Probabilistic Mapping Of An Environment By A Mobile Robot

Sebastian Thrun

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Dieter Fox Wolfram Burgard

Computer Science Department III
University of Bonn
D–53117 Bonn, Germany

Abstract

This paper addresses the problem of building large-scale maps of indoor environments with mobile robots. It proposes a statistical approach that phrases the map building problem as a constrained maximum-likelihood estimation problem, for which it devises a practical algorithm. Experimental results in large, cyclic environments illustrate the appropriateness of the approach.

1 Introduction

The problem of acquiring maps in large-scale indoor environments has received considerable attention in the mobile robotics community. The problem of map building is the problem determining the location of entities-of-interest (such as: landmarks, obstacles) in a global frame of reference (such as a Cartesian coordinate frame). To build a map of its environment, a robot must know where it is. Since robot motion is inaccurate, the robot must solve a concurrent localization problem, whose difficulty increases with the size of the environment (and specifically with the size of possible cycles therein).

This paper investigates a specific version of the map building problem, in which the robot can observe landmarks (which have to be mapped). It presents an algorithm for landmark-based map acquisition and concurrent localization, which is based on a statistical account on robot motion and perception. Our approach poses the problem of map building is as a maximum likelihood estimation problem, where both the location of landmarks and the robot's position have to be estimated. Likelihood is maximized under probabilistic constraints that arise from the physics of robot motion and perception. Following [6, 15, 16], the high-dimensional maximum likelihood estimation problem is solved efficiently using the Baum-Welch (or alpha-beta) algorithm [13]. Baum-Welch alternates an "expectation step" (E-step) and a "maximization step" (M-step). In the E-step, the current map is held constant, the probability distributions are calculated for past and current robot locations. In the M-step, the most likely map is computed based on the estimation result of the E-step. By alternating both steps, the robot simultaneously improves its localization and its map. The probabilistic nature of the estimation algorithm makes it considerably robust to ambiguities and noise, both in the odometry and in perception. It also enables the robot to revise past location estimates as new sensor data arrives.

The paper also describes some experimental results obtained with a RWI B21 robot in several indoor environments. In our experiments, a human operator manually chose an indistinguishable set of landmarks, and informed the robot (via button press) of the presence or absence of a landmark. One of the environments contains a cycle of size 60 by 25 meter, which has been mapped successfully despite significant odometric error. The approach has been integrated into a conventional method for building occupancy grid maps [18], for which results are reported as well.

2 The Probabilistic Model

This section describes our probabilistic model of the two basic aspects involved in mapping: motion and perception. These models together with the data (see next section) define the basic likelihood function, according to which maps are built.

2.1 Robot Motion

Let ξ and ξ' denote robot locations in x-y-θ space, and let u denote a control (motion command), which consists of a combination of rotational and translational motion. Since robot motion is inaccurate, the effect of a control u on the robot's location ξ is modeled by a conditional probability density

$$P(\xi'|u, \xi) \tag{1}$$

which determines the probability that the robot is at location ξ', if it previously executed control u at location ξ. $P(\xi'|u, \xi)$ imposes probabilistic constraints between robot positions at

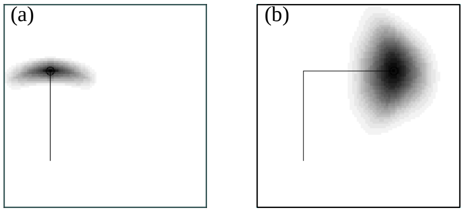

Figure 1: Probabilistic model of robot motion: Accumulated uncertainty after moving as shown: (a) 40 meter, (b) 80 meter.

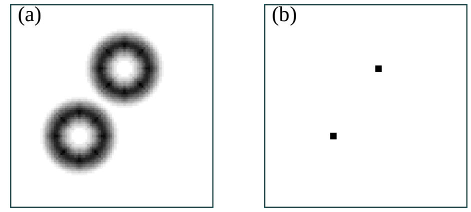

Figure 2: Probabilistic model of robot perception: (a) uncertainty after sensing a landmark in 5 meter distance, (b) the corresponding map.

different points in time. If $P(\xi)$ is the probability distribution for the robot's location before executing an control u,

$$P(\xi') \quad := \quad \int P(\xi'|u,\xi)\ P(\xi)\ d\xi \qquad (2)$$

is the probability distribution after executing that control. Figure 1 illustrates the motion model. In Figure 1a, the robot starts at the bottom location (in a known position), and moves as indicated by the vertical line. The resulting probability distribution is shown by the grey values in Figure 1a: The darker a value, the more likely it is that the robot is there. Figure 1b depicts this distribution after two motion commands. Of course, Figure 1 (and various other figures in this paper) show only 2D projections of $P(\xi)$, as $P(\xi)$ is three-dimensional.

