Proceedings of the 1999 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

Probabilistic Mapping of Unexpected Objects by a Mobile Robot *

Frank Schönherr and **Joachim Hertzberg**†
GMD, AiS.ARC
Schloß Birlinghoven, 53754 Sankt Augustin, Germany
[schoenherr | hertzberg]@gmd.de

Wolfram Burgard
University of Bonn, Computer Science Dept. III,
Römerstr. 164, 53117 Bonn, Germany
wolfram@cs.uni-bonn.de

Abstract

Map learning methods are generally designed to learn from scratch and start with zero knowledge about the state of the world. In this paper, we present a technique for extending a given metric map of the environment by objects of a known type, where localization and perception of the robot is allowed to be uncertain. The advantage of our approach is that it allows the robot to estimate its own position in the given outline of the environment and thus to estimate the position of the objects not contained in the map. The method relies on partially observable Markov decision processes as well as on the Baum-Welch algorithm. It has been implemented and evaluated in several simulation experiments and also in a real-world sewage pipe network. The experimental results demonstrate that our approach can efficiently and accurately estimate the position of unexpected objects. Because of the probabilistic nature of the underlying techniques, our method can deal with noisy sensors as well as with large odometry errors which generally occur when deploying a robot in a sewerage pipe system.

1 Introduction

To perform its tasks efficiently, a mobile robot needs a model of its environment. Using a map, however, requires methods for acquiring and maintaining it as well as techniques for estimating the robot's position in it.

* This work is part of the EC TMR network VIRGO (ERBFMRXCT960049), in which both GMD and Bonn University are nodes. GMD is also supported by the national joint BMBF project MAKRO (02-WK9702/4), the other partners being rhenag, Köln; FZI, Karlsruhe; and Inspector Systems, Rödermark. The support is gratefully acknowledged.

† Hertzberg is also affiliated with Bonn University as a *Privatdozent* (external lecturer).

Two broad map types can be distinguished according to the form of representation. First, metric approaches represent the environment geometry more or less directly. A variant of metric maps, which has proved useful for robot navigation, are occupancy probability maps that decompose the robot environment into a set of grid cells and estimate for each of them the probability that the cell is occupied [6, 13, 12, 19]. Second, topological approaches use a graph-like representation, in which the nodes correspond to places of interest, such as rooms or junctions, and the arcs between these nodes represent a connectivity relation [11, 18, 8].

Several techniques for computing models of the environment from sensor data have been proposed [6, 13, 12, 19]. Most of these approaches are designed to learn maps from scratch, i.e., start with absolute uncertainty about the state of the robot's environment. However, in many practical applications such as surveillance, transportation, or exploration, prior knowledge about the environment structure can be exploited and allows the robot to perform more efficiently than in the case of absolute prior uncertainty.

We present a probabilistic method for extending a given metric map by perceived stationary objects of known types. Data from a simulation as well as experiments in a sewerage pipe network are presented that demonstrate the feasibility of the technique. The method maintains probability distributions about where the objects might be. These distributions are estimated using the Baum-Welch (BW) algorithm [15], which is known to exploit given observations optimally in the sense of yielding models that maximize the observation probabilities under a given set of robot actions.

The BW algorithm has been used before in robot control for map building from scratch [18, 17, 20]. Our method, in contrast, exploits a prior map, where the only change to be made is entering objects of prior known types at a priori unknown places. Using the prior map allows the position of a perceived object to be estimated even in the presence of extreme odometry errors such as occuring when mobile robots operate in sewers [7]. Compared to existing variants of Markov localization [14, 5, 18, 3, 8], our technique extends localization by allowing to estimate past positions.

The next section of this paper recapitulates basic concepts of Markov localization and the BW algorithm. Then, we describe how these techniques are applied to estimate the position of unmapped objects. After that, we illustrate the applicability of our approach in a typical office environment and in a sewer system with its large odometry errors.

