Real-Time Acquisition of Compact Volumetric Maps with Mobile Robots

Christian Martin and Sebastian Thrun
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper describes an online algorithm for generating compact 3D maps of mobile robot environments. Maps generated by our approach consist of small numbers of planar surfaces, which are augmented by fine-grained polygons for non-flat environmental features. Our approach builds on the expectation maximization (EM) algorithm, but develops a new, incremental version that can be executed in real-time. Experimental results obtained in corridor-type environments illustrate that compact and accurate maps can be acquired in real-time, from range and camera data collected by a mobile robot.

1 Introduction

This paper presents an algorithm for generating compact models of indoor environments in real-time from range and camera measurements. The vast majority of successful robot mapping algorithms represent maps by grids [4], raw point measurements [5, 7], or line segments [1]. While such representations appear to be sufficient for navigation, they suffer from two major disadvantages. First, there are intrinsic scaling limitations. For example, while two-dimensional occupancy maps require relatively little memory, the memory requirements of the same approach in 3D are prohibitive (despite recent work on this topic[9]). Second, and more importantly, such maps fail to capture the true nature of indoor environments. Indoor environments are often composed of walls, furniture, doors, windows etc, many of which are characterized by specific geometric features (e.g., are flat). An understanding of such objects and their geometric properties (e.g., flatness) will inevitably lead to new, more powerful mapping algorithms that can generate more accurate maps.

This paper provides a step in the direction of building more compact maps, using a simple object representation. We describe an algorithm that identifies flat rectangular surfaces form the sensor data, such as walls, ceilings, and doors. The mathematical approach for finding such objects is the expectation maximization (EM) algorithm [2]. The EM approach combines a phase of searching for a compact map with planar rectangular surfaces, with one that associates measurements with individual surfaces. By doing so, it can generate accurate surface maps even at the boundary of different surfaces. In addition, our approach allows for objects that are not part of any rectangular surface, which are then represented using fine-grained polygons.

EM has previously been proposed for building 3D maps [6]. A well-known limitation of the classical EM algorithm is its inherent offline nature [8]. This is because EM requires *multiple* passes through the entire data set when generating maps. This is problematic for many application scenarios, such as robot exploration [11], where map building and control is interleaved. To overcome this problem, this paper develops the basic EM algorithm into an online (and real-time) algorithm. The online property is obtained by limiting the number of times a measurement is considered in EM calculations. A carefully crafted strategy for determining when to consider what measurement in the EM procedure leads to an algorithm that can be executed in real-time on a low-cost PC, as documented in the experimental results section of this paper.

Our approach rests on two key assumptions. First, it assumes that 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 [12] to estimate pose; thus, our assumption is not unrealistic at all, but it lets us focus on the 3D 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 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 map may contain non-flat regions in areas that are not sufficiently flat in the physical world.

Our approach has been applied to building multi-planar textural maps of several indoor environments in real-time. The robot shown in Figure 1a 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 Figure 1b). Our results illustrate that the algorithm is highly robust and effective in generating compact maps in real-time.

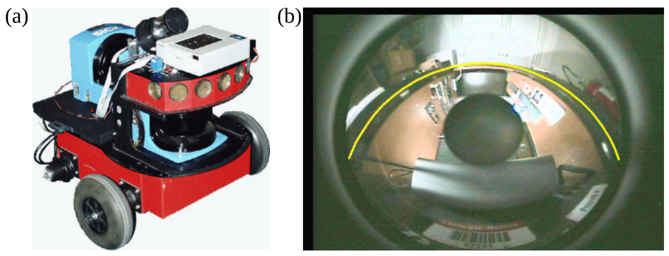

Figure 1: Mobile robot, equipped with two 2D laser range finders and a panoramic camera. The camera uses a panoramic mirror mounted only a few centimeters away from the optical axis of the laser range finder.

