To appear at the *Eighteenth International Conference on Machine Learning*
Williams College, June 28-July 1, 2001

Using EM to Learn 3D Models of Indoor Environments with Mobile Robots

Yufeng Liu YUFENG@ANDREW.CMU.EDU
Center for Automated Learning and Discovery and Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213-3891, USA

Rosemary Emery REMERY@ANDREW.CMU.EDU
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3891, USA

Deepayan Chakrabarti DEEPAY@CS.CMU.EDU
Center for Automated Learning and Discovery, Carnegie Mellon University, Pittsburgh, PA 15213-3891, USA

Wolfram Burgard BURGARD@INFORMATIK.UNI-FREIBURG.DE
Institut für Informatik, Albert-Ludwigs-University Freiburg, D-79110 Freiburg, Germany

Sebastian Thrun THRUN@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891, USA

Abstract

This paper describes an algorithm for generating compact 3D models of indoor environments with mobile robots. Our algorithm employs the expectation maximization algorithm to fit a low-complexity planar model to 3D data collected by range finders and a panoramic camera. The complexity of the model is determined during model fitting, by incrementally adding and removing surfaces. In a final post-processing step, measurements are converted into polygons and projected onto the surface model where possible. Empirical results obtained with a mobile robot illustrate that high-resolution models can be acquired in reasonable time.

1. Introduction

This paper addresses the problem of acquiring volumetric 3D models of indoor environments with mobile robots. A large number of indoor mobile robots rely on environment maps for navigation (Kortenkamp et al., 1998). Almost all existing algorithms for acquiring such maps operate in 2D—despite the fact that robot environments are three-dimensional. Two dimensions are commonly assumed to be sufficient, since the robot is confined to a two-dimensional plane. However, modeling an environment in 3D has two important advantages: First, 3D maps facilitate the disambiguation of different places, since 3D models are much richer than 2D models and hence possess fewer ambiguities. Second, they are of particular interest if the goal of mapping goes beyond robot navigation. 3D models are much better suited for remote users interested in the interior of a building, such as architects, human rescue workers, or fire fighters that would like to familiarize themselves with an environment before entering it.

Moving from 2D to 3D is not just a trivial extension. The most popular paradigm in 2D mapping to date are occupancy grid maps (Elfes, 1989; Moravec, 1988). Occupancy grids represent environments by fine-grained grids. While this is feasible in 2D, in 3D the complexity of these representations pose serious scaling limitations.

This paper proposes an algorithm for recovering low-complexity 3D models from range and camera data, collected by a mobile robot. In particular, our approach fits a probabilistic model that consists of flat surfaces to the data collected by a robot. Such a representation has four advantages over previous non-object representations:

- The resulting maps are less complex, which removes some of the scaling limitations of existing algorithms that are particularly cumbersome in 3D.
- By moving to a low-complexity model, the noise in the resulting maps is reduced—which is a side-effect of the variance reduction by fitting low-complexity models.
- Our approach can utilize prior knowledge about the items in the environment (e.g., number, size and location of walls)
- Finally, an object representation appears to be necessary to track changes in the environment, such as open/close doors and chairs that move.

To identify low-complexity models, the approach presented here uses a variant of the *expectation maximization* (EM) algorithm (Dempster et al., 1977). Our algorithm simultaneously estimates the number of surfaces and their location. Measurements not explained by any surface are retained, enabling us to model non-planar artifacts in the environment as well—but without the benefits of a low-complexity model. The result of the modeling is a low-complexity polygonal model of both structure and texture. The model is represented in VRML, a common virtual reality format.

Our approach rests on two key assumptions. First and foremost, we assume a good estimate of the robot pose is available. The issue of pose estimation (localization) in mapping has been studied extensively in the robotics literature. In all our experiments, we use a real-time algorithm described in (Thrun et al., 2000) to estimate pose; thus, our assumption is not unrealistic at all—but it lets us focus on the 3D modeling aspects of our work. Second, we assume that the environment is largely composed of flat surfaces. The flat surface assumption leads to a convenient close-form solution of the essential steps of our EM algorithm. Flat surfaces are commonly found in indoor environments, specifically in corridors. We also notice that our algorithm retains measurements that cannot be mapped onto any surface and maps them into finer grained polygonal approximations. Hence, the final model may contain non-flat areas.

