Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM

Andreas Nüchter[1], Kai Lingemann[1], Joachim Hertzberg[1],
and Hartmut Surmann[2]

[1] University of Osnabrück, Institute of Computer Science,
Knowledge Based Systems Research Group,
Albrechtstr. 28, D-49069 Osnabrück, Germany
{nuechter, lingemann, hertzberg}@informatik.uni-osnabrueck.de
[2] Fraunhofer Institute for Autonomous Intelligent Systems,
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
hartmut.surmann@ais.fraunhofer.de

Abstract. 6D SLAM (Simultaneous Localization and Mapping) or 6D Concurrent Localization and Mapping of mobile robots considers six dimensions for the robot pose, namely, the x, y and z coordinates and the roll, yaw and pitch angles. Robot motion and localization on natural surfaces, e.g., driving with a mobile robot outdoor, must regard these degrees of freedom. This paper presents a robotic mapping method based on locally consistent 3D laser range scans. Scan matching, combined with a heuristic for closed loop detection and a global relaxation method, results in a highly precise mapping system for outdoor environments. The mobile robot Kurt3D was used to acquire data of the Schloss Birlinghoven campus. The resulting 3D map is compared with ground truth, given by an aerial photograph.

1 Introduction

Automatic environment sensing and modeling is a fundamental scientific issue in robotics, since the presence of maps is essential for many robot tasks. Manual mapping of environments is a hard and tedious job: Thrun et al. report a time of about one week hard work for creating a map of the museum in Bonn for the robot RHINO [25]. Especially mobile systems with 3D laser scanners that automatically perform multiple steps such as scanning, gaging and autonomous driving have the potential to greatly improve mapping. Many application areas benefit from 3D maps, e.g., industrial automation, architecture, agriculture, the construction or maintenance of tunnels and mines and rescue robotic systems.

The robotic mapping problem is that of acquiring a spatial model of a robot's environment. If the robot poses were known, the local sensor inputs of the robot, i.e., local maps, could be registered into a common coordinate system to create a map. Unfortunately, any mobile robot's self localization suffers from imprecision and therefore the structure of the local maps, e.g., of single scans, needs to be

U. Furbach (Eds.): KI 2005, LNAI 3698, pp. 313–328, 2005.

used to create a precise global map. Finally, robot poses in natural outdoor environments involve yaw, pitch, roll angles and elevation, turning pose estimation as well as scan registration into a problem in six mathematical dimensions.

This paper proposes algorithms that allow to digitize large environments and solve the 6D SLAM problem. In previous works we already presented partially our 6D SLAM algorithm [19,23,24]. In [19] we use a global relaxation scan matching algorithm to create a model of an abandoned mine and in [24] we presented our first 3D model containing a closed loop. This paper's main contribution is an octree-based matching heuristic that allows us to match scans with rudimentary starting guesses and to detect closed loops.

1.1 Related Work

SLAM. Depending on the map type, mapping algorithms differ. State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion and uncertain perception models. By integrating of these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, i.e., a second encounter of a previously visited area of the environment, play a special role here. Once detected, they enable the algorithms to bound the error by deforming the already mapped area such that a topologically consistent model is created. However, there is no guarantee for a correct model. Several strategies exist for solving SLAM. Thrun reviews in [26] existing techniques, i.e., maximum likelihood estimation [10], expectation maximization [9,27], extended Kalman filter [6] or (sparse extended) information filter [29]. In addition to these methods, FastSLAM [28] that approximates the posterior probabilities, i.e., robot poses, by particles, and the method of Lu Milios on the basis of IDC scan matching [18] exist.

In principle, these probabilistic methods are extendable to 6D. However no reliable feature extraction nor a strategy for reducing the computational costs of multi hypothesis tracking, e.g., FastSLAM, that grows exponentially with the degrees of freedom, has been published to our knowledge.

3D Mapping. Instead of using 3D scanners, which yield consistent 3D scans in the first place, some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. Thrun et al. [28], Früh et al. [11] and Zhao et al. [31] use two 2D laser scanners finders for acquiring 3D data. One laser scanner is mounted horizontally, the other vertically. The latter one grabs a vertical scan line which is transformed into 3D points based on the current robot pose. Since the vertical scanner is not able to scan sides of objects, Zhao et al. use two additional, vertically mounted 2D scanners, shifted by 45° to reduce occlusions [31]. 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 highly accurate, expensive 3D laser scanners [1,12,22]. The RESOLV project aimed at modeling interiors for virtual reality and telepresence [22]. They used a RIEGL laser range finder on robots and the ICP algorithm for scan matching [4]. The AVENUE project develops a robot for modeling urban environments [1], using a CYRAX scanner and a feature-based scan matching approach for registering the 3D scans. Nevertheless, in their recent work they do not use data of the laser scanner in the robot control architecture for localization [12]. The group of M. Hebert has reconstructed environments using the Zoller+Fröhlich laser scanner and aims to build 3D models without initial position estimates, i.e., without odometry information [14].

Recently, different groups employ rotating SICK scanners for acquiring 3D data [15,30]. Wulf et al. let the scanner rotate around the vertical axis. They acquire 3D data while moving, thus the quality of the resulting map crucially depends on the pose estimate that is given by inertial sensors, i.e., gyros [30]. In addition, their SLAM algorithms do not consider all six degrees of freedom.

Other approaches use information of CCD-cameras that provide a view of the robot's environment [5,21]. Nevertheless, cameras are difficult to use in natural environments with changing light conditions. Camera-based approaches to 3D robot vision, e.g., stereo cameras and structure from motion, have difficulties providing reliable navigation and mapping information for a mobile robot in real-time. Thus some groups try to solve 3D modeling by using planar scanner based SLAM methods and cameras, e.g., in [5].

1.2 Hardware Used in Our Experiments

The 3D Laser Range Finder. The 3D laser range finder (Fig. 1) [23] is built on the basis of a SICK 2D range finder by extension with a mount and a small servomotor. The 2D laser range finder is attached in the center of rotation to the mount for achieving a controlled pitch motion with a standard servo.

