The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

# Robust 3D Scan Segmentation for Teleoperation Tasks in Areas Contaminated by Radiation

A. Roennau, G. Liebel, T. Schamm, T. Kerscher and R. Dillmann
Interactive Diagnosis- and Servicesystems
FZI Research Centre for Information Technology, D-76131 Karlsruhe, Germany
Email: {roennau, liebel, schamm, kerscher, dillmann}@fzi.de

***Abstract*— 3D data collected by a laser scanner has great potential for robotic applications. Exact geometrical models of the environment surrounding the robot can be created from these point clouds. But, before creating any model, the 3D point cloud has to be segmented and depending on the size and quality of the point cloud, this can be a very challenging task. This article describes a robust 3D scan segmentation technique, which is capable of segmenting a 3D point cloud in a short amount of time. The results of the segmentation are used to assist a teleoperator to manoeuvre a robot through an unknown environment. Our segmentation approach copes with indoor and outdoor environments, using only a minimum of assumptions, which makes it very robust. A 3D visualisation illustrates the segmentation results in a clear and user-friendly way.**

***Index Terms*— rotating laser scanner, 3D data segmentation, 3D point cloud, traversability map, 3D visualisation, teleoperation tasks**

## I. INTRODUCTION

Laser scanners are widely used in the field of industrial applications and robotics. They are able to gather precise distance information in a very short amount of time and can be used in almost any environmental condition. In the field of mobile robotics, they are typically used to detect obstacles [1], navigate [2] and for various other tasks. However, many applications, for instance manipulation tasks [3], require three-dimensional information about the environment. By mounting such a fast 2D laser scanner onto a rotating or nodding platform, it is possible to extend the two-dimensional scan lines by a third dimension. In order to maintain the high precision of the 2D scans, it is necessary to detect the rotational or nodding movement of the scanner very exactly. With such a system, it is possible to collect a precise 3D point cloud, where the position of each point represents the distance of a reflecting object in the scanned environment. Fig. 1 shows a typical example for such a 3D point cloud. There are no real surfaces in point clouds and depending on the density, number of points and size of the points in the visualisation it can be difficult to interpret it correctly.

On the one hand, the correct interpretation of a point cloud is a challenging task, especially if this has to be done in reasonable time. But on the other hand, it is worth accepting this challenge, because a point cloud can be used to create good geometrical models. 3D point clouds have the great potential of extending the capabilities and therefore the autonomy of robotic systems. For example, they are used to navigate through narrow doors and small corridors. Object and especially plane detection is a key task when dealing with robot navigation. Planes are so important, because they can be found in almost all application scenarios for mobile robots.

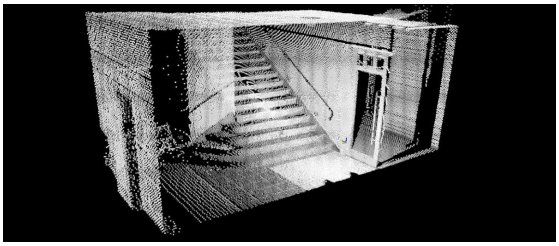

Fig. 1. Typical point cloud of a stairway.

Several different approaches on plane detection in 3D point clouds can be found in literature. In many cases the well-known Random Sample Consensus (RANSAC) method and the Iterative Closest Point (ICP) algorithm are used to detect features such as planes in 3D point clouds (e.g. [4]). The RANSAC method is also used by Schnabel et al. [5] to detect simple shapes such as tori, planes, cylinders or spheres. In a later work, they extend their approach by adding topology graphs to their initial found shapes and perform a constrained subgraph matching with known topology graphs of shapes such as windows or roofs [6]. In the work of Yang and Foerstner [7] a point cloud is divided into small blocks, before again using RANSAC on them in order to detect planes. Region growing methods can also be used to find planes in 3D point clouds. Haehnel et al. [8] find planes by using such a region growing algorithm to calculate the planes' equations with the least squares method. Surmann et al. [9] detect line segments in every scan line of a 2D laser scanner mounted on a nodding platform using standard Hough transform. Afterwards, they merge the line segments to planar structures. Other authors, like A. Sabov and J. Kruger, also use a line segmentation followed by a line combination algorithm to segment range camera images into planar surfaces [10].

