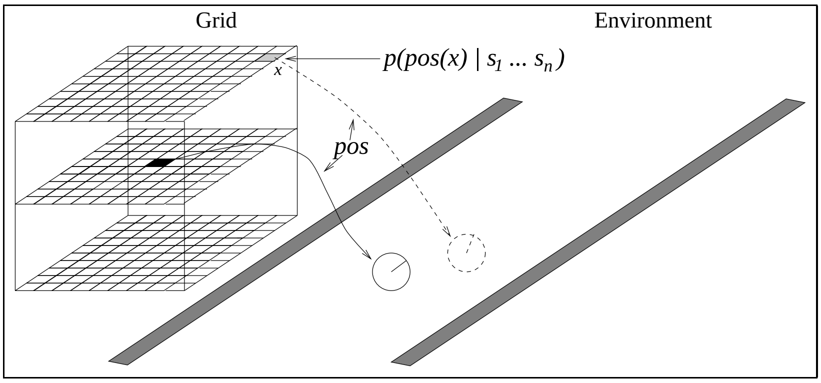

Figure 1. Transformation of grid coordinates into real world coordinates

Whenever the global position of the robot is uniquely determined, the huge state space of the estimation problem can be reduced to a small cube P centered around the robot's estimated position. In this case, tracking the robot's position is equivalent to estimating a function pos transforming the coordinates of the cells x in P to the corresponding coordinates $\text{pos}(x)$ in the environment. This mapping is updated whenever new input is obtained from the wheel encoders or the other sensors of the robot:

1. The coordinate transformation pos is updated according to the movements measured by the wheel encoders of the robot since the last update.
2. After each integration of a sensor input, the cell x containing the maximum probability within P is regarded as referring to the current position of the robot. If necessary, the cells in P are shifted such that the cell with the maximum probability becomes the center of P. The transformation pos is adopted accordingly.

3.1 Integrating sensor readings of proximity sensors

To update P given new sensory input s_n we apply the well-known Bayesian update formula. We compute the likelihood $p(s \mid \mathrm{pos}(x) \wedge m)$ that s is obtained given the robot is in position $\mathrm{pos}(x)$ in the environment, where m is a model of this environment. This likelihood is combined with the probability in x, thus obtaining a new probability that the robot at position $\mathrm{pos}(x)$.

Suppose $p(x \mid s_1 \wedge \ldots \wedge s_{n-1} \wedge m)$ is the (posterior) probability that $\mathrm{pos}(x)$ is the current position of the robot, given m and the sensor readings $s_1, \ldots, s_{n-1}$. Then the probability of x referring to the current position of the robot given new sensory input s_n is defined as

$$p(x \mid s_1 \wedge \ldots \wedge s_n \wedge m) = \frac{p(x \mid s_1 \wedge \ldots \wedge s_{n-1} \wedge m) \cdot p(s_n \mid \mathrm{pos}(x) \wedge m)}{\alpha} \quad (1)$$

where $\alpha = \sum_{x \in P} p(x \mid s_1 \wedge \ldots \wedge s_{n-1} \wedge m) \cdot p(s_n \mid \mathrm{pos}(x) \wedge m)$ is a normalizer ensuring that the position probabilities over all x sum up to 1 [13].

For an implementation of this approach Equation (1) defines the update of P given new sensory input s_n. If $P[x]$ is the value of field x in P, then all we have to do is to multiply $P[x]$ by $p(s_n \mid \mathrm{pos}(x) \wedge m)$ and to store the result in x. After that, we have to normalize P. To initialize $P[x]$ we use the a priori probability $p(\mathrm{pos}(x) \mid m)$ of x referring to the position of the robot given m.

Obviously, the term $p(s_n \mid \mathrm{pos}(x) \wedge m)$ is the crucial component of the update equation. It specifies the likelihood of observing s_n at location $\mathrm{pos}(x)$, for any choice of s_n and x. In [3,10] $p(s_n \mid \mathrm{pos}(x) \wedge m)$ is computed at runtime from a metric model of the environment and a model of sonar sensors. [6,7,12,15] use topological representations of the location space and first scan the sensor data for the presence or absence of certain landmarks. In these approaches the probabilities $p(s_n \mid \mathrm{pos}(x) \wedge m)$ are stored in a lookup table. In Section 4 we show how the sensing probabilities can be computed off-line and compactly stored in a lookup table even for our geometric grid-based approach.

