Position Tracking with Position Probability Grids

Wolfram Burgard Dieter Fox Daniel Hennig Timo Schmidt
Institut für Informatik III, Universtiät Bonn
Römerstr. 164, D-53117 Bonn
{wolfram,fox,hennig,schmidt1}@uran.cs.uni-bonn.de

Abstract

One of the main 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 the tracking of the position of the robot. The main difference of our method to previous approaches lies in the fact that the position probability grid technique is a Bayesian approach which is able to deal with noisy sensors as well as ambiguities and is able to integrate sensor readings of different types of sensors over time. Given a starting position this method estimates the robot's current position by matching sensor readings against a metric model of the environment. Results described in this paper illustrate the robustness of this method against noisy sensors and errors in the environmental model.

1. Introduction

In order to autonomously operate in their environments, mobile robots must always know their position and orientation[1]. 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 [4]. The task of the tracking techniques is the correction of accumulated dead reckoning errors coming from 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 predicted position instead of considering each point in environment as a possible position. In [2] we introduced the *position probability grid* as a robust technique for the estimation of the absolute position of a mobile robot in its environment. We furthermore showed that this method meets the following requirements:

1. **It is 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 is able to deal with ambiguities.** Typical office environments contain several places which cannot be distinguished with a single measurement. As example consider a long corridor, where changes of the position due to the limited range of the sensors do not necessarily result in changes of the measured values. Thus, the set of possible positions of the robot is a region in that corridor.
3. **It allows the integration of sensor readings from different types of sensors over time.** Sensor fusion improves reliability while the integration over time compensates noise and is necessary to resolve ambiguities.

The principle of the position probability grid approach is to accumulate in each cell of a three-dimensional grid the posterior probability that this cell refers to the current position of the robot. Thus the 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 [2] we showed that position probability grids are a robust technique allowing a mobile robot to determine its absolute position in typical office environments within a short time.

[1] In the remainder of this paper we use the notion "position" to refer to "position and orientation" if not stated otherwise.

In this paper we describe a specialization of the position probability grid approach to the tracking problem. Once the absolute position of the robot is determined we extract a small cube out of the complete grid which is centered around the grid cell with the maximum likelihood. Different examples show the robustness of this tracking technique even if noisy sensors are used and if approximative environmental models are given.

After discussing related work Section 3 shows how to build position probability grids for the estimation of the robot's position. In Section 4 we describe an application of this technique to ultrasonic sensors and occupancy probability maps as world model. Finally, Section 5 describes experiments with this application in typical office environments.

2. Related work

Different techniques for the tracking of the position of mobile vehicles by map-matching have been developed in the past (see [3, 4] for overviews). 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. [16] 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 [11] 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 against the global map to produce a position and orientation estimate. This estimate is combined with the previous estimate using a Kalman filter [6], where the uncertainty is represented by the width of the Gaussian distribution. Compared to the approach of Weiß et al., our technique allows an 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. [9] 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. Simmons and Koenig [13] describe a similar approach to position estimation. They 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 [5] 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 described above, 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 its environment or even grasp for objects.

Position probability grids provide a metric discretization of the environment, and are able to accurately estimate 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 certain a priori known objects (e.g. doors, openings etc.). By integrating the likelihoods of measurements over time, it provides a discrete approximation of the position probability density function.

3. Position tracking with position probability grids

The position probability grid approach [2] initially has been designed to estimate the absolute position of a mobile robot in a known environment. The basic idea of this approach is to provide a discrete approximation of the position probability density function for the given environment. From this point of view, position probability grids can be regarded as the pendant to occupancy maps introduced by Elfes and Moravec as a probabilistic grid model for the representation of obstacles [8, 7]. A *position probability grid* is a three-dimensional array containing in each field the posterior probability that this field refers to the current position and orientation of the robot. For a grid field x this certainty 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 environmental model m. We regard the maximum likelihood in this grid as the best estimate for the position of the robot.

To apply this technique to the position tracking problem we simply reduce the size of the grid to a small cube P centered around the robot's estimated position. The position of the center of this local grid is changed according to the position information provided by the wheel encoders of the robot. Thus, each time the robot's sensors are fired, the following steps are carried out:

1. The position of P is shifted according to the movement measured by the wheel encoders of the robot since the last update.
2. For each grid field x of P and each reading s, the likelihood of s is computed under the assumption that x refers to the current position of the robot in m. This likelihood is combined with the probability stored in x to obtain a new probability for x.
3. The cell containing the maximum probability is regarded as referring to the current position of the robot. Thus, if x is the cell with the maximum probability in P, then P is shifted such that x becomes the center of the grid.

