

Furniture Free Mapping using 3D Lidars

Zhenpeng He, Jiawei Hou and Sören Schwertfeger

School of Information Science and Technology

ShanghaiTech University

Shanghai, China

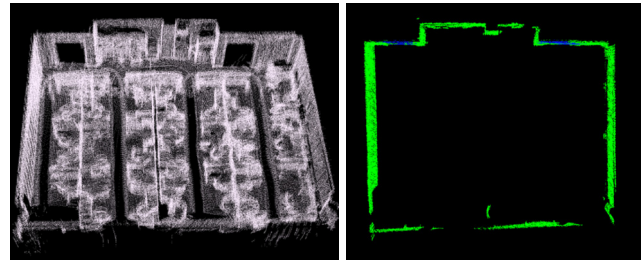
{hezhp, houjw, soerensch}@shanghaitech.edu.cn

Abstract—Mobile robots depend on maps for localization, planning, and other applications. In indoor scenarios, there is often lots of clutter present, such as chairs, tables, other furniture, or plants. While mapping this clutter is important for certain applications, for example navigation, maps that represent just the immobile parts of the environment, i.e. walls, are needed for other applications, like room segmentation or long-term localization. In literature, approaches can be found that use a complete point cloud to remove the furniture in the room and generate a furniture free map. In contrast, we propose a Simultaneous Localization And Mapping (SLAM)-based mobile laser scanning solution. The robot uses an orthogonal pair of Lidars. The horizontal scanner aims to estimate the robot position, whereas the vertical scanner generates the furniture free map. There are three steps in our method: point cloud rearrangement, wall plane detection and semantic labeling. In the experiment, we evaluate the efficiency of removing furniture in a typical indoor environment. We get 99.60% precision in keeping the wall in the 3D result, which shows that our algorithm can remove most of the furniture in the environment. Furthermore, we introduce the application of 2D furniture free mapping for room segmentation.

Index Terms—Furniture Free Map, SLAM, Semantic, Real-time.

I. INTRODUCTION

Indoor maps of buildings are critical in many robotics application such as navigation [1], localization [2], planning [3], room segmentation [4], [5] or topological representations [6]. One difficult challenge in room segmentation is the over-segmentation caused by clutter in the generated map. Some topological maps are created based on segmented maps, which can be used to reduce the computational complexity of large scenarios. In indoor environments, objects like furniture frequently occlude the wall surfaces, which puts more challenges on segmentation. There are some methods to deal with cluttered and occluded environments. Adan et al. [7] used a histogram of the point cloud to detect the wall and generate the furniture free map. Babacan et al. [8] introduced a Convolutional Neural Network (CNN) to filter out the noise in the environment. Mura et al. [9] proposed an efficient occlusion-aware process to extract planar patches as candidate walls, separating them from clutter and coping with missing data. These static solutions need a complete point cloud. Nevertheless, real-time strategies are increasingly required in this field.



(a) One room point cloud with lots of work cubicles. (b) Point cloud of the same room with only walls (green) and doors (blue).

Fig. 1: Furniture free mapping example.

In this study, we propose a real-time method for a mobile robot which has two 3D Lidar (Light Detection And Ranging) sensors: a horizontally scanning and a vertically scanning laser sensor. The robot additionally has nine color cameras, an IMU, and odometry. All sensors are fully synchronized [10] and calibrated [11]. The horizontal scanner is used for Simultaneous Localization And Mapping (SLAM), from which we utilize only the pose information of the robot, while the vertical sensor is key to 3D mapping and our semantic labeling algorithm. There are three steps in our approach, which are all applied to the vertical scanner: 1) point cloud rearrangement, 2) wall plane detection, and 3) semantic label reasoning. In the first step, we remove the ceiling and floor of each frame of the point cloud. Then we detect the wall candidate elements and use a growing algorithm to compute the wall plane coefficient. In the final step, a set of reasoning chains is used to label each point as the wall, ceiling, and others. According to the semantic labels, we can remove the movable objects and only keep the wall points in the result. It is important to mention that an explicit detection of openings (doors) is needed to create 2D maps that correctly represent the connections (doors) between the environments. The pipeline process for our proposed method is shown in Fig. 2.

