Continuous 3D Sensing for Navigation and SLAM in Cluttered and Dynamic Environments*

Dirk Holz
University of Applied Sciences
Bonn-Rhein-Sieg,
Grantham-Allee 20,
53757 Sankt Augustin, Germany
dirk.holz@inf.fh-brs.de

Christopher Lörken
University of Osnabrück,
Albrechtstraße 28,
49069 Osnabrück, Germany
cloerken@uos.de

Hartmut Surmann
Fraunhofer Institute for Intelligent
Analysis and Information Systems,
Schloss Birlinghoven,
53754 Sankt Augustin, Germany
hartmut.surmann@iais.fraunhofer.de

Abstract— **Navigation and mapping is well understood for two-dimensional or static environments since valuable approaches exist for reacting to dynamic obstacles or for extracting static 3D information. However, today's challenge lies in combining the strengths of these approaches to obtain a system capable of performing safe navigation and obstacle avoidance based on rich 3D information of the environment while still being capable of reacting to sudden dynamic changes. In this paper we will present a methodology for continuously sensing environments in 3D and the necessary representations for exploiting the so gathered data in a way efficient enough to perform real-time 3D data based obstacle avoidance and online mapping.**

Keywords: Continuous 3D sensing, Autonomous Navigation, Obstacle Map, Structure Map, Virtual Corridor, SLAM

I. Introduction and Related Work

One of today's main challenges in the field of autonomous mobile robotics is to navigate reliably in cluttered and dynamic environments.

2D laser range-finders that measure, with high frequency and accuracy, the distances to the structures surrounding the robot became pretty much the standard to tackle the problems of simultaneous localization and mapping (SLAM). The spatial information gathered by these laser scanners is frequently being used with various 2D scan matching techniques allowing to localize the robot with three degrees of freedom and to construct two-dimensional models of the environment that are frequently and successfully used for navigation purposes.

The developed approaches (e.g. [3]) are that robust and successful that the problem of navigation and mapping can basically be regarded as being solved. However, this statement only holds for those kinds of environments that are well-structured in a way that they do not contain obstacles of a shape that does not intersect with the applied two-dimensional plane perceived by the laser scanner.

In order to deal with such objects and structures that can be found in all kinds of environments 3D information becomes crucial. Some of the most notable approaches for acquiring 3D environmental information or for mapping in 3D come from Surman et al. [5], Thrun et al. [6], Cole et al. [2], or Wulf et al. [8]. Although all these groups collect three-dimensional information, they use it mainly for map building [5], [6], [2] or for localization [8].

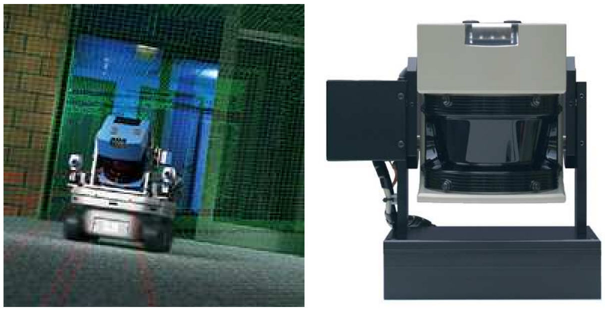


Fig. 1. The *KURT3D* robot (left) and the *IAIS 3D laser scanner* (right).

For instance Wulf et al. use a highly specialized device of a vertically mounted laser scanner that rotates continuously to gather 3D information but they use this information mainly for localization purposes. Their form of obstacle avoidance is solely based on a second 2D laser scanner device mounted in a fixed position in the front of the robot showing the same drawback of being only able to react to obstacles within the particular scan plane of the 2D scanner.

In contrast to Wulf et al., Surmann et al. and Thrun et al. integrate the gathered 3D information into their navigation procedures. Both groups follow the same approach of building environment models by performing 3D scans while the robot is standing at distinct positions. While Surmann et al. use a 2D intersection in a height of roughly one meter above ground to determine traversable areas, Thrun et al. generate a $2\frac{1}{2}$D representation taking obstacles into account for navigation decisions that are characteristic for their kind of application environments. The resulting trajectories are applicable for cluttered but static environments. However, both approaches are restricted to the three-dimensional data acquired while standing. During movement the environment is only perceived in 2D allowing only to avoid dynamic obstacles whose boundaries intersect the 2D scan plane. This kind of *non-continuous* 3D environment sensing does not meet the combination of demands faced in cluttered *and*

*This research was partly funded by the European Commission's 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.

dynamic environments.

