Automatic Model Refinement for 3D Reconstruction with Mobile Robots

Andreas Nüchter, Hartmut Surmann, and Joachim Hertzberg
Fraunhofer Institute for Autonomous Intelligent Systems (AIS)
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
{andreas.nuechter|hartmut.surmann|joachim.hertzberg}@ais.fraunhofer.de

Abstract

Precise digital 3D models of indoor environments are needed in several applications, e.g., facility management, architecture, rescue and inspection robotics. This paper presents a new algorithm that transforms a 3D volumetric model into a very precise compact 3D map and generates semantic descriptions. Our system is composed of a robust, autonomous mobile robot for the automatic data acquisition and a precise, cost effective, high quality 3D laser scanner to gage indoor environments. The reconstruction method consists of reliable scan matching and feature detection algorithms. The 3D scene is matched against a coarse semantic description of general indoor environments and the generated knowledge is used to refine the 3D model.

1 Introduction

Automatic and precise reconstruction of indoor environments is an important task in robotics and architecture. Autonomous mobile robots equipped with a 3D laser range finder are well suited for gaging the 3D data. Due to odometry errors the self localization of the robot is an unprecise measurement and therefore can only be used as a starting point for registration of the 3D scans in a common coordinate system. Furthermore the merging of the views as well as the scanning process itself is noisy and small errors may occur. We overcome these problems by extending the reconstruction process with a new knowledge based approach for the automatic model refinement.

Since architectural shapes of environments follow standard conventions arising from tradition or utility [9] we can exploit knowledge for reconstruction of indoor environments. The used knowledge describes general attributes of the domain, i.e., architectural features as plane walls, ceilings and floors. For various domains different knowledge is needed, e.g., for reverse engineering of CAD parts [20]. We show that applying general knowledge for recovering specific knowledge improves reverse engineering.

In mobile robotics one important task is to learn the environment to fulfill specific jobs. 3D maps are needed for plan execution and obstacle avoidance [23]. Volumetric maps, i.e., 3D point clouds are often large and difficult to use directly in control tasks. Therefore some groups have attempted to generate compact flat 3D models [12, 15] or compact bounding box models [24].

This paper presents algorithms for building compact and precise 3D models and generates a coarse semantic interpretation, thus creates coarse semantic maps. The proposed algorithm consists of three steps: First we extract features, i.e., planes from registered unmeshed range data. The planes are found by an algorithm which is a mixture of the RANSAC (Random Sample Consensus) algorithm and the ICP (Iterative Closest Point) algorithm [1, 5]. Second the computed planes are labeled based on their relative orientation. A predefined semantic net implementing general knowledge about indoor environments is employed to define these orientations. Finally architectural constraints like parallelism and orthogonality are enforced with respect to the gaged 3D data by numerical methods.

The paper is organized as follows. After discussing the state of the art in the following part we present the 3D laser range finder and the autonomous mobile robot. The second section presents the range image registration, followed by a description of the feature extraction algorithm. The algorithms for semantic interpretation of the data is given in section four. In section 5 the model refinement is described. Section 6 summarizes the results and concludes the paper.

1.1 Related Work

Automatic and autonomous reconstruction of environments has received much attention for several years. Some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. Thrun et al. [12, 15, 25], Früh et al. [10] and Zhao et al. [26] use two 2D laser range finder for acquiring 3D data. One laser scanner is mounted horizontally and one is mounted vertically. The latter one grabs a vertical scan line which

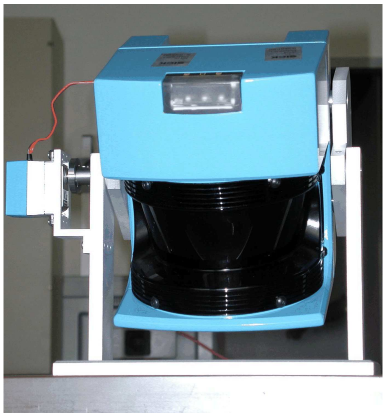

Figure 1. The AIS 3D laser range finder.

is transformed into 3D points using the current robot pose. Since the vertical scanner is not able to scan sides of objects, Zhao et al. use two additional vertical mounted 2D scanner shifted by $45°$ to reduce occlusion [26]. The horizontal scanner is used to compute the robot pose. The precision of 3D data points depends on that pose and on the precision of the scanner.