Our wall plane detection method is based on the assumption that walls are planar and perpendicular to the ground. Artificial indoor scenes often exhibit repeated regularity, especially in the vertical direction. So we also believe that

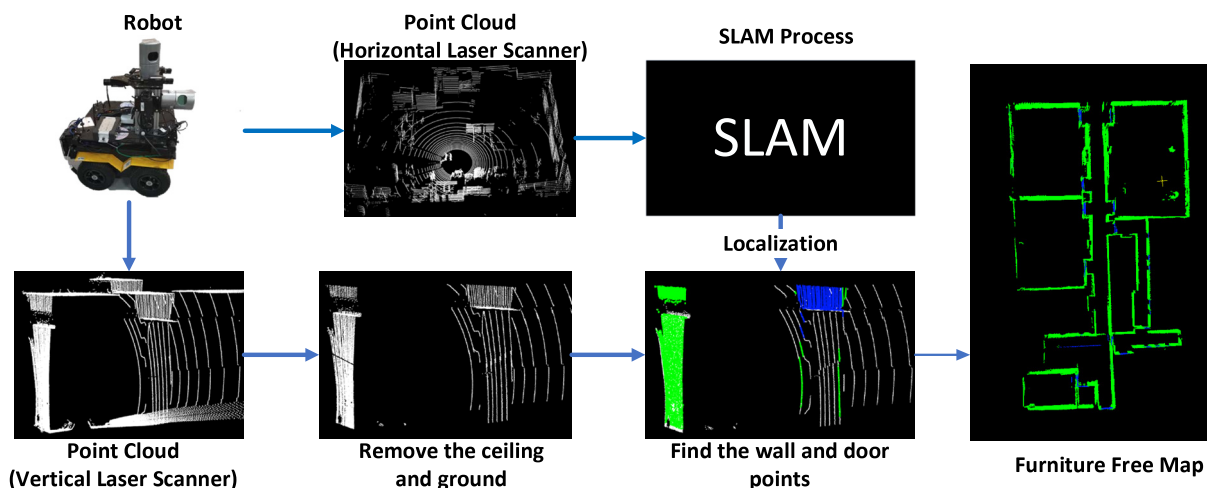


Fig. 2: Pipeline of the proposed algorithm.

identifying semantic labels in the indoor building is mainly a detection problem, rather than a common segmentation problem.

The outputs of our method are a 3D semantic point cloud and a 2D furniture free grid map. The experiment shows that our method has a high precision for finding the wall plane. We also get good performance in noise removal, which is shown in Fig. 1. The first experiment shows the semantic classification accuracy of the 3D point cloud. The second experiment compares the results of the furniture free mapping with a hand-made 2D clean map. The third experiment demonstrates our approach in a 2D room segmentation application.

The main contributions of this paper are as follows:

- An algorithm for large-scale indoor point cloud classification.
- Derived from that a method to generate furniture free 2D grid maps.
- Experimental analysis of the algorithm and application to room segmentation.

II. RELATED WORK

A. Indoor Modeling and Reconstruction

Furniture free mapping is quite related to indoor modeling and reconstruction, because it basically generates the floor plan of a building. Methods for creating such models can be divided into two types: those that use one complete point cloud of the environment as the input and those that work with individual laser scans, typically in real time. Those models are usually higher-order representations of the environment, like planes, meshes or lines.

Real time mapping is most often embedded in SLAM. Chen et al. [12] are using a vertical laser scanner to build the model with a SLAM process. [13] applied a mobile phone

and a set of fixed actions to produce a 2D floor plan and a representative 3D model. Kim et al. [14] used the hand-held depth camera and allowed the user to freely move through a building to capture its important architectural elements. This algorithm needs to be initialized by pointing the sensor to a corner. Turner et al. [15] triangulated sampling points on a 2D planar map and separated these triangles into the interior and exterior sets. The exterior sets are outside of the building, so only the interior sets are kept to create the model.

Creating models from big, static point clouds is well researched. The approach in [16] utilized random sample consensus (RANSAC) and plane normal orientations to extract structures of the building, such as floor, ceiling, and walls. The study of Babacan [17] used minimum description length (MDL) hypothesis ranking to extract the floor blueprint. Oesau [18] presented an energy minimization function for different segment rooms in 2d maps. Okorn [19] used a point histogram and an iterative Hough transform algorithm to generate the map.