We present results of mapping an indoor environment using the robot shown in Figure 1. This robot is equipped with a forward-pointed laser range finder for localization during mapping, and an upward-pointed laser range finder and panoramic camera for measuring the structure and texture of the environment in 3D.

2. Models of 3D Environments

2.1 Flat Surface Model

Our assumed model is a finite collection of flat surfaces (walls doors, ceilings). We denote the model by θ, the number of surfaces in the model by J, and each individual surface by θ_j. Hence, we have:

$$\theta \quad = \quad \{\theta_1, \ldots, \theta_J\} \tag{1}$$

Each surface θ_j is a two-dimensional linear manifold in 3D. Following textbook geometry, we describe θ_j by a tuple

$$\theta_j \quad = \quad \langle \alpha_j, \beta_j \rangle \quad \in \quad \Re^3 \times \Re \tag{2}$$

where α_j is the *surface normal vector* and β_j is the distance of the surface to the origin of the global coordinate system. The surface normal α_j is a vector perpendicular to the surface with unit length, that is, $\alpha_j \cdot \alpha_j = 1$.

The surface normal representation facilitates the calculation of distances. Let z be a point in 3D. Then the distance of this point to the surface θ_j is given by

$$|\alpha_j \cdot z - \beta_j| \tag{3}$$

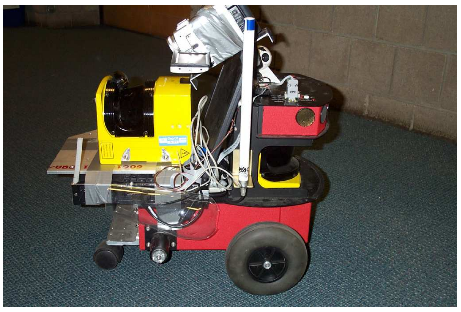

Figure 1. Mobile robot, equipped with two 2D laser range finders and a panoramic camera. The camera uses a panoramic mirror mounted only a few centimeters away from the optical axis of the laser range finder.

where "$\cdot$" denotes the scalar (dot, inner) product of the vectors α and z. The points z on the surface are those for which the following equality holds:

$$\alpha_j \cdot z \quad = \quad \beta_j \tag{4}$$

The reader should notice that our model of surfaces corresponds to planes in 3D, hence they are *unbounded* in size. In practice, however, all objects have finite sizes, and their sizes and locations in 3D are essential components of any model. The advantage of not making size a component of the model is two-fold: First, it simplifies the mathematical derivation of EM, leading to a more efficient algorithm. Second, it enables us to model non-contiguous flat surfaces such as walls separated by doors. It turns out that the size of the surfaces will be obtained almost effortlessly, by mapping sensor measurements onto planes. Hence, we will be content with a model of environments based on 2D planes in 3D.

2.2 Measurements

Our approach assumes that measurements correspond to point obstacles in 3D space. That is, each measurement

$$z_i \quad \in \quad \Re^3 \tag{5}$$

is a 3D coordinate of a point detected by a range measurement. Such point obstacles are easily recovered from range measurements, such as the laser ranger finders used in the experiments below, subject to knowledge of the robot's location. We denote the set of all measurements by

$$Z \quad = \quad \{z_i\} \tag{6}$$

The *measurement model* ties together the world model and the measurements Z. The measurement model is a probabilistic generative model of the measurements given the world:

$$p(z_i | \theta) \tag{7}$$

In our approach, we assume Gaussian measurement noise. In particular, let j be the index of the surface nearest to the measurement z_i. Then the error distribution is given by the following normal distribution with variance parameter σ

$$p(z_i|\theta_j) \quad := \quad \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2}\frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2}} \tag{8}$$

Notice that the log likelihood of this normal distribution is proportional to the squared distance between the measurement z_i and the surface θ_j.

