Probabilistic Methods for Mobile Robot Mapping

Dieter Fox[1] **Wolfram Burgard**[2] **Sebastian Thrun**[1]

[1]Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

[2]Institut für Informatik III
University of Bonn
D-53117 Bonn, Germany

Abstract

The problem of map building is the problem of determining the location of entities-of-interest in a global frame of reference. Over the last years, probabilistic methods have shown to be well suited for dealing with the uncertainties involved in mobile robot map building. In this paper we introduce a general probabilistic approach to concurrent mapping and localization. This method poses the mapping problem as a statistical maximum likelihood problem, and devises an efficient algorithm for search in likelihood space. We furthermore address the problem of using occupancy grid maps for path planning in highly dynamic environments. The approaches have been tested extensively and several experimental results are given in the paper.

1 Introduction

The problem of acquiring maps in indoor environments has received considerable attention in the mobile robotics community. In its most general form, the problem of map building is the problem of determining the location of entities-of-interest (such as: landmarks, obstacles, objects) in a global frame of reference (such as a Cartesian coordinate frame or a topological graph). 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). Thus, the problem of map building is an example of a chicken-and-egg problem: To determine the location of the entities-of-interest, the robot needs to know where it is. To determine where it is, the robot needs to know the locations of the entities-of-interest.

Probabilistic methods have been shown to be well suited for dealing with the uncertainties involved in this problem. In this paper we present a statistical method for dealing with the general problem of concurrent localization and map building. The method is based on a variant of the EM algorithm, which is an efficient hill-climbing method for maximum likelihood estimation in high-dimensional spaces. In the context of mapping, EM iterates two alternating steps: a *localization step*, in which the robot is localized using a previously computed map, and a *mapping step*, which computes the most likely map based on the previous pose estimates. The resulting approach can be applied to different kinds of sensors and is general enough for topological *and* metric map building (see also [Thrun *et al.*, 1998b; 1998c; Burgard *et al.*, 1999]).

A very popular approach to metric maps are occupancy grid maps, which were originally proposed in [Elfes, 1989; Moravec, 1988] and which since have been employed successfully in numerous mobile robot systems. Occupancy grids are designed to estimate the occupancy of all $\langle x, y \rangle$-locations in the environment. The key advantages of such maps are that they (1) are easy to build and maintain, (2) facilitate path planning, and (3) enable accurate position estimates (see e.g. [Burgard *et al.*, 1996; Thrun, 1998]). However, a major drawback of occupancy grids is caused by their pure subsymbolic nature. For example, they provide no framework for representing symbolic entities-of-interest such as doors, desks, etc. They furthermore provide no means for treating static obstacles such as walls differently from dynamic obstacles such as doors. However, in order to operate autonomous mobile robots over long periods of time, it is essential to be able to deal with changes in the environment. For example, in populated environments, a robot must be able to detect blocked passages and to dynamically plan detours.

Therefore, maintaining world models in changing environments is another important aspect of map building. In this paper we introduce an approach that allows to apply occupancy grid maps for path planning in highly dynamic environments. Th key idea of our approach is to represent the world by a combination of different occupancy grid maps. One of these maps models only the static obstacles in the environment. This map is typically built beforehand, when the environment is empty. The other maps are updated on-the-fly as the robot moves and senses. The static and dynamic maps are constantly integrated to allow the robot to quickly react to blocked passages and to plan detours. The common reference frame needed to integrate the different maps is achieved by estimating the position of the robot relative to the static map. Robust position estimation in dynamic environments is attained by using a filter technique to detect sensor measurements corrupted by dynamic obstacles [Fox *et al.*, 1998; Burgard *et al.*, 1998].

The techniques described in this paper have been implemented and extensively tested. We used the mapping techniques to compute large-scale maps of $3000m^2$ containing large open spaces and long cycles. The map updating and sensor filtering techniques have been applied in two different long-term experiments in which the mobile robots Rhino and Minerva were applied for long periods of time as interactive museum tour-guides.

The remainder of this paper is organized as follows. Section 2 introduces our statistical approach to concurrent mapping and localization. Implementations of this general method and results are presented in Section 3. An application of occupancy grid maps in highly dynamic environments is given in Section 4, followed by a review of related work.

2 Concurrent Mapping and Localization

In this section, we propose an algorithm for solving the global alignment problem that occurs in map building with unbounded odometric error and perceptual ambiguity. The approach is an instance of a class of statistical estimation problems, where a robot seeks to find the most likely map from a set of observations and motion commands. This estimation problem is solved by a variant of the EM algorithm, which is an efficient hill-climbing method for maximum likelihood estimation in high-dimensional spaces. In the context of mapping, EM iterates two alternating steps: a *localization step*, in which the robot is localized using a previously computed map, and a *mapping step*, which computes the most likely map based on the previously pose estimates.

The statistical framework is general enough to deal with different types of sensors and for simplicity we will introduce the approach by assuming that the robot can observe landmarks. A different implementation of the approach based on sonar sensors will be discussed in Section 3.2.

2.1 Statistical Foundations

We pose the problem of mapping as a statistical *maximum likelihood estimation problem* [Thrun *et al.*, 1998b]. To generate a map, we assume that a robot is given a stream of data, denoted

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

where $o^{(t)}$ stands for an *observation* that the robot made at time t, and $u^{(t)}$ for an action command that the robot executed between time t to time $t+1$. T denotes the total number of time steps in the data. Without loss of generality, we assume that the data is an alternated sequence of actions and observations.

