

Fast 3D Perception for Collision Avoidance and SLAM in Domestic Environments

Dirk Holz, David Droeschel and Sven Behnke
*Department of Computer Science VI, University of Bonn
Germany*

Stefan May
*Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis
France*

Hartmut Surmann
*Fraunhofer Institute Intelligent Analysis and Information Systems, Sankt Augustin
Germany*

1. Introduction

Autonomous service robots that assist in housekeeping, serve as butlers, guide visitors through exhibitions in museums and trade fairs, or provide care to elderly and disabled people could substantially ease everyday life for many people and present an enormous economic potential (Haegele et al., 2001; Pollack et al., 2002; Siegwart et al., 2003). Moreover, regarding the aging society in most industrialized countries the application of service robots in (elderly) health care might not only be helpful but necessary in the future. However, these service robots face the challenging task of operating in real-world indoor and domestic environments. Domestic environments tend to be cluttered, dynamic and populated by humans and domestic animals. In order to adequately react to sudden dynamic changes and avoid collisions, these robots need to be able to constantly acquire and process, in real-time, information about their environment. Furthermore, in order to act in a goal-directed manner, plan actions and navigate effectively, autonomous mobile robots need an internal representation or *map* of their environment. Nature and complexity of these representations highly depend on the robot's task and workspace.

When operating in preliminary unknown environments, e.g., when it is unfeasible (or simply uncomfortable) to manually model the environment beforehand, the robot needs to construct an internal environment model on its own. Moreover, in dynamic environments the robot further needs to be able to continuously acquire and integrate new sensory information to update the internal environment model in regions where changes have taken place. As integrating new information into the model (*mapping*) requires knowledge about the robot's pose (position and orientation in the environment) and determining the robot's pose requires a map of the environment, these two problems need to be considered jointly and the problem

of constructing or updating an internal environment model is commonly referred to as Simultaneous Localization and Mapping (SLAM). In fact, SLAM is regarded as one of the major prerequisites for truly autonomous robots (Wang, 2004).

Both, collision avoidance and SLAM are well understood in the two-dimensional case, e.g., when acquiring geometric information about surrounding environmental structures with 2D laser range finders. However, as will be shown in the following section, this information is not sufficient in order to adequately navigate in cluttered and dynamic, domestic environments. Presented in this work are methods and means for a fast 3D perception of the robot's surrounding environment as well as for extracting and processing relevant information in the context of robust collision avoidance and SLAM.

Section 3 will provide an overview on methods and means for acquiring three-dimensional information using laser range finders as well as related work. Extracting relevant information from the continuous 3D data stream for the purpose of collision avoidance and SLAM is described in Section 4. Efficient algorithms for collision avoidance based on this information as well as SLAM for constructing two-dimensional and three-dimensional environment models are described in Sections 5 and 6. Extensions for using recent Time-of-Flight cameras are presented in Section 7. Section 8 will contain some concluding remarks and an outlook on future work.

2. 2D-Perception in Domestic Environments

2D laser range-finders became the de facto standard sensor to tackle the problems of SLAM and collision avoidance. These sensors measure, with high frequency and accuracy, the distances to environmental structures surrounding the robot. They emit a laser range beam and measure the time until the emitted beam is received after being reflected on the surface of an object in the robot's vicinity. By means of a rotating mirror these beams are emitted over a two-dimensional plane differing, depending on the used sensor, in apex angle and angular resolution. Typically, 2D laser range finders are mounted horizontally in order to measure distances to surrounding objects in a plane being parallel to the floor. A laser scan S is a set of 2-tuples (d, θ) where d is a distance measurement and θ the angle under which the measurement has been taken, i.e.,

$$S = \{(d_i, \theta_i) \mid i \in [1, N_s]\}. \quad (1)$$

Each tuple (d_i, θ_i) in S forms the polar coordinates of a point \mathbf{p}_i measured on the surface of an object in the surrounding environment, i.e.,

$$\forall i \in [1, N_s] : \quad \mathbf{p}_i = \begin{pmatrix} d_i \cos \theta_i \\ d_i \sin \theta_i \\ 0 \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \quad (2)$$

using a right-handed coordinate frame. That is, the robot is looking along the x -axis with the y -axis extending to the left. The z -axis points upwards and represents the height of objects. Depending on the measurement principle, S can be a totally ordered set with respect to, respectively, i and θ_i in either clockwise or anti-clockwise direction, i.e., for $i < j$: $\theta_i < \theta_j$ or $\theta_i > \theta_j$. A 2D laser range scan is exemplarily depicted in Figure 1(a).

The inherent drawback of 2D laser range finders, in the context of simultaneous localization and mapping (SLAM) as well as collision avoidance, is that objects not intersecting the scanner's measurement plane are not perceived. Consider for example the couch table in Figure

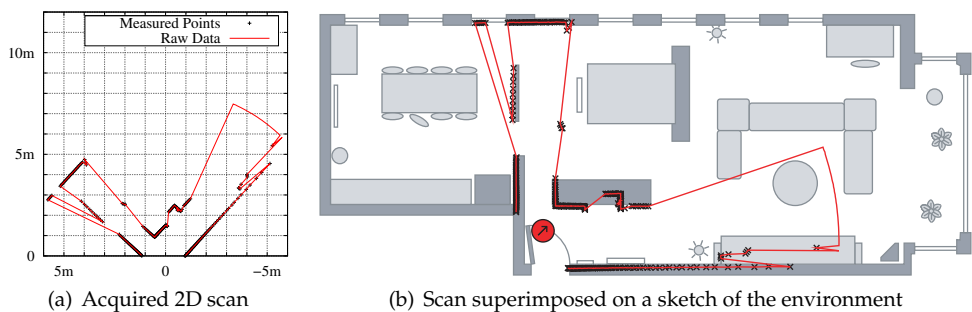


Fig. 1. 2D laser range scan in an example scenario. Depicted is a range scan (a) from a data set recorded by Zivkovic et al. (2007) together with an approximate floor plan of the scenario (b). Note, that the couch table has not been sensed at all since it did not intersect the scanner's measurement plane.