The normal distributed noise is a good model if a range finder succeeds in detecting a flat surface. Sometimes, however, range finders fail to detect the nearest object altogether, or the object detected by a range finder does not correspond to a flat surface. In our approach, we will model such events using a uniform distribution over the entire measurement range:

$$p(z_i|\theta_*) \quad := \quad \begin{cases} 1/z_{\max} & \text{if } 0 \le z_i \le z_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Here θ_* denotes a 'phantom' component of the model θ, which accounts for all measurements not caused by any of the surfaces in θ. The interval $[0; z_{\max}]$ denotes the measurement range of the range finder.

For convenience, we will use the following notation for this uniform distribution, which is reminiscent to a normal distribution (but note the constant exponent!). This notation assumes $z_i \in [0; z_{\max}]$, which is naturally the case for range sensors:

$$p(z_i|\theta_*) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} \tag{10}$$

The reader should quickly verify that for values in $[0; z_{\max}]$, (9) and (10) are indeed identical.

3. Expectation Maximization

Our derivation of the model estimation algorithm begins with the definition of the likelihood function that is being optimized. What follows is the description of EM for this likelihood function, which is tailored towards the perceptual model that combines a Gaussian and a uniform component.

3.1 Log-Likelihood Function

To define the likelihood function, we have to introduce a new set of random variables. These random variables are the *correspondence*, denoted c_{ij} and c_{i*}. Each correspondence variable is a binary random variable. The variable c_{ij} is 1 if and only if the i-th measurement z_i corresponds to the j-th surface in the model, θ_j. Likewise, the correspondence c_{i*} is 1 if and only if the i-th measurement was not caused by any of the surfaces in the model. The correspondence vector of the i-th measurement is given by

$$C_i \quad = \quad \{c_{i*}, c_{i1}, c_{i2}, \ldots, c_{iJ}\} \tag{11}$$

By definition, the correspondences in C_i sum to 1 for all i, since each measurement is caused by exactly one component of the model θ.

If we know the correspondences C_i, we can express the measurement model $p(z_i|\theta)$ as follows

$$p(z_i|C_i,\theta) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2}\left[c_{i*}\ln\frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij}\frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2}\right]} \tag{12}$$

This obviously generalizes our definition in the previous section, since for every measurement z_i only a single correspondence will be 1; all other c-variables will be zero.

Making the correspondence explicit in the measurement model enables us to compute the *joint probability* of a measurement z_i along with its correspondence variables C_i:

$$p(z_i, C_i|\theta) = \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left[c_{i*}\ln\frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij}\frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2}\right]} \tag{13}$$

This formula assumes that all $J+1$ correspondences are equally likely in the absence of measurements; hence the term $J+1$ in the denominator.

Assuming independence in measurement noise, the likelihood of *all* measurements Z and their correspondences $C := \{C_i\}$ is given by

$$p(Z, C|\theta) \tag{14}$$
$$= \prod_i \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left[c_{i*}\ln\frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij}\frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2}\right]}$$

This equation is simply the product of (13) over all measurements z_i.

It is common practice to maximize the log-likelihood instead of the likelihood (14), which is given by

$$\ln p(Z, C|\theta) \quad = \quad \sum_i \ln\frac{1}{(J+1)\sqrt{2\pi\sigma^2}} - \frac{1}{2}c_{i*}\ln\frac{z_{\max}^2}{2\pi\sigma^2} - \frac{1}{2}\sum_j c_{ij}\frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2} \tag{15}$$

The log-likelihood is more convenient for optimization, since it contains a sum where the likelihood contains a product. Maximizing the log-likelihood is equivalent to maximizing the likelihood, since the logarithm is strictly monotonic.

Finally, while the formulas above all compute a joint over model parameters *and* correspondence, all we are actually interested in are the model parameters. The correspondences are only interesting to the extent that they determine the most likely model θ. Therefore, the goal of estimation is to maximize the *expectation* of the log likelihood (15), where the expectation is taken over all correspondences C. This value, denoted $E_C[\ln p(Z, C|\theta)]$, is the expected log likelihood of the data given the model (with the correspondences integrated out!). It is obtained directly from Equation (15):

$$
\begin{aligned}
&E_C[\ln p(Z, C|\theta)] \\
&= \quad E_C\left[\sum_i \ln \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} - \frac{1}{2} c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2}\right. \\
&\qquad \left. -\frac{1}{2}\sum_j c_{ij} \frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2}\right] \qquad (16)
\end{aligned}
$$