A few other groups use 3D laser scanners [2, 13, 21]. A 3D laser scanner generates consistent 3D data points within a single 3D scan. The RESOLV project aimed to model interiors for virtual reality and tele presence [21]. They used a RIEGL laser range finder on robots and the ICP algorithm for scan matching [5]. The AVENUE project develops a robot for modeling urban environments [2], using a CYRAX laser scanner and a feature based scan matching approach for registration of the 3D scans in a common coordinate system [22]. The research group of M. Hebert reconstruct environments using the Zoller+Fröhlich laser scanner and aim to build 3D models without initial position estimates, i.e., without odometry information [13].

1.2 The AIS 3D Laser Range Finder

The AIS 3D laser range finder [23] is built on the basis of a 2D range finder by extension with a mount and a servomotor. The 2D laser range finder is attached to the mount for being rotated. The rotation axis is horizontal. A standard servo is connected on the left side (figure 1) and is controlled by the computer running RT-Linux, a real-time operating system which runs LINUX as a task with lowest priority [23]. The 3D laser scanner operates up to 5h (Scanner: 17 W, 20 NiMH cells with a capacity of 4500 mAh, Servo: 0.85 W, 4.5 V with batteries of 4500 mAh).

The area of $180°$(h) $\times$ $120°$(v) is scanned with different horizontal (181, 361, 721) and vertical (128, 256) resolutions. A plane with 181 data points is scanned in 13 ms by the 2D laser range finder (rotating mirror device). Planes with more data points, e.g., 361, 721, duplicate or quadruplicate this time. Thus a scan with 181 $\times$ 256 data points needs 3.4 seconds. In addition to the distance measurement the AIS 3D laser range finder is capable of quantifying the amount of light returning to the scanner.

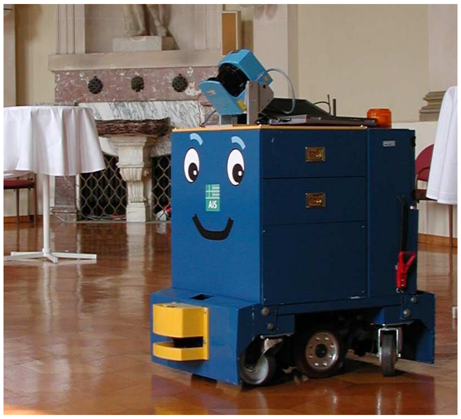

Figure 2. The Ariadne robot platform.

1.3 The Autonomous Mobile Robot

The Ariadne Robot (figure 2) is an industrial robot and is about 80 cm $\times$ 60 cm large and 90 cm high. The mobile platform can carry a payload of 200 kg at speeds of up to 0.8 m/s (about half the speed of a pedestrian). The right and left driving wheels are mounted on a suspension on the center line of the mobile platform. Passive castors on each corner of the chassis ensure stability. The core of the robot is a Pentium-III-800 MHz with 384 MB RAM and real-time Linux. One embedded PC-104 system is used to control the motor, internal display and numerical keyboard and radio link of the robot. The platform is rigged with two 2D safety laser scanners as bumper substitutes, one on the front and the other on the rear of the robot. Each laser scans a horizontal plane of 180° of the environment. The robot has a weight of 250 kg and operates for about 8 hours with one battery charge [24][1].

2 Range Image Registration

We use the well-known Iterative Closest Points (ICP) algorithm to calculate a rough approximation of the transformation while the robot is acquiring the 3D scans. The ICP algorithm calculates iteratively the point correspondence. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transfor-

[1] Videos of the exploration with the autonomous mobile robot can be found at `http://www.ais.fhg.de/ARC/3D/scanner/cdvideos.html`

mation $(\mathbf{R}, \mathbf{t})$ for minimizing the equation

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left|\left| \mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t}) \right|\right|^2, \qquad (1)$$

where N_m and N_d, are the number of points in the model set M or data set D, respectively and w_{ji} are the weights for a point match. The weights are assigned as follows: $w_{ji} = 1$, if $\mathbf{m}_i$ is the closest point to $\mathbf{d}_j$ within a close limit, $w_{ji} = 0$ otherwise.

It is shown that the iteration terminates in a minimum [5]. The assumption is that in the last iteration step the point correspondences are correct. In each iteration, the transformation is calculated by the quaternion based method of Horn [14].

2.1 Matching Multiple 3D Scans

To digitalize environments without occlusions, multiple 3D scans have to be registered. After registration, the scene has to be globally consistent. A straightforward method for aligning several 3D scans is *pairwise matching*, i.e., the new scan is registered against the scan with the largest overlapping areas. The latter one is determined in a preprocessing step. Alternatively, Chen and Medioni [7] introduced an *incremental matching* method, i.e., the new scan is registered against a so-called *metascan*, which is the union of the previous acquired and registered scans. Each scan matching has a limited precision. Both methods accumulate the registration errors such that the registration of many scans leads to inconsistent scenes and problems with the robot localization.

