A Probabilistic Approach for Concurrent Map Acquisition and Localization for Mobile Robots

Sebastian Thrun Wolfram Burgard[†] Dieter Fox[†]
October 1997
CMU-CS-97-183

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper addresses the problem of building large-scale geometric maps of indoor environments with mobile robots. It poses the map building problem as a constrained, probabilistic maximum-likelihood estimation problem. It then devises a practical algorithm for generating the most likely map from data, along with the most likely path taken by the robot. Experimental results in cyclic environments of size up to 80 by 25 meter illustrate the appropriateness of the approach.

[†] These authors are affiliated with the Institut für Informatik III, Universät Bonn, Germany
The research was financially supported by Daimler Benz Research, Germany.

Keywords: autonomous robots, Baum-Welch, mobile robots, navigation, localization, mapping, positioning, probabilistic algorithms, robot mapping

1 Introduction

Over the last two decades or so, 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), often 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 relative to past locations. 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 general 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.

In our experiments, we investigates a restricted version of the map building problem, in which a human operator tele-operates the robot through its environment. In particular, we assume that the operator selects a small number of *significant places* (such as intersections, corners, dead ends), where he pushes (with high likelihood) a button to inform the robot that such a place has been reached. The approach, however, can be applied to the problem of landmark-based map acquisition (using one of the many landmark recognition routines published in the literature). Thus, the paper phrases the approach in the language commonly used in the field of landmark-based navigation. The general problem addressed in this paper is: How can a robot construct a consistent map of an environment, if it occasionally observes a landmark? In particular, the paper addresses situations where landmarks might be entirely indistinguishable, and where the accumulated odometric error might be enormous.

The paper presents an algorithm for landmark-based map acquisition and concurrent localization that is based on a rigorous statistical account on robot motion and perception. In it the problem of map building is posed 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 [12, 24], the high-dimensional maximum likelihood estimation problem is solved efficiently using the Baum-Welch (or alpha-beta) algorithm [22]. Baum-Welch alternates an "expectation step" (E-step) and a "maximization step" (M-step, sometimes also called "modification 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, which leads to a local maximum in likelihood space. 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 surveys results obtained with a RWI B21 robot in indoor environments of size 80 by 25 meter. 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 with a conventional method for building occupancy grid maps [27], for which results are reported as well. Related work is reviewed in Section 8.

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

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

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

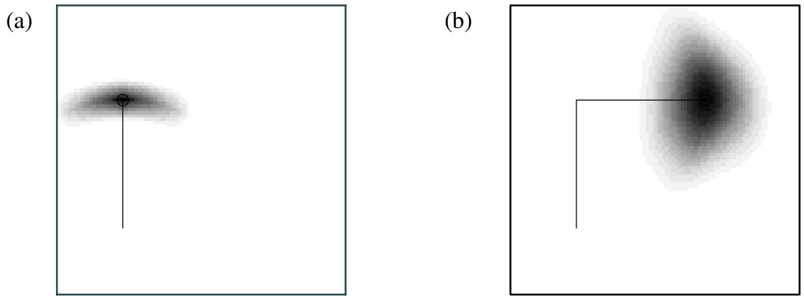

Figure 1: Probabilistic model of robot motion: Accumulated uncertainty after moving as shown: (a) 40 meter, (b) 80 meter.

paper) show only 2D projections of $P(\xi)$, as $P(\xi)$ is three-dimensional. Note that the particular shape of the distributions results from accumulated translational and rotational error as the robot moves.

Mathematically speaking, the exact motion model assumes that the robot accumulates both translational and rotational error as it moves. Both sources of error are distributed according to a triangular distribution, that is centered on the zero-error outcome.[1] The width of these distributions are proportional to the length of the motion command. Of course, the resulting distribution in x-y-θ space is not triangularly distributed, as the curvature in Figure 1a indicates.

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" [6] (see also Kuipers's and Matarić's work [13, 18]), which correspond to intersections or dead ends in corridors and which can be detected very robustly. Various other choices are described in [1].

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

[1] The density function of a triangular distribution centered on μ and with width σ is given by $f(x) = \max\{0, \sigma^{-1} - \sigma^{-2}|x - \mu|\}$.

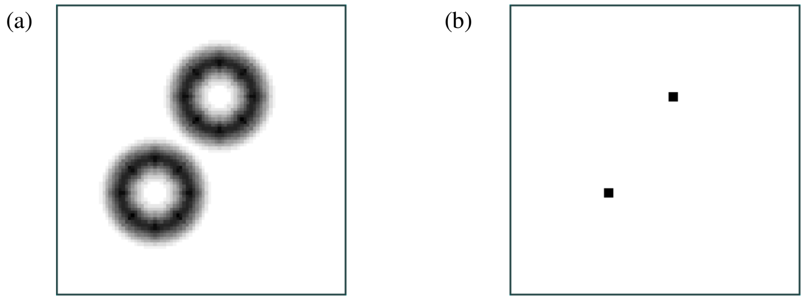

Figure 2: Probabilistic model of robot perception: (a) uncertainty after sensing a landmark in 5 meter distance, (b) the corresponding map.

modeled by a conditional probability:

$$P(o|\xi, m). \tag{3}$$

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

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

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. If the landmark were distinguishable, the resulting density (Figure 2a) would consist of a single circle, instead of two.

Notice that although the perceptual model $P(o|\xi, m)$ assumes exact knowledge of both the robot's location and its environment (which makes it easy to derive), the estimation according to equation (4) does not assume knowledge of ξ. It only assumes knowledge of the map m.