In statistical terms, the problem of mapping is the problem of finding the most likely map given the data. *Maps* will be denoted by $m = \{m_{x,y}\}_{x,y}$. A map is an assignment of "properties" $m_{x,y}$ to each x-y-location in the world. In topological approaches to mapping, the properties-of-interest are usually locations of landmarks [Chown *et al.*, 1995] or, alternatively, location of significant places [Kuipers and Byun, 1991; Choset, 1996]. Metric approaches, on the other hand, usually use the location of obstacles as properties-of-interest [Chatila and Laumond, 1985; Moravec, 1988; Lu and Milios, 1997a].

Our approach assumes that the robot is given two basic, probabilistic models, one that describes robot motion, and one that models robot perception.

- **The motion model**, denoted $P(\xi'|u,\xi)$, describes the probability that the robot's pose is ξ', if it previously executed action u at pose ξ. Here ξ is used to refer to a pose, that is the x-y-location of a robot together with its heading direction. Figure 1 illustrates the motion model, by showing the probability distribution for ξ' upon executing two different actions. Notice that in these and other diagrams, poses are projected into x-y-space (the heading direction is omitted).

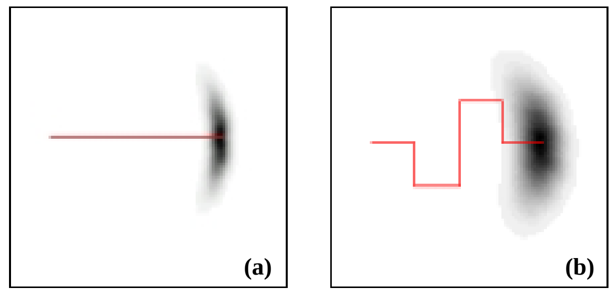


Fig. 1. Motion model. The grayly shaded area shows the pose distributions (projected into 2D) after the motion commands indicated by the solid lines.

- **The perceptual model**, denoted $P(o|m,\xi)$, models the likelihood of observing o in situations where both the world m and the robot's pose ξ are known.

  For low-dimensional sensors such as proximity sensors, perceptual models can readily be found in the literature [Burgard *et al.*, 1996; Moravec, 1988; Konolige, 1999]. Figure 2 illustrates a perceptual model for a robot that can detect landmarks and that can measure, with some uncertainty, their relative orientations and distances. Figure 2 (a) shows an example map m, in which the dark spots indicate the locations of two indistinguishable landmarks. Figure 2 (b) plots $P(o|m,\xi)$ for different poses ξ, for the specific observation that the robot observed a landmark ahead in five meters distance. The darker a pose, the more likely it is under this observation.

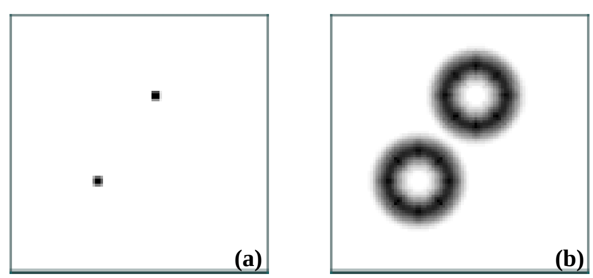


Fig. 2. Perceptual model. (a) Shows a map with two indistinguishable landmarks, and (b) displays the uncertainty after sensing a landmark in 5 meter distance.

These three quantities—the data d, the motion model $P(\xi'|u,\xi)$, and the perceptual model $P(o|m,\xi)$—form the statistical basis of our approach.

2.2 The Map Likelihood Function

In statistical terms, the problem of mapping is the problem of finding the most likely map given the data

$$m^* \quad = \quad \underset{m}{\operatorname{argmax}}\, P(m|d). \tag{2}$$

The probability $P(m|d)$ can be written as

$$P(m|d) \quad = \quad \int P(m|\xi, d)\; P(\xi|d)\; d\xi. \tag{3}$$

Here the variable ξ denotes the *set of all poses* at times $1, 2, \ldots, T$, that is, $\xi := \{\xi^{(1)}, \ldots, \xi^{(T)}\}$, where $\xi^{(t)}$ denotes the robot's pose at time t. By virtue of Bayes rule, the probability $P(m|\xi, d)$ on the right hand side of Equation (3) can be re-written as

$$P(m|\xi, d) \quad = \quad \frac{P(d|m, \xi)\; P(m|\xi)}{P(d|\xi)} \tag{4}$$

Based on the observation that $o^{(t)}$ at time t depends only on the map m and the corresponding location $\xi^{(t)}$, the first term on the right hand side of Equation (4) can be transformed into

$$P(d|m, \xi) \quad = \quad \prod_{t=1}^{T} P(o^{(t)}|m, \xi^{(t)}) \tag{5}$$

Furthermore, $P(m|\xi) = P(m)$ in Equation (4), since in the absence of any data, m does not depend on ξ. $P(m)$ is the Bayesian *prior* over all maps, which henceforth will be assumed to be uniformly distributed. Finally, the term $P(\xi|d)$ in Equation (3) can be re-written as

$$P(\xi|d) \quad = \quad \prod_{t=1}^{T-1} P(\xi^{(t+1)}|u^{(t)}, \xi^{(t)}) \tag{6}$$

