3D Mapping with Semantic Knowledge

Andreas Nüchter[1], Oliver Wulf[2], Kai Lingemann[1], Joachim Hertzberg[1],
Bernado Wagner[2], and Hartmut Surmann[3]

[1] University of Osnabrück, Institute for Computer Science,
Knowledge-Based Systems Research Group, Albrechtstraße 28,
D-49069 Osnabrück, Germany
nuechter@informatik.uni-osnabrueck.de
WWW home page: http://www.informatik.uni-osnabrueck.de/nuechter/
[2] University of Hannover, Institute for Systems Engineering (ISE/RTS),
Appelstraße 9A, D-30167 Hannover, Germany
[3] Fraunhofer Institute for Autonomous Intelligent Systems (AIS),
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany

Abstract. A basic task of rescue robot systems is mapping of the environment. Localizing injured persons, guiding rescue workers and excavation equipment requires a precise 3D map of the environment. This paper presents a new 3D laser range finder and novel scan matching method for the robot Kurt3D [9]. Compared to previous machinery [12], the apex angle is enlarged to 360°. The matching is based on semantic information. Surface attributes are extracted and incorporated in a forest of search trees in order to associate the data, i.e., to establish correspondences. The new approach results in advances in speed and reliability.

1 Introduction

Rescue robotic systems are designed to assist rescue workers in earthquake, fire, flooded, explosive and chemical disaster areas. Currently, many robots are manufactured, but most of them lack a reliable mapping method. Nevertheless, a fundamental task of rescue is to localize injured persons and to map the environment. To solve these tasks satisfactorily, the generated map of the disaster environment has to be three-dimensional. Solving the problem of simultaneous localization and mapping (SLAM) for 3D maps turns the localization into a problem with six degrees of freedom. The x, y and z positions and the roll, yaw and pitch orientations of the robot have to be considered. We are calling the resulting SLAM variant 6D SLAM [10].

This paper addresses the problem of creating a consistent 3D scene in a common coordinate system from multiple views. The proposed algorithms allow to digitize large environments fast and reliably without any intervention and solve the 6D SLAM problem. A 360° 3D laser scanner acquires data of the environment and interprets the 3D points online. A fast variant of the iterative closest points (ICP) algorithm [3] registers the 3D scans in a common coordinate system and relocalizes the robot. The registration uses a forest of approximate

kd-trees. The resulting approach is highly reliable and fast, such that it can be applied online to exploration and mapping in RoboCup Rescue.

The paper is organized as follows: The remainder of this section describes the state of the art in automatic 3D mapping and presents the autonomous mobile robot and the used 3D scanner. Section 2 describes briefly the online extraction of semantic knowledge of the environment, followed by a discussion of the scan matching using forests of trees (section 3). Section 4 presents experiments and results and concludes.

1.1 3D Mapping – State of the Art

A few groups use 3D laser scanners [1,5,11,14,15]. The RESOLV project aimed to model interiors for virtual reality and tele presence [11]. They used a RIEGL laser range finder on robots and the ICP algorithm for scan matching [3]. The AVENUE project develops a robot for modeling urban environments [1], using an expensive CYRAX laser scanner and a feature-based scan matching approach for registration of the 3D scans in a common coordinate system. Nevertheless, in their recent work they do not use data of the laser scanner in the robot control architecture for localization [5]. Triebel et al uses a SICK scanner on a 4 DOF robotic arm mounted on a B21r platform to explore the environment [14].

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 [7,8,13,15]. Thrun et al. [7,13] use two 2D laser range finder 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. 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. Howard et al. uses the restriction of flat ground and structured environments [8]. Wulf et al. let the scanner rotate around the vertical axis. They acquire 3D data while moving, thus the quality of the resulting map crucial depends on the pose estimate that is given by inertial sensors, i.e., gyros [15]. In this paper we let rotate the scanner continuously around its vertical axis, but accomplish the 3D mapping in a stop-scan-go fashion, therefore acquiring consistent 3D scans as well.

Other approaches use information of CCD-cameras that provide a view of the robot's environment. Some groups try to solve 3D modeling by using a planar SLAM methods and cameras, e.g., in [4].