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

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

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. We believe that 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 $t = 1, \ldots, T$. With appropriate assumptions, $P(\xi^{(t)}|d, m)$ can be expressed as the normalized product of two terms

$$\begin{aligned} & P(\xi^{(t)}|d, m) \\ & \quad = \; P(\xi^{(t)}|o^{(1)}, \ldots, o^{(t)}, u^{(t+1)}, \ldots, o^{(T)}, m) \end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{=} \eta_1\, P(o^{(1)},\ldots,o^{(t)}|\xi^{(t)},u^{(t+1)},\ldots,o^{(T)},m)\; P(\xi^{(t)}|u^{(t+1)},\ldots,o^{(T)},m) \\
&\overset{(b)}{=} \eta_1\, P(o^{(1)},\ldots,o^{(t)}|\xi^{(t)},m)\; P(\xi^{(t)}|u^{(t+1)},\ldots,o^{(T)},m) \\
&\overset{(c)}{=} \eta_2\, P(\xi^{(t)}|o^{(1)},\ldots,o^{(t)},m)\; P(o^{(1)},\ldots,o^{(t)}|m)\; P(\xi^{(t)}|u^{(t+1)},\ldots,o^{(T)},m) \\
&= \eta_3\, \underbrace{P(\xi^{(t)}|o^{(1)},\ldots,o^{(t)},m)}_{:=\alpha_\xi^{(t)}}\; \underbrace{P(\xi^{(t)}|u^{(t+1)},\ldots,o^{(T)},m)}_{:=\beta_\xi^{(t)}} \qquad (6)
\end{aligned}
$$

Here η_1, η_2, η_3 are normalizers that ensure that the left-hand side of Equation (6) sums up to one. The derivation of (6) follows from (a) the application of Bayes rule, (b) a commonly-used Markov assumption that specifies the conditional independence of future from past data given knowledge of the current location and the map, and (c) a second application of Bayes rule under the assumption in the absence of data, robot positions are equally likely.

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 our definition of $\alpha_\xi^{(t)}$ and $\beta_\xi^{(t)}$ deviates from the definition usually given in the literature on Hidden Markov Models (c.f., [22]). However, our definition maps nicely into existing localization paradigms. The computation of the α-values is a version of *Markov localization*, which has recently been used with great success by various researchers [3, 10, 12, 20, 25, 28]. 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.

3.1.1 Computation of the 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 $(0,0,0)$:

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

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

$$
\begin{aligned}
\alpha_\xi^{(t)} &= P(\xi^{(t)}|o^{(1)},\ldots,o^{(t)},m) \\
&= \eta\, P(o^{(t)}|\xi^{(t)},o^{(1)},\ldots,u^{(t-1)},m)\; P(\xi^{(t)}|o^{(1)},\ldots,u^{(t-1)},m) \\
&= \eta\, P(o^{(t)}|\xi^{(t)},m)\; P(\xi^{(t)}|o^{(1)},\ldots,u^{(t-1)},m) \qquad (8)
\end{aligned}
$$

where η is again a probabilistic normalizer, and the rightmost term of (8) can be transformed to

$$
\begin{aligned}
&P(\xi^{(t)}|o^{(1)},\ldots,u^{(t-1)},m)\\
&\quad = \int P(\xi^{(t)}|u^{(t-1)},\xi^{(t-1)})\ P(\xi^{(t-1)}|o^{(1)},\ldots,o^{(t-1)},m)\ d\xi^{(t-1)}\\
&\quad = \int P(\xi^{(t)}|u^{(t-1)},\xi^{(t-1)})\ \alpha_\xi^{(t-1)}\ d\xi^{(t-1)} \qquad (9)
\end{aligned}
$$

Substituting (9) into (8) yields a recursive rule for the computation of all $\alpha_\xi^{(t)}$ with boundary condition (7), which uses the data d, the model m, in conjunction with the motion model $P(\xi'|u,\xi)$ and the perceptual model $P(\xi|o,m)$. See [26] for a more detailed derivation.

3.1.2 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:

$$
\begin{aligned}
\beta_\xi^{(t)} &= P(\xi^{(t)}|u^{(t)},\ldots,o^{(T)},m)\\
&= \int P(\xi^{(t)}|u^{(t)},\xi^{(t+1)})\ P(\xi^{(t+1)}|o^{(t+1)},\ldots,o^{(T)},m)\ d\xi^{(t+1)}\\
&= \int P(\xi^{(t+1)}|u^{(t)},\xi^{(t)})\\
&\quad P(\xi^{(t+1)}|o^{(t+1)},\ldots,o^{(T)},m)\ d\xi^{(t+1)} \qquad (10)
\end{aligned}
$$

The rightmost expression is further transformed to:

$$
\begin{aligned}
&P(\xi^{(t+1)}|o^{(t+1)},\ldots,o^{(T)},m)\\
&\quad = \eta\ P(o^{(t+1)}|\xi^{(t+1)},u^{(t+1)},\ldots,o^{(T)},m)\ P(\xi^{(t+1)}|u^{(t+1)},\ldots,o^{(T)},m)\\
&\quad = \eta\ P(o^{(t+1)}|\xi^{(t+1)},m)\ \beta_\xi^{(t+1)} \qquad (11)
\end{aligned}
$$

The derivation of the equations are analogous to that of the computation rule for 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.

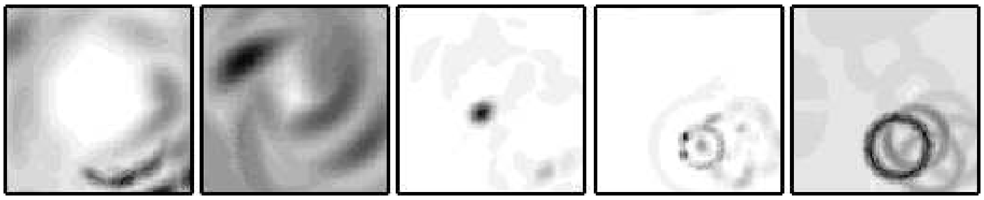

Figure 3: A typical sequence of β-tables (projected into 2D), taken from one of the experiments reported in this paper, illustrates the complex nature of the distributions involved.

Figure 3 shows a sequence of β-values, which arose in one of the experiments described below. This figure illustrates that some of the distributions are multimodal; Kalman filters, which are frequently used in localization, are probably not appropriate here.

In the first computation of the E-step, where no map m is available, $P(o|\xi, m)$ is assumed to be distributed uniformly. This is equivalent to ignoring all observations $\{o^{(1)}, o^{(2)}, \dots o^{(T)}\}$ in the computation of the α- and β-values. The resulting position estimates are only based on the controls $\{u^{(1)}, u^{(2)}, \dots u^{(T-1)}\}$ in d.

