Sonar-Based Mapping With Mobile Robots Using EM

Wolfram Burgard[1] **Dieter Fox**[2] **Hauke Jans**[1] **Christian Matenar**[1] **Sebastian Thrun**[2]

[1]Department of Computer Science
University of Bonn
D-53117 Bonn, Germany

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

Abstract

This paper presents an algorithms for learning occupancy grid maps with mobile robots equipped with range finders, such as sonar sensors. Our approach employs the EM algorithm to solve the concurrent mapping and localization problem. To accommodate the spatial nature of range data, it relies on a two-layered representation of maps, where global maps are composed from a collection of small, local maps. To avoid local minima during likelihood maximization, a softmax version of the M step is proposed that is gradually annealed to the exact maximum. Experimental results demonstrate that our approach is well suited for constructing large maps of typical indoor environments using sensors as inaccurate as sonars.

1 Introduction

Learning maps with mobile robots has frequently been recognized as one of the most fundamental problems in mobile robotics [CW90, KBM98]. This is because the map learning problem, often referred to as *concurrent mapping and localization*, is a chicken-and-egg problem. If the pose (we use the term *pose* to refer to a robot's x-y location and its heading direction θ) of the robot was always known during mapping—which is actually assumed by the majority of work in the field—building maps is relatively easy, and there is a large number of approaches that work well. On the other hand, if a map was available, determining the robot's poses at any time is relatively straightforward, as methods exist that even work reliably if the environment is non-stationary (e.g., densely populated by people) [BAC$^+$98, FBTC98]. In map learning, however, neither the robot's poses nor the map is known initially—so both must be estimated from data. Unfortunately, errors in odometry (wheel encoders) amplify over time. For example, small rotational error can have a huge effect on the robot's x-y-location later in time. This makes it difficult to estimate poses while learning maps.

In the past, the problem of map learning has been tackled by a great number of researchers (see e.g., [CL85, Cho96, CKK95, Elf87, Elf89, KB91, Mat90, Mor88, Ren93, SK97]). Some recent methods, such as the *metric* approaches described in [GN97, LM97, TGF$^+$98], have successfully been applied for learning maps up to 80 by 25 m^2 in size. However, they are closely married to *laser range finders*, a highly accurate and quite expensive sensor technology. Other methods, such as the *topological* approaches described in [Sha98, TFB98], rely on *landmarks* to build maps. They work with less accurate sensors (such as sonars), but they are forced to throw away almost all sensor data, except for extremely scarce landmark data. For example, they cannot exploit the width of a corridor to disambiguate different corridors. Some of the largest topological maps (80 by 25 m^2) have been learned using the algorithm in [TFB98]. However, it has only been demonstrated to work with pre-defined landmarks. It is remarkable, however, that most successful approaches to date use statistical techniques for map learning.

The approach proposed here builds directly on our previous work [TFB98, TGF$^+$98]. As there, the map learning problem is phrased as a maximum likelihood estimation problem, and EM is applied for finding a local maximum in likelihood space. The algorithm proposed here differs in several aspects, crucial for its scalability. First, *small, local maps* are generated from short sequences of sensor data, exploiting the fact that in the short term odometry errors are typically small and can be neglected. These local maps replace the landmarks in [TFB98], thereby overcoming the reliance on manually labeled landmark locations. More importantly, our approach relaxes an independence assumption made in [TFB98] by a weaker one. Our previous approach represented maps in the E-step by a single, global grid; and each grid cell was estimated *independently* of other grid cells. While this is unproblematic if the robot approximately knows its position (in which case the mapping step is identical to occupancy grids [Elf87, Elf89, Mor88, Thr98]), this independence assumption leads to highly ambiguous maps when the robot is uncertain, which makes it difficult to localize the robot with sonar sensors. Here we replace these flat maps using a two-layered representation. Our algorithm replaces the single map by a collection of local maps, annotated with their pose distributions. This approach maintains dependencies between different grid cells within each local map. It enables us to use conventional metric Markov localization [BAC+98, BFHS96, BDFC98] (see also [KCK96, KS96, NPB95, SK95]) in the E-step. As a result, our new algorithm inherits all advantages of the EM approach and simultaneously is able to operate on raw proximity information without relying on pre-defined landmarks. Our approach makes EM applicable to learning maps of typical indoor environments of mobile robots equipped with sonar sensors, or other low-resolution proximity sensors.

Empirical results, described in this paper, illustrate that our approach succeeds in learning large-scale maps purely based on sonar measurements. To demonstrate its robustness, experiments with amplified odometric noise illustrate that our approach is applicable even under significant odometric errors.

