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

Dirk Hähnel [1] Wolfram Burgard [1] Sebastian Thrun [2]

Abstract

This paper presents an algorithm for full 3d shape reconstruction of indoor and outdoor environments with mobile robots. Data is acquired with laser range finders installed on a mobile robot. Our approach combines efficient scan matching routines 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.

Key words: Map building, 3d mapping, Model simplification

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 [1,2,3,4,5], and approaches that do not [6,7]. In the present paper, we are interested in using mobile

[1] Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany
[2] School of Computer Science, Carnegie Mellon University, Pittsburgh PA, USA

Article published in Elsevier Science Special Issue Eurobot'01 (2004) 1–16

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 [6,7] which generate highly complex models, our focus is on generating low-complexity models that can be rendered in real-time. The approach in [7], 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 [8], 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 mobile robots equipped with laser range finders. To estimate the pose of the robot while collecting the data, a probabilistic scan matching algorithm is used. The resulting pre-filtered data is globally consistent but locally 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., [9,10,11]), 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 [9] illustrates that our approach yields significantly more accurate and realistic-looking 3d models.

2 Computing Consistent Maps

Our current system is able to learn 2d and 3d maps using range scans recorded with a mobile robot. In both cases, the approach is incremental. Mathematically,

Figure 1. Two-dimensional map of Sieg Hall at University of Washington, Seattle, constructed out of 8013 2d range scans.

we calculate a sequence of poses $\hat{l}_1, \hat{l}_2, \ldots$ and corresponding maps by maximizing the marginal likelihood of the t-th pose and map relative to the $(t-1)$-th pose and map:

$$\hat{l}_t = \operatorname*{argmax}_{l_t} \{ p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) \cdot p(l_t \mid u_{t-1}, \hat{l}_{t-1}) \} \tag{1}$$

In this equation the term $p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$ is the probability of the most recent measurement s_t given the pose l_t and the map $\hat{m}(\hat{l}^{t-1}, s^{t-1})$ constructed so far. The term $p(l_t \mid u_{t-1}, \hat{l}_{t-1})$ represents the probability that the robot is at location l_t given the robot previously was at position $\hat{l}_{t-1}$ and has carried out (or measured) the motion u_{t-1}. The resulting pose $\hat{l}_t$ is then used to generate a new map $\hat{m}$ via the standard incremental map updating function presented in [12]:

$$\hat{m}(\hat{l}^t, s^t) = \operatorname*{argmax}_{m} p(m \mid \hat{l}^t, s^t) \tag{2}$$

The overall approach can be summarized as follows: At any point $t-1$ in time the robot is given an estimate of its pose $\hat{l}_{t-1}$ and a map $\hat{m}(\hat{l}^{t-1}, s^{t-1})$. After the robot moved further on and after taking a new measurement s_t, the robot determines the most likely new pose $\hat{l}_t$. It does this by trading off the consistency of the measurement with the map (first term on the right-hand side in (1)) and the consistency of the new pose with the control action and the previous pose (second term on the right-hand side in (1)). The map is then extended by the new measurement s_t, using the pose $\hat{l}_t$ as the pose at which this measurement was taken.

It remains to describe how we actually maximize Equation (1). Our system applies two different approaches depending on whether the scans to be aligned are two-dimensional or three-dimensional scans.

2.1 Two-dimensional Scan Alignment

Our algorithm to 2d scan matching is an extension of the approach presented in [7]. To align a scan relative to the previous scans, we use the map $\hat{m}(\hat{l}^{t-1,\Delta t}, s^{t-1})$. Additionally to [7] we integrate over small Gaussian errors in the robot pose when computing the maps. This avoids that many cells remain unknown especially if the scans contain long beams, increases the smoothness of the likelihood function to be optimized, and thus results in better alignments. To maximize the likelihood of a scan with respect to the given map, we apply a hill climbing strategy. A typical map resulting from this process is shown Figure 1. The size of the map is 50m times 14m.

2.2 Aligning Three-dimensional Range Scans

Unfortunately, three-dimensional variants of the maps and likelihood functions described above would consume too much memory. Therefore this approach is not applicable to 3d scan alignment. Instead, we represent 3d maps as triangle meshes constructed from the individual scans. We create a triangle for triples of neighboring scan points, if the maximum length of an edge does not exceed a certain threshold which depends on the length of the beams.