2 Markov Localization and the Baum-Welch Algorithm

This section briefly recapitulates basic concepts of POMDPs and of the BW algorithm, as far as they are relevant for this paper. It may be safely skipped by anyone who is familiar with these.

Partially observable Markov decision processes (POMDPs) [4, 18] are used for localizing autonomous mobile robots under unreliability of sensor readings and of motion. The resulting positional uncertainty is represented as a probability distribution, or *belief state,* over the set of possible discrete positions, or *states.* In navigation according to a topological map, a state corresponds to a robot position at one of the landmarks in the map; in metric maps, larger areas devoid of landmarks are discretized by overlaying a grid, each grid cell inducing a state.

An *action* leads the robot from a state to a state. It is an abstraction from a possibly complicated structure of concrete robot control commands. As assumed, an action may yield an unintended result, such as overshooting in a turn or drifting past a landmark steered for. Estimated a priori probabilities for these faults have to be given in advance. Then, the result of an action executed in a state is described by a probability distribution over possible successor states: Given a belief state $B = [b^1 \dots b^n]$ induced by a set of states $S = \{s^1, \dots, s^n\}$, the successor belief state B' after an action a is computed component-wise for all i by $B' = \left[\Pr(s^i|a, B)\right]^i$.(We assume that all actions are defined for all states, so no normalization is necessary.)

An *observation* is abstract in the same spirit: It represents the type(s) of object(s) that the robot perceives at a certain position. *Object* is understood in a broad sense here; an object may be a landmark as occuring in a map, some a priori known world feature of a priori unknown occurrence, such as a damage in a wall (later, we will attempt to map these objects), or a spatial situation, such as a position in the middle of a corridor, which is remarkable for the absence of other distinguishing features. As assumed, mis-perceptions may occur, such as overlooking an object or mis-classifying it; again, the respective a priori probabilities must be given. The result of an observation in a state is a probability distribution over states; accordingly, an observation o maps a belief state B into a belief state $B' = \left[\sigma b^i \Pr(o|s^i)\right]^i$, where $\sigma = 1/\sum_i b^i \Pr(o|s^i)$ is a scaling factor.

Now assume that actions and observations are executed in *observe-then-act* pairs $\langle o, a\rangle$. Consider a *trace* of length T, i.e., a trace consisting of $T-1$ such pairs with an additional observation o_T at the end. Using the above definitions, we can obviously compute the resulting belief state B_T, doing *forward localization.* In particular, we have the probability of being in a given state s at t (for $1 \leq t \leq T$):

$$\begin{aligned} \alpha_t(s) &= \Pr(s|o_{1\dots t}, a_{1\dots t-1}) \qquad (1)\\ &= \sigma_t \Pr(o_t|s) \sum_{s' \in S} \alpha_{t-1}(s') \Pr(s|a_{t-1}, s') \end{aligned}$$

(σ_t is a scaling factor defined in analogy to σ above.)

Given a trace of length T, we can also perform *backward localization:* Compute the belief state associated with a past point in time t. The probability of having been at s at t is

$$\beta_t(s) = \Pr(s|o_{t+1\dots T}, a_{t+1\dots T-1}) \qquad (2)$$

Putting forward and backward localization together, we can use a trace of length T for computing the probability of having been at position s at time t (where $1 \leq t \leq T$):

$$\begin{aligned} \gamma_t(s) &= \Pr(s|o_{1\dots T}, a_{1\dots T-1}) \qquad (3)\\ &= \alpha_t(s)\beta_t(s)\sigma_t \end{aligned}$$

To denote the probability of transiting from position s to s' at t within a trace, we use

$$\gamma_t(s, s') = \Pr(s, s'|o_{1\dots T}, a_{1\dots T-1}) \qquad (4)$$

Based on these local transition models $\gamma_t(s, s')$, it is possible to recompute the transition probabilities according to the trace, i.e., to not only refine the belief states, but also adapt the underlying POMDP model of the robot's functioning in the environment. The BW algorithm [15] in fact does both.

