International Journal of Advances in Electrical Engineering

E-ISSN: 2708-4582
P-ISSN: 2708-4574
IJAEE 2022; 3(2): 33-47

www.electricaltechjournal.com
Received: 07-04-2022
Accepted: 07-05-2022

Adetunmbi Adewole Oyewale
Department of Electrical and Electronic Engineering, Federal Polytechnic Ile-Oluji Ondo State, Nigeria

Senboyejo Temitope Anthony
Department of Systems Engineering, University of Lagos, Lagos State, Nigeria

Okeke Onyedika Remigius
Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State, Nigeria

Correspondence
Adetunmbi Adewole Oyewale
Department of Electrical and Electronic Engineering, Federal Polytechnic Ile-Oluji Ondo State, Nigeria

Development of an intelligent algorithm for acquiring multiplanar maps in mobile robots

Adetunmbi Adewole Oyewale, Senboyejo Temitope Anthony and Okeke Onyedika Remigius

Abstract
This paper presents a real-time technique for creating small, three-dimensional maps of indoor spaces utilising a mobile robot fitted with imaging and range sensors. Our method expands on earlier work on real-time pose estimation during mapping by applying the well-known expectation-maximisation algorithm to multisurface models and enabling real-time execution. Compact collections of textured polygons that may be interactively visualised make up the maps that our method acquires. Experiments conducted in corridor-like settings showed that accurate and compact maps could be acquired in real-time and with complete automation.

Keywords: Robotic mapping, mobile robots, perception, statistical techniques

1. Introduction
This Paper demonstrates a real-time system for creating three-dimensional (3-D) maps of indoor areas using range and camera data collected by a mobile robot. Numerous indoor mobile robots rely on environment maps for navigation (Kortenkamp *et al.*, 1998) [20] Most available techniques for getting these maps to work are in two dimensions (2-D). Given that most interior mobile robots are restricted to 2-D planes, 2-D maps may seem adequate for navigation. Nevertheless, there are two significant benefits to modelling an environment in three dimensions: First, because 3-D models are more detailed than 2-D models, 3-D maps make it easier to distinguish between different locations. Secondly, 3-D maps are more useful for users interested in a building, such as architects or human rescue workers who want to know a new area.

Manuscript received November 23, 2018; revised May 5, 2003. This paper was recommended for publication by Associate Editor G. Oriolo and Editor S. Hutchinson upon evaluating the reviewers' comments. This work was partly supported by the Defense Advanced Research Projects Agency (DARPA) MARS Program under Contract N66001-01-C-6018 and Contract.
NBCH1020194, partly 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, 2018, and in part at the International Conference on Machine Learning, Williamstown, MA, 2019.
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).
Building 3-D modelling has long been a goal of computer vision researchers (Bajcsy *et al.*, 1998; Debevec *et al.*, 1996; Shum *et al.*, 1998; Teller *et al.*, 2019) [3, 9, 32, 33] for these and other reasons. Such robot-generated maps would allow for the acquisition of maps of areas that are inaccessible to humans (Casper, 2018) [5], (Montemerlo *et al.*, 2018) [26] such as recently 3-D- mapped abandoned mines (Thrun, 2003) [34].

Moving from 2-D to 3-D is not only a simple extension of robotic mapping. Currently, occupancy maps (Elfes, 2019, (Moravec, 1988) [13, 27], which depict environments using fine-grained grids, are the most widely used paradigm in 2-D mapping. While this is possible in 2-D, there are major scaling limits in 3-D due to the complexity of these representations (Moravec, 1988) [27]. Point clouds, line segments, and other common 2-D representations are available (Gutmann and Konolige, 2019, Lu and Milios, 1997, and Chatila, 2020) [18, 23, 6] By representing maps as collections of fine-grained polygons, line representations have been expanded to encompass three dimensions (Thrun, 2019) [35], (Montemerlo *et al.*, 2018) [26]. Off-the-shelf computer graphics techniques for mesh simplification (Garland and Heckbert,, 1998) [16] frequently produce aesthetically incorrect maps (Thrun, 2019) [35], and the resulting maps are frequently highly complicated.

This study provides an algorithm for low-complexity 3-D model recovery using range and camera data that especially uses prior information about the geometry of fundamental construction components. Our method specifically fits a robot-collected data set to a probabilistic model made up of big rectangular, flat surfaces. Small polygons represent sections in the map that cannot be well represented by flat surfaces as in (Montemerlo *et al.*, 2018) [26], allowing our method to account for nonflat parts of the environment. Compared to the maps produced by the earlier methods mentioned above, the ones that result are less complicated. Additionally, switching to a low-complexity model has the added benefit of reducing the noise in the produced maps, a byproduct of the variance reduction achieved by fitting low-complexity models.

The method described here employs a real-time form of the expectation-maximisation (EM) technique to find low-complexity models (Dempster, 1977) [10], (McLachlan and Krishnan, 1997) [25]. Our technique evaluates the number of surfaces and their positions concurrently. We use fine-grained polygonal representations to map measurements that cannot be described by any surface, allowing us to simulate nonplanar environmental phenomena. The final map is shown using the Virtual Reality Markup Language (VRML) format, with a panoramic camera's texture overlay.

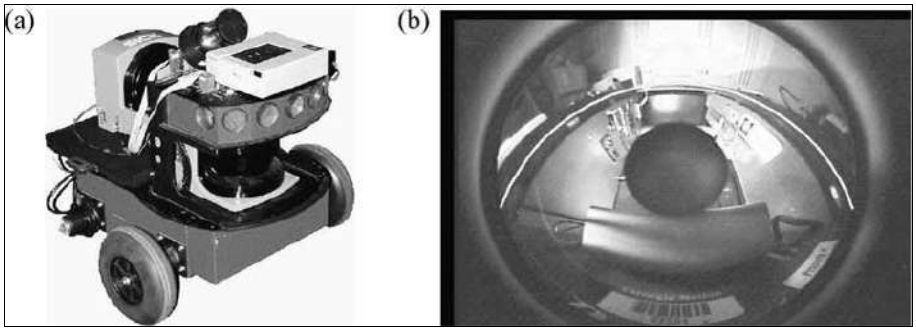

Fig 1: Mobile robot equipped with two 2-D laser range finders and a panoramic camera. The camera uses a panoramic mirror mounted only a few centimetres away from the optical axis of the laser range finder.