To compute the most likely position of a new 3d scan with respect to the current 3d model, we apply an approximative physical model of the range scanning process. Obviously, an ideal sensor would always measure the correct distance to the closest obstacle in the sensing direction. However, sensors and models generated out of range scanners are noisy. Therefore, our system incorporates measurement noise and random noise in order two deal with errors typically found in 3d range scans. First, we generally have normally distributed measurement errors around the distance "expected" according to the current position of the scanner and the given model of the environment. Additionally, we observe randomly distributed measurements because of errors in the model and because of deviations in the angles between corresponding beams in consecutive scans. Therefore, our model consists of a mixture of a Gaussian with a uniform distribution. The mode of the Gaussian corresponds to the distance expected given the current map. Additionally, we use a uniform distribution to deal with maximum range readings. To save computation time, we approximate the resulting distribution by a mixture of triangular distributions.

Whereas this approach saves memory, it requires more computation time than the technique for 2d scan alignment. However, in practical experiments we found out that this technique has two major advantages over the Iterative Closest Point (ICP) algorithm [13,14] and other similar scan-matching techniques. First, it exploits the

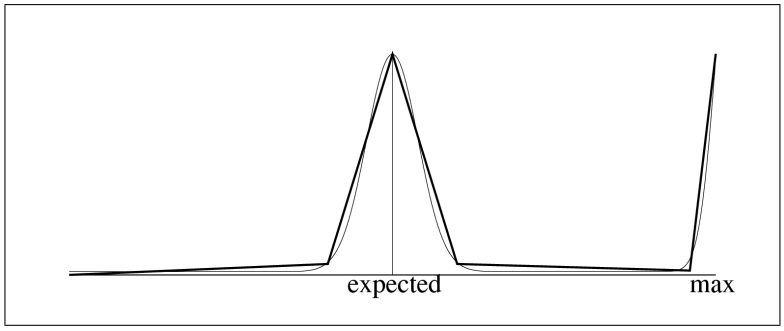

Figure 2. The probabilistic measurement model given as a mixture of a Gaussian and a uniform distribution and its approximation by piecewise linear functions.

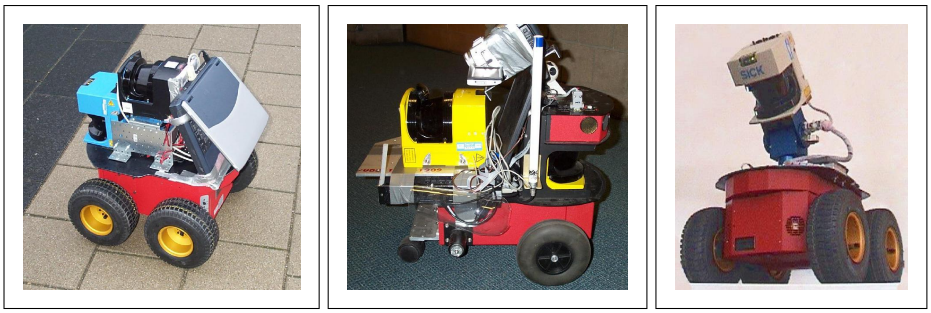

Figure 3. The platforms used for acquiring the 3d data. Outdoor system with two lasers (left), indoor system (middle), outdoor system with pan/tilt unit (right).

fact that each laser beam is a ray that does not go through surfaces and therefore does not require special heuristics for dealing with occlusions. Second, our approach also exploits the information provided by maximum range readings. For example, if such a beam goes through surfaces in the map, it reduces the likelihood of the current alignment.

To compute the likelihood of a beam b given the current map $\hat{m}(\hat{l}^{t-1}, s^{t-1})$, we first determine the expected distance $e(b, \hat{m}(\hat{l}^{t-1,\Delta t}, s^{t-1}))$ to the closest obstacle in the measurement direction. This is efficiently carried using ray-tracing techniques based on a spatial tiling and indexing [15] of the current map. Then we compute the likelihood of the measured distance given the expected distance, i.e. we determine the quantity $p(b \mid e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1})))$ using the mixture computed for $e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1,\Delta t}))$. Assuming that the beams contained in s_t are independent, we compute the likelihood of the whole scan as

$$p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) = \prod_{b \in s_t} p(b \mid e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1}))). \tag{3}$$

To maximize Equation 1 we again apply a hill climbing technique.