2 Flat Surface Model

Mathematically, a map is a finite collection of rectangular flat surfaces, representing doors, walls, ceilings, plus a set of small polygons representing non-flat artifacts in the environment. We will denote the set of rectangular flat surfaces by θ, where

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

Here J is the total number of rectangular surfaces θ_j. Each θ_j is described by a total of 9 parameters, arranged in three groups:

$$\theta_j \quad = \quad \langle \alpha_j, \beta_j, \gamma_j \rangle \tag{2}$$

Here α_j is the three-dimensional surface normal of the rectangular surface, β_j is the one-dimensional 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. The Euclidean distance of any coordinate z in 3D space to any surface θ_j will be denoted

$$d(z, \theta_j) \tag{3}$$

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

Measurements correspond to points in 3D space. The i-th sensor measurement, denoted

$$z_i \quad \in \quad \Re^3 \tag{4}$$

is a 3D coordinate of a point detected by a laser range finder. 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

$$Z \quad = \quad \{z_i\} \tag{5}$$

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:

$$p(z_i | \theta) \tag{6}$$

Our approach assumes 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

$$p(z_i | \theta_j) \quad := \quad \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{1}{2} \frac{d^2(z_i, \theta_j)}{\sigma^2}} \tag{7}$$

Notice that the log likelihood of this normal distribution is proportional to the 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, 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:

$$p(z_i | \theta_*) \quad := \quad \left\{ \begin{array}{cl} 1/z_{\max} & \text{if } 0 \leq z_i \leq z_{\max} \\ 0 & \text{otherwise} \end{array} \right. \tag{8}$$

Here θ_* denotes a 'phantom' component of the map 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.

3 The Log-Likelihood Function

In statistical terms, finding a plausible map is equivalent to maximizing a likelihood function. To define the likelihood function, we have to introduce a new set of random variables, called the *correspondences* 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 map, 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 map θ. The correspondence vector of the i-th measurement is given by

$$C_i \quad = \quad \{c_{i*}, c_{i1}, c_{i2}, \ldots, c_{iJ}\} \tag{9}$$

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

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

$$p(z_i | C_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{1}{2}\left[c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}\right]} \tag{10}$$

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:

$$p(z_i, C_i | \theta) = \frac{1}{(J+1)\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left[c_{i*} \ln \frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij} \frac{d^2(z_i, \theta_j)}{\sigma^2}\right]} \tag{11}$$

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

$$p(Z, C|\theta) \tag{12}$$
$$= \prod_i \frac{1}{(J{+}1)\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left[c_{i*}\ln\frac{z_{\max}^2}{2\pi\sigma^2} + \sum_j c_{ij}\frac{d^2(z_i,\theta_j)}{\sigma^2}\right]}$$

This equation is simply the product of (11) over all measurements z_i.

It is common practice to maximize the log-likelihood instead of the likelihood (12):

$$\ln p(Z, C|\theta) = \sum_i \ln\frac{1}{(J{+}1)\sqrt{2\pi\sigma^2}} - \frac{1}{2}c_{i*}\ln\frac{z_{\max}^2}{2\pi\sigma^2} - \frac{1}{2}\sum_j c_{ij}\frac{d^2(z_i,\theta_j)}{\sigma^2} \tag{13}$$

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 (13), 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 map with the correspondences integrated out. It is obtained directly from Equation (13):

$$E_C[\ln p(Z, C|\theta)]$$
$$= \sum_i \ln\frac{1}{(J{+}1)\sqrt{2\pi\sigma^2}} - \frac{1}{2}E[c_{i*}]\ln\frac{z_{\max}^2}{2\pi\sigma^2} - \frac{1}{2}\sum_j E[c_{ij}]\frac{d^2(z_i,\theta_j)}{\sigma^2} \tag{14}$$

This equation is the basis for the EM algorithm for maximizing the log-likelihood described in turn.

4 Likelihood Maximization Via EM

The expected log-likelihood (14) is maximized using EM, a popular method for hill climbing in likelihood space for problems with latent variables [2]. In essence, EM generates a sequence of maps, $\theta^{[0]}, \theta^{[1]}, \theta^{[2]}, \ldots$. Each map improves the log-likelihood of the data over the previous map until convergence. 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 $E[c_{ij}]$ and $E[c_{i*}]$ are calculated for the n-th map $\theta^{[n]}$, and an M-step, where a new maximum likelihood map $\theta^{[n+1]}$ is computed under these expectations.