1(b). The 2D laser range scan taken in this example scenario does adequately model surrounding environmental structures whereas not a single measurement has been taken on the surface of the couch table. That is, the couch table is not at all perceived. Hence, it cannot be modeled in the robot's internal environment representation. As there is no obstacle in the corresponding model region, the robot might plan a path that directly leads through the couch table. Furthermore, a collision with the couch table cannot be avoided as it does not intersect the scanner's measurement plane. Even when standing directly in front of the table not a single measurement would be reflected. In this example, the scanner is mounted too high so that even the legs of the table do not intersect the measurement plane. However, even when intersecting the scanner's measurement plane, especially table and chair legs are not always perceivable. Depending on material and shape of table legs, e.g., round metal rods, only a portion of emitted laser beams are reflected in a way so that they are received by the scanner. The same holds true for objects whose surface is less reflective. That is, primary reasons for not perceiving an object with a 2D laser ranger finder are:

1. **The object does not intersect the 2D scan plane.** That is, objects below or above the two-dimensional measurement plane cannot be perceived.
2. **The surface of the object is less reflective** or reflects incoming range beams in directions other than the emitting range scanner. Black surfaces, for example, absorb a larger portion of the incoming light. On the other hand, metallic objects with round shape, like for instance table or chair legs, might cause diffuse reflections or completely diffract the emitted beam.

Tables and chairs are not the only objects in domestic environments that are hard to perceive with 2D laser range finders. Objects like for instance table tops, open drawers, small objects lying on the ground or stairs might not be appropriately perceivable by the robot and modeled in its internal environment representation (see Figure 2). When mounting the scanner in another height to perceive a specific class of obstacles, other types of obstacles are still not perceivable. Even the usage of several 2D range scanners in different heights does not appropriately solve this problem. For adequately handling all kinds of obstacles in a cluttered and dynamic environment 3D information becomes crucial.

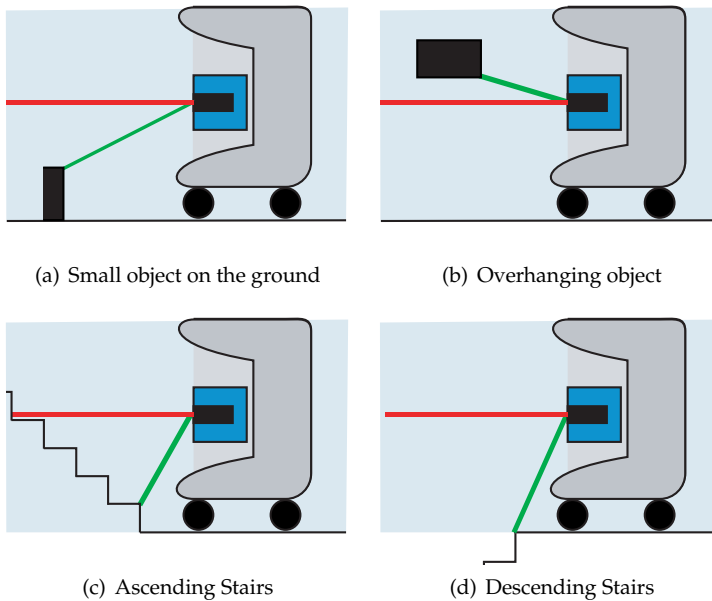


Fig. 2. Different types of obstacles. Shown are four different examples of obstacles in a robot's workspace. They have in common that the robot is not able to reliably avoid them by means of simple 2D perception. The measurement plane of a standard 2D laser range finder is depicted in red only intersecting the ascending stair. By means of 3D perception, a single distance measurement (green line) would allow the robot for detecting the obstacles.

3. Using Laser Range Finders for 3D Perception

Important criteria for the choice of a particular sensor in the design of mobile robots are size, weight, power consumption and price. Several approaches for acquiring 3D information have been proposed that differ, amongst others, in the type, number and setup of sensors. Here, we will focus on approaches that are based on range sensors.

3.1 Related Work

Thrun et al. (2000) use two 2D laser range finders. One scanner is mounted horizontally and used for localizing the robot with three degrees-of-freedom. The other scanner is mounted vertically. The data of the latter is used to compute a volumetric model of the scene based on the poses determined by using the horizontal scanner. However, whereas this approach allows for constructing three-dimensional environment models, it only provides 3D information about objects that are currently passed by the robot and intersect the measurement plane of the vertically mounted scanner. That is, objects in front of the robot are only, if at all, perceivable by the horizontally mounted 2D scanner.

Zhao & Shibasaki (2001) follow a similar approach but use multiple vertically mounted scanners in order to reduce the size of these occlusions. In fact, the usage of multiple 2D range scanners became a commonly found sensor setup in the context of the DARPA Grand and Urban Challenges. Thrun et al. (2006) mounted five 2D laser range scanners on the top of Stanley,

the robot that won the DARPA Grand Challenge. The scanners were mounted in driving direction (just like horizontally mounted 2D scanners), but under different pitch/tilt angles to perceive the surface of the ground in different distances. Similar setups can be found in many other participating teams of the Grand and the Urban Challenge. However, such a battery of 2D range scanners cannot be mounted on smaller household service robots.

Another possibility for acquiring three-dimensional information is the application of commercially available 3D laser scanners as used, e.g., in land surveying. Sequeira et al. (1998) use a RIEGL¹ laser scanner on an autonomous robot to construct 3D models of indoor environments. Allen et al. (2001) use a Leica CYRAX² laser scanner on a car to construct three-dimensional models of urban environments. A Zoller+Fröhlich³ scanner is used by Huber et al. (2000) for reconstructing volumetric models of indoor environments. Urmson et al. (2008) use a Velodyne⁴ laser scanner in addition to a battery of 2D laser range finders for detecting and avoiding obstacles in the immediate vicinity of an autonomous car in the DARPA Urban Challenge. Commercially available 3D laser scanners directly provide highly accurate three-dimensional point clouds. However, compared to 2D laser range finders, they are quite expensive and unwieldy for the application on mobile household service robots.

