

A Real-Time Expectation-Maximization Algorithm for Acquiring Multiplanar Maps of Indoor Environments With Mobile Robots

Sebastian Thrun, Christian Martin, Yufeng Liu, Dirk Hähnel, Rosemary Emery-Montemerlo, Deepayan Chakrabarti, and Wolfram Burgard

Abstract—This paper presents a real-time algorithm for acquiring compact three-dimensional maps of indoor environments, using a mobile robot equipped with range and imaging sensors. Building on previous work on real-time pose estimation during mapping, our approach extends the popular expectation-maximization algorithm to multisurface models, and makes it amenable to real-time execution. Maps acquired by our algorithm consist of compact sets of textured polygons that can be visualized interactively. Experimental results obtained in corridor-type environments illustrate that compact and accurate maps can be acquired in real time and in a fully automated fashion.

Index Terms—Robotic mapping, mobile robots, perception, statistical techniques.

I. INTRODUCTION

THIS PAPER presents a real-time algorithm for generating three-dimensional (3-D) maps of indoor environments from range and camera measurements acquired by a mobile robot. A large number of indoor mobile robots rely on environment maps for navigation [3]. Most all existing algorithms for acquiring such maps operate in two dimensions (2-D). 2-D maps may appear sufficient for navigation, given that most indoor mobile robots are confined to 2-D planes. However, modeling an environment in 3-D has two important advantages: first, 3-D maps facilitate the disambiguation of different places, since 3-D models are richer than 2-D models; second, 3-D maps

are better suited for users interested in the interior of a building, such as architects or human rescue workers that would like to familiarize themselves with an environment before entering it. For these and other reasons, modeling buildings in 3-D has been a long-standing goal of researchers in computer vision [4]–[7]. Generating such maps with robots would make it possible to acquire maps of environments inaccessible to people [8], [9], such as abandoned mines that have recently been mapped in 3-D by mobile robots [10].

In robotic mapping, moving from 2-D to 3-D is not just a trivial extension. The most popular paradigm in 2-D mapping to date are occupancy maps [11], [12], which represent environments by fine-grained grids. While this is feasible in 2-D, in 3-D, the complexity of these representations pose serious scaling limitations [12]. Other popular representations in 2-D involve point clouds [13], [14] or line segments [15]. Line representations have been generalized to 3-D by representing maps through sets of fine-grained polygons [1], [9]. The resulting maps are often quite complex, and off-the-shelf computer graphics algorithms for mesh simplification [16] tend to generate maps that are visually inaccurate [1].

This paper presents an algorithm for recovering low-complexity 3-D models from range and camera data that specifically exploits prior knowledge on the shape of basic building elements. In particular, our approach fits a probabilistic model that consists of large rectangular, flat surfaces to the data collected by a robot. Areas in the map that are not explained well by flat surfaces are modeled by small polygons (as in [9]), enabling our approach to accommodate nonflat areas in the environment. The resulting maps are less complex than those generated by the previous approaches discussed above. Moreover, 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.

To identify low-complexity models, the approach presented here uses a real-time variant of the *expectation-maximization* (EM) algorithm [2], [17]. Our algorithm simultaneously estimates the number of surfaces and their locations. Measurements not explained by any surface are mapped onto fine-grained polygonal representations, enabling our approach to model nonplanar artifacts in the environment. The resulting map is represented in Virtual Reality Markup Language (VRML) format, with texture superimposed from a panoramic camera.

Our approach rests on two key assumptions. First, it assumes that a good estimate of the robot pose is available. The issue of

Manuscript received November 23, 2002; revised May 5, 2003. This paper was recommended for publication by Associate Editor G. Oriolo and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) MARS Program under Contract N66001-01-C-6018 and Contract NBCH1020014, and in part by the National Science Foundation (NSF) under CAREER Grant IIS-9876136 and Grant IIS-9877033. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Washington, DC, 2002, and in part at the International Conference on Machine Learning, Williamstown, MA, 2001.

S. Thrun is with the Computer Science Department, Stanford University, Stanford, CA 94305 USA (e-mail: thrun@stanford.edu).

C. Martin is with the Computer Science and Automation Faculty, Department of Neuroinformatics, Technical University Ilmenau, 98693 Ilmenau, Germany (e-mail: Christian.Martin@MetraLabs.com).

Y. Liu and D. Chakrabarti are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: yufeng@andrew.cmu.edu; deepay@cs.cmu.edu).

D. Hähnel and W. Burgard are with the Institute for Computer Science, Albert-Ludwigs-University, 79110 Freiburg, Germany (e-mail: haehnel@informatik.uni-freiburg.de; burgard@informatik.uni-freiburg.de).

R. Emery-Montemerlo is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: remery@cs.cmu.edu).

Digital Object Identifier 10.1109/TRA.2004.825520

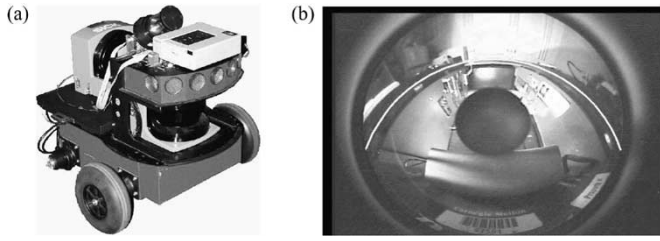


Fig. 1. Mobile robot, equipped with two 2-D laser range finders and a panoramic camera. The camera uses a panoramic mirror mounted only a few centimeters away from the optical axis of the laser range finder.

pose estimation (localization) in mapping has been studied extensively in the robotics literature [18]. In all our experiments, we use a real-time algorithm described in [1] to estimate pose; thus, our assumption is not unrealistic at all, but it lets us focus on the 3-D mapping aspects of our work. Second, we assume that the environment is largely composed of flat surfaces. The flat-surface assumption leads to a convenient closed-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 flat surface, and maps them into finer grained polygonal approximations. Hence, the final map may contain nonflat regions in areas that are not sufficiently flat in the physical world.

Our approach has been fully implemented using the mobile robot shown in Fig. 1(a). This robot is equipped with a forward-pointed laser range finder for localization during mapping, an upward-pointed laser range finder for structural mapping, and a panoramic camera for recording the texture of the environment [see Fig. 1(b)]. The system has been tested in several different buildings. Our results illustrate that the algorithm is effective in generating compact and accurate 3-D maps in real time.