3.2 Integrating the movements of the robot

To integrate the dead-reckoning information of the wheel encoders, we update the coordinate transformation pos according to the measured movement Δ

which is equivalent to shifting P by Δ. In order to deal with possible dead reckoning errors we use a general formula coming from the domain of Markov chains. We regard each cell in P as one possible state of the robot, and determine a state transition probability $p(x \mid \tilde{x} \wedge \Delta)$ for each pair x, $\tilde{x}$ of cells in P, which depends on the trajectory taken by the robot and the time elapsed since the previous update. Then we apply the following update formula:

$$P[x] := \sum_{\tilde{x} \in P} P[\tilde{x}] \cdot p(x \mid \tilde{x} \wedge \Delta) \tag{2}$$

Additionally, we consider how the trajectory taken by the robot fits into the environment. A trajectory leading through free space leads to a higher position probability than a trajectory leading through an obstacle. Therefore, we multiply each field x in P with the a priory position probability $p(\text{pos}(x) \mid m)$. Assuming that the robot does not leave the environmental model m, the update formula is

$$P[x] := \begin{cases} \beta \cdot P[x] \cdot p(\text{pos}(x) \mid m) & \text{if } \text{pos}(x) \in m \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where β again is a normalizing constant. To estimate the a priori probability $p(\text{pos}(x) \mid m)$ that the robot is at position $\text{pos}(x)$, we use an occupancy probability map o of the environment [11,10], which in our case is computed from m. We assume that $p(\text{pos}(x) \mid m)$ directly depends on the occupancy probability $o(\text{pos}(x))$ of the field $\text{pos}(x)$ in o:

$$p(\text{pos}(x) \mid m) = \frac{1 - o(\text{pos}(x))}{\sum_{\tilde{x} \in o} (1 - o(\tilde{x}))} \tag{4}$$

4 A fast sensor model for proximity sensors

To compute the likelihood $p(s \mid x \wedge m)$ that a sensor reading s is received given a position x we analyze the model m of the environment. Since this likelihood has to be estimated for each possible position x of the robot, the necessary computation time has a high impact on the efficiency of the overall approach.

In [3] we applied an approach similar to Moravec's [10] for estimating the probability that a sensor measures a certain distance. One disadvantage of this approach, however, lies in its computational complexity. For each location x one has to compute a generally not Gaussian probability density function over a discrete set of possible distances measured by the sensor. For an environment of $15 \times 15m^2$, a discretization of $15 \times 15cm^2$, and an angular resolution of 1 degree, the state space of the robot contains already 3.600.000 states. Consequently, representing the complete densities for all possible states would by far exceed the memory of typical computers and especially such mobile robots generally are equipped with.

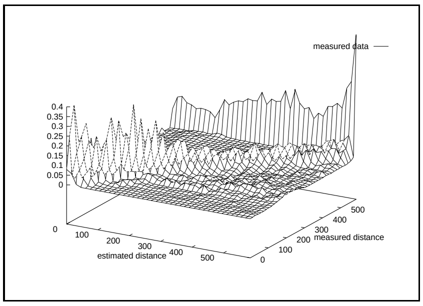

Figure 2. Distances measured by the robot

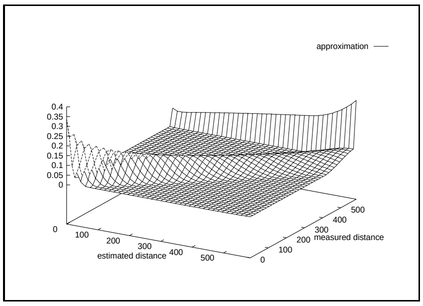

Figure 3. Approximation of the measured data

The key idea of our approach is to use a simplified sensor model which allows us to compute the likelihood of a proximity sensing solely given the distance to the next obstacle in the sensing direction. This sensor model is based on the following observations. First, if an obstacle is detected by the sensor then the measured distance is generally normally distributed around the exact distance to the object. Second, because of the limited accuracy of the sensors and the world models, there is a small chance for any distance to be perceived by the sensor. Finally, there is an additional probability that the beam is absorbed or completely reflected, which results in a maximum range measurement. Figure 2 shows the probability density of measuring a distance d_1 given an expected distance d_2 and justifies these assumptions. This function was obtained from approximately 190.000 sonar sensings in different environments. Figure 3 shows the density obtained by adapting our model to these data.