Yet another possibility to acquire three-dimensional data is to mount a single 2D laser range finder on a mechanical actuator to gain an additional degree of freedom. That is, in addition to the rotating mirror for scanning two-dimensional planes, the actuator rotates the complete scanner. Taking multiple 2D scans at different rotation angles allows for constructing locally consistent 3D point clouds. Different setups have been proposed that differ primarily in field-of-view (FOV) and spatial measurement density. The highest point density lies around the rotation axis. Amongst others, Surmann et al. (2003) and Hähnel et al. (2002) started using a horizontally mounted scanner where a rotation angle of 0° corresponds to acquiring a regular 2D laser range scan, i.e., with the measurement plane being parallel to the floor. Standard servo motors or pan-tilt units are used to rotate, respectively, the scanner and the measurement plane upwards and downwards in a nodding-like fashion. Wulf & Wagner (2003) evaluate this and other sensor setups and refer to it as a *pitching scanner* due to the rotation about the y -axis in a right-handed coordinate frame. For acquiring a locally consistent 3D laser scan, Surmann et al. stop the robot and rotate the scanner over the complete vertical aperture angle of 120°. Multiple 3D scans are then matched and registered into a global coordinate frame to construct a three-dimensional point model (Surmann et al., 2003). Strand & Dillmann (2008) let the scanner continuously rotate about the x -axis (*rolling scanner*) and use the acquired data for exploring and mapping indoor environments.

3.2 Continuously Rotating Laser Range Finders

The aforementioned approaches have in common that the robot is stopped in order to acquire a locally consistent 3D point cloud by rotating the scanner. The resulting behavior of the robot is thus composed of *stop-scan-move* cycles. Wulf et al. (2006) started using a continuously yawing scanner for acquiring 3D data while moving. They segment the data stream into individual point clouds and use the acquired information to localize the robot based on ceiling structures that are normally not occluded by people or objects. Cole & Newman (2006) use a pitching scanner that is continuously rotated over the complete vertical aperture angle

¹ RIEGL Laser Measurement Systems: <http://www.riegl.com>

² Leica Geosystems: <http://www.leica-geosystems.com>

³ Zoller+Fröhlich: <http://www.zf-laser.com>

⁴ Velodyne: <http://www.velodyne.com/lidar>

in a nodding-like fashion. Their robot is, however, not autonomously controlled and the acquired 3D data is solely used to construct three-dimensional environment models. However, augmenting a commercial 2D laser range finder with an additional degree of freedom seems to be the most appropriate approach for acquiring three-dimensional information on smaller mobile robots.

For safe navigation in dynamic and cluttered environments it is crucial to detect, as fast as possible, all obstacles with which the robot could eventually collide. Hence, the area in the robot's movement direction and in the height spanned by the robot's three-dimensional bounding box is of special interest. Since the continuously yawing scanner of Wulf et al. (2006) scans only vertical planes by means of a single 2D range scanner, objects in the robot's movement direction get out of sight when the scanner is sensing environmental structures behind the robot. For the DARPA Urban Challenge we have extended this setup by using two antipodally mounted 2D laser range scanners (Maurelli et al., 2009; Rojo et al., 2007). The two scanners are, furthermore, not mounted vertically but can be adjusted in the rotation angle. The resulting scanner, the Fraunhofer IAIS 3DLS-K is shown in Figure 3.a and consists of two SICK LMS 291 2D laser range finders mounted on a rotatable carrier. This carrier is continuously rotated around the vertical axis. Depending on the current orientation, the 2D laser range scans of the scanners are transformed into a sensor-centric coordinate frame. The transformed scans are aggregated to form a local 3D point cloud (see Figure 3.b). However, even when not waiting for a complete point cloud, but processing every single 2D scan as it arrives, it remains the drawback of having larger areas in the robot's vicinity not in sight until the other scanner arrives at the corresponding rotation angle. Furthermore, larger portions of the acquired information is obtained in regions that do not pose a threat to the robot, e.g., ceiling structures. This is, in fact, the reason why the two scanners are not mounted vertically as for the normal acquisition of 3D laser scans, but almost diagonally. This decreases the size of the unseen region when processing the acquired information scan-wise.

However, for further reducing the sensed regions to those that are relevant for collision avoidance, a pitching 3D laser scanner seems to be more appropriate (Holz et al., 2008; Wulf & Wagner, 2003).



Fig. 3. Continuously rotating 3D laser scanner (a) and example data (b) taken in front of the Robotics Pavilion at Fraunhofer IAIS. A photo of the scene is shown in (c). The color of the 3D points in (b) corresponds to the received remission values.

3.3 A Continuously Pitching Laser Scanner for Collision Avoidance and SLAM

In a *pitching scanner* setup the 2D laser range finder is mounted horizontally and rotated around the y -axis. A pitching scanner, the IAIS 3DLS, is shown in Figure 4. For the RoboCup@Home world championship in Atlanta 2007, it was mounted on a three-wheeled VolksBot RT3 platform allowing to acquire 3D scans of the arena. With the additional rotation axis, driven by a standard servo motor, the scanner has a vertical aperture angle of up to $\Theta_{\text{pitch}} = 120^\circ$ with a maximum angular resolution of $\Delta\theta_{\text{pitch}} = 0.25^\circ$. Taking a 3D scan by rotating the scanner over the complete vertical range and using a horizontal angular resolution of $\Delta\theta_{\text{yaw}} = 0.25^\circ$ results in 3D point clouds containing 346 080 points. However, as a relatively low angular resolution is sufficient for robust collision avoidance and has benefits in terms of speed concerns while still providing a sufficient detail for mapping purposes, an angular resolution of $\Delta\theta_{\text{yaw}} = 1^\circ$ is preferable. For the used SICK LMS 2xx range scanners, a single 2D laser scan of 181 distance measurements is read in approximately 13.32 ms (≈ 75 Hz) in this operating mode. Reliable navigation in domestic environments requires for a fast and continuous 3D perception of surrounding environmental structures and obstacles. Therefore, the scanner is continuously pitched around its horizontal axis in a nodding-like fashion allowing the robot to perceive surrounding environmental structures in 3D while moving through its workspace. Since a rotation over the complete aperture angle Θ_{pitch} might yield a couple of single 2D laser scans primarily containing information being not relevant for collision avoidance, e.g., only floor points if the scanner is directed downwards or ceiling structures if the scanner is directed upwards, we defined an *area of interest (AOI)* (Holz et al., 2008). This area restricts the range of used rotation angles θ_{pitch} to the interval $[\theta_{\text{pitch, min}}, \theta_{\text{pitch, max}}]$ so that it contains primarily relevant information.