II. GENERATIVE PROBABILISTIC MODEL

A. World Model

In our approach, 3-D maps are composed of rectangular flat surfaces, representing doors, walls, and ceilings, plus sets of small polygons representing nonflat surfaces. We will denote the set of rectangular flat surfaces by θ , where

$$\theta = \{\theta_1, \dots, \theta_J\}. \quad (1)$$

Here, J is the total number of rectangular surfaces θ_j . Each θ_j is described by a total of nine parameters, arranged in three groups

$$\theta_j = \langle \alpha_j, \beta_j, \gamma_j \rangle. \quad (2)$$

The vector α_j is the 3-D surface normal of the surface, the value β_j is the 1-D offset between the surface and the origin of the coordinate system, and γ_j are five parameters specifying the size and orientation of the rectangular area within the (infinite) planar surface represented by α_j and β_j .

B. Measurements

Measurements are obtained using a laser range finder. Each range measurement is projected into 3-D space, exploiting the

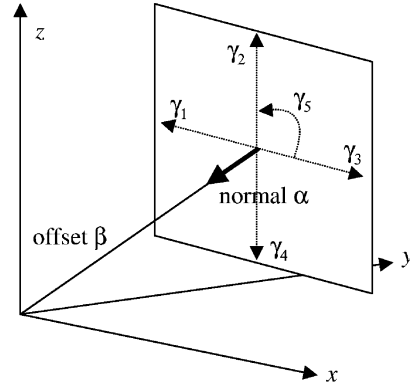


Fig. 2. Illustration of the parameters in the planar surface model, shown here for one surface.

fact that the robot pose is known (see Fig 2). The 3-D coordinate of the i th range measurement will be denoted

$$z_i \in \mathbb{R}^3. \quad (3)$$

We denote the set of all measurements by

$$Z = \{z_i\}. \quad (4)$$

The Euclidean distance of any coordinate z_i in 3-D space to any surface θ_j will be denoted

$$d(z_i, \theta_j). \quad (5)$$

In our implementation, we distinguish two cases; the case where the orthogonal projection of z_i falls into the rectangle, and the case where it does not. In the former case, $d(z_i, \theta_j)$ is given by $\alpha_j \cdot z_i - \beta_j$; in the latter case, $d(z_i, \theta_j)$ is the Euclidean distance between the bounding box of the rectangle and z_i , which is either a point-to-line distance or a point-to-point distance.

C. Correspondences

In devising an efficient algorithm for environment mapping, it will prove convenient to make explicit the relation between individual measurements z_i and the different components θ_j of the model. This is achieved through *correspondence variables*. For each measurement C_i , we define there to be $J + 1$ binary correspondence variables, collectively referred to as C_i

$$C_i = \{c_{i*}, c_{i1}, c_{i2}, \dots, c_{iJ}\}. \quad (6)$$

The vector C_i specifies which part of the model θ “causes” the measurement z_i . Each of the variables in C_i is binary. The variable c_{ij} (for $1 \leq j \leq J$) is 1 if and only if the i th measurement z_i corresponds to the j th surface in the map, θ_j . If the measurement does not correspond to any of the surfaces in the map, the “special” correspondence variable c_{i*} is 1. This might be the case because of random measurement noise, or due to the presence of nonplanar objects in the world.

Naturally, each measurement is caused by exactly one of those $J + 1$ possible causes. This implies that the correspondences in C_i sum to 1

$$c_{i*} + \sum_{j=1}^J c_{ij} = 1. \quad (7)$$

Our algorithm below involves a step in which probabilities over correspondences are calculated from the data.

D. Measurement Model

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

$$p(z_i | C_i, \theta) \quad (8)$$

where C_i is the correspondence vector of the i th measurement, and θ is the set of planar surfaces. Our approach assumes Gaussian measurement noise. Suppose $c_{ij} = 1$, that is, the measurement z_i corresponds to the surface θ_j in the model. The error distribution is then given by the following normal distribution with variance parameter σ :

$$p(z_i | c_{ij} = 1, \theta) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j)}{\sigma^2}}. \quad (9)$$

Notice that the log likelihood of this normal distribution is proportional to the squared Euclidean distance $d(z_i, \theta_j)$ 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, the object detected by a range finder does not correspond to a flat surface, that is, $c_{i*} = 1$. In our approach, we model such events using a uniform distribution over the entire measurement range

$$p(z_i | c_{i*} = 1, \theta) := \begin{cases} 1/z_{\max}, & \text{if } 0 \leq z_i \leq z_{\max} \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

The interval $[0; z_{\max}]$ denotes the measurement range of the range finder. The uniform noise model is clearly just a crude approximation, as real measurement noise is not uniform. However, uniform distributions are mathematically convenient and provide excellent results.

For reasons that shall become apparent below, we note that the uniform density in (10) can be rewritten as follows:

$$\frac{1}{z_{\max}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \log \frac{z_{\max}^2}{2\pi\sigma^2}}. \quad (11)$$

Clearly, a uniform noise model is somewhat simplistic; however, it is mathematically convenient and was found to work well in our experiments.

III. LOG-LIKELIHOOD FUNCTION

To devise a likelihood function suitable for optimization, it shall prove useful to express the sensor model as the following exponential mixture:

$$p(z_i | C_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} [c_{i*} \log \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}]}. \quad (12)$$

This form follows directly from (9) and (11), and the assumption that exactly one variable in C_i is one, whereas all others are zero. This form of the measurement model enables us to devise a compact expression of the *joint probability* of a measurement z_i along with its correspondence variables C_i

$$p(z_i, C_i | \theta) = \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} [c_{i*} \log \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}]}. \quad (13)$$

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

$$p(Z, C | \theta) = \prod_i \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} \times e^{-\frac{1}{2} [c_{i*} \log \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}]}. \quad (14)$$

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

In EM, it is common practice to maximize the log likelihood instead of the likelihood (14), exploiting the fact that the logarithm is monotonic in its argument

$$\log p(Z, C | \theta) = \sum_i \left[\log \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} - \frac{1}{2} c_{i*} \log \frac{z_{\max}^2}{2\pi\sigma^2} - \frac{1}{2} \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2} \right]. \quad (15)$$

Finally, while the formulas above all compute a joint over map parameters *and* correspondence, all we are actually interested in are the map parameters. The correspondences are only interesting to the extent that they determine the most likely map 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 $E_C[\log p(Z, C | \theta)]$, is the expected log likelihood of the data, given the map with the correspondences integrated out. It is obtained directly from (15)

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

In [19], it is shown that maximizing this expectation indeed maximizes the log likelihood of the data.