The area of up to 180°(h) × 120°(v) is scanned with different horizontal (181, 361, 721) and vertical (128, 256, 400, 500) 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 × 256 data points needs 3.4 seconds. Scanning the environment with a mobile robot is done in a stop-scan-go fashion.

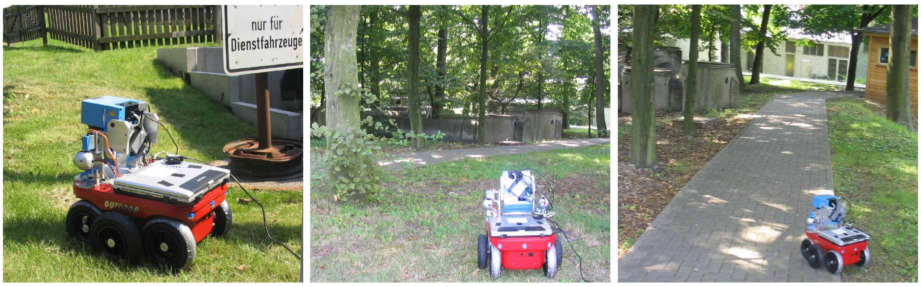

Fig. 1. Kurt3D in a natural environment. Left to right: Lawn, forest track, pavement.

The Mobile Robot. Kurt3D Outdoor (Fig. 1) is a mobile robot with a size of 45 cm (length) × 33 cm (width) × 29 cm (height) and a weight of 22.6 kg. Two 90 W motors are used to power the 6 skid-steered wheels, whereas the front and rear wheels have no tread pattern to enhance rotating. The core of the robot is a Pentium-Centrino-1400 with 768 MB RAM and Linux.

2 Range Image Registration and Robot Relocalization

Multiple 3D scans are necessary to digitalize environments without occlusions. To create a correct and consistent model, the scans have to be merged into one coordinate system. This process is called registration. If the robot carrying the 3D scanner were precisely localized, the registration could be done directly based on the robot pose. However, due to the unprecise robot sensors, self localization is erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration. As a by-product, successful registration of 3D scans relocalizes the robot in 6D, by providing the transformation to be applied to the robot pose estimation at the recent scan point.

The following method registers point sets in a common coordinate system. It is called *Iterative Closest Points (ICP)* algorithm [4]. Given two independently acquired sets of 3D points, M (model set) and D (data set) which correspond to a single shape, we aim to find the transformation consisting of a rotation $\mathbf{R}$ and a translation $\boldsymbol{t}$ which minimizes the following cost function:

$$E(\mathbf{R}, \boldsymbol{t}) = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j} \left|\left| \boldsymbol{m}_i - (\mathbf{R}\boldsymbol{d}_j + \boldsymbol{t}) \right|\right|^2 . \quad (1)$$

$w_{i,j}$ is assigned 1 if the i-th point of M describes the same point in space as the j-th point of D. Otherwise $w_{i,j}$ is 0. Two things have to be calculated: First, the corresponding points, and second, the transformation ($\mathbf{R}$, $\boldsymbol{t}$) that minimizes $E(\mathbf{R}, \boldsymbol{t})$ on the base of the corresponding points.

The ICP algorithm calculates iteratively the point correspondences. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation $(\mathbf{R}, \boldsymbol{t})$ for minimizing equation (1). The assumption is that in the last iteration step the point correspondences are correct. Besl et al. prove that the method terminates in a minimum [4]. However, this theorem does not hold in our case, since we use a maximum tolerable distance $d_{\max}$ for associating the scan data. Such a threshold is required though, given that 3D scans overlap only partially.

In every iteration, the optimal transformation ($\mathbf{R}$, $\boldsymbol{t}$) has to be computed. Eq. (1) can be reduced to

$$E(\mathbf{R}, \boldsymbol{t}) \propto \frac{1}{N} \sum_{i=1}^{N} \left|\left| \boldsymbol{m}_i - (\mathbf{R}\boldsymbol{d}_i + \boldsymbol{t}) \right|\right|^2 , \quad (2)$$

with $N = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j}$, since the correspondence matrix can be represented by a vector containing the point pairs.

Four direct methods are known to minimize eq. (2) [17]. In earlier work [19,23,24] we used a quaternion based method [4], but the following one, based on singular value decomposition (SVD), is robust and easy to implement, thus we give a brief overview of the SVD-based algorithm. It was first published by Arun, Huang and Blostein [2]. The difficulty of this minimization problem is to enforce the orthonormality of the matrix $\mathbf{R}$. The first step of the computation is to decouple the calculation of the rotation $\mathbf{R}$ from the translation $\boldsymbol{t}$ using the centroids of the points belonging to the matching, i.e.,

$$\boldsymbol{c}_m = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{m}_i, \qquad \boldsymbol{c}_d = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{d}_j \tag{3}$$

and

$$M' = \{\boldsymbol{m}'_i = \boldsymbol{m}_i - \boldsymbol{c}_m\}_{1,\dots,N}, \qquad D' = \{\boldsymbol{d}'_i \ = \boldsymbol{d}_i \ - \boldsymbol{c}_d\}_{1,\dots,N}. \tag{4}$$

After substituting (3) and (4) into the error function, $E(\mathbf{R}, \boldsymbol{t})$ eq. (2) becomes:

$$E(\mathbf{R}, \boldsymbol{t}) \propto \sum_{i=1}^{N} \left|\left|\boldsymbol{m}'_i - \mathbf{R}\boldsymbol{d}'_i\right|\right|^2 \quad \text{with} \quad \boldsymbol{t} = \boldsymbol{c}_m - \mathbf{R}\boldsymbol{c}_d. \tag{5}$$

The registration calculates the optimal rotation by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. Hereby, the matrices $\mathbf{V}$ and $\mathbf{U}$ are derived by the singular value decomposition $\mathbf{H} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$ of a correlation matrix $\mathbf{H}$. This 3×3 matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \sum_{i=1}^{N} \boldsymbol{d}'_i \boldsymbol{m}'^T_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}, \tag{6}$$

with $S_{xx} = \sum_{i=1}^{N} \ m'_{ix} d'_{ix}$, $S_{xy} = \sum_{i=1}^{N} \ m'_{ix} d'_{iy}$, $\dots$ [2].