The basic assumptions for the tracking with position probability grids are:

- *The robot must possess a model m of the world the sensor readings can be matched against.* Such models can either come from CAD-drawings of the environment or can themselves be grid representations of occupancy probabilities.
- *The robot does not leave the environmental model.* This assumption allows us to deal with situations where the robot comes close to the border of the environmental model. In this case we simply set the probabilities of grid fields of P outside of m to zero.

3.1. Updating position probability grids

The hypothesis space of the position estimation problem consists of $|P|$ different hypotheses, where $|P|$ is the size of the grid P. Each field x of P contains the probability that the current position of the robot is referred by x, and the sum of the position probabilities over all fields of P must be 1.

Let pos(x) be the position referred to by x. Suppose m is the model of the environment, and $p(x \mid s_1 \wedge \ldots \wedge s_{n-1} \wedge m)$ is the (posterior) probability that x refers to 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 [2]:

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

The constant α simply normalizes the sum of the position probabilities over all x up to 1 [10].

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 $p(s_n \mid \text{pos}(x) \wedge m)$ and to store the result in x. After that, we have to normalize P. To initialize $P[x]$ we use the a priori probability $p(\text{pos}(x) \mid m)$ of x referring to the actual position of the robot given m.

Note that the estimation of the probability $p(\text{pos}(x) \mid m)$ as well as the likelihood $p(s_n \mid \text{pos}(x) \wedge m)$ depends on the given world model and the type of the sensors used for position estimation. In Section 4 we demonstrate how to use occupancy probability maps for position estimation and how sensor readings of ultrasonic sensors are matched against such maps.

3.2. Integrating the movements of the robot

To integrate the dead-reckoning information of the wheel encoders, we

1. update the position of the grid and the grid cells according to the robot's movement measured by the wheel encoders.
2. consider for each cell whether the corresponding position is occupied by an obstacle. For example, a free cell has a higher position probability than an occupied cell.

Let Δ represent the movement measured by the wheel encoders since the previous update. Then the first task is done by adding Δ to the current position of P. 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:

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

Additionally, we consider how the trajectory taken by the robot fits into the environment. This is done by multiplying each field x in P with the a priory position probability $p(\text{pos}(x) \mid m)$. Because the robot does not leave the environmental model m, the update formula is

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

where β is a normalizing constant.

Finally, we have to solve the problem occurring whenever the maximum shifts from the center to another field of the local grid. Because P always has to be centered around the maximum, we shift the grid accordingly. In this case, the values of at least one complete plane of P are lost, and new values for the layer at the opposite side of the grid have to be chosen. In the current implementation this is done by initializing these fields with the a priori position probability of the corresponding positions in the environment.

4. Using sonar sensors and occupancy grid maps

4.1. Matching sonar sensor readings against occupancy grids

In the previous section we described two different procedures to update P. The first one accumulates the likelihood of sensor readings and the second one processes P according to the measurements of the wheel encoders. In this section we apply the position probability grid approach to sonar sensors and occupancy grid maps as world model.

Occupancy probability grids provide a discrete approximation of the occupancy probability function for the given environment [7]. To compute the likelihood $p(s \mid x \wedge m)$ that a sensor reading s is received given an occupancy grid map m and a position x in m we use an approach similar to Moravec's [7]: We consider a discretization $R_1, \ldots, R_k$ of possible distances measured by the sensor. Consequently, $p(R_i \mid x \wedge m)$ is the likelihood that a the distance R_i is measured at the position x in m.

There are two situations in which a sensor beam is reflected by a cell $\tilde{x}$ of m covered by the segment R_i of a sonar cone if the robot is in position x. The usual case is that an occupied cell is hit by the sonar beam. On the other hand, there is a small chance that a sensor beam erroneously is reflected. Possible reasons for such too short readings are cross-talk or inaccuracy of the world model. If $m(\tilde{x})$ is the occupancy probability of field $\tilde{x}$ in m, then we approximate the likelihood that $\tilde{x}$ reflects the sonar beam by $m(\tilde{x}) \cdot p(h(\tilde{x}) \mid x \wedge m)$ where $p(h(\tilde{x}) \mid x \wedge m)$ is the probability that the sensor would detect the occupied cell in $\tilde{x}$ given x is the position of the robot. The likelihood of a too short reading is computed by the product $(1 - m(\tilde{x})) \cdot p(e(\tilde{x}) \mid x \wedge m)$ where $p(e(\tilde{x}) \mid x \wedge m)$ is the probability that the unoccupied cell $\tilde{x}$ reflects the sonar beam. Thus, the likelihood that $\tilde{x}$ does not reflect the sonar beam is $(1 - m(\tilde{x}) \cdot p(h(\tilde{x}) \mid x \wedge m)) \cdot (1 - (1 - m(\tilde{x})) \cdot p(e(\tilde{x}) \mid x \wedge m)))$. Assuming the event that $\tilde{x}$ reflects the sonar beam being conditionally independent of the reflection of all other cells in R_i, we compute the likelihood that R_i reflects a sonar beam by