The E-Step: In the E-step, we are given a map $\theta^{[n]}$ for which we seek to determine the expectations $E[c_{ij}]$ 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):

$$\begin{aligned} E[c_{ij}] &= p(c_{ij}|\theta^{[n]}, z_i) \\ &= \frac{p(z_i|\theta^{[n]}, c_{ij})p(c_{ij}|\theta^{[n]})}{p(z_i|\theta^{[n]})} \\ &= \frac{e^{-\frac{1}{2}\frac{d^2(z_i,\theta_j)}{\sigma^2}}}{e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2}\frac{d^2(z_i,\theta_j)}{\sigma^2}}} \end{aligned} \tag{15}$$

$$E[c_{i*}] = \frac{e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}}}{e^{-\frac{1}{2}\ln\frac{z_{\max}^2}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2}\frac{d^2(z_i,\theta_k)}{\sigma^2}}} \tag{16}$$

The M-Step: In the M-step, we are given the expectations $E[c_{ij}]$ and seek to calculate a map $\theta^{[n+1]}$ that maximizes the expected log-likelihood of the measurements, as given by Equation (14). In other words, we seek surface parameters $\langle\alpha_j, \beta_j\rangle$ that maximize the expected log likelihood of the map.

Obviously, many of the terms in (14) do not depend on the map parameters θ. This allows us to simplify this expression and instead minimize

$$\sum_i\sum_j E[c_{ij}]d^2(z_i, \theta_j) \tag{17}$$

The actual M-step proceeds in two steps. First, the parameters α_j and β_j are computed that specify the principal orientation and location of the rectangular surface, without considerations of the surface boundaries. If walls are assumed to be boundless, minimizing (17) is equivalent to minimizing

$$\sum_i\sum_j E[c_{ij}](\alpha_j \cdot z_i - \beta_j)^2 \tag{18}$$

subject to the normality constraint $\alpha_j \cdot \alpha_j = 1$. This quadratic optimization problem is commonly solved via Lagrange multipliers λ_j for $j = 1, \ldots, J$:

$$L := \sum_i\sum_j E[c_{ij}](\alpha_j \cdot z_i - \beta_j)^2 + \sum_j \lambda_j\alpha_j \cdot \alpha_j \tag{19}$$

This problem is easily solved using eigenvalue decomposition, as described in [3, 6].

Our algorithm proceed by calculating the bounding boxes γ_j. It does so by determining the minimum rectangular box on the surface which includes all points whose maximum likelihood assignment is the j-th surface: θ_j:

$$\{i \text{ such that } i = \underset{k\in\{1,\ldots,k,*\}}{\text{argmax}} E[c_{kj}]\} \tag{20}$$

The optimization searches a finite number of possible surface orientations and then calculates the minimum bounding box for this orientation. Finally, the bounding box with the smallest enclosed surface is selected. This step leads to a rectangular surface of small size, but which nevertheless contains all measurements that most likely correspond to the surface at hand.

Model Selection: In addition to computing the surface parameters, the number of surfaces J has to be determined. Our approach involves a straightforward Bayesian prior that penalizes complex maps using an exponential prior, written here in log-likelihood form:

$$p(\theta|Z) \quad \propto \quad p(Z|\theta) - \kappa J \tag{21}$$

Here κ is a constant factor (e.g., 0.02). The final map estimator is, thus, a maximum posterior probability estimator (MAP), which combines the complexity-penalizing prior with the data likelihood calculated by EM. In practice, this approach implies that surfaces that are not supported by sufficiently many data measurements weighted by their expectation, are discarded. This approach makes it possible to choose the number of map components J (rectangular surfaces) dynamically, during the execution of EM.

Texture Mapping: Textures are extracted from the panoramic camera, along a stripe shown in Figure 1b 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 described in [10]. Textures of the same feature in the environment recorded at different points in time are presently not merged due to tight computational constraints—which should be considered a shortcoming of our present implementation.

5 Online EM

Online mapping is achieved through several modifications of the basic EM algorithm, which ensure that the computation at each time is independent of the data set size, and that leverages past optimization as much as possible when constructing a new map.

In the E-step, the expectation is only calculated for a small (and bounded) number of measurements, in a way that the total number of such calculations does not depend on the amount of available data. Our approach calculates expectations for all current measurements. In addition, expectations are recalculates for older measurements which meet several conditions: They lie at the boundary of two surfaces (judging from their maximum likelihood assignment), or are entirely unexplained by any existing surface *and* they have been considered no more than a fixed number of times in previous E-steps. The latter condition assures that measurements cannot be considered infinitely often in the E-step, hence bounds the number of measurements that can be considered in any E-step. The number of surfaces considered in the E-step is also bounded and typically includes all nearby surfaces.

