Submitted to IJCAI-01

Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot

Dirk Hähnel† **Wolfram Burgard**†‡ **Sebastian Thrun**‡

†Computer Science Department
University of Freiburg
{haehnel,wolfram}@informatik.uni-bonn.de

‡School of Computer Science
Carnegie Mellon Unuversity
thrun@cs.cmu.edu

Abstract

This paper presents an algorithm for full 3D shape reconstruction of indoor and outdoor environments with mobile robots. Data is acquired by a fast-moving robot equipped with two 2D laser range finders. Our approach combines an efficient scan matching routine for robot pose estimation with an algorithm for approximating environments using flat surfaces. On top of that, our approach includes a mesh simplification technique to reduce the complexity of the resulting models. In extensive experiments, our method is shown to produce accurate models of indoor and outdoor environments that compare favorably to other methods.

1 Introduction

The topic of learning 3D models of buildings (exterior and interior) and man-made objects has received considerable attention over the past few years. 3D models are useful for a range of applications. For example, architects and building managers may use 3D models for design and utility studies using virtual reality (VR) technology. Emergency crews, such as fire fighters, could utilize 3D models for planning as to how to best operate at a hazardous site. 3D models are also useful for robots operating in urban environments. And finally, accurate 3D models could be a great supplement to the video game industry, especially if the model complexity is low enough for real-time VR rendering. In all of these application domains, there is a need for methods that can generate 3D models at low cost, and with minimum human intervention.

In the literature, approaches for 3D mapping can be divided into two categories: Approaches that assume knowledge of the pose of the sensors [1; 2; 3; 5; 19], and approaches that do not [11; 20]. In the present paper, we are interested in using mobile robots for data acquisition; hence our approach falls into the second category due to the inherent noise in robot odometry. However, unlike the approaches in [11; 20] which generate highly complex models, our focus is on generating low-complexity models that can be rendered in real-time. The approach in [20], for example, composes models where the number of polygons is similar to the number of raw scans, which easily lies in the hundreds of thousands even for small indoor environments. The majority of existing systems also requires human input in the 3D modeling process. Here we are interested in fully automated modeling without any human interaction. Our approach is also related to [14], 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.

This paper presents an algorithm for generating simplified 3D models of indoor and outdoor environments. The data for generating these models are acquired by a mobile robot, equipped with two 2D laser range finders. The first laser scans horizontally and is used for pose estimation (localization). The second scanner is pointed upwards so that it scans the three-dimensional structures while the robot moves through its environment. To estimate the pose of the robot while collecting the data, a 2D scan matching algorithm is used that is reminiscent of the literature on mobile robot mapping. The resulting pre-filtered data is globally consistent but locally extremely noisy. A recursive surface identification algorithm is then employed to identify large flat surfaces, thereby reducing the complexity of the 3D model significantly while also eliminating much of the noise in the sensor measurement. The resulting 3D models consist of large, planar surfaces, interspersed with small fine-structured models of regions that cannot be captured by a flat-surface model.

The topic of simplification of polygonal models has long been studied in the computer graphics literature (see e.g., [8; 12; 17]), often with the goal of devising algorithms for real-time rendering of complex models. There are two important characteristics of the data generated by robots that differ from the polygonal model studied in the computer graphics literature. First, robot data is noisy. The models studied in the computer graphics literature are usually assumed to be noise-free; hence, simplifications are only applied for increasing the speed of rendering, and not for the reduction of noise. This has important ramifications, as the noise in the data renders a close-to-random fine structure of the initial 3D models. Second, built structure is known to contain large, flat surfaces that are typically parallel or orthogonal to the ground. Such a prior is usually not incorporated in polygon simplification algorithms. Consequently, a comparison with the algorithm presented in [8] illustrates that our approach yields significantly more accurate and realistic-looking 3D models.

