Learning Occupancy Grids with Forward Models

Sebastian Thrun
School of Computer Science
Carnegie Mellon University
Web: http://www.cs.cmu.edu/~thrun

Abstract

This paper presents a new way to acquire occupancy grid maps with mobile robots. Virtually all existing occupancy grid mapping algorithms decompose the high-dimensional mapping problem into a collection of one-dimensional problems, where the occupancy of each grid cell is estimated independently of others. This induces conflicts that can lead to inconsistent maps. This paper shows how to solve the mapping problem in the original, high-dimensional space, thereby maintaining all dependencies between neighboring cells. As a result, maps generated by our approach are often more accurate than those generated using traditional techniques. Our approach relies on a rigorous statistical formulation of the mapping problem using forward models. It employs the expectation maximization algorithm for estimating maps, and a Laplacian approximation to determine uncertainty.

1 Introduction

In the past two decades, occupancy grid maps have become the dominant paradigm for environment modeling in mobile robotics [4, 6, 7]. Occupancy grid maps are spatial representations of robot environments. They represent an environment by a fine-grained, metric grid of variables that reflect the occupancy of the environment. Once acquired, they facilitate various key aspects of mobile robot navigation, such as localization, path planning, collision avoidance, and people finding [1, 5, 9].

Existing occupancy grid mapping algorithms suffer a key problem. They often generate maps that are inconsistent with the data, particularly in cluttered environments and when learned from sonar data. This problem is due to the fact that existing algorithms decompose the high-dimensional mapping problem into many one-dimensional estimation problems—one for each grid cell—which are then tackled independently.

Figure 1 illustrates the problem graphically. In diagram (a), a passing robot might receive the (noise-free) range measurements shown in diagram (b). Inverse sensor models map these beams into probabilistic maps. This is done separately for each grid cell and each beam, as shown in diagrams (c) and (d). Combining both interpretations may yield a map as shown in diagram (e). Obviously, there is a conflict in the overlap region, indicated by the circles in this diagram. The interesting insight is: There exist maps, such as the one in diagram (f), which perfectly explain the sensor measurements without any such conflict. This is because for a sensor reading to be explained, it suffices to assume an obstacle *somewhere* in its measurement cone. Put differently, the fact that cones sweep over multiple grid cells induces important dependencies between neighboring grid cells. A decomposition of the mapping problem into thousands of binary estimation problems—as is common practice in the literature—does not consider these dependencies and therefore may yield suboptimal results. While this consideration uses sonar sensors as motivating example, it is easily extended to other sensor types that may be used for building occupancy maps, such as stereo vision [8].

This paper proposes an alternative approach, which solves the mapping problem in the original, high-dimensional space. In particular, our approach formulates the mapping problem as a maximum likelihood estimation problem in the high-dimensional space of all grid maps. The estimation is carried out using the *expectation maximization algorithm* (in short: EM) [2], which is a popular statistical tool. A key feature of our approach is its reliance on forward probabilistic models of sensors. Forward models describe the physics of sensors. This is in contrast to the literature on occupancy grid mapping, which typically invokes inverse models. To obtain a probabilistic map with uncertainty, we propose to apply a Laplacian approximation. Empirical results in high-noise regimes show that our approach yields considerably more accurate maps.

2 Occupancy Grids with Inverse Models

We begin with a brief review of the classical occupancy grid mapping algorithm [4, 7] to highlight its underlying assumptions. Let m be a map, m_{xy} the occupancy of the grid cell $\langle x, y \rangle$ (a binary random variable), and let $Z =$
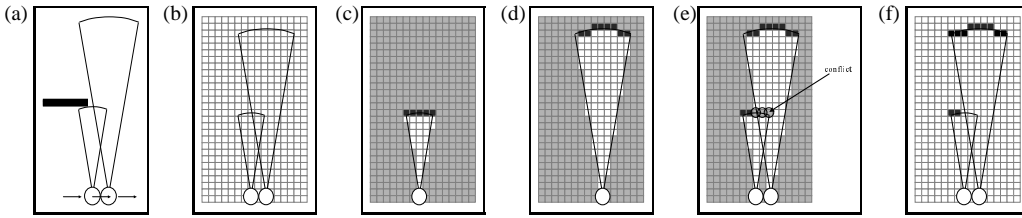
Figure 1. Illustration of the problem with current occupancy grid mapping algorithms. See text.