Besides creating 3D point clouds and extracting objects from these, laser scanners are often used to create traversability maps. These maps can then be used for autonomous

navigation of mobile robots (e.g.[11],[12]).

In this paper we will present an approach to segment a 3D point cloud into traversable planes, including passable doorways, obstacles and walls. First, we extract line segments from each scan line. These line segments are then used to estimate multiple plane equations. After finding the ground planes, a 2D traversability map is created by moving the bounding box of the robot along the ground planes. Based on the traversability map, a 3D visualisation of the segmentation results is created in the original point cloud. Due to the unknown application scenario, we have minimised our assumptions with the target of creating a very robust segmentation algorithm, which can cope with any kind of indoor or outdoor environment. Because our results are used to help a teleoperator to navigate in an unknown scenario, it is very important to present the segmentation results in a user-friendly way (3D visualisation) and in a reasonable short amount of time.

In Section II, we will present the real world setup and our 3D scan segmentation approach. Experimental results, showing the computation time and segmentation quality of our approach, will be given in Section III. We will summarise and present future work in Section IV.

## II. APPROACH

The 3D point clouds we are considering are created by a commercial 2D laser scanner (SICK LMS200), which is mounted on a rotating platform. A rotary feed through with slip-ring is used to transmit the power and data between the laser scanner and a PC in an environmentally sealed box. This box includes several DC-DC converters and an UCoM (Universal Controller Module) [13], controlling the movement of the rotating platform. The left part of Fig. 2 shows the rotating SICK laser scanner (RoSi), which was developed at our institute. Each point $\vec{x} = (x,y,z)^T$ of the 3D point cloud can be determined as follows [14]:

$$x_i = \cos\beta \cdot \cos\alpha_i \cdot d_i \tag{1}$$

$$y_i = \sin\beta \cdot \cos\alpha_i \cdot d_i \tag{2}$$

$$z_i = \sin\alpha_i \cdot d_i \tag{3}$$

where $\alpha_i$ is the measurement angle and $d_i$ is the distance of one point in a single scan line. The angle $\beta$ is the rotational angle of the rotating platform. By changing the rotational velocity $\dot{\beta}$ the number of scan lines per full revolution and therefore the density of the point cloud is also changed. The laser scanner itself scans at a rate of 37.5 Hz and has a viewing angle of 180 degrees. We are only using the half of each scan line with a resulting viewing angle of 90 degrees. This is sufficient to capture the entire scene with one revolution. The RoSi system is operating with an angular resolution of 0.5 degrees and is therefore receiving 180 points per half scan line. Depending on the rotational velocity $\dot{\beta}$, the number of points in a RoSi point cloud varies from approx. 20000 to 80000. The amount of reflected light is used to measure a grey-scale value for each point. This remission value is attached to each point and adds important extra information to the point cloud [14]. It can be used to differentiate between surfaces with different reflectivity values. However, surfaces creating a total internal reflection or transparent objects can not be measured.

Fig. 2. (left) RoSi: Rotating SICK laser scanner system, (right) LMF: radiation-tolerant teleoperated service robot.

The RoSi system is carried by the LMF [15], a robot which is operated by the KHG [16]. The LMF (see right part of Fig. 2) is a radiation-tolerant robot, which was developed by CYBERNETIX, France, with the ability to operate inside a nuclear power plant. The LMF body is a two tracked platform, that is able to climb up and down stairs at 45 degrees and turn around with a diameter of 1.2 m. It is able to manipulate a object with a mass of 100 kg and resist a very high radiation. The LMF can be equipped with heavy-load hydraulic or electrical master-slave manipulation arms.

