Finding Landmarks for Mobile Robot Navigation

Sebastian Thrun

Computer Science Department and Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Localization addresses the problem of determining the position of a mobile robot from sensor data. This paper presents an algorithm, called BaLL, which enables a mobile robot to learn a set of landmarks used in localization and to learn how to recognize them using artificial neural networks. BaLL is based on a statistical localization approach. It is applicable to a large variety of sensors and environments. Experiments with a mobile robot equipped with sonar sensors and a camera illustrate that BaLL identifies highly useful landmarks.

1 Introduction

To operate successfully, mobile robots must know where they are. *Mobile robot localization*, that is the process of determining and tracking the position (location) of mobile robots relative to their environments, has received considerable attention over the past few years. Accurate localization is a key prerequisite for successful navigation in large-scale environments, particularly when global models are used (such as maps, drawings, topological descriptions, CAD models, *cf.* various chapters in [12]).

A recent book on mobile robot localization [2] demonstrates the importance of the problem and provides a unique description of the state-of-the-art. Most successful approaches utilize landmarks. *Landmarks*, for the purpose of this paper, will be defined as stationary objects, places, or families thereof whose locations are usually known (*cf.* [17, 15]). Many of the early landmark-based approaches reviewed in [2] require artificial landmarks such as bar-code reflectors [7], reflecting tape, ultrasonic beacons, or visual patterns that are easy to recognize, such as black rectangles with white dots [1]. Some of the more recent approaches use more natural landmarks for localization, which do not require modifications of the environment (such as ceiling lights [11] or gateways/doors/vertical objects [13, 14, 20, 19]). *Model matching*, an alternative approach to localization, often memorize and compare significant places for localization [5, 18, 23].

It is common practice that a human expert selects the landmarks and provides computer routines for their recognition. As a result, these approaches are tight to a particular sensor type and a particular environment type. For example, if ceiling lights are used as primary landmark—an approach which has become popular in recent years—, localization will fail if the environment does not contain ceiling lights, or the robot does not a possess a sensor capable of detecting them. In addition, the perceptual apparatus of humans differs substantially from that of robots, and landmarks which work well for humans are often not appropriate for robots. All these arguments illustrate the need for methods which enable robots to choose landmarks by themselves.

This paper presents an algorithm called BaLL (short for Bayesian Landmark Learning). BaLL enables a robot to learn its landmarks by itself, and to learn routines for their recognition (see also [9, 22]). In order to do so, the robot has to be provided with a set of sensor snapshots labeled with the position at which they were taken. These data is used (1) to establish correspondence between the sensors of the robot and the world coordinates, and (2) to train artificial neural networks that recognize landmarks. Learning is driven by a simple principle: The minimization of the localization error.

An experimental study compares the performance of landmarks selected by human experts to landmarks selected with the BaLL algorithm. In three experiments, landmarks discovered by BaLL outperform alternative, hand-selected landmarks (doors and ceiling lights).

The results of this study illustrate the two main innovations: First, the burden of having to select landmarks by hand and having to code routines for their recognition is eliminated. Second, our method enables a robot to customize its localization algorithm to a particular environment and to its sensors, making it more widely applicable than methods that rely on static, built-in landmarks.

2 Approach

The input to the BaLL algorithm is a set of sensor snapshots, labeled by the position at which they were taken. The data are used to establish the correspondence between world coordinates and sensor values and to select appropriate landmarks. After learning, our approach enables a robot to estimate is position from sensor data.