1.2 Automatic 3D Sensing

The Robot Platform Kurt3D. Kurt3D (Fig. 1) is a mobile robot platform with a size of 45 cm (length) × 33 cm (width) × 26 cm (height) and a weight of 15.6 kg, both indoor as well as outdoor models exist. Two 90 W motors (short-term 200 W) are used to power the 6 wheels. Compared to the original Kurt3D robot platform, the outdoor version has larger wheels, where the middle ones are shifted outwards. Front and rear wheels have no tread pattern to enhance rotating. Kurt3D operates for about 4 hours with one battery charge (28 NiMH cells, capacity: 4500 mAh) charge. The core of the robot is a laptop computer running a Linux operating system. An embedded 16-Bit CMOS microcontroller is used to process commands to the motor. A CAN interface connects the laptop with the microcontroller.

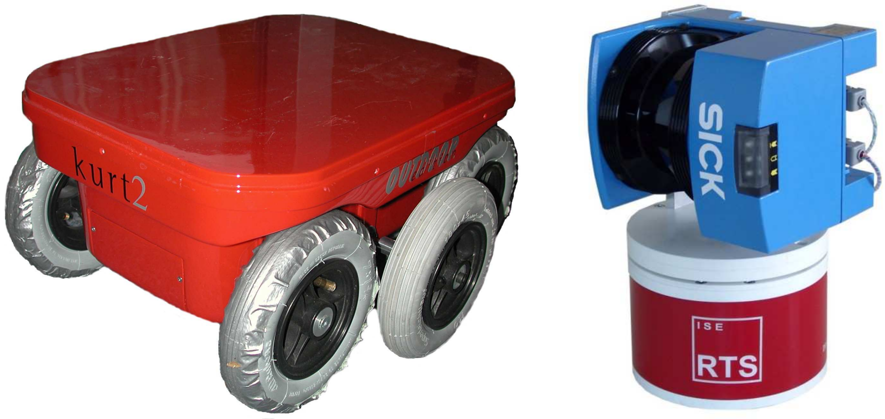

Fig. 1. The mobile robot platform Kurt3D offroad (left) and the 3D laser scanner (right) The scanner rotates around the vertical axis. It's technical basis is a SICK 2D laser range finder (LMS-200).

The 3D Laser Scanner. As there is no commercial 3D laser range finder available that could be used for mobile robots, it is common practice to assemble 3D sensors out of a standard 2D scanner and an additional servo drive [6,12]. The scanner that is used for this experiment is based on a SICK LMS 291 in combination with the RTS/ScanDrive developed at the University of Hannover. Different orientations of the 2D scanner in combination with different turning axes result in a number of possible scanning patterns. The scanning pattern that is most suitable for this rescue application is the yawing scan with a vertical 2D raw scan and rotation around the upright axis (see Fig. 1). The yawing scan pattern results in the maximal possible field of view (360° horizontal and 180° vertical) and an uniform distribution of scan points.

As 3D laser scanner for autonomous search and rescue applications needs fast and accurate data acquisition in combination with low power consumption, the RTS/ScanDrive incorporates a number of improvements. One mechanical improvement is the ability to turn continuously, which is implemented by using slip rings for power and data connection to the 2D scanner. This leads to a homogeneous distribution of scan points and saves the energy and time that is needed for acceleration and deceleration of panning scanners. Another improvement that becomes more important with short scanning times of a few seconds is the compensation of systematic measurement errors. In this case the compensation is done by sensor analysis and hard real-time synchronization, using a

Linux/RTAI operation system. These optimizations lead to scan times as short as 3.2s for a yawing scan with 1.5° horizontal and 1° vertical resolution (240x181 points). For details on the RTS/ScanDrive see [17].

2 Extracting Semantic Information

The basic idea of labelling 3D points with semantic information is to use the gradient between neighbouring points to differ between three categories, i.e., floor-, object- and ceiling-points. A 3D point cloud that is scanned in a yawing scan configuration, can be described as a set of points $\mathbf{p}_{i,j} = (\phi_i, r_{i,j}, z_{i,j})^T$ given in a cylindrical coordinate system, with i the index of a vertical raw scan and j the point index within one vertical raw scan counting bottom up. The gradient $\alpha_{i,j}$ is calculated by the following equation:

$$\tan \alpha_{i,j} = \frac{z_{i,j} - z_{i,j-1}}{r_{i,j} - r_{i,j-1}} \quad \text{with} \quad -\frac{1}{2}\pi \leq \alpha_{i,j} < \frac{3}{2}\pi.$$

The classification of point $\mathbf{p}_{i,j}$ is directly derived from the gradient $\alpha_{i,j}$:

1. floor-points: $\alpha_{i,j} < \tau$
2. object-points: $\tau \leq \alpha_{i,j} \leq \pi - \tau$
3. ceiling-points: $\pi - \tau < \alpha_{i,j}$

with a constant τ that depends on the maximal ascent being accessible by the robot (here: $\tau = 20°$).