2.3 Generating Raw 3d Data

To record the 3d data used throughout this paper we used three different mobile platforms with two different kinds of laser configurations (see Figure 3). In the first configuration the robots carry two lasers. Whereas the first laser scans horizontally, the second laser is upward-pointed (see left and center image in Figure 3). These systems, which are used for in-door mapping, use the front laser to maps an unknown environment in 2d, thereby recovering their pose. At the same time the upward pointed laser scans the 3d structure of the environment. The robot depicted in the right image of Figure 3 is used for out-door environments only. It carries a single laser scanner that is mounted on a pan/tilt unit allowing the robot to dynamically change the scanning direction.

With all robots we obtain a polygonal model by connecting consecutive 3d points. To avoid closing wholes coming from doorways etc, we only create a polygonal surface if the consecutive points are close to each other. A typical model resulting from this process for the Wean Hall at Carnegie Mellon University is depicted in Figure 4. This data set was recorded with the robot depicted in the middle image of Figure 3

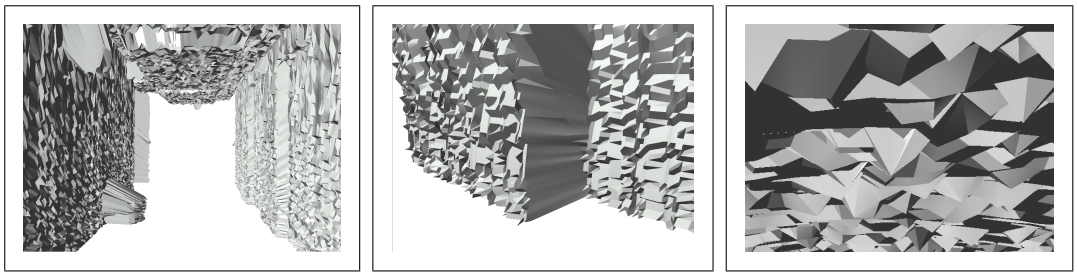

Figure 4. Model learned for a fraction of the Wean Hall at the Carnegie Mellon University (left) and fractions of the raw data for parts of the wall (center) and the ceiling (right).

3 Learning Smooth 3d Models

Although the position estimation techniques described in Sections 2.1 and 2.2 approach described above produces accurate position estimates, the resulting models often lack the appropriate smoothness. Figure 4 shows, in detail, a model of a corridor including parts of a doorway (see center image) and the ceiling (see right image). As it can be 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, consider the fractions of the doorway and the ceiling depicted in the right two images of Figure 4. 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.

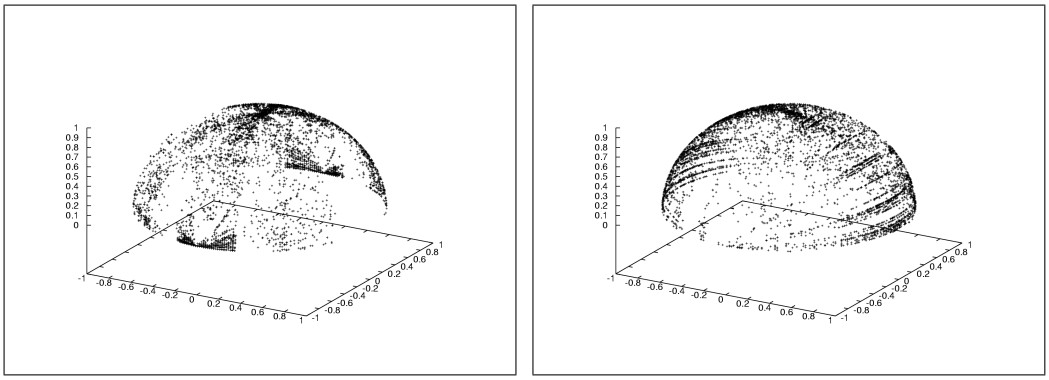

Figure 5. Normed surface normals for a ceiling (left) and a wall (right).