2.2 Robot Perception

Our approach assumes that the robot can observe landmarks. More specifically, we assume that the robot is given a method for estimating the *type*, the *relative angle* and an *approximate distance* of nearby landmarks. For example, such landmarks might be Choset's "meet points" [3] (see also Kuipers's and Matarić's work [7, 10]), which correspond to intersections or dead ends in corridors and which can be detected very robustly.

In our probabilistic framework, landmarks are not necessarily distinguishable; in the most difficult case, landmarks are entirely indistinguishable. It is also assumed that the perceptual component is erroneous—the robot might misjudge the angle, distance, or type of landmark. Thus, the model of robot perception is modeled by a conditional probability:

$$P(o|\xi, m). \qquad (3)$$

Here o denotes a landmark observation, and m denotes the map of the environment (which contains knowledge about the exact location of all landmarks). $P(o|\xi, m)$ determines the likelihood of making observation o when the robot is at location ξ according to the model m.

The perceptual model imposes probabilistic constraints between the map, m, and the robot's location, ξ. According to Bayes rule, the probability of being at ξ when the robot observes o is given by

$$\begin{aligned} P(\xi|o,m) &= \frac{P(o|\xi,m)\ P(\xi|m)}{\int P(o|\xi',m)\ P(\xi'|m)\ d\xi'} \\ &= \eta\ P(o|\xi,m)\ P(\xi|m) \qquad (4) \end{aligned}$$

Here $P(\xi|m)$ measures the probability that the robot is at ξ prior to observing o and η is a normalizer that ensures that the left-hand probabilities in (4) sum up to 1. Equation (4) implies that after observing o, the robot's probability of being at ξ is proportional to the product of $P(\xi|m)$ and the perceptual probability $P(o|\xi, m)$.

Figure 2a illustrates the effect of Equation (4) for a simple example. Shown there is the distribution $P(\xi|o, m)$ that results, if the robot initially has no knowledge as to where it is (i.e., $P(\xi|m)$ is uniformly distributed), and if it perceives a landmark approximately 5 meters ahead of it, in a world m that contains exactly two indistinguishable landmarks. This world is shown in Figure 2b. The circles in Figure 2a indicate that the robot is likely to be approximately 5 meter away from a landmark—although there is a residual non-zero probability for being at other location, since the robot's perceptual routines might err.

3 Maximum Likelihood Estimation

The key idea is to build maps from data by maximizing the likelihood of the map under the data. The data is a sequence of control interleaved with observations. Without loss of generality, let us assume that motion and perception are alternated, i.e., that the data available for mapping is of the form

$$d \quad = \quad \{o^{(1)}, u^{(1)}, o^{(2)}, u^{(2)}, \ldots o^{(T-1)}, u^{(T)}, o^{(T)}\}. \qquad (5)$$

T denotes the total number of steps.

The estimation algorithm alternates two different estimation steps, the E-step and the M-step. In the E-step, probabilistic estimates for the robot's locations at the various points in times are estimated based on the currently best available map (in the first iteration, there is none). In the M-step, a maximum likelihood map is estimated based on the locations computed in the E-step. The E-step can be interpreted as a localization step with a fixed map, whereas the M-step implements a mapping step which operates under the assumption that the robot's locations (or, more precisely, probabilistic estimates thereof) are known. Iterative application of both rules leads to a refinement of both, the location estimates and the map. This algorithm can be shown to converge to a local optimum in likelihood space.

3.1 The E-Step

In the E-step, the current-best map m and the data are used to compute probabilistic estimates $P(\xi^{(t)}|d, m)$ for the robot's position $\xi^{(t)}$ at $t = 1, \ldots, T$. $P(\xi^{(t)}|d, m)$ can be expressed as product of two terms

$$P(\xi^{(t)}|d, m) = \underbrace{P(\xi^{(t)}|o^{(1)}, \ldots, o^{(t)}, m)}_{:=\alpha_\xi^{(t)}} \underbrace{P(\xi^{(t)}|u^{(t)}, \ldots, o^{(T)}, m)}_{:=\beta_\xi^{(t)}} \quad (6)$$

Both terms, $\alpha_\xi^{(t)}$ and $\beta_\xi^{(t)}$, are computed separately, where the former is computed forward in time and the latter is computed backwards in time. The reader should notice that the computation of the α-values is a version of *Markov localization*, which has recently been used with great success by various researchers [1, 6, 12, 17]. The β-values add additional knowledge to the robot's position, typically not captured in Markov-localization. They are, however, essential for revising past belief based on sensor data that was received later in time, which is a necessary prerequisite of building large-scale maps.