$\{z_1, \ldots, z_t\}$ denote the sensors measurements from time 1 through time t along with the corresponding robot coordinates. Put probabilistically, occupancy grid maps seek to estimate the posterior probability over maps m given the data: $p(m|z_1, \ldots, z_t)$. However, the high dimensionality of maps m makes it difficult to estimate this posterior directly. Therefore, standard occupancy grid mapping decompose the problem into a collection of one-dimensional estimation problems $p(m_{xy}|z_1, \ldots, z_t)$, one for each grid cell m_{xy}. Calculating these one-dimensional posteriors is tractable.

A common approach is to represent the posterior using *log-odds ratios*, denoted l_{xy}, which is obtained recursively via Bayes rule (see [4, 7] for a derivation):

$$l_{xy}^t = \log \frac{p(m_{xy}|z_1, \ldots, z_t)}{1 - p(m_{xy}|z_1, \ldots, z_t)} \tag{1}$$
$$= \log \frac{p(m_{xy}|z_t)}{1 - p(m_{xy}|z_t)} + \log \frac{1 - p(m_{xy})}{p(m_{xy})} + l_{xy}^{t-1}$$

As this update equation suggests, the basic occupancy grid mapping algorithm requires a *prior for occupancy*, $p(m_{xy})$, and an *inverse sensor model*, $p(m_{xy}|z)$. The latter is called 'inverse' since it maps sensor measurements back to its causes. Occupancy grid maps commonly rely on such inverse models. Notice that inverse models do not take the occupancy of neighboring cells into account when predicting the occupancy of a cell m_{xy}.

The update equation (1) rests on two important assumptions. The first assumption is the *static world assumption*, which states that past sensor readings are conditionally independent given knowledge of the map m:

$$p(z_t|z_1, \ldots, z_{t-1}, m) = p(z_t|m) \tag{2}$$

This assumption is valid in static environments with a non-changing map. It will also be made by our approach. However, by virtue of the grid decomposition, occupancy grid maps make a much stronger assumption: They assume conditional independence given knowledge of an individual grid cell m_{xy}:

$$p(z_t|z_1, \ldots, z_{t-1}, m_{xy}) = p(z_t|m_{xy}) \tag{3}$$

This is an incorrect assumption even in static worlds, since sensor measurements (e.g., sonar cones) sweep over multiple grid cells, all of which have to be known for obtaining independence. Overcoming this assumption is the motivating factor behind this research.

3 Occupancy Grids with Forward Models

3.1 Forward Models

Our alternative approach sidesteps this independence assumption by using *forward models*. Forward models describe the physics of the environment, from causes (occupancy) to effects (measurements). They will be denoted $p(z|m)$, where z is a measurement and m is the map.

Our forward model is a mixture model specific to range finders, such as sonar sensors. Often, a measurement corresponds to an object in the sensor's perceptual cone. Suppose the generic probability that an occupied grid cell may be detected is given by p_{hit}. Sonars typically report the *nearest* such detection. Thus, multiple occupied grid cells in a sonar cone induce a *geometric distribution with parameter* p_{hit}. Our model convolves this geometric distribution with a Gaussian with zero mean and variance σ to account for small measurement errors. A second situation modeled by our forward model is that of failure to detect an object altogether. The resulting measurement is assumed to be random. In our model, failures of this type may happen with probability p_{rand}.

Figure 2 illustrates our forward model graphically. Diagram (a) shows a map with three obstacles (dark squares) and a sonar cone. Two of the obstacles overlap with the cone. With probability p_{rand}, the measurement is random. The probability for detecting the nearer of the two obstacles is $(1-p_{\text{rand}}) \cdot p_{\text{hit}}$; however, with probability $(1-p_{\text{rand}}) \cdot (1-p_{\text{hit}})$ the robot fails to detect the nearer obstacle. It may then detect the second obstacle, which will happen with probability $(1-p_{\text{rand}}) \cdot (1-p_{\text{hit}}) \cdot p_{\text{hit}}$. Otherwise, its sensor will have missed both obstacles and return a max-range reading, which happens with probability $(1-p_{\text{rand}}) \cdot (1-p_{\text{hit}})^2$. Figure 2 (b) shows the corresponding probability density function $p(z|m)$. Clearly visible are two high-probability peaks, which correspond to the two obstacles in the cone. A third peak at the max-range of the sensor models the case that neither obstacle is detected. Obviously, this model is simplistic, and more detailed physical models will likely yield better results. Nevertheless, it is sufficient to yield good results, as illustrated further below in this paper.