IV. LIKELIHOOD MAXIMIZATION VIA EM

The expected log likelihood (16) is maximized using EM, a popular method for hill climbing in likelihood space for problems with latent variables [2]. EM generates a sequence of maps, $\theta^{[0]}, \theta^{[1]}, \theta^{[2]}, \dots$. Each map improves the log likelihood of the data over the previous map until convergence. More specifically, EM starts with a random map $\theta^{[0]}$. Each new map is obtained by executing two steps: an E-step, where the expectations of the unknown correspondences $E[c_{ij} | \theta^{[n]}, z_i]$ and $E[c_{i*} | \theta^{[n]}, z_i]$ are calculated for the n th map $\theta^{[n]}$; and an M-step, where a new maximum-likelihood (ML) map $\theta^{[n+1]}$ is computed under these expectations.

A. The E-Step

In the E-step, we are given a map $\theta^{[n]}$ for which we seek to determine the expectations $E[c_{ij} | \theta^{[n]}, z_i]$ and $E[c_{i*} | \theta^{[n]}, z_i]$ 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 for mathematical convenience)

$$\begin{aligned} e_{ij}^{[n]} &:= E \left[c_{ij} \mid \theta^{[n]}, z_i \right] = p \left(c_{ij} \mid \theta^{[n]}, z_i \right) \\ &= \frac{p(z_i \mid \theta^{[n]}, c_{ij}) p(c_{ij} \mid \theta^{[n]})}{p(z_i \mid \theta^{[n]})} \\ &= \frac{e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j)}{\sigma^2}}}{e^{-\frac{1}{2} \log \frac{z_{\max}^2}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2} \frac{d^2(z_i, \theta_k)}{\sigma^2}}} \end{aligned} \quad (17)$$

and similarly

$$\begin{aligned} e_{i*}^{[n]} &:= E \left[c_{i*} \mid \theta^{[n]}, z_i \right] \\ &= \frac{e^{-\frac{1}{2} \log \frac{z_{\max}^2}{2\pi\sigma^2}}}{e^{-\frac{1}{2} \log \frac{z_{\max}^2}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2} \frac{d^2(z_i, \theta_k)}{\sigma^2}}} \end{aligned} \quad (18)$$

As pointed out in [17] and [19], substituting these expectations into the log likelihood (16) lower bounds the log likelihood by a function tangent to it at $\theta^{[n]}$.

B. The M-Step

In the M-step, this lower bound is optimized. More specifically, we are given the expectations $e_{ij}^{[n]}$ and $e_{i*}^{[n]}$ and seek to calculate a map $\theta^{[n+1]}$ that maximizes the expected log likelihood of the measurements, as given by (16). In other words, we seek surface parameters $\langle \alpha_j^{[n+1]}, \beta_j^{[n+1]}, \gamma_j^{[n+1]} \rangle$ that maximize the expected log likelihood of the map under fixed expectations $e_{ij}^{[n]}$ and $e_{i*}^{[n]}$.

Obviously, many of the terms in (16) do not depend on the map parameters θ . This allows us to simplify (16) and instead carry out the following minimization:

$$\theta^{[n+1]} = \arg \min_{\theta} \sum_i \sum_j e_{ij}^{[n]} d^2(z_i, \theta_j). \quad (19)$$

The actual M-step proceeds in two steps. First, our approach determines the parameters $\alpha_j^{[n+1]}$ and $\beta_j^{[n+1]}$, which specify the principal orientation and location of the rectangular surface without the surface boundary. If walls are assumed to be boundless, the minimization (19) is equivalent to the minimization

$$\langle \alpha_j^{[n+1]}, \beta_j^{[n+1]} \rangle = \arg \min_{\alpha, \beta} \sum_i \sum_j e_{ij}^{[n]} (\alpha_j \cdot z_i - \beta_j)^2 \quad (20)$$

subject to the normality constraints $\alpha_j \cdot \alpha_j = 1$, for all j . This quadratic optimization problem is commonly solved via Lagrange multipliers λ_j for $j = 1, \dots, J$ [20], [21]

$$L := \sum_i \sum_j e_{ij}^{[n]} (\alpha_j \cdot z_i - \beta_j)^2 + \sum_j \lambda_j \alpha_j \cdot \alpha_j. \quad (21)$$

Obviously, for each minimum of L , it must be the case that $(\partial L)/(\partial \alpha_j) = 0$ and $(\partial L)/(\partial \beta_j) = 0$. Setting the derivatives of L to zero leads to the linear system of equalities

$$\sum_i e_{ij}^{[n]} \left(\alpha_j^{[n+1]} \cdot z_i - \beta_j^{[n+1]} \right) z_i - \lambda_j \alpha_j^{[n+1]} = 0 \quad (22)$$

$$\sum_i e_{ij}^{[n]} \left(\alpha_j^{[n+1]} \cdot z_i - \beta_j^{[n+1]} \right) = 0 \quad (23)$$

$$\alpha_j^{[n+1]} \cdot \alpha_j^{[n+1]} = 1. \quad (24)$$

The values of $\beta_j^{[n+1]}$ are obtained from (22) and (23)

$$\beta_j^{[n+1]} = \frac{\sum_k e_{kj}^{[n]} \alpha_j^{[n+1]} \cdot z_k}{\sum_k e_{kj}^{[n]}}. \quad (25)$$

Substituting those back into (22) gives us

$$\sum_i e_{ij}^{[n]} \left(\alpha_j^{[n+1]} \cdot z_i - \frac{\sum_k e_{kj}^{[n]} \alpha_j^{[n+1]} \cdot z_k}{\sum_k e_{kj}^{[n]}} \right) z_i = \lambda_j \alpha_j^{[n+1]}. \quad (26)$$

This is a set of linear equations of the type

$$A_j^{[n]} \cdot \alpha_j^{[n+1]} = \lambda_j \alpha_j^{[n+1]} \quad (27)$$

where each $A_j^{[n]}$ is a 3×3 matrix whose elements are as follows:

$$a_{st}^{[n]} = \sum_i e_{ij}^{[n]} z_{is} z_{it} - \frac{\sum_i e_{ij}^{[n]} z_{it} \sum_k e_{kj}^{[n]} z_{ks}}{\sum_k e_{kj}^{[n]}} \quad (28)$$

for $s, t \in \{1, 2, 3\}$, subject to (24). It is now easy to see that each solution of (27) must be an eigenvector of $A_j^{[n]}$. The two eigenvectors with the largest eigenvalues describe the principal orientation of the surface. The third eigenvector, which corresponds to the smallest eigenvalue, is the normal vector of this surface, hence, our desired solution for $\alpha_j^{[n+1]}$.