We proposed and evaluated algorithms to accelerate ICP, namely point reduction and approximate kd-trees [19,23,24]. They are used here, too.

3 ICP-Based 6D SLAM

3.1 Calculating Heuristic Initial Estimations for ICP Scan Matching

To match two 3D scans with the ICP algorithm it is necessary to have a sufficient starting guess for the second scan pose. In earlier work we used odometry [23] or the planar HAYAI scan matching algorithm [16]. However, the latter cannot be used in arbitrary environments, e.g., the one presented in Fig. 1 (bad asphalt, lawn, woodland, etc.). Since the motion models change with different grounds, odometry alone cannot be used. Here the robot pose is the 6-vector $\boldsymbol{P} = (x, y, z, \theta_x, \theta_y, \theta_z)$ or, equivalently the tuple containing the rotation matrix and translation vector, written as 4×4 OpenGL-style matrix $\mathbf{P}$ [8].[1] The following heuristic computes a sufficiently good initial estimation. It is based on two ideas. First, the transformation found in the previous registration is applied

[1] Note the bold-italic (vectors) and bold (matrices) notation. The conversion between vector representations, i.e., Euler angles, and matrix representations is done by algorithms from [8].

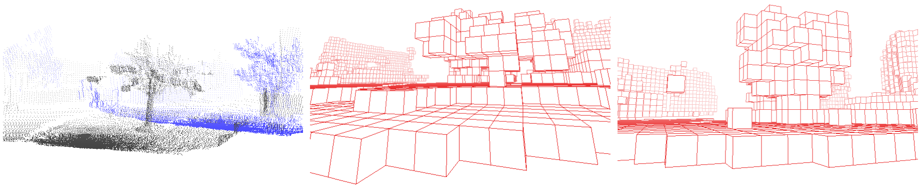

Fig. 2. Left: Two 3D point clouds. Middle: Octree corresponding to the black point cloud. Right: Octree based on the blue points.

to the pose estimation – this implements the assumption that the error model of the pose estimation is locally stable. Second, a pose update is calculated by matching octree representations of the scan point sets rather than the point sets themselves – this is done to speed up calculation:

1. Extrapolate the odometry readings to all six degrees of freedom using previous registration matrices. The change of the robot pose $\Delta\mathbf{P}$ given the odometry information $(x_n, z_n, \theta_{y,n})$, $(x_{n+1}, z_{n+1}, \theta_{y,n+1})$ and the registration matrix $\mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})$ is calculated by solving:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ \theta_{x,n+1} \\ \theta_{y,n+1} \\ \theta_{z,n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ z_n \\ \theta_{x,n} \\ \theta_{y,n} \\ \theta_{z,n} \end{pmatrix} + \left(\begin{array}{c|c} \mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n}) & \mathbf{0} \\ \hline \mathbf{0} & \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \end{array} \right) \cdot \underbrace{\begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta z_{n+1} \\ \Delta\theta_{x,n+1} \\ \Delta\theta_{y,n+1} \\ \Delta\theta_{z,n+1} \end{pmatrix}}_{\Delta\boldsymbol{P}} .$$

   Therefore, calculating $\Delta\boldsymbol{P}$ requires a matrix inversion. Finally, the 6D pose $\mathbf{P}_{n+1}$ is calculated by

$$\mathbf{P}_{n+1} = \Delta\mathbf{P} \cdot \mathbf{P}_n$$

   using the poses' matrix representations.

2. Set $\Delta\boldsymbol{P}_{\text{best}}$ to the 6-vector $(\boldsymbol{t}, \boldsymbol{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})) = (\boldsymbol{0}, \boldsymbol{R}(\boldsymbol{0}))$.
3. Generate an octree $\mathfrak{O}_M$ for the nth 3D scan (model set M).
4. Generate an octree $\mathfrak{O}_D$ for the $(n+1)$th 3D scan (data set D).
5. For search depth $t \in [t_{\text{Start}}, \ldots, t_{\text{End}}]$ in the octrees estimate a transformation $\Delta\boldsymbol{P}_{\text{best}} = (\boldsymbol{t}, \boldsymbol{R})$ as follows:
   (a) Calculate a maximal displacement and rotation $\Delta\boldsymbol{P}_{\text{max}}$ depending on the search depth t and currently best transformation $\Delta\boldsymbol{P}_{\text{best}}$.
   (b) For all discrete 6-tuples $\Delta\boldsymbol{P}_i \in [-\Delta\boldsymbol{P}_{\text{max}}, \Delta\boldsymbol{P}_{\text{max}}]$ in the domain $\Delta\boldsymbol{P} = (x, y, z, \theta_x, \theta_y, \theta_z)$ displace $\mathfrak{O}_D$ by $\Delta\mathbf{P}_i \cdot \Delta\mathbf{P} \cdot \mathbf{P}_n$. Evaluate the matching of the two octrees by counting the number of overlapping cubes and save the best transformation as $\Delta\boldsymbol{P}_{\text{best}}$.

6. Update the scan pose using matrix multiplication, i.e.,

$$\mathbf{P}_{n+1} = \varDelta\mathbf{P}_{\text{best}} \cdot \varDelta\mathbf{P} \cdot \mathbf{P}_n.$$

Note: Step 5b requires 6 nested loops, but the computational requirements are bounded by the coarse-to-fine strategy inherited from the octree processing. The size of the octree cubes decreases exponentially with increasing t. We start the algorithm with a cube size of 75 cm^3 and stop when the cube size falls below 10 cm^3. Fig. 2 shows two 3D scans and the corresponding octrees. Furthermore, note that the heuristic works best outdoors. Due to the diversity of the environment the match of octree cubes will show a significant maximum, while indoor environments with their many geometry symmetries and similarities, e.g., in a corridor, are in danger of producing many plausible matches.