2 Statistical Foundations

2.1 Models

We formulate the problem of concurrent mapping and localization as a statistical *maximum likelihood estimation problem* [TFB98]. To generate a map, we assume that a robot is given a stream of sensor data, denoted

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

where $o^{(t)}$ stands for an *observation* that the robot made at time t (e.g., using its sonar sensors), and $u^{(t)}$ for an odometry reading that characterizes the action 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 can be posed as the problem of finding the most likely map given the data. A *map*, denoted $m = \{m_{x,y}\}_{x,y}$, 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 [CKK95] or, alternatively, locations of significant places [KB91, Cho96]. Metric approaches, on the other hand, usually use the location of obstacles as properties-of-interest [CL85, Mor88, LM97]. Here we follow the metric approach.

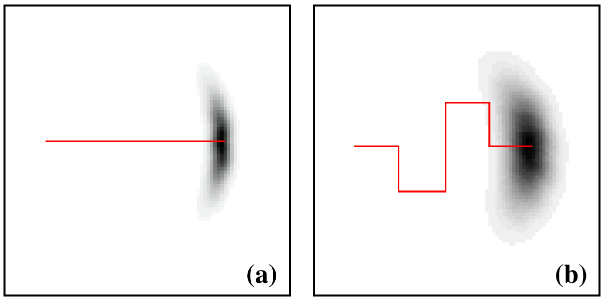

Figure 1: Motion model. The grayly shaded area shows the pose distribution (projected into 2D) after the motion command indicated by the solid line.

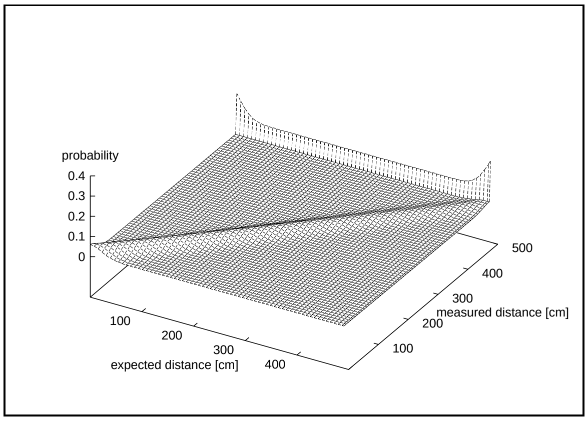

Figure 2: Perception model for ultrasound sensors. Probability of measuring a certain distance given the expected distance according to the model of the environment.

Our approach assumes that the robot is given three basic, probabilistic models, one that describes robot motion, one that models robot perception, and an inverse model of robot perception.

- **The motion model**, denoted $P(\xi' \mid u, \xi)$, 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 (the x-y-location of a robot together with its heading direction θ). Our motion model assumes independent bounded and normally distributed errors in the length of the distance traveled as well as in the rotation carried out. Figure 1 illustrates the

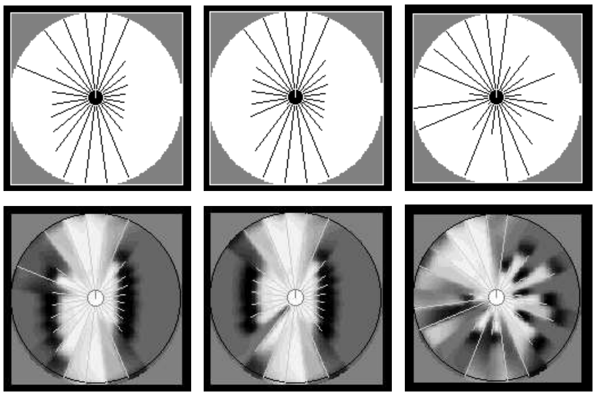

Figure 3: Inverse perceptual models: The top row shows raw sensor data, the bottom row shows a likelihood field (local map): the brighter a pixel, the higher its likelihood for being unoccupied. The inverse perceptual model has been learned from hand-labeled data, using artificial neural networks.

motion model, by showing the probability distribution for ξ' upon executing different motion actions. The images show projections of the tree-dimensional state space of the robot onto the x-y-plane.

- **The perception model**, denoted as $P(o \mid m, \xi)$, models the likelihood of observing o in situations where both the world m and the robot's pose ξ are known. For sonar sensors, as used throughout our experiments, this model is shown in Figure 2. Shown there is the probability of a sonar reading (vertical axis) as a function of the *correct* distance (determined using ray tracing) and the measured distance. The graph in Figure 2 has been generated from several millions of raw sonar readings (see [FBT98, Fox98]). It is the result of fitting a model consisting of a mixture of of a linear-Gaussian (centered around the correct distance), a Geometric distribution (modeling overly short readings) and a Dirac distribution (modeling max-range readings) to this data.

- **The inverse perception model** is denoted as $P(m \mid o, \xi)$. While the inverse perception model can be derived from the perception model using Bayes rule, it is computationally convenient to have an explicit inverse model for mapping. Here we use a model previously described in [Thr98], which applies neural network learning to compute the likelihood of occupancy from sonar scans. Figure 3 shows examples.