Finally, the M-step calculates new bounding boxes $\gamma_j^{[n+1]}$. It does so by determining the minimum rectangular box on the surface which includes all points whose ML assignment is the j th surface θ_j

$$\left\{ z_i, \text{ such that } j = \arg \max_{k \in 1, \dots, J, *} e_{i,k}^{[n]} \right\}. \quad (29)$$

This optimization problem does not possess an easy closed-form solution. Our approach probes the orientation in one-degree intervals, then calculates the tightest bounding box for each orientation. The bounding box with the smallest enclosed surface volume is finally selected. This step leads to a near-optimal rectangular surface that contains all measurements that most likely correspond to the surface at hand.

C. Determining the Number of Surfaces

Parallel to computing the surface parameters, our approach determines the number of surfaces J . Our approach is based on a straightforward Bayesian prior that penalizes complex maps using an exponential prior, written here in log-likelihood form

$$p(\theta \mid Z) \propto p(Z \mid \theta) - \kappa J. \quad (30)$$

Here, κ is a constant factor. The final map estimator is, thus, a maximum *a posteriori* (MAP) probability estimator, which combines the complexity penalizing prior with the data likelihood calculated by EM. In practice, this approach implies that surfaces not supported by sufficiently many data measurements (weighted by their expectation) are discarded. This makes it possible to choose the number of rectangular surfaces J concurrently with the execution of the EM algorithm.

In our implementation, the search for the best J is interleaved with running the EM algorithm. The search involves a step for creating new surfaces, and another one for terminating surfaces,

both executed in regular intervals (every 20 iterations in our offline implementation). In the surface creation step, new surfaces are created based on measurements z_i that are poorly explained by the existing model. A measurement z_i is considered poorly explained if its value e_{i*} exceeds a threshold, indicating that none of the planar surfaces in the model explain the measurement well. A new surface is started if three adjacent measurements are poorly explained; the initial parameters of this new surface are then uniquely determined through the coordinates of these three measurements. The number of new surfaces is kept limited at each iteration, using random selection among all candidates if the number of candidate surfaces exceeds the total limit. The new surfaces are then added to the model and treated identically to all other surfaces in subsequent iterations of EM.

In the termination step, each surface undergoes a posterior evaluation using the criterion set forth in (30). If removing a surface or fusing it with a nearby surface *increases* the posterior in (30), the corresponding action is taken. Otherwise, it is retained in the model. In this way, only surfaces supported by sufficiently many data points survive the selection process, avoiding the overfitting that inevitably would occur without a complexity penalty term.

D. Texture Mapping and Visualization

Textures are extracted from the panoramic camera, along a stripe shown in Fig. 1(b) that corresponds to the range measurement taken by the vertical laser range finder. These stripes are collected at frame rate and pasted together into raw texture maps. These maps are then mapped onto the planar surfaces in real time, using a technique analogous to the one described in [22]. Textures of the same feature in the environment recorded at different points in time are presently not merged due to tight computational constraints, which is a clear shortcoming of our present implementation.

The final model contains all surfaces in θ ; however, it lacks information on nonflat objects in the environment. The final 3-D visualization of the environment is enriched by fine-grained polygonal models of such nonflat objects. In particular, our approach analyzes all measurements whose most likely correspondence is the class “*.” For any occurrence of three nearby such measurements, a small triangle is introduced into the final 3-D visualization. In this way, the visualization contains not only large flat surfaces, but also fine-grained polygonal models of nonflat objects in the environment. The final output of our program is a VRML file, which makes it possible to render the 3-D model using off-the-shelf software.

V. ONLINE EM

Our approach has been extended into an online algorithm that enables robots to acquire compact 3-D models in real time [23]. EM, as presented so far, is inherently offline. It required multiple passes through the data set. As the data set grows, so does the computation per iteration of EM. This limitation is a common limitation of the vanilla EM algorithm [19].

The key insight that leads to an online implementation is that during each fixed time interval, only a constant number of new range measurements arrive (collectively referred to as z_i). If we

begin our estimation with the model acquired in the first $i - 1$ measurements, many new measurement values have to be constantly incorporated into the model in response to observing z_i . Such a routine would be strictly incremental; however, it would fail to adjust correspondence variables based on data acquired at a later point in time. Our approach is slightly more sophisticated in that it adjusts past expectations, but it does this in a way that still conforms to constant update time.

A. Online E-Step

The online E-step considers only a finite subset of all measurements, and only calculates expectations for the correspondence variables of those measurements. In particular, it includes all new range measurements, of which there are only finitely many. Additionally, it includes past measurements that meet the following condition. They lie at the boundary of two surfaces (judging from their ML assignments), or they are entirely unexplained by any existing surface. These conditions alone are insufficient to assure constant time computation. Thus, we also attach a counter to each measurement that keeps track of the total number of times it was considered in an E-step calculation. If this counter exceeds a threshold, the expected correspondence value is frozen and never recalculated again. It is easy to see that the last condition ensures constant time update (in expectation). The former condition identifies “interesting” measurements, which greatly reduces the constant factor in the computational complexity.

B. Online M-Step

The online M-step is slightly more complex. This is because as the data set grows, so do two things: the number of model components J ; and the number of measurements z_i associated with each model component. Each of these expansions have to be controlled to ensure that the M-step can be executed in real time.

In the online M-step, only a small number of “active” surfaces are re-estimated. We say that a surface θ_j is active at time i if the E-step executed at that time changed, for any of the updated measurements, the ML correspondence to or away from the surface θ_j . Clearly, this is an approximation, in that any adjustment of the expected correspondence affects all model parameters, at least in principle. However, our approach is a good approximation that limits the number of surfaces considered in the M-step to a constant number.