Computation of α-values: Since initially, the robot is assumed to be at the center of the global reference frame, $\alpha_\xi^{(1)}$ is given by a Dirac distribution centered at $(0, 0, 0)$:

$$\alpha_\xi^{(1)} = P(\xi^{(1)}|d, m) = \begin{cases} 1, & \text{if } \xi^{(1)} = (0,0,0) \\ 0, & \text{if } \xi^{(1)} \neq (0,0,0) \end{cases} \quad (7)$$

All other $\alpha_\xi^{(t)}$ are computed recursively:

$$\alpha_\xi^{(t)} = \eta\, P(o^{(t)}|\xi^{(t)}, m) \int P(\xi^{(t)}|u^{(t-1)}, \xi^{(t-1)})\; \alpha_\xi^{(t-1)}\; d\xi^{(t-1)} \quad (8)$$

Here η is a normalizer that ensures that the resulting probabilities sum up to 1. A detailed derivation of (8), which follows directly from (2) and (4), can be found in [19].

Computation of the β-values: The computation of $\beta_\xi^{(t)}$ is completely analogous, but backwards in time. The initial $\beta_\xi^{(T)}$, which expresses the probability that the robot's final position is ξ is uniformly distributed ($\beta_\xi^{(T)}$ does not depend on data). All other β-values are computed in the following way:

$$\beta_\xi^{(t)} = \eta \int P(\xi^{(t+1)}|u^{(t)}, \xi^{(t)})\; P(o^{(t+1)}|\xi^{(t+1)}, m)\; \beta_\xi^{(t+1)} d\xi^{(t+1)} \quad (9)$$

The derivation of this equation is analogous to that of the α-values. The result of the E-step, $\alpha_\xi^{(t)} \cdot \beta_\xi^{(t)}$, is an estimate of the robot's locations at the various points in time t.

3.2 The M-Step

The M-step computes the most likely map based on the probabilities computed in the E-step.

Without loss of generality, we assume that there are n different *types* of landmarks (for some value n), denoted $l_1, \ldots, l_n$. The set $L = \{l_1, \ldots, l_n, l_*\}$ is the set of *generalized landmarks types*, which includes l_*, the "no-landmark." A probabilistic *map* of the environment is an assignment of probabilities $P(m_{xy} = l)$ for $l \in L$, where $\langle x, y\rangle$ is a location measured in global coordinates, and m_{xy} is a random variable that corresponds to the generalized landmark type at $\langle x, y\rangle$. The M-step computes the most likely map under the assumption that $\alpha_\xi^{(t)} \cdot \beta_\xi^{(t)}$ accurately reflects the likelihood that the robot was at $\xi^{(t)}$ at time t.

Following [13], the maximum likelihood map is computed according to the weighted likelihood ratio

$$P(m_{xy} = l|d) = \frac{\text{\# of times } l \text{ was observed at } \langle x, y\rangle}{\text{\# of times something was observed } \langle x, y\rangle}$$

which is obtained by

$$\frac{\sum_{t=1}^{T} \int P(m_{xy} = l|o^{(t)}, \xi^{(t)})\; \alpha_\xi^{(t)}\beta_\xi^{(t)}\; d\xi^{(t)}}{\sum_{t=1}^{T} \sum_{l' \in L} \int P(m_{xy} = l'|o^{(t)}, \xi^{(t)})\; \alpha_\xi^{(t)}\beta_\xi^{(t)}\; d\xi^{(t)}} \quad (10)$$

where

$$P(m_{xy} = l|o^{(t)}, \xi^{(t)}) = \frac{P(o^{(t)}|m_{xy} = l, \xi^{(t)})}{\sum_{l'} P(o^{(t)}|m_{xy} = l', \xi^{(t)})} \quad (11)$$

While these equations look complex, they basically amount to a frequentist maximum-likelihood estimation. Equation (10) counts how often the generalized landmark l was observed for location $\langle x, y\rangle$, divided by the number *some* generalized landmark was observed for that location. Each count is weighted by the probability that the robot's was at a location ξ where it could observe something about $\langle x, y\rangle$. Frequency counts are maximum likelihood estimators. Thus, the M-step determines the most likely map from the position estimates computed in the E-step. By alternating both steps, both the localization estimates and the map are gradually improved (see also [13]).