Please note that this 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 sufficiently small. However, the smaller is the distance between consecutive scanning positions, the higher is 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, more exhaustive techniques are required. In our system, we apply a randomized search technique 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

$$A = \sum_{i=1}^{n} (v_i - m)^T \cdot (v_i - m), \quad \text{where} \quad m = \frac{1}{n} \sum_{i=1}^{n} v_i \tag{4}$$

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 from 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.

Table 1. The plane extraction algorithm.

```
select point tuple v1, v2
Π := {v1, v2}
WHILE (new point can be found) BEGIN
    Select point v' with pointDist(Π, v') < δ
    if error(Π ∪ {v'}) < ε && ||(Π ∪ {v'}, v')|| < γ
        Π := Π ∪ {v'}
END WHILE
```

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.

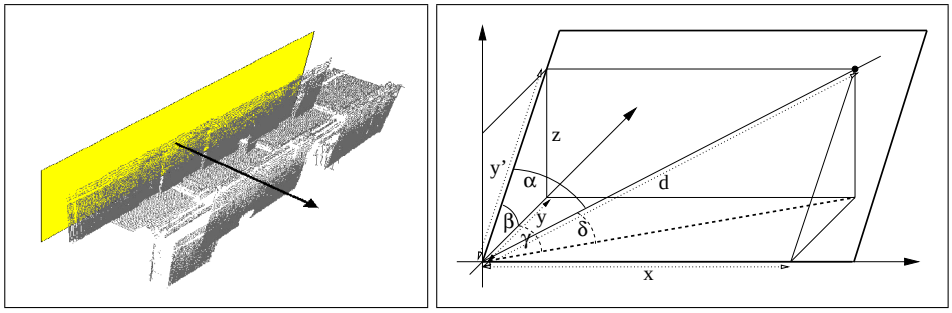

Figure 6. Sweeping a plane through a point set (left) and computing a “virtual” vertical 2d scan from a 3d scan.

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 [6].

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 [9,10], 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 [9,10] 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.

3.3 Improving the Efficiency

One of the major problems with the approach described above is its time complexity. For a typical data set consisting of 200,000 surfaces, a naive implementation on a standard PC requires over 10 hours to extract all planes. For environments like the ones considered here, a major issue therefore is the reduction of the overall search space.

To speed-up the plane extraction process, our system extracts lines out of the individual two-dimensional range-scans using the split-and merge technique which has been applied with great success in the past [16]. Given the Hessian normal form of the extracted lines we compute a histograms of the line angles. Thereby each line

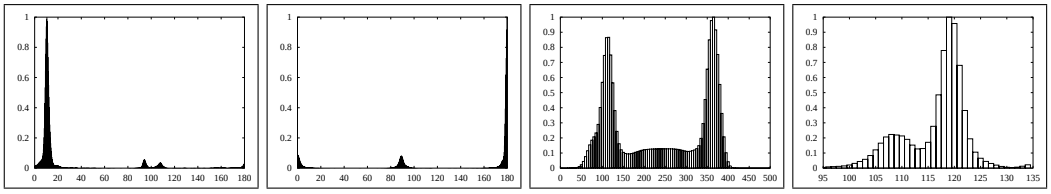

Figure 7. Histogram of the angles of line segments found in all scans of the horizontal (left image) and vertical (second image) scanner. Points belonging a plane for one sweep of a plane through the point set (third image). Points close to a plane found during a high-resolution sweep for the first peak in the resulting histogram (rightmost image).

is weighted proportionally to the number points that belong to it.

For the system with two range scanners, we compute a separate histogram for each scanner. Since the scans are obtained asynchronously, we treat both histograms independently and compute the histogram for possible plane directions based on the product of these two distributions. Figure 7 shows the histograms obtained for a fraction of Wean Hall data set (see Figure 4). Whereas the left image shows the predominant angles found in the horizontal scans, the right image shows the same for the vertical scanner. To extract the planes, our approach proceeds as follows. For each local maximum found in the histogram for vertical lines and each peak in the histogram for horizontal scans we construct the corresponding plane in 3d. We then sweep this plane along its normal through the data set (see Figure 6). For each possible plane, we perform two different sweeps with two different discretizations of the plane positions along its normal. The first sweep is carried out using a discretization of 5cm. The second sweep is then carried out for each peak found during the first sweep.