To turn these considerations into a concrete mathematical model, let K_i the number of obstacles in the sensor cone of the i-th measurement. Let $D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,K_i}\}$
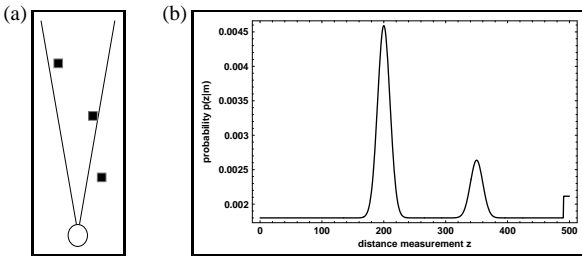
Figure 2. Example of a probabilistic forward model. Diagram (a) depicts a map with three obstacles (dark squares), two of which lie in the sensor cone of a sonar sensor. Diagram (b) depicts the probability of measuring z, for different ranges z. The large peak in diagram (b) corresponds to the nearest obstacle in the cone. The second (smaller) peak corresponds to the obstacle further away, which is only detected if the near obstacle is missed. If neither of the obstacles is detected, the sensor returns a max-range reading, which is indicated by the peak on the far right. Finally, this probability density is mixed with a uniform density that models random measurement errors.

denote the distances to these obstacles computed with ray tracing and ordered in increasing order. To describe the multiple causes of a sensor measurement z_i, it is necessary to introduce new variables, called *correspondence variables*. For each sensor measurement, we define $K_i + 2$ binary variables $C_i = \{c_{i,1}, c_{i,2}, \ldots, c_{i,K_i}, c_{i,K_i+1}, c_{i,*}\}$. Exactly one of those variables is 1; all others must be 0. Each correspondence variable corresponds to exactly one cause: If $c_{i,k}$ is 1 (for $1 \le k \le K_i$), the measurement is caused by the k-th obstacle in the obstacle list D_i. If $c_{i,K+1} = 1$, none of the obstacles in the list were detected. Finally, the random variable $c_{i,*}$ corresponds to the failure case, where a measurement was purely random.

The correspondence variables enable us to describe each possible cause of a measurement separately. If $c_{i,k} = 1$, that is, if the $k - th$ obstacle in D_i caused the measurement z_i, the forward model is a straightforward Gaussian noise variable with variance σ and mean $d_{i,k}$:

$$p(z_i|m, c_{i,k} = 1) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z_i - d_{i,k})^2}{\sigma^2}} \qquad (4)$$

If the measurement is entirely random (failure), we obtain a uniform distribution:

$$p(z_i|m, c_{i,*} = 1) \quad = \quad \frac{1}{z_{\max}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} \quad (5)$$

Finally, the sensor measurement might be caused by missing all obstacles in the cone, hence becomes a max-range reading $z_{\max}$, convolved with Gaussian noise:

$$p(z_i|m, c_{i,K_i+1} = 1) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z_i - z_{\max})^2}{\sigma^2}} \quad (6)$$

Putting all these distributions together gives us the following forward model:

$$p(z_i|m, C_i) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} \qquad (7)$$

$$e^{-\frac{1}{2}\left[c_{i,*} \ln\frac{z_{\max}^2}{2\pi\sigma^2} + \sum_{k=1}^{K_i} c_{i,k}\frac{(z_i - d_{i,k})^2}{\sigma^2} + c_{i,K_i+1}\frac{(z_i - z_{\max})^2}{\sigma^2} \right]}$$

Equation (7) presupposes knowledge of the correspondence, which in practice we do not have. It is therefore convenient to calculate the *joint probability* over measurements and correspondences C_i:

$$p(z_i, C_i|m) \quad = \quad p(z_i|m, C_i)\, p(C_i|m) \qquad (8)$$

Here $p(C_i|m) = p(C_i)$ is the *prior* over correspondences. This prior arises from our consideration above, where we described the different causes of sensor measurements:

$$p(C_i) \quad = \quad \begin{cases} p_{\text{rand}} & \text{if } c_{i,*} = 1 \\ (1-p_{\text{rand}})(1-p_{\text{hit}})^{k-1}p_{\text{hit}} & \text{if } c_{i,k} = 1 \quad (9) \\ (1-p_{\text{rand}})(1-p_{\text{hit}})^k & \text{if } c_{i,K_1+1} = 1 \end{cases}$$

The resulting joint probability $p(z_i, C_i|m)$ implements a simplistic generative model of sonar measurements, factoring in different causes of range measurements through the correspondence variables.