Exploiting the linearity of the expectation, we can rewrite (16) as follows:

$$
\begin{aligned}
= \quad &\sum_i \ln \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} - \frac{1}{2} E[c_{i*}] \ln \frac{z_{\max}^2}{2\pi\sigma^2} \\
&-\frac{1}{2}\sum_j E[c_{ij}] \frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2} \qquad (17)
\end{aligned}
$$

Notice that this equation factors in the expectation of the individual correspondences $E[c_{ij}]$ and $E[c_{i*}]$, to calculate the expected log likelihood of the measurements. This equation is the basis for the EM algorithm for maximizing the log-likelihood described in turn.

4. Expectation Maximization

The problem addressed in this section is the maximization of the expected data log-likelihood, as specified in (17). The EM algorithm is a popular method for hill climbing in likelihood space (Dempster et al., 1977) that is directly applicable to this problem.

In essence, EM generates a sequence of models, $\theta^{[0]}, \theta^{[1]}, \theta^{[2]}, \ldots$. Each model improves the log-likelihood of the data over the previous model, until convergence. EM starts with a random model $\theta^{[0]}$. Each new model is obtained by executing two steps: an E-step, where the expectations of the unknown correspondences $E[c_{ij}]$ and $E[c_{i*}]$ are calculated for the n-th model $\theta^{[n]}$, and an M-step, where a new maximum likelihood model $\theta^{[n+1]}$ is computed under these expectations. An important result is that for our planar model with uniform noise, both of these steps can be solved in closed form.

4.1 The E-Step

In the E-step, we are given a model $\theta^{[n]}$ for which we seek to determine the expectations $E[c_{ij}]$ and $E[c_{i*}]$ for all i, j. Bayes rule, applied to the sensor model, gives us a way to calculate the desired expectations (assuming a uniform prior over correspondences):

$$
\begin{aligned}
E[c_{ij}] &= p(c_{ij}|\theta^{[n]}, z_i) \\
&= \frac{p(z_i|\theta^{[n]}, c_{ij}) p(c_{ij}|\theta^{[n]})}{p(z_i|\theta^{[n]})} \\
&= \frac{e^{-\frac{1}{2}\frac{(\alpha_j \cdot z_i - \beta_j)^2}{\sigma^2}}}{e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2}\frac{(\alpha_k \cdot z_i - \beta_k)^2}{\sigma^2}}} \qquad (18)
\end{aligned}
$$

Similarly, we obtain

$$
E[c_{i*}] = \frac{e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}}}{e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2}\frac{(\alpha_k \cdot z_i - \beta_k)^2}{\sigma^2}}} \qquad (19)
$$

Thus, to summarize, in the E-step the expectation that the i-th measurement corresponds to the j-th surface is proportional to the Mahalanobis distance between the surface and the point, with an additional variable in the denominator that accounts for unexplainable phantom measurements.

4.2 The M-Step

In the M-step, we are given the expectations $E[c_{ij}]$ and seek to calculate a model $\theta^{[n+1]}$ that maximizes the expected log-likelihood of the measurements, as given by Equation (17). In other words, we seek surface parameters $\langle \alpha_j, \beta_j \rangle$ that maximize the expected log likelihood of the model.

Obviously, many of the terms in (17) do not depend on the model parameters θ. This allows us to simplify this expression and instead minimize

$$
\sum_i \sum_j E[c_{ij}](\alpha_j \cdot z_i - \beta_j)^2 \qquad (20)
$$

The reader should quickly verify that minimizing (20) is indeed equivalent to maximizing (17). The minimization of (20) is subject to the normality constraint $\alpha_j \cdot \alpha_j = 1$; since otherwise α_j would not be a surface normal. Hence, the M-step is a quadratic optimization problem under equality constraints for some of the variables.

