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

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

Christopher Lörken  
Institute of Computer Science  
University of Osnabrück  
Osnabrück, Germany  
cloerken@uos.de

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

Abstract—Navigation and mapping is well understood for dynamic 2D or static 3D environments since elaborated approaches exist that address the problems faced in standard mobile robot applications. 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 SLAM.

Keywords: SLAM, Autonomous Navigation, Obstacle Map, Structure Map, Virtual Corridor

I. INTRODUCTION AND RELATED WORK

Internal world representations or maps are prerequisites for autonomous mobile robots acting in real world environments in order to plan actions and navigate effectively. Nature and complexity of these representations highly depend on the robot’s task and application space.

2D laser range-finders that measure, with high frequency and accuracy, the distances to environmental structures surrounding the robot became the de facto standard to tackle the problem 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.

However, another essential prerequisite for the use of autonomous mobile robots is their ability to react to environmental characteristics or sudden dynamic changes that are not always perceivable by means of 2D sensors.

Figure 2 gives an overview over the different kinds of approaches that can be found in recent robotics literature. Class I of that table stands for those systems, that use standard 2D sensors and work with robot poses that have three degrees of freedom i.e. position and orientation in the plane and in 2D environment models respectively. Valuable navigation and mapping approaches that belong to this well understood class are available (for an overview see Thrun [1]). However, these approaches are limited to planar navigation and mapping based on the 2D data of the sensor. They obviously lack the ability to detect objects outside of that one scan plane like, for instance, small objects, open drawers, or descending stairs.

To overcome this limitation some groups build 3D volumetric representations of the environment with two fixed 2D laser scanners (class III in Fig. 2) e.g. [2]–[4]. Another approach are slice-wise local 3D map approaches based on several 2D scanners and a 6DOF pose representation (class II in Fig. 2), e.g. from the Stanford racing team used for terrain classification [5].

On the other hand some groups including us present full 6DOF approaches based on 3D sensors (class IV in Fig. 2) e.g. [6]–[9].

In general, all these approaches show sophisticated solutions to specific problems in mobile robot applications e.g. localization in preliminary known environments, construction

*This research was partially 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.
of precise and consistent 3D environment models or navigation techniques capable of avoiding dynamic obstacles.

However, most of these systems require a considerable amount of hardware or processing time. This paper presents our approach that aims at combining the strengths of the SLAM and navigation approaches categorized in Table 2. We use an affordable, single-device 3D laser sensor in a novel way to continuously sense the environment and to extract rich three-dimensional information that we use for both autonomous navigation and mapping.

The resulting system originates on our approach [10] and has successfully been tested at the 2007 RoboCup@Home contest performing especially well in navigational tasks.

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

A. Previous Work

In our previous work we have introduced a closed system for autonomous exploration and mapping of real-world environments. The system achieves robust consistent 3D modeling applying an elaborated 6D-SLAM algorithm [11]. 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 cm (length) × 33 cm (width) × 26 cm (height). The robot’s maximum velocity is 5.2 m/s. Two 90 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 kg. Equipped with the IAIS 3DLS the height increases to 47 cm and the weight to 22.6 kg.

The IAIS 3DLS is based on a standard SICK 2D laser range-finder. It supports a horizontal aperture angle of $\Theta_{\text{yaw}} = 180^\circ$ with an angular resolution of up to $\Delta\theta_{\text{yaw}} = 0.25^\circ$ (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 $\Delta\theta_{\text{yaw}} = 1^\circ$. In this operating mode a single 2D laser scan of 181 distance measurements is read in approximately 13.32 ms ($\approx$ 75 Hz). To take three-dimensional scans of the environment, the scanner can be rotated around its horizontal axis. The device has a vertical angular range of up to $\Theta_{\text{pitch}} = 120^\circ$ with a maximum resolution of $\Delta\theta_{\text{pitch}} = 0.25^\circ$.