Given this model, it suffices to store for each possible state of the robot the distance which is expected to be measured by the sensor. These expected distances can be computed off-line given metric representations of the environment. Because of a compact representation of the expected distances in a table only two lookup operations are necessary to compute the likelihood of a given sensing at runtime. Although this sensor model, in contrast to the original model introduced by Moravec and Elfes [10], has only a small number of parameters, we obtained a significant speed up which results in a better positioning performance for global localization as well as position tracking. In [4] we successfully utilized this model even for active self-localization of mobile robots.

5 Experimental results

The grid-based tracking technique described above has been implemented and extensively tested. The current system is able to interpret sensor readings of ultrasonic sensors and match them with occupancy probability grid maps. The experiments described in this section were carried out with our mobile

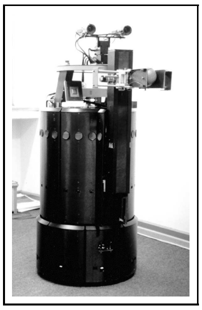

Figure 4. The mobile robot *RHINO*

Figure 5. *RHINO's* sister *Amelia*

robot *RHINO* [2,17] (see Figure 4). as well as *Amelia* (see Figure 5) which is one of the mobile robots in the mobile robot laboratory of the computer science department of the Carnegie Mellon University. Both robots are B21 robots manufactured by Real World Interface Inc. They are equipped with a ring of 24 Ultrasonic sensors, each having a cone of 15 deg. Throughout the experiments we used a position probability grid of $15 \times 15 \times 15$ fields, each $15\text{cm} \times 15\text{cm} \times 1\text{deg}$ in size. The integration of a sweep of all 24 sensors takes about 0.2 seconds on a Pentium 120 computer.

5.1 Position tracking in large environments

As a complex example we used a typical run of the mobile robot *Amelia* in the Wean Hall of the Carnegie Mellon University. Figure 6 shows the map of the environment as well as *Amelia's* trajectory as measured by the wheel-encoders. The size of the environment is a about $75 \times 35m^2$, and the length of the trajectory is over $200m$. The starting point is in the northern corridor facing east. As can be seen in the figure the error of the orientation permanently increases up to a value of more than 30 degree. Obviously the model of the environment would be useless after short time if the position tracking would rely on the wheel encoders only.

Figure 7 shows the corrected trajectory of the robot computed with our position probability grid approach based on the information coming from the wheel encoders and the 24 ultrasonic sensors. The likelihoods of the sonar measurements were computed by matching them with the displayed map. At this point it should be noted, that this map is very approximative as it only represents an outline of the environment. For example, the hallway in the south-west of the Wean Hall contains a cafeteria with two bars as well as tables and chairs. Furthermore, the corridors contain several objects not included in the map such as thrash bins, information boards etc. Finally, the state of doors was not represented correctly by the map, and there were many people which walked through the environment and additionally produced unexpected readings. Nevertheless, our approach is able to reliably keep track of the robot's position.