Our 3D scan segmentation problem is specified as follows. Given a point cloud, which was collected by the RoSi system, we want to support the operator of the LMF while working in an unknown scene by displaying the areas, which are traversable for the LMF. Because the RoSi system is carried by an arm of the LMF, we cannot assume a fixed orientation or position of the scanning system. In this article the LMF will follow a "scan-and-stop" strategy, therefore we will not address any problems originated from a movement of the platform. It is necessary to segment the captured point cloud in a small amount of time, in order to not disturb the workflow of the operator. The results have to be displayed in an user-friendly and clear way. This visualisation is supposed to give the operator a better spatial feeling for the scene and simplify decisions concerning the navigation of the LMF, e.g. if a doorway is passable for the robot. The next subsections will give some details on the developed algorithm.

### A. Line detection

Each 3D point cloud consists of several scan lines. We rotate every scan line by its given angle $\beta$ around the Z-axis in the Cartesian coordinate system (that is, the rotational axis of the RoSi system):

$$\vec{x}'^T := \begin{pmatrix} x' & y' & z' \end{pmatrix} = \vec{x}^T \cdot \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$y' = -\sin\beta \cdot x + \cos\beta \cdot y \tag{5}$$

This rotation of a scan line is shown in Fig. 3. Inserting equations (1)-(2) in (5) we can see that the Y-coordinate $y'$ is always zero:

$$\begin{aligned} y' &= -\sin\beta \cdot (\cos\beta \cdot \cos\alpha \cdot d_i) + \cos\beta \cdot (\sin\beta \cdot \cos\alpha \cdot d_i) \\ &= \cos\alpha \cdot d_i \cdot (\sin\beta \cdot \cos\beta - \sin\beta \cdot \cos\beta) \\ &= 0 \end{aligned}$$

Every 3D scan line is considered as a single 2D images by choosing the X- and Z-coordinate as the 2D coordinate axes (U- and V-coordinate). Then we use a standard Hough transform on these images to find single line segments in every scan line (see Fig. 4). These line segments are given by their start and end points. Now, the found line segments are transformed back into 3D space by adding the Y-coordinate with $y = 0$ again. Finally, the start and end points of every line segment are rotated back using equation (4) with the inverted angle $-\beta$.

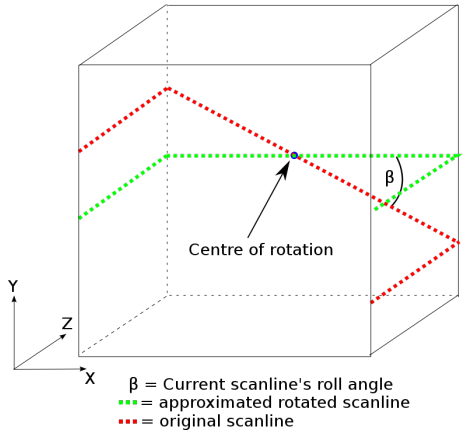


Fig. 3. Rotation of a scan line around the Z-axis, which is the rotational axis of the RoSi system.

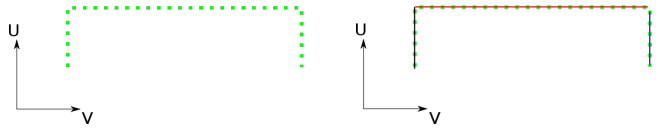


Fig. 4. (left) Rotated 3D scan line as 2D image, (right) found line segments with Hough transform.