3.2 Expected Data Log-Likelihood

The definition of the forward model enables us to define the data likelihood—which will be maximized to recover the map. Exploiting the static world assumption, the likelihood of the data Z and all correspondences C is given by the product $p(Z, C|m) = \prod_i p(z_i, C_i|m)$, or in logarithmic form:

$$\log p(Z, C|m) \quad = \quad \sum_i \log p(z_i, C_i|m) \qquad (10)$$

In practice, we are interested in maximizing the likelihood of the data D, and not the joint over data D and correspondences C. This is achieved by maximizing the *expected log-likelihood* $E[\log p(Z, C|m)]$, where the expectation is taken over the correspondence variables C. We obtain the expected log-likelihood by substituting Equations (10), (9), (8), and (7) into the expression $E[\log p(Z, C|m)]$, and subsequently exploiting the linearity of the expectation:

$$E[\log p(Z, C|m)] \qquad\qquad\qquad\qquad (11)$$
$$= \sum_i \left[E[\log p(C_i)] + \log\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left[E[c_{i,*}]\ln\frac{z_{\max}^2}{2\pi\sigma^2} \right.\right.$$
$$\left.\left. + \sum_{k=1}^{K_i} E[c_{i,k}]\frac{(z_i - d_{i,k})^2}{\sigma^2} + E[c_{i,K_i+1}]\frac{(z_i - z_{\max})^2}{\sigma^2} \right]\right]$$

Most of the terms in this expression do not depend on the map m, hence can be omitted in the optimization. Maximizing $E[\log p(Z, C|m)]$ is thus equivalent to minimizing

$$\sum_i \sum_{k=1}^{K_i} E[c_{i,k}]\frac{(z_i - d_{i,k})^2}{\sigma^2} \quad \longrightarrow \quad \min \qquad (12)$$

Here the ray tracing distances $d_{i,k}$ and the expectations $E[c_{i,k}]$ depend non-linearly on the map m. Optimization problems of this type are known as maximum likelihood optimization with *latent* variables, where the latent variables are the correspondences C. Unfortunately, we know of no closed form solution for the optimization problem (12).

3.3 Finding Maps via EM

A common way to optimize functions with latent variables is the *expectation maximization algorithm* (in short: EM) [2]. EM is an iterative algorithm that gradually maximizes an expected log-likelihood. The optimization begins with an entirely unoccupied map m. EM then iterates two steps, an *expectation step* (E-step) and a *maximization step* (M-step), thereby gradually increasing the likelihood of the data until a local maximum is reached.

The **E-step** calculates the expected correspondences for a given map m. These expectations are simply the probabilities for each of the possible "causes" of the sensor measurements:

$$E[c_{i,*}] = \eta\, p_{\text{rand}}\, e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} \tag{13}$$

$$E[c_{i,k}] = \eta\, (1-p_{\text{rand}})(1-p_{\text{hit}})^{k-1} p_{\text{hit}}\, e^{-\frac{1}{2}\frac{(z_i - d_{i,k})^2}{\sigma^2}}$$

$$E[c_{i,K_i+1}] = \eta\, (1-p_{\text{rand}})(1-p_{\text{hit}})^{K_i}\, e^{-\frac{1}{2}\frac{(z_i - z_{\max})^2}{\sigma^2}}$$

Here η is a normalizer, which ensures that all expectations in C_i add up to one for each individual value of i. Only the values of $E[c_{i,k}]$ are needed for the optimization, though all of them are required to calculate the normalizer η.

The **M-step** generates a new map for a fixed set of expectations $E[c_{i,k}]$. This is done by minimizing (12). The maximization of this expression is performed by hill climbing in the space of all maps. More specifically, the (discrete) occupancy of individual grid cells is flipped' whenever doing so decreases the target function (12). This discrete search is terminated when no additional flipping can further decrease the target function. It is easy to show that our algorithm terminates after finitely many steps. In practice, we found that maximizing (12) takes only a few seconds on a low-end PC, for the types maps shown in this paper.