Applied to real data, this simple definition causes two problems. As can be seen in Fig. 2(a) noisy range data can lead to wrong classifications of floor- and ceiling-points. Changing the differential quotient as follows solves this problem:

$$\tan \alpha_{i,j} = \frac{z_{i,j} - z_{i,j-k}}{r_{i,j} - r_{i,j-k}}$$

with $k \in \mathbb{N}, k \geq 1$ the smallest number so that

$$\sqrt{(r_{i,j} - r_{i,j-k})^2 + (z_{i,j} - z_{i,j-k})^2} > d_{\min}$$

for a constant $d_{\min}$ depending on the scanner's depth accuracy σ (here: $\sigma =$ 30 mm, $d_{\min} = 2\sigma$).

The second difficulty is the correct computation of the gradient across jumping edges (see Fig. 2(b)). This problem is solved with a prior segmentation [16], as the gradient $\alpha_{i,j}$ is only calculated correctly if both points $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,j-k}$ belong to the same segment. The correct classification result can be seen in Fig. 2(c). Fig. 3 shows a 3D scan with the semantic labels.

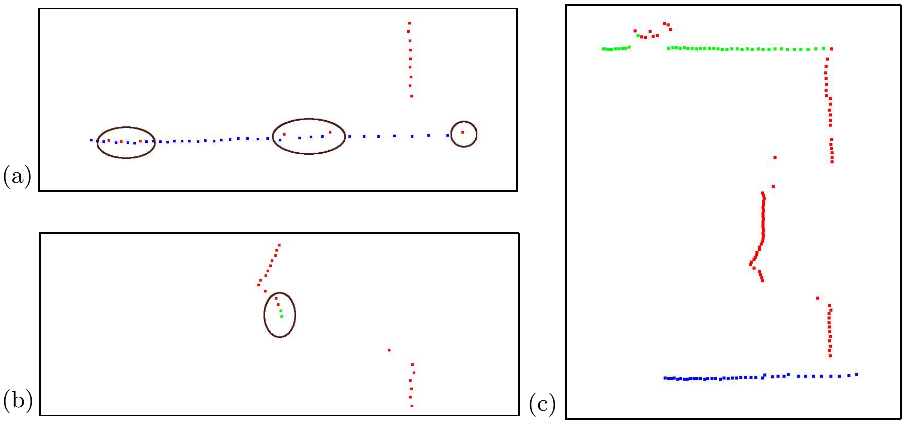

Fig. 2. Extracting semantic information using a slice of a 3D scan. (a) Problems with simple gradiant definition, marked with circles. (b) Problems with jump edges. (c) Correct semantic classification.

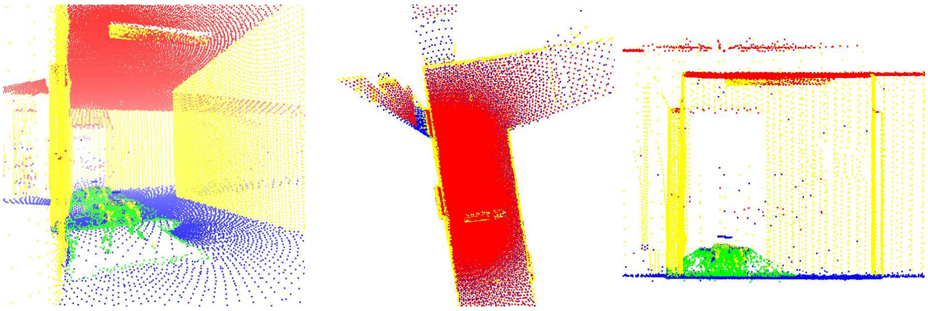

Fig. 3. Semantically labeled 3D point cloud from a single 360° 3D scan. Red points mark the ceiling, yellow points objects, blue points the floor and green points correspond to artefacts from scanning the RTS/ScanDrive and the robot.