### B. Line clustering

The line clustering algorithm iterates through all found line segments of one point cloud. By comparing the Euclidean distances between the start and end points of two line segments and their angle $\alpha$, it is possible to find similar line segments. By using the direction vectors of the line segments $\vec{m}$ and $\vec{n}$ in three-dimensional space, we can easily calculate their angle $\alpha$ by using the dot product (indicated by $*$):

$$\begin{aligned} \vec{m} * \vec{n} &= \|\vec{m}\| \cdot \|\vec{n}\| \cdot \cos\alpha \\ \Longleftrightarrow \alpha &= \arccos\left(\frac{\vec{m} * \vec{n}}{\|\vec{m}\| \cdot \|\vec{n}\|}\right) \end{aligned}$$

Because each plane consists of a set of similar lines, it is possible to find these planes using line clustering (see Fig. 5). Here, the scan lines of the RoSi system lie very close to each other and all meet at the centre of rotation. By adapting the definition of similarity this approach can be transferred to other point clouds, e.g. collected by a nodding platform. If the algorithm finds enough similar lines, a least squares plane equation is estimated. Then the rest of the line segments are no longer compared to each other, but to the plane equation. This minimises the possible drift, which can occur when only comparing the line segments to each other. The plane is refitted every time the number of new additional similar line segments reaches a threshold value.

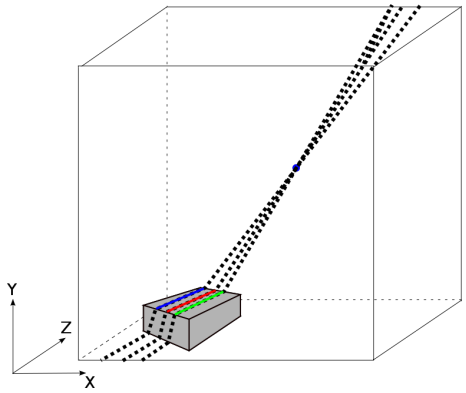


Fig. 5. Line clustering: Due to the scan lines created by the rotating scanner, a plane has a number of very similar line segments (blue, red, green line). With the help of such similar line segments it is possible to find the corresponding planes.

### C. Plane generation and validation

If the number of line segments exceeds a threshold value (in our case 25), the start and end points of all similar line segments are used to calculate the final plane equation. Because the lines corresponding to one plane fulfil the similarity definition of the line clustering algorithm, the number of outlier points is very small. In this case, a least squares estimation for the plane equation is able to determine a very exact equation. It can still happen, that the line clustering algorithm finds planes, which are not really existent in the point cloud. The validation of a plane equation in a noisy point cloud is a difficult task [8] and the environment of teleoperation tasks in contaminated areas can vary strongly. This is why our approach can only use few assumptions and we cannot eliminate any planes in advance by referring to usual indoor geometrical structures. But it is still necessary to validate the plane equations received by the line clustering algorithm. Therefore, we have developed a fast and robust plane validation technique. We create a discrete grey-scale image corresponding to the actual plane and its supporting points (see Fig. 6). As supporting points we consider points with distances to the plane, which are smaller than a threshold value. Each pixel represents a region (in our case approx. 0.2 x 0.2 m) of a plane. The grey-scale value of each pixel is related to the number of supporting

points lying in this region. More supporting points make a pixel brighter. If the size of a area with enough supporting points exceeds a threshold size, the plane can be assumed to be existent in the real 3D point cloud. Otherwise the plane is deleted. With the help of typical image processing algorithms, this is calculated in a very fast and efficient way. When segmenting indoor environments, nearly all planes have a big number of supporting points. Even if the tested plane does not exist in the point cloud, the plane will cross several walls and other real planes, creating many supporting points at these crossings. This is why we do not only consider the number of supporting points, but also their spreading.

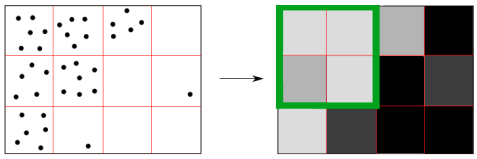

Fig. 6. Fast and robust plane validation technique: Find areas of a minimum size with enough supporting points (green surrounded area).