2.1 Markov Localization

BaLL is based on Markov localization. Markov localization has recently been employed successfully in various state-of-the-art mobile robot systems (*e.g.*, [3, 10, 13, 16, 20, 21]). Figure 1 illustrates the basic method. The key idea of Markov localization is to compute a probability density over all positions, denoted by $Bel(\xi)$. $Bel(\xi)$ expresses the robot's subjective belief (uncertainty) that its current position is ξ, where ξ denotes an arbitrary position. Initially, $Bel(\xi)$, reflects the a priori uncertainty: If the robot knows its initial position and the goal of localization is to compensate slippage and drift, $Bel(\xi)$ is initialized with a point-centered distribution that has a peak at the correct position. If the robot has no initial knowledge about its position, $Bel(\xi)$ is initialized with a uniform distribution. The latter situation is depicted in Figure 1a.

The belief $Bel(\xi)$ is updated whenever the robot senses or moves:

Sensing: The robot's sensors provide information concerning the presence or absence of landmarks. Suppose the robot can recognize n different landmarks. Let $s \in \{0,1\}^n$ denote the *landmark configuration vector* for all n landmarks, *i.e.*, the i-th component of s is 1 if and only if the i-th landmark has been observed in the most recent sensor reading. Then, s is used to refine the position density in the following way:

$$Bel(\xi) \quad \longleftarrow \quad \frac{P(s|\xi)\ Bel(\xi)}{P(s)} \tag{1}$$

$$\text{where}\ \ P(s) \quad = \quad \int P(s|\bar{\xi})\ Bel(\bar{\xi})\ d\bar{\xi} \tag{2}$$

$P(s)$ is a normalizer which ensures that $Bel(\xi)$ integrates to 1. Here $Bel(\xi)$ on the right-hand side of (1) denotes the belief prior to sensing, and $P(s|\xi)$ denotes the probability of observing the landmark configuration s at position ξ. Update equation (1) is illustrated by the transition of Figure 1a to Figure 1b. Assuming that the robot just observed the landmark "door," it multiples its prior belief by the probability of observing a door. Since the environment contains three indistinguishable doors, the resulting belief has three peaks at the three door positions.

The probability $P(s|\xi)$ in Equation (2) can be viewed as a *map of the environment*, since it specifies the likelihood of observing each individual landmark at each position ξ. For the example in Figure 1, $P(s|\xi)$ specifies the chances of observing doors at the various positions ξ. In general, $P(s|\xi)$ can be modeled by parametric densities such as Gaussians (Kalman filters) or mixtures thereof [21], or by splines or piecewise constant functions. In our implementation, $P(s|\xi)$ is represented by a piecewise constant function that is estimated from data [22].

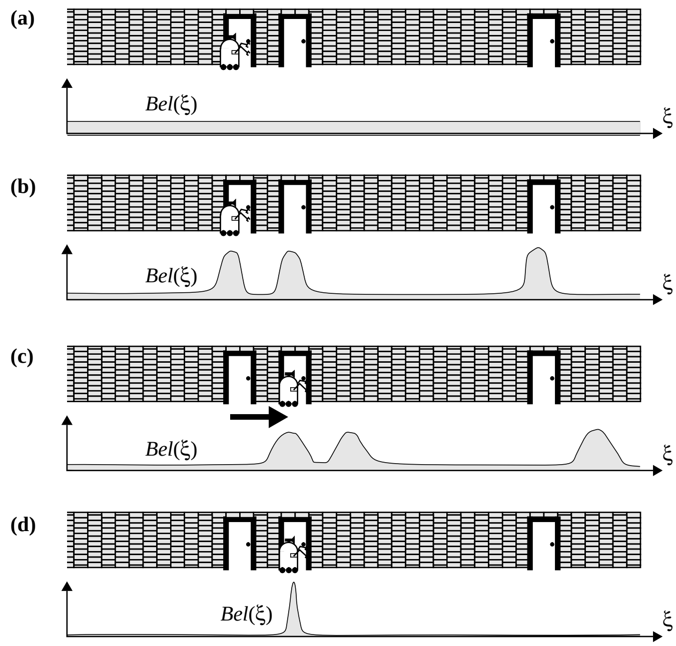

Figure 1: Markov localization, see text.