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 $l_1, \dots, l_n$. The set $L = \{l_1, \dots, l_n, l_*\}$ 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 [22], the maximum likelihood map under fixed position estimates is computed according to the weighted likelihood ratio

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

which is obtained by

$$
\begin{aligned}
P(m_{xy} = l|d) &= \frac{\sum_{t=1}^{T} \int P(m_{xy} = l|o^{(t)}, \xi^{(t)})\ \alpha_\xi^{(t)} \beta_\xi^{(t)}\ d\xi^{(t)}}{\sum_{t=1}^{T} \sum_{l' \in L} \int P(m_{xy} = l'|o^{(t)}, \xi^{(t)})\ \alpha_\xi^{(t)} \beta_\xi^{(t)}\ d\xi^{(t)}} \\
&= \frac{\sum_{t=1}^{T} \int P(m_{xy} = l|o^{(t)}, \xi^{(t)})\ \alpha_\xi^{(t)} \beta_\xi^{(t)}\ d\xi^{(t)}}{\sum_{t=1}^{T} \int \alpha_\xi^{(t)} \beta_\xi^{(t)}\ d\xi^{(t)}} \\
&= \frac{1}{T} \sum_{t=1}^{T} \int P(m_{xy} = l|o^{(t)}, \xi^{(t)})\ \alpha_\xi^{(t)} \beta_\xi^{(t)}\ d\xi^{(t)} \qquad (12)
\end{aligned}
$$

where

$$
P(m_{xy} = l|o^{(t)}, \xi^{(t)}) = \frac{P(o^{(t)}|m_{xy} = l, \xi^{(t)})\ P(m_{xy} = l|\xi^{(t)})}{\sum_{l' \in L} P(o^{(t)}|m_{xy} = l', \xi^{(t)})\ P(m_{xy} = l'|\xi^{(t)})} \qquad (13)
$$

Since we assume that m_{xy} does not depend on the robot's position ξ (and hence in the absence of data: $P(m_{xy} = l|\xi) = P(m_{xy} = l'|\xi)\ \forall l, l' \in L$), expression (13) can be simplified to

$$
\begin{aligned}
P(m_{xy} = l|o^{(t)}, \xi^{(t)}) &= \frac{P(o^{(t)}|m_{xy} = l, \xi^{(t)})}{\sum_{l' \in L} P(o^{(t)}|m_{xy} = l', \xi^{(t)})} \\
&= \eta\ P(o^{(t)}|m_{xy} = l, \xi^{(t)}) \qquad (14)
\end{aligned}
$$

Here η is the usual normalizer. While these equations look complex, they basically amount to a frequentist maximum-likelihood estimation (also called: counting). Equation (12) 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 [22]).

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. 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.

All experimental results described below were obtained on a 200Mhz Pentium Pro equipped with 64mb RAM in less than two hours per run. On average, the computation of a probability $P(\xi^{(t)})$—which includes the computation of the corresponding α- and β-table—took less than 10 seconds for the size environments considered here. Data collection required between 15 and 20 minutes for each dataset. The (worst-case) memory complexity and computational complexity are linear in the size of d and in the size of the environment.

5 Results

The approach was tested using a B21 mobile robot, manufactured by Real World Interface, Inc (see Figure 4). Data was collected by joy-sticking the robot through its

Figure 4: The RWI B21 robot used in our research.

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 in the robot's environment (which roughly corresponded to the meet-points described in [6]). These were used as landmarks. When the robot came across a landmark location, the human operator pushed a button. Button presses were translated into two types of observations: A landmark observation for the robot's current position, and no-landmark observations for all other locations within a perceptual radius of 10 meter. The choice of 10 meter was somewhat ad hoc—in many cases, the "true" sidtance between landmark locations was smaller, causing somewhat of a perceptual conflict. To test the most difficult case, we assumed that the landmarks were generally indistinguishable.

Figure 5a 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 circle for the first time, the robot cannot exploit landmarks to improve its location estimates;thus,

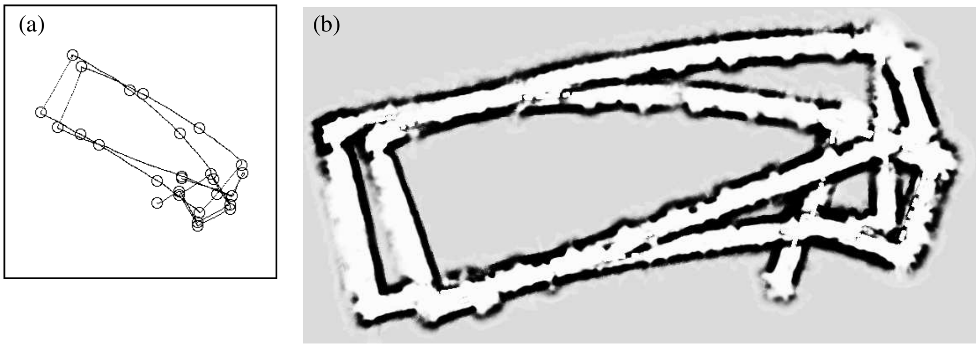

Figure 5: (a) Raw data (2,972 controls). The box size is 90 by 90 meters. Circles indicate the locations where landmarks were observed. The data indicates systematic drift, in some of the corridors. The final odometric error is approximately 24.9 meter. (b) Occupancy grid map, constructed from sonar measurements.

Figure 6: (a) Maximum likelihood map, along with the estimated path of the robot. (b) Occupancy grid map constructed using these estimated locations.