$$\begin{aligned} p(R_i \mid x \wedge m) = \\ 1 - \prod_{\tilde{x} \in R_i} \Big[\big[1 - m(\tilde{x}) \cdot p(h(\tilde{x}) \mid x \wedge m)\big] \\ \cdot \big[1 - (1 - m(\tilde{x})) \cdot p(e(\tilde{x}) \mid x \wedge m)\big] \Big] \end{aligned} \quad (4)$$

Before the beam reaches R_i, it traverses $R_1, \ldots, R_{i-1}$. Supposed that the sonar reading s is included by range R_i, $p(s \mid x \wedge m)$ equals the likelihood that R_i reflects the sonar beam given that none of the ranges $R_{<i}$ reflects it. Thus, we have

$$\begin{aligned} p(s \mid x \wedge m) &= p(R_i \mid x \wedge m) \\ &\quad \cdot \prod_{j=1}^{i-1} (1 - p(R_j \mid x \wedge m)) \end{aligned} \quad (5)$$

4.2. Computing position estimates using occupancy grids

It remains to estimate the probability $p(x \mid m)$ that the robot is at position x in m. We assume that this probability directly depends on the occupancy probability $m(x)$ of the field x in m: the higher the occupancy probability, the lower the position probability and vice versa. Therefore, the value $p(x \mid m)$ is computed as follows:

$$p(x \mid m) = \frac{1 - m(x)}{\sum_{\tilde{x} \in m} (1 - m(\tilde{x}))} \quad (6)$$

5. Experimental results

The position probability grid technique has been implemented and extensively tested for the estimation of the absolute position [2] and the tracking problem. The current system is able to interpret sensor readings of ultrasonic sensors and match them against occupancy probability grid maps. The experiments described in this section were carried out with our mobile robot *RHINO*, a B21 robot manufactured by Real World Interface Inc. (see Figure 1). *RHINO* is 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 new sensor sweep for all 24 sensors takes about 1 second on a Pentium 90 computer. This turned out to be fast enough for real-time tracking of the robot's position.

5.1. Position tracking in large environments

As a complex example we used a typical run of *RHINO* in the arena of the 1994 Mobile robot competition [12]. Fig-