Model update is done in the regular way; we refer to [10] for the exact procedure. For our work, we restrict the model update to changing only the observation probabilities, and keeping fixed the transition probabilities. The reason lies in our current target application domain: Local and temporal motion conditions (dirt, material, weather effects) in parts of a sewer can be dramatically different for a sewer robot. In consequence, positions may exist in the sewer that provide very atypical motion conditions. A transition model averaging over a long trace would practically exclude to localize the robot at such an outlier position. This assumption is debatable. The general method presented in this paper does in no way require it.

The computational complexity for updating all belief states is $O(|S|^2T)$ in one iteration of the BW algorithm. Practically, computation can be saved by exploiting structural properties of the environment, such as symmetries among map areas.

3 Mapping Unexpected Objects

This is the general idea of our method for entering objects into a metric map even in the presence of large odometry error: We use a grid-based representation; the aim is to find the correct grid cell for entering the object. Navigation of the robot is modeled in a POMDP with grid cells corresponding to the state set. That means, whenever an object U is perceived unexpectedly, the recent probability distribution over states yields a first clue about its position. Conceptually, the whole "life history" trace of the robot is recorded; the BW algorithm is used to compute from it a maximally likely POMDP model for explaining the past actions and observations. (Practically, relatively short traces should suffice, with the rest of the robot history encoded in the form of an intermediate version of the map. We will not go into that here, but note that the optimality of the BW algorithm is theoretically lost then.)

The process is iterated for the continuation of the robot's travel through its environment. Over the time, and particularly so if an updated trace includes more passes along U, its position estimation will grow more reliable. In addition to this standard behavior of the BW algorithm, we wash out possible positions of very low probability by maintaining a threshold for the minimum probability of positions to be considered that starts out very low and is increased with every iteration of the algorithm. This helps γ converge towards unique position estimations for U.

We now turn to describing this method technically. For ease of presentation, we assume there is just one type U of unexpected objects for mapping, such as fire extinguishers for an office environment, or inlets in sewers. Of this object type, the environment may contain an arbitrary, previously unknown amount of instances. The generalization to handling more than one object type is straightforward.

We start out with an initial metric map M_0, which remains constant and available all the time. This map is assumed to be correct in the sense that all landmarks included are in fact present in the world, observable, and mapped at the right location. Note that a *landmark* here is taken to mean a distinguishable object from a fixed set of classes; pragmatically, a navigation map should contain only relatively few of them. For example, we use shafts as landmarks in sewers, and doors in office floors. The map is overlayed with a grid of appropriate, constant cell size. The initial POMDP environment model is given, with states corresponding to grid cells.

Unexpected objects must be different from landmarks. By definition, they may be observed at any position. To shield initially localization from the observation of such an object, we add to the initial POMDP a schema for all (initially unknown) unexpected objects U that $\Pr(o = U|s) = 1.0$ for all s.

Given a trace of length T, we use the BW algorithm to maintain an updated version M of the original map M_0 that includes for each position s an expectation of the existence of an object U. This expectation is defined as the ratio of the sum of probabilities of being at s after any t steps and observing U, and the sum of probabilities of all visits at s. The ratio compensates for differences in frequency of visiting different states s. In particular, for T,

$$\mathrm{E}_U(s) = \frac{\sum_{t=1}^{T} \gamma_t(s) \text{ such that } o_t = U}{\sum_{t=1}^{T} \gamma_t(s)} \tag{5}$$

Practically, observations are not always definite; we have used a pseudo-observation *dontknow* in these cases. It makes sense then to sum up in the denominator only those observations that differ from *dontknow*.