Finally, our approach addresses the fact that the number of points involved in each maximization grows over time. In conventional EM, every measurement participates in the calculation of all surface parameters, though most make only a negligible contribution. To reduce the number of measurements used in the M-step, our online approach considers only those whose ML data association corresponds to the model component in question. This is analogous to the well-known k-means algorithm [24] for clustering, which approximates the EM algorithm using hard assignments. Furthermore, if the number of ML measurements exceeds a threshold, our approach randomly subsamples those measurements. By doing so, the amount of computation when recalculating the parameters of a model component is bounded by a constant number, as is the overall computational complexity of the M-step.

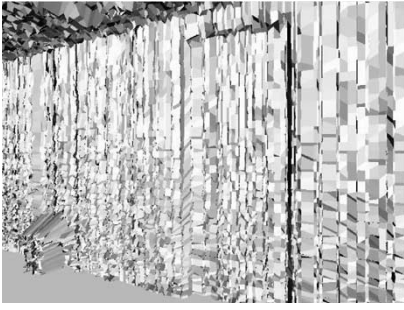


Fig. 3. Polygonal map generated from raw data, not using EM. The display without texture shows the level of noise involved. In particular, it illustrates the difficulty of separating the door from the nearby wall (as achieved by EM).

C. Online Model Selection

Finally, our approach also implements Bayesian model selection in real time. New surfaces are introduced for new measurements which are not “explained” by any of the existing surfaces in the map. A measurement is explained by an existing surface if its likelihood of having been generated from that surface is above a threshold; otherwise, it will be used to seed a new surface. Surfaces are removed from the map if after a fixed number of iterations, they are not supported by sufficiently many measurements, in accordance with the map complexity penalty factor κ set forth in (30). As a result, the number of surfaces J increases with the complexity of the environment, while all computation can still be carried out online, irrespective of the total map size.

The resulting algorithm is strictly incremental. It has been implemented on a low-end laptop PC, where it is able to construct compact 3-D maps in real time.

VI. EXPERIMENTAL RESULTS

Our experiments were carried out in two stages. First, we implemented and evaluated the offline EM algorithm. This evaluation served us to establish the basic capability of extracting large planar surfaces from complex data sets using EM. Our second set of experiments was carried out in real time using the online version of our approach.

All computation in these experiments was carried out onboard the moving robot platform shown in Fig. 1. As noted above, the robot is equipped with two SICK PLS laser range finders, one pointed forward and one pointed upward, perpendicular to the robot’s motion direction. The SICK sensor covers an area of 180 degrees with a one-degree angular resolution. The range accuracy is described by the manufacturer as ± 5 cm. The sampling rate in our experiments is approximately 5 Hz (entire scans).

A. Offline EM

The offline version of EM was tested using a data set acquired inside a university building. The data set consists of 168 120 range measurements and 3270 camera images, collected in less than two minutes. In a prealignment phase, our software extracted 3 220 950 pixels, all of which directly correspond to range measurements. Nearby scans were converged into polygons. Fig. 3 shows a detail of the resulting map without the texture superimposed. This figure clearly illustrates the high level

(a) Fine-grained polygonal maps generated from raw data

(b) Low-complexity maps generated using EM



Fig. 4. 3-D map. (a) Generated from raw sensor data. (b) Generated using the offline version of EM. In this map, 94.6% of all measurements are explained by seven surfaces. Notice that the map in (b) is smoother and appears to be visually more accurate than the one in (a).

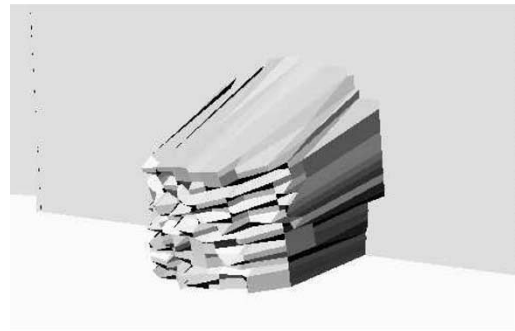


Fig. 5. Model of a trash bin in the final map, with a large planar rectangular surface patch in the background. Our algorithm recognizes that this object cannot be explained by planar surfaces with sufficient likelihood, hence, it retains the fine-grained polygonal representation.

of noise in the raw data. The left column of Fig. 4 shows three views of the map with texture superimposed.

The result of our EM algorithm is shown in the right column of Fig. 4. This specific map contains of $J = 7$ flat surfaces, which together account for 94.6% of all measurements. Clearly, the new map is smoother and visually more accurate. It also models nonplanar regions well, as manifested by a trashbin in the top panel of that figure, which is not modeled by a planar surface. The corresponding measurements are not mapped to any of the surfaces θ_i . Fig. 5 shows the corresponding map segment, illustrating the importance of merging flat surfaces with fine-grained polygons for modeling building interiors. Notice also that the wall surface is different from the door surface. A

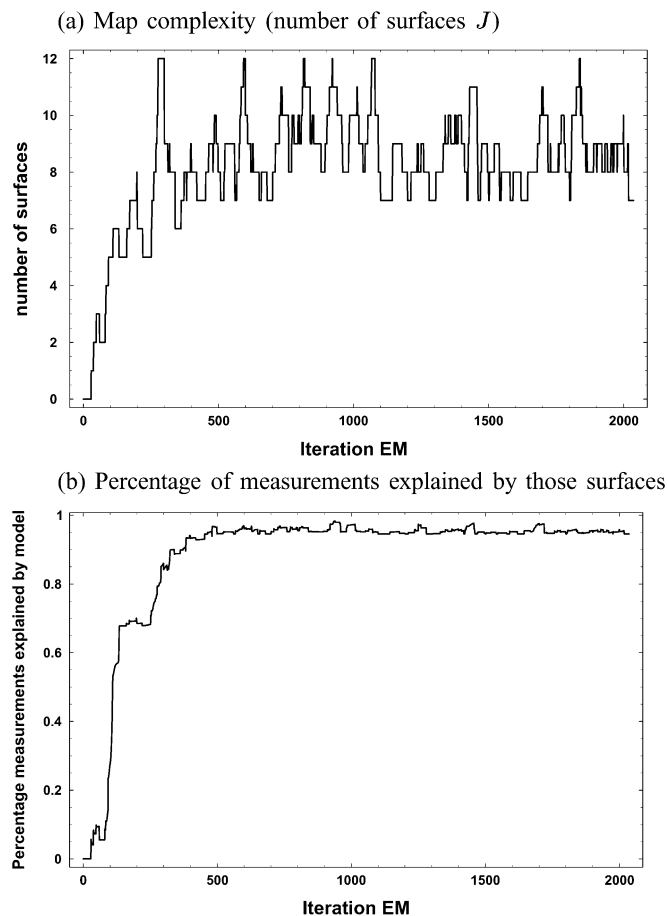


Fig. 6. Offline EM. (a) Number of surfaces J . (b) Percentage of points explained by those surfaces as function of the iterator. A good map is available after 500 iterations.