### D. Traversability check

Next, we need to find the ground planes among all real planes. Only planes with a slope smaller than the maximum drivable slope of the LMF (45 degrees) are possible ground planes. In this way we can remove all planes belonging to walls. The ceiling is removed by assuming that the scanner's position is higher than the ground. The found ground planes are then used to create a traversability map. First obstacle and walls are detected. The traversability algorithm iterates through all points and if the distance of a point to the ground plane is smaller than the maximum height of the robot, the position of the corresponding point is marked in the traversability map. Depending on the exact distance, it is marked as a probably passable obstacle (orange) or as an impassable wall (red). If the point is a supporting point of the ground plane and if there is no obstacle or wall already marked in the map, the map is marked as traversable (green) at this position. After creating this basic map with respect to the height of the robot, it is then used to check the traversability according to the x- and y- dimension. A bounding box with different kinds of orientations is moved along the 2D traversability map, marking those parts of the map, which are traversable and not. This is done with the help of image processing algorithms, because the traversability map is represented as a standard colour image. Problems occur at narrow doors. Due to discretisation errors, it is hard to determine if the robot can pass a doorway, which is nearly the size of the robot. We solved this problem by inserting an intermediate discretisation step. First, we create a rather rough map with a rough discretisation. Then we enlarge the bounding box of the robot by half a discretisation step and create a second rough map. The parts of the two maps that differ are then analysed with a more precise map with a smaller discretisation. This is how we are able to find narrow passages with a size close to size of the robot.

TABLE I

EXPERIMENTAL RUNTIME RESULTS

| scan no. | points in scan | drawn triangles | av. time (sec) |
|---|---|---|---|
| 1 | 14400 | approx. 49000 | 8.4 |
| 2 | 24480 | approx. 77000 | 9.2 |
| 3 | 26460 | approx. 75000 | 9.0 |
| 4 | 70200 | approx. 213000 | 16.8 |
| 5 | 72900 | approx. 277000 | 12.8 |
| 6 | 75420 | approx. 133500 | 10.8 |
| 7 | 75600 | approx. 134600 | 10.2 |
| 8 | 76140 | approx. 282000 | 14.2 |

### E. Visualisation

The 2D traversability map, which is mainly used for calculations, is also provided to the teleoperator. It gives a quick overview of the actual scene. The 2D traversability map is then transferred into the 3D point cloud. This more precise 3D visualisation gives a better spatial feeling for the real environment than the 2D map.

A standard 2D Delaunay triangulation is used to display the traversable parts of the 3D point cloud. The colours of the 3D visualisation correspond to the colours of the 2D traversability map. Impassable walls are approximated by red cuboids and obstacles are represented by several small cubes. In this way all sorts of objects can be approximated. The obstacles are highlighted in orange. The drivable parts (green) of the ground planes are triangulated using a Delaunay triangulation. As all supporting points of a drivable area lie roughly on a plane, we can use the 2D triangulation instead of a complex 3D triangulation, achieving a worst-case complexity of $O(n \log n)$ instead of $O(n^3)$ [17]. Using rotation and projection of the 3D points which both lie in $O(n)$ we can get an approximated 3D triangulation using a fast 2D Delaunay triangulation.

## III. RESULTS AND EXPERIMENTS

The described 3D scan segmentation algorithm was developed for teleoperation tasks. Therefore, the runtime has to be in an acceptable range. Depending on the received point cloud and complexity of the scene, we achieved a runtime considerably smaller than one minute on an Intel Core 2 Quad 2.66 GHz. Table I shows the measured runtime results for 5 randomly picked point clouds. The runtime results were averaged over 5 runs.

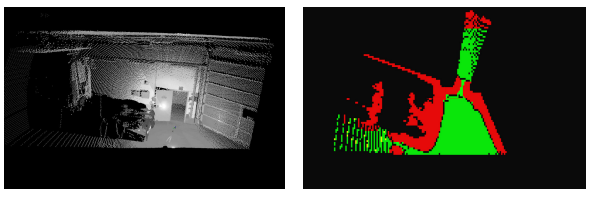

Fig. 7. (left) Collected point cloud of a garage with open door, (right) created 2D traversability map for the smallest configuration of the robot.