Pulli presents a registration method that minimizes the global error and avoids inconsistent scenes [18]. This method distributes the global error while the registration of one scan is followed by registration of all neighboring scans. Other matching approaches with global error minimization have been published, e.g., by Benjemaa et al. [3, 4] and Eggert et al. [8].

Based on the idea of Pulli we designed a method called *simultaneous matching*. Thereby, the first scan is the masterscan and determines the coordinate system. This scan is fixed. The following steps register all scans and minimize the global error:

1. Based on the robot odometry, pairwise matching is used to find a start registration for a new scan. This step speeds up computation.
2. A queue is initialized with the new scan.
3. Three steps are repeated until the queue is empty:
   (a) The current scan is the first scan of the queue. This scan is removed from the queue.
   (b) If the current scan is not the master scan, then a set of neighbors (set of all scans that overlap with the current scan) is calculated. This set of neighbors form one point set M. The current scan forms the data point set D and is aligned with the ICP algorithms.
   (c) If the current scan changes its location by applying the transformation, then each single scan of the set of neighbors that is not in the queue, is added to the end of the queue.

Note: One scan overlaps with another, iff more than 250 corresponding point pairs exist. To speed up the matching, kD trees and *reduced points* are used [23, 24].

In contrast to Pulli's approach, the proposed method is totally automatic and no interactive pairwise alignment has to be done. Furthermore the point pairs are not fixed [18]. The computed transformations are applied to the robot pose and thus a relocalization of the robot is done after every 3D scan. The *simultaneous localization and mapping problem (SLAM)* is solved.

3 Feature Detection

A common technique for plane extraction is the region growing based approach, e.g., used by Hähnel et al. [12]. Starting from an initial mesh, neighbored planar triangles are merged iteratively. The drawback of this approach is the high computational demand. Alternatively the approach of online surfaces detection based on line detection in scan slices of a 3D scans [23] reduces the computational requirements [23], but extending this approach to multiple 3D scans leads to difficulties.

Another well known algorithm for feature extraction from data sets is the RANSAC algorithm [1], used by Cantzler et al. [6]. RANSAC (Random Sample Consensus) is a simple algorithm for robust fitting of models in the presence of many data outliers. RANSAC first selects N data items randomly and uses them to estimate the parameters of the plane. The next step computes the number of data points fitting the model based on a user given tolerance. RANSAC accepts the fit, if the computed number exceeds a certain limit. Otherwise the algorithm iterates with other points [1].

Liu et al. proposes another technique for plane extraction from range data. They use expectation maximization (EM) for generating a surface model [15]. Their algorithm adjusts the number of planes and estimates the location and orientation, by maximizing the expectation of a logarithmic likelihood function. Plane parameters are efficiently calculated by reducing the problem to a computation of eigenvalues by introducing Lagrange multipliers. This approach is not inherently able to determine the number of planes in the data set [12].

Our algorithm is a mixture of the RANSAC and the ICP algorithm, and provides fast plane extraction for a point cloud. No prior meshing algorithms needs to be applied. A plane p is defined by three 3D points $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}^3)$ or by one 3D point and the surface normal $(\mathbf{p}_1,\ \mathbf{n}$ with $||\mathbf{n}|| = 1,\ \mathbf{p}_1, \mathbf{n} \in \mathbb{R}^3)$. To detect a surface the algorithm randomly selects a point and estimates a plane through two neighbored data points. Now the data points $\mathbf{x} \in \mathbb{R}^3$ are calculated that fulfill:

$$|(\mathbf{x} - \mathbf{p}_1) \cdot \mathbf{n}| < \epsilon. \qquad (2)$$

If this set of points exceeds a limit, e.g., 50 points, an ICP based optimization is started. All data points satisfying eq. (2) form the model set M and are projected to the plane to form the data set D for each iteration of the ICP algorithm. Minimizing the ICP error function (1) by transforming the plane with this point-to-plane metric takes only a few iterations. The time consuming search is replaced by direct calculation of the closest point and the transformation $(\mathbf{R}, \mathbf{t})$ is efficiently calculated [14]. Given the best fit, all plane points are marked and subtracted from the original data set. The algorithm terminates after all points have been tested according to eq. (2).

The extracted 3D planes are unbounded in size. Surfaces are finally extracted from the points by mapping them onto the planes. A quadtree based method generates the surfaces. Figure 4 shows an example with 12 extracted planes of a single 3D scan containing 184576 range data points.