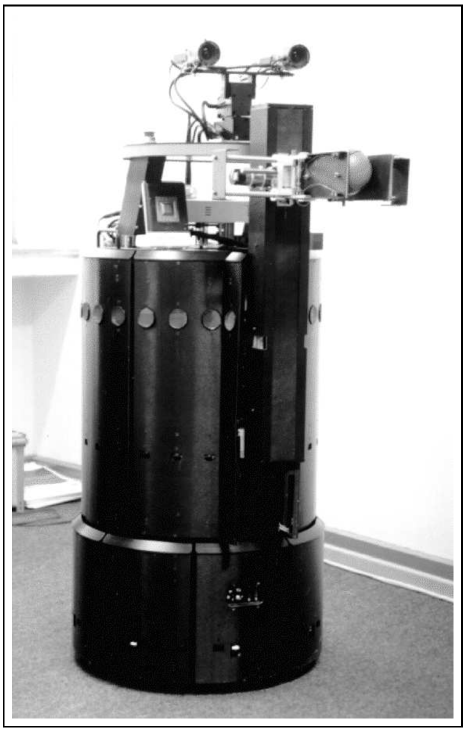

Figure 1. The mobile robot RHINO

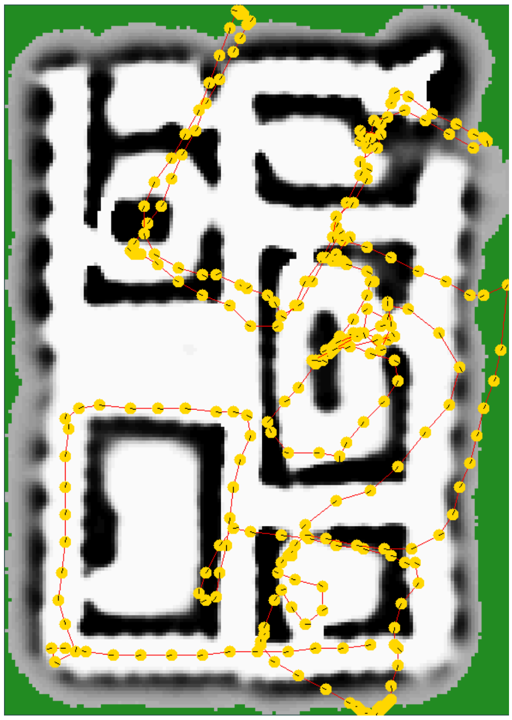

Fig. 2. Trajectory measured by the wheel-encoders

ures 2 and 3 show the occupancy grid map of the arena constructed with the map-building tool of the *RHINO* system [1, 15]. The size of the environment amounts $20 \times 30m^2$. The sonar sensor measurements as well as the position information of the wheel encoders were recorded during a run through this arena. Figure 2 includes the path of the robot as measured by the wheel-encoders. The starting point is in the lower right corner of the environment. As can be seen in the figure the error of the orientation permanently increases up to a value of more than 30 deg. Obviously the model of the environment would be useless for the robot without successful position tracking. Figure 3 shows the corrected positions of the robot computed with our position probability grid approach. The likelihoods of the sonar measurements were computed by matching them against the displayed map. Notice that this world model is not very accurate (e.g. the outer walls and corners of the arena are modeled only vaguely). The path shows that the robot successfully keeps track of its position despite of this deficiency.

Figure 4 illustrates an example position probability grid. The grid shows the position probability distribution of the robot in the lower left corridor of Figure 3. The different layers represent the position probabilities for different orientations. The layers are marked with the deviation of the orientation from the current orientation. Each layer represents a 225×225 cm^2 area centered around the current position of the robot. For simplicity only five of the 15 orientations are plotted. Notice that the maximal value lies in the center of the cube.

5.2. Combining position tracking with absolute position estimation

The next example is designed to demonstrate the combination of absolute position estimation with position tracking in a typical office of our department. In this experiment we steered the robot through the office and stopped it in the middle of the doorway leading to the corridor. The task of the robot was to estimate its absolute position in the office. As soon as the absolute position was determined the robot had to keep track of its current position.

Figure 5 shows an outline of this office. The occupancy grid map used to compute the likelihoods of the sensor readings is contained in Figure 6. The dotted path in Figure 5 represents the trajectory of the robot given by the dead-reckoning information of the wheel encoders. To determine its absolute position *RHINO* used 12 sweeps of 8 sonar readings.

Figures 7–10 show plots of the maximum probabilities for all positions in the map of the office for the first, second, fourth, and twelfth reading sets. The positions of the ro-

