Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach

Sebastian Thrun[1] **Jens-Steffen Gutmann**[2] **Dieter Fox**[3] **Wolfram Burgard**[3] **Benjamin J. Kuipers**[4]

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

[2]Institut für Informatik
Universität Freiburg
D-79110 Freiburg, Germany

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

[4]Computer Science Department
University of Texas at Austin
Austin, TX 78712

Abstract

The problem of concurrent mapping and localization has received considerable attention in the mobile robotics community. Existing approaches can largely be grouped into two distinct paradigms: topological and metric. This paper proposes a method that integrates both. It poses the mapping problem as a statistical maximum likelihood problem, and devises an efficient algorithm for search in likelihood space. It presents an novel mapping algorithm that integrates two phases: a topological and a metric mapping phase. The topological mapping phase solves a global position alignment problem between potentially indistinguishable, significant places. The subsequent metric mapping phase produces a fine-grained metric map of the environment in floating-point resolution. The approach is demonstrated empirically to scale up to large, cyclic, and highly ambiguous environments.

Introduction

Over the past two decades, the problem of building maps in indoor environments has received significant attention in the mobile robotics community. The problem of building maps is the problem determining the location of certain entities, such as landmarks or obstacles, in a global frame of reference. To build a map of its environment, a robot must know where it is. Since robot motion is inaccurate, constructing maps of large indoor environments requires a robot to solve an inherent concurrent localization problem.

As of to date, there exist two major paradigms for mobile robot mapping: *Metric and topological*. Approaches in the *metric paradigm* generate fine-grained, metric descriptions of a robot's environment (Moravec 1988; Lu & Milios 1997). Approaches in the *topological paradigm*, on the other hand, generate coarse, graph-like descriptions of environments, where nodes correspond to significant, easy-to-distinguish places or landmarks, and arcs correspond to actions or action-sequences that connect neighboring places (Matarić 1990; Dudek *et al.* 1991). Examples of metric and topological methods are provided towards the end of this paper.

It has long been recognized (Chatila & Laumond 1985; Kuipers & Byun 1991) that either paradigm alone, metric or topological, has significant drawbacks. In principle, topological maps should scale better than metric maps to large-scale environments, because a coarse-grained, graph-structured representation is much more compact than a dense array, and more directly suited to problem-solving algorithms (Kuipers & Byun 1991; Dudek *et al.* 1991). However, purely topological maps have difficulty distinguishing adequately among different places, and have not, in practice, been applied successfully to large environments. Recent progress in metric mapping (Lu & Milios 1997; Thrun 1998) has made it possible to build useful and accurate metric maps of reasonably large-scale environments, but memory and time complexity pose serious problems.

In this paper, we propose and evaluate a new algorithm that integrates the topological and metric approach. We show that both the topological and the metric mapping problems can be solved as different instances of a class of statistical estimation problems, in which a robot seeks to find the most likely map from a set of observations and motion commands. These estimation problems are solved by a variant of the EM algorithm, which is an efficient hill-climbing method for maximum likelihood estimation in high-dimensional spaces. In the context of mapping, EM iterates two alternating steps: a *localization step*, in which the robot is localized using a previously computed map, and a *mapping step*, which computes the most likely map based on the previously pose estimates.

The statistical framework is the foundation for two algorithmic phases, one that builds topological maps and one that builds metric maps. Both components possess orthogonal strengths and weaknesses. The topological map builder is capable of solving global alignment problems that occur in datasets with unbounded odometric error and perceptual ambiguity. Its result, however, is only approximate, partially because it ignores much of the sensor data. The metric approach, on the other hand, builds fine-grained metric maps in floating-point resolution. However, it can only compensate small odometric errors. By integrating both, topological and metric mapping, the algorithm can build high-resolution maps in large and highly ambiguous indoor environments.

Statistical Foundations

This paper poses the problem of mapping as a statistical *maximum likelihood estimation problem* (Thrun, Fox, & Burgard 1998). To generate a map, we assume that a robot is given a stream of data, denoted

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

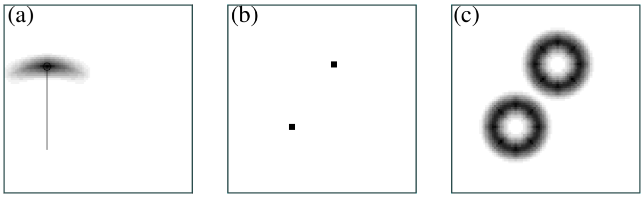

Figure 1: Basic probabilistic models: (a) Robot motion model. Shown here is the accumulated uncertainty after moving 40 meter, starting at a known pose. (b-c) Perceptual model. (b) shows a map with two indistinguishable landmarks, and (c) displays the uncertainty after sensing a landmark in 5 meter distance.