These three quantities, i.e., the given set d of data, the motion model $P(\xi' \mid u, \xi)$, and the perception model $P(o \mid m, \xi)$ form the statistical basis of our approach.

2.2 The Map Likelihood Function

Following [TFB98], the problem of mapping is the problem of finding the most likely map m^* given the data

$$m^* = \operatorname*{argmax}_m P(m \mid d). \tag{2}$$

As shown there, the probability $P(m \mid d)$ can be rewritten as

$$P(m \mid d) = \lambda \int \dots \int \prod_{t=1}^{T} P(o^{(t)} \mid m, \xi^{(t)}) \prod_{t=1}^{T-1} P(\xi^{(t+1)} \mid u^{(t)}, \xi^{(t)})\; d\xi^{(1)}, \dots d\xi^{(T)}. \tag{3}$$

We are only interested in maximizing $P(m \mid d)$, hence the normalizer λ can safely be dropped. The remaining expression is a function of the data d, the perceptual model $P(o \mid m, \xi)$, and the motion model $P(\xi' \mid u, \xi)$. Maximizing this expression is equivalent to finding the most likely map.

2.3 EM for Learning Maps

Computing the *global* maximum of (3) is computationally challenging. The currently best-known solutions perform hill climbing in likelihood space. Following [TFB98], our approach uses the EM algorithm [DLR77, MK97] for maximizing likelihood. 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 [KS96, SK97]):

1. In the E-step, the robot computes probabilities $P(\xi \mid m, d)$ 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 most likely map is determined by maximizing $\operatorname*{argmax}_m P(m \mid \xi, d)$, 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.

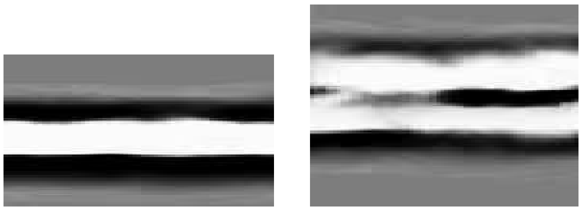

Figure 4: Local map (left) and misleading global map (right), generated by convolving the local map with a bi-modal pose distribution. Such maps occur during optimization, when the robot has not yet determined its position. They make Markov localization fail. Our approach, hence, maintains the local maps during EM, and builds the global map only at the end.

3 Two-layered Map Representations

In our previous work, maps were represented by flat grids. Each grid cell represented the posterior probability of it containing a landmark. Unfortunately, this is problematic when applying EM to maps constructed by proximity sensors. Figure 4 illustrates why: Suppose the robot's sensor readings suggest a map like the one shown on the left side of Figure 4, but it has two distinct hypotheses as to where it might be. Integrating a single global map (which involves convolution with the robot's uncertainty) may yield the map shown on the right side of Figure 4. While for each individual map pixel, this map *correctly* estimates the likelihood of occupancy, the map as a whole is not usable: In some areas, it shows 3 walls, instead of 2; in others, the corridor appears to be twice as wide. Such maps are plainly not usable for localization with proximity sensors, as any specific sensor scan will not fit well anywhere into such maps. The situation is even worse when the robot is more globally uncertain, in which case the resulting maps often do not contain any visible structure.

This effect, which in fact is a consequence of the independence assumption in occupancy grids [Mor88], can be avoided by adopting a richer [JM98], layered representation. Our approach represents the world by a collection of small, local maps annotated by their location. Formally, a map m is a conjunction of N small maps, denoted $m_{[i]}$, annotated by a coordinate transform, denoted $\xi_{[i]}$:

$$m \quad = \quad \{\langle m_{[i]}, \xi_{[i]} \rangle\}_{i=1,\ldots,N} \tag{4}$$

We will refer to $\xi_{[i]}$ as the *pose* of the i-th local map (poses of maps are essentially the same as robot poses). For brevity, let us write

$$M_{[i]} \quad := \quad \langle m_{[i]}, \xi_{[i]} \rangle \tag{5}$$

and hence

$$m \quad = \quad \{M_{[i]}\}_{i=1,\ldots,N} \tag{6}$$

By representing maps in two layers, we eliminate the problems arising from using a single, monolithic map. This is because local maps are not convolved with their pose distribution; instead, the pose distribution is represented explicitly.