Motion: Whenever the robot executes a motion command, denoted by u, it updates its internal belief according the following rule:

$$Bel(\xi) \quad \longleftarrow \quad \int P(\xi|u,\xi')\ Bel(\xi')\ d\xi' \tag{3}$$

Here $Bel(\xi')$ on the right-hand side of (3) is the robot's belief prior to executing u. The conditional probability $P(\xi|u,\xi')$ denotes the probability that the motion command u, if executed at ξ', makes the robot move to ξ. For example, if u is the command for moving 1 meter forward, $P(\xi|u,\xi')$ will be a bell-shaped density centered around $\xi = \xi' + 1\text{m}$. An approximate version of $P(\xi|u,\xi')$ can easily be derived from the robot's kinematics.

Figure 1c illustrates the effect of robot motion on the internal belief $Bel(\xi)$. Here the robot moves forward. As a result, the density is shifted towards the right and flattened out a little, since robot motion introduces uncertainty. Figure 1d illustrates the effect of a second sensor query. Let us suppose that the robot now detects a second door next to it and applies Equation (1) to update $Bel(\xi')$. The resulting density contains a single peak at the correct position, which can be interpreted that the robot now knows where it is.

2.2 The Average Localization Error

The key to automatically selecting landmarks is to *minimize the localization error after taking a sensor snapshot*. To measure this error, let us examine the update Equation (1), which specifies the way BaLL updates the internal belief in response to a sensor reading. Equation (1) transforms the belief $Bel(\xi)$ prior to sensing to a new, refined belief, which is usually more accurate. The error in this new (posterior) belief depends on the landmarks employed by the robot. It is this error that BaLL seeks to minimize.

This argument will now be formalized. Let ξ^* denote the *true* position of the robot, and let $e(\xi^*, \xi)$ denote an error function for measuring the error between the true position ξ^* and an arbitrary other position ξ (e might be the Kullback-Leibler divergence or a metric distance). The *average (Bayesian) localization error at* ξ^*, denoted by $E(\xi^*)$, is obtained by integrating the error e over all position ξ, weighted by the corresponding belief $Bel(\xi)$:

$$E_{\text{pri}}(\xi^*) \quad := \quad \int e(\xi^*, \xi)\ Bel(\xi)\ d\xi \tag{4}$$

If this error is computed *before* taking a sensor snapshot, it is called the *prior error at* ξ^* with respect to the next sensor reading; hence the notation $E_{\text{pri}}(\xi^*)$.

We will now derive an expression that describes the error after taking a sensor snapshot. By definition, the robot will sense the landmark configuration s with probability $P(s|\xi^*)$ when being at ξ^*. In response, it will update its belief according to Equation (1). The *posterior error at* ξ^*, which is the error the robot is expected to suffer at ξ^* *after* sensing, is obtained by applying the update rule (1) to the error (4)

$$E_{\text{post}}(\xi^*) \ = \sum_s \int e(\xi^*, \xi) \frac{P(s|\xi)\ Bel(\xi)}{P(s)} P(s|\xi^*)\ d\xi \tag{5}$$

E_{post} is averaged over all possible landmark vectors s, weighted by their likelihood $P(s|\xi^*)$. The normalizer $P(s)$ is computed just like in Equation (1).

Thus far, the posterior error E_{post} corresponds to a single position ξ^* only. By averaging over all possible positions ξ^*, weighted by their likelihood of occurrence $P(\xi^*)$, we obtain the *average posterior error*:

$$E_{\text{post}} \quad := \quad \int E_{\text{post}}(\xi^*)\ P(\xi^*)\ d\xi^* \tag{6}$$

In the absence of any better knowledge, we will assume that $P(\xi^*)$ is *uniformly distributed*, *i.e.*, each robot position is equally likely. The posterior error E_{post} enables a robot to *compare* landmarks with one another. Put differently, by comparing the error E_{post} for different landmarks, the robot can decide which one yields more accurate localization.