In our previous work we used the system to construct both accurate and consistent 3D maps out of single 3D laser scans acquired while standing at distinct positions by taking scans over the full vertical aperture angle. 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 [12] 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. 3). 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 (see Fig. 3). Lower bound ($\theta_{\text{pitch, min}}$) and upper bound ($\theta_{\text{pitch, max}}$) of the AOI as well as the scanner’s pitch rate ($\Delta\theta_{\text{pitch}}/13.32$ ms) thereby depend on the robot’s current velocity and can be adjusted by applying a PI-controller. Thereby, $\theta_{\text{min}}$ 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$^\circ$ of $\Theta_{\text{pitch}}$.

A scan point is represented by the tuple $(d_i; \theta_{\text{yaw},i}; \theta_{\text{pitch}})$ with $d_i$ being the $i$-th distance measurement in the current laser scan while $\theta_{\text{yaw},i}$ and $\theta_{\text{pitch}}$ are the $i$-th measurement angle 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 (w.r.t. the robot’s center of rotation) is taken into account with the translational part $(x_s, z_s, y_s)^T$.

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

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. This continuous data flow has to be interpreted online in a way that allows to react in real-time to obstacles in or, suddenly appearing in, the aforementioned AOI. Real-time capability demands for highly efficient navigation algorithms and existing state-of-the-art approaches that show this capability normally perform on less complex and less information bearing 2D laser data (cf. e.g. [12]).

In order to combine these well-studied and well-performing navigation algorithms with our rich continuously gathered 3D data, we chose to break down the three-dimensionality of
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. Of course, only those points whose height above ground would intersect with the robot’s bounds and the virtual corridor respectively are inserted into the map. This explicitly includes obstacles like small objects lying around or open drawers that are not perceivable only by 2D perception (see Fig. 3).

In order to represent non-traversable areas, i.e. areas that correspond to holes in the ground like descending stairs, the intersection point of a laser beam that has an end point below floor level with the floor itself is being computed and inserted as an artificial obstacle into the map if it is the closest detected point in that direction.\(^1\)

In this first approach, our selection mechanism assumes a flat ground structure what is, after all, a feasible assumption for indoor environments.

By this method, we obtain a local map containing all obstacles and non-traversable areas close to 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 [13]. 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.

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 \( \Theta \in [0^\circ, \ldots, 360^\circ] \) 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 \((\Delta x, \Delta z)^T\). The egocentric maps thus need to be transformed according to Eq. (2):

\[
\begin{pmatrix}
    x_{i,t+1} \\
    z_{i,t+1}
\end{pmatrix} =
\begin{pmatrix}
    \cos \Delta \theta & -\sin \Delta \theta \\
    \sin \Delta \theta & \cos \Delta \theta
\end{pmatrix}
\begin{pmatrix}
    x_{i,t} \\
    z_{i,t}
\end{pmatrix} +
\begin{pmatrix}
    \Delta x \\
    \Delta z
\end{pmatrix}
\]

(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. 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, \( \approx 5 \text{s} \)). The same procedure can be applied to both kinds of maps for removing erroneous points caused by inaccuracies in the pose shift estimation. Larger pose shift errors that may arise from imprecise odometry can be corrected by matching consecutive laser scans with the algorithm presented in chapter IV-B.

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 \( s_i^\theta \) equals the discretized angle \( p_i^\theta \) 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.

C. Navigation Using Obstacle Maps

Obstacle maps bare all the information necessary for performing reactive behavior-based robot control. To show the applicability of the approach, we apply a simple set of behav-
iors implementing a set of algorithms introduced previously in [12]. The behaviors consist of:

- **Steer** orients the robot towards the direction of maximally free space.
- **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. This steers the robot out of dead ends.

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 are to small to intersect with the 2D scan plane. In the second case they were perceived due to the pitch movement and thus integrated in the obstacle map. As a result the robot’s trajectory leads around those obstacles avoiding them reliably.

Although it is not covered in this example, it is to note that the robot robustly perceives sudden dynamic changes in the environment due to the fast pitch rate of the scanner while driving. It therefore detects and avoids obstacles like, for instance, suddenly opened drawers.

IV. DATA SEGMENTATION AND MAPPING

In the previous chapters we showed that continuous 3D environment sensing together with the concepts of an area of interest and the virtual corridor enables an autonomous mobile robot to perceive and react to various obstacles being typical for home or office environments. In this chapter we will show how the same continuous 3D data flow of the pitching laser scanner can be used for Simultaneous Localization and Mapping (SLAM) both with 3DOF and 6DOF robot poses.