Together, the E- and the M-step perform hill climbing in the space of all maps, by maximizing the likelihood of the data. EM may terminate in a non-global maximum, although in our experiments we never observed this to be the case. The resulting map is a maximum likelihood map generated via forward models, which is not subject to the intercell independence assumption that conventional occupancy grid algorithms make.

3.4 Modeling Map Uncertainty

In many applications we would like to know how *certain* we are in the map. This is an important feature of conventional occupancy grid maps, which calculate a posterior probability of occupancy. EM, on the other hand, calculates only a single map and not an entire posterior. Our approach approximates the residual uncertainty by a Laplacian method. The basic idea is to calculate the second derivative (curvature) of the log-likelihood function (11) with respect to changes in occupancy of individual grid cells: $\frac{\partial^2 E[\log p(Z,C|m)]}{\partial m_{xy}^2}$. Since maps are discrete, the derivative is approximated by the discrete difference

$$E[\log p(Z,C|m : m_{xy} = 0)] - E[\log p(Z,C|m : m_{xy} = 1)]$$

The absolute values of this difference approximates the uncertainty in the grid cell. It can be viewed as a discrete sensitivity analysis of the final map. When building an uncertainty map, these values are 'translated' into posterior probabilities using a simple ad-hoc algorithm.

4 Experimental Results

Our approach was successfully applied to learning grid maps using simulated and real robot data. In the real-world experiments, pose estimates we obtained via a separate mapping algorithm described in [10].

Simulation results: Figure 3 shows an example set of sonar measurements of a simulated robot traversing a hallway with an open door. Of particular interest is the number n of sensor readings that correctly detect the open door—instead of being bounced off the door posts. With as little as $n = 1$ door detections, standard occupancy grids fail entirely to map the door, as shown in diagram (b). Our approach, in contrast, succeeds. Diagram (c) shows the result of applying EM (a single map) and diagram (d) is obtained by adding uncertainty extracted via the Laplacian approach. The same result still holds for $n = 3$, as visualized in diagrams (e) through (g). Only for values of n as large as $n = 16$ does the sensor evidence suffice to show an open door in the regular occupancy grid mapping approach, as shown in diagrams (h). It is interesting to note that while the standard occupancy grid mapping algorithm uses probabilistic means to accommodate conflicting sensor interpretations, there are no such conflicts in our approach. All data sets are perfectly explained by any of the maximum likelihood maps shown in Figure 3 (c), (f), or (i). In addition, the standard occupancy grid maps exhibit stripes in the map that are almost perpendicular to the wall. These stripes result from measurements that hit the wall at steep angles.

Real world results: Experiments with two real-world data sets, collected with a RWI B21 robot in an office environment, validate the findings of our simulation experiments. Results are shown in Figures 4 and 5. Figure 4 compares the standard occupancy grid mapping algorithm (diagram (a)) to the one using forward models (diagram (b)). Our approach correctly identifies two narrow doors, which are missed by the standard approach. Several dots in the Laplacian map stem from the fact that a person walked by the robot during data collection. This highlights one of
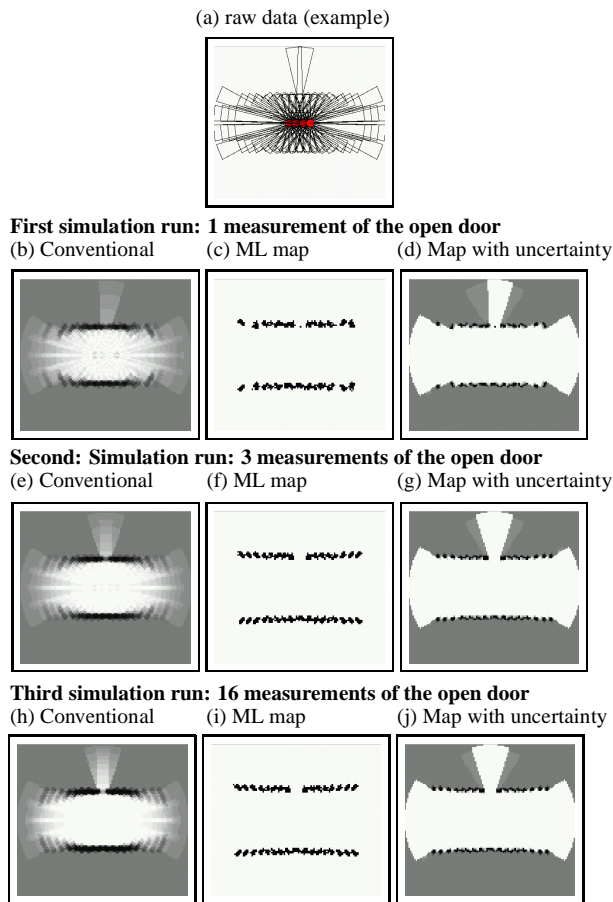
(a) raw data (example)