The error E_{post} cannot be computed directly. BaLL approximates E_{post} using the data, as described in [22].

2.3 Neural Network Training

BaLL recognize landmarks with artificial neural networks, which map sensor scans (*e.g.*, camera images, sonar scans) to values between 0 and 1. Thus, every feature of the environment which can be recognized by a neural network can potentially be a landmark. Landmarks need not to correspond to individual objects in the environment; in fact, they might correspond to groups of objects or places with similar features (such as doors), even to features which would not necessarily be used as landmark by humans (such as a reflection on a shiny wall). The number of networks (denoted by n) and hence the number of different types of landmarks is specified by the user.

BaLL trains these networks by minimizing the average a posterior error E_{post}. The learning approach utilizes the fact that E_{post} is differentiable in the weights of the neural networks (see [22] for an exact derivation of the derivatives of E_{post}). To train the networks, each weight w_{ij} is adjusted in proportion to the negative gradient $-\nabla_{w_{ij}} E_{\text{post}}$ (just like weights and biases are adjusted in Back-Propagation): Repeated application of this gradient descent update rule leads to the nearest local minimum in weight space. The interesting thing here to notice is that the specific choice of landmarks emerges through minimizing the localization error. The BaLL training scheme differs from the conventional use of Backpropagation in that it does not require explicit *target patterns* for the individual networks; instead, the networks are trained so as to minimize the localization error.

3 Experimental Results

3.1 Three Setups

The central question addressed in this section is: *Will landmarks learned by our algorithm outperform landmarks carefully selected by a human expert?* In our experimental study we were therefore attempted to compare our method to other state-of-the-art localization algorithms.

Figure 2 depicts the robot used in our experiments. A total of three experimental comparisons was conducted, using three different experimental setups (and three different datasets). The first two comparisons were motivated by a mobile robot control system built by a different research team in our department. This team is primarily interested in reliable long-term mobile robot operation; for which reason it has operated a mobile robot almost on a daily basis over the last two years [20, 19]. Accurate localization is essential for reliable point-to-point navigation. Their localization algorithm is a version of Markov localization, using *doors* as their primary landmarks. In our building, doors are by far the most regular and the most easily distinguishable feature. In experimental setup #1, the camera of our robot was directed towards the outside wall of the building. Here the number of doors is

Figure 2: The RWI B21 robot used in our research.

considerably large. In setup #2, the camera was directed towards the interior of the building. Here doors are about twice as rare and approximately 50% wider. In both cases, the environment was dynamic and did not obey the Markov assumption: People frequently blocked the robot's sensors, doors were sometimes open, sometimes closed, and since our corridor possesses two large windows, the illumination depended on the time of day at which the data was recorded. Just like the other research team, we trained a neural network (with supervised learning) to recognize doors. After training, the network recognized doors with 97.7% accuracy, measured on an independent test set. The reader should notice, however, that the experimental setup was *not* identical to the one reported in [20, 19].

In the third comparison (setup #3), we used ceiling lights as landmark. Here the camera was pointed upward. Since ceiling lights are considerably easy to recognize, stationary, and barely blocked by obstacles, they have become popular for mobile robot localization. As in the other two comparisons, we trained a single network to recognize such lights.

We collected a total of 9,815 sensor snapshots, which were divided about equally between the three different datasets (3,232+3,110+3,473). In each experiment, approximately half the data was used for training, and the other half for testing. To collect the data efficiently, we repeatedly had our robot navigate in a 89 meter long segment of our building, using its built-in obstacle avoidance and wandering routines [8]. In each "run," the robot was started at the same initial position and sent down the same corridor in the same direction. Instead of measuring the exact position of the robot by hand (which is practically infeasible for the large number of positions used here), we used the robot's odometry to derive the position label. While odometry is certainly inaccurate in the long run, we found it to be sufficiently accurate in the short run. After some cleanup using the position tracking method described in [23], the position data was accurate within an estimated $\pm.5$m. Color camera images were pre-processed using various low-level feature extraction techniques. The final dimension of all sensor values (vision and sonar) was 164; thus, each neural network had 164 input units. We employed layered networks with six hidden units and a single output unit.

