6D SLAM — PRELIMINARY REPORT ON CLOSING THE LOOP IN SIX DIMENSIONS

Hartmut Surmann   Andreas Nüchter
Kai Lingemann   Joachim Hertzberg

Fraunhofer Institute for Autonomous Intelligent Systems
Schloss Birlinghoven
D-53754 Sankt Augustin, Germany

Abstract: To create 3D volumetric maps of scenes with autonomous mobile robots it is necessary to gage several 3D scans and to merge them into one consistent 3D model. This paper provides a new solution to the simultaneous localization and mapping (SLAM) problem with six degrees of freedom. Robot motion on natural surfaces has to cope with yaw, pitch and roll angles, turning pose estimation into a problem in six mathematical dimensions. A fast variant of the Iterative Closest Points (ICP) algorithm registers the 3D scans in a common coordinate system and relocalizes the robot. Finally, consistent 3D maps are generated using closing loop detection and a global relaxation.

Keywords: autonomous mobile robots, 3D laser scanner, 3D scan matching, simultaneous localization and mapping (SLAM), closing loop.

1. INTRODUCTION

The problem of automatic environment sensing and modeling in 3D is complex, because a number of fundamental scientific issues are involved. One is the control of an autonomous mobile robot and scanning the environment with a 3D sensor. Another question is how to create a volumetric consistent scene in a common coordinate system from multiple views. This problem is addressed here: The proposed algorithms allows to digitize large environments fast and reliably without any intervention and solve the simultaneous localization and mapping (SLAM) problem. Finally, robot motion on natural outdoor surfaces has to cope with yaw, pitch and roll angles, turning pose estimation as well as scan matching or registration into a problem in six mathematical dimensions. This paper presents a new solution to the SLAM problem with six degrees of freedom and focuses on closing the loops in 6D. A fast variant of the iterative closest points algorithm registers the 3D scans in a common coordinate system and relocalizes the robot. The computational requirements are reduced by two new methods: First we reduce the 3D data and second we approximate the closest point search. Overall consistent 3D maps are generated using a global relaxation after a closed loop has been detected.

The paper is organized as follows: The next section describes the state of the art. Section 3 presents the autonomous mobile robot and the 3D laser range finder. Section 4 presents the registration algorithms, a ground truth experiments and solution to the SLAM problem. Furthermore we show how data reduction and access methods speed up computation and the methods become feasible. The next section presents the results. Finally section 7 concludes the paper. The paper is accompanied by a video including the 3D map. The video is available for download at `www.ais.fraunhofer.de/ARC/3D/6D/`.

2. RELATED WORK

Some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. In (Thrun *et al.*, 2000) two 2D laser range finders are used for acquiring 3D data. One laser scanner is mounted horizontally and one is mounted vertically. The latter one grabs a vertical scan line which is transformed into 3D points using the current robot pose. This approach has difficulties to navigate around 3D obstacles with jutting out edges. They are only detected while passing them. The published 2D probabilistic localization approaches, e.g., Markov models or Monte Carlo methods work well in flat and structured 2D environments but an extension in the third dimension is still missing since the algorithm do not scale with additional dimensions.

A few other groups use 3D laser scanners (Sequeira *et al.*, 1999; Allen *et al.*, 2001). 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 telepresence (Sequeira *et al.*, 1999). They used a RIEGL laser range finder on a robot, and the ICP algorithm for scan matching. The AVENUE project develops a robot for modeling urban environments using a CYRAX laser scanner and a feature-based scan matching approach for registration of the 3D scans in a common coordinate system (Allen *et al.*, 2001). The research group of M. Hebert has reconstructed environments using the Zoller+Fröhlich laser scanner and aims to build 3D models without initial position estimates, like without odometry information (Hebert *et al.*, 2001).

3. AUTOMATIC 3D SCANNING

3.1 The AIS 3D Laser Range Finder

The AIS 3D laser range finder (Surmann *et al.*, 2001) 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 (Fig. 1) (Surmann *et al.*, 2001).

The area of $180°$(h) $\times$ $120°$(v) is scanned with different horizontal (181, 361, 721) and vertical (128, 256, 512) 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.

Fig. 1. The robot platform Kurt3D with the 3D scanner.