Our strategy is based on two major presumptions. It begins by assuming that a reliable estimation of the robot's position is known. In-depth research has been done on the problem of pose estimation (localisation) in mapping in the robotics literature (Leonard, *et al.* 2018) [21]. Our assumption is not implausible because we employ a real-time approach like the one described in (Thrun, 2019) [35] in all of our trials to estimate posture, allowing us to concentrate on the 3-D mapping parts of our research. Second, we presume that most of the environment comprises flat surfaces. The fundamental operations of our EM algorithm may be conveniently solved in closed form thanks to the flat-surface assumption. Indoor spaces frequently have flat surfaces, particularly in hallways. Additionally, our system keeps measurements that any flat surface cannot approximate, translating into more accurate polygonal approximations. As a result, the final map can have nonflat sections in places that are not so flat.

Utilising the mobile robot in Fig. 1, our strategy has been fully accomplished (a). This robot has three types of range finders: a forward-pointing laser range finder for localisation during mapping, an upward-pointing laser range finder for structural mapping, and a panoramic camera for capturing the texture of the environment [see Fig. 1(b)]. The technology has undergone testing in a variety of structures. Our findings show that the algorithm efficiently produces quick and precise 3-D maps.

2. Generative Probabilistic Model

A. World Model

Our method uses sets of tiny polygons to depict nonflat surfaces and rectangular flat surfaces to represent doors, walls, and ceilings. We will denote the set of rectangular flat surfaces by, where θ

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

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 $\alpha_j and \beta_j$.

B. Measurements

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

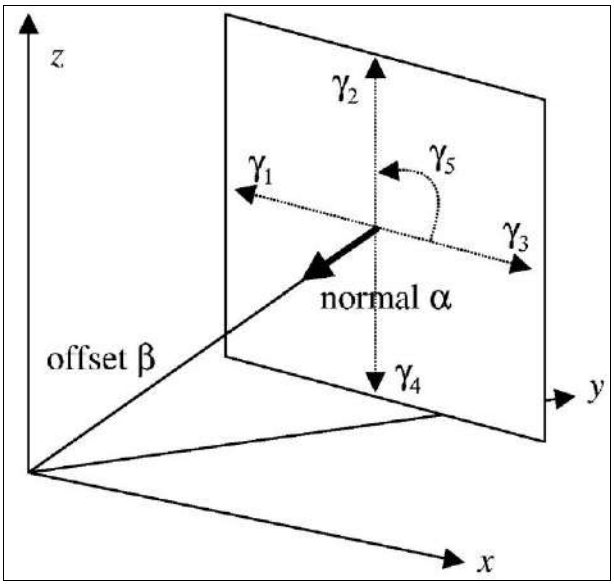

Fig 2: Illustration of the parameters in the planar surface model, shown here for one surface.

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

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

We denote the set of all measurements by

$$Z = \{z_i\}. \tag{4}$$

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

$$d(z_i, \theta_j) \tag{5}$$

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, $d(z_i, \theta_j)$ it is given by $\alpha_j.z_i - \beta_j$; 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

$$c_i = \{c_{i*}, c_{i1}, c_{i2}, ..., c_{ij}\}. \tag{6}$$

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} ($(for 1 \le j \le j)$) 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 on the map, the "special" correspondence variable c_{i*} is 1. This might be the case because of random measurement noise or 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

$$c_{i*} + \sum_{j=1}^{J} c_{ij} = 1 \tag{7}$$

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 to the world.

$$p(z_i \mid C_i, \theta) \tag{8}$$

C_i is the correspondence vector of the ith 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:

$$p(z_i \mid c_{i*} = 1, \theta) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{d^2(z_i, \theta_j)}{\sigma^2}} \tag{9}$$

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

The normally 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

$$p(z_i \mid c_{i*} = 1, \theta) := \left\{ \begin{matrix} 1/z_{\max}, \\ 0 \end{matrix}, if\ 0 \le z_i \le z_{\max} \right\} \tag{10}$$

The interval $[0; z_{\max}]$ denotes the measurement range of the range finder. The uniform noise model is 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:

$$\frac{1}{z_{\max}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\log\frac{z^2 \max}{2\pi\sigma^2}} \tag{11}$$

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

3. Log-likelihood function

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

$$p(z_i \mid C_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left[c_{i*}\log\frac{z^2 \max}{2\pi\sigma^2} + \sum j^{\,c_{ij}\frac{d^2(z_i, \theta_j)}{\sigma^2}}\right].} \tag{12}$$

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.

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

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

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

This equation is simply the product of (13) overall measurements z_i. In EM, it is common practice to maximise the log-likelihood instead of the likelihood (14), exploiting the fact that the logarithm is monotonic in its argument.

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

Finally, while the formulas above all compute a joint over map parameters *and* correspondence, we are all interested in 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 maximise the *expectation* of the log-likelihood (15), where the expectation is taken over all correspondences *C*. This value, denoted $E_C\left[\log p(Z, C \mid \theta)\right]$, is the expected log-likelihood of the data, given the map with the correspondences integrated out. It is obtained directly from (15)

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

In (Neal and Hinton, 1998) [29], it is shown that this expectation indeed maximises the log-likelihood of the data.

4. Likelihood maximisation Via EM

The expected log-likelihood (16) is maximised using EM, a popular method for hill climbing in likelihood space for problems with latent variables [2]. EM generates a sequence of maps, $\theta^{[0]}, \theta^{[1]}, \theta^{[2]},$ 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 $E\left[c_{ij} \mid \theta^{[n]}, z_i\right]$ $E\left[c_{i*} \mid \theta^{[n]}, z_i\right]$ are calculated for the nth $\theta^{[n]}$ map, 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 $E\left[c_{ij} \mid \theta^{[n]}, z_i\right]$ $E\left[c_{i*} \mid \theta^{[n]}, z_i\right]$ 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)

$$e_{ij}^{[n]} := E\left[c_{ij} \mid \theta^{[n]}, z_i\right] = p\left(c_{ij} \mid \theta^{[n]}, z_i\right) = \frac{p\left(z_i \mid \theta^{[n]}, c_{ij}\right) p\left(c_{ij} \mid \theta^{[n]}\right)}{p\left(z_i \mid \theta^{[n]}\right)}$$

$$= \frac{e^{-\frac{1}{2}\frac{d^2(z_i,\theta_j)}{\sigma^2}}}{e^{-\frac{1}{2}\log\frac{z^2 \max}{2\pi\sigma^2}} + \sum_k e^{-\frac{1}{2}\frac{d^2(z_i,\theta_k)}{\sigma^2}}} \quad (17)$$

And similarly

$$e_{i*}^{[n]} := E\left[c_{i*} \,\middle|\, \theta^{[n]}, z_i \right] = \frac{e^{-\frac{1}{2}\log_{2\pi\sigma^2} z^2 \max}}{e^{-\frac{1}{2}\log_{2\pi\sigma^2} z^2 \max} + \sum_k e^{-\frac{1}{2}\frac{d^2(z_i,\theta_k)}{\sigma^2}}}. \tag{18}$$

As pointed out in, (McLachlan and Krishnan, 1997) [25] and (Neal and Hinton, 1998) [29], substituting these expectations into the log-likelihood (16) lower bounds the log-likelihood by a function tangent to it $\theta^{[n]}$.