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

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

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

- **The motion model**, denoted $P(\xi'|u,\xi)$, describes the probability that the robot's pose is ξ', if it previously executed action u at pose ξ. Here ξ is used to refer to a pose, that is the x-y-location of a robot together with its heading direction. Figure 1a illustrates the motion model, by showing the probability distribution for ξ' upon executing the action *"go forward 40 meters."* Notice that in these and other diagrams, poses are projected into x-y-space (the heading direction is omitted).
- **The perceptual model**, denoted $P(o|m,\xi)$, models the likelihood of observing o in situations where both the world m and the robot's pose ξ are known. For low-dimensional sensors such as proximity sensors, perceptual models can readily be found in the literature (Burgard *et al.* 1996; Moravec 1988). Figure 1b&c illustrates a perceptual model for a robot that can detect landmarks and that can measure, with some uncertainty, their relative orientations and distances. Figure 1b shows an example map m, in which the dark spots indicate the locations of two indistinguishable landmarks. Figure 1c plots $P(o|m,\xi)$ for different poses ξ, for the specific observation that the robot observed a landmark ahead in five meters distance. The darker a pose, the more likely it is under this observation.

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

The Map Likelihood Function

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Efficient Estimation

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

Fortunately, there exists an efficient and well-understood technique for hill-climbing in likelihood space: the *EM algorithm* (Dempster, Laird, & Rubin 1977), which in the context of Hidden Markov Models is often referred to as *Baum-Welch* or *alpha-beta algorithm* (Rabiner & Juang 1986). EM is a hill-climbing routine in likelihood space, which alternates two steps, an *expectation step* (E-step) and a *maximization step* (M-step). In the context of robot mapping, these steps correspond roughly to a localization step and a mapping step (see also (Koenig & Simmons 1996; Shatkay & Kaelbling 1997)):

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

The E-step corresponds to a localization step with a fixed map, whereas the M-step implements a mapping step which operates under the assumption that the robot's locations (or, more precisely, probabilistic estimates thereof) are known. Iterative application of both rules leads to a refinement of both, the location estimates and the map. Our approach is, thus, a hill-climbing procedure that does not provide a guarantee of global optimality; given the complexity of the problem, however, it is unclear whether a globally optimal routine exists that is computationally tractable.

The E-Step

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

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

Here η is a normalizers that ensure that the left-hand side of Equation (9) sums up to one (see (Thrun, Fox, & Burgard 1998) for a mathematical derivation). Both terms, $\alpha^{(t)}$ and $\beta^{(t)}$, as defined in (9), are computed separately, where the former is computed forward in time and the latter is computed backwards in time. Notice that $\alpha^{(t)}$ and $\beta^{(t)}$ are analogous to those in the alpha-beta algorithm (Rabiner & Juang 1986).

The computation of the α-values is a version of *Markov localization*, which has recently been used with great success for robot localization in *known* environments by various researchers (Burgard *et al.* 1996; Kaelbling, Cassandra, & Kurien 1996; Koenig & Simmons 1996; Nourbakhsh, Powers, & Birchfield 1995; Simmons & Koenig 1995). The β-values add additional knowledge to the robot's pose, typically not captured in Markov-localization. They are, however, essential for revising past belief based on sensor data that was received later in time, which is a necessary prerequisite of building large-scale maps.

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

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

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

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

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

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

Substituting (12) into (11) yields a recursive rule for the computation of all $\alpha^{(t)}$ with boundary condition (10). See (Thrun, Fox, & Burgard 1998) for a more detailed derivation.

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

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

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

The M-Step

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

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

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

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

$$\underset{m}{\text{argmax}}\; P(m|\xi, d) = \left\{ \underset{m_{x,y}}{\text{argmax}} \prod_{t=1}^{T} P(o^{(t)}|\xi^{(t)}, m_{x,y}) \right\}_{x,y} \tag{15}$$

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

The Mapping Algorithm

A key feature of the statistical approach is that it can equally be applied to both topological and metric maps. The mapping algorithm proposed in this paper first constructs coarse-grained topological maps, based on which it then builds detailed metric maps. Both of mapping steps are specialized versions of the same statistical approach described above.