The latter transformation is based on the observation that the robot's pose $\xi^{(t+1)}$ depends only on the robot's pose $\xi^{(t)}$ one time step earlier and the action $u^{(t)}$ executed there. Putting all this together leads to the likelihood function

$$P(m|d) \quad = \quad \int \frac{\prod_{t=1}^{T} P(o^{(t)}|m, \xi^{(t)})\; P(m)}{P(d|\xi)} \prod_{t=1}^{T-1} P(\xi^{(t+1)}|u^{(t)}, \xi^{(t)})\; d\xi. \tag{7}$$

Since we are only interested in maximizing $P(m|d)$, not in computing an exact value, we can safely drop the constants $P(m)$ and $P(d|\xi)$. The resulting expression,

$$\underset{m}{\operatorname{argmax}} \int \prod_{t=1}^{T} P(o^{(t)}|m, \xi^{(t)}) \prod_{t=1}^{T-1} P(\xi^{(t+1)}|u^{(t)}, \xi^{(t)})\; d\xi, \tag{8}$$

is a function of the data d, the perceptual model $P(o|m, \xi)$, and the motion model $P(\xi'|u, \xi)$. Maximizing this expression is equivalent to finding the most likely map.

2.3 Efficient Estimation

Unfortunately, computing (8) is computationally challenging. This is because finding the most likely map involves search in the space of all maps. For large-scale environments, this space often has 10^6 dimensions or more if crude approximations are used. To make matters worse, the evaluation of a single map would require integrating over all possible poses at all points in time, which for the datasets considered in this paper would require integration over more than 10^5 independent pose variables, each of which can take 10^8 values or so.

Fortunately, there exists an efficient and well-understood technique for hill-climbing in likelihood space: the *EM algorithm* [Dempster *et al.*, 1977], which in the context of Hidden Markov Models is often referred to as *Baum-Welch* or *alpha-beta algorithm* [Rabiner, 1989]. EM is a hill-climbing routine in likelihood space, which alternates two steps, an *expectation step* (E-step) and a *maximization step* (M-step). In the context of robot mapping, these steps correspond roughly to a localization step and a mapping step (see also [Koenig and Simmons, 1996; Shatkey and Kaelbling, 1997]):

1. In the E-step, the robot computes probabilities $P(\xi|m, d)$ for the robot's poses ξ at the various points in times, based on the currently best available map m (in the first iteration, there will be no map).
2. In the M-step, the robot determines the most likely map by maximizing $\operatorname{argmax}_m P(m|\xi, d)$, using the location estimates computed in the E-step.

The E-step corresponds to 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. Our approach is, thus, a hill-climbing procedure that does not provide a guarantee of global optimality; given the complexity of the problem, however, it is unclear whether a globally optimal routine exists that is computationally tractable.

The E-Step

The E-step uses the current-best map m along with the data to compute probabilities $P(\xi^{(t)}|d, m)$ for the robot's poses at times $t = 1, \ldots, T$. With appropriate assumptions, $P(\xi^{(t)}|d, m)$ can be expressed as the normalized product of two terms

$$P(\xi^{(t)}|d, m) \quad = \quad \eta\, \underbrace{P(\xi^{(t)}|o^{(1)}, \ldots, o^{(t)}, m)}_{:=\alpha^{(t)}}\; \underbrace{P(\xi^{(t)}|u^{(t+1)}, \ldots, o^{(T)}, m)}_{:=\beta^{(t)}} \tag{9}$$

Here η is a normalizers that ensure that the left-hand side of Equation (9) sums up to one (see [Thrun *et al.*, 1998b] for a mathematical derivation). Both terms, $\alpha^{(t)}$ and $\beta^{(t)}$, as defined in (9), are computed separately, where the former is computed forward in time and the latter is computed backwards in time. Notice that $\alpha^{(t)}$ and $\beta^{(t)}$ are analogous to those in the alpha-beta algorithm [Rabiner, 1989].

The computation of the α-values is a version of *Markov localization*, which has recently been used with great success for robot localization in *known* environments by various researchers [Burgard *et al.*, 1996; Kaelbling *et al.*, 1996; Koenig and Simmons, 1996; Nourbakhsh *et al.*, 1995; Simmons and Koenig, 1995]. The β-values add additional knowledge to the robot's pose, 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.

Computing the α-Values: Initially, the robot is assumed to be at the center of the global reference frame and $\alpha^{(1)}$ is given by a Dirac distribution centered at $(0,0,0)$:

$$\alpha^{(1)} \;=\; P(\xi^{(1)}|o^{(1)},m) \;=\; \begin{cases} 1, \text{ if } \xi^{(1)} = (0,0,0) \\ 0, \text{ if } \xi^{(1)} \neq (0,0,0) \end{cases} \qquad (10)$$

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

$$\alpha^{(t)} \;=\; \eta\, P(o^{(t)}|\xi^{(t)},m)\; P(\xi^{(t)}|o^{(1)},\ldots,u^{(t-1)},m) \qquad (11)$$

where η is again a probabilistic normalizer. The rightmost term of (11) can be transformed to

$$\begin{aligned} &P(\xi^{(t)}|o^{(1)},\ldots,u^{(t-1)},m) \\ &\quad = \int P(\xi^{(t)}|u^{(t-1)},\xi^{(t-1)})\; \alpha^{(t-1)}\; d\xi^{(t-1)} \end{aligned} \qquad (12)$$

Substituting (12) into (11) yields a recursive rule for the computation of all $\alpha^{(t)}$ with boundary condition (10). See [Thrun *et al.*, 1998b] for a more detailed derivation.