4 Semantic Scene Interpretation

The scene interpretation uses the features, i.e., planes found by the algorithm described in the previous section. The background for interpretation comprises generic architectural knowledge. A model of an indoor scene is implemented as a semantic net based on the idea of Grau et al. [11] and also used by Cantzler et al. [6].

Nodes of a semantic net represent entities of the world / model. The relationship between the entities are encoded using different connections. Possible labels of the nodes are $L = \{$`Wall`, `Floor`, `Ceiling`, `Door`, `No_Feature`$\}$. The relationships between the features are $R = \{$`parallel`, `orthogonal`, `above`, `under`, `equal height`$\}$. The labels `above` and `under` are relative to their plane and hence not commutative. Figure 3 shows the entities and the relation. The reader should notice that in our semantic net a `door` is an *open* door. The semantic net can easily be extended to more entities which have be accompanied by a more sophisticated feature detection. This paper concentrates on plane detection so that the semantic net is a subset of all indoor environments.

A depth first search (backtracking) is implemented to assign the labels to the set of planes P according to the constraints in the semantic net. The search starts by assigning the first label from L to the first plane. The second plane is labeled and tested with the constraints given by the net. If all constraints are satisfied the search continues with the next plane. Otherwise backtracking starts with further labels. This process terminates after the whole search tree is tested and all consistent combinations are generated. A consistent labeling exists if each plane is assigned with a label and the model graph is arc consistent. From all consistent labelings our algorithm chooses the labeling that maximizes

$$\sum_{p \in P} f(p), \qquad (3)$$

where $f(p) = 0$ if plane p is assigned to No_Feature, $f(p) = 1$ if the plane is assigned to Wall, Door, Floor or Ceiling. The maximization of (3) ensures correct labelings containing Floor, Ceiling and Walls with the minimal number of No_Features and requires a complete tree search.

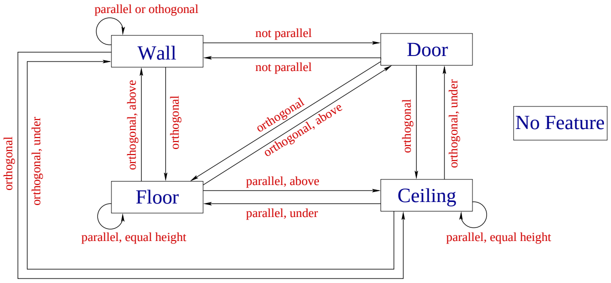

Figure 3. Semantic net for scene interpretation.

The computational expense is reduced by backtracking pruning and reusing (caching) of constraint tests, e.g., the verification that two planes are orthogonal. Especially the constraints ”under” and ”above” require a distance computation with all points of the plane. Figure 4 shows the interpretation of extracted planes from a point cloud acquired in the GMD Robobench, a standard office environment for the evaluation of autonomous mobile robots. The plane labeled with door is an slightly opened office door.

5 Model Refinement

Due to unprecise measurements or registration errors, the 3D data might be erroneous. These errors lead to inaccurate 3D models. The semantic interpretation enables us to refine the model. The planes are adjusted such that the planes explain the 3D data and the semantic constraints like parallelism or orthogonality are enforced.

To enforce the semantic constraints the model is first simplified. A preprocessing step merges neighboring planes with equal labels, e.g., two ceiling planes. This simplification process increases the point to plane distance, which

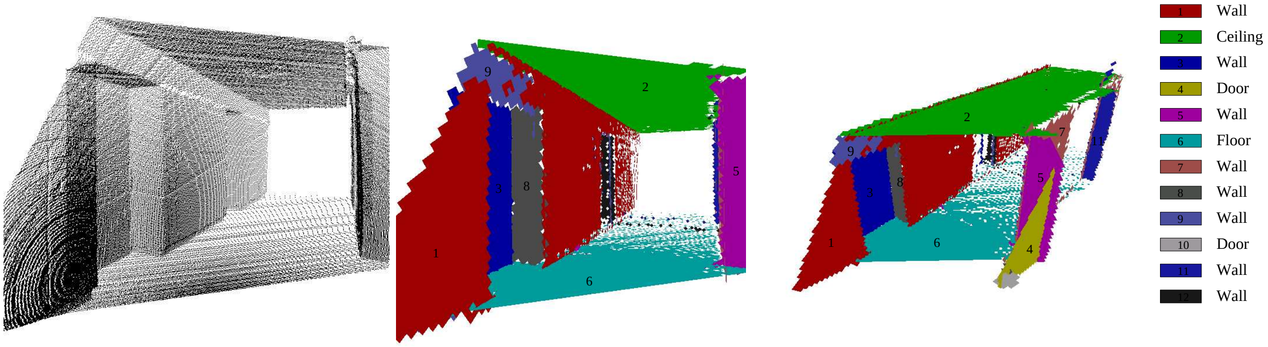

Figure 4. Left: Point cloud. Middle and right: Extracted planes and semantic interpretation.