First simulation run: 1 measurement of the open door

(b) Conventional   (c) ML map   (d) Map with uncertainty

Second: Simulation run: 3 measurements of the open door

(e) Conventional   (f) ML map   (g) Map with uncertainty

Third simulation run: 16 measurements of the open door

(h) Conventional   (i) ML map   (j) Map with uncertainty

Figure 3. Simulation results: The robot travels along a corridor with an open door. When only a single measurement detects the open door (diagrams (b) through (d)), conventional occupancy grid maps do not show the open door (diagram (b)), whereas our approach does (diagram (d)). The same is the case for a data set where the open door was detected three times (diagrams (e) through (g)). With 16 measurements of the open door, the regular occupancy map show an open door (diagram (h)).

the nice features of standard occupancy grid maps, which typically do not show traces of people as long as they do not remain at a single location in the majority of measurements. Our approach appears to be more sensitive to such violations of the static world assumption, in its attempt to maximize the likelihood of the data.

Figure 5 (a) shows a robot trace in a 50 meters long corridor, characterized by several open doors on one side and a smooth wall on the other, which causes many false, long measurements. Under our parameter setting, standard occupancy grid maps miss the smooth wall entirely. A different parameter setting (not shown here) identifies large fractions of this wall, but none of the open doors are detected. Our approach generates a better map, where the only visible inaccuracy stems from the person walking behind the robot as the data was collected. Generating maps of the size shown here typically takes a few minutes of processing time on a 850Mhz Pentium PC.
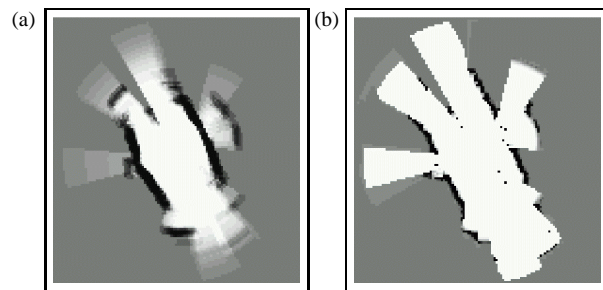
(a)   (b)

Figure 4. Comparison of (a) conventional occupancy grid mapping with (b) our approach using forward models. The open doors are only visible in (b). However, several dot obstacles are shown in (b) that stem from a person walking by the robot during data collection.

5   Discussion

This paper proposed a new approach for generating occupancy grid maps. Our approach relies on physical forward models, instead of the inverse models that are commonly used for learning occupancy grid maps. Instead of breaking down the map learning problem into a multitude of independent binary estimation problems—as done by virtually all existing occupancy grid mapping algorithms—our approach searches maps in the high-dimensional space of all maps. To perform this search, we have adopted the popular EM algorithm to the map estimation problem. Uncertainty maps are finally obtained by applying a Laplacian approximation to the maximum likelihood map found by EM.

The advantage of our new approach is two-fold: First, we conjecture that forward models are more natural to obtain than inverse models, since forward models are descriptive of the physical phenomena that underly the data generation. Second, and more importantly, our approach yields more consistent maps in many situations. This is because it does not rely on a cell-wise independence assumption, and instead treats the mapping problem for what it is: A search in a high dimensional space. The disadvantage of our approach, on the other hand, is an increased sensitivity to changes in non-stationary environments; such environments violate a basic independence assumption that in fact underlies both occupancy grid mapping paradigms. A second disadvantage is a need to go through the data multiple times, which prohibits its use in real-time applications.

Experimental results illustrate that more accurate maps can be built in situations with seemingly conflicting sensor information. Such situations include environments with narrow openings and environments were sonars frequently fail to detect obstacles. These advantages are somewhat set of by our approach's increased sensitivity to environment dynamics, such as a person walking nearby the robot.