As an example consider the histograms shown in Figure 7. The left image of this Figure shows a plot of the number of points that are closer than 40cm to the corresponding plane for different positions the plane along its surface normal. Here we used the data set shown in Figure 4. The second plot shows the histogram that is computed from the data acquired with the vertical scanner. The histogram obtained by the first sweep is illustrated in the third image. Finally, the rightmost image in Figure 7 shows the histogram obtained using a sweep with a discretization of 1cm for the first peak in the third image of the same figure. As can be seen, the finer discretization shows a second peak. This peak comes from the doors in the environment which constitute planes that are slightly displaced from the walls.

For each peak in the high-resolution histogram, we collect all data points which are close to the corresponding plane and apply the plane extraction process described above. Because the number of points close to a plane generally much smaller than the overall number of points, we obtain an speed-up, which turned out to exceed one order of magnitude in all our experiments.

Please note that the same technique can be applied to 3d data gathered with the laser mounted on the pan-tilt unit. However, it requires certain geometric transfor-

mations to correctly identify vertical and horizontal lines from these data. For the sake of brevity, we only present the corresponding equations for the extraction of the angles of vertical lines. To compute angles of vertical lines we transform the data into a set of "virtual" vertical 2d-scans. Figure 6 shows the different parameters characterizing a single beam d of a 3d scan. The term α denotes the angle between d and center beam y' of the laser with in the currently scanned plane. The quantity β is the tilt of the laser scanner. To efficiently compute the vertical 2d-scan we sort all beams according to their vertical tilt angle δ and their angle γ to the center beam y' after both have been projected onto the horizontal plane. These two quantities can be computed out of α and β by straightforward geometric computations. Obviously, we have:

$$x = d \cdot \sin(\alpha), \quad y' = d \cdot \cos(\alpha), \quad y = y' \cdot \cos(\beta), \quad z = y' \cdot \sin(\beta)$$

Accordingly, $\tan(\gamma)$ and $\sin(\delta)$ can be computed as follows:

$$\tan(\gamma) = \frac{x}{y} = \frac{d \cdot \sin(\alpha)}{y' \cdot \cos(\beta)} = \frac{d \cdot \sin(\alpha)}{d \cdot \cos(\alpha) \cdot \cos(\beta)} = \frac{\tan(\alpha)}{\cos(\beta)}$$

$$\sin(\delta) = \frac{z}{d} = \frac{y' \cdot \sin(\beta)}{d} = \frac{d \cdot \cos(\alpha) \cdot \sin(\beta)}{d} = \cos(\alpha) \cdot \sin(\beta).$$

Thus, we obtain:

$$\gamma = \arctan(\frac{\tan(\alpha)}{\cos(\beta)}), \quad \delta = \arcsin(\cos(\alpha) \cdot \sin(\beta)) \tag{5}$$

The vertical 2d-scans are then extracted by selecting those points for which γ lies within a small interval around the currently considered angle γ^*.

4 Experimental Results

Our approach has been implemented and tested using three different platforms (see Figure 3), in indoor and outdoor environments. The robots were equipped with two 2d laser-range scanners or one 2d scanner mounted on a pan/tilt unit. Whereas the angular resolution of the laser used on the outdoor system is 0.25 degree, the angular resolutions of the lasers mounted on the indoor systems were 1 degree and 0.5 degrees. The resolution of the measured distances is 1cm and the measurement error of these systems lies between 0 and 20cm (SICK PLS) and 0 and 5cm (SICK LMS). The speed of the first two mobile platforms were 10cm/s.

The first experiment was carried out in the Wean Hall at the Carnegie Mellon University. Here the robot traveled 10m through a corridor and measured 140,923

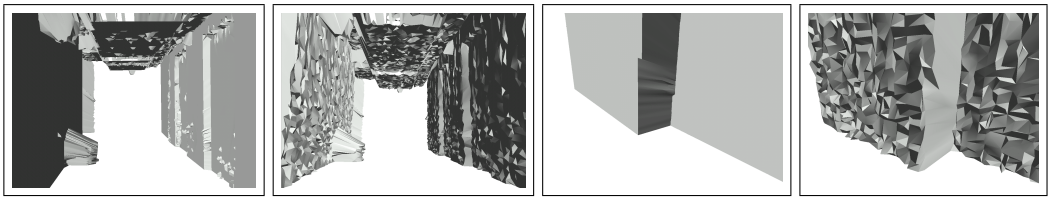

Figure 8. Approximations for the Wean Hall data set: Our approach (left) and QSlim (right). Magnified view of a doorway in the corridor environment: Our approach (left) and QSlim (right).