In all experiments, the robot moved primarily along the center of the corridor. This is a result of the navigation routines employed [8, 23], which keep the robot centered in the middle of the corridor unless a human blocks its path (which happened frequently). Using such a navigation system greatly reduced the number of examples required to estimate the density $P(s|\xi)$. Locations were represented in three dimensions. The particular error function $e(\xi^*, \xi)$ employed to compare different locations measured the distance a robot has to travel from ξ^* to ξ.

As far as the experimental methodology is concerned, it is important to notice that different landmarks cannot be compared by just a single experiment. This is because the utility of a landmark greatly depends on the prior uncertainty $Bel(\xi)$. Thus, to conduct as insightful a comparison as possible, we generally evaluated the utility of landmarks under several different uncertainties $Bel(\xi)$, all of which were uniformly distributed and centered around the true (unknown) position. In particular, these uniform distributions had the following widths: $[-1\text{m}, 1\text{m}]$, $[-2\text{m}, 2\text{m}]$, $[-5\text{m}, 5\text{m}]$, $[-10\text{m}, 10\text{m}]$, $[-50\text{m}, 50\text{m}]$, and $[-89\text{m}, 89\text{m}]$ (*i.e.*, global uncertainty).

3.2 Single Step Results

The central result of our study is: The landmarks selected by BaLL uniformly outperform the ones chosen manually. This result was confirmed in every single comparison.

In an initial series of experiments, only a single network was used, *i.e.*, $n = 1$. Figures 3-5 survey the main results obtained in the three different experimental setups (camera pointed to the outside wall, camera pointed to the inside wall, camera pointed towards the ceiling). The vertical axes measure the *error reduction*: If, for example, the prior error is four times as large as the posterior error after taking a sensor snapshot, the corresponding error reduction is 75%. The advantage of plotting the error reduction (over the absolute error) is that all results are in the same scale, which facilitates their comparison. The vertical axes in Figures 3-5 show results obtained for different prior uncertainties $Bel(\xi)$. In all diagrams, the dashed lines depict the error reduction obtained with the human-selected landmarks (door and ceiling lights), whereas the solid lines show the error reduction for landmarks selected by BaLL. 95% confidence bars are also plotted.

As can be seen from these figures, landmarks selected by BaLL are uniformly superior to landmarks selected by hand. For example, the dashed line in Figure 3 indicates that doors are best when the uncertainty of the robot is $\pm$2m. Here a single sensor reading reduces the error on average by 8.31%. BaLL finds a network (and hence a set of landmarks) which reduces the error by 14.9% for $\pm$2m uncertainty, under oth-

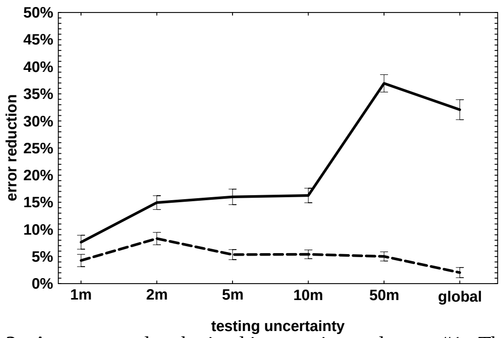

Figure 3: Average results obtained in experimental setup #1. The dashed (solid) line indicates the error reduction of manually (automatically) selected landmarks, for different prior uncertainties.