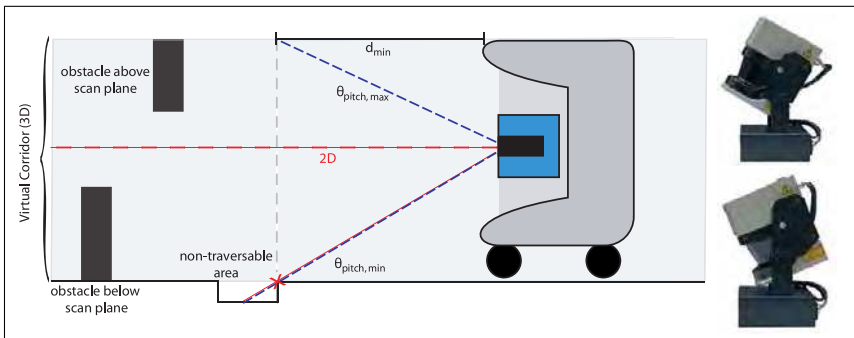


Fig. 4. Continuously pitching scanner and virtual corridor. By continuously rotating the scanner in a nodding-like fashion over the area of interest (AOI), all obstacles in the virtual corridor are perceived.

To be able to react to suddenly appearing obstacles in front of the robot even a restricted but fixed area might be too large to timely perceive especially small objects intersecting only a single measurement plane during one rotation. Therefore, the boundaries of the AOI ($\theta_{\text{pitch, min}}$ and $\theta_{\text{pitch, max}}$) as well as the scanner's pitch rate ($\Delta\theta_{\text{pitch}} / 13.32$ ms corresponding to the number of consecutive 2D laser scans taken during one pitch movement), can be adjusted, e.g., to depend on the robot's current velocity or special characteristics of the environment. Increasing the upper bound $\theta_{\text{pitch, max}}$ when moving slow allows to perceive more information about the environmental boundaries such as walls due to the height of the measured points.

For moving faster it can be decreased so that only the volume corresponding to the robot's height is sensed. We refer to this minimal area as the *virtual corridor*. It extends the idea of *virtual roadways* (Lingemann et al., 2005a) to the third dimension, i.e., with respect to the robot's boundaries (in 3D) and thus possible areas of collision. The concept of the virtual corridor is depicted in Figure 4. Here lower bound $\theta_{\text{pitch}, \min}$ and upper bound $\theta_{\text{pitch}, \max}$ of the AOI correspond to the size of the virtual corridor and thus to the robot's boundaries (marked with the slightly colored background). The length d_{\min} corresponds to the distance from which on the full virtual corridor can be perceived during one pitch movement. It has to be chosen appropriately, e.g., for a dense or narrow environment it has to be rather small whereas $d_{\min} = 1$ m is absolutely sufficient when driving fast along an uncluttered corridor. 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} . This allows to construct *complete* 3D models of the environment containing all perceivable information as in (Surmann et al., 2003). For the pitching scanner, 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 latest 2D laser scan while $\theta_{\text{yaw}, i}$ and θ_{pitch} are the i -th measurement angle and the current pitch angle of the laser scanner respectively. The coordinates of the 3D points corresponding to acquired distance measurements result from rotating the 2D measurement plane by the current pitch angle θ_{pitch} . Here, the scanner's position on the robot (w.r.t. the robot's center of rotation) is taken into account with the translational part $\mathbf{t}_s = (\mathbf{t}_s^x, \mathbf{t}_s^y, \mathbf{t}_s^z)^T$. Note that the scanner's orientation with respect to the robot's movement direction, has to be taken into account using additional rotations (not necessary here).

$$\mathbf{p}_i = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \underbrace{\begin{pmatrix} \cos \theta_{\text{pitch}} & 0 & \sin \theta_{\text{pitch}} \\ 0 & 1 & 0 \\ -\sin \theta_{\text{pitch}} & 0 & \cos \theta_{\text{pitch}} \end{pmatrix}}_{\mathbf{R}_y(\theta_{\text{pitch}})} \underbrace{\begin{pmatrix} d_i \cos \theta_{\text{yaw}, i} \\ d_i \sin \theta_{\text{yaw}, i} \\ 0 \end{pmatrix}}_{\mathbf{R}_z(\theta_{\text{yaw}, i})} + \mathbf{t}_s \quad (3)$$

By continuously sensing and monitoring the virtual corridor the different types of obstacles (see Figure 2) can be perceived. Small obstacles lying on the ground and overhanging objects do not intersect the measurement plane when using simple 2D perception, i.e., when holding the scanner in a fixed horizontal position (red line). With the continuously pitching scanner they are perceived and can thus be avoided. The ascending stair is perceivable with 2D perception but depending on the scanner's height, the measured distance is larger than the distance to the first step. With 3D perception, the first step is perceived just like a small object lying on the ground. For descending stairs, however, a little trick needs to be applied that is presented in the following section.

4. Extracting Relevant Information from 3D Data

Real-time applicability does not only necessitate a fast acquisition of information but also to efficiently process the acquired information. Due to the larger amount of data and the higher dimensionality of information, directly processing raw 3D data is not feasible in many applications. Especially in the context of navigation, existing state-of-the-art approaches that show real-time applicability normally perform on less complex and less information bearing 2D laser data (see e.g. Lingemann et al., 2005a). In order to combine these well-studied and well-performing algorithms with the rich continuously gathered 3D data it is suggestive to break down the three-dimensionality of the data into a slim two-dimensional representation that still holds all necessary 3D information but is nevertheless efficient enough to apply these

efficient algorithms. For this purpose we are using *virtual maps* that have been initially presented in (Holz et al., 2008) and that are based on the idea of virtual 2D scans (Wulf et al., 2004). These egocentric maps, in which, respectively, the robot and the sensor form the origin of the coordinate frame, store only relevant information that has been extracted from 3D data. Two types of virtual 2D maps are distinguished: *2D obstacle maps* and *2D structure maps*. Both are generated from consecutive single 2D laser range scans acquired during the continuous pitching movement of the laser scanner presented above or extracted, for example, from one depth image of a Time-of-Flight camera (see Section 7). Note that the following descriptions will focus on continuously pitching lasers scanners. The actual implementation, however, is the same for all kinds of 3D sensors.