has to be reduced in the following main optimization process. This optimization uses an error function to enforce the parallelism or orthogonality constraints. The error function consists of two parts. The first part accumulates the point to plane distances and the second part accumulates the angle differences given through the constraints. The error function has the following form:

$$E(P) = \sum_{p_i \in P} \sum_{\mathbf{x} \in p_1} ||(\mathbf{x} - \mathbf{p}_{i1}) \cdot \mathbf{n}_i|| + \gamma \sum_{p_i \in P} \sum_{p_j \in P} c_{i,j}, \quad (4)$$

where $c_{i,j}$ expresses the parallelism (5) or orthogonality (6) constraints according to

$$c_{i,j} = \min\{|\arccos(\mathbf{n}_i \cdot \mathbf{n}_j)|, |\pi - \arccos(\mathbf{n}_i \cdot \mathbf{n}_j)|\} \quad (5)$$

and

$$c_{i,j} = |\frac{\pi}{2} - \arccos(\mathbf{n}_i \cdot \mathbf{n}_j)|. \quad (6)$$

Minimization of eq. (4) is a nonlinear optimization process.

The time consumed for optimizing eq. (4) increases with the number of plane parameters. To speed up the process, the normal vectors $\mathbf{n}$ of the planes are specified by spherical coordinates, i.e., two angles α, β. The point $\mathbf{p}_1$ of a plane is reduced to a fixed vector pointing from the origin of the coordinate system in the direction of $\mathbf{p}_1$ and its distance d. The minimal description of all planes P consists of the concatenation of p_i, with $p_i = (\alpha_i, \beta_i, d_i)$, i.e., a plane p_i is defined by two angles and a distance.

A suitable optimization algorithm for eq. (4) is Powell's method [16], because the optimal solution is close to the starting point. Powell's method finds search directions with a small number of error function evaluations of eq. (4). Gradient descent algorithms have difficulties, since no derivatives are available. Cantzler et al. use a time consuming genetic algorithm for the optimization [6].

Powell's method computes directions for function minimization in one direction [16]. From the starting point P_0 in the n-dimensional search space (the concatenation of the 3-vector descriptions of all planes) the error function (4) is optimized along a direction $\mathbf{i}$ using a one dimensional minimization method, e.g., Brent's method [17].

Conjugate directions are good search directions, while unit basis directions are inefficient in error functions with valleys. At the line minimum of a function along the direction $\mathbf{i}$ the gradient is perpendicular to $\mathbf{i}$. In addition, the n-dimensional function is approximated at point P by a Taylor series using point P_0 as origin of the coordinate system. It is

$$E(P) = E(P_0) + \sum_k \frac{\partial E}{\partial P_k} P_k + \sum_{k,l} \frac{\partial^2 E}{\partial P_k \partial P_l} P_k P_l + \cdots (7)$$

$$\approx \quad c - \mathbf{b} \cdot P + \frac{1}{2} P \cdot \mathbf{A} \cdot P \quad (8)$$

with $c = E(P_0)$, $\mathbf{b} = \nabla E|_{P_0}$ and $\mathbf{A}$ the Hessian matrix of E at point P_0. Given a direction $\mathbf{i}$, the method of conjugate gradients is to select a new direction $\mathbf{j}$ so that $\mathbf{i}$ and $\mathbf{j}$ are perpendicular. This selection prevents interference of minimization directions. For the approximation (8) the gradient of E is $\nabla E = A \cdot P - \mathbf{b}$. From the differentiation $(\delta(\nabla E) = \mathbf{A}(\delta P))$ it follows for directions $\mathbf{i}$ and $\mathbf{j}$ that

$$0 = \mathbf{i} \cdot \delta(\nabla E) = \mathbf{i} \cdot \mathbf{A} \cdot \mathbf{j}. \quad (9)$$

With the above equation conjugate directions are defined and Powell's method produces such directions, without computing derivatives.

The following heuristic scheme is implemented for finding new directions. Starting point is the description of the planes and the initial directions $\mathbf{i}_l$, $l = 1, \ldots, n$ are the unit basis directions. The algorithm repeats the following steps until the error function (4) reaches a minimum [17]:

1. Save the starting position as P_0.
2. For $l = 1, \ldots, n$, minimize the error function (4) starting from P_{l-1} along the direction $\mathbf{i}_l$ and store the minimum as the next position P_l. After the loop, all P_l are computed.