small number of measurements in the door surface are erroneously assigned to the wall surface. The poster board shown in various panels of Fig. 4 highlights once again the benefits of the EM approach over the raw data maps. Its visual accuracy is higher, thanks to the planar model on which the texture is being projected.

Fig. 6 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. 2000 iterations require approximately 20 min computation on a low-end PC.

B. Real-Time Implementation

The real-time implementation using online EM was evaluated more thoroughly and in several different buildings. Overall, we found our approach to be highly reliable in generating accurate maps. Fig. 7 illustrates the online creation of compact maps. Shown in the left column are maps generated directly from the raw measurement data by creating polygons for any set of nearby measurements. The right column shows a sequence of maps built in real time, using our incremental EM algorithm. We note that this specific illustration shows only vertical surfaces; however, our implementation handles surfaces in arbitrary orientations.

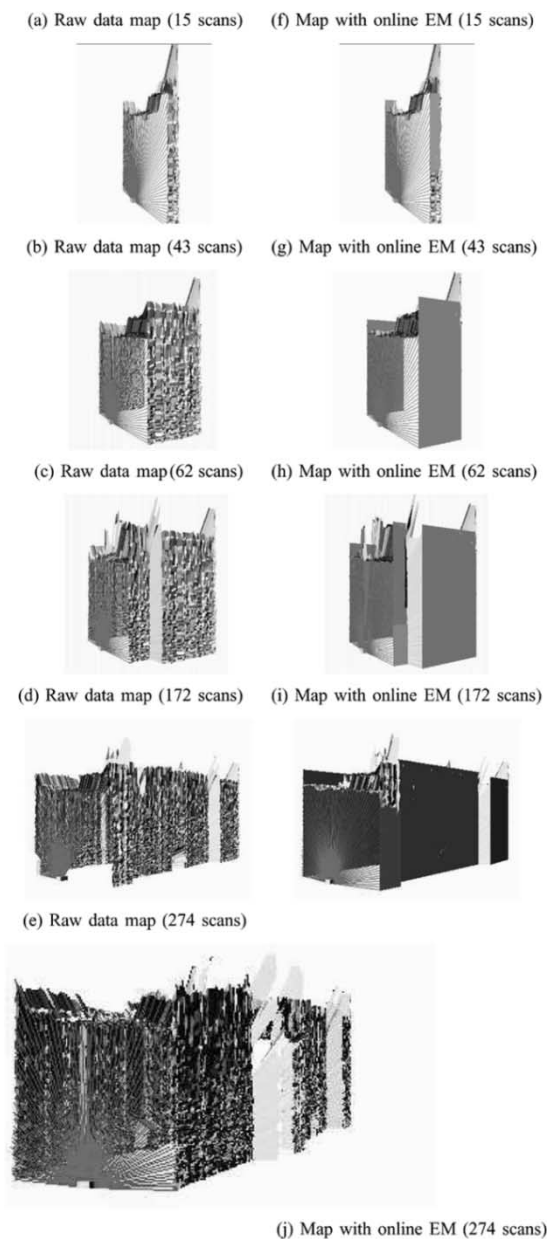


Fig. 7. Raw data (left column) and maps generated using online EM (right column). The lines visualize the most recent range scan. Some of the intermediate maps exhibit suboptimal structures, which are resolved in later iterations of EM. Despite this backward correction of past estimates, the algorithm still runs in real time, due to careful selection of measurements that are considered in the EM estimation.

As it is easily seen, the new map is more compact than the raw data map. A small number of surfaces suffices to model the vast amounts of data. Close inspection of Fig. 7 illustrates EM at work. For example, a small region in Fig. 7(i) has not been



Fig. 8. Views of a compact 3-D texture map built in real time with an autonomously exploring robot.

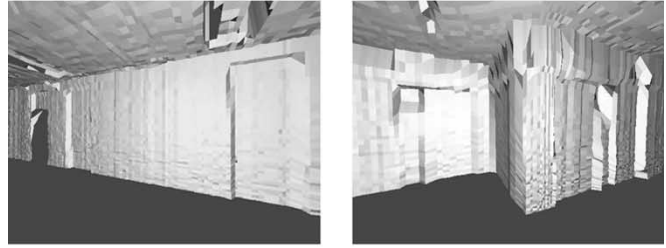
identified as belonging to a flat wall, simply because the amount of noise in the rectangular surface below this patch is too large. Further optimization of this past data leads to a more compact map, as illustrated in Fig. 7(j).

Fig. 8 shows views of a compact 3-D map acquired in real time, for the same environment as reported previously. The entire data collection requires less than 2 min, during which all processing occurs.

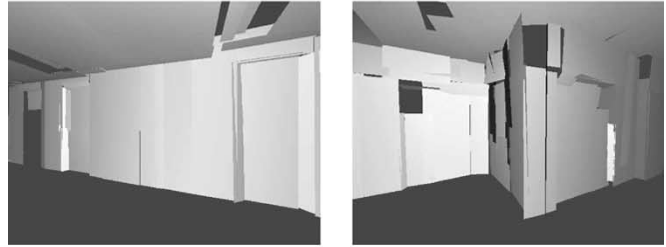
As is easily seen, the identification of rectangular surfaces in the environment has a positive effect on the visual acuity of the map. Fig. 9 shows inside projections of a 3-D map built by EM and compares it with a map built without EM, using fine-grained polygonal maps, and using data sets obtained in two different university buildings. Obviously, the visual accuracy of the texture projected onto flat surfaces is higher than the renderings obtained from the fine-grained polygonal map. This illustrates once again that the resulting map is not only more compact, but it also provides more visual detail than the map created without our approach.