4 Efficiency Considerations

In our implementation, all probabilities are represented by discrete grids. Thus, all integrals are replaced by sums in all equations above. Maps of size 90 by 90 meter with a spatial resolution of 1 meter and an angular resolution of 5° were used throughout all experiments reported here (unless otherwise noted). Our implementation employs a variety of "tricks" for efficient storage and computation:

- **Caching.** The motion model $P(\xi|u,\xi')$ is computed in advanced for each control in d and cached in a look-up table.
- **Exploiting symmetry.** Symmetric probabilities are stored in a compact manner.
- **Coarse-grained temporal resolution.** Instead of estimating the location at each individual micro-step, locations are only estimated if at least one landmark has been observed, or if the robot moved 20 meter. In between, position error is interpolated linearly.
- **Selective computation.** Computation focuses on locations ξ whose probability $P(\xi)$ is larger than a threshold: $P(\xi)$ must be larger or equal to $.001 \max_{\xi'} P(\xi')$.
- **Selective memorization.** Only a subset of all probabilities are stored for each $P(\xi)$, namely those that are above the threshold described above. This is currently implemented with a generalized version of bounding boxes.

These algorithmic "tricks" were found to lower memory requirements by a factor of $2.98 \cdot 10^8$ (in our largest experiment) when compared to a literal implementation of the approach. The computation was accelerated by a similar factor. The resulting algorithm does not work in real-time; however, all maps shown in this paper were produced in less than an hour on a low-end PC.

5 Results

The approach was tested using a B21 mobile robot, manufactured by Real World Interface, Inc. Data was collected by joy-sticking the robot through its environment and using odometry (shaft encoders) to re-compute the corresponding control. While joy-sticking the robot, a human chose and marked a collection of significant locations (which roughly corresponded to the meet-points described in [3, 7]), and informed the robot by pressing a button every time the robot crossed a landmark location. To test the most difficult case, we assumed that the landmarks were generally indistinguishable.

Figure 3a shows one of our datasets, collected in our university buildings. The circles mark landmark locations. What makes this particular environment difficult is the large circular hallway (60 by 25 meter). When traversing the

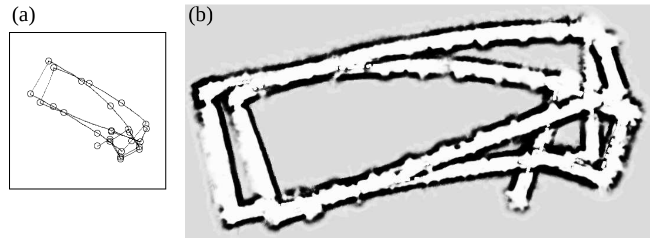

Figure 3: (a) Raw data (2,972 controls). The box size is 90 by 90 meters. Circles indicate the locations where landmarks were observed. The data indicates systematic drift, in some of the corridors. The final odometric error is approximately 24.9 meter. (b) Occupancy grid map, constructed from sonar measurements.

Figure 4: (a) Maximum likelihood map, along with the estimated path of the robot. (b) Occupancy grid map constructed using these estimated locations.

circle for the first time, the robot cannot exploit landmarks to improve its location estimates;thus, it accumulates odometric error. As Figure 3a illustrates, the odometric error is quite significant; the final odometric error is approximately 24.9 meter. Since landmarks are indistinguishable, it is difficult to determine the robot's position when the circle is closed for the first time (here the odometric error is larger than 14 meter). Only as the robot proceeds through known territory it can use its perceptual clues to estimate where it is (and was), in order to build a consistent map.

Figure 4a shows the maximum likelihood map along with the estimated path of the robot. This map is topologically correct, and albeit some bents in the curvature of the corridors (to avoid those, one has to make further assumptions), the map is indeed good enough for practical use. This result demonstrates the power of the method. In a series of experiments with this data set, we consistently found that the principle topology of the environment was already known after two iterations of the Baum-Welch algorithm; after approximately four iterations, the location of the landmarks were consistently known with high certainty.