To solve this problem, let us introduce the Lagrange multipliers λ_j for $j = 1, \ldots, J$, and define

$$
L \; := \; \sum_i \sum_j E[c_{ij}](\alpha_j \cdot z_i - \beta_j)^2 + \sum_j \lambda_j \alpha_j \cdot \alpha_j \qquad (21)
$$

Obviously, for each minimum of L, it must be the case that $\frac{\partial L}{\partial \alpha_j} = 0$ and $\frac{\partial L}{\partial \beta_j} = 0$. This leads to the linear system of equalities (first derivatives of L):

$$
\sum_i E[c_{ij}](\alpha_j \cdot z_i - \beta_j) z_i - \lambda_j \alpha_j \; = \; 0 \qquad (22)
$$

$$\sum_i E[c_{ij}](\alpha_j \cdot z_i - \beta_j) \;=\; 0 \tag{23}$$

$$\alpha_j \cdot \alpha_j \;=\; 1 \tag{24}$$

The values of β_j can be calculated from Equations (22) and (23):

$$\beta_j \quad = \quad \frac{\sum_k E[c_{kj}]\alpha_j \cdot z_k}{\sum_k E[c_{kj}]} \tag{25}$$

which, substituted back into (22) gives us

$$\sum_i E[c_{ij}] \left(\alpha_j \cdot z_i - \frac{\sum_k E[c_{kj}]\alpha_j \cdot z_k}{\sum_k E[c_{kj}]} \right) z_i - \lambda_j \alpha_j \quad = \quad 0 \tag{26}$$

This is a set of *linear equations* of the type

$$A_j \cdot \alpha_j \quad = \quad \lambda_j \alpha_j \tag{27}$$

where each A_j is a 3×3 matrix whose elements are as follows:

$$a_{st} = \sum_i E[c_{ij}] z_{is} z_{it} - \frac{\sum_i E[c_{ij}] z_{it} \sum_k E[c_{kj}] z_{ks}}{\sum_k E[c_{kj}]} \tag{28}$$

for $s, t \in \{1, 2, 3\}$.

The solution to our problem of calculating the values of α_j is now the eigenvector of (27) with the smallest eigenvalue. Why the smallest? It is easy to see that each solution of (27) must be an eigenvector of A_j. The two eigenvectors with the largest eigenvalues describe the desired surface. The third eigenvector, which is a normal and orthogonal to the first two eigenvectors, is therefore the desired surface normal. Thus, we now have a solution to the problem of calculating the maximum likelihood model $\theta^{[n+1]}$ under the expectations $E[c_{ij}]$. This completes the derivation of the EM algorithm.

4.3 Starting and Terminating Model Components

The EM algorithm assumes knowledge of the number of surfaces J. In practice, however, J is unknown and has to be estimated as well. The number of surfaces J depends on the environment and vary drastically from environment to environment.

Our approach uses an incremental strategy for introducing new surfaces (increasing J) and terminating other, unneeded surfaces (decreasing J). This step is an *outer loop* to EM: Every time the model complexity J is changed, EM is run for a certain number of time steps (e.g., 20) to find the most likely model given J. As usual in EM, the results of likelihood maximization is sensitive to the initialization of a new model component, which determines the quality of the local maximum found by EM.

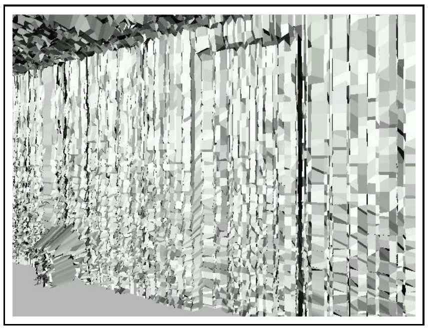

Figure 2. Polygonal model generated from raw data, not using EM. The display without texture shows the level of noise involved, which makes it difficult to separate the door from the nearby wall.

Introducing New Surfaces: New surfaces are introduced as follows: Select a random measurement z_i. Select the two nearest measurements to z_i, where nearness is measured by Euclidean distance in 3D. These three measurements together define a surface, which is added to the model. Notice that our approach of generating new surfaces does not consider how well a measurement z_i is already 'explained' by the existing model—an alternative for starting new model components might be to give preference to measurements z_i distant to all of the surface θ_j in the model. We found our approach to be superior in environments with several nearby surfaces, such as the corridor environment described below, where the wall surface and the door surface are only 7 centimeters apart.