3 Scan 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 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. Furthermore, 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 ICP algorithm registers the 3D scans in a common coordinate system and relocalizes the robot. The basic algorithm was invented in 1992 and can be found, e.g., in [3].

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 aim to find the transformation consisting of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ which minimizes the following cost function:

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

$w_{i,j}$ 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 the point correspondences. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation $(\mathbf{R},\mathbf{t})$ 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 [3]. However, this theorem does not hold in our case, since we use a maximum tolerable distance $d_{\max}$ for associating the scan data. Here $d_{\max}$ is set to 15 cm for the first 15 iterations and then this threshold is lowered to 5 cm. Fig. 4 (left) shows two 3D scans aligned only according to the error-prone odometry-based pose estimation. The point pairs are marked by a line.

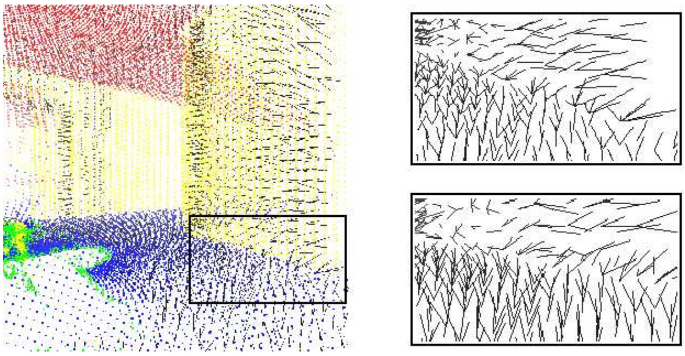

Fig. 4. Point pairs for the ICP scan matching algorithm. The left image show parts of two 3D scans and the closest point pairs as black lines. The right images show the point pairs in case of semantically based matching (top) whereas the bottom part shows the distribution with closest points without taking the semantic point type into account.

3.1 Computing the Optimal Rotation and Translation in 6D

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

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

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

In earlier work [10] we used a quaternion based method [3], 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 algorithms. It was first published by Arun, Huang and Blostein [2]. The difficulty of this minimization problem is to enforce the orthonormality of matrix $\mathbf{R}$. The first step of the computation is to decouple the calculation of the rotation $\mathbf{R}$ from the translation $\mathbf{t}$ using the centroids of the points belonging to the matching, i.e.,

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

and

$$M' = \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1,\ldots,N}, \tag{4}$$

$$D' = \{\mathbf{d}'_i \ = \mathbf{d}_i \ - \mathbf{c}_d\}_{1,\ldots,N}. \tag{5}$$

After replacing (3), (4) and (5) in the error function, $E(\mathbf{R},\mathbf{t})$ eq. (2) becomes:

$$E(\mathbf{R},\mathbf{t}) \propto \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i - \underbrace{(\mathbf{t} - \mathbf{c}_m + \mathbf{R}\mathbf{c}_d)}_{=\tilde{\mathbf{t}}}||^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i||^2 \tag{6a}$$

$$-\frac{2}{N}\tilde{\mathbf{t}} \cdot \sum_{i=1}^{N} (\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i) \tag{6b}$$

$$+\frac{1}{N}\sum_{i=1}^{N} ||\tilde{\mathbf{t}}||^2 . \tag{6c}$$

In order to minimize the sum above, all terms have to be minimized. The second sum (6b) is zero, since all values refer to centroid. The third part (6c) has its minimum for $\tilde{\mathbf{t}} = \mathbf{0}$ or

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d. \tag{7}$$

Therefore the algorithm has to minimize only the first term, and the error function is expressed in terms of the rotation only:

$$E(\mathbf{R}, \mathbf{t}) \propto \sum_{i=1}^{N} ||\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i||^2 .$$

Theorem: The optimal rotation is calculated by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. Herby the matrices $\mathbf{V}$ and $\mathbf{U}$ are derived by the singular value decomposition $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ of a correlation matrix $\mathbf{H}$. This 3×3 matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \sum_{i=1}^{N} \mathbf{m}_i'^T \mathbf{d}'_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix} ,$$

with $S_{xx} = \sum_{i=1}^{N} m'_{ix} d'_{ix}$, $S_{xy} = \sum_{i=1}^{N} m'_{ix} d'_{iy}$, $\ldots$. The analogous algorithm is derived directly from this theorem.
Proof: See [2] or [9].
Finally, the optimal translation is calculated using eq. 7) (see also (6c)).