B. The M-Step

In the M-step, this lower bound is optimised. More specifically, we are given the expectations $e_{ij}^{[n]}$ $e_{i*}^{[n]}$ and seek to calculate a map $\theta^{[n+1]}$ that maximises the expected log-likelihood of the measurements, as given by (16). In other words, we seek surface parameters $\left[a^{[n+1]}, \beta^{[n+1]}, \gamma^{[n+1]} \right]$ that maximise the expected log-likelihood of the map under fixed expectations $e_{ij}^{[n]}$ and $e_{i*}^{[n]}$.

Many terms in (16) do not depend on the map parameters. This allows us to simplify (16) and instead carry out the following minimisation:

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

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

$$\left[\alpha^{[n+1]}, \beta^{[n+1]} \right] = \arg\min_{\alpha,\beta} \sum_i \sum_j e_{ij}^{[n]} \left(\alpha_j . z_i - \beta_j \right)^2 \tag{20}$$

This quadratic optimisation problem is commonly solved via Lagrange multipliers λ_j for $j = 1, \ldots, J$ [20, 21]

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

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

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

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

$$\alpha_j^{[n+1]} . \alpha_j^{[n+1]} = 1. \tag{24}$$

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

$$\beta_j^{[n+1]} = \frac{\sum_i e_{ij}^{[n]} \alpha_j^{[n+1]} .. z_k}{\sum_k e_{kj}^{[n]}} \tag{25}$$

Substituting those back into (22) gives us)

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

This is a set of linear equations of the type

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

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 $A_j^{[n]}$. 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 $\alpha_j^{[n+1]}$.

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 jth surface θ_j

$$\{z_i,\} \text{ such that } \begin{Bmatrix} j = \arg\max e_{i,k}^{[n]} \\ k \in 1,...,J,* \end{Bmatrix} \quad (29)$$

There is no simple closed-form solution to this optimisation issue. Our method calculates the tightest bounding box for each direction after probing it at one-degree intervals. The final choice is the bounding box with the least contained surface volume. This process produces an almost optimum rectangular surface with all measures that most likely match the current surface.

C. Determining the Number of Surfaces

Our method establishes the number of surfaces concurrently with computing the surface characteristics. Our method uses an easy-to-understand Bayesian prior that penalises complicated maps with an exponential prior, shown below in the log-likelihood form.

$$p\left(\theta \mid d \mid \alpha p\left(d \mid \theta \mid \leftrightarrow k <\right)\right) \quad (30)$$

Is a constant in this situation. In order to integrate the complexity penalising prior with the data likelihood determined by EM, the final map estimator is a maximum a posteriori (MAP) probability estimator. According to this method's practical application, surfaces not supported by an adequate number of data measurements (weighted by their expectation) are disregarded. As a result, selecting the number of rectangular surfaces while the EM algorithm is being run is possible.

The EM algorithm execution and the quest for the best are interspersed in our implementation. The search contains two steps carried out at regular intervals: one for terminating surfaces and one for building new surfaces (every 20 iterations in our offline implementation). New surfaces are formed during the surface creation process using measurements that the current model does not adequately account for. If the measurement value exceeds a certain threshold, it is said that the measurement is not adequately described by any of the flat surfaces in the model. Suppose three adjacent observations cannot be satisfactorily explained. In that case, a new surface is created, and the starting parameters of this new surface are then uniquely defined using the coordinates of these three measurements. Suppose the total number of candidate surfaces is more than the limit. In that case, a random selection is made from all candidates to maintain the maximum number of new surfaces at each iteration. In the next rounds of EM, the new surfaces are included in the model and treated in the same manner as the previous surfaces.

Each surface is evaluated posteriorly in the last stage using the criteria outlined in (30). The same action is conducted if removing a surface or merging it with another surface close by raises the posterior in (30). If not, it is kept in the model. In this method, over fitting that would unavoidably happen without a complexity penalty term is prevented by only surfaces supported by sufficient data points making it through the selection process.

D. Texture Mapping and Visualization

Textures are pulled out of the panoramic camera along a stripe in Fig. 1(b) corresponding to the vertical laser range finder's distance measurement. Raw texture maps are created by combining these stripes at frame rate. Then, using a method similar to

the one described in (Nayar, 1997), these maps are overlapped on the planar surfaces in real-time. Weak computational limitations prevent us from currently merging textures of the same feature in the environment that were captured at separate times in time, which is a glaring flaw in our current solution.
All surfaces are included in the final model. However, it is missing information on the environment's non flat items. Fine-grained polygonal representations of these nonfat things enhance the final 3-D depiction of the surroundings. Our method, in particular, evaluates all measures whose class has its most probable relationship. The final 3-D display adds a tiny triangle for every instance of three similar close measurements. In this approach, the visualization includes fine-grained polygonal representations of non flat objects in the surroundings and massive flat surfaces. Our tool generates a VRML file as its final result, allowing third-party software to render the 3-D model.

V. Online EM

With the help of an online algorithm built on top of our methodology, robots can now acquire small 3-D models instantly (Martin, C., and S. Thrun, 2018). In its current form, EM is fundamentally offline. Multiple iterations through the data set were necessary. The calculation of each EM iteration increases as the size of the data collection increases. The vanilla EM algorithm has this drawback often (Neal and Hinton, 1998) [29].
The crucial realisation of an online implementation is that only a defined number of fresh range measurements come at each fixed period (collectively referred to as). If we start our estimation with the model we received from the initial data, the model must continually be updated to account for new measurement values as they are observed. Such a process would be strictly incremental but would not modify correspondence variables in light of information obtained later. Our method is a little more advanced because it modifies previous expectations but complies with the constant update time.