III. METHOD

In our method, the goal is to distinguish the points of the wall, doors, and furniture. These points belong to different regular and irregular components in the point cloud. The regular components in the indoor point cloud can be divided into two geometrically significant clusters, according to the distribution of surface normals: horizontal structure and vertical structure. The horizontal structure includes mainly the ceiling and the floor. As mentioned above, we are using the vertical laser scanner of the robot for this part, so typically, there will always be a ceiling and floor element to be found. The vertical structure consists primarily of walls. Irregular components have no apparent geometric distribution. Our pipeline to classify these mixed data is to start by extracting

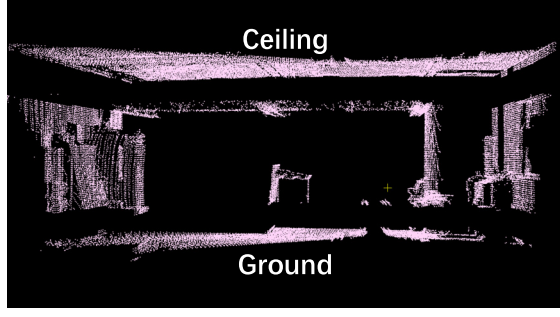


Fig. 3: Point cloud from the vertical laser scanner after removing the ceiling and floor.

the ceiling and floor from the horizontal structure, and further determine wall planes from the remaining cloud. Semantic classification is then conducted according to the relative pose between the object and the wall, as shown in Fig. 2.

A. Point Cloud Rearrangement

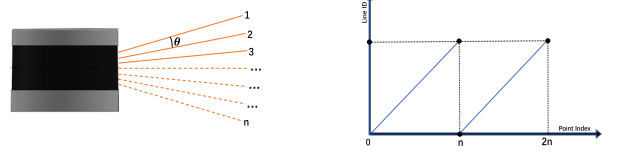
Firstly, we want to classify the position of the floor and ceiling. On the one hand, our algorithm is running on a ground robot so that we can straightforward locate where the floor is. We use a height filter to remove the floor from each frame of the scanner. We also assume that the ceiling is parallel to the floor. We use a Random Sample Consensus (RANSAC) algorithm to extract the biggest plane whose surface normal is orthogonal to the floor. As shown in Fig. 3, the remaining point cloud is mainly consisting of vertical structures after removing the ceiling and floor. We divide the point cloud in the vertical direction to many lines, as shown in Fig. 5a. The 3D laser scanner is composed of discrete laser emitters (32 for our robot). We extract the points from each emitter and then get an ordered point cloud. Fig. 4a shows the layout of the beams of the laser scanner. There is a fixed angular resolution in the point cloud. We sort these points based on their angle value a to the position of the laser scanner. Let $\mathcal{P} = \{p_i, i \in \mathcal{N}\}$ be the set of points in a point cloud, which is from one frame of the vertical laser scanner. (x_i, y_i, z_i) are the measured coordinates of p_i . Then the angle value a is computed by:

$$a = \arctan \frac{\sqrt{x_i^2 + z_i^2}}{y_i} \quad (1)$$

Points that have the same angle a are categorized into the same line. There are n lines of the laser scanner, such that we can divide each frame of point cloud to $2n$ vertical point lines in Fig 4b. To speed up the further computations we then, based on the angle, subsample the point lines to have exactly 200 points.

B. Wall Plane Detection

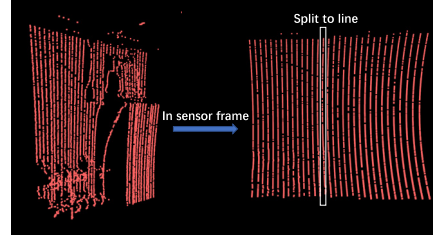
In this step, we extract wall planes from point lines $\mathcal{L} = \{l_i, i \in \mathcal{N}\}$. Firstly, we find the wall candidate elements



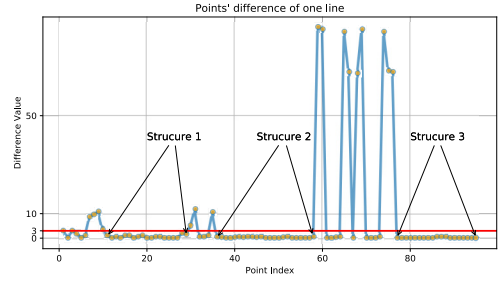
(a) Beam layout of the laser scanner.

(b) Return mode.

Fig. 4: Beams of the 3D laser sensor.



(a) Extraction of a point line from the point cloud of one scan.



(b) Forward difference of one point line l_i . The red line is threshold. The bottom is on the left at index 0 and the top on the right.