Terminating Unsupported Surfaces: A surface is considered "unsupported" if after convergence of EM, it fails to meet any of the following criteria:

- Insufficient number of measurements: The total expectation for surface θ_j is smaller than a threshold $E_{\min}$:

$$\sum_i E[c_{ij}] \quad < \quad E_{\min} \tag{29}$$

- Insufficient density: Let us define as $Z_j(\theta)$ the set of measurements who select the surface θ_j as their most probable surface:

$$Z_j \quad := \quad \{j : \operatorname*{argmax}_{j'} E[c_{ij'}]\} \tag{30}$$

Then our approach rejects each surface j for which the average distance between measurements in Z_j and their nearest neighbor in Z_j is larger than a threshold $x_{\max}$

$$\frac{1}{|Z_j|} \sum_{z \in Z_j} \min_{z' \in Z_J : z' \neq z} |z' - z| \quad > \quad x_{\max} \tag{31}$$

Figure 3. 3D Model generated (a) from raw sensor data, and (b) using our algorithm, in which 94.6% of all measurements are explained by 7 surfaces. Notice that the model in (b) is much smoother and appears more accurate.

This criterion can be interpreted as a minimum density criterion. It leads to the elimination of surfaces whose supporting points are spread out too far.

Fusing Surfaces: Finally, pairs of nearby surfaces are fused into one new surface if they are too close together. This is necessary to eliminate surfaces that compete for the same set of measurements. Our approach fuses surfaces whose angle is smaller than a threshold, and whose distance is smaller than a threshold at each of the supporting points.

5. Post-Processing

5.1 Smoothing

In a final step, the result is post-processed by a smoothing step. This step brings to bear knowledge that nearby measurements are likely to belong to the same surface. The final assignment of points to surfaces is based on a mixture of the assignments of the neighbors. In particular, if three or more of the nearest neighbors of a measurement z_i are matched to surface j (as most likely assignment), z_i is also assigned to surface j.

The final smoothing improves the quality of the matching in the presence of nearby surfaces. In the testing example described below, the wall and the door surface are only 7 cm apart, whereas the (cumulative) measurement error of the laser range finder is 10 cm. Without the final smoothing step, a number of points on the door would be associated to the wall surface, and vice versa.

We note that there are two possible ways to smooth the assignments: As a prior for the model likelihood, or as a post-smoothing step. In the former case, one would have to augment the log-likelihood function, which, unfortunately, implies that the E-step cannot be solved in closed form. This motivates our choice of making smoothing a post-processing step.

5.2 3D Reconstruction

In a final, mechanical reconstruction process, the measurements z_i are mapped into polygons and the exact size of the surfaces is determined. Our approach exploits the fact that measurements are taken in a sequence. If in two time-adjacent range scans, four measurements are ‘nearby’ (within 50 cm range) they are mapped into a polygon, and the texture data recorded along the way is superimposed to this polygon. The exact size and shape of a surface is then obtained by projecting the measurements (and hence the polygons) onto the surface model θ. See (Thrun et al., 2000) for more detail.

6. Results

Our approach has been evaluated in the context of the DARPA Tactical Mobile Robotics Project (TMR). The specific data set has been acquired inside a university building, using the robot shown in Figure 1. It consists of 168,120

(a) Model complexity (number of surfaces J) (b) Percentage of measurements explained by model

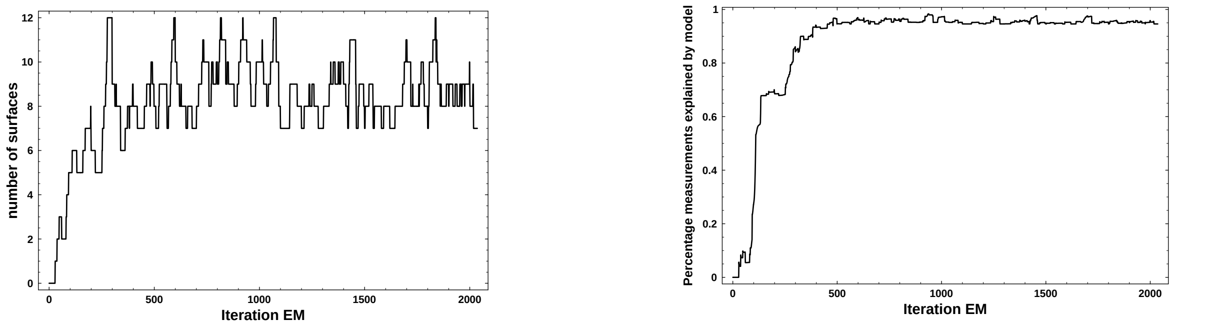

Figure 4. (a) number of surfaces J and (b) percentage of points explained by these surfaces as function of the iterator. A usable model is available after 500 iterations.