With the point clouds collected by the RoSi system our 3D segmentation algorithm has a runtime smaller than 20

seconds. Table I also shows that the runtime is not directly proportional to the size of the point cloud. Although scan no. 1 only has approx. one-fifth of the size of scan no. 7, the runtime difference is very small (approx. 2 seconds). The number of triangles created for the 3D visualisation only has a small influence on runtime. This is obvious, when comparing the runtimes of scan no. 4 and 5. Our analyses have shown that the number of line segments and planes found in the point cloud have the biggest influence on the runtime. The LMF is using a "scan-and-stop" strategy, therefore the achieved runtime fulfils the teleoperation requirements for the LMF.

Hundreds of scans of all kind of indoor and outdoor scenarios have been used to evaluate the developed 3D segmentation algorithm. The following figures will illustrate some of the results. Fig. 7 shows the collected point cloud of a garage with a truck standing in it and an open door. The door has a width of approx. 0.9 m and can be seen as a kind of a benchmark for this algorithm. The LMF has a width of 0.85 m in its smallest configuration. The 2D traversability map created for the smallest configuration is presented on the right side of Fig. 7. This map shows that the robot is able to pass through the doorway. The corresponding 3D visualisation can be found in Fig. 8. The way through the door is coloured in orange, because it was found with the intermediate discretisation step, meaning that the passage is very narrow and the operator of the LMF has to be careful passing this door.

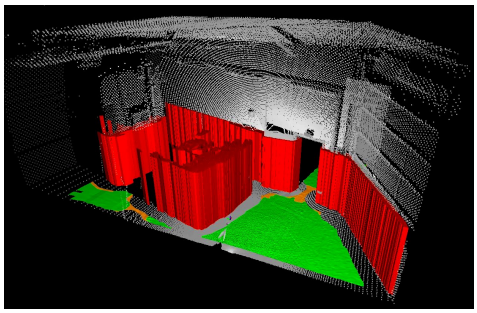

Fig. 8. Rotated 3D visualisation of segmentation results for the garage point cloud.

Then the configuration of the robot is changed. Now, the robot is wider, because one of its arms was moved. The 2D traversability map indicates that the robot can no longer drive through the door (see left part of Fig. 9). The 3D visualisation (right part of Fig. 9) shows this by removing the orange way through the door.

The LMF can also change its height. With this configuration there is no visible change on the 2D traversability map, but an obvious change in the 3D visualisation (see Fig. 10). Now, the doorway is too low for the robot and it is visualised as a part of the impassable wall.

Fig. 11 illustrates the segmentation results for a different kind of point cloud. Here, the scene shows a hallway with a slope and some stairs on the right side. In this point cloud, the segmentation algorithm found more than one ground plane

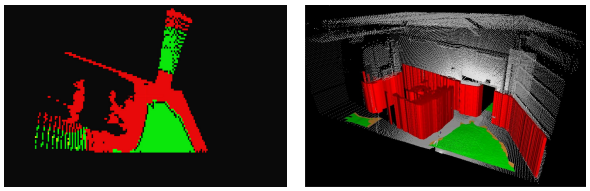

Fig. 9. (left) 2D traversability map for robot with wider configuration - robot cannot pass door anymore, (right) 3D visualisation of the left traversability map showing the open door, which cannot be passed.