A. The Iterative Closest Point algorithm

For the purpose of mapping and relative robot localization in both dimensionalities we incrementally build models of the environment and match sensory information against these maps i.e. newly acquired data is aligned with or mapped onto the model and used to augment the information contained therein. To use raw sensor data in the alignment process, i.e. points in 2D or 3D Cartesian space, we use a fast variant of Iterative Closest Point (ICP) algorithm by Besl and McKay [14]:

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

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

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.

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 to update the robot pose and to correct a former pose estimation obtained e.g. via odometry, respectively. A detailed description of our algorithms and solutions to the above optimization problem for both dimensionalities ($n = 2$ and $n = 3$) can be found in [11].

B. 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 thus use firstly the aforementioned 2D structure maps that are, being constructed from consecutive laser scans and holding only the information to structural static features of the environment, highly applicable in this context.

Secondly, we follow the straightforward approach of extracting exactly those 2D scans during the pitch movement that have been taken in the horizontal position ($\theta_{pitch} \approx 0^\circ$).

Both, the horizontal scans and the structure maps, can directly be used as data set $D$ and matched against an incrementally built 2D map $M$; 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. Fig. 6 shows a two-dimensional obtained by matching 2D structure maps, whereas the map in Fig. 7 has been constructed by matching those laser scans that have been taken in the scanner’s horizontal position. Note, that in both examples all acquired points have been added to the map. As one can see, the resulting maps do not rank behind 2D maps shown in approaches where the laser scanner is mounted in a fixed position although in this approach the robot performed autonomous 3D data based obstacle avoidance while acquiring the maps. Due to the online re-localization by means of 3DOF-SLAM the robot was able to correct its odometric pose estimations (solid red lines) resulting in corrected trajectories (dotted green lines).

![Uncorrected and Corrected Trajectory](image)

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

In our previous work we used a 6DOF-SLAM algorithm together with methodologies for the detection of loop closures (the robot reaches an already visited and modeled area) and error distribution if the robot detects inconsistencies in the model when trying to match new information against formerly modeled environmental structures [11]. In that work, we took 3D scans of the surrounding environment by grabbing single 2D scans over the full vertical aperture angle of the 3D laser scanner. In order to keep these scans locally consistent, a 3D scan was only acquired while standing. The spatial information gathered by this means was then integrated into single 3D point clouds to be used as data sets in the matching process together with the robot’s pose where the scan has been taken.

Since the robot is now moving and continuously pitching the laser scanner over the AOI we have to segment the continuous data flow in a way to obtain single three-dimensional point clouds that are each referenced to a distinct robot pose – the base pose $P_b$. I.e., we have to construct 3D point clouds as they were generated inherently in the previously used 3D scan procedure. We therefore strip the robot’s movement in space during one complete pitch movement by transforming successively the thereby gathered 2D scans according to the estimated relative pose shift between the current robot pose and $P_b$. The scans are then combined to form one 3D point cloud that, depending on the currently used pitch rate and size of the AOI, consists of 15 to 500 single 2D laser scans. 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 [8].

The generated point clouds are then registered into an incrementally built 3D model using the 6DOF-SLAM algorithm of our previous work.

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 $P_b$ 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 overlap of the point clouds for registration. 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.

Although the matching of single 3D point clouds is efficient enough to be run online in a separate process, we performed only online data segmentation in this particular case, running the matching algorithm afterwards since the information of the complete 3D model was not required during runtime.

V. SUMMARY AND OUTLOOK

In this paper we have presented a novel sensor setup for continuously sensing a robots’ surrounding environment in 3D during movement. We have introduced methodologies for representing the so gathered three-dimensional data efficiently in the form of 2D obstacle and structure maps together with two applications of these egocentric representations:

- **Obstacle avoidance**: By combining 2D obstacle maps with the concepts of an area of interest and what we call the virtual corridor we demonstrated a system that performs reactive real-time 3D data based obstacle avoidance. The system detects and reacts to obstacles that are not perceivable by means of 2D perception; i.e. those classes of obstacles that do not intersect the sensor’s scan.
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