individual range measurements and 3,270 camera images, of which our software in a pre-alignment phase extracts 3,220,950 pixels that correspond to range measurements. The data was collected in less than two minutes. Figure 2 shows a detail of this scene without the texture superimposed, to illustrate the level of noise in the raw model. Figure 3a shows three rendered views of a 3D model generated *without* EM. Here the 3D reconstruction step has been applied to the raw data, as described in Section 5.2. The resulting model is extremely rugged due to the noise in the sensor measurements.

Figure 3b depicts views rendered from the resulting low-complexity model. This model consists of $J = 7$ surfaces that account for an 94.6% of all measurements. Clearly, the new model is smoother, which makes various details visible. Notice that the left view shows a trash-bin, which is *not* captured by the surface model. The corresponding measurements are not mapped to any of the surfaces but instead correspond to the phantom component in the sensor model. Notice also that the wall surface is different from the door surface. A small number of measurements in the door surface are erroneously assigned to the wall surface. The poster-board shown in the other columns of Figure 3 highlights the benefits of the EM approach over the unfiltered data.

Figure 4 shows the number of surfaces J and the number of measurements explained by these surfaces, as a function of the iteration. Every 20 steps (in expectation), surfaces are terminated and restarted. After only 500 iterations, the number of surfaces settles around a mean (and variance) of 8.76 ± 1.46, which explains a steady $95.5 \pm 0.006\%$ of all measurements. 2,000 iterations require approximately 20 minutes computation on a low-end PC.

7. Related Work

Building 3D models from robot data has previously been proposed in (Thrun et al., 2000), who describes an online algorithm for pose estimation during mapping using a forward-pointed laser range finder. The 3D maps given there are (vast) collections of local polygons obtained directly from the raw data, without any additional processing or modeling. In our implementation, these algorithms are used for obtaining initial measurement estimates for the approach proposed here. The present work, thus, continues this line of investigation and yields a practical algorithm for finding low-complexity surface models from those raw measurements.

The vast majority of robot modeling research has focused on building maps in 2D. Our approach is reminiscent of an early paper by Chatila and Laumond (Chatila & Laumond, 1985), who proposed to reconstruct low-dimensional line models in 2D from sensor measurements but provided no practical algorithm. Our work is also related to work on line extraction from laser range scans (Lu & Milios, 1998). However, these methods address the two-dimensional case, where lines can be extracted from a single scan. Another common approach to 2D mapping are occupancy grids, which decompose the environment using a fine-grained grid. The resulting models are more complex than the surface models generated here, for which reason occupancy grid maps are commonly only applied in 2D. Recently, Moravec successfully extended his approach to the full three-dimensional case (Moravec & Martin, 1994). However, this approach requires excessive memory and is unable to exploit prior knowledge about the typical shape of indoor features (e.g., our flat surface assumption). Our approach is also related to (Iocchi et al., 2000), which reconstructs planar models of indoor environments using stereo vision, using some manual guidance in the reconstruction process to account for the lack of visible structure in typical indoor environments.

EM has been applied in the context of robot mapping (Shatkay & Kaelbling, 1997; Thrun et al., 1998). These approaches, however, address a different problem: The localization problem in mapping. In their work, the hidden variables correspond to robot locations relative to sensor measurements, and the resulting models are complex. In contrast, the present approach uses EM to recover low-complexity models, where the hidden parameters are correspondence parameters between model components and measurements.

In the area of computer vision, 3D scene reconstruction has been studied by several researchers. Approaches for 3D modeling can be divided into two categories: Approaches that assume knowledge of the pose of the sensors (Allen & Stamos, 2000; Bajcsy et al., 2000; Becker & Bove, 1995; Debevec et al., 1996; Shum et al., 1998), and approaches that do not (Hakim & Boulanger, 1997). Our approach uses mobile robots for data acquisition; hence our approach falls into the second category due to the inherent noise in robot odometry (even after pose estimation). However, unlike the approaches in (Hakim & Boulanger, 1997; Thrun et al., 2000) which generate highly complex models, our focus is on generating low-complexity models that can be rendered in real-time.