Our layered map representation requires a new definition of the perceptual model $P(o \mid m, \xi)$. Our approach makes the following conditional independence assumption:

$$P(M_{[i]} M_{[j]} \mid o, \xi) \quad = \quad P(M_{[i]} \mid o, \xi)\; P(M_{[j]} \mid o, \xi) \tag{7}$$

for all $i \neq j$. Clearly, this assumption is only approximately correct, but it is necessary to keep the computation manageable and in practice seems to work well (it only introduces errors in regions where local maps overlap, which results in overly confident estimates). The independence assumption allows us to extend our perceptual model to maps represented in layers:

$$\begin{aligned}
&P(o \mid \xi, m) \\
&= P(o \mid \xi, \bigotimes_{i=1}^{N} M_{[i]}) \\
&= \frac{P(\bigotimes_{i=1}^{N} M_{[i]} \mid o, \xi) P(o \mid \xi)}{P(\bigotimes_{i=1}^{N} M_{[i]} \mid \xi)} \\
&= \frac{\prod_{i=1}^{N} P(M_{[i]} \mid o, \xi) P(o \mid \xi)}{\prod_{i=1}^{N} P(M_{[i]} \mid \xi)} \\
&= \frac{1}{P(o \mid \xi)^{(N-1)}} \prod_{i=1}^{N} \frac{P(M_{[i]} \mid o, \xi) P(o \mid \xi)}{P(M_{[i]} \mid \xi)} \\
&= \frac{1}{P(o \mid \xi)^{(N-1)}} \\
&\qquad \prod_{i=1}^{N} \frac{P(o \mid M_{[i]}, \xi) P(M_{[i]} \mid \xi)}{P(o \mid \xi)} \; \frac{P(o \mid \xi)}{P(M_{[i]} \mid \xi)} \\
&= \frac{1}{P(o \mid \xi)^{(N-1)}} \prod_{i=1}^{N} P(o \mid \xi, M_{[i]})
\end{aligned} \tag{8}$$

Here the expressions $P(o \mid \xi, M_{[i]})$ are computed as described in Section 2, using the learned sensor model shown in Figure 2b. The term $P(o \mid \xi)^{(N-1)}$ represents the probability of observing o given the position ξ of the robot. This probability is a constant, since it is independent of the map. In the subsequent analysis, it will be subsumed in other normalizers and hence be dropped from consideration at this point.

As Equation (8) shows, our approach computes the likelihood of observations separately in all maps, and then combines the resulting density multiplicatively. In practice, maps which cannot overlap with the robot's position can be safely eliminated, thereby speeding up the computation while not affecting the final result.

4 Mapping

We are now ready to present the map learning algorithm. In a first pre-processing phase, our algorithm extracts local maps from the data d. It then applies EM, iterating localization (E-step) and local map alignment (M-step). Finally, a single global map is generated from the result of EM.

Figure 5 illustrates the difference between our past approach [TFB98, TGF+98], and the one presented here. Previously, EM was used to directly generate the most likely map, which works well for landmarks, but purely for proximity sensors (such as sonars). Our new approach uses an intermediate, two-layered representation, in which EM is used to estimate the position of many local maps.

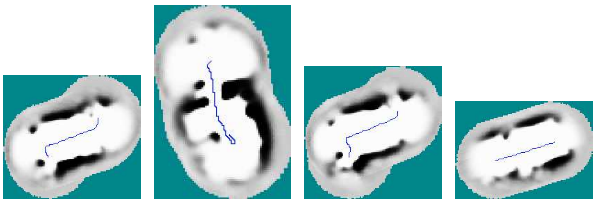

Figure 6: Examples of local maps, annotated by robot trajectories. These maps have been constructed from sonar measurements.

4.1 Pre-Processing

First, our approach generates the local maps $m_{[i]}$. This is done using conventional occupancy grid mapping [Elf87, Mor88, YLS+98], exploiting the fact that odometric errors can typically be neglected in the short range. In our current implementation, a new map is generated after every 5 meter of robot travel, during which the accumulated odometry error can be safely neglected. As mentioned above, our current implementation applies Backpropagation networks and Bayes rule to estimate the probability of occupancy in regions covered by the robot's sensors [Thr98]. Odometric errors are not corrected when building local maps. Figure 6 shows examples of such local maps. Local maps are computed only once in the beginning.

4.2 The E-Step

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

$$\begin{aligned} P(\xi^{(t)} \mid d, m) \quad &= \quad \eta \underbrace{P(\xi^{(t)} \mid o^{(1)}, \ldots, o^{(t)}, m)}_{:=\alpha^{(t)}} \\ &\qquad \underbrace{P(\xi^{(t)} \mid u^{(t+1)}, \ldots, o^{(T)}, m)}_{:=\beta^{(t)}} \end{aligned} \tag{9}$$

Here η is a normalizer that ensure that the left-hand side of Equation (9) sums up to one (see [TFB98] 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 [RJ86].

The computation of the α-values and the β-values are modified versions of *Markov localization*, using multiple (local) maps. Markov localization has recently been used with great success for robot localization in *known* environments by various researchers [BAC+98, KCK96, KS96, NPB95, SK95].