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

$$\begin{aligned} \beta^{(t)} \;=\; \eta \int & P(\xi^{(t+1)}|u^{(t)},\xi^{(t)}) \\ & P(o^{(t+1)}|\xi^{(t+1)},m)\; \beta^{(t+1)}\; d\xi^{(t+1)} \end{aligned} \qquad (13)$$

The derivation of the equations are analogous to that of the computation rule for α-values and can be found in [Thrun *et al.*, 1998b]. The result of the E-step, the products $\alpha^{(t)}\beta^{(t)}$, are estimates of the robot's locations at the various points in time t.

The M-Step

The M-step maximizes $P(m|\xi,d)$, that is, in the M-step the robot computes the most likely map based on the pose probabilities computed in the E-step. Generating maps with *known* robot poses, which is basically what the M-step amounts to, has been studied extensively in the literature on mobile robot mapping (see e.g., [Borenstein and Koren, 1991; Elfes, 1989; Moravec, 1988]).

By applying Bayes rule and with the appropriate assumptions, the estimation problem can be temporally decomposed into

$$P(m|\xi,d) \;=\; \alpha \prod_{t=1}^{T} P(o^{(t)}|\xi^{(t)},m) \qquad (14)$$

where α is a normalizer that can safely be ignored in the maximization. It is common practice to decompose the problem spatially, by solving the optimization problem independently for different x-y-locations:

$$\operatorname*{argmax}_m P(m|\xi,d) = \left\{ \operatorname*{argmax}_{m_{x,y}} \prod_{t=1}^{T} P(o^{(t)}|\xi^{(t)},m_{x,y}) \right\}_{x,y} \qquad (15)$$

While technically speaking, this independence assumption is not warranted for sensors that cover many x-y-locations (such as sonar sensors), it is typically made in the literature to make the estimation problem tractable. The resulting local maximum likelihood estimation problems are highly tractable, since each of them involves only a single, discrete random variable.

3 Generating Metric Maps

A key feature of our statistical approach is that it can equally be applied to both topological and metric maps. In this section we will show how to generate metric maps using this technique. This can be done in two different ways: First, using the landmark-based approach presented in the previous section to solve the *global alignment problem*, thus generating a topologically correct path of the robot. Such a corrected path can then be used to build a detailed metric map using the proximity data collected during map building (see Section 3.1). The disadvantage of this technique is that is relies on the detection of landmarks. Another method is to use the same statistical framework to directly generate metric maps without the need for landmark detections. An approach to solving this more challenging problem will be proposed in Section 3.2.

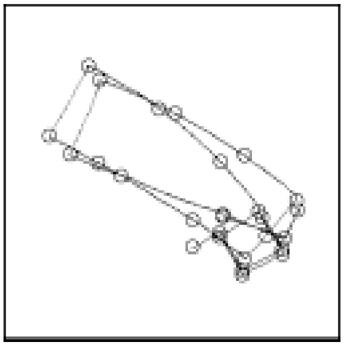
3.1 Landmark-based Mapping

This mapping algorithm first constructs coarse-grained topological maps, based on which it then builds detailed metric maps. Both of the mapping steps are specialized versions of the same statistical approach described above.

Generating Topologically Correct Maps

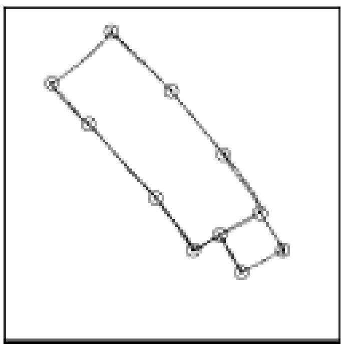
Following [Kuipers and Byun, 1991; Choset, 1996; Matarić, 1990; Shatkey and Kaelbling, 1997], the topological component of our algorithm seeks to determine the location of *significant places* in the environment, along with an order in which these places were visited by the robot.

In the topological mapping step, the robot can only observe whether or not it is at a *significant place*. Our definition of significant places follows closely the notion of "distinctive places" in Kuipers's Spatial Semantic Hierarchy [Kuipers and Byun, 1991], and the notion of "meetpoints" in Choset's Generalized Voronoi Graphs [Choset, 1996]. In our experiments, we simulated these methods by manually pressing a button whenever the robot crossed a significant place. To test the most general (and difficult) case, our approach assumes that the significant places are *indistinguishable*. Thus, the robot observes only a single bit of information, namely whether or not it is at a significant place. This deviates from Kuipers and Byun's work, in which places are assumed to be locally


(a)


(b)

Fig. 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 using the raw odometry data.