The observation probablities in a given position have to change accordingly. Note that the ground map M_0 and the recent map M have the same set of states; they may differ only in what is expected to be observed in the same state. For an observation o using the map M, change of observation probabilities, then, is modeled by weighting U with E_U, and with $1 - E_U$ the observation that would have been expected according to the ground map M_0:

$$\Pr(o|s) = E_U(s)\Pr(o|s\text{ in }M)+(1-E_U(s))\Pr(o|s\text{ in }M_0) \quad (6)$$

If odometry is *really* bad, as we have assumed for this work, then the position estimation in Markov localization grows "shallow" after only short travel distances. Accordingly, the position estimations for a newly detected object is normally very fuzzy, and remains so over quite some time even if using the γ_t forward-and-backward localization. To speed up this process, we have used a minimum likelihood threshold $0 \leq \epsilon \leq 1$ for the $E_U(s)$ values. ϵ starts out low and is increased after every iteration of the BW algorithm. All $E_U(s) < \epsilon$ are then set to 0.

Handling ϵ is obviously a heuristic matter. In our experiments, we have achieved good results with a start value of 0.01 and increments of 0.01. On the other hand, note that $E_U(s) = 1$ is practically never reached, because the traces are finite and the observations are assumed to be uncertain. In consequence $\epsilon \approx 1$ is dangerous, as it would lead to deleting many correct new map entries. In our experiments, we have achieved good results with constraining $\epsilon \approx 0.7$.

4 Experimental Results

To describe how our method performs, we present results from real-world robot experiments in a sewage pipe network and from simulated runs in an indoor environment. We start with a quantitative description of the odometry error from that the sewage pipe experiments suffer and that we have reproduced in the indoor simulations. It is sewer robotics [7], particularly, autonomous sewer inspection, that has originally motivated this work on probabilistic mapping. Sewer inspection is a huge practical real-world problem (German public sewers are about 400,000 km long, with private ones about twice that size [1]), and the conventional procedure, which must be done regularly by law, costs enormous money every year. Obviously, probabilistic mapping makes sense in other applications as well.

4.1 Odometry errors

Sewer pipes are an extremely demanding environment for odometry. They are curved, sloped, slippery, and dirty. Worse, these conditions vary tremendously within a sewer according to pipe material, wear-out, and direction of travel. Needless to say, GPS signals are inaccessible in a sewer.

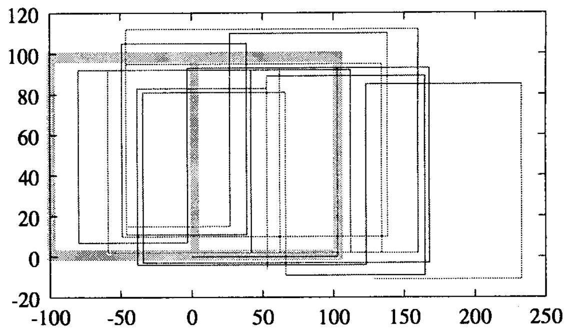


Figure 2: Error of calibrated odometry of the sewer test platform KURT in a tour through an 8-shaped part (grey shade) of the LAOKOON-net. All turns were performed correctly. Distances given in decimeters; the tour started at the (0,0) coordinate.