4.1 Structure of Virtual Maps

To be able to apply the same algorithms for collision avoidance, mapping and localization purposes to both standard 2D laser scans and virtual 2D maps constructed by means of 3D perception, the representation of the virtual maps is chosen to extend the representation of standard laser scans. That is, they are organized as a vector of distance measurements d_i ordered by the discretized measurement angle ($\theta_{yaw,i}$). This extended representation has, compared to a 2D laser scanner, a variable aperture angle $\Theta \in [0^\circ, \dots, 360^\circ]$ and a variable angular resolution $\Delta\theta_{yaw}$. It is implemented as a vector of $N = \Theta/\Delta\theta_{yaw}$ points indexed by the accordingly discretized angle in which the measured point is lying from the robot's perspective. Furthermore, each point is represented by means of Cartesian and polar coordinates to avoid algorithm-dependent transformations. An example of a virtual map modeling nearest obstacles in a cluttered environment is visualized in Figure 5.

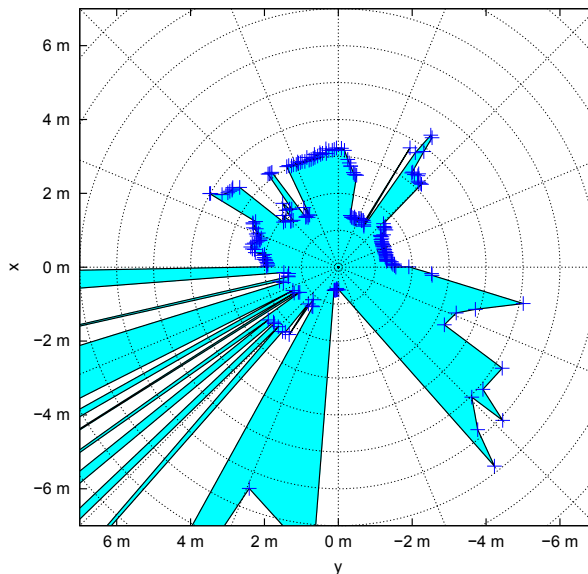


Fig. 5. Visualization of a 360° obstacle map containing information about nearest obstacles aggregated while moving through a cluttered indoor environment.

The virtual maps are ego-centric, i.e., the robot's center of rotation forms the origin of the map's coordinate frame and the coordinates of points stored in the maps are referenced to the base coordinate frame $\{B\}$ of the robot. In addition to being an efficient representation of relevant 2.5D information from 3D data, the maps can also be used to fuse information from several sensors, e.g., different 2D laser range finders or a combination of 2D and 3D laser scanners. In fact, the concept of virtual maps has recently been adopted by Stiene & Hertzberg (2009) to fuse information of different 2D range scans. Furthermore, as the representation (and the actually used implementation) does not differ from that of a standard 2D laser range scan, basically all algorithms working on 2D laser data can be applied to the aggregated or fused information in a virtual map. The remainder of this section will describe the two types of virtual maps as well as the procedures for updating them.

4.2 2D Obstacle Maps

In the case of the obstacle maps, the point corresponding to the *minimum* distance in each scan direction ($\theta_{yaw,i}$), projected into the xy -plane in which the robot is moving, is extracted and inserted into the virtual map. These measurements correspond to the closest objects or obstacles in that particular direction regardless of the actual pitch angle θ_{pitch} of the scanner. Of course, only those points whose height above ground would intersect, respectively, the robot's bounds and the virtual corridor are inserted into the map. This explicitly includes obstacles like small objects lying on the ground or overhanging objects like open drawers as shown in Figure 4. Objects not intersecting the virtual corridor pose no threat to the robot and can thus be ignored. In the update procedure of *2D structure maps* they are, of course, used since a lot of information especially on higher environmental structures would be neglected otherwise.

In order to represent non-traversable areas and especially areas that correspond to holes in the ground, like for instance descending stairs in the examples of Figure 2, artificial obstacles are inserted into the map. Such non-traversable areas are characterized by those distance measurements that correspond to points in the real environment that are located below floor level. Note that the robot is assumed to be only able to traverse almost flat floor areas what is, after all, a feasible assumption for domestic indoor environments. Once a measurement below floor level occurs in an acquired laser scan its intersection with the floor plane is computed and an artificial measurement is added to the obstacle map at exactly this point. Thereby, the robot is able to perceive descending stairs and stop before the first step is reached. Such an intersection point and artificial distance measurement representing the stair as an obstacle is depicted with a red cross in Figure 4.

An important issue when constructing virtual obstacle maps is to not add points to the map that correspond to traversable surface as the robot would otherwise avoid to move through that particular region. Under the assumption of a perfect flat floor and minimum measurement inaccuracies, the most straightforward way for determining which points belong to the floor plane, is to apply a simple height thresholding. That is, all points whose height lies approximately at $z = 0$ are filtered out and ignored in the update procedure, i.e., $floor(\mathbf{p}_i) = \mathbf{p}_i^z \in [-\epsilon_z, \epsilon_z]$ where ϵ_z represents a tolerance according to the sensor's accuracy. Amongst others, Yuan et al. (2009) follow this approach. A value of $\epsilon_z = 2.5$ cm, for example, is an appropriate tolerance for the aforementioned sensor setup leading to a robust removal of floor points but would also neglect measurements on the surface of small objects lying on the ground if their height does not exceed 2.5 cm.

A more reasonable, still simple and efficient, way for determining floor points is to evaluate the neighborhood of individual measurements. Nüchter et al. (2005), for example, apply a segmentation algorithm that exploits the order of points in a range scan, the order of range scans in the 3D point cloud and the 3D sensor setup to classify points regarding their correspondence to floor, wall, object or ceiling structures. It originates from the work in (Wulf et al., 2004) and will be used in the remainder of this section.

By the aforementioned means, i.e., filtering out minimum distances in each direction while ignoring points belonging to traversable surfaces, the robot obtains an egocentric map containing all obstacles and non-traversable areas close to the robot. An obstacle map that exemplarily shows how small objects are perceived is depicted in Figure 6.a.

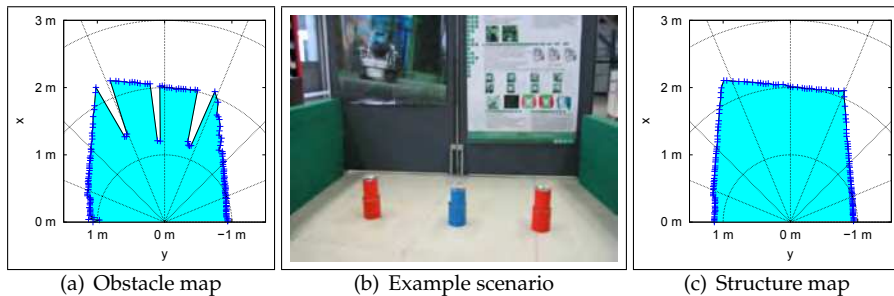


Fig. 6. Demonstration of the virtual 2D map types in an example scenario. The *obstacle map* (a) is generated by extracting minimum distances (projected into 2D) in the continuously acquired 3D data. Extracting maximum distances results in the *structure map* (c).