3.2 Computing Point Correspondences

As mentioned earlier, the strategy of ICP is to always use closest points. To speed up computation, kd-trees have been proposed [3]. kD-trees are a generalization of binary search trees. Every node represents a partition of a point set to the two successor nodes. For searching points we use optimized, approximate kd-tree. The idea behind this is to return as an approximate nearest neighbor the closest point in the bucket region where the query point lies. This value is determined from the depth-first search, thus expensive ball-within-bounds tests and backtracking are not used. Here, optimization means to choose the split axis during construction perpendicular to the longest axis to minimize the amount of backtracking.

A forest of kd-trees is used to search the point correspondences. For every color, i.e., semantic label, a separate search kd-tree is created. The algorithm computes point correspondences according to the color. E.g., points belonging to the wall are paired with wall points of previous 3D scans. Fig. 4 shows the point correspondences in case of semantic based matching (top) in comparison with normal closest point matching (bottom). The points at the change of colors are paired differently. Fig. 5 shows extracted slices of the kd-trees for the colors red and yellow.

Using semantic information helps to identify the correct correspondences, thus the number of ICP iterations for reaching a minimum is reduced. In addition, maximizing the number of correct point pairs guides the ICP algorithm to the correct (local) minimum leading to a more robost algorithm.

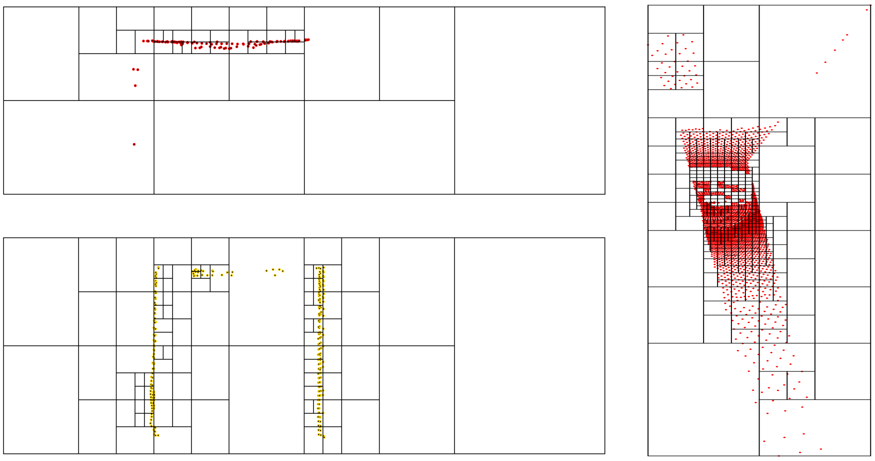

Fig. 5. A forest of kd-trees based on the semantic interpretation is used to compute the point correspondence. Left: Vertical slices through the trees (top: Ceiling points. Bottom: Wall points). Right: Horizontal slice using ceiling points.

4 Results and Conclusion

The proposed methods have been tested on a data set acquired at RTS, Hannover. Fig. 3 shows a single 3D scan with semantic labeling. Fig. 6 presents the final map, consisting of five 3D scans, each containing 43440 points. Table 1 shows the computing time for matching of two 3D scans. Using semantically labeled points results in a speedup of up to 30% with no loss of quality.

Table 1. Computing time and number of ICP iterations to align all 32 3D scans (Pentium-IV-3200). In addition, the computing time for scan matching using reduced points are given. Point reduction follows the algorithm given in [10].

used points	search method	computing time	number of iterations
all points	kd-trees	17151.00 ms	49
reduced points	kd-trees	2811.21 ms	38
all points	forest of kd-trees	12151.50 ms	39
reduced points	forest of kd-trees	2417.19 ms	35

Fig. 7 shows a detailed view of the ceiling. 3D points belonging to the lamp at the ceiling are also colored yellow. The correct match will be in no way affected by this fact. In fact, the semantic meaning is that data points of the lamp will be matched correctly with their correspondents.