---

it accumulates odometric error. As Figure 5a 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 6a 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

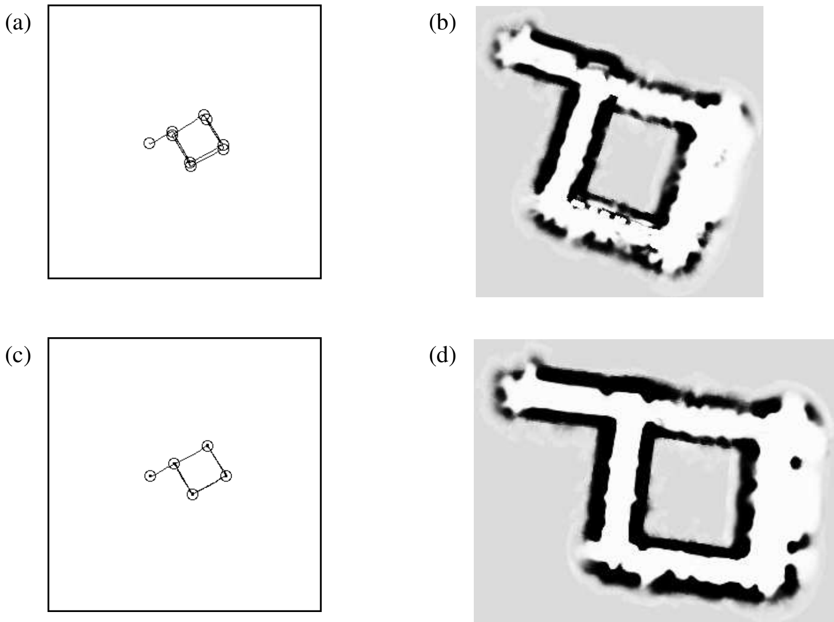

Figure 7: Even in this simple case (small cycle, only minor odometric error), our approach improves the quality of the map: (a) raw data, (b) occupancy grid map built from raw data, (c) corrected data, and (4) the resulting occupancy grid map.

method. In a series of experiments with this dataset, 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 [7, 19]. Figure 6b 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 [27]. In comparison, Figure 5b 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 navigation software (see [3, 28] for a description of the navigation routines).

Figures 7 to 11 show the map at different stages of the data collection. Figure 7 shows results for mapping the small cycle in the environment. Here most published methods should work well, since the odometric error is considerably

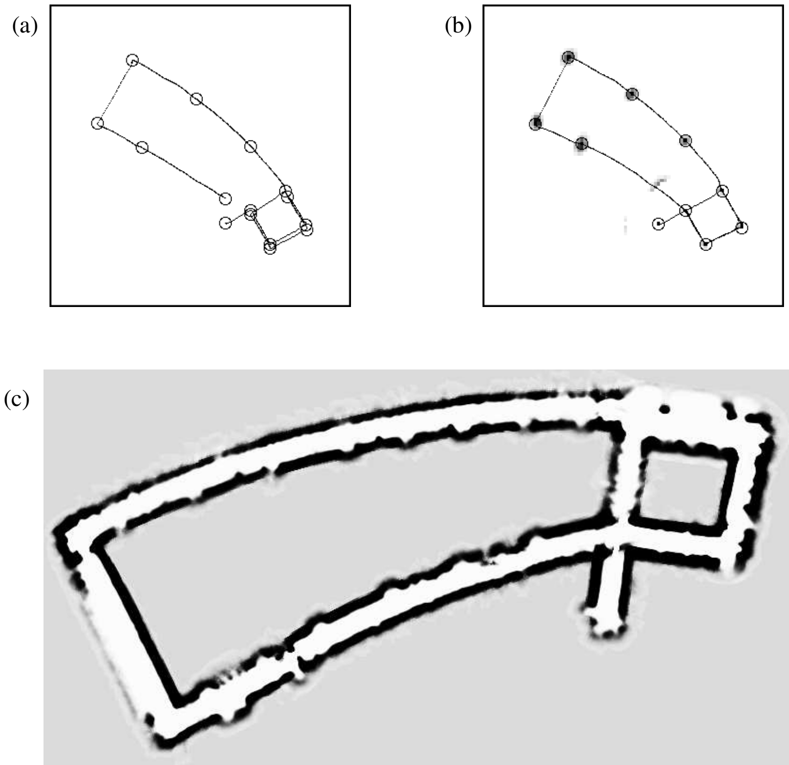

Figure 8: After observing the 15th landmark, the most plausible map is topologically incorrect, due to string odometric error. (a) Raw data, (b) Map and estimated trajectory, (c) occupancy grid map. The irregular dark areas in (b) indicate that the approach assigns high probability to several locations for the last step.

small. The quality of the occupancy grid map benefits from our approach, as shown in Figure 7b&d. In particular, the corrected occupancy grid map (Figure 7d) shows an obstacle on the right that is missing in the map constructed from raw data). The importance of revising past location estimates based on data collected later in time becomes more apparent when the robot maps the second circle in the environment. Here the odometric error is quite large (more than 14 meter). Figure 8-11 shows consecutive results after observing the 15th, 16th, 17th, and 20th landmark, respectively. While the maximum likelihood is topologically incorrect in the first two Figures, with 17 observations or more the most likely map is topologically correct. We conjecture that any incremental routine that does not

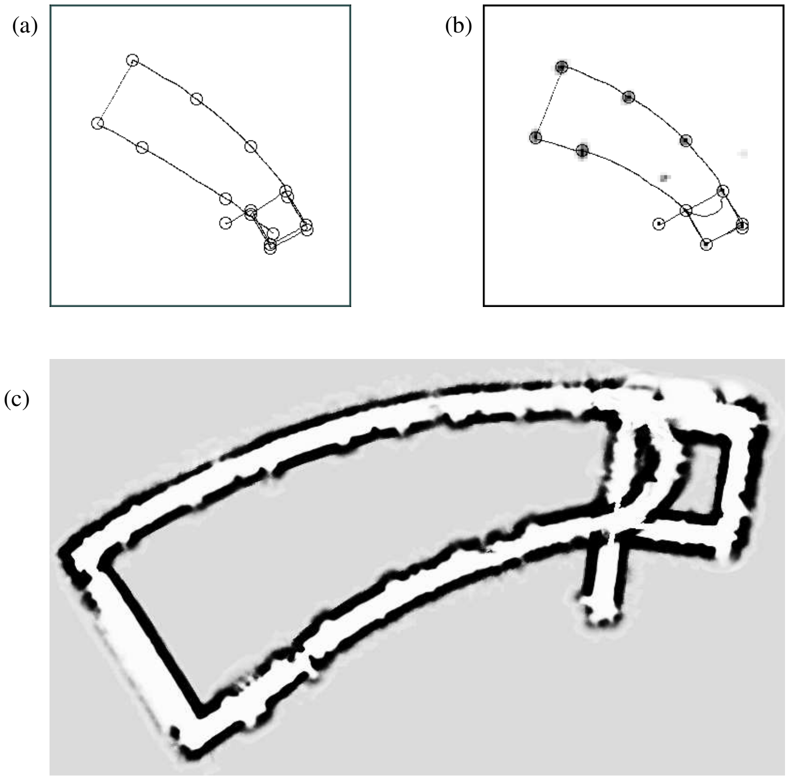

Figure 9: After observing the 16th landmark, the most plausible map is topologically still incorrect.

revise past location estimates would be bound to fail in such a situation.