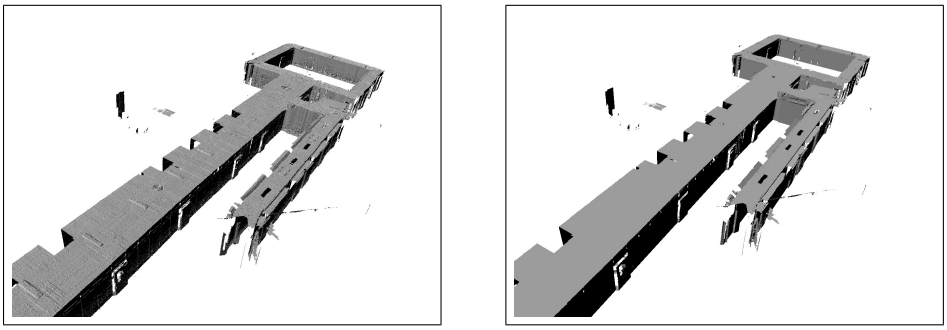

Figure 9. 3D data gathered in Sieg-Hall, University of Washington (left image) and simplified model (right image).

points in 3d using a SICK PLS laser scanner. The corresponding raw 3d data shown in Figure 4 consisted of 267,355 triangles. The result of our simplification technique is shown in the left image of Figure 8. For this data set our approach needed 6 minutes to compute the planes and generated 3613 polygons or quads (see Table 2). Only 34,227 triangles could not be approximated by larger planar structures. As a result, we obtained a significant reduction by 86% of the input data. The right image of Figure 8 shows the result of the QSlim system [9] 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 the quality obtained by the QSlim system. Figure 8 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.

The second experiment was carried out in a floor of the Sieg Hall at the University of Washington, Seattle. The robot measured 1,933,018 3d data points which are shown in the left image of Figure 9. Our approach needed 52 minutes to reduce the overall data set to 2,471 polygons and and 242 quads. 151,624 triangles could not be approximated by larger planar structures. The resulting smooth model is depicted in the right image of Figure 9.

Additionally, we applied our approach to the data set depicted in the left image of Figure 9. The right image shows the result after a planar approximation of the data. In this case our algorithm reduced the number of objects in the data by more than

Table 2. Statistics for the three data sets obtained in the experiments.

	Wean Hall	Sieg Hall	Campus
number of points	140,923	1,933,018	210,921
time [min]			
plane extraction	6:16	51:42	7:14
polygon merging	0:20	9:43	0:48
raw model			
number of triangles	267,355	3,765,072	377,896
reduced model			
number of polygons	2,626	2,471	255
number of quads	987	242	18
number of triangles	34,227	151,624	33,312
compression rate	86%	96%	91%

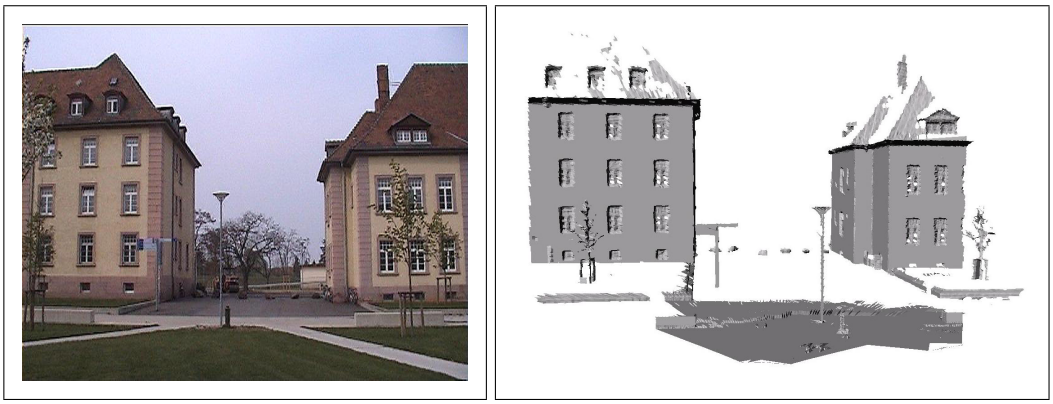

Figure 10. Photography of parts of two buildings at the University of Freiburg (left) and learned model (right).

one order of magnitude. A further, quite complex example is shown in the right image of Figure 10. The size of this area at the University of Freiburg is 40×60m. A photography of the same campus area is depicted in the left image of Figure 10. Obviously, the quality of the resulting model is quite good. In spite of the fact that the model contains several non-planar structures like tress, we obtained a significant reduction by 91%.