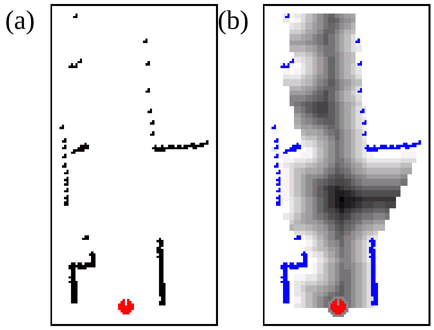

Figure 1: (a) 2D laser scan of the forward-pointed scanner (bird's eye perspective). (b) Log-likelihood function for detecting obstacles in free-space: the darker a location, the less likely it is that another range scan detects an obstacle at this location. Notice that occluded areas are left blank, hence do not contribute to the gradient ascent scan adjustment.

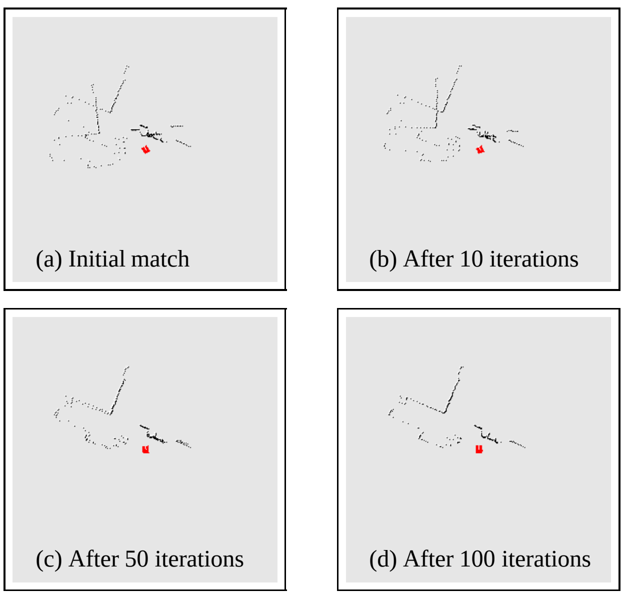

Figure 2: Example of gradient ascent for aligning scans. The initial translational error is 10 cm along each axis, and the rotational error is 30 degrees. The gradient ascent algorithm safely aligns the scans.

2 Concurrent Mapping and Localization in 2D

2.1 2D Scan Alignment

The first step towards building 3D maps with mobile robotics is a 2D pose alignment procedure. The problem is as follows: Robot odometry is erroneous. Small error in odometry, caused by effects such as drift and slippage, multiply over time. Such effects are relatively easy to compensate if a model of the environment is readily available [9]. However, in the absence of such a model, the robot faces a chicken-and-egg problem in that it has to simultaneously estimate both the model and its path.

In the robot mapping literature, this problem is known as the *simultaneously localization and mapping problem* (SLAM) [6]. Powerful statistical techniques have been proposed for this problem, most of which, however, require the extraction of unique features [15] or are computationally very expensive [4]. However, these approaches demonstrate that the pose of a robot can be recovered from 2D information in indoor and leveled outdoor environments, for robots confined to a flat 2D surface.

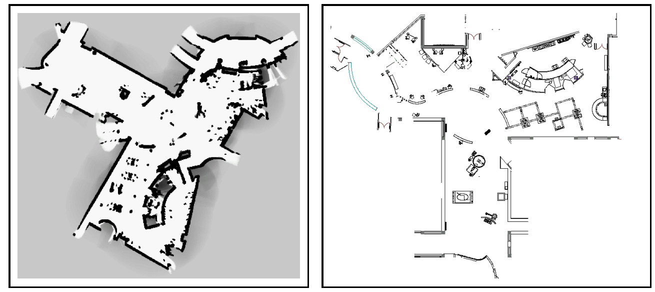

Figure 3: Occupancy grid map (left) and architectural blueprint of a recently constructed museum in San Jose (right). Notice that the blueprint is inaccurate in certain places.