A. Online E-Step

The online E-step only computes expectations for the correspondence variables of those data and only considers a restricted fraction of all measurements. It specifically covers all newly discovered range measurements, of which there is a finite number. It also contains earlier readings that satisfy the next requirement. They are either completely unrelated to any existing surface or, based on their ML assignments, located at the intersection of two surfaces. These prerequisites by themselves cannot guarantee continuous-time computation. In order to keep track of how many times each measurement was taken into account in an E-step computation, we added a counter to each measurement. The predicted correspondence value is locked and never again calculated if this counter rises beyond a predetermined level. The final requirement provides consistent time updating (in expectation). The former circumstance finds "interesting" measurements, which significantly lowers the constant component of computational complexity.

B. Online M-Step

The M-step online is a little trickier. This is because both the number of model components and the number of measurements linked to each model component increase as the size of the data collection increases. Each of these expansions must be regulated to ensure that the M-step can be carried out in real-time.
Only a few "active" surfaces are re-estimated in the online M-step. If the E-step conducted at that time moved the ML correspondence to or away from the surface for any of the updated data, we say that a surface is active at that moment. This is an approximation because, theoretically, changing the anticipated correspondence will change all model parameters. Though our method is a decent approximation, it only considers a fixed number of surfaces in the M-step.
Finally, our strategy takes into account the fact that each maximising involves an increasing amount of points. Every measurement in traditional EM contributes to the computation of every surface parameter, but the majority do so insignificantly. Our online technique only considers measurements whose ML data connection matches the questioned model component to minimise the number of measurements required in the M-step. This is comparable to the well-known clustering technique k-means (Duda and Hart, 1973) which uses hard data to approximate the EM process. 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.

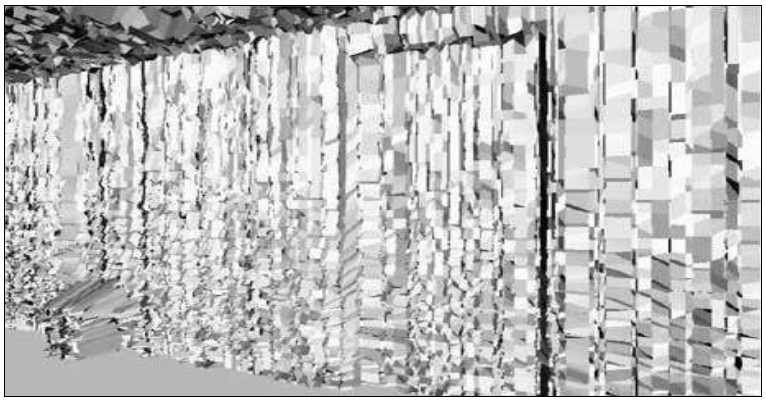

Fig 3: Polygonal map generated from raw data, not using EM. The display without texture shows the level of noise involved. In particular, it illustrates the difficulty separating the door from the nearby wall (as achieved by EM).

C. Online Model Selection

Finally, our method simultaneously applies Bayesian model selection. For new observations that no current surface in the map can "explain," new surfaces are added. If a measurement's likelihood of coming from an existing surface exceeds a certain threshold, that measurement is used to explain the surface; otherwise, it is used to seed a new surface. Surfaces are deleted from the map in line with the map complexity penalty factor described if, after a certain number of iterations, they are not supported by enough data (30). As a result, the number of surfaces rises as the environment becomes more complicated, yet no matter how big the entire map is, all calculations can still be done online.

The outcome is a purely incremental algorithm. It has been used on a low-end laptop PC, where it can instantly create small 3-D maps.

5. Experimental Results

Our tests were conducted in two steps. First, we put the offline EM algorithm into practice and assessed it. This assessment helped us to establish the fundamental capacity to extract huge planar surfaces using EM from complicated data sources. The online version of our method was used to conduct our second round of tests in real-time.

In these trials, all computing took place on the moving robot platform depicted in Fig. 1. As mentioned above, the robot has two SICK PLS laser range finders, one oriented forward and the other up, parallel to the robot's direction of travel. The SICK sensor has a one-degree angular resolution and a 180-degree field of view. The manufacturer describes the range accuracy as cm. The sampling rate in our experiments is approximately 5 Hz (Entire scans).

A. Offline EM

A data collection collected within a university building served as the basis for testing the offline version of EM. The data collection, gathered in about two minutes, comprises 168 120 range measurements and 3270 camera pictures. Our programme retrieved 3220950 pixels during a realignment phase, which precisely matched range measurements. Polygons were created by converging nearby scans. A detail of the generated map sans the texture is shown in Fig. 3. This graph demonstrates the high degree of noise in the raw data. The left column of Fig. 4 shows three map views with a superimposed texture.