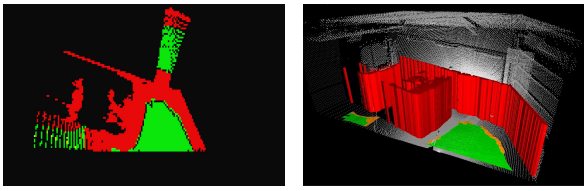

Fig. 10. (left) same 2D traversability map for robot with higher configuration - robot can also not pass door, (right) 3D visualisation of the traversability map showing that the doorway is now visualised as impassable wall.

and many obstacles, which probably can be passed by the robot. The traversability check is able to cope with ground planes having different kind of slopes. Both planes were classified as traversable (green). The stairs, which are part of these obstacles, are marked orange. In these cases the operator has to be careful when driving over the obstacles. The 3D visualisation in Fig. 12 shows the same scene from a different perspective. The ability to change the perspective is a major advantage of a 3D visualisation. Here, we can also see, that the drivable ground plane is not drawn completely up to the walls. The reason is that we are using the centre of the robot to calculate the traversability map. In this way the thickness of a way through a passage illustrates how narrow a passage is. These are only some of many other interesting results. For instance, the segmentation of very dense or sparse point clouds also works very well.

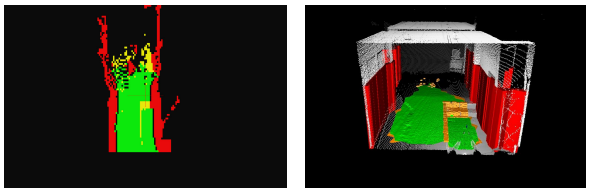

Fig. 11. (left) 2D traversability map of a hallway with two ground planes, (right) 3D visualisation of the traversability map.

Walls and obstacles made of transparent material like glass create no points in the RoSi point clouds. Therefore, we have no possibility to segment these transparent objects. In some rare cases our approach is not able to find a ground plane. This happens when the point cloud contains very few ground points. Then, the line clustering algorithm is not able to find enough line segments on the ground plane or the plane validation algorithm removes the plane, because there were not enough supporting points. In these cases, the algorithm delivers no results. But in most cases, after moving the LMF

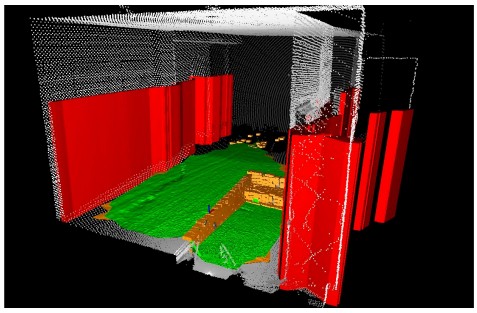

Fig. 12. Rotated 3D visualisation of the traversability map of the hallway showing the two found ground planes.

or the arm of the robot a little, the algorithm is able to find the ground plane again and shows correct results.

## IV. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

This work has presented a 3D scan segmentation approach, which is able to segment a point cloud in traversable and non-traversable parts. It can easily be transferred to many other systems and other application scenarios, because only very few assumptions have been made. Therefore, it is not limited to applications in radioactive facilities.

Hundreds of different scans have been evaluated, with very good results. Although the segmentation algorithm had some problems with a few scans, we never visualise false results. If the scan segmentation is not able to deliver correct results, this is recognised and nothing is shown in the 3D visualisation. This is an important aspect, when working in hazardous environments. Our approach can cope with indoor and outdoor environments, with multiple ground planes and with many kinds of different point clouds (sparse, dense, rotated, etc.). The segmentation results are visualised in less than 20 seconds. So, the developed approach is robust and reasonable fast at the same time. Summing up, the presented 3D scan segmentation approach is able to support a teleoperator with a clear 3D visualisation of the traversable areas in a very effective way.

### B. Future Works

Our future work will concentrate on segmenting our point clouds into more types of objects. Now, we are only using three types: drivable floor, probably passable obstacle and impassable wall. We are planning on adding at least two more types: stairways and doors. The LMF is able to drive up stairways and we want to find these stairways. We are also working on an assisted door pass manoeuvre based on a more detailed door detection (orientation, door frame thickness, etc.). The traversability check is the slowest part of the 3D segmentation. We are working on an improved traversability check. Besides this we are still improving the runtime performance of the algorithm by splitting the calculations into multiple tasks and then using more than just one core of the CPU. Then the plane detection results could be visualised before the traversability results in less than one second. Until now, the visualisation of the walls and obstacles is only approximating. Soon, we will replace this approximation by a fast and more detailed visualisation.