(a)

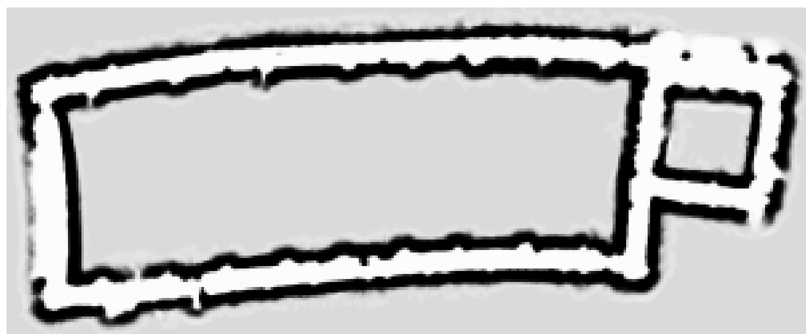

(b)

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

distinctive. Nodes in the topological map correspond to *significant places*. Arcs between nodes are created if nodes are adjacent in the data set d. The robot is not told how many significant places exist in its environment; neither does it know whether or not it visited a significant place more than once. Instead, it guesses the number of nodes as a side-effect of maximizing the map likelihood function.

The topological mapper represents all probabilities (poses, maps, ...) with evenly-spaced, piecewise constant functions (also called: grids). In all our experimental results, the spatial resolution was 1 meter and the angular resolution was 5°.

Figures 3 (a) and 4 (a) illustrate the capabilities of this approach. Figure 3 (a) shows a dataset collected in our university building, in which circles indicate significant place observations. Here the final odometric error is approximately 24.9 meter. What makes this dataset challenging is the large circular hallway (60 by 25 meter). When traversing the circle for the first time, the robot cannot exploit landmarks to improve its pose estimates; thus, it accumulates odometric error. Since significant places 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 can it use its perceptual clues to estimate where it is (and was), in order to build a consistent map. Figure 4 (a) shows the result of topological mapping, including the "corrected" path taken by the robot.

Building Metric Maps

The result of the topological estimation routine can be used to build more accurate occupancy grid maps [Elfes, 1989; Moravec, 1988]. Figure 4 (b) 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 [Thrun, 1998]. Here, sonar measurements are converted into local occupancy grid maps using neural networks (see Figure 6). These maps are integrated using Bayesian integration rules.

To see the benefit of using the path corrected by the topological mapper, Figure 3 (b) shows a map using the raw, uncorrected data. The map constructed from raw data is unusable for navigation, whereas the corrected map is sufficient for mobile robot navigation (see [Burgard *et al.*, 1996; Thrun *et al.*, 1998a] for a description of our navigation routines).

When using data collected by laser range-finders, even more accurate maps can be built. This metric mapper, which is based on the same statistical framework as the topological mapper, is a modified version of an approach previously proposed in [Gutmann and Schlegel, 1996; Lu and Milios,

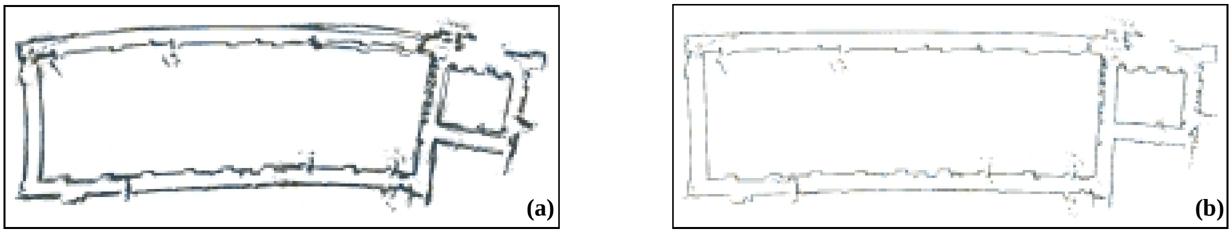


Fig. 5. (a) The metric map, generated using laser measurements and the pose estimates derived in the topological mapping phase, is still imperfect, although the errors are small. (b) The metric mapper finally generates a highly detailed map.

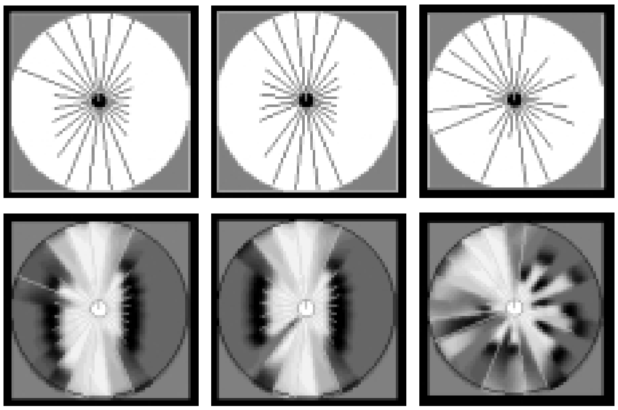


Figure 6: The top row shows raw sensor data, the bottom row shows a likelihood field (local map): the brighter a pixel, the higher its likelihood for being unoccupied. This perceptual model has been learned from hand-labeled data, using artificial neural networks.

1997a].