The result of the estimation routine can be used to build more accurate occupancy grid maps [4, 11]. Figure 4b shows an occupancy grid map constructed from sonar measurements (using a ring of 24 Polaroid sonar sensors), using the guessed maximum likelihood positions as input to the mapping software described in [18]. In comparison, Figure 3b shows the same map using the raw, uncorrected data. The map constructed from raw data is unusable for navigation, whereas the corrected map is sufficient for our current navi-

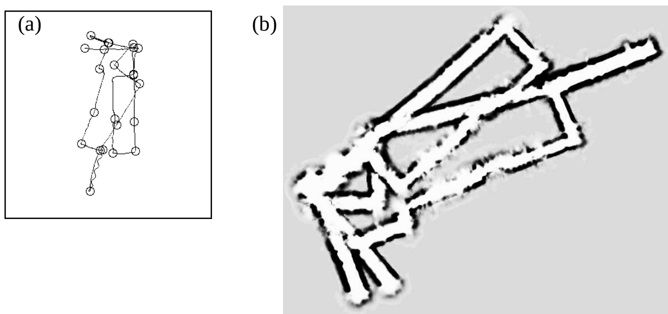

Figure 5: (a) A second data set (2,091 controls, box size 90 by 90 meter), and (b) occupancy grid map, constructed from sonar measurements.

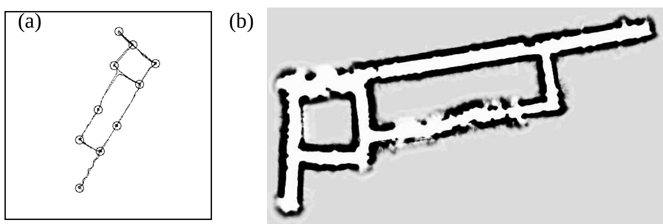

Figure 6: (a) Maximum likelihood map, along with the estimated path of the robot, and (b) the resulting occupancy grid map.

gation software.

Figures 5 and 6 show results obtained in a different part of the building. In this run, one of the corridors was extremely populated, as the “fuzziness” of the occupancy grid map suggests. The floor material in both testing environments consisted of carpet and tiles.

Figures 7 and 8 show results obtained in the National Museum of American History, obtained in preparation for an upcoming robot exhibition. During data collection, the robot suffered a final odometric error of 70 meters and 180 degrees. This dataset was first pre-filtered by a recently developed routine for self-calibrating of kinematic models [14], then processed using the algorithm described in this paper. The final result is a accurate map. No experiments were performed using the raw dataset.

More results can be found in [19].

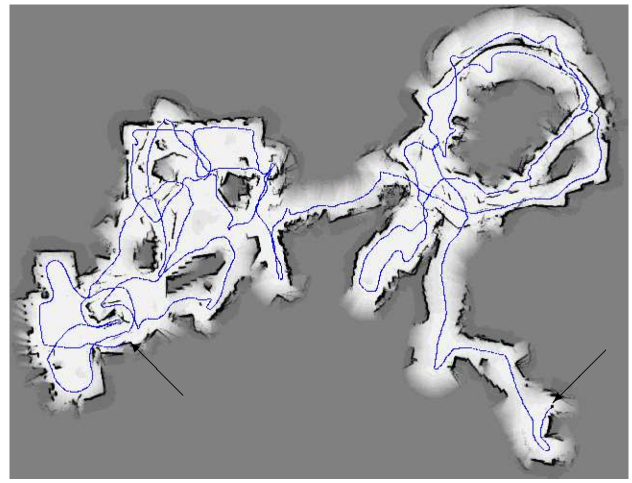

Figure 7: Dataset collected in the National Museum of American History (Smithsonian Institution, Washington, DC). The dataset contains 50 landmark observations and 2,667 controls. The final odometric error is approximately 70 meters and 180 degree; the two marked locations (arrows) are actually identical.

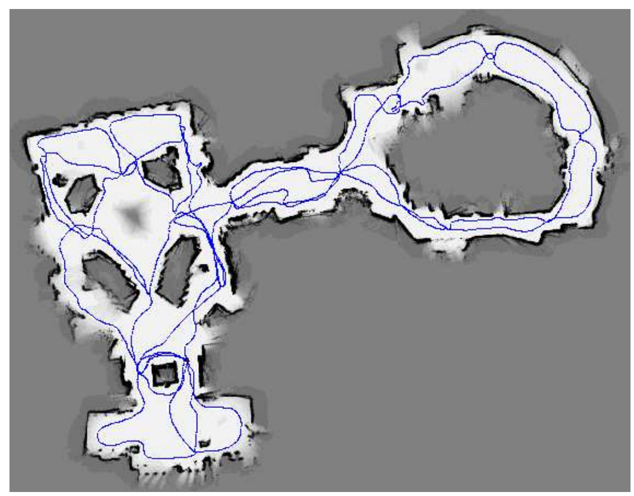

Figure 8: Mapping result, after first compensating for systematic rotational drift and then applying the algorithm described here. This map is approximately 85 meters long.