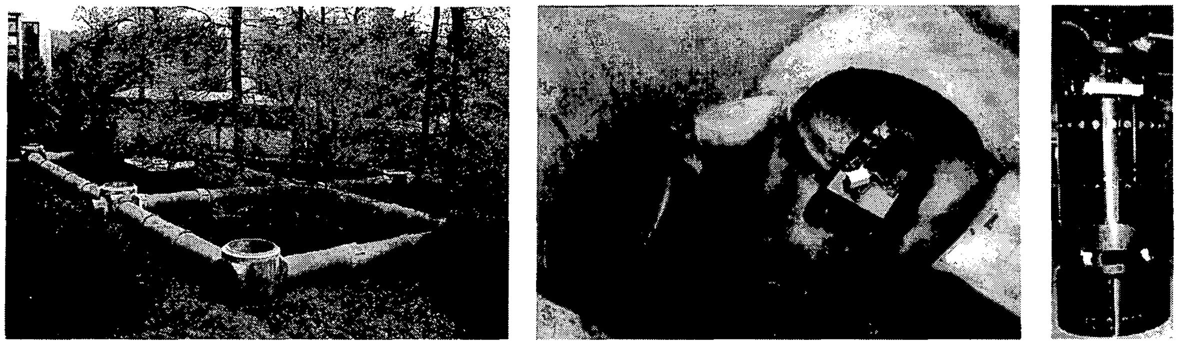
To give an idea of the error quantity, we have recorded test runs of about 1 km overall distance of a sewer robot test platform KURT ([9]; Fig. 1b) in a dry sewer network made of concrete pipes of 60 cm inner diameter (the LAOKOON-net, installed at the GMD site, Fig. 1a). We have calibrated KURT's odometry[1] to perform correctly on the average of this 1 km trip. Applying the so-calibrated odometry to single runs through individual sewer segments between manholes yields significant inter- and intra-segment differences. For example, for one segment (called Seg 1 in the following) with expected odometry value of 10 m, different runs produced an average of 10.7 m, with values varying between 8.9 m and 12.2 m. For another segment, also with 10 m expected, the average was 9.5 m, with values varying between 8.6 m and 9.9 m. The overall standard deviation is 17%, i.e., 17 cm on 1 m. (For details, see [16].)

To give a visual impression of the odometry error, Fig. 2 shows measured distances of a travel through a part of the LAOKOON-net overlayed with the real pipe system. Note that all travel segments are straight and all turns at junctions in the recorded trip are rectangular, because that is how the pipes are built: Drift and non-rectangular turns are physically impossible.[2]

To study odometry errors in environments where

[1] To be precise, KURT's odometry is better desribed as dead-reckoning. The difference is of no importance for our method.

[2] In fact, turning errors may happen for KURT that lead to overshooting by multiples of 90 deg. They are not recorded in the data shown. Moreover, KURT's tilt control maneuvers while passing a pipe, which are a recurrent source of odometry error, are not recorded as such here.



(a) (b) (c)

Figure 1: Pictures of our experimental settings: The LAOKOON net (a), KURT turning in a manhole of the LAOKOON net (b), and the RWI B21 robot RHINO (c)

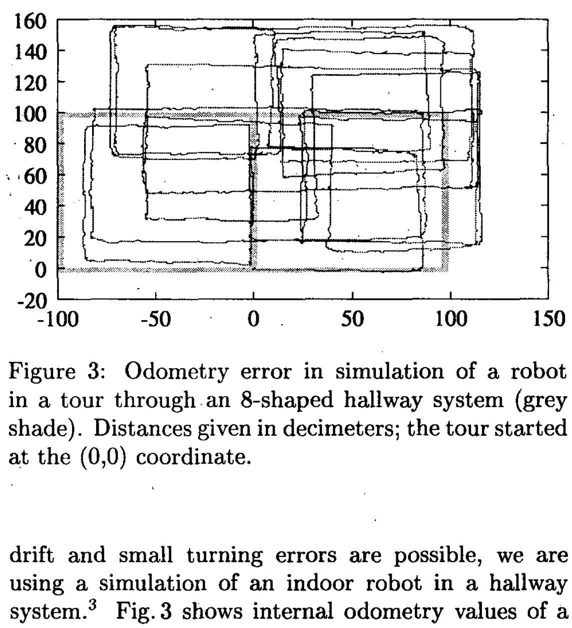


Figure 3: Odometry error in simulation of a robot in a tour through an 8-shaped hallway system (grey shade). Distances given in decimeters; the tour started at the (0,0) coordinate.

drift and small turning errors are possible, we are using a simulation of an indoor robot in a hallway system.[3] Fig. 3 shows internal odometry values of a simulated tour in a hallway system overlayed with the simulated space. The amount and distribution of odometry error used in the simulation is identical to the one reported above for the sewer segment Seg 1; in addition, we have induced slight drift and turning inaccuracies. This typical odometry error for the LAOKOON-net is untypically large for real indoor robots where traction is normally better than in sewer pipes. In consequence, we have run other simulations with smaller errors. We will come back to that later.