3. Let $\mathbf{i}_l$ be the direction of the largest decrease. Now this direction $\mathbf{i}_l$ is replaced with the direction given by $(P_n - P_0)$. The assumption of the heuristic is that the substituted direction includes the replaced direction so that the resulting set of directions remains linear independent.
4. The iteration process continues with the new starting position $P_0 = P_n$, until the minimum is reached.

Experimental evaluations for the environment test settings show that the minimization algorithm finds a local minimum of the error function (4) and the set of directions remains linear independent. The computed description of the planes fits the data and the semantic model.

The semantic description, i.e., the ceiling and walls, enable to transform the orientation of the model along the coordinate axis. Therefore it is not necessary to transform the model interactively into a global coordinate system or to stay in the coordinates given by the first 3D scan.

6 Results and Conclusion

6.1 Experimental Results

The proposed methods have been tested in several experiments with our autonomous mobile robot in the GMD Robobench. Figure 4 shows an example 3D point cloud (single 3D scan with 184576 points) and the semantic interpretation. The corresponding original and refined model is given in figure 5 (top: Original model, $E(P)$ = 14.57 + γ 173.09, bottom: Refined model, $E(P)$ = 26.68 + γ 2.35, γ was set to 100.0). The figure shows the reduction of the jitters at the floor and ceiling (circled). The orientation of the model in the bottom image is transformed along the axis of the coordinate system and the meshing algorithm produces flat walls. The total computation time for the optimization is about one minute (Pentium-IV-2400).

An octree-based algorithm [19] generates the mesh (cube width: 5cm) to visualize the differences between the images. Starting from a cuboid surrounding the whole scene the mesh generation recursively divides the scene into 8 smaller cubes. Empty nodes of the resulting octree are pruned.

The second example in figure example 6 consists of eight merged scans acquired by the autonomous mobile robot driving in the GMD Robobench. The scene consists of a 32 meter corridor connecting 15 offices. Two persons are standing inside at the beginning. Figure 6 top, left shows the 3D data and reflectance values. The next two pictures (top middle and right) show the extracted and labeled planes. The two persons and other non-flat objects, e.g., dynamic objects, are not explained by the semantic net and therefore filtered from the plane model. The door behind the right person becomes visible. Figure 6 bottom shows the original (left) and refined (right) octree model with marked differences. The images contains the silhouette of two persons, because all points not assigned to planes are unchanged and included.

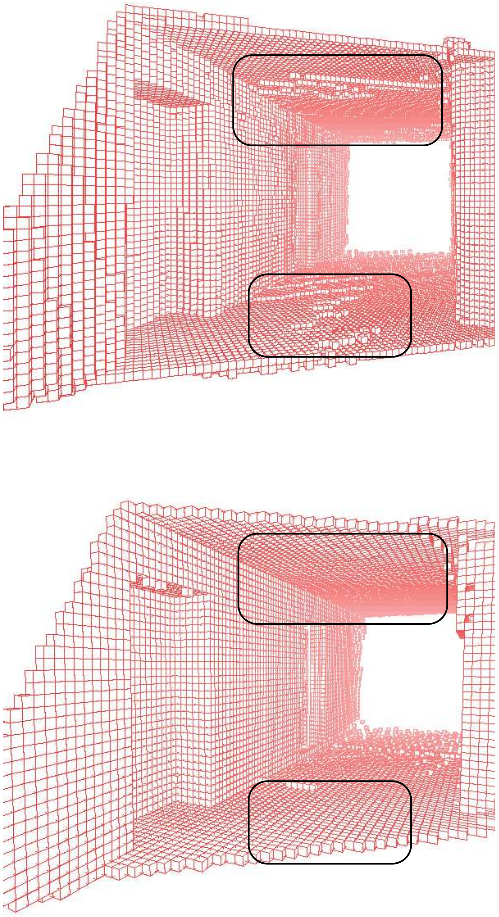

Figure 5. Top: Unconstrained mesh. Bottom: Constrained mesh.

6.2 Future work

Needless to say, much work remains to be done. Future work will concentrate on three aspects:

- Integrate a camera and enhance the semantic interpretation by fusing color images with range data. The aperture angle of the camera will be enlarged using a pan and tilt unit to acquire color information for all measured range points.
- Build an explicit knowledge base, i.e., specifying the semantic net and labels in a file, such that easy adoption to different domains with templates is possible.