Fig. 5: Point line computation.

in those point lines. There are both noise and vertical structures in a line. We compute the forward difference of adjacent points along the line to determine vertical structures. Let $p_j, p_{j-1} \in l_i$ be the neighboring points. The forward difference d is computed by:

$$d = \left| \frac{\sqrt{x_j^2 + y_j^2} - \sqrt{x_{j-1}^2 + y_{j-1}^2}}{z_j - z_{j-1}} \right| \quad (2)$$

We show the forward difference of one point line l_i in Fig. 5b. It is the difference in the measured distance along the Z axis. We apply a *threshold* (shown as a red line) to separate planar structures of a certain minimum number of points (10 in these experiments). In the example, we can find three vertical structures in the line. Next, we need to choose the vertical structure that is most likely to be the wall. We assume that the highest detected structure is more likely to be the wall. This assumption is based on architectural design

and space using practices: Typically, furniture does not go all the way up to the ceiling, so the top planar part should be the wall. If the furniture goes all the way to the ceiling, our algorithm will classify it as a wall. We think this is OK, because a cabinet that is that high is very unlikely to be moved. Fig. 6a shows the highest detected vertical structure in green and, for comparison, Fig. 6b highlights the bottom-most detected structure. We can see that Fig. 6a with the upper vertical structure shows a good representation of the wall. So we keep the upper detected vertical structure to be the wall candidate elements.

Afterwards, the growing method Alg. 1 is used to find the wall planes, which consist of many point lines. We pick a wall candidate as a seed and perform an iterative search for wall candidate elements that are coplanar with it, as shown in Fig. 7. In the end, we filter out the wall candidate elements which do not belong to the wall and calculate the wall plane parameters.

Algorithm 1 Wall Plane Detection

Input: Point lines set: \mathcal{L}

Output: The wall planes set \mathcal{W}

Data:

$\mathcal{W} \leftarrow \emptyset$

$\mathcal{L} \leftarrow \{l_i, i \in \mathcal{N}\}$

Current plane parameters $\mathcal{C} \leftarrow \emptyset$

Current plane $\mathcal{P} \leftarrow \emptyset$

Ransac function $\Omega(\cdot)$

Similarity function $\Theta(\cdot, \cdot)$

Similarity threshold σ_{th}

begin

for $i=1$ to $size(\mathcal{L})$ **do**

$l_i \leftarrow$ Wall candidate points p_i

end

for $i=1$ to $size(\mathcal{L})$ **do**

if $\mathcal{P} == \emptyset$ **then**

$\mathcal{P} \leftarrow l_i$

end

else if $\mathcal{C} == \emptyset$ **then**

$\mathcal{C} = \Omega(\{\mathcal{P}\})$

end

else if $\Theta(\mathcal{C}, l_i) < \sigma_{th}$ **then**

$\mathcal{P} \leftarrow l_i$

$\mathcal{C} = \Omega(\mathcal{P})$

end

else

$\mathcal{W} \leftarrow \{\mathcal{P}\}$

$\mathcal{P} \leftarrow \emptyset$

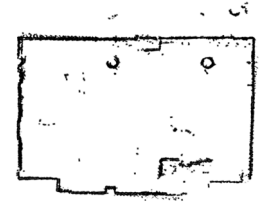
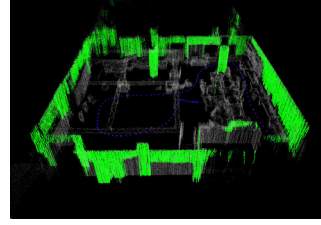
$\mathcal{C} \leftarrow \emptyset$

end

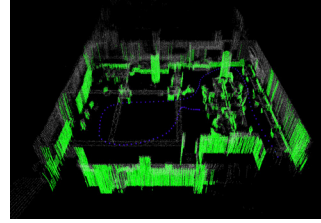
end

 Return \mathcal{W}

end



(a) Up: The upmost planar structure of point lines in a scan. The 3D point cloud generated from several scans on the left and the 2D map on the right.



(b) Down: The bottom-most planar structure of point lines in a scan.

Fig. 6: Planar structures in point lines and the corresponding 2D maps.