4.3 2D Structure Maps

In the case of virtual structure maps, the *maximum* distance in each scan direction ($\theta_{yaw,i}$), projected into the xy -plane, is extracted and inserted into the map. Extracting 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 thus only contain those points that most probably correspond to the environmental bounds while all points that belong to smaller or overhanging obstacles are filtered out as are those that belong to dynamic obstacles. Such a structure map is exemplarily depicted in Figure 6.c. Whereas the obstacle map shows the red and blue cans representing the obstacle type of small objects lying on the ground, the structure map only contains the environmental boundaries. When updating structure maps points belonging to the floor or to traversable surfaces do not need to be ignored. They are inherently replaced by points being measured on environmental structures farther away from the robot. Instead, maximum range readings need to be sorted out as they would otherwise replace shorter but valid measurements on environmental structures. However, compared to determining floor points, an according procedure is straightforward.

While the obstacle maps are very valuable when it comes to local collision 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. When sensing un-occluded parts of walls

and ceilings, using structure maps in a localization algorithm such as Monte-Carlo Localization will have similar effects as localization based on ceiling structures (Wulf et al., 2006). The obstacle maps will surely not be appropriate for such purposes as they would miss a lot of information about environmental structures.

4.4 Update Procedures of Obstacle and Structure Maps

The steps to keep both representations egocentric and to update them, according to newly acquired range scans and the definitions above, are:

- 1.) **Transformation** of the map to keep it sensor-centric (e.g., according to odometry or a known pose).
- 2.) **Removal** of obsolete points to handle dynamics and inaccurate pose shift estimates.
- 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. The same holds true if the virtual maps are used as efficient representations of single 3D sensor readings, e.g., as obtained from 3D cameras (see Section 7). In its initial state, the map is filled with *dummy points* that are chosen in a way that they are immediately replaced during in the first update, i.e., points corresponding to the maximum measurable distance for obstacle maps and distances of 0 m for structure maps. As soon as a valid measurement has been taken in the direction of the dummy point, it replaces the dummy. As these measurements correspond to either 0 m or maximum range readings, they are naturally ignored in algorithms processing 2D laser range scans (and the virtual maps respectively).

4.4.1 Transformation of the map to keep it egocentric

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

$$\begin{pmatrix} x_{i,t+1} \\ y_{i,t+1} \end{pmatrix} = \begin{pmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{pmatrix} \begin{pmatrix} x_{i,t} \\ y_{i,t} \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (4)$$

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

As Eq. (4) transforms the map based on Cartesian coordinates, the values of the polar coordinates have to be adjusted accordingly. Note, that the polar coordinates are not only stored and kept up to date in order to avoid unnecessary transformations in individual algorithms but also to avoid repetitive re-calculations in the update procedure itself. That is, even if not a single algorithm operates on the polar coordinates of the points stored in the virtual maps, this step cannot be skipped without a degradation of performance.

Due to the discretization of the $N = \Theta / \Delta\theta_{yaw}$ valid angles two points can potentially fall into the same vector index. In this specific case the point being more relevant with respect to the map type has priority. That is, in an obstacle map, for example, a point with a smaller distance replaces a point in the same angular interval that is farther away from the robot. Furthermore, vector indices being unassigned after the transformation are filled with dummy points.

4.4.2 Removal of obsolete Points to handle Dynamics

If the maps are not intended to only represent the gathered 3D information of one rotation over the AOI or a single range image as acquired by a 3D camera, but to be used endlessly, i.e., updated with every sensor reading, it is suggestive to remove points after a certain while. Therefore, 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 certain threshold (e.g., 500 transformations, approx. 5 s in the case of the continuously pitching IAIS 3DLS). This is because an obstacle passing by or crossing the robot's path leaves a trace of non-existent points in the obstacle map. This is not a drawback of the approach but a simple accounting for the uncertainty in the obstacle's movement. Points being removed by this means that correspond to static obstacles will immediately be measured again if the obstacle is still in the virtual corridor and thus still originates a possible source for a collision. The effect of applying this step can be seen in the region behind the robot in Figure 5. Here the point density is smaller than in the angular region in front of the robot, as some object points have already been removed and not sensed again since the removal.

Furthermore, if pure odometry is used to estimate the robot's pose shift for the transformation in step 1.) instead of pose tracking and scan matching to accurately determine the pose shift, errors and inconsistencies in both types of maps may arise from imprecise odometry. These are, in the same way, removed from the map. As a side note, it is to remark that even the rotated single 2D laser scans during the nodding-like movement of the sensor can be used for fast pose tracking algorithms like the one presented in (Lingemann et al., 2005b) if floor points are filtered out and the number of residual points is still sufficient for matching a newly acquired scan against the last one. Here, we apply the scan matching algorithm that is going to be described in Section 6.

4.4.3 Replacement of already saved Points

The final update procedure highly depends on the map type as described above. In a nutshell, a point \mathbf{p}_i stored in an obstacle map is replaced by a point \mathbf{s}_i in the current laser scan S if the angle of acquisition \mathbf{s}_i^θ falls into the discretized angular interval of \mathbf{p}_i^θ and the measured distance \mathbf{s}_i^d is less than or equal to \mathbf{p}_i^d . In the same way a point \mathbf{p}_i stored in a structure map is overwritten with \mathbf{s}_i if $\mathbf{s}_i^\theta = \mathbf{p}_i^\theta$ and $\mathbf{s}_i^d \geq \mathbf{p}_i^d$. The height \mathbf{s}_i^z of an acquired point in a perceived environmental structure is used as an additional information in both types of maps resulting in a 2.5D representation. That is, the virtual maps store for each discrete angle the polar coordinates $(d_i, \theta_{yaw,i})$ as well as the Cartesian coordinates $(\mathbf{p}_i^x, \mathbf{p}_i^y, \mathbf{p}_i^z)$. This will be extended in future work to not only store the particular height of the most recent points but to store minimum and maximum height of all points measured within a range of approximately 10 cm around that most recent point or in a way comparable to *Multi-Level Surface Maps* (Triebel et al., 2006). By this simple extension a complete egocentric 3D model of the surrounding environmental structures can be reconstructed on the basis of the virtual 2D maps. In the case of obstacle maps the height information \mathbf{p}_i^z of an acquired point \mathbf{p}_i is also used to neglect those points that do not lie within the virtual corridor and are hence not relevant for representing nearby obstacles. This has the effect, that the robot can, for example, underpass a table as long as the table top is higher than the highest point on the robot and the passage between the table legs is not too narrow.