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 $(0, 0, 0)$:

$$\begin{aligned} \alpha^{(1)} \quad &= \quad P(\xi^{(1)} \mid o^{(1)}, m) \\ &= \quad \left\{ \begin{array}{l} 1, \text{ if } \xi^{(1)} = (0,0,0) \\ 0, \text{ if } \xi^{(1)} \neq (0,0,0) \end{array} \right. \end{aligned} \tag{10}$$

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

$$\begin{aligned} &\alpha^{(t)} = \\ &\quad \eta \; P(o^{(t)} \mid \xi^{(t)}, m) \; P(\xi^{(t)} \mid o^{(1)}, \ldots, u^{(t-1)}, m) \end{aligned} \tag{11}$$

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

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

Substituting (12) into (11) yields a recursive rule for the computation of all $\alpha^{(t)}$ with boundary condition (10). See [TFB98] for a more detailed derivation.

Computing the β-Values: The computation of $\beta^{(t)}$ is completely analogous but takes place backwards in time.

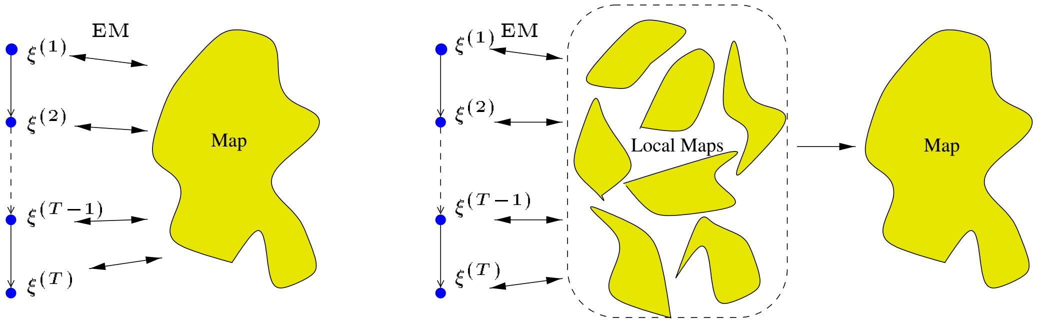

Figure 5: EM using a flat map representation (left) and using our new two-layered representation (right). In the layered approach, EM is applied for aligning small, local maps, which after the optimization are projected into a single, global map.

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:

$$\beta^{(t)} = \eta \int P(\xi^{(t+1)} \mid u^{(t)}, \xi^{(t)}) \\ P(o^{(t+1)} \mid \xi^{(t+1)}, m)\ \beta^{(t+1)}\ d\xi^{(t+1)} \quad (13)$$

The derivation of the equations are analogous to that of the computation rule for α-values and can be found in [TFB98]. 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.

4.3 The M-Step with Deterministic Annealing

The M-step calculates the most likely poses of the local maps. Let $t_{[i]}$ be the index of the first data point in the i-th local map. In the most generic setting, the M-step calculates

$$\begin{aligned} \xi_{[i]} &= \operatorname*{argmax}_{\xi} P(\xi^{(t_{[i]})} \mid d, m) \\ &= \operatorname*{argmax}_{\xi} \alpha^{(t_{[i]})}\ \beta^{(t_{[i]})} \end{aligned} \quad (14)$$

Unfortunately, this approach is problematic in practice. EM is a hill climbing method, which only converges to *local* maxima. If the odometric error is large, the initial map will be erroneous, and subsequent iterations of EM might not be able to recover.

The danger of getting stuck in a local maximum can be reduced significantly by a modified M-step. Instead of keeping track of the most likely model, our approach generates a distribution over models that slowly converges to the most likely one. More specifically, the M-step generates a distribution over poses, denoted μ:

$$\mu(\xi_{[i]}) \quad = \quad \eta\ \left(\alpha^{(t_{[i]})}\beta^{(t_{[i]})}\right)^{\frac{1}{\theta}} \quad (15)$$

Here η is a (different) normalizer that ensures that the probabilities integrate to 1. The parameter θ is a control parameter in $(0, 1]$ which will, in analogy to the rich literature on annealing, be referred to as *temperature*. Equation (15) is a version of the softmax function [Now91]. When $\theta = 1$, the full distribution over all poses is retained. When $\theta = 0$, our approach is equivalent to (14). To accommodate the fact that the E-step generates a distribution over maps, rather than a single map only, the expression $P(o \mid \xi, M_{[i]})$ in Equation (8) has to be substituted by

$$\int P(o \mid \xi, \langle m_{[i]}, \xi_{[i]} \rangle)\ \ \mu(\xi_{[i]})\ \ d\xi_{[i]} \quad (16)$$

In our approach, θ is initialized with 1 and annealed towards zero using an exponential cooling schedule. The effect of the annealed softmax function in the E-step is to avoid early commitment to a single map. Instead, one can think of our approach as moving from density estimation (over the pose parameters in the map) to maximum likelihood estimation. The reader should notice that our approach is not the first to employ annealing to avoid local maxima in EM [HPJ99, Ros98].