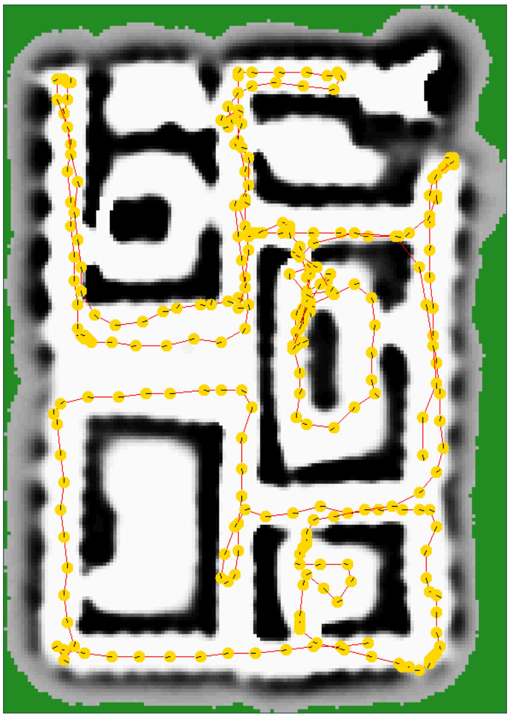

Fig. 3. Corrected path

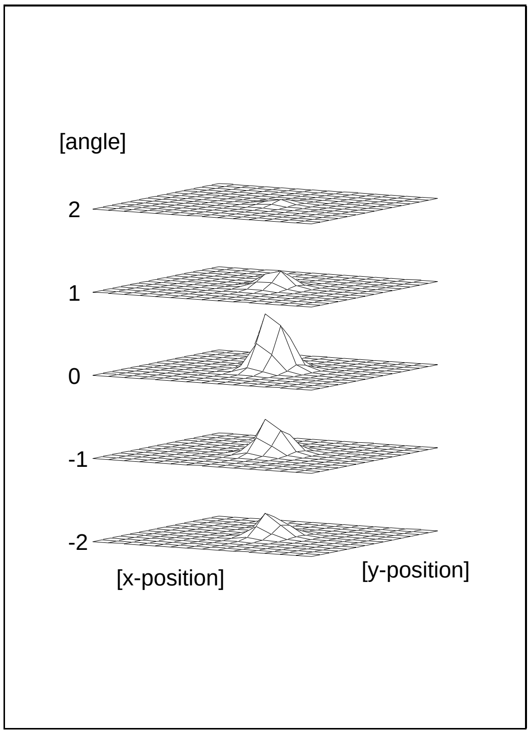

Figure 4. Local position probability grid

bot where the readings were taken are represented by small dots in Figure 5. The vertical lines in Figures 7–10 illustrate the corresponding position of the robot as given by the dead reckoning. For the sake of simplicity only the maximum probability over all orientations at each position is shown. After the first reading we have a multi-modal distribution with only a small peak for the current position of the robot. Figures 8, 9, and 10 show how the ambiguities are resolved, and the result is a significant and unique peak. Note the slight deviation of the estimated and the measured position already after two meters. The time needed to interpret these 12 readings is about 60 seconds on a Pentium 90 computer.

At this point the tracking mode is started. The tracking is initialized with the values centered around the peak shown in Figure 10. The solid line in Figure 5 represents the trajectory of the robot estimated with the position probability grid. The indentations in the corrected trajectory indicate the points where the grid is shifted in order to center it around the maximum. This corresponds to a correction of the estimated position of the robot. At the end of the corrected trajectory, the robot is close to the middle of the doorway. The difference is caused by the limited resolution of the grid. As illustrated, the deviation between the corrected and the measured position amounts to about 30 cm.

6. Discussion

In this paper we presented an application of position probability grids to the problem of tracking a mobile robot's position in a known environment. The position probability grid approach originally was designed as a robust technique for the fast and reliable estimation of the absolute position of mobile robots in known environments. The advantage of position probability grids is the ability to deal with ambiguities, noisy sensors (e.g. ultrasonic sensors) and approximative environmental models (e.g. occupancy grid maps), and to integrate sensor readings from different types of sensors over time. In this paper we developed a specialized variant of this method which allows a fast tracking of the robot's position.

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 with the $600m^2$ wide AAAI '94 robot competition arena shows, even trajectories longer than 150 m are tracked successfully. We furthermore demonstrated how absolute position estimation and position tracking can be combined. This is done by reducing the grid to a small local grid centered around the estimated position of the robot. The evaluation of a complete sweep consisting of 24 ultrasonic sensors meas-

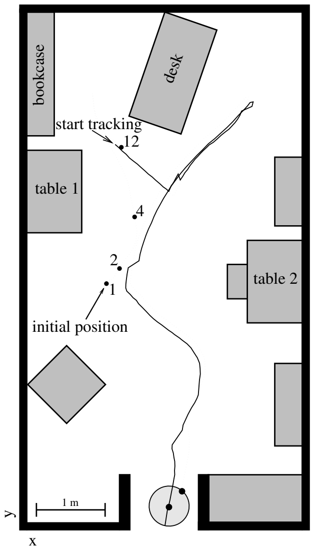

Fig. 5. Outline of the office

Fig. 6. Occupancy grid used for sensor matching

urements into a grid consisting of more than 3000 cells takes about 1 second on a Pentium 90 computer, which turned out to be sufficient for a reliable tracking of the robot's position.

An interesting question concerns the relation of the position probability grid technique to Kalman filters [6, 14] as applied in [11]. 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 approximate the overall distribution by a Gaussian density function. There are at least two situations which cannot be represented by single Gaussian distributions. First, we often observe multimodal distributions coming from ambiguities in the environment. Second, consider a situation where the robot is close to a small obstacle. Using a single Gaussian density function, one is not able to represent the fact that the robot cannot be in the position occupied by the obstacle.