In the M-step, only surfaces are re-estimated whose maximum likelihood assignments were changed in the E-step. This nat-

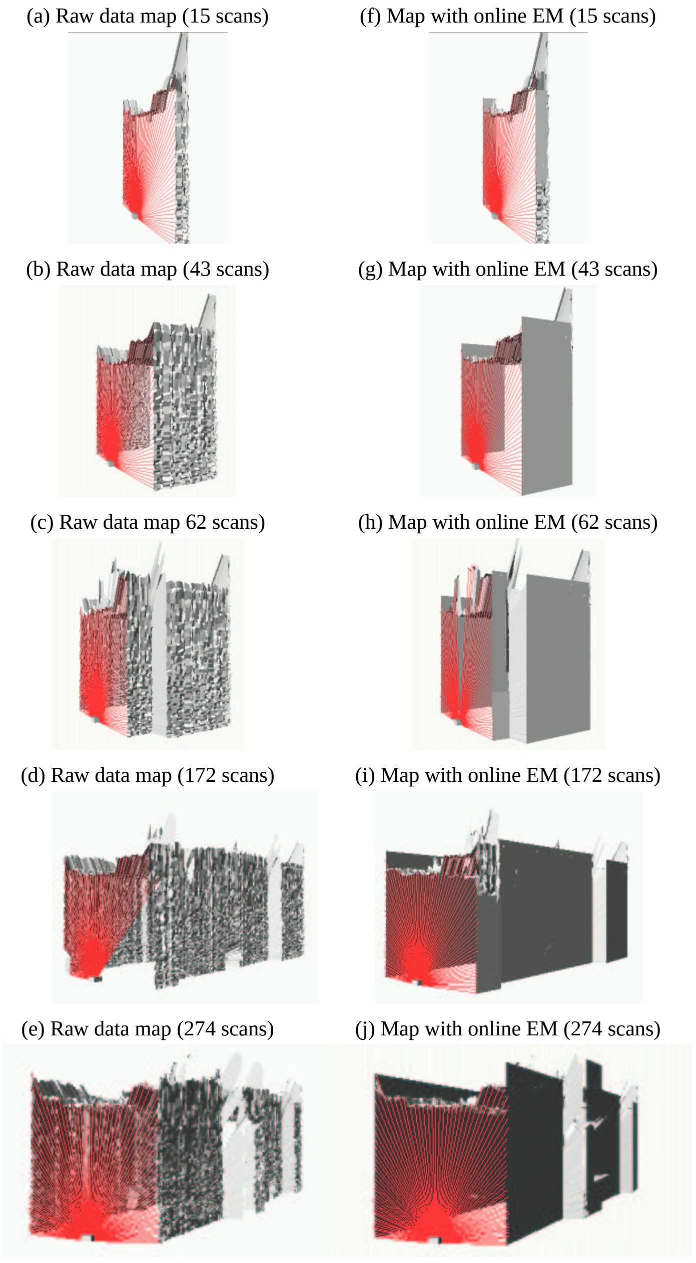
(a) Raw data map (15 scans)
(f) Map with online EM (15 scans)
(b) Raw data map (43 scans)
(g) Map with online EM (43 scans)
(c) Raw data map 62 scans)
(h) Map with online EM (62 scans)
(d) Raw data map (172 scans)
(i) Map with online EM (172 scans)
(e) Raw data map (274 scans)
(j) Map with online EM (274 scans)

Figure 2: Raw data (left column) and maps generated using online EM (right column). Some of the intermediate maps exhibit suboptimal structures, which are resolved in later iterations of EM. Despite this backwards correction of past estimates, the algorithm presented in this paper still runs in real-time, due to careful selection of measurements that considered in the EM estimation.

urally guarantees that only a small number of surfaces is recalculated in the M-step, with the total number being independent of the total number of surfaces J. However, each such surface might still be supported by an unbounded number of measurements, hence the calculation of surface parameters may take time that grows with the total number of measurements. To avoid the latter, the number of measurements involved in re-computing surface parameters in the M-step is subsampled so as to not to exceed a fixed number. This is necessary to guarantee the incrementally of the EM algorithm.