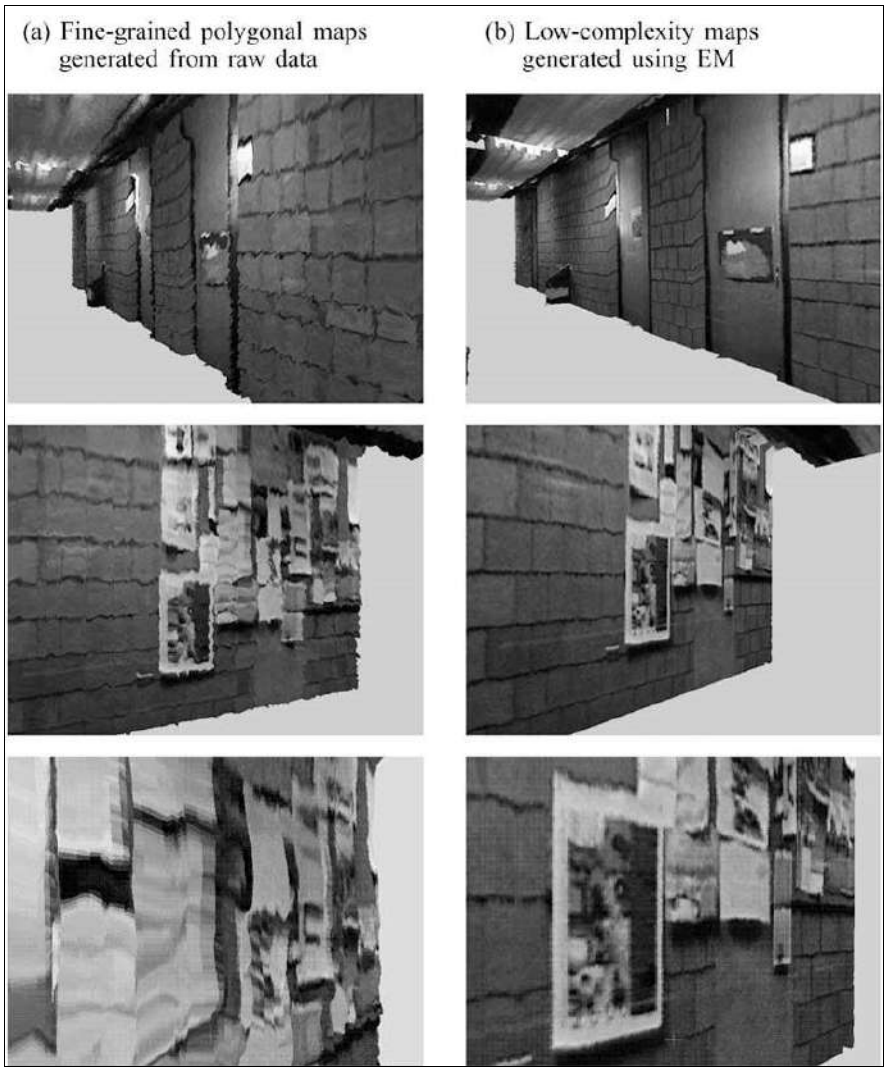

Fig 4: 3-D map. (a) Generated from raw sensor data. (b) Generated using the offline version of EM. This map explains 94.6% of all measurements by seven surfaces. Notice that the map in (b) is smoother and appears visually more accurate than the one in (a).

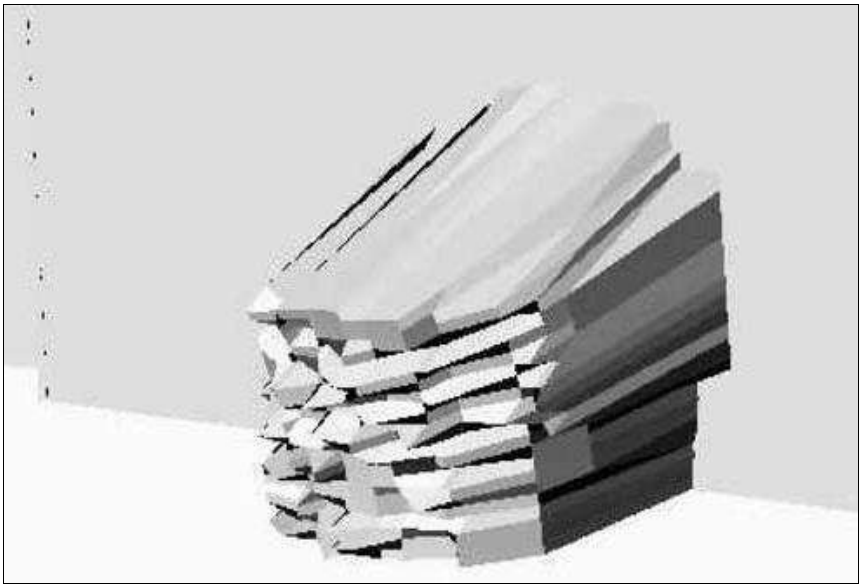

Fig 5: Model of a trash bin in the final map, with a large planar rectangular surface patch in the background. Our algorithm recognises that planar surfaces cannot explain this object with sufficient likelihood. Hence, it retains the fine-grained polygonal representation.

The right column of Fig. 4 displays the output of our EM method. Flat surfaces are present on this particular map, making up 94.6% of all measurements. The new map is undeniably more realistic and fluid. A garbage can in the top panel of that illustration, which is not depicted by a flat surface, demonstrates how effectively it mimics non planar locations. None of the surfaces has the necessary measurements mapped to them. The corresponding map segment is shown in Fig. 5, which highlights how crucial it is to combine flat surfaces with fine-grained polygons when modelling the interior of buildings. Observe how the surface of the wall differs from that of the door.

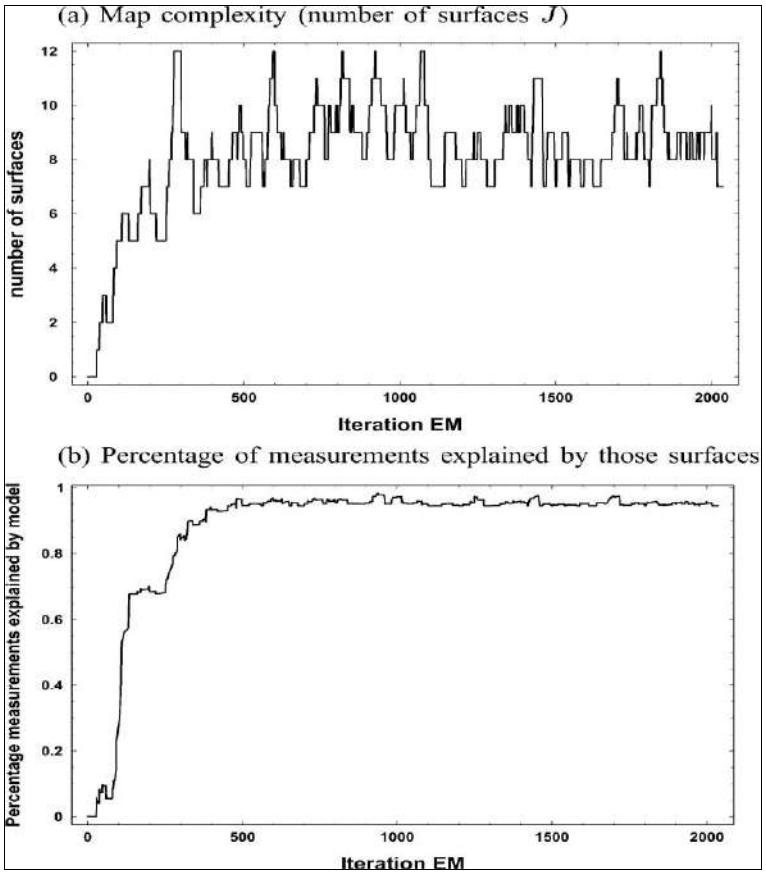

Fig 6: Offline EM. (a) Several surfaces [J]. (b) Percentage of points explained by those surfaces as a function of the iterator. A good map is available after 500 iterations.

A few measurements in the door surface are misclassified as belonging to the wall surface. The benefits of the EM technique over the raw data maps are again shown by the poster board displayed in various panels of Fig. 4. Because the texture is projected onto a planar model, it has better visual accuracy.

As a function of iteration, Fig. 6 displays the number of surfaces and the number of measurements that these surfaces explain. Surfaces are stopped and resumed approximately every 20 steps. A stable % of all measurements is explained by the number of surfaces stabilising at approximately a mean (and variance) after just 500 iterations. On a basic PC, 2019 iterations take around 20 minutes to compute.