5. Simple Reactive Collision Avoidance using Virtual Obstacle Maps

In addition to the goal-directed motion control of a mobile robot, e.g., to reach a certain position, reactive collision avoidance is important in dynamic and human-populated environments. That is, the motion of the robot needs to be adapted in the presence of obstacles suddenly appearing in the robot's vicinity. The virtual obstacle maps can be used with any collision avoidance procedure known from processing standard 2D range scans. As a simple example, we use a set of three simple reactive behaviors controlling and adapting the robot's translational and rotational velocities based on the robot's current movement direction and speed as well as surrounding objects modeled in the obstacle map. The obstacle map is constructed and updated during navigation. These behaviors and the corresponding algorithms have been initially introduced in (Lingemann et al., 2005a).

The first behavior slows down the robot if obstacles appear, respectively, in the virtual corridor and in front of the robot in the virtual obstacle map. If the distance to the nearest obstacle in the virtual corridor falls below a certain threshold, the robot is completely stopped. Another behavior turns the robot on the spot once it has been completely stopped. This avoids that the robot gets caught in dead ends or corners. Alternatively, the robot can be moved backwards so that it is positioned in free space again. Then an alternative path can be planned to reach the position that is to be approached.

The third behavior is more complex compared to the aforementioned ones. It slightly adapts the rotational velocity of the robot so that it prefers moving along free space. Consider for example, the robot has planned a path along a longer corridor. As this path results from searching for the shortest path between two positions, it can run directly along one of the walls. When exactly following such a path, this third behavior causes that the robot is not directly moving along the wall, but instead along the center of the corridor. The concept of the behavior is to steer the robot towards a *freespace* orientation.

5.1 Determining the Freespace Orientation

The origin of determining the freespace orientation lies in the early work of Surmann & Peters (2001) for fuzzy-based control of autonomous mobile robots. A comparable behavior was obtained by applying a fuzzy controller with fuzzy rules like the following:

```
IF COMMAND is straight-ahead
  AND IF FRONT-SENSOR is very-near
    AND FRONT-LEFT-SENSOR is very-near
    AND FRONT-RIGHT-SENSOR is near
  THEN SPEED is positive-small, ANGLE is negative-small
```

An adaption to 2D laser range scans has been presented in (Lingemann et al., 2005a). Here the following fuzzy rule is applied to every single distance measurement:

```
IF (angle_i is in driving direction) AND (distance_i is large)
  THEN drive in this direction.
```

The actual driving direction of the robot, the wanted freespace orientation α_{free} , further adapted to meet our requirements, results as follows and is based on the above rule.

$$\alpha_{\text{free}} = \text{atan2} \left(\sum_{i=1}^N \sin \mathbf{s}_i^\theta \cdot f_\theta(\mathbf{s}_i^\theta) \cdot f_d(\mathbf{s}_i^d), \sum_{i=1}^N \cos \mathbf{s}_i^\theta \cdot f_\theta(\mathbf{s}_i^\theta) \cdot f_d(\mathbf{s}_i^d) \right) \quad (5)$$

The functions $f_\theta(\mathbf{s}_i^\theta)$ and $f_d(\mathbf{s}_i^d)$ relate the i -th range reading in the form of the polar coordinates $(\mathbf{s}_i^\theta, \mathbf{s}_i^d)_{i=1\dots N}$ as obtained from a 2D laser scanner or an obstacle map to the fuzzy sets "angle is in driving direction" and "distance is large". N is the number of points in the map and the laser range scan respectively.

$$f_\theta(\theta) = \cos\left(\frac{\theta}{1.2}\right) \quad (6)$$

$$f_d(d) = \frac{1}{1 + \exp\left(-\left(\frac{d - dto_{\max}}{dto_{\min}}\right)\right)} \quad (7)$$

Here dto_{\min} and dto_{\max} determining the slope and the inflection point of the exponential, correspond to the thresholds of the behavior that slows down the robot. That is, if an object appears in front of the robot within a range dto_{\max} , the behavior starts slowing down the robot according to the distance to that object. If the distance to the object falls below dto_{\min} , the robot is completely stopped or moved backwards. Plots of the weighting functions $f_\theta(\theta)$ and $f_d(d)$ as well as the resulting application of the fuzzy AND by means of a multiplication are shown in Figure 7.

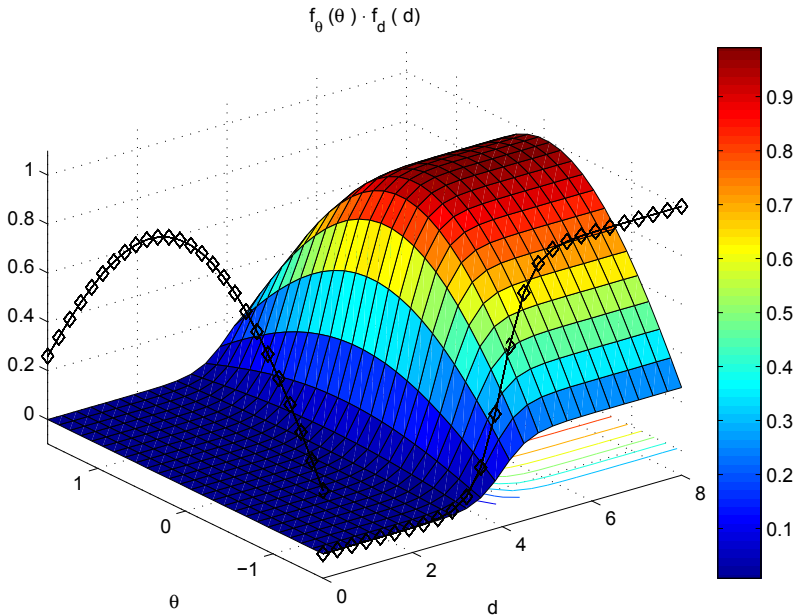


Fig. 7. Composed weighting function $f_\theta(\mathbf{s}_i^\theta) \cdot f_d(\mathbf{s}_i^d)$ to determine the freespace orientation in a laser scan or obstacle map 5.