The perceptual model of the laser-based mapper is defined through a geometric map matching method, which determines the likelihood of laser scans based on the proximity of perceived obstacles in x-y-coordinates. The metric mapper represents all densities (poses, maps, motion model, and perceptual model) by Gaussian distributions (Kalman filters). Gaussians have a dual advantage: First they permit determining robot poses and location of obstacles with floating-point resolution, yielding high-resolution maps. Second, they make it possible to apply highly efficient linear programming methods when maximizing the likelihood function [Lu and Milios, 1997a]. However, Gaussians are uni-modal; Thus, the metric mapper cannot represent two distinct hypotheses, as can the topological mapper. As a direct consequence, the metric mapper can only be applied if the initial odometric error is small (e.g., smaller than 2 meters), so that the correct solution lies in the vicinity of the initial guess. Fortunately, the topological mapper, if successful, generates maps that meet this criterion.

Technically, the metric mapper builds a *network of spatial relations* among all poses where range scans have been taken. Spatial probabilistic constraints between poses are derived from matching pairs of range scans and from odometry measurements. In the E-step, the metric mapper estimates all poses. In the M-step, it re-maps the scans based on the previously estimated poses. Both steps are iterated. In our experiments, we found that the metric mapper consistently converged to the limit of machine accuracy after four or five iterations of EM.

The metric mapper is computationally highly efficient. As noticed above, our approach employs linear programming for efficient likelihood maximization. To do so, it approximates the likelihood function linearly, which leads to a closed form solution for all pose variables. Figure 5 (a) shows a map based on laser measurements and the positions estimated by the topological approach (c.f. 4 (a)). As can be seen there, the position estimates are only approximately correct. Figure 5 (b) shows the map subsequently generated by the laser-based metric mapper.

3.2 Sonar-based Mapping

In the previous section we described how the EM method can be used to compute a metric map of the environment by first computing a topological map. The disadvantage of this approach is that it requires a human operator or special detectors to identify significant places. Fortunately, this drawback can be eliminated by introducing a two-layered representation in which a global occupancy grid map is built from a set of local maps generated from short sequences of sensor data [Burgard *et al.*, 1999], a technique also used in [Yamauchi, 1996; Schiele and Crowley, 1994]. Examples of such local maps are shown in Figure 7. For such maps, which replace the landmarks used above, we assume that the odometric error can be neglected since the robot only moves a short distance. In the E-step, the localization is not carried out with respect to a global map but rather with respect to the different local maps (see [Burgard *et al.*, 1999] for details).

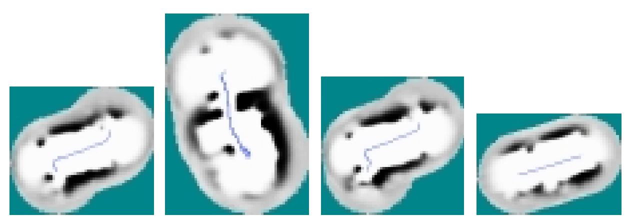
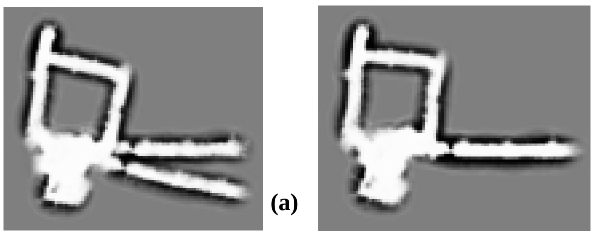


Fig. 7: Examples of local maps, annotated by robot trajectories. These maps have been constructed from sonar measurements.

This approach has two advantages. First, the local maps integrate data over time and thus allow to deal with noisy range sensors such as ultra-sound sensors. Thus, this technique does not rely on expensive and highly accurate sensors like laser range-finders as used in [Gutmann and Schlegel, 1996; Lu and Milios, 1997a]. Second, the use of static local grid maps maintains dependencies between individual grid cells.

This for example allows the approach to accurately represent the width of a corridor or the structure of rooms.

Figure 8 illustrates the capabilities of this approach. The experiment was carried out with the robot Amelia in the Wean Hall of the Carnegie Mellon University. Figure 8 (a) shows the map constructed from ultrasound data using the raw odometry measurements. Figure 8 (b) shows the result of applying the EM method to our layered map representation. As can be seen from the figure, the odometry errors are corrected and the corridor in the lower right corner is now mapped correctly.



(a) (b)

Fig. 8. Maps built in Wean Hall of Carnegie Mellon University, (a) using raw odometry, and (b) using our new algorithm. These maps are comparable to those generated by our previous EM method, but without reliance on manually labeled reference positions.

4 Dealing with Dynamic Environments

In the previous sections we showed how to build metric maps from sensor data. To keep the mapping problem tractable, our approach – as well as virtually all existing approaches to concurrent mapping and localization – makes the assumption that the environment is *static*. While this assumption might be reasonable during the phase of initial map acquisition, it certainly does not hold for robots that operate over long periods of time in populated environments. Especially for path planning in changing environments, a robot's ability to modify its map and hence its paths on-the-fly is essential for efficient navigation.

Consider, for example, our mobile robot RHINO in the "Deutsches Museum Bonn" (German Museum Bonn), where it served the function of an interactive robotic tour-guide (cf. 10). The robot's task was to engage visitors and guide them through the exhibition, providing verbal explanations for the various exhibits. Here, updating the world model is essential, since entire passages can be blocked by visitors making it necessary to plan detours. Please note that office delivery robots face similar situations, e.g. when doors are closed.