Both mapping steps exhibit orthogonal strengths and weaknesses, arising from differences in representations and the ways sensor data is processed. The topological step is specialized to solve a *global alignment problem*. It can deal with arbitrarily large errors in the robot's odometry; yet it only produces maps with low spatial resolution. The metric step addresses a *local alignment problem*. It assumes that the odometric error is already small, and generates metric maps of the robot's environment with floating-point resolution.

Topological Maps

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

In the topological mapping step, the robot can only observe whether or not it is at a *significant place*. Our definition of significant places follows closely the notion of "distinctive places" in Kuipers's Spatial Semantic Hierarchy (Kuipers & Byun 1991), and the notion of "meetpoints" in Choset's Generalized Voronoi Graphs (Choset 1996). In our experiments, we simulated these methods by manually pressing a button whenever the robot crossed a significant place. To test the most general (and difficult) case, our approach assumes that the significant places are *indistinguishable*. Thus, the robot observes only a single bit of information, namely whether or not it is at a significant place. This deviates from Kuipers and Byun's work, in which places are assumed to be locally distinctive. Nodes in the topological map correspond to *significant places*. Arcs between nodes are created if nodes are adjacent in the data set d. The robot is not told how many significant places exist in its environment; neither does it know whether or not it visited a significant place more than once. Instead, it guesses the number of nodes as a side-effect of maximizing the map likelihood function.

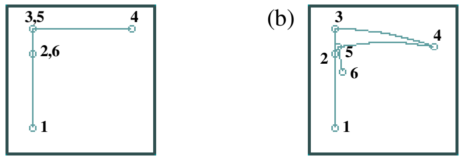

Figure 2: Illustrative example. (a) A path taken by the robot, along which it observed six indistinguishable landmarks (in the order of the numbers). (b) The robot's odometry yields erroneous readings between the third and the fifth landmark observation.

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

Figures 2 and 3 illustrate the topological mapper for an artificial example. Figure 2a shows a path of a robot, in a world that possesses four significant places. The numbers indicate the order in which the robot traverses these places. Figure 2b shows the odometry information. In our example, the robot suffers significant odometric error between its third and its fifth place observation. Based on the odometry information alone, the fifth place appears to be closest to the second, although the robot really went back to the third.

To illustrate the algorithm, let us first consider the situation after the robot made its fifth place observation. Figure 3a shows $\alpha^{(t)}$, $\beta^{(t)}$, and $\alpha^{(t)}\beta^{(t)} \propto P(\xi^{(t)}|d, m)$ after the *first* iteration of EM, for the time steps $t = 2, \ldots, 5$ ($t = 1$ is omitted since the pose $\xi^{(1)}$ is known by definition). Since in the first iteration, there is no map available, the α-values are directly obtained by iteratively "chaining together" the motion model, and the β-values are fairly unspecific. Figure 3b shows the same values two EM iterations later, along with the "corrected" odometry information. The most important density is the one in the upper right corner of Figure 3b, labeled $\alpha^{(5)}$, $\beta^{(5)}$. As can be seen there, the maximum likelihood estimate is incorrect in that it assumes the fifth landmark observation corresponds to the second one (and not the third, which would be correct). This comes at no surprise, as the robot's odometry suggested that the fifth place is much closer to the second, than it is to the third. It is interesting to notice, however, that the algorithm assigns non-zero likelihood to both possibilities, as indicated by the bimodal distribution in the upper right corner-diagram in Figure 3b.

As the robot moves on to the next (=sixth) place observation, the picture changes. The final densities are shown in Figure 3c. Now the robot assigns higher likelihood to the correct topological assignment, and the resulting path (and map) is topologically correct.

This example illustrates two important aspects: First, our algorithm uses future data to revise beliefs backwards in time.