Both the E-step and the M-step are alternated until convergence (using a criterion like a threshold on the minimum amount of change). In practice, less than 7 iterations sufficed in all our experiments.

4.4 Post-Processing

To come up with a single global map of the environment, the local maps $m_{[i]}$ are integrated based on their final poses $\xi_{[i]}$. This is done using Bayes rule, following the standard approach to building occupancy maps, as described in [Mor88, Thr98].

More specifically, let $\langle x, y\rangle$ be arbitrary coordinates of a grid cell in the global map. Let $\langle x_{[i]}, y_{[i]}\rangle$ specify the same location in the local coordinate system if the i-th local map $m_{[i]}$, which is easily obtained by applying the coordinate transformation induced by $\xi_{[i]}$. Let us furthermore use the indicator function

$$I_{\langle x,y\rangle \in m_{[i]}} \tag{17}$$

to indicate whether the local map $m_{[i]}$ covers the coordinate $\langle x, y\rangle$. Then the occupancy of $\langle x, y\rangle$ in global coordinates is given by:

$$P(occ_{x,y} \mid d) = P(occ_{x,y} \mid \bigotimes_{i=1}^{N} \langle m_{[i]}, \xi_{[i]}\rangle) \quad = \tag{18}$$

$$1 - \left(1 + O(occ) \prod_{i: I_{\langle x,y\rangle \in m_{[i]}}} \frac{m_{[i],\langle x_{[i]}, y_{[i]}\rangle}}{1 - m_{[i],\langle x_{[i]}, y_{[i]}\rangle}} \, O(occ)^{-1}\right)^{-1}$$

where

$$O(occ) \quad = \quad \frac{P(occ)}{1 - P(occ)}. \tag{19}$$

$P(\mathrm{occ})$ is the *prior* for the occupancy of a grid cell (it vanishes when set to 0.5), and $m_{[i],\langle x_{[i]}, y_{[i]}\rangle}$ denotes the likelihood for occupancy at $\langle x_{[i]}, y_{[i]}\rangle$ in the i-th local map. The derivation of (19) is completely analogous to the familiar occupancy grid approach, which is used to construct the local maps in the pre-processing phase). It can be found in [Thr98].

5 Experimental Results

Our approach was evaluated using various data sets collected in two indoor environments. Here we show results for two of them, one collected in the University of Bonn, and one at Carnegie Mellon University. The first is difficult because of the large number of offices, which makes it challenging to produce a straight corridor. The other contains a cycle—which is more challenging to map than non-cyclic environments [GN97, LM98]. All data has been collected using RWI B21 robots (RHINO at the University of Bonn and AMELIA at Carnegie Mellon University), equipped with 24 Polaroid sonar sensors with 15° opening angle (main cone). The reader may notice that building maps from sonar data is more difficult than building maps with high-precision laser range finders, due to their limited resolution and accuracy. Most previous approaches that successfully mapped circular environments required laser range finders (see e.g., [GN97, LM97, TGF+98]).

In our implementation, we used grids to represent probability densities, with a spatial resolution of 15 cm and an angular resolution of 2 degrees. The "tricks" described in [TFB98] were applied to keep the computation manageable.

Figure 7 shows on the left side a synthetic map of the first environment, along with the robot's path. To understand the best possible map that can be built from sonar sensors in this environment, we initially provided the robot with a correct map and used Markov localization to localize the robot. As reported in [BDFC98, GBFK98], this approach yields accurate position estimates even for ultrasound sensors. We then built a new map using the previously corrected position estimates. This map is shown on the right side of Figure 7. It can be viewed as the best possible result for EM.

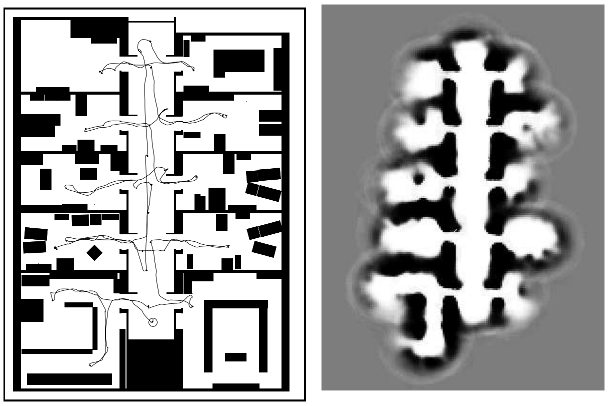

Figure 7: CAD-map of the first testing environment (left). Desks and tables are below the height of the sonars. Occupancy grid map constructed using the CAD map for localization (right). This map reflects the best possible outcome of EM.