In an attempt to quantitatively evaluate our approach, we mapped three different corridor environments in different buildings. The complexity of those environments was comparable to the maps shown here. The number of initial polygons was between $3.5 \cdot 10^4$ and $6.5 \cdot 10^4$. The final maps contained, on average, 0.60% as many polygons (0.69%, 0.80%, and 0.32%), which corresponds to an average compression ratio of 1:192. Overall, we found that the online version generated an order of

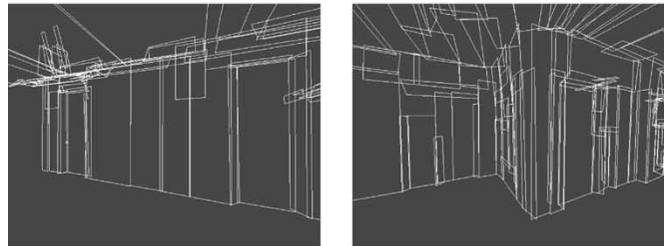
(a) Raw data map (using a high-accuracy range finder)



(b) Planes, extracted from the map using EM



(c) Plane boundaries



(d) Full texture map (planes only)

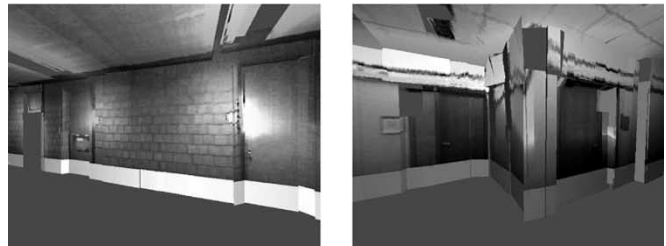


Fig. 9. Maps generated in real time of office environments at Carnegie Mellon University's Wean Hall (left column) and Stanford University's Gates Hall (right column).

magnitude more flat surfaces J than the offline version. This increased number was partially due to a lower penalty term κ . An additional cause of the increase in J was the fact that the online version considers splitting and fusion decisions only for a brief time period. Overall, the increased number of surfaces J led to a decrease in the overall complexity of the map, thanks to the fact that significantly more measurements were explained by surfaces θ_j found by EM.

We finally note that all computation in our experiment was carried out on a single laptop on board the robot. The views in Figs. 7–9 were rendered using a standard software package for rendering VRML models.

VII. RELATED WORK

As argued in the introduction to this paper, the vast majority of robot modeling research has focused on building maps in 2-D.

Our approach is reminiscent of an early paper by Chatila and Laumond [15] and a related paper by Crowley [25], who proposed to reconstruct low-dimensional line models in 2-D from sensor measurements. Our work is also related to work on line extraction from laser range scans [26]. However, these methods address the 2-D case, where lines can be extracted from a single scan.

In the area of computer vision, object recognition under geometric constraints (such as the planar surface assumption) is a well-researched field [27]–[29]. Multiple researchers have studied the topic of 3-D scene reconstruction from data. Approaches for 3-D modeling can roughly be divided into two categories: methods that assume knowledge of the pose of the sensors [4]–[6], [30], [31]; and methods that do not [32]. Of great relevance is a paper by Roth and Wibowo [33], who have also proposed the use of a mobile platform to acquire textured 3-D models of the environment. Their systems include a 3-D range sensor and eight cameras. Just like our system, theirs uses sensor information to compensate odometric errors. Their approach, however, does not consider the uncertainty of individual measurements when generating the 3-D model.

Our approach is somewhat related to [34], 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. Like ours, the environment model is composed of flat surfaces; however, due to the use of stereo vision, the range data is too incomplete to allow for a fully automated modeling process to take place. Consequently, the technique in [34] relies on manual guidance in the process of identifying planar surfaces. Related work on outdoor terrain modeling can be found in [7] and [35].

VIII. CONCLUSION

We have presented an online algorithm for building compact maps of building interiors. This approach utilizes the EM algorithm for finding rectangular surface patches in 3-D data, acquired by a moving robot equipped with laser range finders and a panoramic camera. While EM is traditionally an offline algorithm, a modified version of EM was presented, which is capable of generating such maps online, while the robot is in motion. This approach retains the key advantage of EM, namely, the ability to revise past assignments and map components based on future data, while simultaneously restricting the computation in ways that make it possible to run the algorithm in real time, regardless of the size of the map. Experimental results illustrate that this approach enables mobile robots to acquire compact and accurate maps of corridor-style indoor environments in real time.

All results shown in this paper were obtained using accurate 2-D laser range finders. The mathematical approach can accommodate a wider array of range finders, such as sonars and range cameras, assuming that the sensor model is adjusted appropriately. However, we suspect that the high spatial and angular resolution of our laser range finder plays an important role in the success of our approach.

Although stated here in the context of finding rectangular flat surfaces, the EM algorithm is more general in that it can easily

handle a richer variety of geometric shapes. The extension of our approach to richer classes of objects is subject to future research. Another topic of future research is to augment the EM algorithm to estimate the robot's location during mapping, as described in [36].

Finally, it would be worthwhile to include both lasers' data in the mapping process. Unfortunately, the upward-pointed laser may never observe the same aspect of the environment twice; hence, further assumptions (such as smoothness) would be necessary to use its data. It would also be interesting to add the robot pose variables as latent variables in the EM algorithm, so that localization and modeling can be interleaved, as in [36]. Unfortunately, the added computational complexity incurred by such extensions would make it difficult to execute the resulting algorithm online and in real time.

ACKNOWLEDGMENT

The authors thank three anonymous reviewers for insightful comments on an earlier version of this manuscript.