Our algorithm estimates poses in 2D using a real-time scan matching algorithm, similar to the ones described in [16; 20]. Our scan matching algorithm uses gradient ascent in a log-likelihood function defined over pairs of scans. Figure 1a shows an example of a sensor scan, taken with a forward-pointed 2D laser range finder. The probability function of detecting an obstacle in the range of this scan is shown by the grayly shaded area in Figure 1b: The darker a location in this diagram, the less likely it is that an obstacle is detected at some other point in time. This log-likelihood function is differentiable. It is maximal at locations where an object has been detected in the scan (occupied region), or in occluded regions as shown. It is minimal in areas where the present scan fails to detect an obstacle (free-space), with the log-likelihood decreasing in proportion to the distance to the nearest detected obstacle. Thus, this probabilistic perception model is similar to potential fields [13], but it also carries a notion of occlusion.

2.2 Search in Pose Space

Clearly, when aligning a scan to one or more previously collected scans, the total log-likelihood depends on the pose of the scan, where *pose* refers to the scan's x-y-coordinates together with its orientation θ. Exploiting the differentiability of our log-likelihood function, scans are aligned relative to previously recorded scans by adjusting the pose in proportion to the negative gradient:

$$\langle x, y, \theta \rangle \quad \longleftarrow \quad \langle x, y, \theta \rangle - \eta \frac{\partial E}{\partial \langle x, y, \theta \rangle} \tag{1}$$

where E denotes the total log likelihood (a sum over all measurements of a scan), and $\eta > 0$ is a step-size parameter. Figure 2 shows an example of aligning two scans, using the gradient ascent scan matching routine. In this example, the two scans are initially misaligned by translational errors of 10cm in each coordinate axis, and a rotational misalignment of 30 degrees—these errors exceed practical errors by a factor

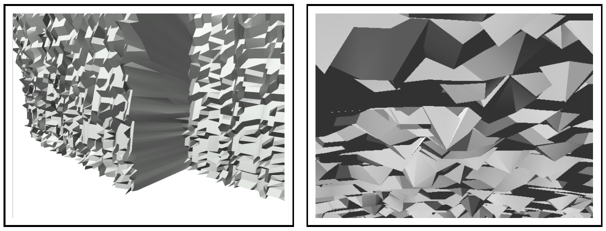

Figure 4: Fractions of the raw data for parts of the ceiling and the wall.

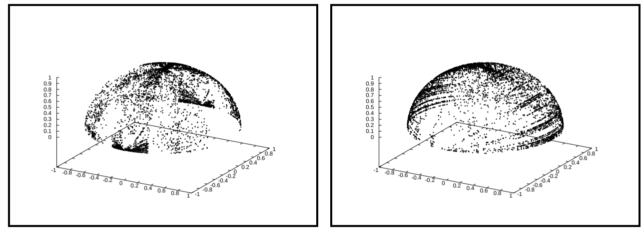

Figure 5: Normed surface normals for a ceiling (left) and a wall (right).

of 10. After 100 iterations, the scans are aligned with sufficient accuracy for our 3D modeling task.

To perform the scan alignment in real-time, our approach pre-computes occlusion and all necessary distances necessary for calculating gradients using a fine-grained 2-dimensional grid (typical resolution: 10 cm). After this pre-computation, which takes approximately 0.1 seconds, each iteration of gradient ascent requires in the order of 1ms on a low-end PC—which is fast enough to align scans accurately as the robot moves. Two related papers [10; 20] discuss the use of probabilistic posterior estimation techniques to build consistent maps in cyclic environments, where our incremental scan matching approach would be doomed to fail.

Figure 3 illustrates the accuracy of the resulting maps. Shown there is a 2D occupancy grid map [7; 18] of a large indoor environment acquired in real-time, along with a blueprint of the same environment. The pose estimates of these 2D maps form the very basis of the topic centrally addressed in this paper: Building 3D maps.

2.3 Generating 'Raw' 3D Data

The 3D data is generated using the upward-pointed laser, as shown in Figure 6. While the robot traverses and maps an unknown environment in 2D, thereby recovering its pose, the upward pointed laser scans the 3D structure of the environment. We then obtain a polygonal model by connecting consecutive 3D points. To avoid closing wholes such as doorways etc, we only create a polygonal surface if the consecutive points are close to each other.