3.2 The Autonomous Mobile Robot Kurt3D

Kurt3D (figure 1) is a mobile robot platform with a size of 45 cm (length) $\times$ 33 cm (width) $\times$ 26 cm (height) and a weight of 15.6 kg. Equipped with the 3D laser range finder the height increases to 47 cm and weight increases to 22.6 kg. Kurt3D's maximum velocity is 5.2 m/s (autonomously controlled 4.0 m/s). Two 90W motors are used to power the 6 wheels, whereas the front and rear wheels have no tread pattern to enhance rotating. Kurt3D operates for about 4 hours with one battery (28 NiMH cells, capacity: 4500 mAh) charge. The core of the robot is a Pentium-III-600 MHz with 384 MB RAM running Linux. An embedded 16-Bit CMOS microcontroller is used to control the motor.

4. 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 localization of the robot with the 3D scanner were precise, the registration could be done directly by the robot pose. However, due to the unprecise robot sensors, the self localization is erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration.

The matching of 3D scans can either operate on the whole three-dimensional scan point set or can be reduced to the problem of scan matching in 2D by extracting, e.g., a horizontal plane of fixed height from both scans, merging these 2D scans and applying the resulting translation and rotation matrix to all points of the corresponding 3D scan.

Matching of complete 3D scans has the advantage of having a larger set of attributes (either pure data points or extracted features) to compare the scans. This results in higher precision and lowers the possibility of running into a local minimum of the cost function. Furthermore, using three dimensions enables the robot control software to recognize and take into account changes of height and roll, yaw and pitch angles of the robot. This 6D robot relocalization is essential for robots driving cross country.

4.1 Matching as an Optimization Problem

The following method for registration of point sets is part of many publications, so only a short summary is given here. The complete algorithm was invented in 1992 and can be found, e.g., in (Besl and McKay, 1992). The method is called *Iterative Closest Points (ICP) algorithm.*

Given two independently acquired sets of 3D points, M (model set, $|M| = N_m$) and D (data set, $|D| = N_d$) which correspond to a single shape, we want to find the transformation consisting of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ which minimizes the following cost function:

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

$w_{i,j}$ 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 ($\mathbf{R}$, $\mathbf{t}$) that minimize $E(\mathbf{R}, \mathbf{t})$ on the base of the corresponding points.

The ICP algorithm calculates iteratively a local minimum of equation (1). In each iteration step, the algorithm selects the closest points as correspondences $w_{i,j}$ and calculates the transformation $(\mathbf{R}, \mathbf{t})$ for minimizing equation (1). Fig. 2 shows three steps of the ICP algorithm. Besl et al. proves that the method terminates in a minimum (Besl and McKay, 1992). The assumption is that in the last iteration step the point correspondences are correct.

In each ICP iteration, the transformation is calculated by the quaternion based method of Horn (Horn, 1987). A unit quaternion is a 4 vector $\dot{q} = (q_0, q_x, q_y, q_z)^T$, where $q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1, q_0 \geq 0$. It describes a rotation axis and an angle to rotate around that axis. The rotation expressed as quaternion that minimizes equation (1) is the largest eigenvalue of the cross-covariance matrix (Besl and McKay, 1992). This calculation is computationally inexpensive. The transformation $\mathbf{t}$ is computed using the rotation and the two centroids of the 3D points sets (Horn, 1987). Fig. 2 shows two 3D scans in their initial, i.e., odometry-based pose, after 5 iterations, and the final pose. 55 iterations are needed to align these two 3D scans correctly.

In order to use the ICP algorithm for mobile robot applications an analysis of the registration process has been made. To compare the registration with the ground truth several experiments have to be made. We have made an experiment in a flat, very structured environment (Fig. 2) and measured the ground truth with a meter rule (scan pose 1: (x, z, θ) = (0 cm, 0 cm, 0°), scan pose 2 (120 cm, 301 cm, 0°). It turned out, that the automatic alignment of the 3D scans (116 cm, 301 cm, 0°) is close to that ground truth pose. Fig. 3 shows the results of the alignment of two 3D scans with different starting guesses. It turned out that the angle of the robot is more important than the position. Nevertheless, the main parameter of 3D scan matching is the form of the 3D surface.

4.2 Matching Multiple 3D Scans

To digitalize environments, 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, in (Chen and Medoni, 1991) 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 many scans leads to inconsistent scenes and to problems with the robot localization.