Cole et al. have started to address this problem by introducing an approach of 3D map building based on *continuous* 3D environment sensing; i.e. while moving. Nevertheless, their robot seems to act in a completely non-autonomous way and they show no ambition to exploit this rich data for navigational purposes. Furthermore, their system does not yet perform in real-time.

This paper presents a first approach to use the rich information of continuous 3D environment sensing for both autonomous navigation and mapping. The following section provides a description of the system and the applied method of 3D sensing. An efficient data representation for safe navigation in cluttered and dynamic environments is presented in section III while section IV shows how the gathered information is used for online 2D and 3D mapping and localization.

II. Continuous 3D Environment Sensing

In our previous work we have introduced a closed system for autonomous exploration and mapping of real-world environments [5]. The system achieves robust consistent 3D modeling applying an elaborated *6D-SLAM* algorithm showing its performance at RoboCup Rescue by becoming vice world champion in Lisbon 2004. The algorithm allows for constructing three-dimensional models of the environment and to localize the robot with six *degrees of freedom (DOF)*. The system is built on the basis of the *KURT3D* robot platform and uses the *IAIS 3D laser scanner (3DLS)* to acquire spatial information about the robot's surrounding environment. Both are shown in Fig. 1.

KURT3D is a mobile robot platform with a size of $45\,\mathrm{cm}$ (length) × $33\,\mathrm{cm}$ (width) × $26\,\mathrm{cm}$ (height). The robot's maximum velocity is $5.2\,\mathrm{m/s}$. Two $90\,\mathrm{W}$ motors are used to power the 6 wheels, whereas the front and rear wheels have no tread pattern to enhance rotating. The robot has a weight of $15.6\,\mathrm{kg}$. Equipped with the IAIS 3DLS the height increases to $47\,\mathrm{cm}$ and the weight to $22.6\,\mathrm{kg}$.

The IAIS 3DLS is based on a standard SICK 2D laser range-finder. It supports a horizontal apex angle of $\Theta_{\mathrm{yaw}} = 180°$ with an angular resolution of up to $\Delta\theta_{\mathrm{yaw}} = 0.25°$ (rotating mirror device). Nevertheless, as a relatively low resolution is adequate for robust obstacle avoidance and has benefits in terms of speed concerns we use an angle resolution of $\Delta\theta_{\mathrm{yaw}} = 1°$. In this operating mode a single 2D laser scan of 181 distance measurements is read in approximately $13.32\,\mathrm{ms}$ ($\approx 75\,\mathrm{Hz}$).

In order to take three-dimensional scans of the environment, the scanner is attached to a rotatable horizontal axis. This allows to pitch the scanner over a vertical angular range of up to $\Theta_{\mathrm{pitch}} = 120°$ with a maximum resolution of $\Delta\theta_{\mathrm{pitch}} = 0.25°$. In our previous work we used the system to perform accurate and locally consistent 3D scans for mapping by taking scans over the full vertical angular range while standing. However, for a fast and continuous perception of the environmental structures relevant for obstacle avoidance while moving we now restrict this range by defining an *Area of Interest (AOI)* extending the idea of virtual roadways [4] to the third dimension, i.e. with respect to the robot's boundaries (in 3D) and thus possible areas of collision. We call this the *virtual corridor* (see Fig. 2). Its upper limit is formed by the robot's height while the lower limit is bound by the maximum size of obstacles the robot can still handle or simply by the relative floor height. Narrowing the AOI and furthermore reducing the number of consecutive 2D scans that form a complete 3D scan results in an increase of the scanner's pitch rate. Pitching the laser scanner in a continuous nodding-like fashion allows for sensing the surrounding environmental structures lying in the AOI as well as to monitor the virtual corridor for dynamic obstacle detection. Lower bound ($\theta_{\text{pitch, min}}$) and upper bound ($\theta_{\text{pitch, max}}$) of the AOI as well as the scanner's pitch rate ($\Delta\theta_{\mathrm{pitch}}/13.32\,\mathrm{ms}$) thereby depend on the robot's current velocity and can be adjusted by applying a PI-controller (see Fig. 3). Thereby, d_{min} corresponds to the distance from which on the full virtual corridor can be perceived during the pitch movement. It has to be chosen appropriately. The minimum size of the AOI for driving fast covers exactly the virtual corridor while the maximum size corresponds to a complete 3D scan over the full 120° of Θ_{pitch}.