B. Real-Time Implementation

Several buildings were used to examine the real-time online EM implementation further. Overall, our method was quite trustworthy for producing precise maps. Fig. 7 shows how tiny maps may be created online. Maps made from the raw measurement data are displayed in the left column, and they were made by drawing polygons around any cluster of close values. A series of maps created using our incremental EM technique is displayed in the right column. Although this particular Picture only illustrates vertical surfaces, our technology supports surfaces with any orientation.

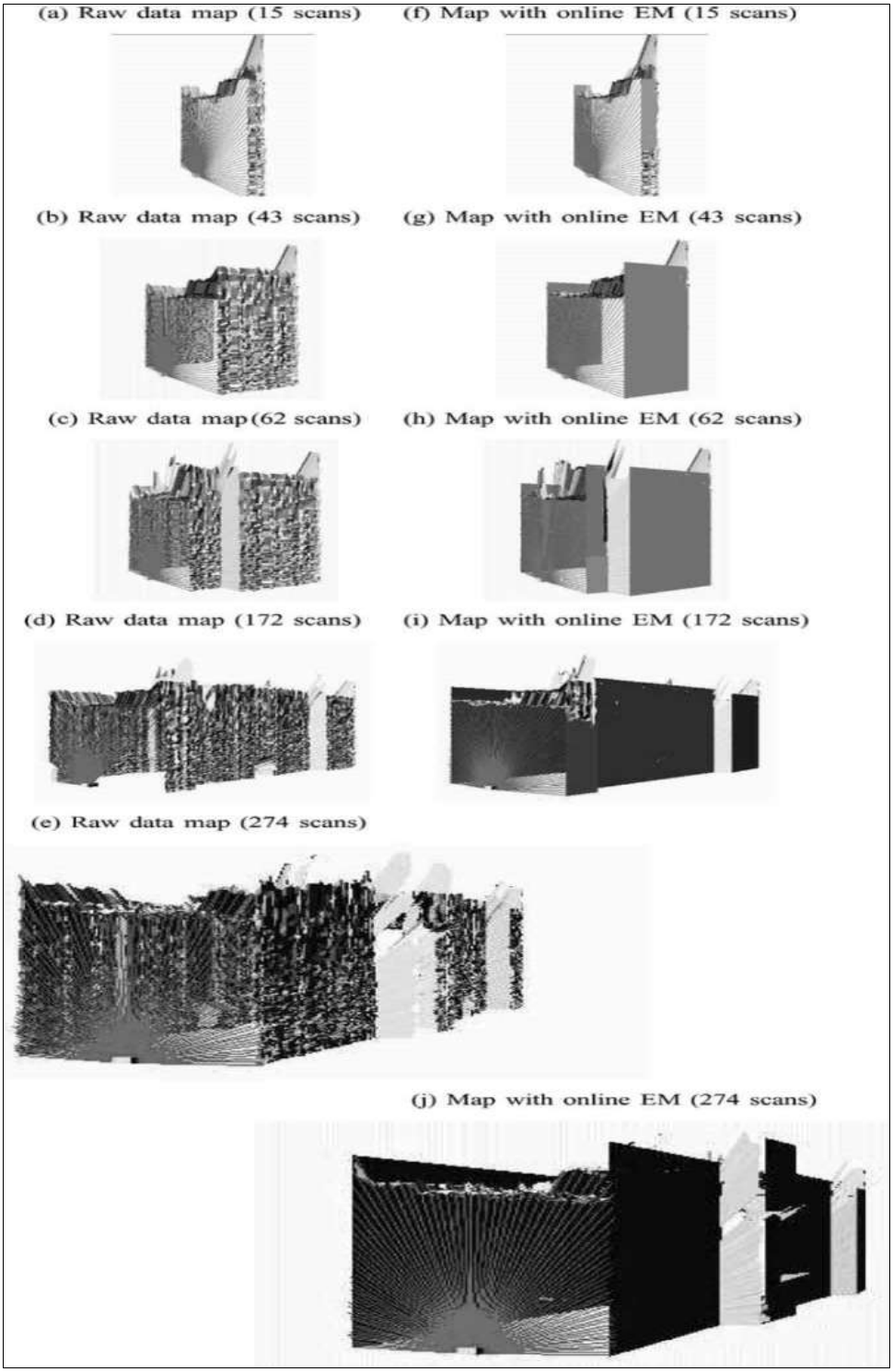

Fig 7: Raw data (left column) and maps generated using online EM (right column). The lines visualise the most recent range scan. Some intermediate maps exhibit suboptimal structures, resolved in later iterations of EM. Despite this backward correction of past estimates, the algorithm still runs in real-time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

The new map is smaller than the original data map. The enormous volumes of data may be modelled with a small number of surfaces. A close examination of Fig. 7 demonstrates EM in action. For instance, a little portion of Fig. 7(i) has not been classified as a flat wall merely because there is excessive noise on the rectangular surface underneath this patch. As seen in Fig. 7, further optimisation of this historical data results in a more condensed map (j).A small, 3-D map created in real-time for the same environment is shown in Fig. 8. Less than 2 minutes are needed for the complete data gathering procedure, which includes all processing.

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

Recognising rectangular surfaces in the surroundings has a favourable impact on the map's visual acuity. Fig. 9 compares a 3-D map constructed using EM to one constructed without it with fine-grained polygonal maps and data sets collected from two separate university buildings. The renderings produced from the fine-grained polygonal map have worse visual accuracy than the texture projected onto flat surfaces. This demonstrates once again that the final map is not only more compact but also offers more visual detail compared to the map made without our method.

We mapped three distinct corridor settings in various buildings to assess our methodology statistically. Those landscapes were just as complicated as the ones depicted on these maps. The original polygon count was between and. The finished maps had an average compression ratio of 1:192 and had, on average, 0.60% as many polygons (0.69%, 0.80%, and 0.32%). Overall, we discovered that compared to the offline version, the online version created orders of magnitude more flat surfaces. This rise in numbers was partly caused by a shorter penalty period[6].