6 Related Work

Over the last decade, there has been a flurry of work on map building for mobile robots (see e.g., [2, 8, 18]). As noticed by Lu and Milios [9], the dominating paradigm in the field is *incremental:* Robot locations are estimated as they occur; the majority of approaches lacks the ability to use sensor data for revising past location estimates. A detailed survey of recent literature on map building can be found in [18]. The approach proposed there, however, is also incremental and therefore incapable of dealing with situations such as the ones described in this paper.

Recently, several groups have proposed algorithms that revise estimates backwards in time [5, 6, 15, 9]. These approaches are similar to the one proposed here in that they use similar statistical methods for constructing maps. They differ in the way the represent locations and maps. The approaches in [6, 15] use Hidden Markov Models as their baseline representation, which make it difficult to embed geometry in a mathematically correct way. The approaches in [5, 9], on the other hand, represent locations of the robot and the obstacles using Kalman filters. While those approaches can generate maps with floating-point accuracy, they are limited in their ability to represent ambiguities. As a result, the odometric error must be quite limited.

7 Discussion

This paper proposed a probabilistic approach to building large-scale maps of indoor environments with mobile robots. It phrased the problem of map building as a maximum likelihood estimation problem, where robot motion and perception impose probabilistic constraints on the map. It then devised an efficient algorithm for maximum likelihood estimation. Simplified speaking, this algorithm alternates localization and mapping, thereby improving estimates of both the map and the robot's locations. Experimental results in large, cyclic environments demonstrate the appropriateness and robustness of the approach.

The basic approach can be extended in several interesting directions.

The current approach is "passive", i.e., it does not restrict in any way how the robot is controlled. Thus, the approach can be combined with one of the known sensor-based exploration techniques. One possibility, which has not yet been implemented, would be to combine the current approach with Choset's sensor-based covering algorithm [3].

Our current implementation also relies on humans to identify landmarks. While this is reasonable when mapping an environment collaboratively with a human, it is impractical if the robot is to operate autonomously. The lack of a landmark-recognizing routine is purely a limitation of our current implementation, not of the general algorithm. For example, Choset's sensor-based covering algorithm [3] automatically detects and navigates to so-called *meet-points*. Meet-points correspond to intersections, corners, and dead-ends (see also [7]). We conjecture that a combined algorithm, using Choset's approach for exploration and meet-point detection and our approach for mapping, would yield an algorithm for fully autonomous exploration and mapping.

Acknowledgment

This research benefited from discussions with Howie Choset, Keiji Nagatani, and Hagit Shatkay, which are gratefully acknowledged.

This research was sponsored in part by Daimler-Benz Research (via Frieder Lohnert) and the Defense Advanced Research Projects Agency (DARPA) via the Airforce Missile System Command under contract number F04701-97-C-0022. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of Daimler-Benz Research, the Airforce Missile System Command, or the United States Government.

References

[1] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Menlo Park, August 1996. AAAI, AAAI Press/MIT Press.

[2] R. Chatila and J.-P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985.

[3] H. Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, California Institute of Technology, 1996.

[4] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.

[5] J.-S. Gutmann. Vergleich von algorithmen zur selbstlokalisierung eines mobilen roboters. Master's thesis, University of Ulm, Ulm, Germany, 1996. (in German).

[6] S. Koenig and R. Simmons. Passive distance learning for robot navigation. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

[7] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.

[8] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):89–96, 1992.

[9] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[10] M.J. Matarić. Interaction and intelligent behavior. Technical Report AI-TR-1495, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA, 1994.

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

[12] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, Summer 1995.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*. IEEE, 1989. IEEE Log Number 8825949.

[14] N. Roy and S. Thrun. Online self-calibration for mobile robots. Submitted for publication, 1998.

[15] H Shatkay and L. Kaelbling. Learning topological maps with weak local odometric information. In *Proceedings of IJCAI-97*. IJCAI, Inc., 1997. 1997.

[16] H. Shatkay and L.P. Kaelbling. Learning hidden markov models with geometric information. Technical Report CS-97-04, Computer Science Department, Brown University, Providence, RI, April 1997.

[17] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of IJCAI-95*, pages 1080–1087, Montreal, Canada, August 1995. IJCAI, Inc.

[18] S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, to appear.

[19] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, to appear.