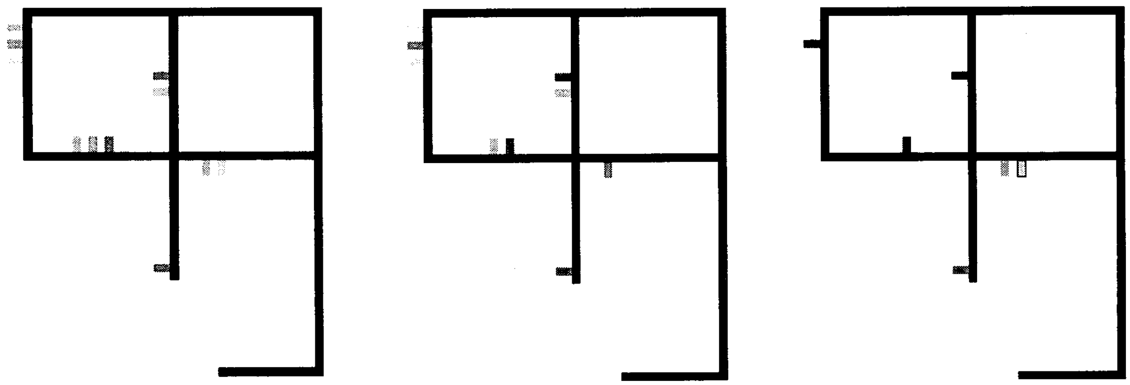
[3]The simulator is the one used to study the RHINO robot ([2]; Fig. 1c).

4.2 Application in sewerage pipes

The method for mapping unexpected objects has been applied to tours of KURT through the LAOKOON-net. The given map included as landmarks all manholes; by law, a manhole must be built in a sewer wherever the direction of the main pipe changes, or wherever two main pipes join. We have used a metric discretization of 1 m, with real pipe length between manholes between 7 and 15 m, most of them 10 m.

As objects to be detected and mapped, we have used small inlets that lead into the mains independent from manholes. Such inlets occur in real sewers to connect single houses or drains from streets; the LAOKOON-net includes five of them, 15 cm in diameter each, which join the mains about the height of the pipe axis. We used an ultrasound transducer on a pan-tilt unit to detect the joins, with KURT's standard inclinometers providing an "artificial horizon" for determining the required tilt value; [16] gives the details. Given this poor sensor, there is a non-0 chance of overlooking joins or of perceiving artefacts.

Fig. 4 presents flashlights of a development of object position estimations over 32 iterations through the BW algorithm, based on sensor data recorded over 8 physical tours through the sewer network. In Fig. 4(a), after the first iteration, all five objects have been perceived plus (compare with the nearly correct positions in Fig. 4(c)) two artefacts—or "real" objects perceived with a significantly faulty position/orientation estimation. After 10 iterations, Fig. 4(b), the probability mass has started to concentrate; the faulty object at the top middle pipe is still a plausible candidate. After 32 iterations, Fig. 4(c), all objects except one have



(a) (b) (c)

Figure 4: Position estimations of observed laterals after 1 iteration (a), 10 iterations (b), and 32 iterations (c) of the BW algorithm. Black/grey boxes represent estimated positions of the relative object or objects, with darkness increasing with probability mass.

just one single plausible position, with no more incorrect objects present; for the ambiguous object, the position slightly framed in the figure has fallen out in the next iteration. All position estimations are then correct, modulo the 1 m grid size. However, note that the procedure is anytime in the sense that after one physical robot tour and one BW iteration, a preliminary result exists, which is only improved by further robot tours and BW runs.