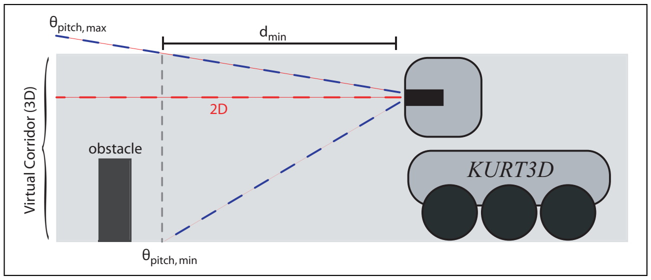


Fig. 2. Perception of the Virtual Corridor. Obstacles of different sizes can be perceived in contrast to standard 2D perception. $\theta_{\text{pitch, min}}$ and $\theta_{\text{pitch, max}}$ are chosen in a way to represent the minimal *Area Of Interest (AOI)*.

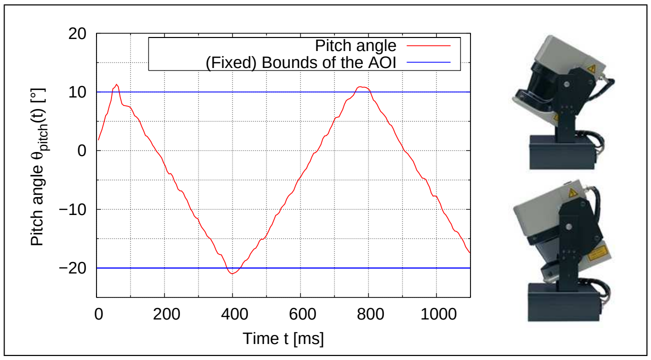


Fig. 3. Sequence of pitch angles θ_{pitch} to continuously sense the *Area Of Interest (AOI)*. Lower and upper bound of the AOI as well as the scanner's pitch rate are exemplarily fixed.

A scan point is represented by the tuple $(d_i; \theta_{\mathrm{yaw},i}; \theta_{\mathrm{pitch}})$ with d_i being the i-th distance measurement in the current laser scan while $\theta_{\mathrm{yaw},i}$ and θ_{pitch} are the yaw angle of that distance measurement and the current pitch angle of the laser scanner respectively. The cartesian coordinates of that point, with respect to the robot's left-handed coordinate frame, result from applying the homogeneous transformation in Eq. 1. The scanner's position on the robot is taken into account with the translational part $\Delta t = (x_s, z_s, y_s)^T$.

$$\begin{pmatrix} x \\ z \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & x_s \\ 0 & \cos(\theta_{\text{pitch}}) & -\sin(\theta_{\text{pitch}}) & z_s \\ 0 & \sin(\theta_{\text{pitch}}) & \cos(\theta_{\text{pitch}}) & y_s \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} d_i \cos(\theta_{\text{yaw},i}) \\ d_i \sin(\theta_{\text{yaw},i}) \\ 0 \\ 1 \end{pmatrix} \tag{1}$$

This 3D sensing design allows the robot to continuously perceive the defined AOI and detect dynamic and static obstacles in real-time.