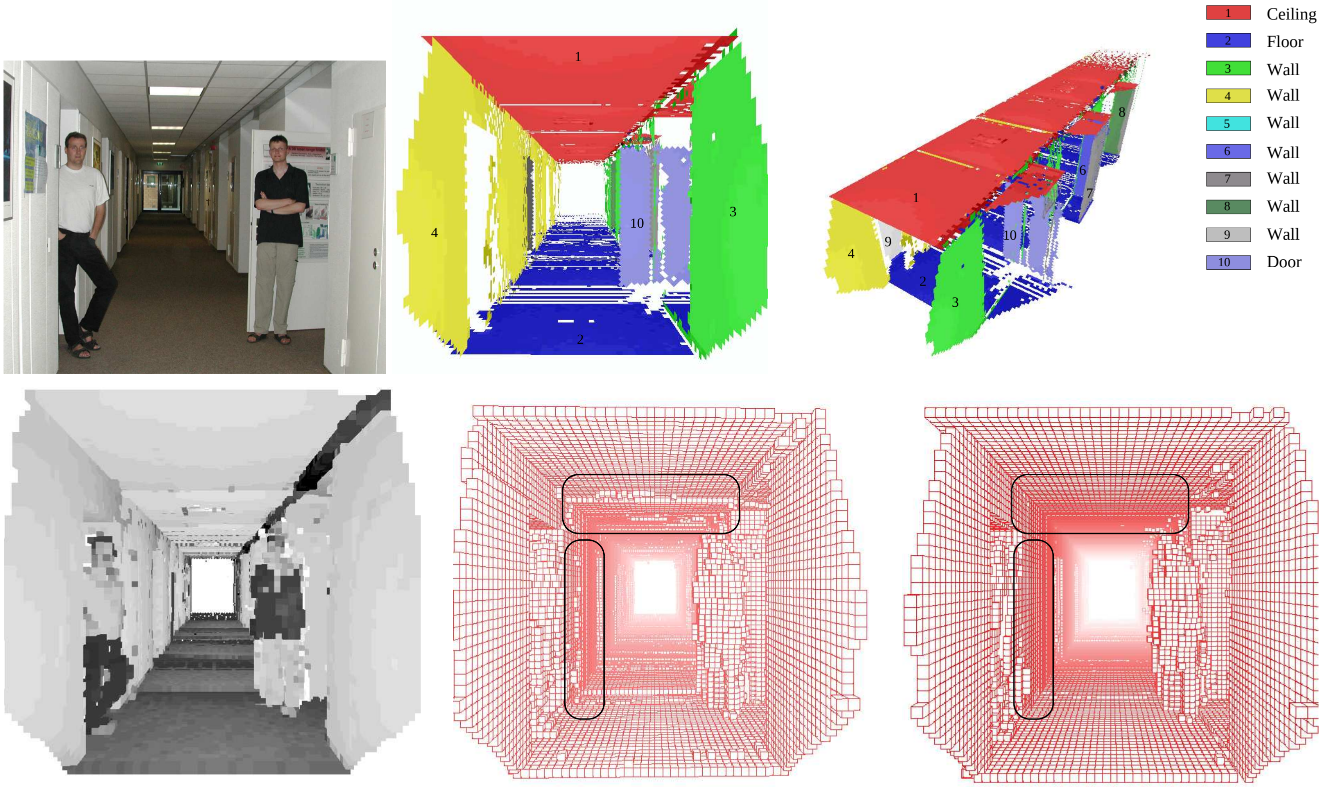

Figure 6. Example of a compact flat surface model reconstructed by an autonomous mobile robot (eight merged 3D scans). The persons in the scene are filtered out through the plane detection. Top left: Photo of the corridor scene. Top middle and right: Extracted surfaces with their semantic interpretation. Bottom left: Rendered scene with reflectance values. Bottom middle: Unconstrained mesh. Bottom right: Constrained mesh.

- Generate high level descriptions and semantic maps including the 3D information, e.g., in XML format. The semantic maps contain spatial 3D data with descriptions and labels.

6.3 Conclusion

This paper has presented a new approach to sensor and knowledge based reconstruction of 3D indoor environments with autonomous mobile robots. The proposed method consists of three steps and can be applied after the 3D data is acquired:

- The first step is a fast feature extraction, i.e., plane detection.
- Second the computed planes are labeled with a predefined semantic net. The semantic net contains and implements general knowledge of indoor scenes.
- Third the model is refined with the constraints arising from the semantic labeling. An numerical algorithm based on Powell's method is used for the 3D model improvement.

The proposed method is included in the robot control architecture for the automatic gaging of indoor environments.