4.3 Application in an indoor simulation

The results just reported for the experiments with KURT in the LAOKOON-net reproduce analogously in the indoor environment simulation, which we have conducted to achieve reproducible results in a standard setting. Note that the possibility of drift or even involuntary turns on a flat office floor, which are excluded in the sewer, provide even more sources of blurring the position estimation than in the sewer experiments.

We have used the simulation to experiment with variants of robot and environment parameters, in particular, with

- the odometry error,
- the probability of involuntary turns/non-turns,
- the frequency and positions of unmapped objects,
- the error rate in object detection.

Most of the overall results are intuitive:

- the method works with all reasonable amounts of odometry and motion control errors;

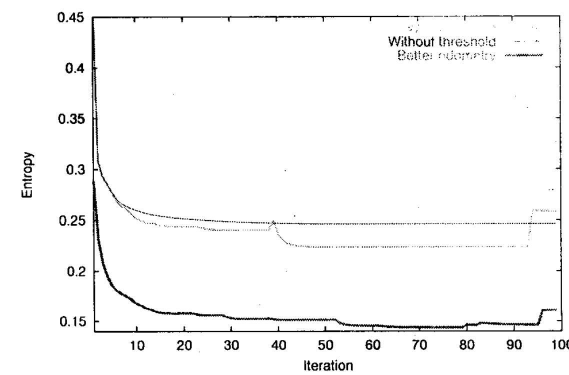


Figure 5: γ entropy values over iterations of the BW algorithm for three different settings in the simulation.

- the more correct odometry, motion control, and object recognition are, the faster do object positions practically converge towards unique grid cells;
- the position estimation for an object close to a landmark converges faster than for an object "in the middle of nowhere";
- whenever the position of an object converges to a unique value, this value is mostly correct;
- an error more likely to occur is that, due to false object classification, an existing object does not get mapped in the end.

For details, we refer to [16]. To get an impression of some common effects of the method, consider the γ

entropy $H(\gamma) = -\sum_{s \in S} \gamma(s) \lg \gamma(s)$ as a measure for the quality of the recent POMDP. The upper curve in Fig. 5 shows how $H(\gamma)$ develops using the sewer-type odometry error with $\epsilon = 0$. Unsurprisingly, the entropy decreases monotonically; invisible from its value alone, the model never arrives at unique positions for detected objects. The middle curve is based on the same odometry error, but uses ϵ as described, i.e., start at 0.01 with 0.01 increments. In principle, the entropy is lower, except for two regions. First, it increases drastically towards the end, corresponding to deleting stabilized (in, fact correct) object positions: As mentioned, ϵ has been pulled too high here. Second, the peak around the 40th run corresponds to deleting (in fact, correctly) an object position; by that, the recent POMDP deteriorates and needs a little time to re-adjust, yielding a considerably better entropy value.

The data for the lowest curve still contains motion errors (90% correctness) and drift, but is based on a more favorable odometry error that is realistic for office navigation and that is used in the RHINO simulator. Accordingly, the localization is much more exact from the start, and the entropy considerably lower. Manipulating the ϵ threshold yields effects as before, albeit in a smoother way.

5 Conclusions and Discussion

In many service applications of autonomous mobile robots, detailed prior knowledge about the environment is available by maps; at least some of these applications, such as inspection and surveillence tasks, involve localizing objects. In addition to being a huge and expensive real-world problem, our target application, autonomous inspection of sewers, certainly is of that type. In consequence, there seems to be a practical problem of mapping at unforseeable locations objects of known types, such as damages. The features of this mapping process blend map learning with navigation: It is more general than navigation in that static objects have to be entered into the map; it is more special than map learning from scratch in that detailed prior knowledge is available and should be exploited; it all is supposed to work under the deficiencies of odometry and sensing that are found in application environments that cannot be tailored to robot operation.

Sewers are a hairy application area in that they induce extreme odometry errors. Our probabilistic approach allows this problem to be handled. As examined in a simulated environment and tested in a real sewage pipe network, we are able to map objects correctly, where the estimation of an object position typically converges fast; precision of initial object localizations as well as speed of position estimation convergence directly benefit if the odometry can be made more accurate. Our work builds upon well-understood ingredients, namely, Markov localization and the Baum-Welch algorithm; yet we are not aware of equally powerful approaches for mapping unexpected objects.