After an initial starting guess is found, the range image registration from section 2 proceeds and the 3D scans are precisely matched.

3.2 Computing Globally Consistent Scenes

After registration, the scene has to be correct and globally consistent. A straightforward method for aligning several 3D scans is *pairwise matching*, i.e., the new scan is registered against a previous one. Alternatively, an *incremental matching* method is introduced, i.e., the new scan is registered against a so-called *metascan*, which is the union of the previously acquired and registered scans. Each scan matching has a limited precision. Both methods accumulate the registration errors such that the registration of a large number of 3D scans leads to inconsistent scenes and to problems with the robot localization. Closing loop detection and error diffusing avoid these problems and compute consistent scenes.

Closing the Loop. After matching multiple 3D scans, errors have accumulated and loops would normally not be closed. Our algorithm automatically detects a to-be-closed loop by registering the last acquired 3D scan with earlier acquired scans. Hereby we first create a hypothesis based on the maximum laser range and on the robot pose, so that the algorithm does not need to process all previous scans. Then we use the octree based method presented in section 3.1 to revise the hypothesis. Finally, if a registration is possible, the computed error, i.e., the transformation ($\mathbf{R}$, $\boldsymbol{t}$) is distributed over all 3D scans. The respective part is weighted by the distance covered between the scans, i.e.,

$$c_i = \frac{\text{length of path from start of the loop to scan pose } i}{\text{overall length of path}}$$

1. The translational part is calculated as $\boldsymbol{t}_i = c_i\boldsymbol{t}$.
2. Of the three possibilities of representing rotations, namely, orthonormal matrices, quaternions and Euler angles, quaternions are best suited for our interpolation task. The problem with matrices is to enforce orthonormality and Euler angles show Gimbal locks [8]. A quaternion as used in computer graphics is the 4 vector $\dot{\boldsymbol{q}}$. Given a rotation as matrix $\mathbf{R}$, the corresponding quaternion $\dot{\boldsymbol{q}}$ is calculated as follows:

$$\dot{\boldsymbol{q}} = \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sqrt{\operatorname{trace}(\mathbf{R})} \\ \frac{1}{2}\frac{r_{3,3}-r_{3,2}}{\sqrt{\operatorname{trace}(\mathbf{R})}} \\ \frac{1}{2}\frac{r_{2,1}-r_{2,3}}{\sqrt{\operatorname{trace}(\mathbf{R})}} \\ \frac{1}{2}\frac{r_{1,2}-r_{1,1}}{\sqrt{\operatorname{trace}(\mathbf{R})}} \end{pmatrix}, \quad \text{with the elements } r_{i,j} \text{ of } \mathbf{R}.\text{[2]} \tag{7}$$

The quaternion describes a rotation by an axis $\boldsymbol{a} \in \mathbb{R}^3$ and an angle θ that are computed by

$$\boldsymbol{a} = \begin{pmatrix} \frac{q_x}{\sqrt{1-q_0^2}} \\ \frac{q_y}{\sqrt{1-q_0^2}} \\ \frac{q_z}{\sqrt{1-q_0^2}} \end{pmatrix} \qquad \text{and} \qquad \theta = 2\arccos q_o.$$

The angle θ is distributed over all scans using the factor c_i and the resulting matrix is derived as [8]:

$$\mathbf{R}_i = \left(\begin{matrix} \cos(c_i\theta) + a_x^2(1-\cos(c_i\theta)) & a_z\sin(c_i\theta) + a_x a_y(1-\cos(c_i\theta)) \\ -a_z\sin(c_i\theta) + a_x a_y(1-\cos(c_i\theta)) & \cos(c_i\theta) + a_y^2(1-\cos(c_i\theta)) \\ a_y\sin(c_i\theta) + a_x a_z(1-\cos(c_i\theta) & -a_x\sin(c_i\theta) + a_y a_z(1-\cos(c_i\theta)) \end{matrix} \right.$$

$$\left. \begin{matrix} -a_y\sin(c_i\theta) + a_x a_z(1-\cos(c_i\theta)) \\ -a_x\sin(c_i\theta) + a_y a_z(1-\cos(c_i\theta)) \\ \cos(c_i\theta) + a_z^2(1-\cos(c_i\theta)) \end{matrix} \right). \tag{8}$$

The next step minimizes the global error.

[2] If trace $(\mathbf{R})$ (sum of the diagonal terms) is zero, the above calculation has to be altered: Iff $r_{1,1} > r_{2,2}$ and $r_{1,1} > r_{3,3}$ then,

$$\dot{\boldsymbol{q}} = \begin{pmatrix} \frac{1}{2}\frac{r_{2,3}-r_{3,2}}{\sqrt{1+r_{1,1}-r_{2,2}-r_{3,3}}} \\ \frac{1}{2}\sqrt{1+r_{1,1}-r_{2,2}-r_{3,3}} \\ \frac{1}{2}\frac{r_{1,2}+r_{2,1}}{\sqrt{1+r_{1,1}-r_{2,2}-r_{3,3}}} \\ \frac{1}{2}\frac{r_{3,1}+r_{1,3}}{\sqrt{1+r_{1,1}-r_{2,2}-r_{3,3}}} \end{pmatrix}, \text{if } r_{2,2} > r_{3,3}\ \dot{\boldsymbol{q}} = \begin{pmatrix} \frac{1}{2}\frac{r_{3,1}-r_{1,3}}{\sqrt{1-r_{1,1}+r_{2,2}-r_{3,3}}} \\ \frac{1}{2}\frac{r_{1,2}+r_{2,1}}{\sqrt{1-r_{1,1}+r_{2,2}-r_{3,3}}} \\ \frac{1}{2}\sqrt{1-r_{1,1}+r_{2,2}-r_{3,3}} \\ \frac{1}{2}\frac{r_{2,3}+r_{3,2}}{\sqrt{1-r_{1,1}+r_{2,2}-r_{3,3}}} \end{pmatrix},$$