Acknowledgement

Special thanks to Kai Lingemann and Kai Pervölz for supporting our work and to the unknown reviewers for their hints and suggestions.

References

[1] The RANSAC (Random Sample Consensus) Algorithm. `http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/FISHER/RANSAC/`, 2003.

[2] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. AVENUE: Automated Site Modelling in Urban Environments. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling (3DIM '01)*, Quebec City, Canada, May 2001.

[3] R. Benjemaa and F. Schmitt. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In *Proceedings IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM '97)*, Ottawa, Canada, May 1997.

[4] R. Benjemaa and F. Schmitt. A Solution For The Registration Of Multiple 3D Point Sets Using Unit Quaternions. *Computer Vision – ECCV '98*, 2:34 – 50, 1998.

[5] P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, February 1992.

[6] H. Cantzler, R. B. Fisher, and M. Devy. Quality enhancement of reconstructed 3D models using coplanarity and constraints. In *Proceedings of annual Symposium for Pattern Recognition (DAGM '02)*, pages 34 – 41, Zürich, Switzerland, September 2002.

[7] Y. Chen and G. Medoni. Object Modelling by Registration of Multiple Range Images. In *Proceedings of the IEEE Conferenc on Robotics and Automation (ICRA '91)*, pages 2724 – 2729, Sacramento, CA, USA, April 1991.

[8] D. Eggert, A. Fitzgibbon, and R. Fisher. Simultaneous Registration of Multiple Range Views Satisfying Global Consistency Constraints for Use In Reverse Engineering. *Computer Vision and Image Understanding*, 69:253 – 272, March 1998.

[9] R. B. Fisher. Applying knowledge to reverse engeniering problems. In *Proceedings of the international conference. Geometric Modeling and Processing (GMP '02)*, pages 149 – 155, Riken, Japan, July 2002.

[10] C. Früh and A. Zakhor. 3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR '01)*, Kauai, Hawaii, USA, December 2001.

[11] O. Grau. A Scene Analysis System for the Generation of 3-D Models. In *Proceedings IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM '97)*, pages 221 – 228, Ottawa, Canada, May 1997.

[12] D. Hähnel, W. Burgard, and S. Thrun. Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. In *Proceedings of the fourth European workshop on advanced mobile robots (EUROBOT '01)*, Lund, Sweden, September 2001.

[13] M. Hebert, M. Deans, D. Huber, B. Nabbe, and N. Vandapel. Progress in 3–D Mapping and Localization. In *Proceedings of the 9th International Symposium on Intelligent Robotic Systems, (SIRS '01)*, Toulouse, France, July 2001.

[14] B. Horn. Closed–form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629 – 642, April 1987.

[15] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to Learn 3D Models of Indoor Environments with Mobile Robots. In *Proceedings of the 18th Conference on Machine Learning*, Williams College, July 2001.

[16] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7:155 – 162, 1964.

[17] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, January 1993.

[18] K. Pulli. Multiview Registration for Large Data Sets. In *Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling (3DIM '99)*, pages 160 – 168, Ottawa, Canada, October 1999.

[19] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle. Robust meshes from multiple range maps. In *Proceedings IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM '97)*, Ottawa, Canada, May 1997.

[20] C. Robertson, R. B. Fisher, N. Werghi, and A. P. Ashbrook. Fitting of Constrained Feature Models to Poor 3D Data. In *Proceedings of the Adaptive Computing in Design and Manufacture (ACDM '00)*, pages 149 – 160, Plymouth, UK, April 2000.

[21] V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan–views. In *Proceedings of SPIE, Electronic Imaging '99, The Society for Imaging Science and Technology /SPIE's 11th Annual Symposium*, San Jose, CA, USA, January 1999.

[22] I. Stamos and P. Allen. 3-D Model Construction Using Range and Image Data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '00)*, USA, June 2000.

[23] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the of the 32nd International Symposium on Robotics (ISR '01)*, pages 153 – 158, Seoul, Korea, April 2001.

[24] H. Surmann, A. Nüchter, and J. Hertzberg. *Autonomous Mobile Robots for 3D Digitalization of Indoor Environments, GMD Report 147*. GMD - Forschungszentrum Informationstechnik GmbH, 2003.

[25] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA, USA, April 2000.

[26] H. Zhao and R. Shibasaki. Reconstructing Textured CAD Model of Urban Environment Using Vehicle-Borne Laser Range Scanners and Line Cameras. In *Second International Workshop on Computer Vision System (ICVS '01)*, pages 284 – 295, Vancouver, Canada, July 2001.