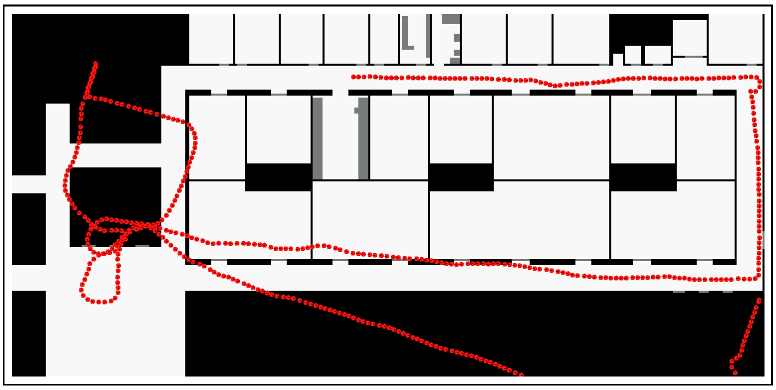

Figure 6. Trajectory measured by the wheel-encoders

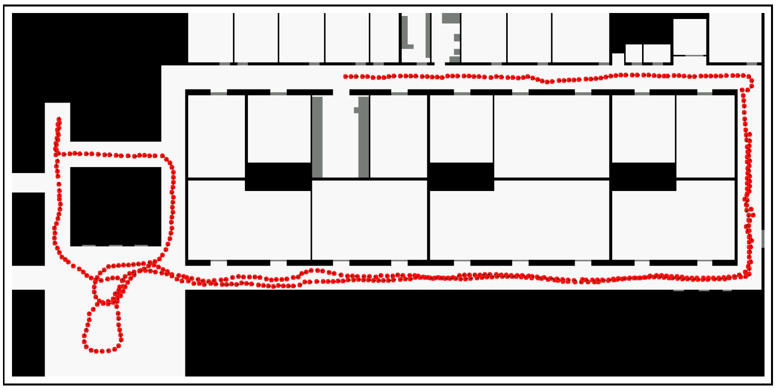

Figure 7. Corrected path

Figure 8 illustrates an example position probability grid. The grid shows the position probability distribution of the robot in a corridor. The different layers represent the position probabilities for 9 orientations. The layers are marked with the deviation of the orientation from the current orientation. Each layer represents a $225 \times 225 cm^2$ area centered around the currently estimated position of the robot. For simplicity only 9 of the 15 orientations are plotted. Notice that the maximal value lies in the center of the cube.

5.2 Accuracy of the approach

To demonstrate the accuracy of the approach we performed 8 different runs with our mobile robot *RHINO* in a part of our department which has a s size of $20 \times 25m^2$. In all theses runs shown in Figure 9 the robot's starting position was in the southern office. We steered *RHINO* on different trajectories through the corridor and measured the final position in the northern office. In this experiment we used only 12 ultrasonic sensors. As in the previous

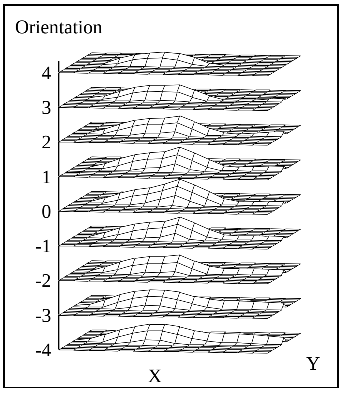

Figure 8. Typical density in a corridor

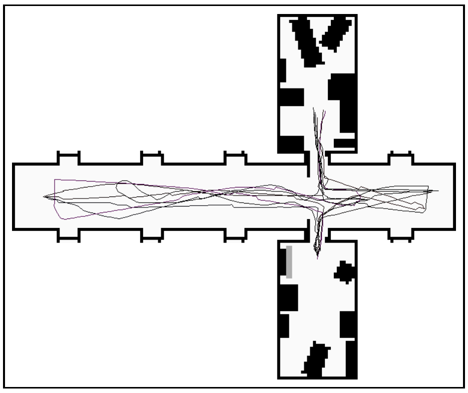

Figure 9. Trajectories of *RHINO* in eight test runs

experiment the side length of one grid cell was $15cm$. The average distance between the estimated and the measured position of the robot was less than the resolution of the grid. This shows that the position of the robot can be tracked accurately even if noisy sensors are used.

6 Discussion

In this paper we presented a fast grid-based technique for tracking a mobile robot's position in a known environment. The advantage of this technique is the ability to deal with noisy sensors (e.g. ultrasonic sensors) and approximative environmental models, and to integrate sensor readings from different types of sensors over time. Based on a fast and adaptable sensor model the approach quickly integrates new sensory input. Because it directly uses the proximity information coming from the sensors, it is able to exploit arbitrary features of the environment which are visible by the sensors.