otherwise the quaternion $\dot{\boldsymbol{q}}$ is calculated as

$$\dot{\boldsymbol{q}} = \begin{pmatrix} \frac{1}{2}\frac{r_{1,2}-r_{2,1}}{\sqrt{1-r_{1,1}-r_{2,2}+r_{3,3}}} \\ \frac{1}{2}\frac{r_{3,1}+r_{1,3}}{\sqrt{1+r_{1,1}-r_{2,2}-r_{3,3}}} \\ \frac{1}{2}\frac{r_{2,3}+r_{3,2}}{\sqrt{1-r_{1,1}-r_{2,2}+r_{3,3}}} \\ \frac{1}{2}\sqrt{1-r_{1,1}-r_{2,2}+r_{3,3}} \end{pmatrix}.$$

Diffusing the Error. Pulli presents a registration method that minimizes the global error and avoids inconsistent scenes [20]. The registration of one scan is followed by registering all neighboring scans such that the global error is distributed. Other matching approaches with global error minimization have been published, e.g., [3,7]. Benjemaa et al. establish point-to-point correspondences first and then use randomized iterative registration on a set of surfaces [3]. Eggert et al. compute motion updates, i.e., a transformation $(\mathbf{R}, \mathbf{t})$, using force-based optimization, with data sets considered as connected by groups of springs [7].

Based on the idea of Pulli we designed the relaxation method *simultaneous matching*[23]. The first scan is the masterscan and determines the coordinate system. It is fixed. The following three steps register all scans and minimize the global error, after a queue is initialized with the first scan of the closed loop:

1. Pop the first 3D scan from the queue as the current one.
2. 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 forms one point set M. The current scan forms the data point set D and is aligned with the ICP algorithms. One scan overlaps with another iff more than p corresponding point pairs exist. In our implementation, $p = 250$.
3. If the current scan changes its location by applying the transformation (translation or rotation) in step 2, then each single scan of the set of neighbors that is not in the queue is added to the end of the queue. If the queue is empty, terminate; else continue at step 1.

In contrast to Pulli's approach, our method is totally automatic and no interactive pairwise alignment has to be done. Furthermore the point pairs are not fixed [20]. The accumulated alignment error is spread over the whole set of acquired 3D scans. This diffuses the alignment error equally over the set of 3D scans [24].

4 Experiment and Results

The following experiment has been made at the campus of Schloss Birlinghoven with Kurt3D. Fig. 3 (left) shows the scan point model of the first scans in top view, based on odometry only. The first part of the robot's run, i.e., driving on asphalt, contains a systematic drift error, but driving on lawn shows more stochastic characteristics. The right part shows the first 62 scans, covering a path length of about 240 m. The heuristic has been applied and the scans have been matched. The open loop is marked with a red rectangle.

At that point, the loop is detected and closed. More 3D scans have then been acquired and added to the map. Fig. 4 (left and right) shows the model with and without global relaxation to visualize its effects. The relaxation is able to align the scans correctly even without explicitly closing the loop. The best visible difference is marked by a red rectangle. The final map in Fig. 4 contains 77 3D scans, each consisting of approx. 100000 data points (275 × 361). Fig. 5 shows two detailed views, before and after loop closing. The bottom part of Fig. 4 displays an aerial view as ground truth for comparison. Table 1

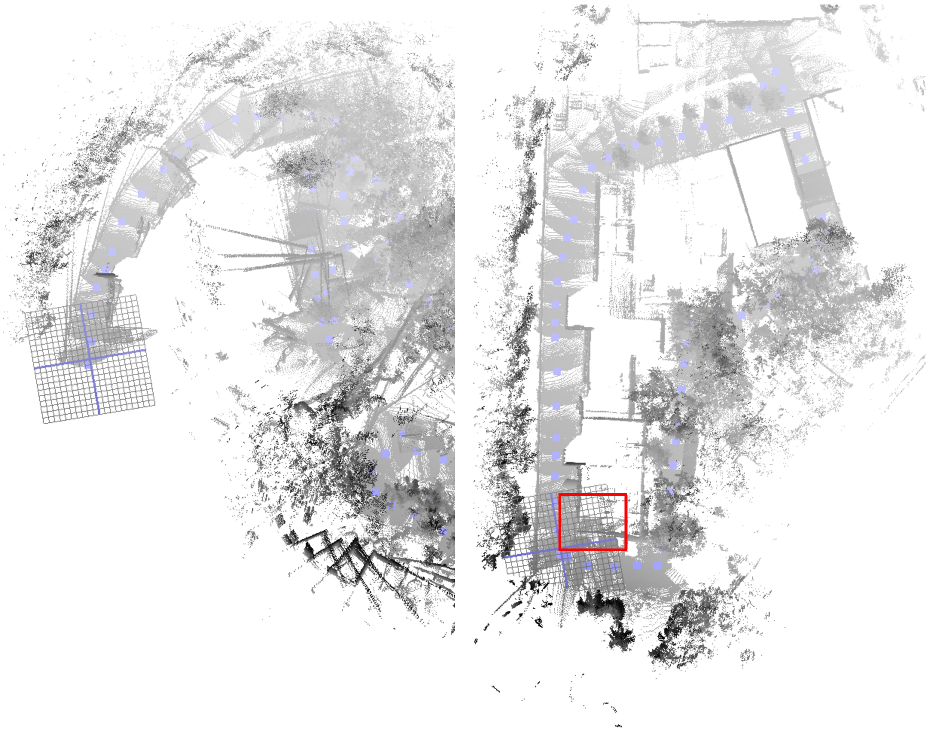

Fig. 3. 3D model of an experiment to digitize part of the campus of Schloss Birlinghoven campus (top view). Left: Registration based on odometry only. Right: Model based on incremental matching right before closing the loop, containing 62 scans each with approx. 100000 3D points. The grid at the bottom denotes an area of 20×20m^2 for scale comparison. The 3D scan poses are marked by blue points.

compares distances measured in the photo and in the 3D scene. The lines in the photo have been measured in pixels, whereas real distances, i.e., the (x, z)-values of the points, have been used in the point model. Taking into account that pixel distances in mid-resolution non-calibrated aerial image induce some error in ground truth, the correspondence show that the point model at least approximates reality quite well.