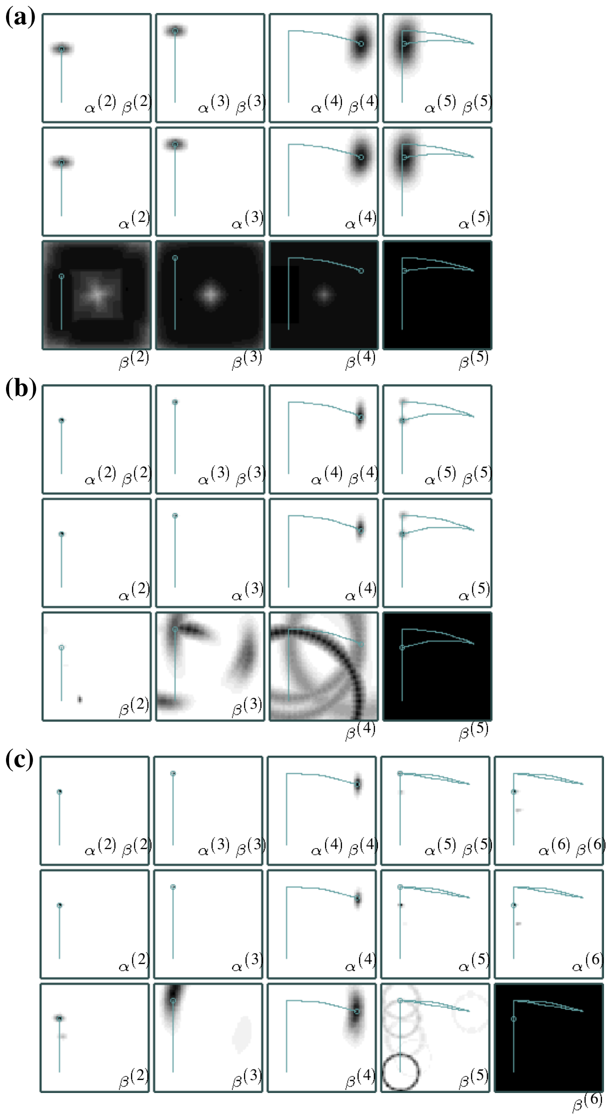

Figure 3: The pose probabilities $P(\xi)$ and their factors α, β for the example. (a) Five-step dataset, first iteration of EM, (b) Five-step dataset, third iteration, (c) Six-step dataset, third iteration.

Second, it considers multiple hypotheses, not just a single one. Both aspects differ from the vast majority of work in the field. We conjecture that both are essential for scaling up mapping to larger environments.

Metric Maps

The metric mapper, which is based on the same statistical framework as the topological mapper, is a modified version of an approach previously proposed in (Gutmann & Schlegel 1996; Lu & Milios 1997).

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

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

The metric mapper is computationally highly efficient. As noticed above, our approach employs linear programming for efficient likelihood maximization. To do so, it approximates the likelihood function linearly, which leads to a closed form solution for all pose variables. Each iteration of the procedure involves solving a linear equation with a $3T$ x $3T$ matrix, which can be computationally challenging for large datasets. This matrix, however, is usually sparse (unless the robot moves only on the spot). Our implementation employs a highly efficient linear programming package that exploits this sparseness. It also employs an efficient grouping mechanisms for reducing the effective number of range scans, prior to constructing the map.

Experimental Results

The mapping algorithm was applied to various datasets obtained in indoor environments, using the RWI B21 robot equipped with 24 sonar sensors and a SICK laser range finder. Figures 4a&b show a dataset collected in our university building, in which circles indicate significant place observations. Here the final odometric error is approximately 24.9 meter.

What makes this dataset challenging is the large circular hallway (60 by 25 meter). When traversing the circle for the first time, the robot cannot exploit landmarks to improve its pose estimates; thus, it accumulates odometric error. Since significant places are indistinguishable, it is difficult to determine the robot's position when the circle is closed for the first time (here the odometric error is larger than 14 meter). Only as the robot proceeds through known territory can it use its perceptual clues to estimate where it is (and was), in order to build a consistent map.

Figure 4c shows the result of topological mapping, including the "corrected" path taken by the robot. While this map is topologically correct, the position estimates are only ap-