III. Navigation and Efficient Egocentric World Representations

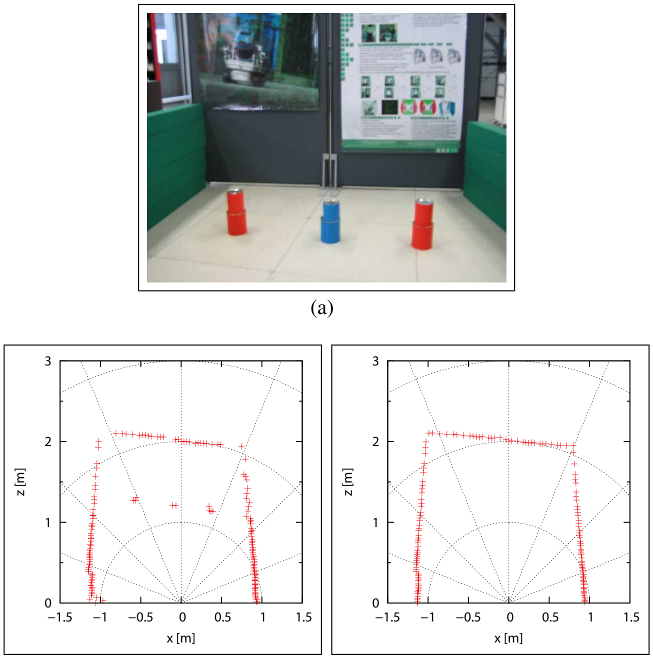
The sensor setup described in section II delivers continuous 3D data. The data flow has to be interpreted in a way that allows to react in real-time to obstacles suddenly appearing in the aforementioned AOI. Real-time capability demands for highly efficient navigation algorithms and the existing approaches that show this capability normally perform on 2D data (cf. e.g. [4]). Hence, a most promising approach is to break down the three-dimensionality of the gathered data into a two-dimensional representation that still holds all necessary 3D information but allows for the application of these existing efficient navigation algorithms.

A. Representing Three-Dimensionality in 2D

To compress the three-dimensionality of the data delivered by the scanner for real-time applicability we introduce the concepts of *2D obstacle maps* and *2D structure maps*. Both kinds of maps are local and egocentric environment representations generated from consecutive pitching laser scans.

2D Obstacle Maps: In the case of the obstacle maps the *minimum* distance in each scan direction ($\theta_{\text{yaw},i}$) is extracted and inserted into the map. These values correspond to the closest objects or obstacles in that particular direction regardless of the actual vertical angle of the scanner. Hereby, only those points are taken into account whose height above ground would intersect with the robot's bounds and the virtual corridor respectively. In this first approach, our selection mechanism assumes a flat ground structure what is, after all, a feasible assumption for indoor environments. Thus, we obtain a local map containing all obstacles in sight of the robot. Such a map is exemplarily depicted in Fig. 4(b) for the scene shown in Fig. 4(a).

2D Structure Maps: The structure maps, on the other hand, only contain those values that correspond to the *maximum* distance readings of the scanner in that particular direction, an approach inspired by the concept of *virtual 2D scans* introduced by Wulf et al. in [7]. Extracting the maximum distances automatically filters out all objects that do not extend over the full height of the AOI since the scanner will eventually look above or beneath these objects. The robot thereby replaces a previously measured smaller distance value with the newly obtained larger distance reading in that direction. The resulting map will only contain points that most probably correspond to the environmental bounds while all points that belong to smaller obstacles are filtered out as are those that belong to dynamic obstacles. Fig. 4(c) shows such a structure map.



(a)

(b)

(c)

Fig. 4. Demonstration of the map types in an example scenario (a). The *obstacle map* (b) is generated by extracting minimum distances (projected into 2D) in the continuously acquired 3D data. Extracting maximum distances results in the *structure map* (c).

While the obstacle maps are very valuable when it comes to local obstacle avoidance, the structure maps are, for instance, very suitable for robotic self-localization, i.e. for tasks that need large scale information about an environment. The obstacle maps would fail for such purposes as they would miss a lot of environmental information.

The procedure of generating these maps is quite intuitive when thinking of a standing robot. This yields, however, the question of how the maps are represented and updated while moving.

B. Representation and Update Procedure

As a great part of the algorithms that we have developed in our previous work is designed to process two-dimensional laser scan data we decided to extend the representation of standard laser scans in order to keep the algorithms compatible. The standard representation is a vector of distance measurements d_i ordered by the discretized measurement angle ($\theta_{\text{yaw},i}$). The extended representation has a variable apex angle $\Theta \in [0°, \ldots, 360°]$ and a variable angle resolution $\Delta\theta_{\text{yaw}}$. It is implemented as a vector of $N = \Theta / \Delta\theta_{\text{yaw}}$ points indexed by the accordingly discretized angle in which the measured point is lying from the robot's perspective. To minimize the computational costs of transforming points each time as input for the various algorithms the representation always maintains cartesian as well as polar coordinates.

The map update procedure consists of the following three fundamental steps and is applied for every incoming laser scan:

1) *Transformation* of the map to keep it egocentric (according to odometry).
2) *Removal* of obsolete points to handle dynamics.
3) *Replacement* of already saved points using more relevant points from the current laser scan.