3 Learning Smooth 3D Models

Although the approach described above produces accurate position estimates, the resulting models often lack the appropriate smoothness. Figure 4 shows, in detail, some of the data. As is easily seen, the data is extremely rugged. Whereas some of the ruggedness arises from remaining errors in the pose estimation, the majority of error stems from measurement noise in laser range finders. However, the key characteristic here is that all noise is local, as the scans have been globally aligned by the 2D mapping algorithm. As a result, global structures cannot be extracted by considering small areas of the data. Rather, one has to scan larger fractions of the model in order to find such structures.

For example, the fractions of the models illustrated in Figure 4 show data obtained for a wall (left image) and a ceiling (right image). Although the corresponding objects are planar in the real world, this structure cannot be extracted from the local surfaces. Figure 5 shows the surface normals for 5000 surfaces of the wall and the ceiling partly shown in Figure 4. As can be seen from the figure, the normals are almost uniformly distributed. Accordingly, first experiments with a standard clustering algorithm lead to unsatisfactory results because the surface normals could not be separated appropriately.

The noise problem is inherent to the sensor-based acquisition of high-resolution 3D-models. In order to scan an object with high resolution, the distance of consecutive scanning positions must be extremely small. However, the smaller the distance between consecutive scanning positions, the higher the influence of the measurement noise on the deviation between surface normals of neighboring shapes (and also to the true surface normal).

Since approximations of larger structures cannot be found by a local analysis, our method performs a randomized search to find larger planar structures in the data. If such a planar structure is found, our approach maps the corresponding 3D-points on this planar surface. In a second phase neighboring surfaces in the mesh which lie on the same plane and satisfy further constraints described below are merged into larger polygons.

3.1 Planar Approximation of Surfaces

The algorithm to find planes for sets of points is a randomized approach. It starts with a randomly chosen point in 3D and applies a region growing technique to find a maximum set of points in the neighborhood to which a fitting plane can be found. As the optimal plane we chose that plane which minimizes the sum of the squared distances to the points v_i in the current region. The normal of this plane is given by the eigenvector corresponding to the smallest eigenvalue of the 3×3 matrix

$$A = \sum_{i=1}^{n} (v_i - m)^T \cdot (v_i - m) \qquad (2)$$

where

$$m = \frac{1}{n} \sum_{i=1}^{n} v_i \qquad (3)$$

is the center of mass of the points v_i. The minimum eigenvalue corresponds to the sum of the squares of the distances between the plane and the points v_i.

Our approach proceeds as follows. It starts with a random point v_1 and its nearest neighbor v_2. It then repeatedly tries to extend the current set Π of points by considering all other points in increasing distance form this point set. Suppose v' is a point such that the point distance $\texttt{pointDist}(\Pi, v')$ between v' and one point in Π is less than δ (which is 30cm in our current implementation). If the average squared error $\texttt{error}(\Pi \cup \{v'\})$ to the optimal plane for $\Pi \cup \{v'\}$ is less than ϵ (which was 2.8 in all our experiments) and if the distance of v' to the optimal plane for $(\Pi \cup \{v'\})$ is less than γ ($\gamma = 10$cm in our implementation) then v' is added to Π. As a result, regions are grown to include nearby points regardless of the surface normal of the polygons neighboring these points (which are assumed to be random). This process is described more precisely in Table 1. To find the best planes, this process is restarted for different randomly chosen starting points v_1 and v_2. Our approach always selects the largest plane found in each round. If no further plane can be found, the overall process is terminated

select point tuple v_1, v_2
$\Pi := \{v_1, v_2\}$
WHILE (new point can be found) BEGIN
    Select point v' with $\texttt{pointDist}(\Pi, v') < \delta$
    if $\texttt{error}(\Pi \cup \{v'\}) < \epsilon$ && $||(\Pi \cup \{v'\}, v')|| < \gamma$
        $\Pi := \Pi \cup \{v'\}$
END WHILE

Table 1: Algorithm to search for a plane through a point tuple.