Table 1. Length ratio comparison of measured distances in the aerial photographs with distances in the point model as shown in Fig. 4

1st line	2nd line	ratio in aerial views	ratio in point model	deviation
AB	BC	0.683	0.662	3.1%
AB	BD	0.645	0.670	3.8%
AC	CD	1.131	1.141	0.9%
CD	BD	1.088	1.082	0.5%

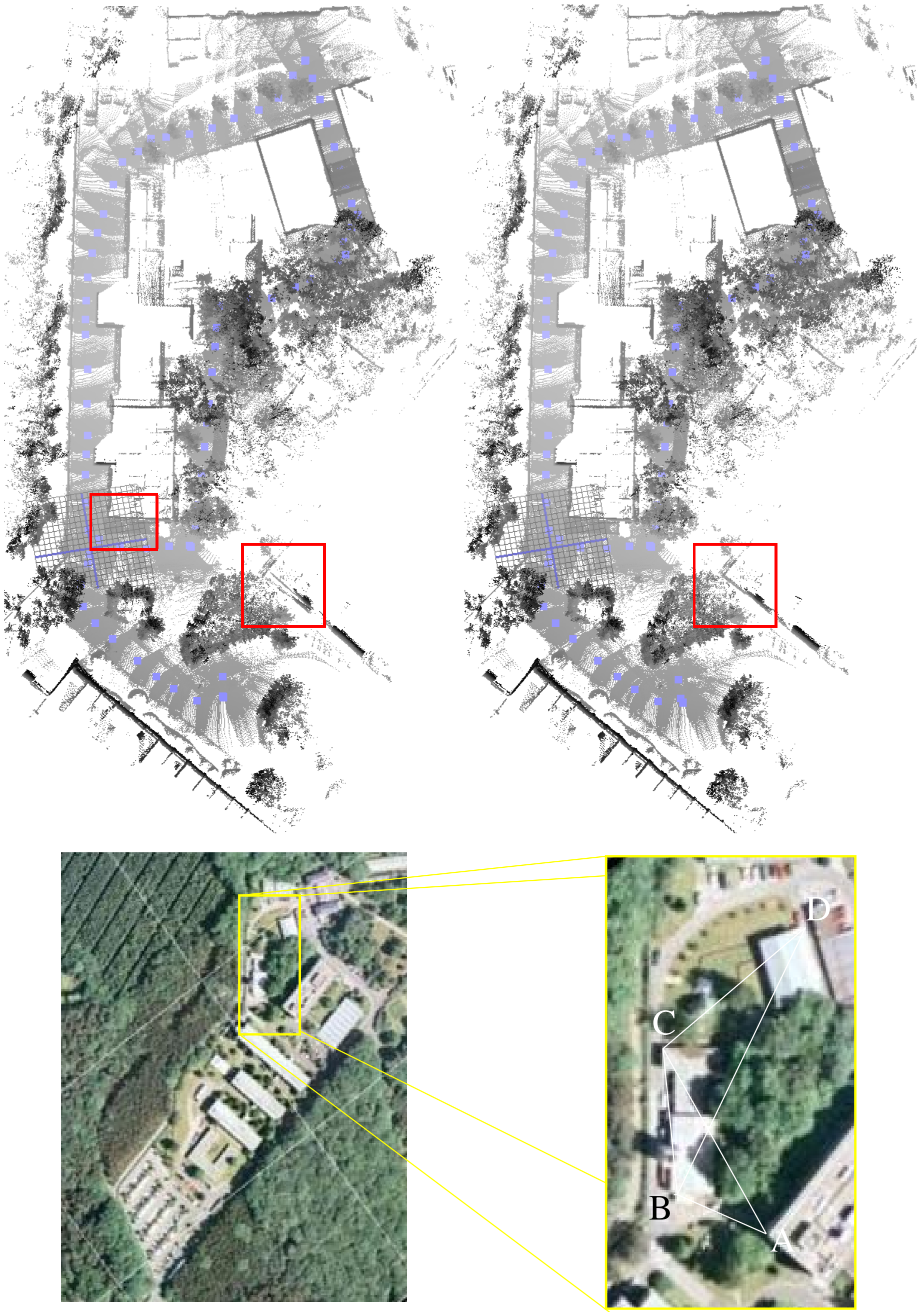

Fig. 4. Top left: Model with loop closing, but without global relaxation. Differences to Fig. 3 right and to the right image are marked. Top right: Final model of 77 scans with loop closing and global relaxation. Bottom: Aerial view of the scene. The points A – D are used as reference points in the comparison in Table 1.