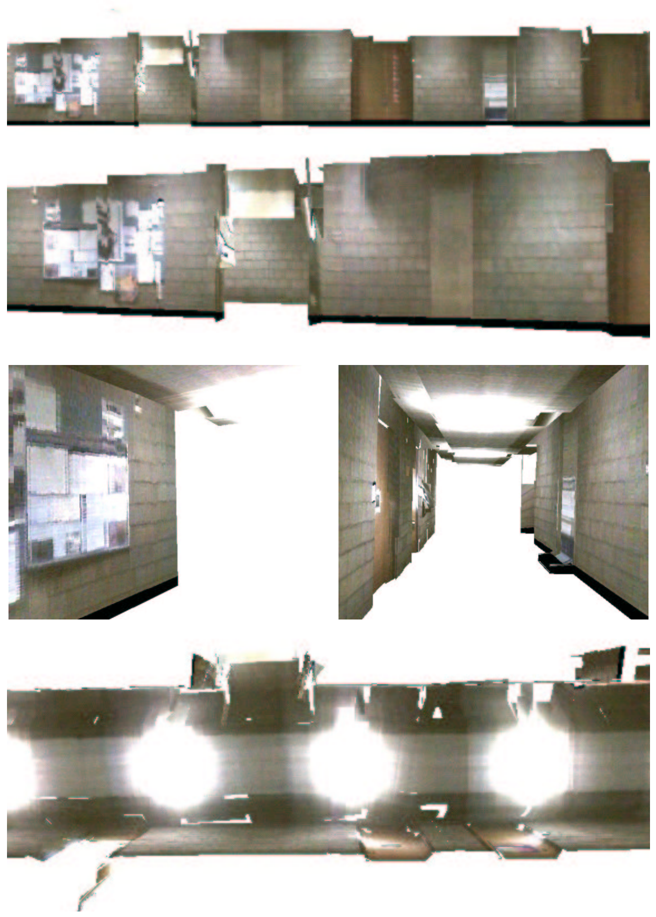

Figure 3: Views of a compact 3D texture map built in real time with an autonomously exploring robot.

To guide the search in the likelihood space, the map generated at time $t-1$ is used as initial map at time t. This greatly reduces the number of iterations required at any point in time.

Finally, our approach implements Bayesian model selection in real-time. New surfaces may be introduces based on new measurements which are not "explained" (in an ML fashion) by any of the existing surfaces in the map. Surfaces are removed form 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 (see Equation (21)).

6 Experimental Results

Our online EM algorithm for 3D multi-planar mapping was successfully tested in several corridor-style environments, some of which are shown in various figures in this paper. All results shown in this paper were generated in real-time, in some cases using an autonomously exploring robot, in others using a manually controlled robot.

Figure 2 illustrates the online creation of a compact map. Shown in the left column are maps generated directly from the raw measurement data by creating polygons for any set of nearby measurements. Views of a map with full texture are shown in Figure 3. This map, including the texture information, was entirely built in real-time. A comparison of raw data maps, the flat surfaces extracted in real-time, and the final texture maps is shown in Figure 5. Figure 4 shows the structure of a non-planar object (a trash bin) in one of the final maps. This example illustrates that the ability to merge flat surfaces with fine-grained polygons makes it possible to build complete maps of building interiors while exploiting the building's planarity.

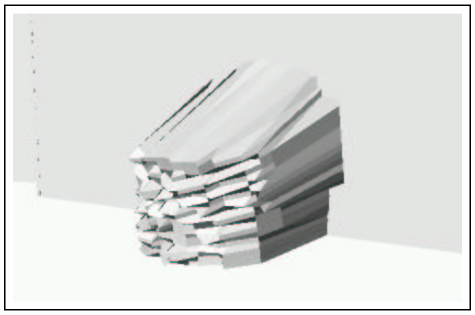

Figure 4: View of a trash bin in the final map, with a large planar rectangular surface patch in the background. Our algorithm recognizes that this object cannot be explained by planar surfaces with sufficient likelihood, hence it retains the polygonal representation.

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 $1:192$.

7 Discussion

We have presented an online algorithm for building compact maps of building interiors. This approach utilizes the popular *expectation maximization* algorithm for finding rectangular surface patches in 3D 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 maps of corridor-style environments.