Fig. 10. Rhino as it gives a tour through the "Deutsches Museum Bonn".

One way to deal with such dynamic environments would be to constantly update the map of the environment using an approach for concurrent mapping and localization. Unfortunately, this solution is intractable in practice since it entails concurrent estimation of the position of the robot, the positions of all static obstacles, and the positions (and motion vectors!) of all non-static obstacles. Furthermore, occupancy grid maps are not designed to distinguish between static and dynamic obstacles. Therefore, whenever new sensor information arrives, grid cells containing walls are updated the same way as grid cells containing people passing by.

To deal with these problems, we decouple the task of learning a model of the environment from the task of updating it according to changes of the environment. Our approach maintains different maps of the environment. One map only contains *static* obstacles such as walls. This map can either be built by a method discussed in the previous sections or it can be a hand-crafted CAD map (as used during this museum tour-guide project). In addition to this map, occupancy grids are built on-the-fly to model the dynamic obstacles in the environment (note that these maps can be adapted much faster than the static map). All grid maps are combined to get complete information about the obstacles in the environment. Figure 9 illustrates this integration of maps built from different sensor modalities (see [Burgard *et al.*, 1998] for more details).

In order to integrate the different occupancy grids into a single map, the position of the robot is estimated with respect to the global coordinate frame of the static map. To allow reliable position estimation in dynamic environments using only a static world model, we extended our implementation of Markov localization by filter techniques to detect measurements corrupted by non-modeled, i.e. dynamic, obstacles. The resulting technique has been shown to be able to robustly estimate the position of mobile robots over long periods of time even in densely crowded environments such as museums and office environments [Fox *et al.*, 1998; Fox, 1998; Burgard *et al.*, 1998; Thrun *et al.*, 1999].

To allow the robot to react to unforeseen blockage of passages, the path planner consults the integrated map to determine shortest paths to arbitrary target points (see [Thrun, 1998] for a detailed description). Since this map is updated continuously, the motion planner continuously revises its plans. Figure 11 shows a typical output of the path planner. Here, the target location is in the lower left corner of the map and the gray shading indicates the distance of each location to this target point under consideration of the occupancy probability of grid cells.

Figure 12 illustrates the benefit of our approach. It shows an integrated map recorded during peak traffic hours in the museum. In this case, a massive congestion made it impossible for the robot to progress along the original path. Due to the map integration, the blocked path is detected and the robot is able to chose a detour. A further advantage of our distributed map representation is the ability to adapt the maps

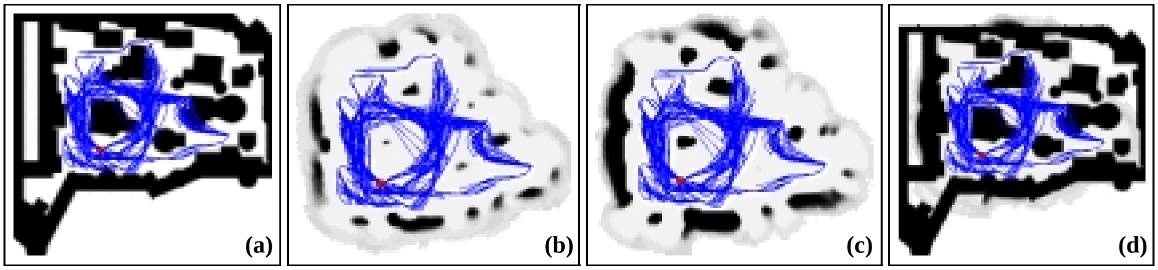


Fig. 9. Integrating multiple maps: (a) CAD map of the museum ($21 \times 20m^2$) modeling only the static obstacles, (b) laser map, (c) sonar map, and (d) the integrated map used for path planning.

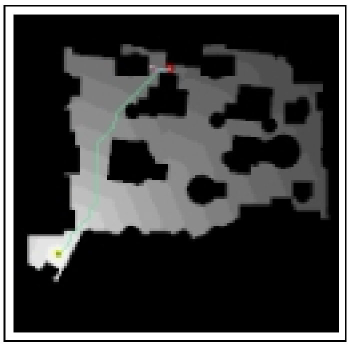


Fig. 11. The motion planner uses dynamic programming to compute the shortest path to the nearest goal(s) for every location in the unoccupied space, as indicated by the gray shading. Once the distance has been computed, paths are generated by hill-climbing in distance space.

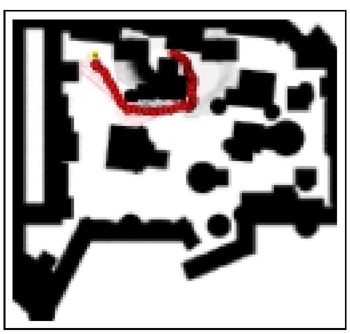


Fig. 12. An integrated map, acquired in a situation where a massive congestion of the museum forced the robot to take a detour.

at different time scales. Once the robot reached a target location, all obstacles were deleted from the dynamic maps and path planning to the next exhibit was based only on the static map shown in Figure 9 (a). However, since the dynamic maps are updated as the robot moves, this plan can be adapted to unforeseen situations.