4.2.1. Closing the Loop. After matching multiple 3D scans, errors have accumulated and a closed loop will be inconsistent (Fig. 4). Our algorithm detects a closing loop by registering the last acquired 3D scan with earlier acquired scans, e.g., the first scan. If a registration is possible, the computed error is in a first step divided by the number of 3D scans in the loop and distributed over all scans (Fig. 4). A second step minimizes the global error with the following algorithm.

4.2.2. Diffusing the Error. Pulli presents a registration method that minimizes the global error and avoids inconsistent scenes (Pulli, 1999). This method distributes the global error while the registration of one scan is followed by registration of

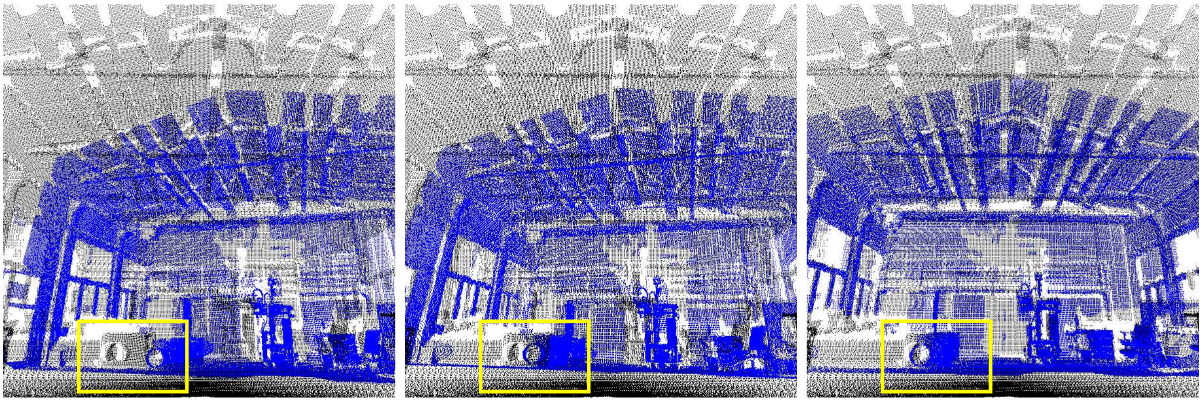

Fig. 2. Left: Initial odometry based pose of two 3D scans. Middle: Pose after five ICP iterations. Right: final alignment.

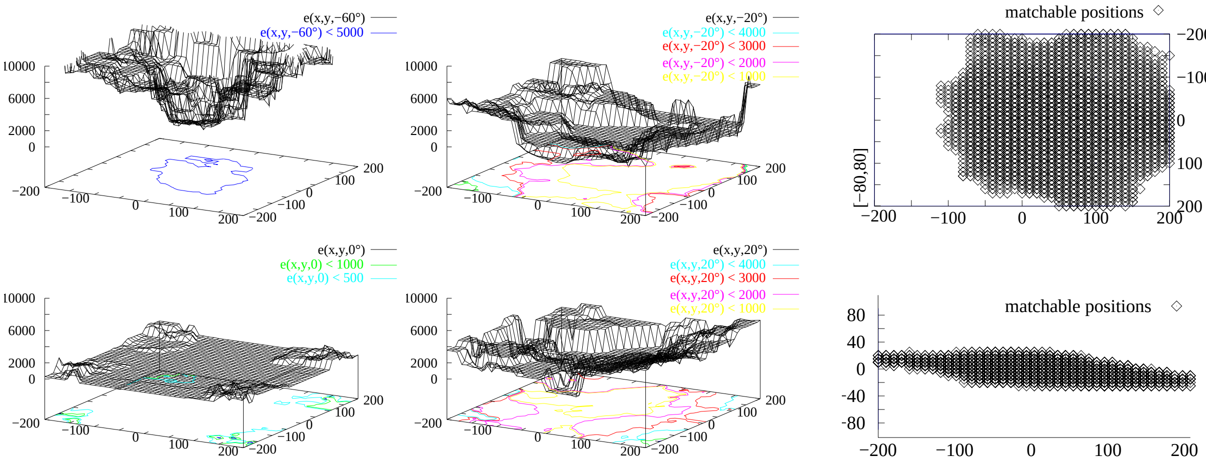

Fig. 3. Left and middle: Error measurements for 3D scan registration (Fig. 2) against the ground truth pose (120 cm, 301 cm, -45°). The Error function is $e(x, z, \theta) = ||(x, z) - (120, 301)|| + \gamma \left|\left|\theta - \frac{\pi}{4}\right|\right|$. Results with $\theta \in \{-60, -20, 0, 20\}$ are given. Right: The poses are marked in (x, z, θ) from which a correct alignment of two 3D scans is possible.