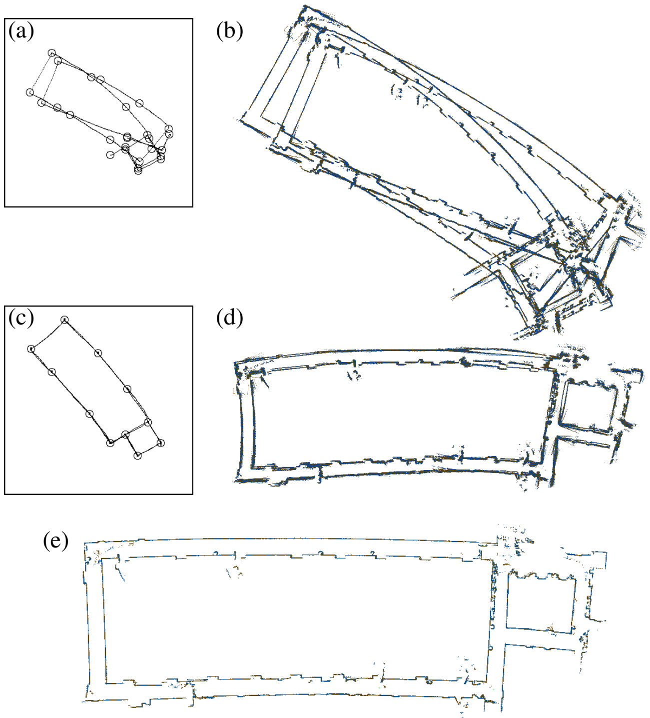

Figure 4: (a) Raw data, obtained in an environment of size 80 by 25 meters. Circles indicate (indistinguishable) places. (b) Sensor map generated from laser range finder data using the raw odometric data. (c) The topological mapper generates a map that correctly describes the topology of the environment. (d) The initial metric map, generated using the pose estimates derived in the topological mapping phase, is still imperfect, although the errors are small. (e) The metric mapper finally generates a highly detailed map.

proximately correct, as demonstrated by Figure 4d. Figure 4e shows the map subsequently generated by the metric mapper. Although the final metric map is slightly bent (which, in fact, appears more plausible given the original data), it is sufficient for our current navigation routines (Gutmann & Nebel 1997; Thrun *et al.* 1998).

Other examples are shown in Figure 5. As can be seen there, our approach produces high-accuracy topological-metric maps in all cases. So far, we did not observe a single failure of the mapping routines. All maps displayed in this paper were generated in less than one hour of computational time, using a 200Mhz Pentium PC for topological mapping and a Sparc Ultra-30 for metric mapping. In all cases, the data collection required less than 20 minutes.

Related Work

Probably the most successful metric approach to date are occupancy grids, which were originally proposed in (Elfes 1989; Moravec 1988; Borenstein & Koren. 1991) and which since have been employed in numerous systems. Other metric approaches, such as those described in (Chatila & Laumond 1985; Cox 1994; Lu & Milios 1997), describe the environment by geometric atoms such as straight lines (walls) or points (range scans). Approaches that fall strictly into the topological paradigm can be found in (Chown, Kaplan, & Kortenkamp 1995;

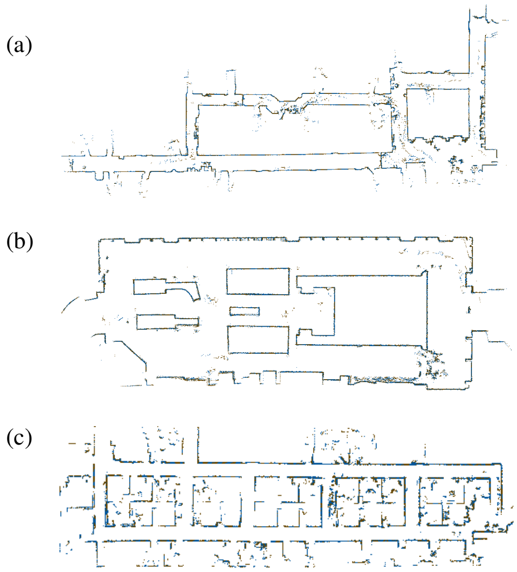

Figure 5: Maps generated in other large-scale environments of sizes (a) 75m, (b) 45m, and (c) 50m. In some of these runs, the cumulative odometric error exceeds 30 meters and 90 degrees.

Kortenkamp & Weymouth 1994; Kuipers & Byun 1991; Matarić 1990; Shatkay & Kaelbling 1997). Some of these approaches do not annotate topological maps with metric information at all; instead, they rely on procedural knowledge for moving from one topological entity to another.

Topological approaches often face severe problems of disambiguating places that look alike. The need to integrate both metric and topological representations has been long recognized (Chatila & Laumond 1985; Kuipers 1978; Thrun 1998), with the current approach just being one example. Other related work is reviewed in (Thrun, Fox, & Burgard 1998).

Our approach differs from most work in the field (see also the surveys in (Thrun 1998; Lu & Milios 1997)) in two important technical aspects. First, robot poses are revised forward *and* backwards in time—as pointed out by Lu and Milios (Lu & Milios 1997), most existing approach do not revise pose estimations backwards in time. Second, by using probabilistic representations, the approach considers multiple hypotheses as to where a robot might have been, which facilitates the recovery from errors. The approaches in (Lu & Milios 1997) and (Shatkay & Kaelbling 1997) are similar in this respect to the one proposed here. Shatkay/Kaelbling proposed to use the alpha-beta algorithm for learning topological maps, based on (Koenig & Simmons 1996), who used the same algorithm for a restricted version of the mapping problem. However, both methods face severe scaling limitations. The approach by Lu/Milios, which in essence constitutes the metric phase of our algorithm, is only able to compensate small odometric error; thus, it alone is insufficient for the problems investigated here. The method by Shatkay/Kaelbling does not represent poses in Cartesian coordinates. It is unclear whether the approach can produce topologically correct maps for the data used in this paper. To the best of our knowledge, no other method has been demonstrated to generate maps of similar size and accuracy.