If the robot stands still and no pose shift has been estimated respectively steps 1) and 2) are skipped. In the initial state the map is filled with *dummy points* that are chosen in a way that they are replaced during the first update.

1) Transformation: According to the robot's movement the pose shift between the current and the last map update (i.e. current and last reception of a laser scan) consists of a rotation $R_{\Delta\theta}$ around the y-axis by an angle $\Delta\theta$ and a translation $\Delta t = (\Delta x, \Delta z)^T$. The egocentric maps thus need to be transformed according to Eq. (2):

$$\begin{pmatrix} x_{i,t+1} \\ z_{i,t+1} \end{pmatrix} = \begin{pmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{pmatrix} \begin{pmatrix} x_{i,t} \\ z_{i,t} \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta z \end{pmatrix} \quad (2)$$

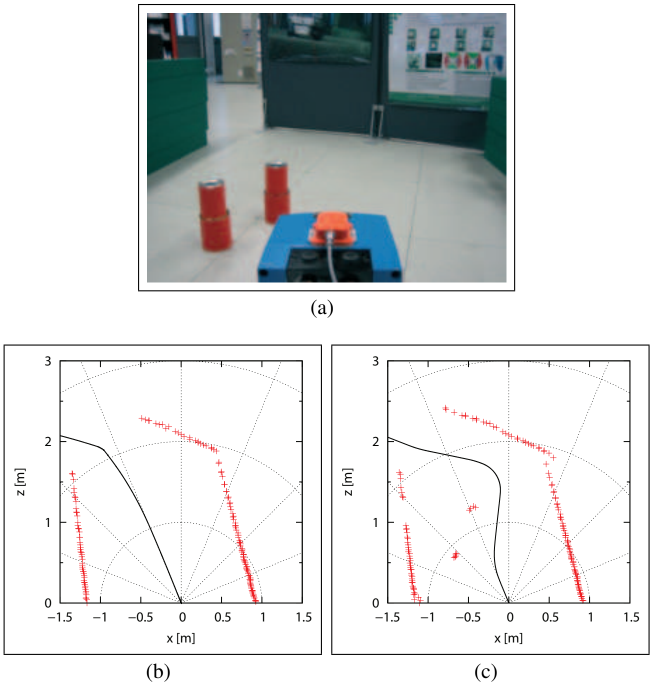
where t and $(t+1)$ represent discrete points in time.

As Eq. (2) transforms the map based on cartesian coordinates the values of the polar coordinates have to be adjusted accordingly. But due to the discretization of the N valid angles two points could fall into the same vector index. In this specific case the point being more relevant with respect to the map type has priority. Vector indices being unassigned after the transformation are filled with dummy points.

2) Removing Obsolete Points: The number of transformations applied during step 1 is stored for every single point. To deal with dynamic obstacles a saved point is removed and replaced by a dummy point after its count of transformations exceeds a threshold (e.g. 500 transformations, $\approx 5\,\mathrm{s}$). This procedure also removes erroneous points from the map that may arise due to inaccuracies in the odometric pose shift estimation.

3) Point Replacement: The final update procedure highly depends on the map type. In a nutshell a point p_i stored in an obstacle map is replaced with a point s_i in the current laser scan S if the angle of acquisition s_i^θ equals the discretized angle p_i^θ and the measured distance s_i^d is less than or equal to p_i^d; just as a point p_i stored in a structure map is overwritten with s_i if $s_i^\theta = p_i^\theta$ and $s_i^d \geq p_i^d$. When updating an obstacle map the height y of an acquired point in a perceived environmental structure is used as an additional information. If a point does not lie within the range being relevant for obstacle avoidance (virtual corridor) it will be ignored in the update procedure.

With these obstacle and structure maps the robot maintains computationally and space efficient 2D representations of a three-dimensional environment. Due to this kind of continuous 3D environment sensing and its adaption to the robot's velocity dynamic obstacles can be perceived relatively fast. Integration of this information in the obstacle map allows for reliable 3D data-based obstacle avoidance while the generation of the structure maps shows benefits in terms of localization.