all neighboring scans. Other matching approaches with global error minimization have been published, e.g., (Benjemaa and Schmitt, 1997) and (Eggert *et al.*, 1998).

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 three steps register all scans and minimize the global error, after a queue is initialized with the first scan of the closed loop:

(1) The current scan is the first scan of the queue. This scan is removed from the queue.
(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 if more than 250 corresponding point pairs exist.
(3) If the current scan changes its location by applying the transformation (translation or rotation), then each single scan of the set of neighbors that is not in the queue is added to the end of the queue.

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 (Pulli, 1999). 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 (Fig. 4).

4.3 Data Reduction and Access

The computational expense of the ICP algorithm mainly depends on the number of points. In a brute force implementation the point pairing is in $O(n^2)$. Data reduction reduces the time required for matching. The reduction proposed here considers the procedure of the scanning process, i.e., the spherical and continuous measurement of the laser. Scanning is noisy and small errors may occur. Two kinds of errors mainly occur: Gaussian noise and so called salt and pepper noise. The latter one occurs for example at edges, where the laser beam of the scanner hits two surfaces, resulting in a mean and erroneous data value. Furthermore reflections lead to suspicious data. Without filtering, only a few outliers lead to mul-

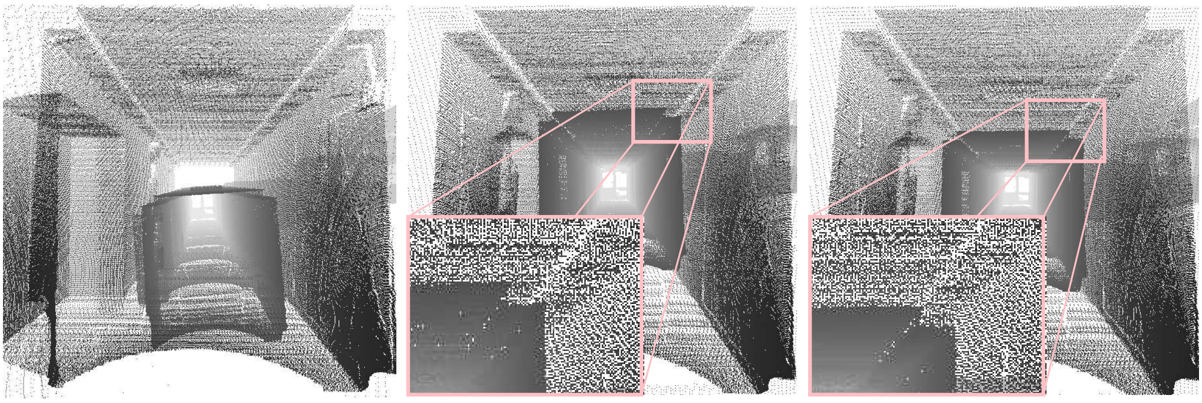

Fig. 4. Left: Initial poses of 3D scans when closing the loop. Middle: Poses after detecting the loop and equally sharing the resulting error. Right: Final alignment after error diffusion with correct alignment of the edge structure at the ceiling.

tiple wrong point pairs during the matching phase and results in an incorrect 3D scan alignment.

We propose a fast filtering method to reduce and smooth the data for the ICP algorithm. The filter is applied to each 2D scan slice, containing up to 721 data points. It is a combination of a median and a reduction filter. The median filter removes the outliers by replacing a data point with the median value of the n surrounding points (here: $n = 7$). The neighbor points are determined according to their number within the 2D scan slice, since the laser scanner provides the data sorted in a counter-clockwise direction. The median value is calculated with regards to the Euclidian distance of the data points to the point of origin. The data reduction works as follows: The scanner emits the laser beams in a spherical way, such that the data points close to the source are more dense. Multiple data points located close together are joined into one point. This reduction lowers the Gaussian noise. Finally, the data for the scan matching is collected from every third scan slice. This fast vertical reduction yields a good surface description. The number of these so called *reduced points* is one order of magnitude smaller than the original one. Data reduction and filtering are online algorithms and run in parallel to the 3D scanning.