C. Semantic Labeling

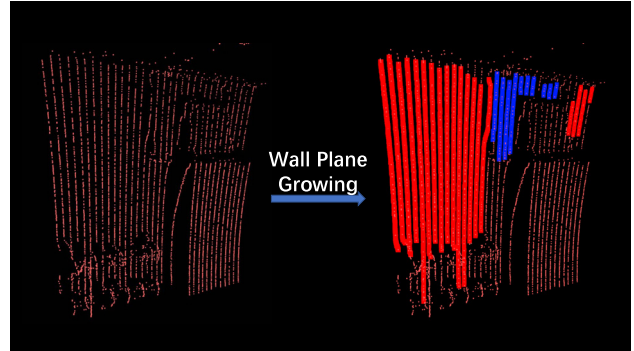


Fig. 7: Find wall plane form the wall candidate elements.

In the previous sections, we already labeled the ceiling, floor, and walls. The remaining points are labeled clutter. For robotics and mapping, doors are essential, as they are the passages between rooms, corridors, and other areas. Up to now, the wall segments above doors will just be classified as the wall. We are thus applying a dedicated door detection algorithm, that works for open and closed doors.

The algorithm uses rule-based context knowledge to find the door. Usually, closed doors are a few centimeters deeper than the wall plane. Our algorithm can only detect this kind of closed door in the wall. Open doors have a very high hollow and are connected to the floor, whereas a closed door is just a depression on the wall. The algorithm works on the point lines from Section III-A. In order for that line to be classified as being above a door, all of the following conditions have to be met:

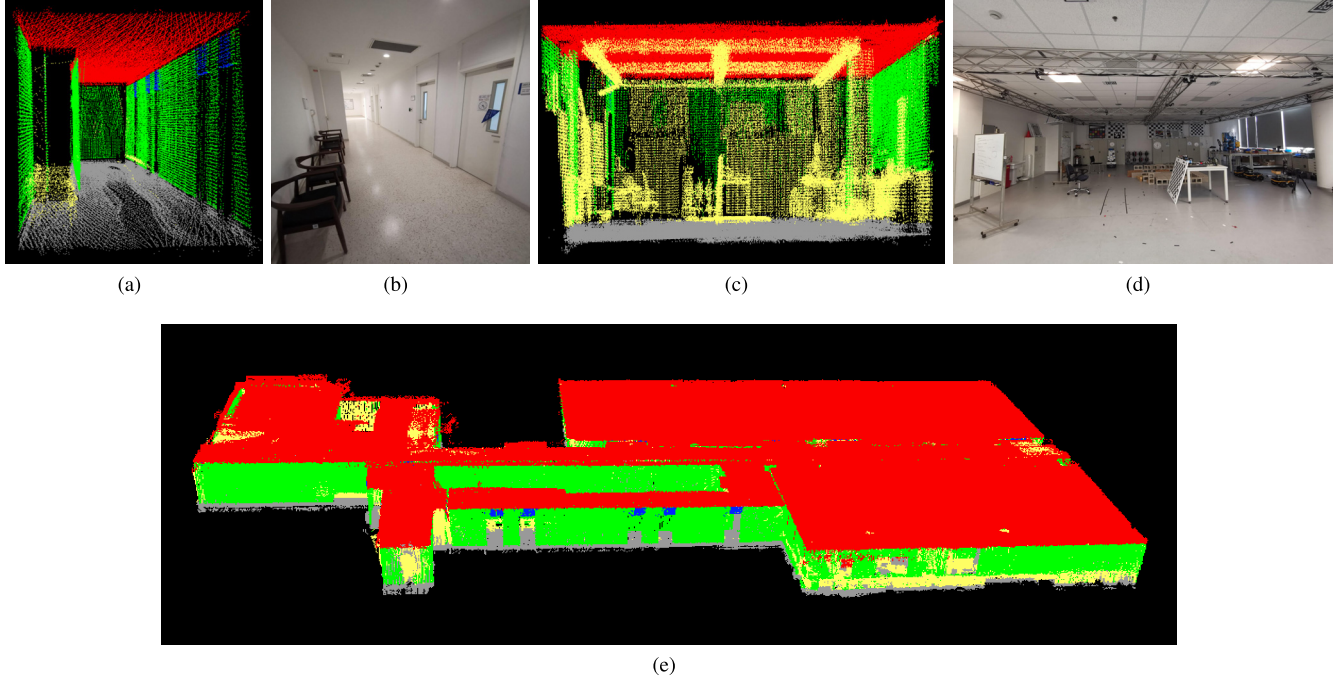


Fig. 8: (a) A corridor in our 3D result. (b) Real scene of the corridor. (c) Our robotics lab as 3D result. (d) Real scene of the lab. (e) 3D result of our process. All point clouds are the combined results from many individual, classified scans. Point color index: red: ceiling; gray: floor; green: wall; blue: door; yellow: other points.