The majority of existing systems also requires human input in the 3D modeling process. Here we are interested in fully automated modeling without any human interaction. Our approach is somewhat related to (Iocchi et al., 2000), which reconstructs planar models of indoor environments using stereo vision, using some manual guidance in the reconstruction process to account for the lack of visible structure in typical indoor environments. EM was previously proposed for scene reconstruction in computer vision (Ayer & Sawhney, 1995), but not using robots. The idea of planar layers in the scene, reminiscent of our planar surface model, can also be found in (Baker et al., 1998)—the latter two approaches require exact knowledge of a robot's pose and do not use range sensing. Related work on outdoor terrain modeling can be found in (Cheng et al., 2000).

8. Conclusion

We have presented an algorithm for recovering 3D low-complexity object models from range and camera data collected by mobile robots. The approach combines an algorithm for fitting mixtures of planar surfaces to 3D data, with an algorithm for modifying the complexity the model. In a post-processing step, measurements are transformed into polygons and projected onto the low-dimension model where possible.

Our approach has been successfully applied to generating a low-complexity 3D structural and texture model from robot data in an indoor corridor environment. Future work includes broadening our model to include non-planar objects—which appears to be possible within the statistical framework put forward in this paper.

Acknowledgment

This research is sponsored by DARPA's TMR Program (contract number DAAE07-98-C-L032), DARPA's MARS Program, and DARPA's CoABS Program (contract number F30602-98-2-0137). It is also sponsored by the National Science Foundation (regular grant number IIS-9877033 and CAREER grant number IIS-9876136), all of which is gratefully acknowledged.

References

Allen, P., & Stamos, I. (2000). Integration of range and image sensing for photorealistic 3D modeling. *ICRA-2000*.

Ayer, S., & Sawhney, H. (1995). Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. *ICCV-95*.

Bajcsy, R., Kamberova, G., & Nocera, L. (2000). 3D reconstruction of environments for virtual reconstruction. *WACV-2000*.

Baker, S., Szeliski, R., & Anandan, P. (1998). A layered approach to stereo reconstruction. *CVPR-98*.

Becker, S., & Bove, M. (1995). Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. *SPIE-95*.

Chatila, R., & Laumond, J.-P. (1985). Position referencing and consistent world modeling for mobile robots. *ICRA-85*.

Cheng, Y.-Q., Riseman, E., Wang, X., Collins, R., & Hanson, A. (2000). Three-dimensional reconstruction of points and lines with unknown correspondence across images. *International Journal of Computer Vision*.

Debevec, P., Taylor, C., & Malik, J. (1996). Modeling and rendering architecture from photographs. *SIGGRAPH-96*.

Dempster, A., Laird, A., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*.

Elfes, A. (1989). *Occupancy grids: A probabilistic framework for robot perception and navigation*. Ph.D. Thesis, ECE, CMU.

Hakim, S., & Boulanger, P. (1997). Sensor based creation of indoor virtual environment models. *VSMM-97*.

Kortenkamp, D., Bonasso, R., & Murphy, R. (Eds.). (1998). *AI-based mobile robots: Case studies of successful robot systems*. MIT Press.

Iocchi, L., Konolige, K., & Bajracharya, M. (2000). Visually realistic mapping of a planar environment with stereo. *ISER-2000*.

Lu, F., & Milios, E. (1998). Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, *18*.

Moravec, H., & Martin, M. (1994). Robot navigation by 3D spatial evidence grids. CMU, Internal Report.

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine* Summer 1998.

Shatkay, H., & Kaelbling, L. (1997). Learning topological maps with weak local odometric information. *IJCAI-97*.

Shum, H., Han, M., & Szeliski, R. (1998). Interactive construction of 3D models from panoramic mosaics. *CVPR*-98.

Thrun, S., Burgard, W., & Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *ICRA-2000*.

Thrun, S., Fox, D., & Burgard, W. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, *31*.