5 Related Work

Over the last decade, there has been a flurry of work on metric map building for mobile robots. Probably the most successful metric approach to date are occupancy grids, which were originally proposed in [Elfes, 1989; Moravec, 1988; Borenstein and Koren, 1991] and which since have been employed successfully in numerous systems. Other metric approaches, such as those described in [Chatila and Laumond, 1985; Cox and Leonard, 1994; Lu and Milios, 1997a], describe the environment by geometric atoms such as straight lines (walls) or points (range scans).

Approaches that fall strictly into the topological paradigm can be found in [Chown *et al.*, 1995; Kortenkamp and Weymouth, 1994; Kuipers and Byun, 1991; Matarić, 1990; Shatkey and Kaelbling, 1997]. Some of these approaches do not annotate topological maps with metric information at all; instead, they rely on procedural knowledge for moving from one topological entity to another.

Our approach to concurrent map building and localization differs from most work in the field (see also the surveys in [Thrun, 1998; Lu and Milios, 1997a]) in three important technical aspects. First, robot poses are revised forward *and* backwards in time—as pointed out by [Lu and Milios, 1997a], most existing approaches do not revise pose estimations backwards in time. Second, it does not rely on highly accurate proximity sensors such as laser range-finders. Third, by using probabilistic representations, the approach considers multiple hypotheses as to where a robot might have been, which facilitates the recovery from errors.

The approaches in [Lu and Milios, 1997a; Cox and Leonard, 1994; Shatkey and Kaelbling, 1997] are similar in this respect to the one proposed here. [Shatkey and Kaelbling, 1997] proposed to use the alpha-beta algorithm for learning topological maps, based on [Koenig and Simmons, 1996], who used the same algorithm for a restricted version of the mapping problem. [Cox and Leonard, 1994] introduced a mapping algorithm that estimates the positions of geometric features. Here, Bayesian methods and Kalman filters are applied to deal with sensor uncertainty. Their approach keeps track of possible feature locations by using multiple hypothesis trees. To avoid exponential growth of the number of hypotheses, these trees are pruned after a short period of time, which limits the ability to revise estimations backwards in time. The approach by Lu/Milios matches laser scans into partially built maps, using Kalman filters for positioning. Together with [Gutmann and Schlegel, 1996], they demonstrated the appropriateness of this algorithm for mapping environments with cycles. Their approach, however, is incapable of representing ambiguities and multi-modal densities. It can only compensate a limited amount of odometric error in x-y-space, due to the requirement of a “sufficient overlap between scans” [Lu and Milios, 1997a]. In all cases studied in [Gutmann and Schlegel, 1996; Lu and Milios, 1997a; 1997b], the odometric error was an order of magnitude smaller than the one reported here. In addition, the approach is largely specific to robots equipped with laser range finders. It is unclear if the approach can cope with less accurate sensors such as sonars.

6 Conclusion

This paper presented a general probabilistic approach to concurrent mapping and localization. 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 that appropriateness of the approach to topological and metric map building.

We furthermore showed how to apply occupancy grid maps to path planning in highly dynamic environments. The key idea of this approach is to treat static and dynamic obstacles differently by representing them in different maps. Thereby we decouple the problems of estimating the position of the robot and estimating the positions of static/dynamic obstacles in the environment. This approach resulted in robust navigation of a mobile robot in a crowded museum over extended periods of time.

References

[Borenstein and Koren, 1991] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.

[Burgard *et al.*, 1996] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1996.

[Burgard *et al.*, 1998] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

[Burgard *et al.*, 1999] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping of large-scale mobile robot environments using EM. In *Proc. of the International Conference on Machine Learning (ICML)*, 1999.

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

[Choset, 1996] H. Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, Caltech, 1996.

[Chown *et al.*, 1995] E. Chown, S. Kaplan, and D. Kortenkamp. Prototypes, location, and associative networks (plan): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19:1–51, 1995.

[Cox and Leonard, 1994] I.J. Cox and J.J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66:311–344, 1994.

[Dempster *et al.*, 1977] Dempster, Laird, and Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[Elfes, 1989] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, CMU, 1989.

[Fox *et al.*, 1998] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

[Fox, 1998] D. Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Naviagation*. PhD thesis, Dept. of Computer Science, University of Bonn, Germany, December 1998.

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

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

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

[Konolige, 1999] K. Konolige. Markov localization using correlation. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[Kortenkamp and Weymouth, 1994] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1994.

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

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

[Lu and Milios, 1997b] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18, 1997.

[Matarić, 1990] 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.

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

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

[Rabiner, 1989] 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.

[Schiele and Crowley, 1994] B. Schiele and J.L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1994.

[Shatkey and Kaelbling, 1997] H. Shatkey and L.P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

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

[Thrun *et al.*, 1998a] 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 David Kortenkamp, R.P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, Cambridge, MA, 1998.

[Thrun *et al.*, 1998b] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. Also appeard in *Autonomous Robots* 5, pp. 253–271, joint issue.

[Thrun *et al.*, 1998c] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

[Thrun *et al.*, 1999] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second generation mobile tour-guide robot. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

[Thrun, 1998] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):27–71, 1998.

[Yamauchi, 1996] B. Yamauchi. Mobile robot localization in dynamic environments using dead reckoning and evidence grids. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1996.