- 1) The points from that line belong to a wall plane.
- 2) Going from top to bottom, from some point onwards, all points of that line are located behind the plane of 1).
- 3) This starting point from 2) has to be at least a minimum height above the ground.

Because our plane parameters are accurate enough, our method can still detect closed doors.

IV. EXPERIMENTS AND DISCUSSION

A. Setup

We evaluated our process in the robot with two Velodyne 32-line laser scanners. The robot mapping scenario includes several offices and corridors from the ShanghaiTech Automation and Robotics Center (STAR Center). Those are quite diverse spaces: cluttered offices, long and mostly empty hallways, a busy lab and a more empty lab. The dataset has closed and open doors. We cannot use publicly available datasets for comparison, as we rely on the vertical Lidar data, which other datasets don't have. Table I shows some statistics from the experiments. We show the run time to compute the wall plane and door points of our single-threaded ROS C++ implementation on an Intel 3.60GHz 8 core laptop. We also show the number of points in the raw point cloud and the total number of points in the point lines (without ceiling and floor) per scan.

TABLE I: Experiment Statistics

	average	std. err.	min	max
Run-time per cloud (ms)	74	26	32	187
#points in cloud	96,058	5,743	60,130	100,538
#points in lines	40,595	13,047	14,885	75,698

The labeling results for point clouds from this environment are shown in Fig. 8e. These segmented point clouds are labeled into the ceiling (red), wall (green), door (blue), floor (gray), and clutter/ furniture (yellow). We are using the BLAM [20] SLAM algorithm with the horizontal Velodyne to provide the poses of the scans. With this information, we can then transform all labeled point clouds from the vertical scanner into a common coordinate system and thus create big point clouds, which we show in Fig. 8e. By hand, we removed error points outside of the room, which are mainly a result of reflections of the laser beams on the windows. In the future we will do this automatically using this [21] approach.

B. Evaluation of 3D Point Cloud Semantic Labeling

We use the recall, precision, and F_1 -measure value to quantitatively evaluate the correctness and accuracy of our 3D labeling result. The precision value describes the percentage of true objects in detected result, and the recall describes the

percentage of true objects in all objects in the environment. The equations for recall, precision, and F_1 -measure are shown below:

$$\begin{aligned} recall &= \frac{TP}{TP + FN} \\ precision &= \frac{TP}{TP + FP} \\ F_1\text{-measure} &= \frac{2 * precision * recall}{precision + recall} \end{aligned} \quad (3)$$

Where FP, TP, FN are the number of false positives, true positives and false negatives, respectively. Because our algorithm is based on each frame of the point cloud, some data can be too noisy to detect the label. We use the detected area instead of the number of points to calculate these measurements. The quantitative evaluation is shown in Table II. The recall and precision values of the wall are 95.26% and 99.60%, respectively. There are only a few wrong labels in the 3D wall labeling result. However, our method cannot handle corners that are not fully scanned. There are also some problems with surfaces such as columns. For the ceiling, the recall and precision values are 85.25% and 99.92%, respectively. The precision value is not high enough because we are not dealing with the points that are reflected by the window. In the future we will remove these outside points using [21]. With a height filter, we can get a high precision on the floor data. This approach cannot handle the uneven ground. The recall and precision values for door detection are 80.76% and 82.82%, respectively. This is because some of the point lines are not vertical to the floor. In some situations those lines bend so that we mark the wall next to the door as a door. In this experiment there are 19 doors and we only missed one far away door in this scenario. So our approach works well to find out where the door is.

C. Evaluation of Furniture Free 2D Maps

One main part of our result is the 2D furniture free map. We show several different 2D maps in Fig. 9. Fig. 9a is a hand-made 2D map which comes from FARO. FARO is a high precision static 3D laser scanner with very low error. The FARO scans were taken when there was no furniture in the building, and all the doors were opened, so it is the perfect comparison. Fig. 9b is our result. We observe that some doors are not open, even though they were classified as open in the 3D map. This error comes from the fact that doors should be detected as doors from both sides if there are scans at both sides. This is something to improve in the future. Fig. 9c is obtained by slicing 50 cm to 60 cm under the ceiling. In this map most of the furniture is removed, but there is still much noise in the environment. At the same time, the doors are not detected. Fig. 9d is obtained by slicing the point cloud in the middle. It can keep the open doors open, but it is full of furniture, and closed doors are obstacles. In Fig. 9e we intuitively compare the 2D furniture free map with the

TABLE II: 3D semantic labelling result of our method.