The behavior simply adapts the rotational velocity, e.g., as set by a motion controller for following a planned path, so that the robot slightly moves towards free space thereby swerving to avoid collisions. The influence of the behavior can be adapted and is kept rather small so that the robot can enter narrow passages and follow paths that lead away from the maximally free space in the robot's workspace. Referring to the resulting weighting function $f_\theta(\theta) \cdot f_d(d)$, the robot prefers moving straight and not adjusting its translational velocity.

5.2 Typical Results

A typical result of applying the three behaviors during navigation is shown in Figure 8. The robot was put into an example scenario bounded by movable walls with an exit in the opposite corner. The experiment was repeated two times. In the first run the scanner was held in a horizontal position (2D perception) comparable to a standard 2D laser range finder. In the second run, the scanner was continuously rotated over the aforementioned area of interest in a nodding-like fashion (3D perception). In both experiments, a constant translational velocity of 0.3 m s^{-1} was set with no rotational velocity. That is, the robot was commanded to move forward. The application of the behaviors successfully moved the robot through the exit and outside the scenario. With 2D perception, small test objects lying on the ground were not perceived. The robot crushed into these objects and pushed them through the exit. With 3D perception, the objects have been perceived and the robot successfully avoided them. Although it is not explicitly 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 dynamic obstacles like, for instance, suddenly opened drawers or people passing by.

What can also be seen in the figure is that simply commanding the robot to move forward while enabling the three collision avoidance behaviors leads to an emergent behavior of wandering around. This strategy can, amongst other applications, be used to perform a random exploration.

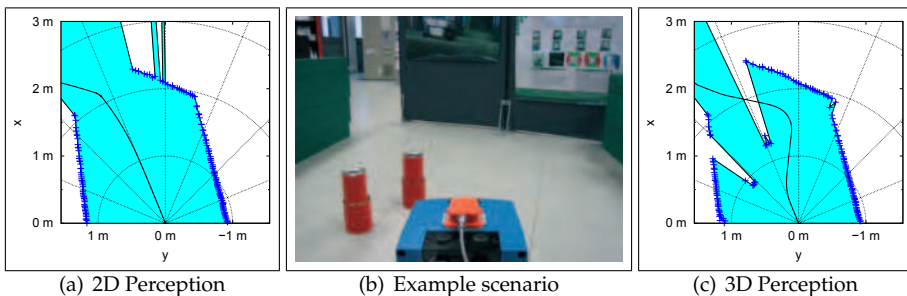


Fig. 8. Behavior-based collision avoidance in an example scenario by means of 2D perception (2D laser range finder in a fixed horizontal position) and continuous 3D perception together with the concept of obstacles maps. The small test objects are successfully avoided by the robot when using 3D perception. A video showing the robot perform in a similar experiment is available under <http://www.b-it-bots.de/media>.

6. Simultaneous Localization and Mapping

In the previous sections 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 domestic environments, like for instance open drawers, small obstacles lying on the floor or descending stairs. In this section we will show how the same continuous 3D data flow of the pitching laser scanner can be used for 2D and 3D *Simultaneous Localization and Mapping (SLAM)*, i.e., constructing two-dimensional and three-dimensional environment representations and localizing the robot

with three and six degrees-of-freedom (DOF) respectively. Central questions in this context are:

1. **Which information** can be used for 3DOF- and 6DOF SLAM respectively?
2. **Which SLAM algorithms** can be used to process this information?

6.1 Extracting Information and Forming Local Point Clouds for SLAM

The pitching laser scanner delivers a continuous stream of 3D data acquired during the robot's movement through the environment. The first question addresses the segmentation of this data stream and the extraction of locally consistent information about environmental structures usable in a SLAM approach. In the two-dimensional case (3DOF-SLAM) this is quite straightforward. By storing maximum distances in each measurement direction the virtual structure maps store aggregated information about boundaries of the environment, like for instance walls, as well as larger static obstacles. As these maps are already represented in the form of a 2D laser range scan, the contained information can be used with any known SLAM algorithm working on range scans. Constructing individual structure maps for every full rotation over the area of interest (AOI) results in locally consistent virtual range scans. Due to the fact that the rotational velocity of the scanner as well as the size of the AOI are adjusted according to the robot's current velocities, these virtual range scans have a sufficient overlap, e.g., for applying scan matching algorithms. However, the range scans are only locally consistent under the assumption that the pose shift estimates, used for keeping the structure maps egocentric during construction, are not too inaccurate. Solely transforming the maps based on inaccurate or erroneous odometric pose shift estimates can lead to inconsistent information, distorted environmental structures when turning fast for example. To account for that, we interrupt the construction of a structure map when the pose shift estimates suggest that the robot is turning fast and start building a new structure map after that turn. A similar procedure is followed by Cole & Newman (2006). However, matching consecutive scans, as described later in this section, can correct false or inaccurate pose shift estimates and the aforementioned interruption is only applied when solely using odometry information.

Another straightforward way of extracting information for 3DOF-SLAM is to take those 2D range scan that have a measurement plane being parallel to the ground, i.e., scans taken at $\theta_{\text{pitch}} = 0$ for a flat floor. Such a scan can be taken as is and is comparable to a range scan acquired by a fixed horizontally mounted 2D laser range finder. Both the horizontal laser scans as well as the constructed structure maps can be used with any SLAM algorithm processing range scans, e.g., Rao-Blackwellized Particle Filters (Grisetti et al., 2007).

For being able to construct three-dimensional environment models and to localize the robot with six degrees-of-freedom (6DOF-SLAM) we need to extract locally consistent 3D point clouds out of the continuous data stream delivered by the pitching scanner while moving. That is, we want to construct individual 3D point clouds that are each referenced to a distinct vehicle pose, the base pose \mathcal{P}_b , just like 3D range scans acquired while standing (as in Surmann et al., 2003). 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 \mathcal{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 Figure 9.

The figure also shows the results of different processing steps that are performed for every 2D laser range scan acquired during the nodding-like movement of the scanner: 1.) reducing

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