Let us summarize the assumptions and requirements of our method. The objects that can be recorded must be stationary in the sense that they remain at their position long enough to allow several independent observations. Moreover, they must be different from the landmarks included in the prior map, which is assumed to be correct. Using a grid representation of distance, object positions can be recorded exactly only modulo grid size, and given the allowed uncertainties in object detection and odometry, there can be no "hard" guarantee that even unique position estimations are correct—which is normal for probability-based approaches.

Some issues are open. Can the method be changed to delete false entries in the ground map? How about scaling up state spaces such as by covering larger areas or shrinking grid cells? For the latter issue, it seems particularly useful to "localize" the computation of some analog of the BW algorithm to work on just a small record of the immediate past and on a condensed representation of the more distant past. We are not aware of a suitable theoretical framework for that.

All this said, our method seems to have some interesting properties. It can run passively in the background in the sense that data can be recorded incidentally while the robot is working, and can be processed at a suitable time later. The data need not even be collected by one single robot, provided that the action and observation characteristics among several robots are comparable. The method is able to work unsupervised, which is mandatory for environments that—like sewers—are not accessible for humans.

References

[1] ATV (German Association for the Water Environment). *Der Zustand der Kanalisation in der Bundesrepublik Deutschland*, April 1998. http://www.atv.de.

[2] J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, Th. Hoffmann, F.E. Schneider, J. Strikos, and

S. Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, 1995.

[3] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. Fourteenth Natl. Conf. on Art. Intell. (AAAI-96)*, pages 896–901. AAAI Press, 1996.

[4] A. Cassandra, L. Pack Kaelbling, and M. Littman. Acting optimally in partially observable stochastic domains. In *Proc. Twelwth Natl. Conf. on Art. Intell. (AAAI-94)*, pages 1023–1028. AAAI Press, 1994.

[5] A. Cassandra, L. Pack Kaelbling, and J. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of IEEE/RSI Int. Conf. on Intelligent Robots and Systems (IROS-96)*, 1996.

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

[7] J. Hertzberg, Th. Christaller, F. Kirchner, U. Licht, and E. Rome. Sewer robotics. In R. Pfeifer, B. Blumberg, J.-A. Meyer, and S.W. Wilson, editors, *From Animals to Animats 5 – Proc. 5th Intl. Conf. on Simulation of Adaptive Behavior (SAB-98)*, pages 427–436. MIT Press, 1998.

[8] J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proc. 1st Euromicro Workshop on Advanced Mobile Robots* (EUROBOT'96), pages 68–73. IEEE Press, 1996.

[9] F. Kirchner and J. Hertzberg. A prototype study of an autonomous robot platform for sewerage system maintenance. *J. Autonomous Robots*, 4(4):319–331, 1997.

[10] S. Koenig and R. Simmons. Unsupervised learning of probabilistic models for robot navigation. In *Proc. IEEE Conf. on Robotics and Automation (ICRA-96)*, pages 2301–2308, 1996.

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

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

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

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

[15] L.R. Rabiner and B.H. Juang. An introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 1:4–16, January 1986.

[16] F. Schönherr. Ergänzung topologischer Roboternavigationskarten im Falle schwacher Odometrie. Diploma thesis, Univ. Bonn., Computer Sci. Dept., May 1998.

[17] H. Shatkay and L. Pack Kaelbling. Learning topological maps with weak local odometric information. In *Proc. 15th Int. Joint Conf. on Artificial Intelligence (IJCAI-97)*, pages 920–927. Morgan Kaufmann, 1997.

[18] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. IJCAI-95*, pages 1080–1087, 1995.

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

[20] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. Joint issue with J. Autonomous Robots, 5:253-271.