Label	Label(area/ m^2)			Result		
	FP	TP	FN	Precision	Recall	F1-Measure
Ceiling	141.68	819.15	0.60	85.25	99.92	92.00
Floor	72.631	622.20	0.96	89.55	99.85	94.42
Wall	2.80	703.24	34.99	99.60	95.26	97.37
Door	3.77	18.17	4.33	82.82	80.76	81.77

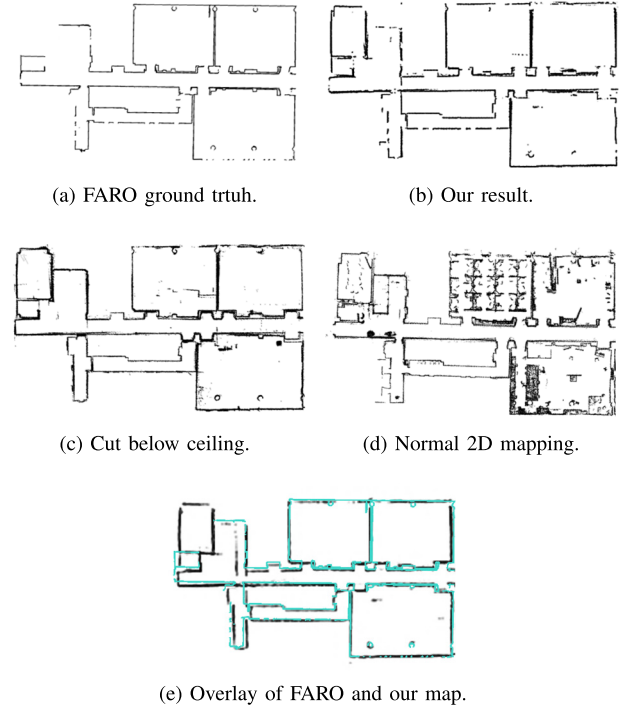


Fig. 9: Comparison of the different 2D grid maps of the STAR Center.

2D map, which comes from FARO. The overlaid map shows that the room location is not fully aligned. This is because we have some drift in the SLAM process, and the poses from the SLAM Lidar are not from the same time as the scans of the vertical scanner. In the future we will interpolate the vertical Lidar pose from the surrounding horizontal Lidar poses. Our map accuracy is completely dependent on the accuracy of the SLAM process.

D. Application to Map Segmentation

In [5] we presented a topological map representation based on areas, the Area Graph. Figure 10 shows the Area Graphs generated from the normal map with furniture (Fig. 9d) and the furniture free map generated by our algorithm (Fig. 9b). The parameter to segment the map is set according to the strategy described in [5] i.e., $w = 1.9m$. We used an alpha



Fig. 10: Room segmentation applied on (a) the normal map with furniture and (b) the furniture free map, where each color presents an area.

shape removal approach to remove noise for the map with furniture, while we did not need use it for the furniture free map. It is straightforward to observe that the Area Graph segmentation based on the normal map suffers from heavy over-segmentation. The segmentation on the furniture free map involves no over-segmentation of the rooms at all. We can also apply Area Graphs of furniture free maps in 2D map matching [22].

V. CONCLUSIONS

This paper presented an efficient approach to automatically generate furniture free maps of indoor buildings at real-time speed. Removing the furniture in the rooms allows for simple tasks such as segmentation, map matching, and long-term localization. We achieve good labeling, also for doors, which is important for applications like path planning or topology generation. The method relies on a dense enough point cloud with coverage from the floor to the ceiling. Some limitations of our algorithm are: first, we need a good SLAM process to get localization, because the accuracy of the furniture free map depends on the accuracy of the slam algorithm. Second, there are usually lots of windows in buildings. The reflection of the laser beams window is hard to deal with. So we are also working on detecting and removing those erroneous beams. We also need to generate the model of rooms to optimize our result and to improve the door representation of furniture 2D grid maps, for example, by explicitly opening a door as soon as it is detected from one side.