Obviously, our approach solves a mesh simplification problem that has been studied throughly in the computer graphics literature. The important difference between our approach and mesh simplification algorithms from computer graphics, such as [8; 12], lies in the way the input data is processed. In contrast to our method, which tries to fit a plane to a larger set of points, the techniques presented in [8; 12] only perform a local search and consider only pairs of surfaces. Neighbored surfaces are simplified by minimizing an error or energy function which specifies the visual discrepancy between the original model and simplified model in terms of discontinuities in the surface normals. Because of the local noise in our data these techniques cannot distinguish between areas with a higher level of detail such as corners and areas with few details such as planar structures corresponding to walls. Thus, the simplification is carried out uniformly over the mesh. Our approach, in contrast, simplifies planar structures and leaves a high level of detail where it really matters.

3.2 Merging of Surfaces

In a second phase, neighboring polygons belonging to the same plane are merged to larger polygons. A polygon belongs to a plane, if all of its edges belong to this plane. Two polygons of the same plane can be combined, if

1. both polygons have exactly one sequence of common edges and
2. if both polygons do not overlap.

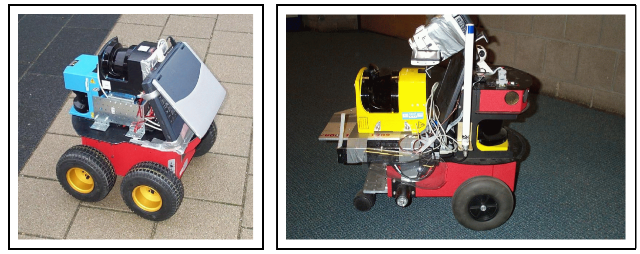

Figure 6: The platforms used for the experiments. Outdoor system (left), indoor system (right).

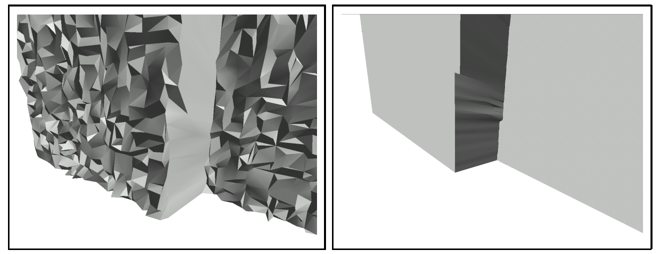

Figure 7: Magnification of a doorway in the corridor environment after the approximation with QSlim (left) and with our approach (right).

Our approach repeatedly performs this merging process until there do not exist any further polygons that can be merged. Please note that both conditions are sufficient to ensure that each merging operation leads to a valid polygon. Furthermore, the resulting polygons are not necessarily convex, i.e. our approach does not close wholes in the model coming from doors or windows, such as the technique describes in [11].

4 Experimental Results

Our approach has been implemented and tested using two different platforms (see Figure 6), and in indoor and outdoor environments. As pointed out above, both robots were equipped with two 2D laser-range scanners. Whereas the angular resolution of the laser used on the outdoor system is 0.5 degrees, the resolution of the lasers mounted on the indoor system is 1 degree. The resolution of the measured distances is 1cm and the measurement error of these systems lies between 0 and 20cm.

The first experiment was carried out in our office environment. Here the robot traveled 10m through a corridor and measured 82,592 points in 3D. The corresponding raw 3D data consisted of 163,336 triangles. This input model is depicted in the left image of Figure 8. The result of our simplification technique is shown in the right image of Figure 8. Our approach needed 6 hours to compute the planes and generated 868 polygons. Only 24,996 triangles could not be approximated by larger planar structures. As a result, we obtained a significant reduction by 84% of the input data. The center of Figure 8 shows the result of the QSlim system [8] which applies computer graphics algorithms to reduce the complexity of 3D models. Please note that this model contains the same number of polygons as obtained with our approach. Obviously, the quality of our model is significantly higher than