Although stated here in the context of finding rectangular flat surfaces, the EM algorithm is more general in that it can easily handle a much richer variety of geometric shapes. The extension of our approach to richer classes of objects is subject to

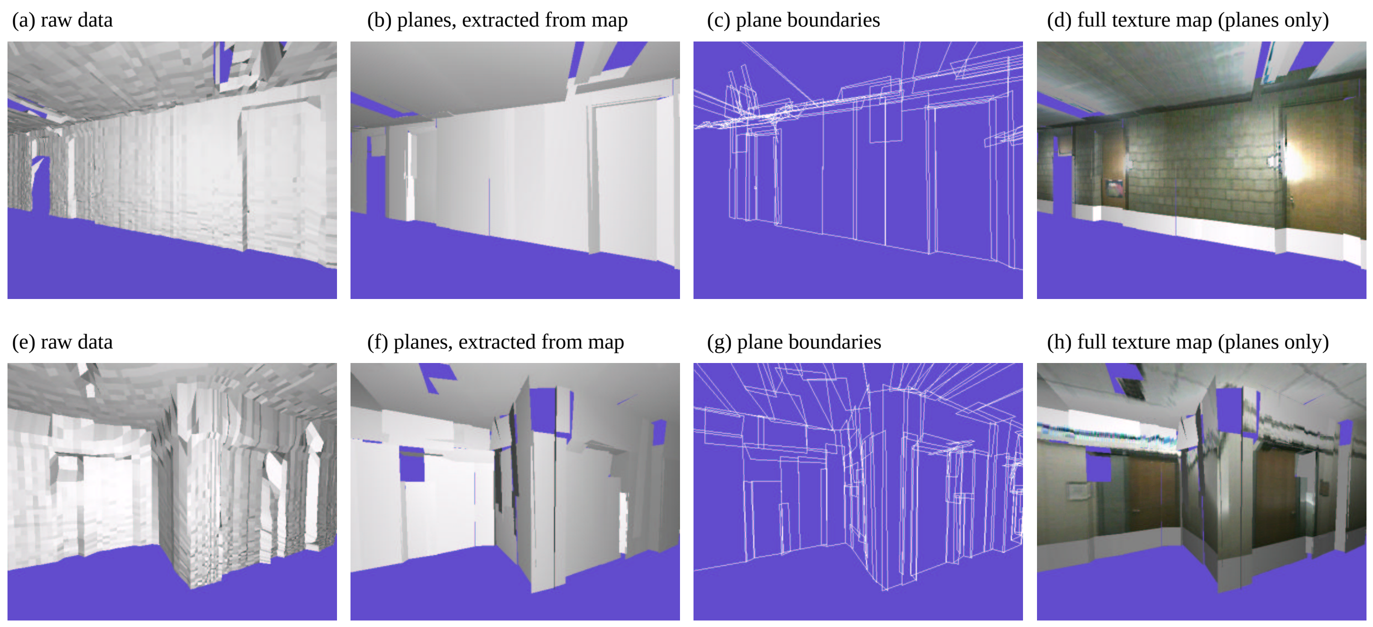
(a) raw data
(b) planes, extracted from map
(c) plane boundaries
(d) full texture map (planes only)
(e) raw data
(f) planes, extracted from map
(g) plane boundaries
(h) full texture map (planes only)

Figure 5: Maps generated in real-time, of office environment at Carnegie Mellon University (top row) and Stanford University (bottom row).

future research. Another topic of future research is to augment the EM algorithm to estimate the robot's location during mapping, as described in [13].

Acknowledgment

This research is sponsored by by DARPA's MARS Program (Contract number N66001-01-C-6018) and the National Science Foundation (CAREER grant number IIS-9876136 and regular grant number IIS-9877033), all of which is gratefully acknowledged. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the United States Government or any of the sponsoring institutions.

References

[1] R. Chatila and J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots," In *Proc. of ICRA-1985*.

[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*, vol. 39, no. 1, pp. 1–38, 1977.

[3] D. Eberly, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*, Morgan Kaufman, 2001.

[4] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, RA-3, pp. 249–265, 1987.

[5] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," In *Proc. of CIRA-2000*.

[6] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun, "Using EM to learn 3D models with mobile robots," in *Proc. of ICML-2001*.

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

[8] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley, New York, 1997.

[9] H.P. Moravec and M.C. Martin, "Robot navigation by 3D spatial evidence grids," Internal Report, CMU, 1994.

[10] S.K. Nayar, "Omnidirectional vision," In *Proc. of ISRR-1997*.

[11] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," In *Proc. of AAAI-2000*.

[12] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

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