The ICP algorithms spends most of its time during creating of the point pairs. kD-trees (here $k = 3$) have been suggested for speed up the data access (Simon *et al.*, 1994; Zhang, 1992). The kd-trees are a binary tree data structure with terminal buckets. The data is stored in the buckets and the keys are selected, such that a data space is divided into two equal parts. This ensures that a data point can be selected in $O(\log n)$. In addition the time spent on creating the tree is important.

Recently Approximate kd-trees (Apr-kd-tree) have been introduced (Greenspan and Yurick, 2003). The idea behind this is to return as an approximate nearest neighbor $\mathbf{p}_a$ the closest point $\mathbf{p}_b$

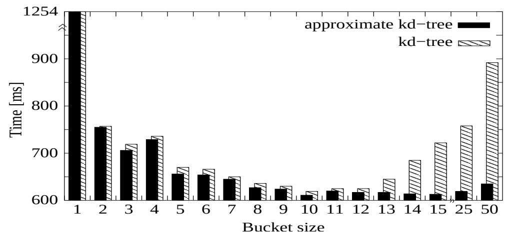

Fig. 5. Running time of scan registration using kd-tree search and approximate kd-tree search.

in the bucket region of the kD-tree where $\mathbf{p}$ lies. This value is determined from the depth-first search, thus expensive Ball-Within-Bounds tests and backtracking are not used (Greenspan and Yurick, 2003). In addition to these ideas we avoid the linear search within the bucket. During the computation of the Apr-kd-tree the mean values of the points within a bucket are computed and stored. Then the mean value of the bucket is used as approximate nearest neighbor, replacing the linear search.

The search using the Apr-kd-tree is applied until the error function (1) stops decreasing. To avoid misalignments due to the approximation, the registration algorithm is restarted using the normal kd-tree search. A few iterations (usually 1-3) are needed for this final corrections. Figure 5 shows the registration time of two 3D scans in dependence to the bucket size using kd-tree or Apr-kd-tree search. Both search methods have their best performance with a bucket size of 10 points.

5. RESULTS

To visualize the scanned 3D data, a viewer program based on OpenGL has been implemented. The task of this program is to project a 3D scene to the image plane, i.e., the monitor, such that the data can be drawn and inspected from every perspective. Fig. 2, 4 and 6 show rendered 3D scans.

Table 1. Computing time and number of ICP iterations to align all 32 3D scans (Pentium-IV-2400). In addition the computing time for the SLAM algorithm (closed loop detection and simultaneous matching) is given. About 1 min per 3D scan is needed to do error diffusion in 6D.

points & search method	time	iter.
all pts. & brute force	144 h 5 min	2080
all pts. & kD–tree	12 min 23 s	2080
all pts. & Apx-kD–tree	10 min 1 s	2080
red. pts. & Apx-kD–tree	<1 min 32 s	2176
6D SLAM with *reduced pts.* & Apx-kD–tree	38 min	16000

The proposed algorithms have been applied to a data set acquired on Birlinghoven Castle's institute campus. 32 3D scans, each containing 302820 range data points, were taken by the mobile robot Kurt3D. The robot had to cope with a height difference between the two buildings of 1.05 meter, covered, on the one hand, by a sloped driveway in open outdoor terrain, and, on the other hand, by a ramp of 12° inside the building. The 3D model was computed after acquiring all 3D scans. Table 1 summarizes the computational time of our 6D SLAM algorithms. The final model with the closed loop is presented in Fig. 6. Please refer to the website `http://www.ais.fraunhofer.de/ARC/3D/6D/` for a computed animation and video through the scanned 3D scene.

6. CONCLUSION

This paper has presented a new solution to the simultaneous localization and mapping (SLAM) problem with six degrees of freedom. Based on the ICP algorithm the registration error is globally spread over all 3D scans and thus minimized. The presented algorithms are speeded up with data reduction that maintains the surface structure and with approximate kd-trees for closest point search. The computational amount of about 1 min per scan is acceptable, but further improvement is needed.