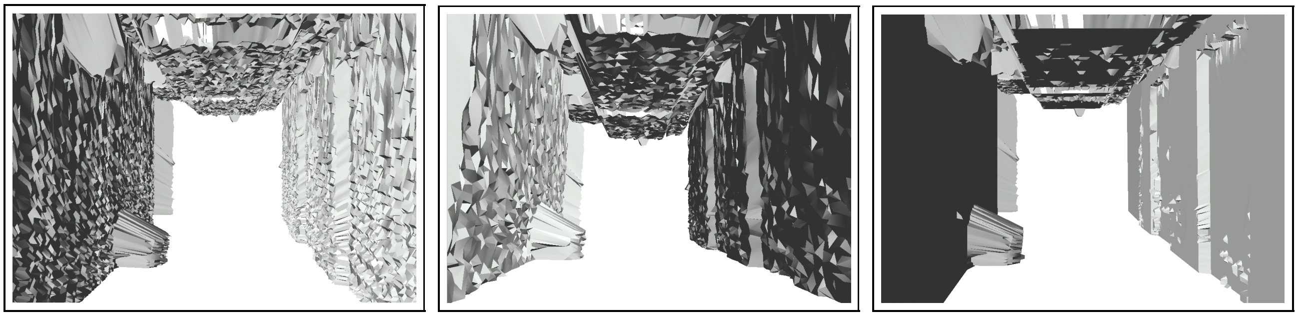

Figure 8: Models learned for an indoor environment. From left to right: Raw data, QSlim-approximation, our approach.

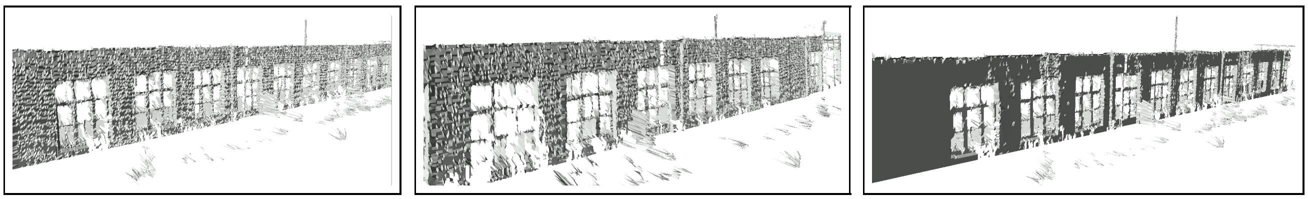

Figure 9: Models learned for a building: Raw data (left), QSlim-simplification (center), our approach (right).

the quality obtained by the QSlim system. Figure 7 shows magnified parts of these models which correspond to the data shown in the left image of Figure 4. Apparently, our approach provides accurate approximations of the planar structures and computes models with a seriously lower complexity than the QSlim system.

Additionally, we applied our approach to different data sets obtained with our outdoor system. The left image of Figure 9 shows one such 3D data set. Again the local noise is clearly visible. Whereas the right image of the figure shows the result of our planar approximation, the center image displays the approximation obtained with QSlim. Both models contain the same number of shapes.

Finally, the left image of Figure 10 shows the model obtained for a building with a size of 40×70m. The right image shows a photo of the same building. Due to the high number of non-planar structures on the surface of the building, the reduction was only 40.7% in this case (from 800,869 to 474,921 surfaces). The overall computation time was 10 hours.

5 Related Work

Due to the various application areas like virtual reality, telepresence, access to cultural savings, the problem of constructing 3D models has recently gained serious interest. The approaches described in [2; 3; 5; 19] rely on computer vision techniques and reconstruct 3D models from sequences of images. Allen et al. [1] construct accurate 3D-models with stationary range scanners. Their approach also includes techniques for planar approximations in order to simplify the models. However, their technique computes the convex hull of polygons in the same plane and therefore cannot deal with windows or doors. Furthermore, their approach to region clustering assumes that the relative positions between consecutive scans are exactly known. Systems similar to ours have been presented in [11] and [20]. Both techniques use a mobile platform to construct 3D models of an environment using range sensors. However, they do not include any means for planar approximation. Accordingly our models have a significantly lower complexity.

The problem of polygonal simplification has been studied intensively in the computer graphics area [8; 12; 17]. The primary goal of these methods is to simplify a mesh so that the visual appearance of the original model and the simplified model is almost identical. Typical criteria used for simplification are the discontinuity of the surface normals of neighboring surfaces as well as the relative angle between the surface normal and the viewing direction. Because of the local noise in the data, however, these methods fail to extract planar structures. Accordingly, our approach provides significantly better approximations in such areas.