REFERENCES

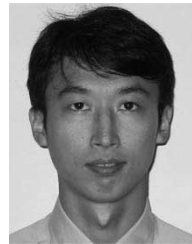
- [1] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 335–363, 2001.
- [2] A. Dempster, A. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
- [3] D. Kortenkamp, R. Bonasso, and R. Murphy, Eds., *AI-Based Mobile Robots: Case Studies of Successful Robot Systems*. Cambridge, MA: MIT Press, 1998.
- [4] R. Bajcsy, G. Kamberova, and L. Nocera, "3-D reconstruction of environments for virtual reconstruction," in *Proc. 4th IEEE Workshop on Applications of Computer Vision*, Oct. 1998, pp. 160–167.
- [5] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs," in *Proc. 23rd Int. Conf. Computer Graphics and Interactive Techniques*, 1996, CD-ROM.
- [6] H. Shum, M. Han, and R. Szeliski, "Interactive construction of 3-D models from panoramic mosaics," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, June 1998, pp. 427–433.
- [7] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master, "Calibrated, registered images of an extended urban area," in *Proc. Conf. Computer Vision and Pattern Recognition*, Dec. 2001, pp. 1-813–1-820.
- [8] J. Casper, "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," Master's thesis, Dept. Comput. Sci. Eng., Univ. South Florida, Tampa, FL, 2002.
- [9] M. Montemerlo, D. Hähnel, D. Ferguson, R. Triebel, W. Burgard, S. Thayer, W. Whittaker, and S. Thrun, "A system for three-dimensional robotic mapping of underground mines," *Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-02-185*, 2002.
- [10] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sept. 2003, pp. 4270–4275.
- [11] A. Elfes, "Occupancy grids: A probabilistic framework for robot perception and navigation," Ph.D. dissertation, Dept. Elec. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, 1989.
- [12] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, pp. 61–74, 1988.
- [13] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*, 2000, CD-ROM.
- [14] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, 1997.
- [15] R. Chatila and J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Mar. 1985, pp. 138–145.
- [16] M. Garland and P. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," in *Proc. IEEE Visualization*, Oct. 1998, pp. 263–269.

- [17] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York, NY: Wiley, 1997.
- [18] "Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)," in *Proc. IEEE Int. Conf. Robotics and Automation*, J. Leonard, J. Tardós, S. Thrun, and H. Choset, Eds., Washington, DC, 2002.
- [19] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. Jordan, Ed. Norwell, MA: Kluwer, 1998.
- [20] D. Eberly, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*. New York and San Mateo, CA: Morgan Kaufman/Academic, 2001.
- [21] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun, "Using EM to learn 3-D models with mobile robots," in *Proc. Int. Conf. Machine Learning*, 2001, CD-ROM.
- [22] S. Nayar, "Omnidirectional vision," in *Proc. 9th Int. Symp. Robotics Research*, Shonan, Japan, 1997, CD-ROM.
- [23] C. Martin and S. Thrun, "Real-time acquisition of compact volumetric 3D maps with mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, May 2002, pp. 311–316.
- [24] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
- [25] J. Crowley, "World modeling and position estimation for a mobile robot using ultrasonic ranging," in *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 674–680.
- [26] F. Lu and E. Miliot, "Robot pose estimation in unknown environments by matching 2-D range scans," *J. Intell. Robot. Syst.*, vol. 18, pp. 249–275, 1998.
- [27] W. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA: MIT Press, 1991.
- [28] O. Faugeras, *Three-Dimensional Computer Vision, A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.
- [29] A. R. Pope, "Model-based object recognition—a survey of recent research," Univ. British Columbia, Dept. Comput. Sci., Vancouver, BC, Canada, Tech. Rep. TR-94-04, 1994.
- [30] P. Allen and I. Stamos, "Integration of range and image sensing for photorealistic 3-D modeling," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2000, pp. 1435–1440.
- [31] S. Becker and M. Bove, "Semiautomatic 3-D model extraction from uncalibrated 2-D camera views," in *Proc. SPIE Symp. Electronic Imaging*, San Jose, CA, 1995, CD-ROM.
- [32] S. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin, "Sensor-based creation of indoor virtual environment models," in *Proc. 4th Int. Conf. Virtual Systems and Multimedia*, Geneva, Switzerland, 1997, CD-ROM.
- [33] G. Roth and R. Wibowo, "A fast algorithm for making mesh models from multiple-view range data," in *Proc. DND/CSA Robotics and Knowledge-Based Systems Workshop*, 1995, CD-ROM.
- [34] L. Iocchi, K. Konolige, and M. Bajracharya, "Visually realistic mapping of a planar environment with stereo," in *Proc. Int. Symp. Experimental Robotics*, Waikiki, HI, 2000, CD-ROM.
- [35] Y.-Q. Cheng, E. Riseman, X. Wang, R. Collins, and A. Hanson, "Three-dimensional reconstruction of points and lines with unknown correspondence across images," *Int. J. Comput. Vis.*, vol. 45, no. 2, pp. 129–156, 2000.
- [36] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Mach. Learning, Auton. Robots*, vol. 31, pp. 29–53, 1998.



Christian Martin is currently working toward the M.Sc. degree in computer science at the Technical University of Ilmenau, Ilmenau, Germany.

He is interested in research on intelligent mobile robots.



Yufeng Liu received the Ph.D. degree in physics and the M.S. degree in computer science in 2003.

He is interested in machine learning, robotics, and data mining.



Dirk Hahnel studied computer science at the University of Bonn. He is currently working toward the Ph.D. degree at the Department of Computer Science, University of Freiburg, Freiburg, Germany.

His areas of interest lie in mobile robot navigation and especially in the learning of two-dimensional and three-dimensional maps.



Rosemary Emery-Montemerlo received the B.S. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1999. She is currently working toward the Ph.D. degree at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Her research interests include machine learning and multirobot systems.



Deepayan Chakrabarti received the B.Tech. degree in computer science from the Indian Institute of Technology, Kanpur, India, in 2000. He is currently working toward the Ph.D. degree at the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

He currently works on mining patterns from large graph datasets.



Sebastian Thrun received the Ph.D. degree from the University of Bonn, Bonn, Germany, in 1995.

He is currently an Associate Professor at Stanford University. Previously, he spent eight years on the faculty of Carnegie Mellon University, Pittsburgh, PA. He pursues research on artificial intelligence, robotics, and machine learning.



Wolfram Burgard studied computer science at the University of Dortmund, Dortmund, Germany, and received the Ph.D. degree from the University of Bonn, Bonn, Germany, in 1991.

Currently, he is an Associate Professor with the Department of Computer Science, University of Freiburg, Freiburg, Germany, where he is head of the Lab for Autonomous Intelligent Systems. His fields of interest lie in robotics and artificial intelligence.