There are several opportunities for follow-up research: We believe that our approach can be extended into a concurrent mapping and localization approach [3], by introducing additional hidden variables that correspond to the robot's pose. We conjecture that at least local errors in robot odom-
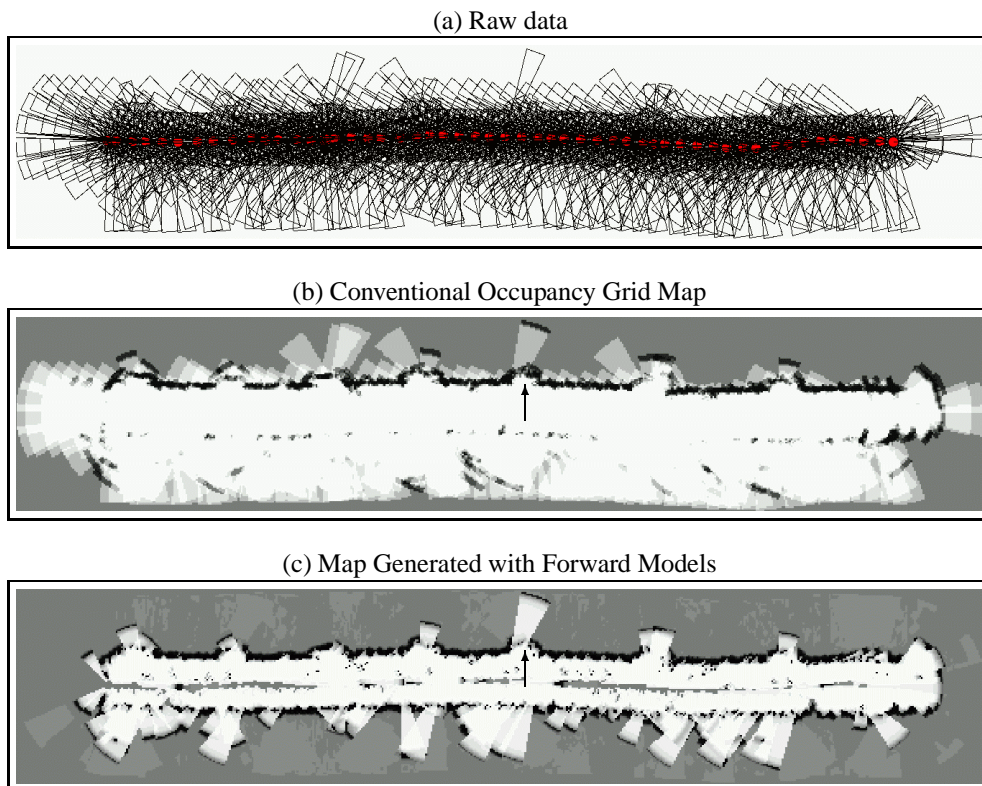
(a) Raw data

(b) Conventional Occupancy Grid Map

(c) Map Generated with Forward Models

Figure 5. Results obtained for a larger data set, shown in (a). While the wall located at the top of each diagram is easy to detect, the wall shown located below the robot is extremely smooth, producing a large number of erroneous readings. The occupancy grid map shown in (b) doesn't model the wall, while at the same time open doors like the one marked by the arrow are falsely modeled as closed. (c) Forward modeling yields an improved model of the bottom wall and also models the door correctly. In addition, this map shows a trace of a person walking behind the robot.

etry can be compensated by such an extension; however, no experimental results are currently available. A different direction of research involves the estimation of environment dynamics. Such dynamics are relatively easily modeled in the forward modeling approach, which describes the physics of the world—as opposed to its inverse. Finally, a worthwhile goal would be the extension to real-time optimization algorithms, so that maps can be generated as the robot moves.

Acknowledgment

References

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

[2] 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.

[3] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*. In Press.

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

[5] K. Konolige and K. Chou. Markov localization using correlation. *Proceedings of IJCAI*, 1999

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

[7] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.

[8] D. Murray and J. Little. Interpreting stereo vision for a mobile robot. *Autonomous Robots*, To Appear.

[9] R. Simmons. Where in the world is Xavier, the robot? *Machine Perception*, 5(1), 1996.

[10] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.