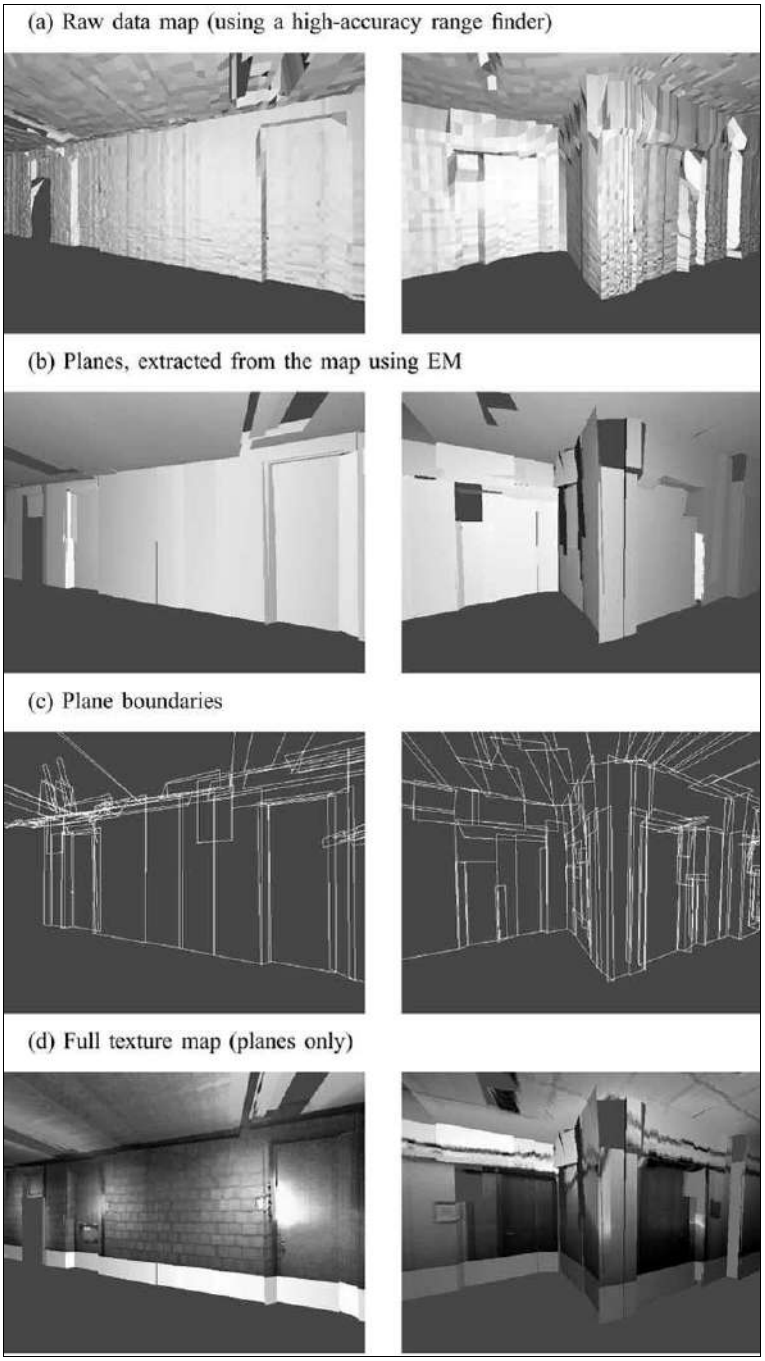

Fig 9: Maps generated in real time of office environments at Carnegie Mellon University's Wean Hall (left column) and Stanford University's Gates Hall (right column).

The online version's short-term consideration of splitting and fusion decisions contributed to the rise. Overall, the map's complexity decreased as the number of surfaces rose since surfaces discovered by EM could explain more observations.
Finally, we want to point out that the whole calculation for our experiment was done on a single laptop mounted to the robot. A typical programme for producing VRML models was used to produce the views in Figs. 7-9.

6. Related Work

The main thesis of this work is that most robot modelling research has concentrated on creating 2-D maps.
Our strategy is similar to early papers by Chatila and Laumond (2020) [6] and a related study by Crowley [2019], which claimed to rebuild low-dimensional line models in 2-D using sensor readings. Our work on line extraction from laser range scans is linked to ours (Lu and Milios, 1998) [24]. However, the 2-D scenario, where lines may be retrieved from a single scan, is addressed by these approaches.
The topic of object recognition under geometric limitations, such as the assumption of a flat surface, has received much attention in computer vision (Grimson, 1991; Faugeras, 1993; Pope, 1994) [17, 15, 30]. Several researchers have investigated the issue of 3-D scene reconstruction from data. The approaches that assume knowledge of the posture of the sensors ((Bajcsy *et al.*, 1998; Debevec *et al.*, 1996; Shum *et al.*, 1998) [3, 9, 32], (Allen and Stamos, 2019) [1], (Becker and Bove, 2021) [4]; and the methods that do not (El-Hakim, *et al.* 1997) [14] may be loosely classified into two types. An article by Roth and Wibowo (2021) [31], who also suggested using a mobile platform to get textured 3-D representations of the surroundings, is of

considerable significance. Eight cameras and a 3-D range sensor are part of their systems. Their method, like ours, utilises sensor data to correct odometric inaccuracies. Their approach, however, does not consider the uncertainty of individual measurements when generating the 3-D model.

Our method is somewhat similar to (Iocchi, 2019) [19] which employs manual guidance throughout the reconstruction process to compensate for the absence of apparent structure in common interior situations when reconstructing planar models of indoor environments using stereo vision. The environment model is made out of flat surfaces, just like ours, but because stereo vision is being used, the range data is insufficient to support completely automated modelling. As a result, the method in (Iocchi, 2019) [19] requires manual assistance to recognise flat surfaces. You may find related work on simulating outdoor landscapes in (Teller, 2019) [33] and (Cheng, *et al.* 2019) [7]

7. Conclusion

We have provided a web-based technique for creating compact maps of interior building spaces. In this method, rectangular surface patches are located using the EM algorithm in 3-D data obtained by a moving robot outfitted with laser range finders and a panoramic camera. Although EM is typically an offline technique, a modified version of EM was given that may produce such maps while the robot is moving online. While simultaneously limiting the computation in ways that enable the algorithm to be executed in real-time, regardless of the map size, this method preserves the main benefit of EM, namely the capacity to modify past assignments and map components depending on future inputs. Experiments show that this method enables mobile robots to acquire accurate and compact maps of interior areas, including corridors.

Accurate 2-D laser range finders were used to obtain all the results presented in this paper. If the sensor model is properly adjusted, the mathematical approach can accommodate a wider range of range finders, including sonars and range cameras. 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 because 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 subject is improving the EM algorithm to determine the robot's location during future studies.

Finally, it would be beneficial to include the data from both lasers in the mapping process. Sadly, it is possible that the upward-pointing laser would never witness the same part of the world twice; as a result, additional presumptions (such smoothness) would be required to use its data. Additionally, it would be intriguing to incorporate the robot posture variables as latent variables into the EM algorithm, allowing for the interleaving of localisation and modelling as in (Thrun, 1998) Sadly, the additional computational complexity brought on by such modifications would make it challenging to use the resultant method in real-time and online.