(a)

(b)

(c)

Fig. 5. Behavior based obstacle avoidance in (a) an example scenario, (b) using obstacle maps by means of 2D perception with the laser scanner in a fixed horizontal position, and (c) with the continuously pitching laser scanner.

C. Navigation Using Obstacle Maps

Obstacle maps bare all the information necessary for performing reactive behavior-based robot control. We apply a simple set of behaviors implementing a set of algorithms introduced previously in [4]. The behaviors consist of:

- $Steer$ orients the robot towards the direction of maximally free space α.
- $Brake$ stops the robot in front of obstacles by examining the occupancy of the virtual corridor.
- $Turn$ turns the robot into a free direction if $Brake$ is active.

Fig. 5(b) depicts the resulting robot trajectory with the laser scanner in a fixed horizontal position (2D perception) and Fig. 5(c) the resulting trajectory with the approach presented here. In the first case, the robot was not able to perceive the obstacles as they did not intersect with the 2D scan plane. In the second they were perceived due to the pitch movement and thus integrated in the obstacle map. The robot was therefore able to avoid them successfully.

IV. Data Segmentation and Mapping

For the purpose of mapping and relative robot localization we incrementally build models of the environment and match extracted information against these maps. The particular challenge is to segment the continuously acquired 3D data flow into chunks applicable for this matching process. We thereby distinguish 2D mapping (3DOF-SLAM) and 3D mapping (6DOF-SLAM). Both algorithms are based on the *Iterative Closest Point (ICP)* algorithm by Besl and McKay [1].

Given two sets of points or *point clouds* – a *model set* $M = \{m_i \mid m_i \in \mathbb{R}^n, i = 1, \dots, N_m\}$ and a *data set* $D = \{d_i | d_i \in \mathbb{R}^n, i = 1, \dots, N_d\}$ – with dimension n, the algorithm searches for a transformation, consisting of a rotation R and a translation Δt that map D onto M. Both are determined by minimizing the error function

$$E(R, \Delta t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left|\left|m_i - (Rd_j + \Delta t)\right|\right|^2 \quad (3)$$

with the weighting factor $w_{i,j}$ encoding point correspondences; i.e. $w_{i,j} = 1$ iff m_i corresponds to d_j and $w_{i,j} = 0$ otherwise. A detailed description of our solutions to this optimization problem for both dimensionalities ($n = 2$ and $n = 3$) can be found in [5].

Once a transformation is found that minimizes Eq. 3 and maps newly acquired data onto the so far built model of the environment, the same transformation can be applied, respectively, to update the robot pose and to correct the former pose estimation obtained via odometry.

A. 3DOF-SLAM

Since our continuously acquired three-dimensional data flow is not applicable for 2D mapping as a whole we have to extract the relevant two-dimensional information. We follow the straightforward approach of extracting exactly those 2D scans during the pitch movement that have been taken in the horizontal position ($\theta_{\text{pitch}} = 0°$) during the pitch movement. These can directly be used as data set D and matched against an incrementally built 2D map; i.e. new scan points d_j that do not show a correspondence to already existing points m_i in M are added to M. Points that already have an equivalent corresponding point, i.e. those points that were matched, will be neglected as they do not provide additional information. In accordance to the approach of the obstacle and structure maps this representation is also relatively space efficient as it avoids duplicate entries in the map. Two-dimensional maps obtained by this means are shown in Fig. 6 and 7. As one can see, the resulting map does not rank behind 2D maps shown in approaches where the laser scanner is mounted in a fixed position although in this approach the robot was able to perform 3D data based obstacle avoidance while acquiring the map.

B. 6DOF-SLAM

When building a three-dimensional environment model and re-localizing the robot with all six degrees of freedom we have not only to select the appropriate scans for the matching process but to extract and transform the continuously gathered 3D data. As aforementioned, our previous work used 6DOF-SLAM algorithms on 3D data that was acquired while standing. But since the robot now moves while performing its scans we have to segment the continuous data flow in a way to obtain single three-dimensional point clouds at discrete points that are each referenced to a distinct robot pose – the *base pose* $\mathcal{P}_b$.