Contrary to previous works, every single 3D scan is a full 360° scan. They are acquired in a stop-scan-go fashion to ensure consistency within the single 3D scans. In RoboCup Rescue the operator drives the robot and acquires 3D scans. In the 2004 competition we encountered that the overlap between two consecutive scans was sometimes too low, so that the operator had to intervene in

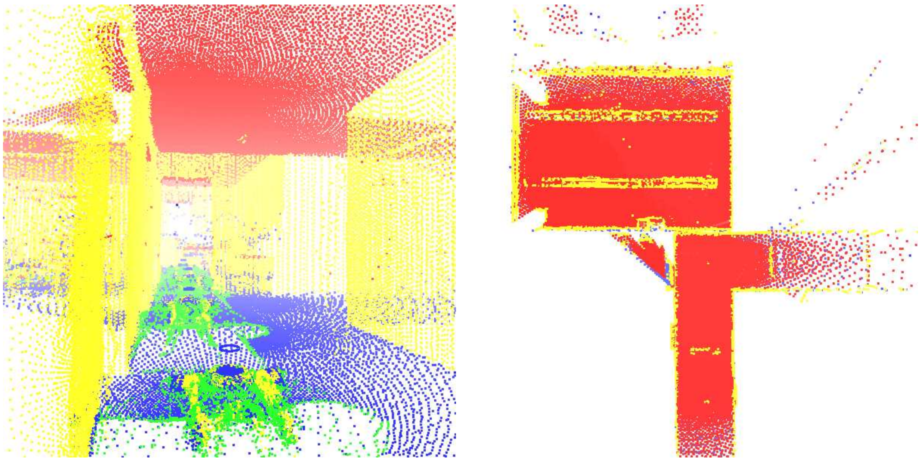

Fig. 6. The final 3D map of an office corridor / laboratory environment. The map consists of 5 3D scans and contains 217200 3D points. Left: Front view. Right: Top view.

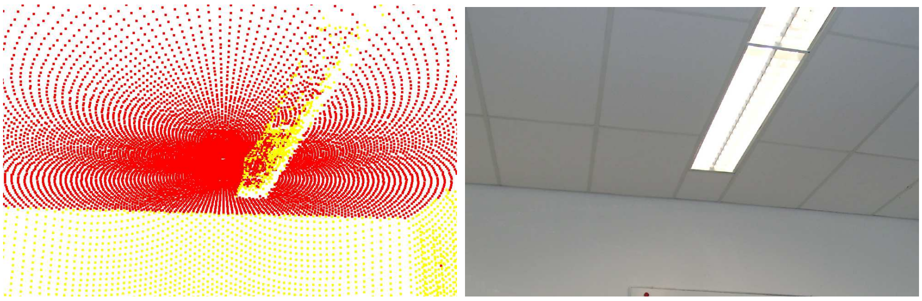

Fig. 7. Left: scanned 3D points of the ceiling including a lamp. Some 3D points of the lamp are marked yellow. The fingerprint like structure is the result of the scanning process. On plane surfaces the laser beam describes a circle. Right: Photo of the scene.

the matching process. The new scanner will reduce this problem, since it provides backward vision. Fig. 8 shows the analogous map of Fig. 6 without backwards vision, i.e., the algorithm uses only points that lie in front of the robot. The 3D scans can no longer be matched precisely, the map shows inaccuracies for example at the lab door. In fact, doors and passages are a general problem of mapping algorithms, due to the small overlap. Fig. 9 shows the final map of an additional experiment with 9 3D scans and 434400 data points.

This paper presented a robotic 3D mapping system consisting of a robot platform and a 3D laser scanner. The laser scanner provides a 360° vision; the scan matching software, based on the well-known ICP algorithm, uses semantic labels to establish correspondences. Both approaches improve previous work, e.g., [9, 10], in terms of computational speed and stability.

The aim of future work is to combine the mapping algorithms with mechatronic robotic systems, i.e., building a robot system that can actually go into the third dimension and can cope with the red arena in RoboCup Rescue. Furthermore, we plan to include semi-autonomous planning tools for the acquisition of 3D scans in this years software.