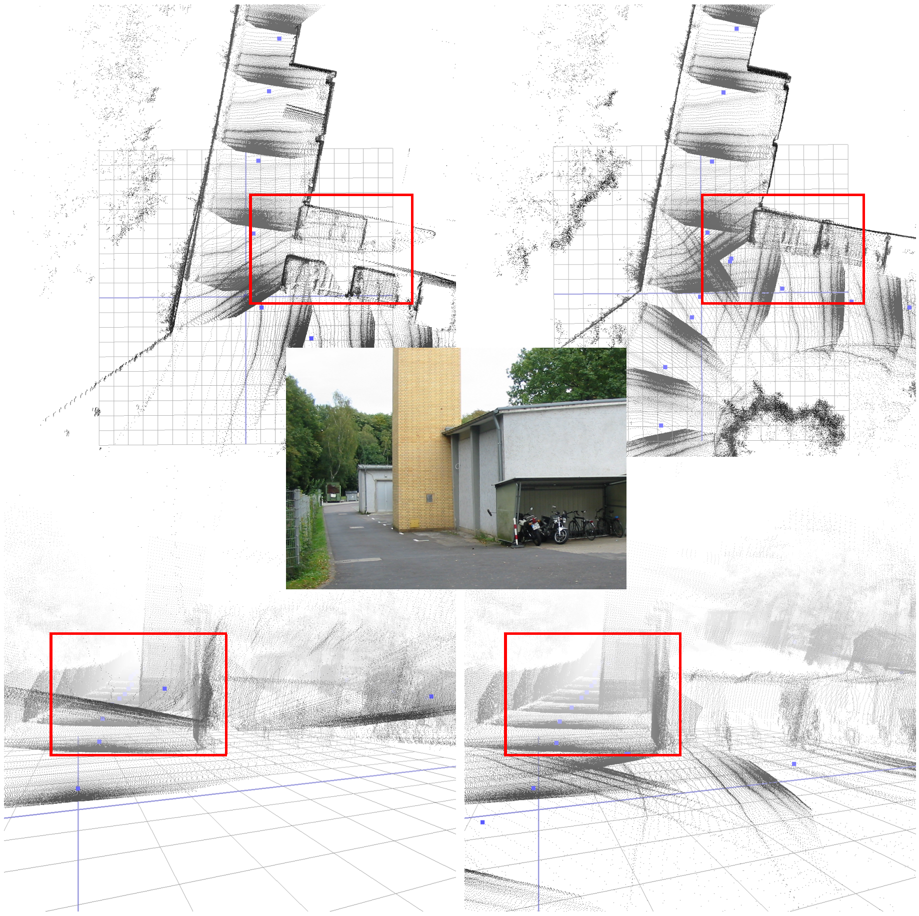

Fig. 5. Detailed view of the 3D model of Fig. 4. Left: Model before loop closing. Right: After loop closing, global relaxation and adding further 3D scans. Top: Top view. Bottom: Front view.

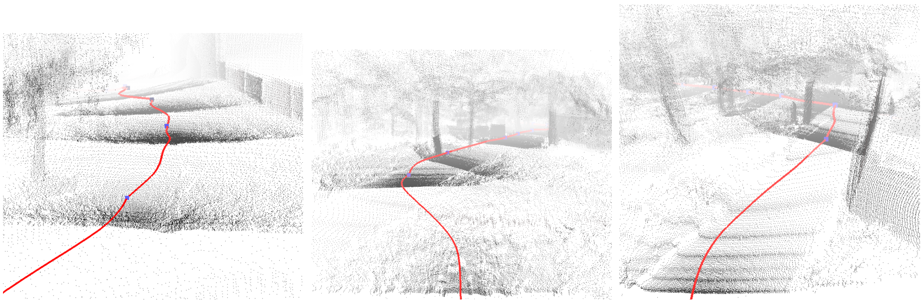

Fig. 6. Detailed views of the resulting 3D model corresponding to robot locations of Fig. 1

Mapping would fail without first calculatingheuristic initial estimations for ICP scan matching, since ICP would likely converge into an incorrect minimum. The resulting 3D map would be some mixture of Fig. 3 (left) and Fig. 4 (right).

Fig. 6 shows three views of the final model. These model views correspond to the locations of Kurt3D in Fig. 1. An updated robot trajectory has been plotted into the scene. Thereby, we assign every 3D scan that part of the trajectory which leads from the previous scan pose to the current one. Since scan matching did align the scans, the trajectory initially has gaps after the alignment (see Fig. 7).

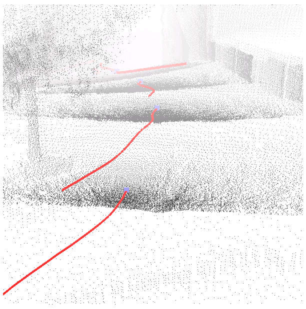

Fig. 7: The trajectory after mapping shows gaps, since the robot poses are corrected at 3D scan poses

We calculate the transformation $(\mathbf{R}, \boldsymbol{t})$ that maps the last pose of such a trajectory patch to the starting pose of the next patch. This transformation is then used to correct the trajectory patch by distributing the transformation as described in section 3.2. In this way the algorithm computes a continuous trajectory. An animation of the scanned area is available at `http://kos.informatik.uni-osnabrueck.de/6Doutdoor/`. The video shows the scene along the trajectory as viewed from about 1 m above Kurt3D's actual position.
The 3D scans were acquired within one hour by teleoperation of Kurt3D. Scan registration and closed loop detection took only about 10 minutes on a Pentium-IV-2800 MHz, while we did run the global relaxation for 2 hours. However, computing the flight-thru-animation took about 3 hours, rendering 9882 frames with OpenGL on consumer hardware.

In addition we used the 3D scan matching algorithm in the context of RoboCup Rescue 2004. We were able to produce online 3D maps, even though we did not use closed loop detection and global relaxation. Some results are available at `http://kos.informatik.uni-osnabrueck.de/download/Lisbon_RR/`.

5 Discussion and Conclusion

This paper has presented a solution to the SLAM problem considering six degrees of freedom and creating 3D maps of outdoor environments. It is based on ICP scan matching, initial pose estimation using a coarse-to-fine strategy with an octree representation and closing loop detection. Using an aerial photo as ground truth, the 3D map shows very good correspondence with the mapped environment, which was confirmed by a ratio comparison between map features and the respective photo features.