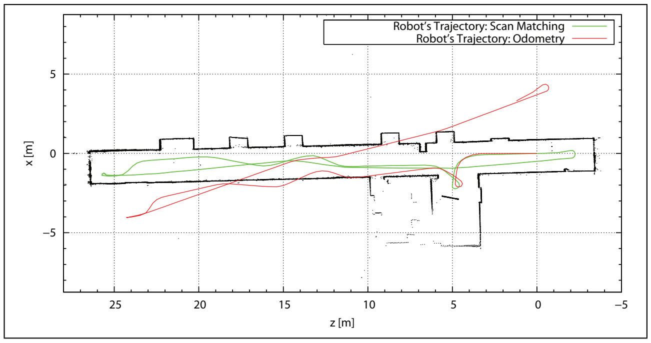

Robot's Trajectory: Scan Matching
Robot's Trajectory: Odometry
x [m]
5
0
−5
25
20
15
10
5
0
−5
z [m]

Fig. 6. 2D Map of a 30 m long corridor. The data for matching was extracted from the continuous 3D data flow.

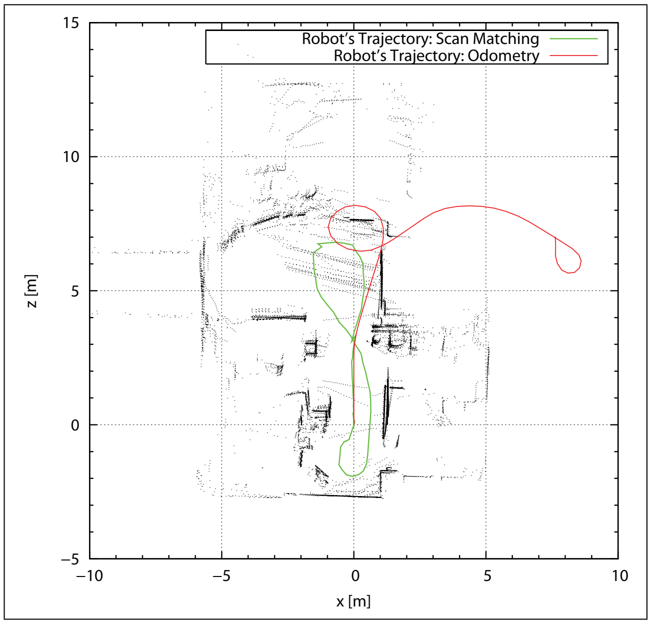

Robot's Trajectory: Scan Matching
Robot's Trajectory: Odometry
z [m]
15
10
5
0
−5
−10
−5
0
5
10
x [m]

Fig. 7. 2D Map of a cluttered example scenario. The according 3D map and photos of the modeled environment are shown in Fig. 9. Due to the online re-localization the scan matching trajectory is corrected and thus reflects that the robot drove back to its starting position.

The robot's movement in space during the phase of acquiring the scan, i.e. during the pitching movement, is stripped from the point cloud by transforming the successively gathered 2D scans that make up the cloud according to the estimated relative pose shift between the current robot pose and $\mathcal{P}_b$. Thereby a point cloud consists of approximately 15 to 500 single 2D laser scans depending on the currently used pitch rate and AOI since we segment the continuous data flow to reflect one complete pitch movement per point cloud. A point cloud being generated by this means is shown in Fig. 8. A similar approach in terms of data segmentation has been presented by Cole and Newman [2].

In order to reduce the computational load we do not build every possible point cloud for the matching process but only those whose robot base poses $\mathcal{P}_b$ are further than 2 m away from each other or that correspond to a rotation of more than 45°. This selection mechanism still guarantees a sufficient

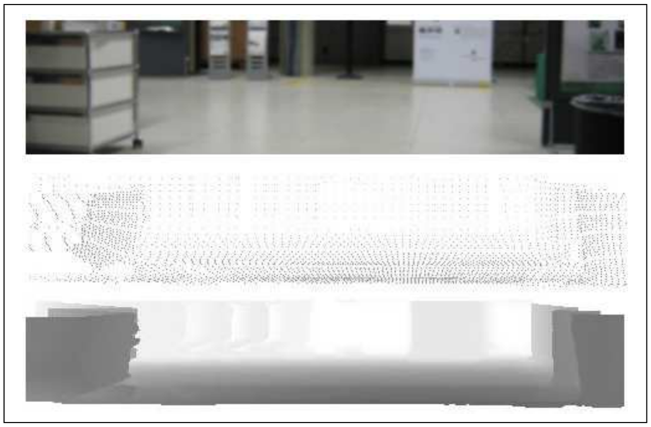


Fig. 8. A point cloud being constructed from 50 consecutive laser scans while moving approximately 1m forward. The upper part shows a photo of the scene, the middle shows the already transformed scans forming the point cloud. The bottom shows the same point cloud rendered as a depth image for visualization.

overlap of the point clouds for the matching process.