The only precondition for the applicability of the position probability grid approach is an environmental model which allows to determine the likelihood of a sensor reading at a certain position in the environment. In our implementation we used occupancy probability grids in combination with ultrasonic sensors. Alternatively one could use a CAD-model of the environment and cameras for edge detection or integrate simple features like the color of the floor.

References

[1] 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.

[2] 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 Thirteenth National Conference on Artificial Intelligence*, 1996.

[3] I.J. Cox and G.T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[4] L. Feng, J. Borenstein, and H.R. Everett. "Where am I?" Sensors and Methods for Autonomous Mobile Robot Positioning. Technical Report UM-MEAM-94-21, University of Michigan, December 1994.

[5] 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.

[6] Peter S. Maybeck. The Kalman filter: An introduction to concepts. In Cox and Wilfong [3].

[7] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.

[8] 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.

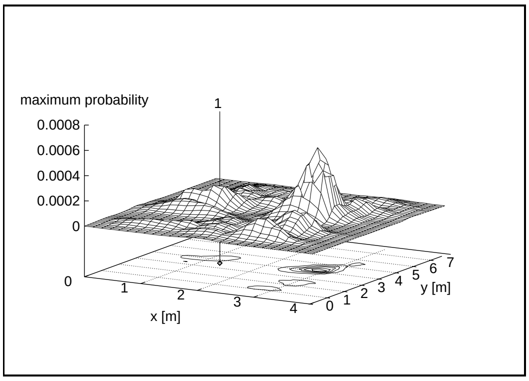

Figure 7. Probability distribution after 1 step

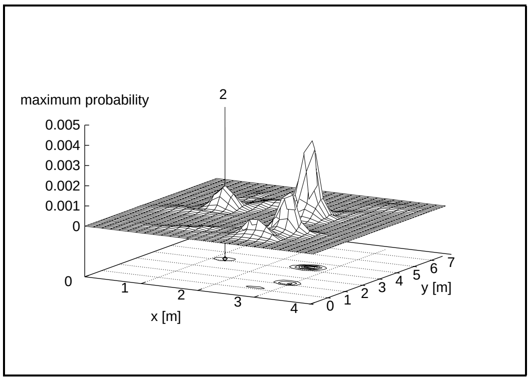

Figure 8. Probability distribution after 2 steps

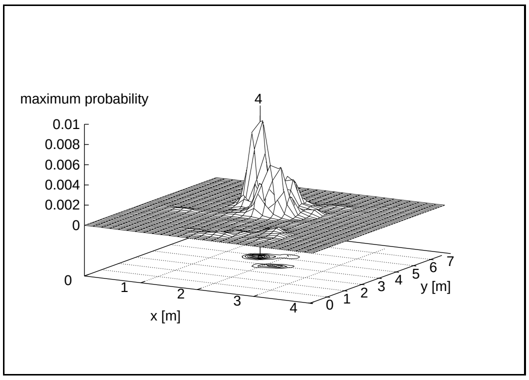

Figure 9. Probability distribution after 4 steps

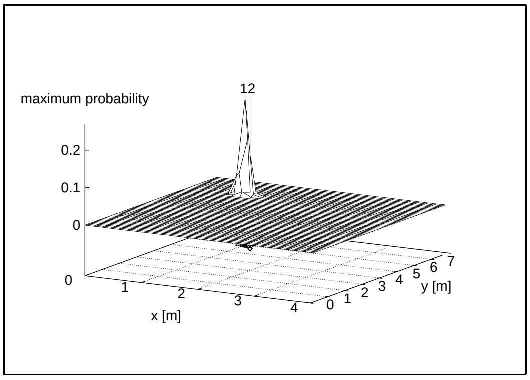

Figure 10. Probability distribution after 12 steps

[9] Illa Nourbakhsh, Rob Powers, and Stan Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, Summer 1995.

[10] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[11] 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.

[12] Reid Simmons. The 1994 AAAI robot competition and exhibition. *AI Magazine*, 16(2):19–29, Summer 1995.

[13] Reid Simmons and Sven Koenig. Probabilistic robot navigation in parially observable environments. In *Proc. International Joint Conference on Artificial Intelligence*, 1995.

[14] Harold W. Sorensen. *Kalman Filtering: Theroy and Application*. IEEE Press, 1985.

[15] Sebastian Thrun. Exploration and model building in mobile robot domains. In *Proceedings of the ICNN-93*, pages 175–180, San Francisco, CA, March 1993. IEEE Neural Network Council.

[16] 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.