Figures 12 and 13 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.

After convergence of the Baum-Welch algorithm, the β values demonstrate nicely the connection of the current approach and Markov localization. This is because the β-values globally localize the robot (with d in reverse order): The final value, $\beta^{(T)}$, is uniformly distributed, indicating that in the absence of any sensor data the robot's location is unknown. As T, decreases, an increasing number of observations and controls are incorporated into the estimation. Figure 14 shows an example, obtained using the second dataset. Here the last four β-tables

(a) (b) (c)

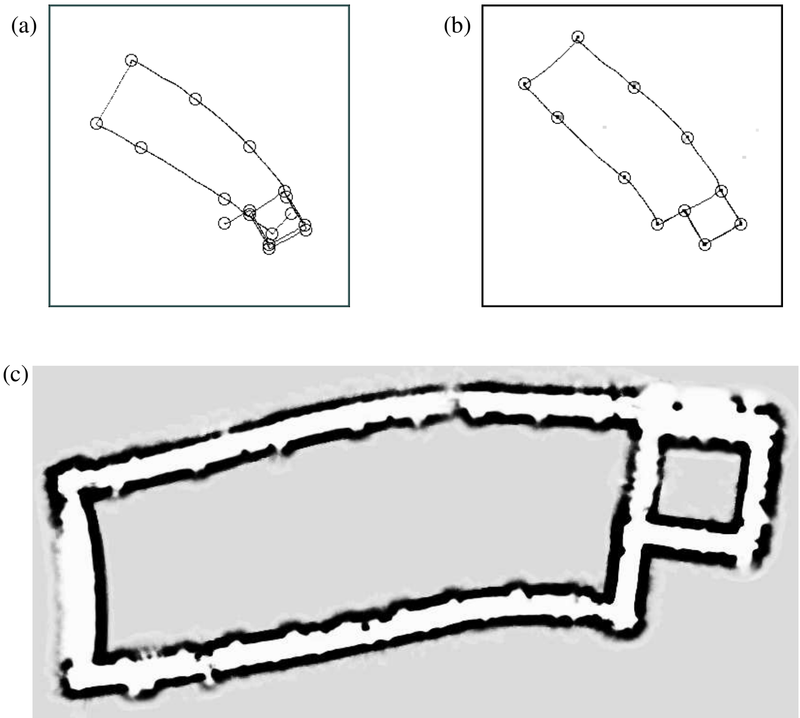

Figure 10: After observing the 17th landmark, our approach finds a topologically correct map. From this point on, the maximum likelihood map is *always* topologically correct.

$(\beta^{(24)}, \ldots, \beta^{(21)})$ are shown, after convergence of the map building algorithm. The final value, $\beta^{(24)}$, which is shown on the left in Figure 14, is uniformly distributed. With every step in the computation the uncertainty is reduced. After three steps, the approach has already uniquely determined the robot's position with high certainty (rightmost diagram). The α values, in contrast, effectively track a robot's position under the assumption that the initial position is known.

6 Application

The mapping algorithm was successfully employed in a practical problem, involving the fast acquisition of a map for a museum. In the past [8], we successfully deployed a robot in the "Deutsches Museum Bonn," with the task of engaging

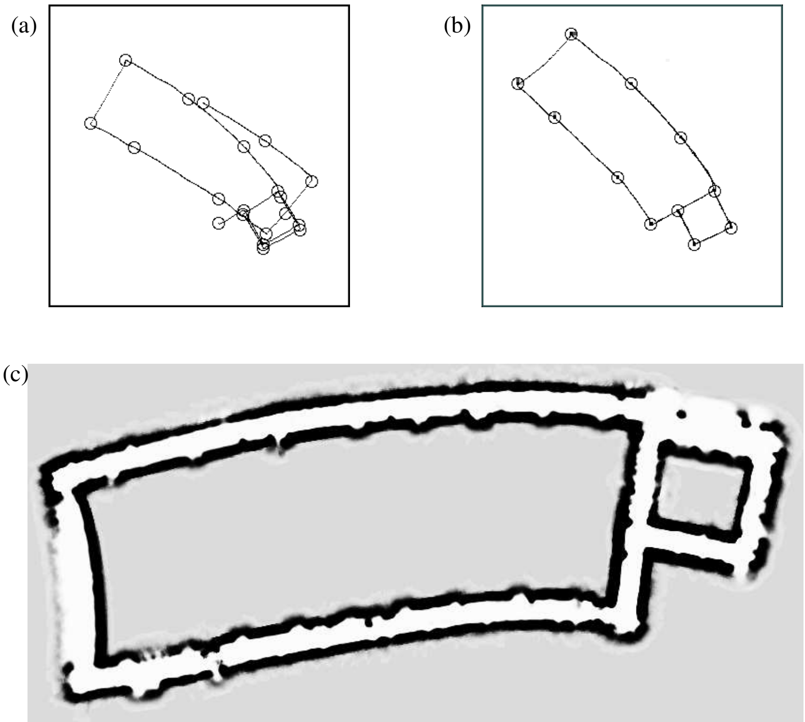

Figure 11: The map obtained after observing the 20th landmark is topologically correct.

people and providing interactive guided tours through the museum. During six days of operation, the robot entertained and guided more than 2,000 visitors of the museum, and an additional 600 "virtual" that commanded the robot through the Web. During those tours, it traversed approximately 18.5 km at an average speed of approximately 36.6 cm/sec. The reliability of the robot in reaching its destination was 99.75% (averaged over 2,400 tour goals).