The generated point clouds are registered into an incrementally built 3D model using our 6DOF-SLAM algorithm [5]. A typical result for such a model generated while roaming the environment is shown in Fig. 9. Note that the so built model does not contain the full three-dimensional information of the environment but only the area covered in the AOI. In the depicted example the AOI was again chosen to correspond to the robot's virtual corridor. While the resulting model is slightly more distorted compared to those that can be achieved by performing 3D scans while standing, the represented information is still very substantive and consistent and thus usable for higher level robotic applications.

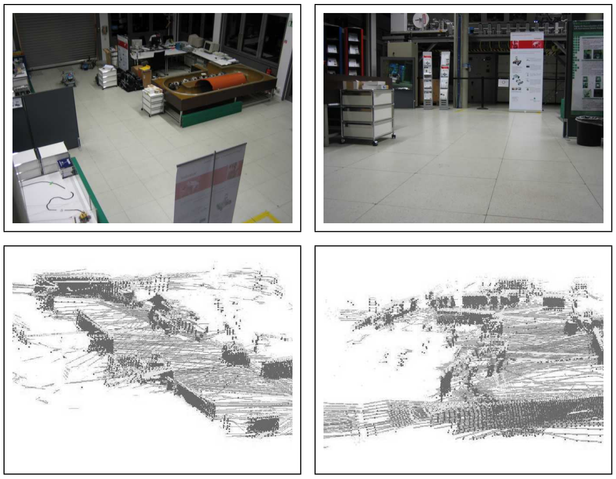


Fig. 9. 3D model of an example scenario. The model was obtained by matching generated 3D point clouds. The robot's trajectory is depicted in Fig. 7.

V. Summary and Outlook

In this paper we have presented a novel sensor setup for continuously sensing the environment in 3D. We have introduced methodologies for representing the so gathered three-dimensional data efficiently in the form of 2D obstacle and structure maps. By combining these maps with the concepts of an area of interest and the virtual corridor we demonstrated a system capable of performing reactive real-time 3D data based obstacle avoidance. With the presented data segmentation mechanisms we have furthermore provided the means for processing the same continuous 3D data in real-time for correcting the robot's pose estimations and for building 2D and 3D models of the environment. Due to this novel approach the used autonomous mobile robot was able to deal with cluttered *and* dynamic environments.

One of the shortcomings that we have already addressed concerns the selection mechanism of filtering out floor points in the update procedure of the 2D obstacle and structure maps. In our previous work we have already presented approaches for evaluating the slope of neighboring points in 3D laser scans. Such methods can easily be integrated into the system in order to determine different kinds of surfaces traversable to the robot.

Furthermore, we proposed a first approach for filtering out dynamics in the update procedures of the 2D obstacle and structure maps but we still need to tackle the same question in the 2D and 3D mapping process.

Moreover, we reduced complex navigation strategies in this first approach to mere behavior-based obstacle avoidance. Future work will thus concentrate on integrating the information provided by the 2D obstacle and structure maps into more sophisticated path planning and following as well as exploration approaches. Thereby we want to evaluate how the benefits of the proposed representations are transferable to these more complex tasks and what additional information may show to be crucial or meaningful to add.

References

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

[2] D. Cole and P. Newman. Using Laser Range Data for 3D SLAM in Outdoor Environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[3] D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

[4] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann. About the Control of High Speed Mobile Indoor Robots. In *Proceedings of the Second European Conference in Mobile Robotics*, 2005.

[5] H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. Simultaneous Mapping and Localization of Rescue Environments. *it - Information Technology*, October 2005:282–291, 2005.

[6] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, 11(4):79–91, 2004.

[7] O. Wulf, K O. Arras, H. I. Christensen, and B. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *IEEE/RAS International Conference on Robotics and Automation (ICRA)*, 2004.

[8] O. Wulf, D. Lecking, and B. Wagner. Robust Self-Localization in Industrial Environments based on 3D Ceiling Structures. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.