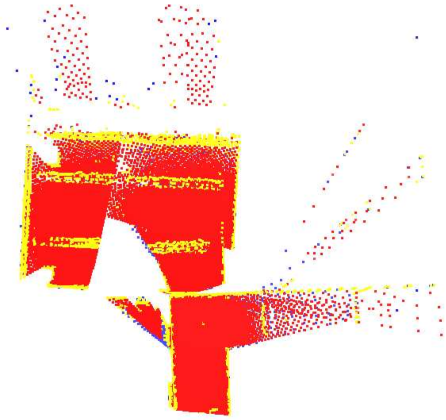

Fig. 8. Resulting 3D map (top view) if the scan matching algorithm uses only 3D points in front of the robot, i.e., the 3D scan is restricted to 180°.

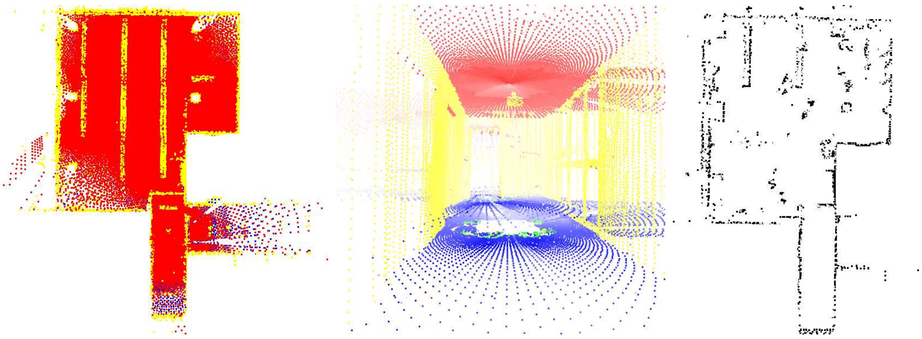

Fig. 9. Results of a second experiment. Left: 3D point cloud (top view). Middle: Some view of the points. Right: A floor plan extracted from the 3D model, i.e., only points with a height of 125 cm ± 15 cm are drawn.

References

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

2. K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. *Transactions on Pattern Analysis and Machine Intelligence*, 9(5), 1987.
3. P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, 1992.
4. 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 *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, Sendai, Japan, September 2004.
5. A. Georgiev and P. K. Allen. Localization methods for a mobile robot in urban environments. *Transaction on Robotics and Automation*, 20(5):851 – 864, 2004.
6. D. Hähnel and W. Burgard. Map Building with Mobile Robots in Populated Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '02)*, Lausanne, Switzerland, September 2004.
7. D. Hähnel, W. Burgard, and S. Thrun. Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. In *Proceedings of the fourth European workshop on advanced mobile robots (EUROBOT '01)*, September 2001.
8. A. Howard, D. F. Wolf, and G. S. Sukhatme. Towards Autonomous 3D Mapping in Urban Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, Sendai, Japan, September 2004.
9. A. Nüchter, K. Lingemann, J. Hertzberg, H. Surmann, K. Pervölz, M. Hennig, K. R. Tiruchinapalli, R. Worst, and T. Christaller. Mapping of Rescue Environments with Kurt3D. In *Proceedings of the IEEE International Workshop on Rescue Robotics (SSRR '05)*, Kobe, Japan, June (submitted).
10. A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an Application in Autonomous Mine Mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1998 – 2003, New Orleans, USA, April 2004.
11. V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan–views. In *Proceedings of SPIE, Electronic Imaging '99, The Society for Imaging Science and Technology /SPIE's 11th Annual Symposium*, San Jose, CA, USA, January 1999.
12. H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the of the 32nd International Symposium on Robotics (ISR '01)*, pages 153 – 158, Seoul, Korea, April 2001.
13. S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA, USA, April 2000.
14. R. Triebel, B. Frank, J. Meyer, and W. Burgard. First steps towards a robotic system for flexible volumetric mapping of indoor environments. In *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Portugal, 2004.
15. O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pages 4204 – 4209, New Orleans, USA, April 2004.
16. O. Wulf, C. Brenneke, and B. A. Wagner. Colored 2D Maps for Robot Navigation with 3D Sensor Data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, Sendai, Japan, September 2004.
17. O. Wulf and B. A. Wagner. Fast 3D Scanning Methods for Laser Measurment Systems. In *Proceedings of the International Conference on Control Systems (CSCS '03)*, Bucharest, Romania, July 2003.