8. References

1. Allen P, Stamos I. Integration of range and image sensing for photorealistic 3-D modelling, in Proc. IEEE Int. Conf. Robotics and automation; c2019. p. 1435-1440.
2. Artin C, Thrun S. Real-time acquisition of compact volumetric 3D maps with mobile robots, in Proc. IEEE Int. Conf. Robotics and Automation, Washington, DC; c2018. p.311-316.
3. Bajcsy R, Kamberova G, Nocera L. 3-D reconstruction of environments for virtual reconstruction, in Proc. 4th IEEE Workshop on Applications of Computer Vision; 1998, p.160-167.
4. Becker S, Bove M. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views, in Proc. SPIE Symp. Electronic Imaging, San Jose, CA, CD-ROM; c2021.
5. Casper J. 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; c2018.
6. Chatila R, Laumond JP. Position referencing and consistent world modelling for mobile robots, in Proc. IEEE Int. Conf. Robotics and Automation; c2020, p.138-145.
7. Cheng YQ, Riseman E, Wang X, Collins R, Hanson A. Threedimensional reconstruction of points and lines with unknown correspondence across images, Int. J Comput. Vis. 2019;45(2):129-156.
8. Crowley J. World modelling and position estimation for a mobile robot using ultrasonic ranging, in Proc. IEEE Int. Conf. Robotics and Automation, Scottsdale, AZ; c2019, p. 674-680.
9. Debevec P, Taylor C, Malik J. Modeling and rendering architecture from photographs," in Proc. 23rd Int. Conf. Computer Graphics and Interactive Techniques, CD-ROM; c1996.
10. Dempster A, Laird A, Rubin D. Maximum-likelihood from incomplete data via the EM algorithm, J Roy. Statist. Soc., ser. B. 1977;39(1):1-38.
11. Duda R, Hart P. Pattern Classification and Scene Analysis. New York: Wiley; c1973.
12. Eberly D, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. New York and San Mateo, CA: Morgan Kaufman/Academic; c2019.
13. Elfes A. Occupancy grids: A probabilistic framework for robot perception and navigation," PhD dissertation, Dept. Elec. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA; c2019.
14. El-Hakim S, Boulanger P, Blais F, Bernardin JA. Sensor-based creation of indoor virtual environment models, in Proc. 4th Int. Conf. Virtual Systems and Multimedia, Geneva, Switzerland, CD-ROM; c1997.
15. Faugeras O. Three-Dimensional Computer Vision, A Geometric Viewpoint. Cambridge, MA: MIT Press; c1993.
16. Garland M, Heckbert P. Simplifying surfaces with colour and texture using quadric error metrics, in Proc. IEEE Visualization; c1998 Oct. p.263-269.
17. Grimson W. Object Recognition by Computer: The Role of Geometric Constraints. Cambridge, MA: MIT Press; c1991.

18. Gutmann JS, Konolige K. Incremental mapping of large cyclic environments, in Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation, CD-ROM; c2019.
19. Iocchi L, Konolige K, Bajracharya M. Visually realistic mapping of a planar environment with stereo, in Proc. Int. Symp. Experimental Robotics, Waikiki, HI, CD-ROM; c2019.
20. Kortenkamp D, Bonasso R, Murphy R. Eds., AI-Based Mobile Robots: Case Studies of Successful Robot Systems. Cambridge, MA: MIT Press; c1998.
21. Leonard J, Tardós J, Thrun S, Choset H. Eds., Washington, DC, Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localisation for Autonomous Mobile Robots (W4), in Proc. IEEE Int. Conf. Robotics and Automation; c2018.
22. Liu Y, Emery R, Chakrabarti D, Burgard W, Thrun S. Using EM to learn 3-D models with mobile robots, in Proc. Int. Conf. Machine Learning, CD-ROM; c2019.
23. Lu F, Milios E. Globally consistent range scan alignment for environment mapping, Auton. Robots. 1997;4:333-349.
24. Lu F, Milios E. Robot pose estimation in unknown environments by matching 2-D range scans, J Intell. Robot. Syst. 1998;18:249-275.
25. McLachlan G, Krishnan T. The EM Algorithm and Extensions. New York, NY: Wiley; c.1997.
26. Montemerlo M, Hähnel D, Ferguson D, Triebel R, Burgard W, Thayer S, *et al.* A system for three-dimensional robotic mapping of underground mines, Comput. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-02-185; c2018.
27. Moravec HP. Sensor fusion in certainty grids for mobile robots, AI Mag. 1988;9(2):61-74.
28. Nayar S. Omnidirectional vision, in Proc. 9th Int. Symp. Robotics Research, Shonan, Japan, CD-ROM; c1997.
29. Neal R, Hinton G. A view of the EM algorithm that justifies incremental, sparse, and other variants, in Learning in Graphical Models, M. Jordan, Ed. Norwell, MA: Kluwer; c1998.
30. Pope AR. Model-based object recognition a survey of recent research, Univ. British Columbia, Dept. Comput. Sci., Vancouver, BC, Canada, Tech. Rep. TR-94-04; c1994.
31. Roth G, Wibowo. A fast algorithm for making mesh models from multiple-view range data, in Proc. DND/CSA Robotics and KnowledgeBased Systems Workshop, CD-ROM; c2021.
32. Shum H, Han M, Szeliski R. Interactive construction of 3-D models from panoramic mosaics, in Proc. Int. Conf. Computer Vision and Pattern Recognition; c1998 Jun. p.427-433.
33. Teller S, Antone M, Bodnar Z, Bosse M, Coorg S, Jethwa M, *et al*. Calibrated, registered images of an extended urban area, in Proc. Conf. Computer Vision and Pattern Recognition; c2019 Dec. p.I-813-I-820.
34. Thrun S, Hähnel D, Ferguson D, Montemerlo M, Triebel R, Burgard W, *et al*. Thayer, and W. Whittaker, A system for volumetric robotic mapping of abandoned mines, in Proc. IEEE Int. Conf. Robotics and Automation; c2003 Sept. p.4270-4275.
35. Thrun S. A probabilistic online mapping algorithm for teams of mobile robots, Int. J Robot. Res. 2019;20(5):335-363.
36. Thrun S, Fox D, Burgard W. A probabilistic approach to concurrent mapping and localisation for mobile robots, Learning, Auton. Robots. 1998 Mach 31. p.29-53.