6 Conclusions

We have presented an algorithm for acquiring 3D models with mobile robots. The algorithm proceeds in two stages: First, the robot pose is estimated using a fast scan matching algorithm. Second, 3D data is smoothed by identifying large planar surface regions. The resulting algorithm is capable of producing 3D maps without manual intervention, as demonstrated using data sets of indoor and outdoor scenes.

The work raises several follow-up questions that warrant future research. Most importantly, the current 3D model is limited to flat surfaces. Measurements not representing flat objects are not corrected in any way. As a consequence, the resulting model is still fairly complex. An obvious extension involves broadening the approach to include atoms other than flat surfaces, such as cylinders, poles, etc. Additionally, an interesting question concerns robot exploration. The issue of robot exploration has been studied extensively for building 2D maps, but we are not aware of robot exploration algorithms that apply to the full three-dimensional case. This case introduces the challenge that the robot cannot move arbitrarily close to objects of interest, since it is confined to a two-

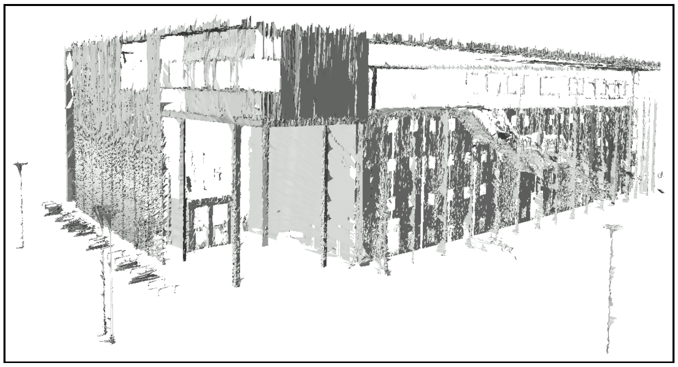

Figure 10: Building model learned by our robot (left) and photo of the same building (right).

dimensional manifold. Finally, extending this approach to multi-robot mapping and arbitrary outdoor terrain (e.g., planetary exploration) are worthwhile goals of future research.

Acknowledgment

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

References

[1] P.K. Allen and Ioannis Stamos. Integration of range and image sensing for photorealistic 3D modeling. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1435–1440, 2000.

[2] R. Bajcsy, G. Kamberova, and Lucien Nocera. 3D reconstruction of environments for virtual reconstruction. In *Proc. of the 4th IEEE Workshop on Applications of Computer Vision*, 2000.

[3] S. Becker and M. Bove. Semiautomatic 3-d model extraction from uncalibrated 2-d camera views. In *Proc. of the SPIE Symposium on Electronic Imaging, San Jose*, 1995.

[4] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping of large-scale mobile robot environments using EM. In *Proceedings of the International Conference on Machine Learning*, Bled, Slovenia, 1999.

[5] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.

[6] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. Working notes of ICRA'2000 Workshop W4: Mobile Robot Navigation and Mapping, April 2000.

[7] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Transactions on Robotics and Automation*, 3(3):249–265, 1987.

[8] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proc. of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 209–216, 1997.

[9] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[10] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.

[11] S.E. Hakim, P. Boulanger, and F. Blais. A mobile system for indoors 3-d mapping and positioning. In *Proc. of the 4th Conference on Optical 3-D Measurement Techniques*, 1997.

[12] H. Hoppe. Progressive meshes. In *Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.

[13] O. Khatib. Real-time obstacle avoidance for robot manipulator and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[14] M. Bajracharya L. Iocchi, K. Konolige. Visually realistic mapping of a planar environment with stereo. In *Proceedings of the 2000 International Symposium on Experimental Robotics*, Waikiki, Hawaii, 2000.

[15] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In J. Hollerbach and D. Koditschek, editors, *Proceedings of the Ninth International Symposium on Robotics Research*, Salt Lake City, Utah, 1999.

[16] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 1998. to appear.

[17] D. Luebke and C. Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proc. of the 24rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1997.

[18] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.

[19] H. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[20] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000. IEEE.