Conclusion

This paper proposes a statistical approach to building large scale maps in indoor environments. This approach integrates topological and metric representations: It first constructs a coarse-grained topological map, based on which it generates a detailed metric map. The topological approach tackles the global alignment problem, thereby correcting large odometric errors. The metric approach fine-tunes the estimations locally, resulting in maps with floating-point resolution.

Both the topological and the metric approaches are based on the same statistical framework, which treats the problem of concurrent mapping and localization as a maximum likelihood estimation problem. An efficient hill-climbing algorithm was devised to maximize the likelihood function. The algorithm has empirically been validated in cyclic environments of size up to 80 by 25 meters. While strictly speaking, EM is not guaranteed to converge to the global optimum, visual inspection of the results suggest that the most likely map is indeed found in all experiments. In fact, in every single experiment that we ran so far, the approach successfully identified maps that were topologically correct and sufficiently accurate for navigation.

We believe that the results reported here shed new light onto the problem as to how to achieve scalability in mobile robot mapping. First, they demonstrate that both paradigms, metric and topological, can be expressed in the same statistical framework. Under this model, the difference between topological and metric mapping appears to be rather minor. Second, this work illustrates that topological approaches indeed scale up to large and highly ambiguous environments. The environments tested here are difficult in that they possess large cycles, and in that local sensor information is insufficient to disambiguate locations. Our approach managed to produce larger maps than, to the best of our knowledge, any other approach ever has. Third, this paper demonstrates that relatively basic statistical methods are well-suited to solve large-scale mapping problems, suggesting that they might be fit for other high-dimensional state estimation problems in robotics.

Acknowledgment

Partial financial support by Daimler-Benz Research and the Defense Advanced Research Projects Agency (DARPA) via the Airforce Missile System Command under contract number F04701-97-C-0022 is gratefully acknowledged.

References

Borenstein, J., and Koren., Y. 1991. The vector field histogram–fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation* 7(3):278–288.

Burgard, W.; Fox, D.; Hennig, D.; and Schmidt, T. 1996. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of AAAI-96*.

Chatila, R., and Laumond, J.-P. 1985. Position referencing and consistent world modeling for mobile robots. In *Proceedings of ICRA-85*.

Choset, H. 1996. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. Ph.D. Dissertation, Caltech.

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

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

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

Dudek, G.; Jenkin, M.; Milios, E.; and Wilkes, D. 1991. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation* 7(6):859–865.

Elfes, A. 1989. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. Ph.D. Dissertation, CMU.

Gutmann, J.-S., and Nebel, B. 1997. Navigation mobiler roboter mit laserscans. In *Autonome Mobile Systeme*. Springer Verlag.

Gutmann, J.-S., and Schlegel, C. 1996. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*.

Kaelbling, L.; Cassandra, A.; and Kurien, J. 1996. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of IROS-96*.

Koenig, S., and Simmons, R. 1996. Passive distance learning for robot navigation. In *Proceedings ICML-96*.

Kortenkamp, D., and Weymouth, T. 1994. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of AAI-96*.

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

Kuipers, B. J. 1978. Modeling spatial knowledge. *Cognitive Science* 2:129–153.

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

Matarić, M. J. 1990. A distributed model for mobile robot environment-learning and navigation. Master's thesis, MIT.

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

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

Rabiner, L., and Juang, B. 1986. An introduction to hidden markov models. In *IEEE ASSP Magazine*.

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

Simmons, R., and Koenig, S. 1995. Probabilistic robot navigation in partially observable environments. In *Proceedings of IJCAI-95*.

Thrun, S. et al. 1998. Map learning and high-speed navigation in RHINO. In Kortenkamp, D.; Bonassi, R.; and Murphy, R., eds., *AI-based Mobile Robots: Case studies of successful robot systems*. Cambridge, MA: MIT Press.

Thrun, S.; Fox, D.; and Burgard, W. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* and *Autonomous Robots* (joint issue). To appear.

Thrun, S. 1998. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*. To appear.