To elucidate the scaling properties of our new algorithm, we artificially added random noise to the robot's odometry. Figure 8 shows in the upper row maps generated with different levels of artificial noise added. Here the raw odometry is insufficient to generate meaningful maps. These results are similar to those one would expect when using a low-accuracy robot, or when operating robots under less optimal conditions (e.g., on dirty surfaces that promote slippage). The lower row of this figure shows the resulting maps after applying our method. While in fact

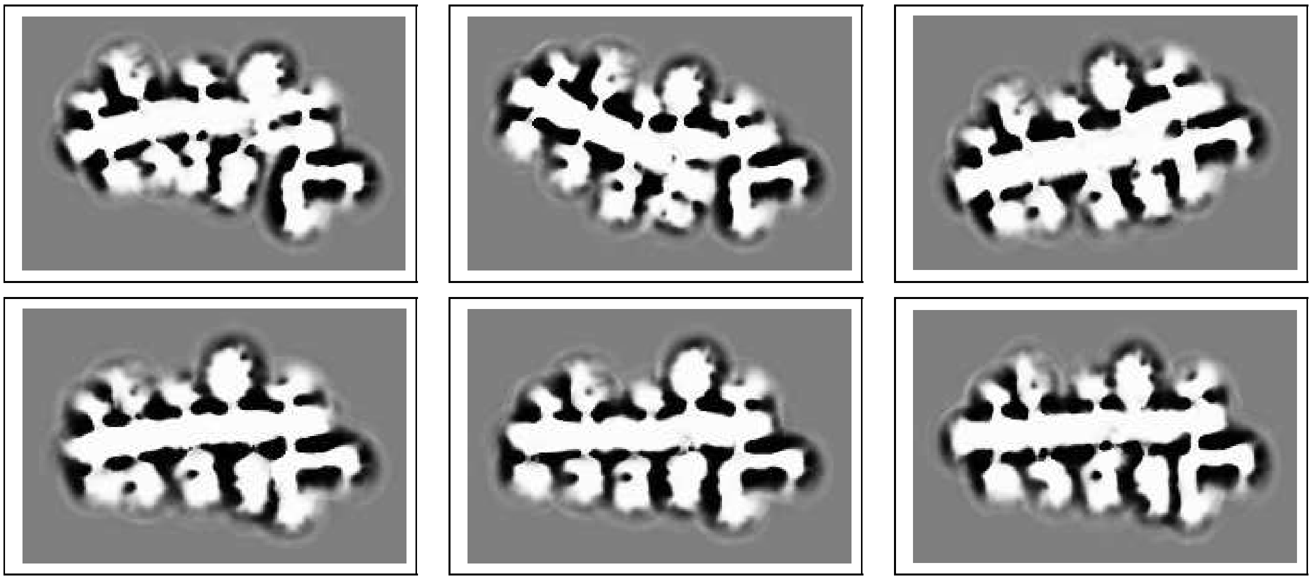

Figure 8: Maps of the first testing environment. The maps in the upper row been generated by adding different levels of noise to the odometry data. The lower row contains the maps obtained as output from applying EM.

our sonar data is insufficient to completely compensate the noise, our approach nevertheless generates maps that are basically sufficient for navigation.

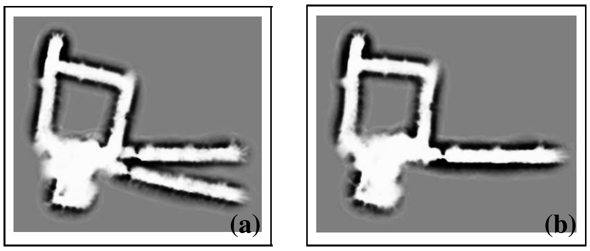

Figure 9: Maps built in Wean Hall of Carnegie Mellon University, (a) using raw odometry, and (b) using our new algorithm. These maps are comparable to those generated by our previous EM method, but without reliance on manually labeled reference positions.

Figure 9 shows similar results for the second environment. Here the robot is asked to map a cyclic environment. Odometric error induces an angular error of approximately 25°. The map shown in Figure 9b is highly accurate, demonstrating that our approach can handle circular environments.

To evaluate our approach quantitatively, we again used a

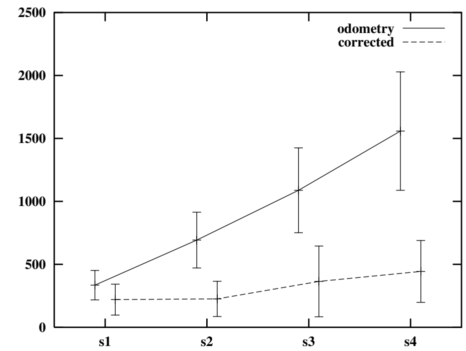

Figure 10: Square error of manually determined reference positions before and after mapping for different levels of random noise added to the odometry data. The error bars represent the 95% confidence interval.

data file which was corrected using a CAD-map of the environment. From this data file we chose several data points and computed the relative distances of the corresponding x-y-locations in the map. We then added noise to the data and applied our mapping system to compute a map. After mapping we measured the deviation of these relative distance from those in the reference map and computed the overall error consisting of the sum of squared differences. Figure 10 shows the average error before (top solid curve) and after (bottom dashed graph) mapping, for different levels of odometric noise. As is to be expected, mapping significantly reduces the error in x-y-space.