Future work will concentrate on the integration of two color cameras (Fig. 1) to acquire textured 3D models. The aperture angle of the camera will be enlarged using a pan and tilt unit to acquire color information for all measured range points.

Acknowledgment: Special thanks to Kai Pervölz and Matthias Hennig for supporting our work.

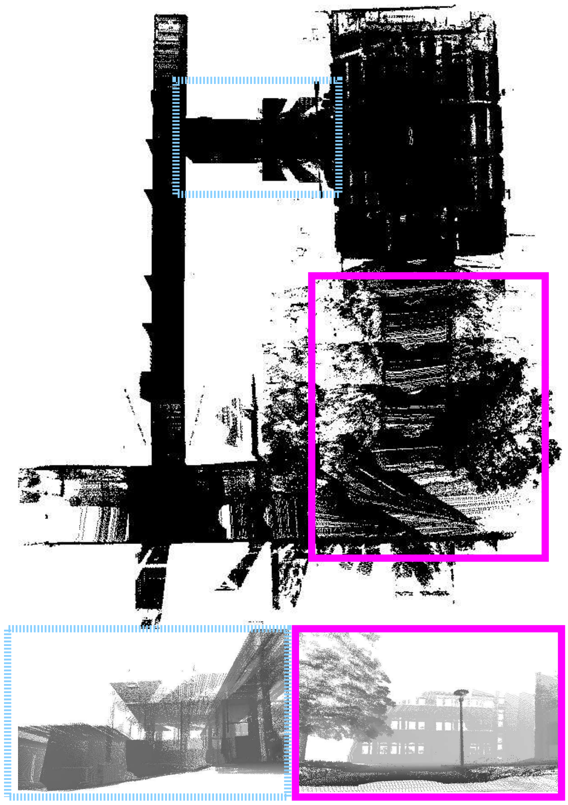

Fig. 6. The closed loop with a viewing position 40 m above. Bottom left: A ramp connecting two buildings is correctly modeled (height difference 1.05 m). Bottom right: Outdoor environment modeling.

REFERENCES

Allen, P., I. Stamos, A. Gueorguiev, E. Gold and P. Blaer (2001). AVENUE: Automated Site Modeling in Urban Environments. In: *Proc. 3rd 3DIM.* Canada.

Benjemaa, R. and F. Schmitt (1997). Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In: *Proc. 1st 3DIM.* Canada.

Besl, P. and N. McKay (1992). A method for Registration of 3–D Shapes. *IEEE Transactions on PAMI* **14**(2), 239 – 256.

Chen, Y. and G. Medoni (1991). Object Modelling by Registration of Multiple Range Images. In: *Proc. IEEE ICRA '91.* USA.

Eggert, D., A. Fitzgibbon and R. Fisher (1998). Simultaneous Registration of Multiple Range Views Satisfying Global Consistency Constraints for Use In Reverse Engineering. *CVIU* **69**, 253 – 272.

Greenspan, M. and M. Yurick (2003). Approximate K-D Tree Search for Efficient ICP. In: *Proc. 4th 3DIM.* Canada.

Hebert, M., M. Deans, D. Huber, B. Nabbe and N. Vandapel (2001). Progress in 3–D Mapping and Localization. In: *Proc. 9th SIRS '01.* France.

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

Pulli, K. (1999). Multiview Registration for Large Data Sets. In: *Proc. of the 2nd 3DIM '99.* Canada.

Sequeira, V., K. Ng, E. Wolfart, J. Goncalves and D. Hogg (1999). Automated 3D reconstruction of interiors with multiple scan–views. In: *Proceedings of SPIE, Electronic Imaging.* USA.

Simon, D., M. Hebert and T. Kanade (1994). Real–time 3–D pose estimation using a high–speed range sensor. In: *Proc. IEEE ICRA '94.* USA.

Surmann, H., K. Lingemann, A. Nüchter and J. Hertzberg (2001). Fast acquiring and analysis of three dimensional laser range data. In: *Proc. VMV '01.* Germany.

Thrun, S., D. Fox and W. Burgard (2000). A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In: *Proc. IEEE ICRA '00.* USA.

Zhang, Z. (1992). Iterative point matching for registration of free–form curves. Technical Report RR-1658. INRIA–Sophia Antipolis. France.