One of the bottlenecks of this installation was the requirement for accurate maps. Our navigation software [28] requires highly accurate maps for reliable navigation. In fact, the map used in this exhibition was acquired by hand, and it took as approximately a week of (quite painful) tape-measuring, interleaved with data collection and tedious hand-tuning of the map, to come up with an accurate map. Accurate maps were of uttermost importance, since the robot had to be able to navigate even in extremely crowded environments (see Figure 15), while at the

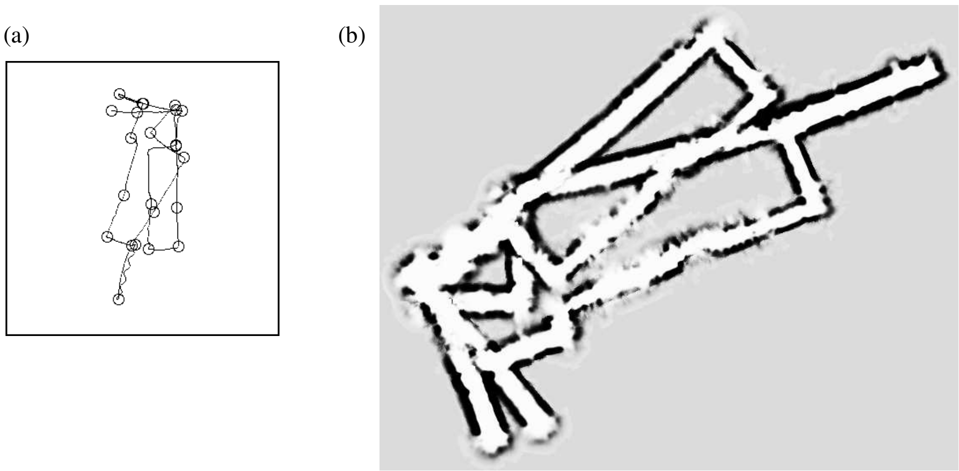

Figure 12: (a) A second dataset (2,091 controls, box size 90 by 90 meter), and (b) occupancy grid map, constructed from sonar measurements.

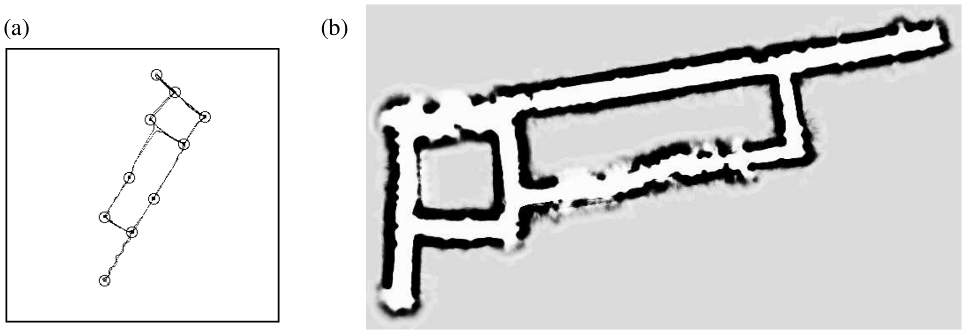

Figure 13: (a) Maximum likelihood map, along with the estimated path of the robot, and (b) the resulting occupancy grid map.

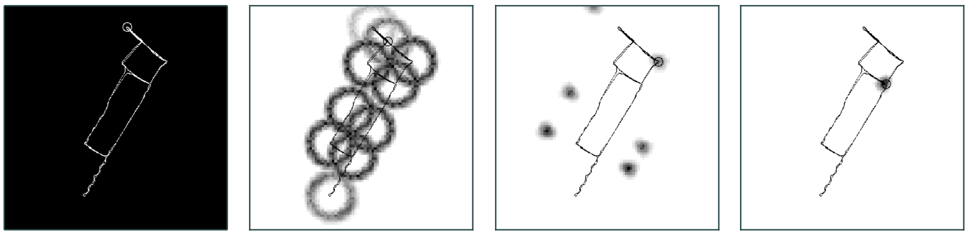

Figure 14: The last four β-tables ($\beta^{(24)}, \ldots, \beta^{(21)}$) after convergence. Here the circle marks the location of the robot, which is being estimated. See text.

Figure 15: The robotic tour-guide in action, in the "Deutsches Museum Bonn." To navigating safely through dense crowds while avoiding collisions with "invisible" obstacles (such as a metal plate shown in the center of this image) required accurate maps. The area in (a) measures only 60 by 60 meters, and the total map is only approximately 45 meter long.

---

same time a large number of obstacles were practically "invisible" to the robot's sensors (such as glass cages). In fact, three of the seven collision that our robot encountered during the exhibition were caused by inaccuracies in the map, which we than manually improved after the fact.

The current approach has already been useful in this context. We are currently installing a similar tour-guide in the Carnegie Museum of Natural Science in Pittsburgh, PA. Figure 16 shows a raw dataset, collected in the "Dinosaur Hall" of that museum. The Dinosaur Hall is significantly smaller than our testing environments. It is about 45 meter long, and the area shown in Figure 16a measures only 60 by 60 meters. The dataset was collected in less than 15 minutes: In about 3 minutes, we marked nine locations on the museum's floor using tape, and in an additional 11 minutes we joy-sticked the robot through the museum, pressing a button whenever it traversed one of the markers. We did not measure anything by hand (of course, the relative location of the markers to each other is estimated by the algorithm; it does not have to be measured manually). The final odometric error is approximately 25.1 meter and almost 90 degrees.