5 Related Work

Due to the various application areas like virtual reality, tele-presence, access to cultural savings, the problem of constructing 3d models has recently gained serious interest. The approaches described in [2,3,4,5] 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 [6] and [7]. 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. Recently, [17] developed an approach based on the EM-algorithm to determine planar approximations. In contrast to this method, our approach is inherently able to determine the number of planes in the data set.

The problem of polygonal simplification has been studied intensively in the computer graphics area [9,10,11]. 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.

Furthermore, several researchers have worked on the problem of range-image registration in the context of the construction of three-dimensional models and in the field of reverse engineering. A popular approach to combine several range-images into a single model is the ICP-algorithm [13]. This technique iteratively computes computes the displacement between two scans by matching the corresponding meshes in a point-wise. Due to the fact that it does not exploit the odometry information and the model of the robots motions, resulting estimates are not as good as with our method. In practice we observed several situations in which the ICP-algorithm diverges whereas our 2d and 3d scan matching techniques provides a correct result.

A further class of approaches addresses the problem of data segmentation and surface approximation out of segmented range data. For example, [18] presented tensor-voting to detect surfaces in range data. The goal is to compute a segmentation of the data and than extract features out of the segmented range data. Furthermore, there are several approaches that consider the problem of shape matching [19]. Finally there a approaches that consider the problem of template matching in range scans. Among them are feature-based techniques [20] and other techniques that use binary decision trees to speed-up the search [21]. Compared to these techniques, our approach can be regarded as a pre-processing step that first reduces the data to planar and non-planar structures. In this case, the techniques mentioned need only to be applied to the non-planar parts of the models generated with our system.

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

References

[1] P. Allen, I. Stamos, Integration of range and image sensing for photorealistic 3D modeling, in: Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2000, pp. 1435–1440.

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

[3] S. Becker, 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] P. Debevec, C. Taylor, J. Malik, Modeling and rendering architecture from photographs, in: Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), 1996.

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

[6] S. Hakim, P. Boulanger, 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.

[7] S. Thrun, W. Burgard, D. Fox, A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping, in: Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2000.

[8] M. B. L. Iocchi, K. Konolige, Visually realistic mapping of a planar environment with stereo, in: Proc. of the 2000 International Symposium on Experimental Robotics, Waikiki, Hawaii, 2000.

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

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

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

[12] H. Moravec, Sensor fusion in certainty grids for mobile robots, AI Magazine (1988) 61–74.

[13] P. Besl, N. McKay, A method for registration of 3d shapes, Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.

[14] M. Greenspan, G. Godin, A nearest neighbor method for efficient ICP, in: Proc. of the 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM01), 2001.

[15] H. Samet, Applications of Spatial Data Structures, Addison-Wesley Publishing Inc., 1989.

[16] J.-S. Gutmann, Robuste Navigation autonomer mobiler Systeme, Ph.D. thesis, University of Freiburg, in German (2000).

[17] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, S. Thrun, Using EM to learn 3D models of indoor environments with mobile robots, in: Proc. of the International Conference on Machine Learning (ICML), 2001.

[18] G. Medioni, M. Lee, C. Tang, A Computational Framework for Segmentation and Grouping, Elsevier Science, 2000.

[19] R. Veltkamp, M. Hagedoorn, State-of-the-art in shape matching, in: M. Lew (Ed.), Principles of Visual Information Retrieval, Springer Verlag, 2001.

[20] F. Ferrie, S. Mathur, G. Soucy, Feature extraction for 3-d model building and object recognition, in: A. Jain, P. Flynn (Eds.), 3D Object Recognition Systems, Elsevier, 1993.

[21] M. Greenspan, The sample tree: A sequential hypothesis testing approach to 3d object recognition, in: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1998.