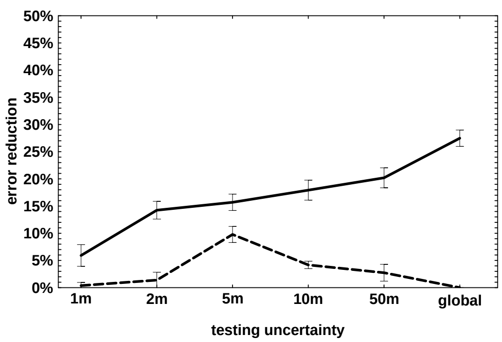

Figure 4: Results obtained for experimental setup #2.

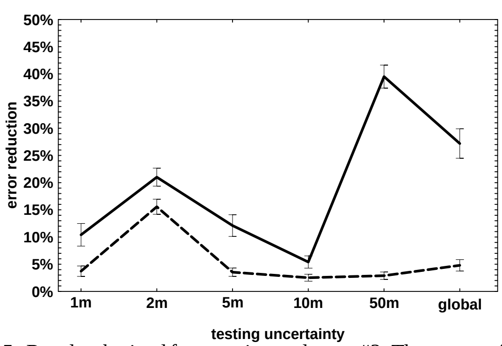

Figure 5: Results obtained for experimental setup #3. The poor performance for $\pm$10m is currently being analyzed in our lab.

erwise equal conditions, which is 1.8 times as good as the 8.31% reported above. The advantage of the BaLL-selected landmarks becomes even larger for more global uncertainties. For example, if the robot's uncertainty is $\pm$50m, the approach reduces the error by 36.9%, which is approximately 7.4 times as good the average reduction when doors are used as landmarks (5.02%).

The same result can be found in Figures 4 and 5. Here, too, the landmarks selected by BaLL are uniformly superior to the hand-selected ones. All differences in Figures 3-5 are statistically significant at the 95% level.

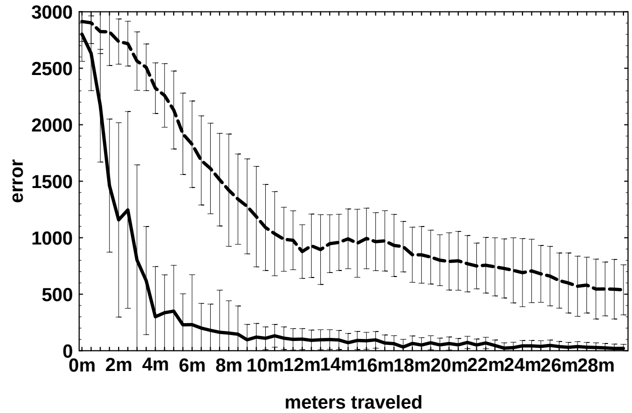

Figure 6: Absolute error as a function of meters traveled, averaged over 35 different experiments (with different starting points). The dashed line indicates the error when doors are used as landmarks, and the solid line corresponds to the error obtained when using BaLL.

3.3 Global Localization Results

In global localization, the robot does not know initially where it is; thus, its initial uncertainty is uniformly distributed [6]. As the robot moves, the internal belief is refined based on sensor readings. In our environment, a single sensor snapshot is usually insufficient to determine the position of the robot uniquely. Thus, multiple sensor readings must be integrated over time.

We conducted a series of 35 global localization experiments comparing the door-based localization with BaLL. In each experiment, the robot was started at a random position in the corridor. Sensor snapshots were taken every 0.5 meter. Figure 6 summarizes the result of the systematic comparison in setup #1. It shows the average error (in cm) as a function of the distance traveled, averaged over 35 different runs (with randomly chosen starting positions). The dashed line shows the localization error when doors are used as landmarks, whereas the solid line shows the error for landmarks found with BaLL ($n = 4$, $\pm$2m uncertainty). 95% confidence intervals are also shown. The results in Figure 6 demonstrate clearly the relative advantage of learning landmarks. After 30m of robot travel, the average error of the door-based approach is 7.46m. In comparison, BaLL attains the same accuracy after 4m, making it approximately 7.5 times as data-efficient.