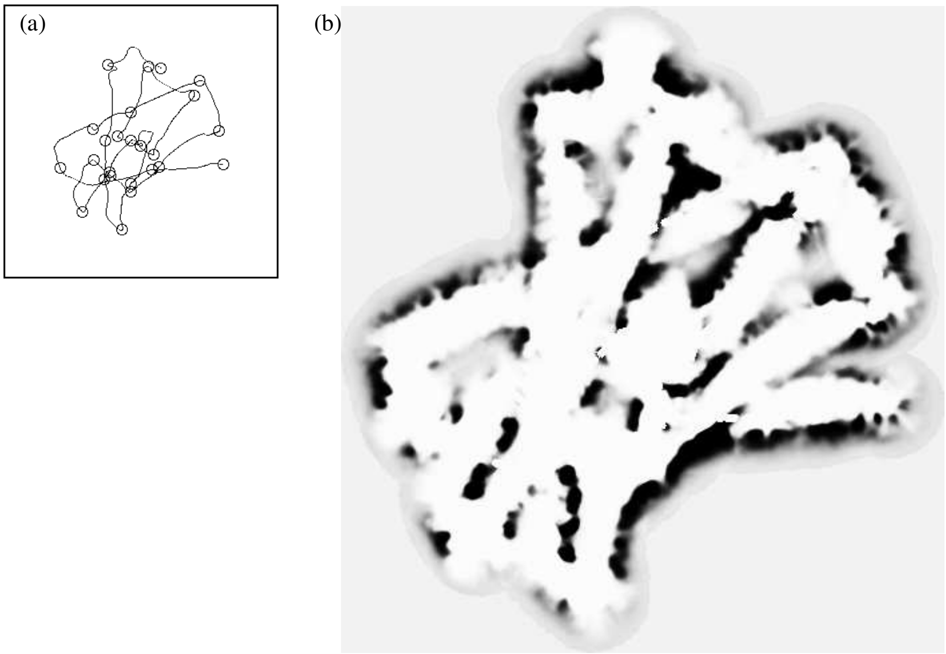

Figure 16: Raw data collected in the Carnegie Museum of Natural History of Natural Science in Pittsburgh, PA.

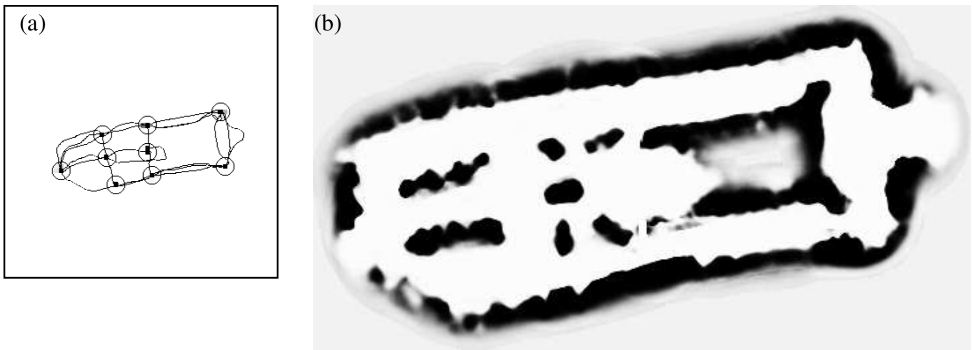

Figure 17: The corrected map of the Carnegie Museum of Natural History of Natural Science is good enough for the robotic tour-guide.

In approximately 41 minutes of computation (on a busy Pentium PC), our

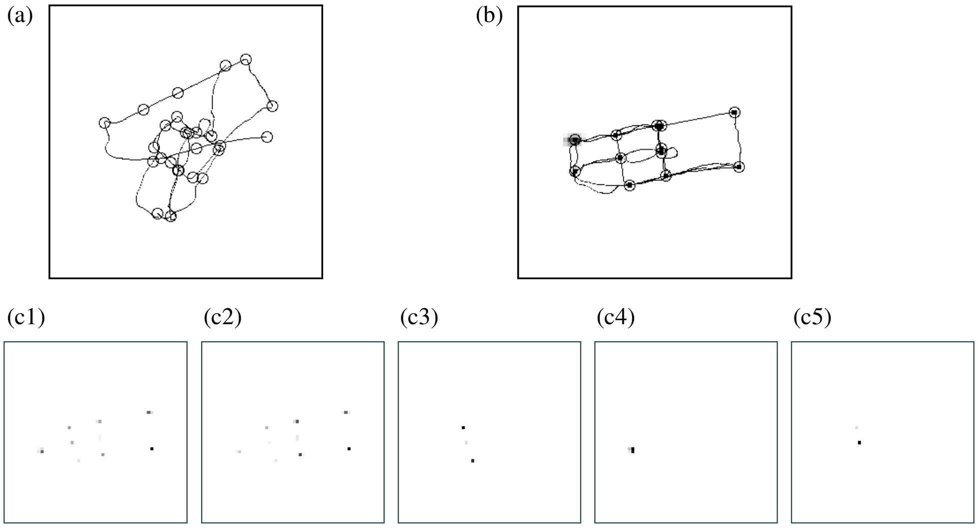

Figure 18: A second dataset is integrated into the first, both collected in the museum. The relative location of the second set with respect to the first is unknown. (a) Raw data, (b) result after a single iteration (less than 2 minutes computation), (c1)-(c5) The alpha values (in the first iteration of the estimation algorithm) demonstrate the localization under global uncertainty. After only four iterations, the robot knows with fairly high confidence where it is.

approach generated the map shown in Figure 17. While this map is not perfect, it is sufficient for navigation (once we draw in "invisible" obstacles by hand). Thus, our approach reduced the time to acquire a map from approximately a week to an hour or so. This is important to us since in the past we have frequently installed robots at various sites, often at conferences (most recently at IJCAI-97 in Japan), where time pressure prohibits modeling environments by hand. We conjecture that similar time savings can be achieved in installing robots in other indoor environments, such as hospitals [11].

7 Suitability for Collaborative Multi-Robot Mapping

Multi-robot collaboration is a topic that is currently gaining significant attention in the scientific community (see e.g., [17, 21]). A sub-problem of multi-robot collaboration is multi-robot map acquisition. In the most general problem, one would like to place robots at arbitrary locations in an unknown environment and have the robots build a single, consistent map thereof. In the most difficult case, the relative location of the robots to each other is unknown. Thus, to build a single map, the robots have to determine their position relative to each other, .i.e., there is a *global* localization problem.

As noticed above, our approach is a generalization of Markov localization, which has been demonstrated to localize robots globally [3]. To cope with multiple robots whose relative location is unknown, our basic approach has to be extended slightly. In particular, the initial position of the second robot relative to the first one is unknown. Thus, the initial belief $P(\xi^{(0)})$, and hence $\alpha_\xi^{(0)}$, is initialized *uniformly* for the second robot (and in fact, every other robot but the first). As in the single-robot case, the initial position of the first robot is defined as $(0, 0, 0)$, and $\alpha_\xi^{(0)}$ is initialized using a Dirac distribution (c.f., Equation (7)). With this extension, our approach is fit for collaborative multi-robot map acquisition.