Compared with related approaches from the literature [6,10,26,27,28,29] we do not use a feature representation of the environment. Furthermore our algorithm manages registration without fixed data association. In the data association step, SLAM algorithms decide which features correspond. Wrong correspondences result in unprecise or even inconsistent models. The global scan

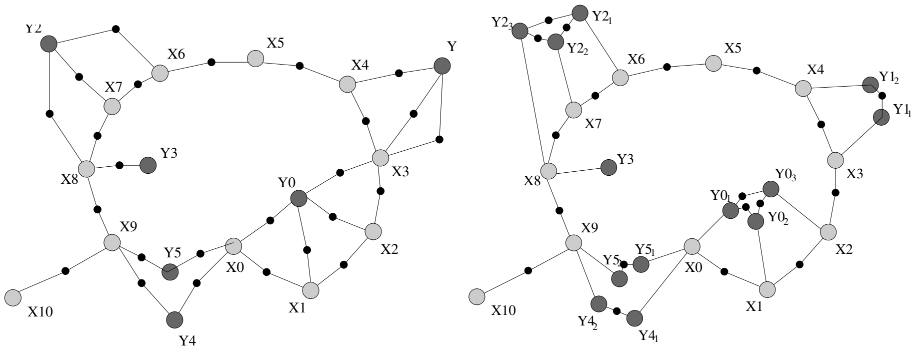

Fig. 8. Abstract comparison of SLAM approaches. Left: Probabilistic methods. The robot poses X_i as well as the positions of the associated landmarks Y_i are given in terms of a probability distribution. Global optimization tries to relax the model, where the landmarks are fixed. Small black dots on lines mark adjustable distances. Right: Our method with absolute measurements Y_i (note there are no black dots between scan poses and scanned landmarks). The poses X_i are adjusted based on scan matching aiming at collapsing the landmark copies Y_{i_k} for all landmarks Y_i. Data association is the search for closest points.

matching based relaxation computes corresponding points, i.e., closest points, in every iteration. Furthermore, we avoid using probabilistic representations to keep the computation time at a minimum. The model optimization is solved in a closed form, i.e., by direct pose transformation. As a result of these efforts, registration and closed loop detection of 77 scans each with ca. 100000 points took only about 10 minutes.

Fig. 8 compares the probabilistic SLAM approaches with ours on an abstract level as presented by Folkesson and Christensen [9]. Robot poses are labeled with X_i whereas the landmarks are the Y_i. Lines with black dots correspond to adjustable connections, e.g., springs, which can be relaxed by the algorithms. In our system, the measurements are fixed and data association is repeatedly done using nearest neighbor search.

Needless to say, a lot of work remains to be done. We plan to further improve the computation time and to use sensor uncertainty models. In addition, semantic labels for sub-structures of the resulting point model will be extracted.

References

1. P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. AVENUE: Automated Site Modelling in Urban Environments. In *Proc. 3DIM*, Canada, May 2001.
2. K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. *IEEE Transactions on PAMI*, 9(5):698 – 700, 1987.
3. R. Benjemaa and F. Schmitt. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In *Proc. 3DIM*, Ottawa, Canada, May 1997.
4. P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on PAMI*, 14(2):239 – 256, February 1992.

5. P. Biber, H. Andreasson, T. Duckett, and A. Schilling. 3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera. In *Proc. IROS*, Sendai, Japan, September 2004.
6. M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229 – 241, 2001.
7. 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.
8. Matrix FAQ. Version 2, `http://skal.planet-d.net/demo/matrixfaq.htm`, 1997.
9. J. Folkesson and H. Christensen. Outdoor Exploration and SLAM using a Compressed Filter. In *Proc. ICRA*, pages 419–426, Taipei, Taiwan, September 2003.
10. U. Frese and G. Hirzinger. Simultaneous Localization and Mapping – A Discussion. In *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2001.
11. C. Früh and A. Zakhor. 3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans. In *Proc. CVPR*, Hawaii, USA, December 2001.
12. A. Georgiev and P. K. Allen. Localization Methods for a Mobile Robot in Urban Environments. *IEEE Transaction on RA*, 20(5):851 – 864, October 2004.
13. M. Golfarelli, D. Maio, and S. Rizzi. Correction of dead-reckoning errors in map building for mobile robots. *IEEE Transaction on TRA*, 17(1), May 2001.
14. 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.
15. P. Kohlhepp, M. Walther, and P. Steinhaus. Schritthaltende 3D-Kartierung und Lokalisierung für mobile Inspektionsroboter. In *18. Fachgespräche, AMS*, 2003.
16. K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann. High-Speed Laser Localization for Mobile Robots. *J. Robotics and Aut. Syst.*, (accepted), 2005.
17. A. Lorusso, D. Eggert, and R. Fisher. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. In *Proc. BMVC*, England, 1995.
18. F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333 – 349, October 1997.
19. A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an Application in autonomous mine mapping. In *Proc. ICRA*, 2004.
20. K. Pulli. Multiview Registration for Large Data Sets. In *Proc. 3DIM*, 1999.
21. S. Se, D. Lowe, and J. Little. Local and Global Localization for Mobile Robots using Visual Landmarks. In *Proc. IROS*, Hawaii, USA, October 2001.
22. V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan–views. In *Proc. of SPIE, Electronic Imaging, 11th Annual Symposium*, San Jose, CA, USA, January 1999.
23. H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor en vironments. *Journal Robotics and Autonomous Systems*, 45(3 – 4):181 – 198, December 2003.
24. H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6D SLAM A Preliminary Report on Closing the Loop in Six Dimensions. In *Proc. IAV*, 2004.
25. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
26. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium.* Morgan Kaufmann, 2002.
27. S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Auton. Rob.*, 1997.

28. S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proc. ICRA*, 2000.
29. S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. F. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Machine Learning and Autonomous Robots*, 23(7 – 8):693 – 716, July/August 2004.
30. O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proc. ICRA*, 2004.
31. H. Zhao and R. Shibasaki. Reconstructing Textured CAD Model of Urban Environment Using Vehicle-Borne Laser Range Scanners and Line Cameras. In *Proc. ICVS*, pages 284 – 295, Vancouver, Canada, July 2001.