Further results, which include a description of the particular landmarks learned by BaLL, can be found in [22].

4 Conclusion

We have presented BaLL, a localization method that enables a mobile robot to find its own landmarks, and to learn artificial neural network for their recognition. Equipped with a set of sensor snapshots labeled by their (approximate) position, our method trains neural networks to minimize the expected localization error after taking a sensor snapshot (posterior error). The resulting networks recognize landmarks which, according to the data, are most informative for estimating a robot's position. In an experimental study, landmarks selected by BaLL were empirically found to outperform landmarks selected by human experts.

A potential limitation of the current approach is its computational complexity. Training the networks consumes a considerable amount of time (between 30 minutes and 12 hours on a Pentium Pro). Once the networks are trained, estimating the map and updating the belief can easily be done in real-time. It should usually suffice to train he networks only once, in the beginning, so that after an initial training phase, all relevant computation can be performed in real-time. Another disadvantage is that BaLL does not allow people to influence which landmarks are selected. If certain landmarks are more reliable than others (*e.g.*, in dynamic environments), BaLL can only take this into account if this is reflected in the data set used for training.

BaLL enables robots to customize themselves to their environments. Any routine that relies on a static, built-in type of landmarks is bound to fail in environments which do not possess the respective landmarks. By providing a method that supports the automatic customization of a robot to its environment, we hope that BaLL provides a new level of flexibility for the design of future service robots (*e.g.*, service robots operated in private homes, whose design varies greatly from home to home).

Acknowledgments

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 author 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] J. Borenstein. *The Nursing Robot System*. PhD thesis, Technion, Haifa, Israel, June 1987.

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

[3] 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.

[4] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proceedings of IJCAI-97*. IJCAI, Inc., 1997. 1997.

[5] 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.

[6] S. Engelson and D. McDermott. Error correction in mobile robot map learning. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2555–2560, Nice, France, May 1992.

[7] H.R. Everett, D.W. Gage, G.A. Gilbreth, R.T. Laird, and R.P. Smurlo. Real-world issues in warehouse navigation. In *Proceedings of the SPIE Conference on Mobile Robots IX*, Boston, MA, November 1994. Volume 2352.

[8] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation*, 4(1), 1997.

[9] R. Greiner and R. Isukapalli. Learning to select useful landmarks. In *Proceedings of 1994 AAAI Conference*, pages 1251–1256, Menlo Park, CA, 1994. AAAI Press / The MIT Press.

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

[11] S. King and C. Weiman. Helpmate autonomous mobile robot navigation system. In *Proceedings of the SPIE Conference on Mobile Robots*, pages 190–198, Boston, MA, November 1990. Volume 2352.

[12] D. Kortenkamp, R.P. Bonassi, and R. Murphy, editors. *AI-based Mobile Robots: Case studies of successful robot systems*, Cambridge, MA, 1998. MIT Press. to appear.

[13] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 979–984, Menlo Park, July 1994. AAAI, AAAI Press/MIT Press.

[14] B. Kuipers and Y.-T. Byun. A robust qualitative method for spatial learning in unknown environments. In *Proceeding of Eighth National Conference on Artificial Intelligence AAAI-88*, Menlo Park, Cambridge, 1988. AAAI Press / The MIT Press.

[15] M. J. Matarić. A distributed model for mobile robot environment-learning and navigation. Master's thesis, MIT, Cambridge, MA, January 1990. also available as MIT AI Lab Tech Report AITR-1228.

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

[17] C.C. Presson and D.R. Montello. Points of reference in spatial cognition: Stalking the elusive landmark. *British Journal of Developmental Psychology*, 6:378–381, 1988.

[18] B. Schiele and J. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 1628–1634, San Diego, CA, May 1994.

[19] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, February 1997.

[20] 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.

[21] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehnicles*, pages 167–193. Springer-Verlag, 1990.

[22] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, to appear.

[23] 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.