Our technique has been implemented and tested in several complex real-world environments. The experiments presented here demonstrate the robustness of this method in tracking the position of a mobile robot. As the experiment in the over $2000m^2$ wide office environment shows, trajectories longer than $200m$ are tracked successfully even if only approximative models are given. Due to the fast sensor model, the evaluation of a complete sweep consisting of 24 ultrasonic sensors measurements into a grid consisting of more than 3000 cells takes about 0.2 seconds on a Pentium 120 computer, which turned out to be sufficient for a reliable on-line tracking of the robot's position.

An interesting question concerns the relation of the position probability grid technique to Kalman filters [9,16]. The difference between both approaches lies in the approximation of the position probability density function. Whereas our method provides a discrete approximation of this function, Kalman filters generally approximate the overall distribution by a Gaussian density function. There are different situations which cannot be represented by single Gaussian distributions. Consider a situation where the robot is close to a wall or another kind of obstacle. In Figure 10, the robot is in corridor and in front of a pillar. This situation leads to a non-Gaussian distribution

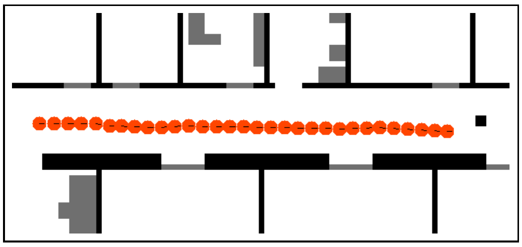

Figure 10. Trajectory of the robot

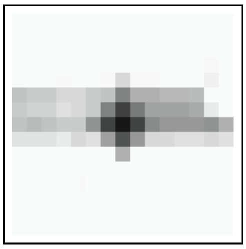

Figure 11. Resulting density

of the position probability. As illustrated in Figure 11 (more likely positions are darker) our technique adequately represents that the robot cannot be at the same position as the pillar. A similar situation is given when the robot is close to a wall. The resulting density is non-Gaussian, because the robot is more likely in the corridor than in the wall. Thus, a Kalman filter tends to shift the position estimate to the center of the corridor, which is obviously wrong. On the contrary, our approach is able to deal with such situations accordingly (see Figure 11).

Despite these encouraging results, there are further warrants for future research. The most important question concerns the integration of global position estimation and position tracking. In our experiments we sometimes observed a failure of our tracking technique, if the robot was in a large hallway for a long time. In such situations the accumulated dead reckoning errors cannot be corrected due to the lack of features detectable by the proximity sensors. To deal with such cases, we are currently working on a combination of this fast tracking approach with our global position estimation technique described in [3].

References

1. J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques.* A. K. Peters, Ltd., Wellesley, MA, 1996.
2. Joachim Buhmann, Wolfram Burgard, Armin B. Cremers, Dieter Fox, Thomas Hofmann, Frank Schneider, Jiannis Strikos, and Sebastian Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, Summer 1995.
3. Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo 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)*, pages 896–901, 1996.

4. Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *Proc. of the Fifteenth International Conference on Artificial Intelligence (IJCAI-97)*, 1997.
5. I.J. Cox and G.T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer Verlag, 1990.
6. Joachim Hertzberg and Frank Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROMICRO '96)*, pages 68–73. IEEE Computer Society Press, 1996.
7. Leslie Pack Kaelbling, Anthony R. Cassandra, and James 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.
8. David Kortenkamp and Terry Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proc. of the Twelfth National Conference on Artificial Intelligence*, pages 979–984, 1994.
9. Peter S. Maybeck. The Kalman filter: An introduction to concepts. In Cox and Wilfong [5].
10. Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.
11. Hans P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 116–121, 1985.
12. Illa Nourbakhsh, Rob Powers, and Stan Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, Summer 1995.
13. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.
14. Bernt Schiele and James 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.
15. Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. International Joint Conference on Artificial Intelligence*, 1995.
16. Harold W. Sorensen. *Kalman Filtering: Theroy and Application*. IEEE Press, 1985.
17. 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.P. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case studies of successful robot systems*. MIT Press, Cambridge, MA, to appear.
18. Gerhard Weiß, Christopher Wetzler, and Ewald von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 595–601, 1994.