REFERENCES

- [1] Saurabh Taneja, Burcu Akinci, James H Garrett Jr, and Lucio Soibelman, "Algorithms for automated generation of navigation models from building information models to support indoor map-matching," *Automation in Construction*, vol. 61, pp. 24–41, 2016.
- [2] Kunjin Ryu, Lakshitha Dantanarayana, Tomonari Furukawa, and Gamini Dissanayake, "Grid-based scan-to-map matching for accurate 2d map building," *Advanced Robotics*, vol. 30, no. 7, pp. 431–448, 2016.
- [3] Paulo Pinheiro, Eleri Cardozo, Jacques Wainer, and Eric Rohmer, "Cleaning task planning for an autonomous robot in indoor places with multiples rooms," *International Journal of Machine Learning and Computing*, vol. 5, no. 2, pp. 86, 2015.
- [4] Richard Bormann, Florian Jordan, Wenzhe Li, Joshua Hampp, and Martin Hagele, "Room segmentation: Survey, implementation, and analysis," pp. 1019–1026, 2016.
- [5] Jiawei Hou, Yijun Yuan, and Sören Schwertfeger, "Area graph: Generation of topological maps using the voronoi diagram," in *2019 19th International Conference on Advanced Robotics (ICAR)*, 2019.
- [6] Sören Schwertfeger and Andreas Birk, "Map evaluation using matched topology graphs," *Autonomous Robots*, p. 1–27, 2015.
- [7] Antonio Adan and Daniel Huber, "3d reconstruction of interior wall surfaces under occlusion and clutter," in *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE, 2011, pp. 275–281.
- [8] K Babacan, L Chen, and G Sohn, "Semantic segmentation of indoor point clouds using convolutional neural network," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.
- [9] Claudio Mura, Oliver Mattauch, Alberto Jaspe Villanueva, Enrico Gobbetti, and Renato Pajarola, "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts," *Computers & Graphics*, vol. 44, pp. 20–32, 2014.
- [10] Hongyu Chen, Xiting Zhao, Jianwen Luo, Zhijie Yang, Zehao Zhao, Haochuan Wan, Xiaoya Ye, Guangyuan Weng, Zhenpeng He, Tian Dong, and Sören Schwertfeger, "Towards generation and evaluation of comprehensive mapping robot datasets," in *Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR, 2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2019, IEEE Press.
- [11] Hongyu Chen and Sören Schwertfeger, "Heterogeneous multi-sensor calibration based on graph optimization," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2019, IEEE.
- [12] Jingdao Chen and Yong K Cho, "Real-time 3d mobile mapping for the built environment," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*. Vilnius Gediminas Technical University, Department of Construction Economics, 2016, vol. 33, p. 1.
- [13] Giovanni Pintore and Enrico Gobbetti, "Effective mobile mapping of multi-room indoor structures," *The visual computer*, vol. 30, no. 6–8, pp. 707–716, 2014.
- [14] Young Min Kim, Jennifer Dolson, Mike Sokolsky, Vladlen Koltun, and Sebastian Thrun, "Interactive acquisition of residential floor plans," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3055–3062.
- [15] Eric Turner and Avidesh Zakhori, "Floor plan generation and room labeling of indoor environments from laser range data," in *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*. IEEE, 2014, pp. 1–12.
- [16] Victor Sanchez and Avidesh Zakhori, "Planar 3d modeling of building interiors from point cloud data," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 1777–1780.
- [17] K Babacan, J Jung, A Wichmann, BA Jahromi, M Shahbazi, G Sohn, and M Kada, "Towards object driven floor plan extraction from laser point cloud," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, 2016.
- [18] Sven Oesau, Florent Lafarge, and Pierre Alliez, "Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 90, pp. 68–82, 2014.
- [19] Brian Okorn, Xuehan Xiong, Burcu Akinci, and Daniel Huber, "Toward automated modeling of floor plans," in *Proceedings of the symposium on 3D data processing, visualization and transmission*, 2010, vol. 2.
- [20] Erik Nelson, "Blam," <https://github.com/erik-nelson/blam>.
- [21] Xiting Zhao, Zhijie Yang, and Sören Schwertfeger, "Mapping with reflection-detection and utilization of reflection in 3d lidar scans," *arXiv preprint arXiv:1909.12483*, 2019.
- [22] Jiawei Hou, Haoqi Kuang, and Sören Schwertfeger, "Fast 2d map matching based on area graphs," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, IEEE.