These experiments highlight some of the maps we have built over the last months. They are not as large as previously published maps (see e.g., [TFB98, TGF+98]), but they do not rely on high-accuracy laser sensors, and they do not require pre-defined landmarks. The results presented here demonstrate that the basic EM approach can be applied to a much wider array of scenarios than previously suspected.

6 Conclusion

This paper has presented a statistical method for map learning for robots equipped with low-resolution range finders such as sonars. Our approach builds on a previously developed method which uses EM for searching the most likely map. It uses a new two-layered representation of maps during EM. This way, it is able to operate on raw proximity data and does not require any pre-defined landmarks. Maps are learned in two stages: first, small, local maps are learned under the assumption that odometry is locally correct. EM is then applied to integrate these local maps, using Markov localization and the layered representation of maps. To avoid local maxima, a softmax function is applied during the M-step, and deterministic annealing is used to slowly move from density estimation to likelihood maximization. Experimental results demonstrate that our approach is well suited for constructing maps using sensors as inaccurate as sonars, without relying on human assistance.

Acknowledgment

The authors would like to thank Thomas Hofmann and various members of CMU's Robot Learning Lab for useful suggestions and comments.

This research is sponsored in part by NSF (CAREER Award IIS-9876136) and DARPA via TACOM (contract number DAAE07-98-C-L032), and Rome Labs (contract number F30602-98-2-0137), which is gratefully acknowledged. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of NSF, DARPA, TACOM, Rome Labs, or the United States Government.

References

[BAC+98] 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 *Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[BDFC98] W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98)*, 1998.

[BFHS96] 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*. AAAI Press/MIT Press, August 1996.

[Cho96] H. Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, California Institute of Technology, 1996.

[CKK95] 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.

[CL85] 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.

[CW90] I.J. Cox and G.T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[DLR77] A.P. Dempster, A.N. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[Elf87] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.

[Elf89] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.

[FBT98] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.

[FBTC98] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[Fox98] D. Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. PhD thesis, Dept of Computer Science, Univ. of Bonn, Germany, 1998.

[GBFK98] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98)*, 1998.

[GN97] J.-S. Gutmann and B. Nebel. Navigation mobiler Roboter mit Laserscans. In *Autonome Mobile Systeme*. Springer Verlag, Berlin, 1997. In German.

[HPJ99] T. Hofmann, J. Puzicha, and M. Jordan. Learning from dyadic data. In *Advances in Neural Information Processing Systems 11 (NIPS)*, Cambridge, MA, 1999. MIT Press.

[JM98] H. Jans and Ch. Matenar. Korrektureverfahren zum verbesserten Kartenbau mobiler Roboter. Master's thesis, Dept. of Computer Science, Univ. of Bonn, Germany, 1998. In German.

[KB91] 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.

[KBM98] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *AI-based Mobile Robots: Case studies of successful robot systems*, Cambridge, MA, 1998. MIT Press.

[KCK96] 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.

[KS96] 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.

[LM97] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[LM98] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 1998. to appear.

[Mat90] 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.

[MK97] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, New York, 1997.

[Mor88] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.

[Now91] S. J. Nowlan. *Soft competitive adaption: neural network learning based on fitting statistical mixtures*. PhD thesis, Carnegie Mellon University, 1991.

[NPB95] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, Summer 1995.

[Ren93] 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.

[RJ86] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. In *IEEE ASSP Magazine*, 1986.

[Ros98] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, D, November 1998.

[Sha98] H. Shatkay. *Learning Models for Robot Navigation*. PhD thesis, Computer Science Department, Brown University, Providence, RI, 1998.

[SK95] 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.

[SK97] H Shatkay and L. Kaelbling. Learning topological maps with weak local odometric information. In *Proceedings of IJCAI-97*. IJCAI, Inc., 1997.

[TFB98] 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 appeared in Autonomous Robots 5, 253–271.

[TGF+98] 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 *Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[Thr98] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[YLS+98] B. Yamauchi, P. Langley, A.C. Schultz, J. Grefenstette, and W. Adams. Magellan: An integrated adaptive architecture for mobile robots. Technical Report 98-2, Institute for the Study of Learning and Expertise (ISLE), Palo Alto, CA, May 1998.