To evaluate our approach empirically, we collected a second dataset in the Carnegie Museum of Natural Science. This dataset is shown in Figure 18a. Strictly speaking, this dataset was collected with the same robot. However, there is no difference to a dataset collected with a different robot of the same type, so that the results should directly transfer over to the multi-robot case.

Figure 18b shows the resulting position estimates after a single iteration of the EM algorithm, if the map generated using the first dataset is used as an initial map (shown in Figure 17a). After a single iteration, which requires less than two minutes of computation time, the robot has correctly determined its position relative to the first robot (with high confidence), and the resulting map incorporates observations made by both robots. Figures 18c1-c5 illustrate the efficiency with which the robot localizes itself relative to the existing map. Here the first alpha values, $\alpha_\xi^{(1)}, \alpha_\xi^{(2)}, \ldots, \alpha_\xi^{(5)}$, are depicted, in the first iteration of EM. Initially, after incorporating a single observation, the robot does not yet know where it is, but it assigns high likelihood to positions that were previously marked in the map. After only four steps, the robot knows where it is, as indicated by the unimodal distribution in 18c4. Not shown in Figure 18 are the corresponding β-values. After computing and incorporating those, the robot knows with high certainty where it was for any point in time.

The availability of an initial map greatly improves the computational efficiency of the approach. Our approach required 1 minute and 57 seconds for estimating the location of the robot when the previously acquired map was used, for a dataset that took 12 minutes and 19 seconds to collect. Thus, once a map is known, our approach appears to be fast enough to localize and track a robots as they move.

8 Related Work

Over the last decade, there has been a flurry of work on map building for mobile robots (see e.g., [5, 14, 23, 27]). As noticed by Lu and Milios [15], 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 [27]. 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. Koenig and Simmons investigated the problem under the assumption that an topologically correct sketch of the environment is available, which simplifies the problem somewhat [12]. They proposed a probabilistic framework similar to the one described here, which also employs the Baum-Welch algorithm for estimation. Shatkay and Kaelbling [24] generalized this approach for mapping in the absence of prior information. Their approach consults local geometric information to disambiguate different locations. Both approaches differ from ours in that they build topological maps. They do not explicitly estimate global geometric information (i.e., x-y-θ positions). As acknowledged in [24], the latter approach fails to take the cumulative nature of rotational odometric error into account. It also violates a basic "additivity property" of geometry (see [24]). Even in the absence of odometric error, it is unclear if the approach will always produce the correct map.

Lu and Milios [15, 16] have proposed a method that matches laser scans into partially built maps, using Kalman filters for positioning. Together with Gutmann [9], they have demonstrated the appropriateness of this algorithm for mapping environments with cycles. Their approach 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" [15]. In all cases studied in [9, 15, 16], 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.

To the best of our knowledge, the problem of multi-robot map acquisition has not been investigated before.

The approach proposed in this paper also relates to work in the field of Markov localization, which requires a map to be given. Recently, Markov localization has been employed by various groups with remarkable success [3, 10, 12, 20, 25, 28].

In our own work, Markov localization played a key role in a recent installation in the Deutsches Museum Bonn, where one of our robots provided interactive tours to visitors. In more than 18.5km of autonomous robot navigation in a densely crowded environment (top speed 80 cm/sec, average speed 36 cm/sec), Markov localization was absolutely essential for the robot's safety and success [8]. The method proposed here directly extends this approach. In future installations of the tour-guide robot, maps do not have to be crafted manually but can now be generated by joy-sticking a robot through its environment. This will reduce the installation time from several days to only a few hours.

9 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. We have already integrated the approach with our previously developed algorithm for greedy occupancy-grid-based exploration described in [2, 27, 28] (see also [29]); however, no systematic results are available at this point in time. Another possibility, which has not yet been implemented, would be to combine the current approach with Choset's sensor-based covering algorithm [6].

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. Recent research on landmark-based navigation has produced a large number of methods for recognizing specific landmarks (see, e.g., [1]). In particular, Choset's sensor-based covering algorithm [6] automatically detects and navigates to so-called *meet-points*. Meet-points correspond to intersections, corners, and dead-ends (see also [13]). 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.

One interesting extension would be to apply the proposed method to other types representations, with different sensor models. The perceptual model used here, which is based on landmarks, is just one choice out of many possible choices. A different choice would be the probabilistic sensor model described in [4, 3], which specifically applies to proximity sensors, such as sonars or laser range finders. The inverse sensor model (also called sensor interpretation), which is employed in the map building step (M-step), can be realized by the approach described in [27], where neural networks are used to extract occupancy grid maps from sensor data. As a result, proximity sensor readings would directly incorporated in the position estimation, which is currently not the case. Such an approach would also obliviate the need for landmarks. The extension of the current approach to such more complex sensor models is subject to future work.

Acknowledgment

This research benefited from discussions with Howie Choset, Keiji Nagatani, and Hagit Shatkay. The authors specifically thank Hagit Shatkay for insightful comments on an earlier version of this paper.

References

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

[2] J. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot Rhino. *AI Magazine*, 16(1), 1995.

[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, D. Hennig, and T. Schmidt. Position tracking with position probability grids. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

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

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

[8] D. Fox, W. Burgard, and Thrun. S. A hybrid collision avoidance method for mobile robots. submitted for publication.

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

[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] 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.

[13] 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.

[14] 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.

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

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

[17] M. J Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, January 1997.

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

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

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

[21] L. E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 10(6), 1996.

[22] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. In *IEEE ASSP Magazine*, 1986.

[23] W.D. Rencken. Concurrent localisation and map building for mobile robots using ultrasonic sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2129–2197, Yokohama, Japan, July 1993.

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

[25] 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.

[26] S. Thrun. A bayesian approach to landmark discovery and active perception for mobile robot navigation. Technical Report CMU-CS-96-122, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213, April 1996.

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

[28] 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.

[29] B. Yamauchi and R. Beer. Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Special Issue on Learning Autonomous Robots, 1996. also located at http://www.aic.nrl.navy.mil/~yamauchi/.