## V. ACKNOWLEDGMENTS

Major parts of this work were founded by the KHG Kerntechnische Hilfsdienst GmbH as part of the 'Modellbasierte Wegplanung' research project. We would like to thank the KHG for giving us such great support while developing the presented 3D scan segmentation approach.

## REFERENCES

[1] D. Amarasinghe and G. K. I. Mann and R. G. Gosine, "Moving Object Detection in Indoor Environments Using Laser Range Data", *in proceedings of Intelligent Vehicles Symposium, 2007 IEEE Digital Object Identifier*, pp 969 - 974.

[2] R. Stahn and G. Heiserich1 and A. Stopp, "Laser Scanner-Based Navigation for Commercial Vehicles", *in proceedings of International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ*, pp 802 - 807.

[3] R.B. Rusu and N. Blodow and Z.C. Marton and M. Beetz, "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments", *in proceedings of International Conference on Intelligent Robots and Systems 2009 (IROS '09)*, pp 1-6.

[4] A. Nuechter and H. Surmann and J. Hertzberg, "Automatic model refinement for 3D reconstruction with mobile robots", *in proceedings of Fourth International Conference on 3D Digital Imaging and Modeling 2003 (3DIM '03)*, pp 394-401.

[5] R. Schnabel and R. Wahl and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection", *in Computer Graphics Forum*, vol. 26, 2007, pp 214-226.

[6] R. Schnabel and R. Wessel and R. Wahl and R. Klein, "Shape Recognition in 3D Point-Clouds", *in proceedings of 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2008 (WSCG '08)*.

[7] M.Y. Yang and W. Foerstner, "Plane Detection in Point Cloud Data", *Technical Report*, Department of Photogrammetry, University of Bonn, 2010

[8] D. Haehnel and W. Burgard and S. Thrun, "Learning compact 3D models of indoor and outdoor environments", *in Robotics and Autonomous Systems*, vol. 44, pp 15-27, 2003.

[9] H. Surmann and K. Lingemann and A. Nuechter and J. Hertzberg, "A 3D laser range finder for autonomous mobile robots", 2001.

[10] A. Sabov and J. Kruger, "Segmentation of 3D points from range camera data using scanlines", *in proceedings of 15th International Conference on Systems, Signals and Image Processing 2008 (IWSSIP 2008)*, pp 429-432.

[11] C. Ye , "Navigating a Mobile Robot by a Traversability Field Histogram", *in IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* , vol.37, no.2, pp.361-372, April 2007.

[12] R. Valencia and E.H. Teniente and E. Trulls and J. Andrade-Cetto, "3D mapping for urban service robots," *in IEEE/RSJ International Conference on Intelligent Robots and Systems 2009 (IROS '09)*, pp.3076-3081, 10-15 Oct. 2009.

[13] K. Regenstein and T. Kerscher and C. Birkenhofer et al., "Universal Controller Module (UCoM) - component of a modular concept in robotic systems", *in proceedings of 2007 IEEE International Symposium on Industrial Electronics, 2007*, pp 2089 - 2094.

[14] M. Strand and R. Dillmann, "Using an attributed 2D-grid for next-best-view planning on 3D environment data for an autonomous robot", *in proceedings International Conference on Information and Automation, 2008. ICIA 2008*, pp 314 - 319.

[15] J. Perret, "Service robots for nuclear safety: new developments by CYBERNETIX", *in proceedings IEEE International Conference on Robotics and Automation, 1998*, vol. 3, pp 2106 - 2109.

[16] KHG Kerntechnische Hilfsdienst GmbH, German company active in the field of nuclear safety, *www.khgmbh.de*.

[17] P. Cignoni and C. Montani and R. Scopigno, "DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed", *in Computer-Aided Design*, vol. 30, 1998, pp 333-341.