Artificial Life and Robotics (2023) 28:314–322
https://doi.org/10.1007/s10015-022-00834-y

ORIGINAL ARTICLE

Point cloud data map creation from factory design drawing for LiDAR localization of an autonomous mobile robot

Ryutaro Kaneko[1] · Yuji Nakamura[2] · Ryosuke Morita[2] · Satoshi Ito[2]

Received: 4 April 2022 / Accepted: 17 November 2022 / Published online: 17 December 2022

Abstract

When using an autonomous mobile robot, an environmental map should be created in advance. In this study, we propose a method for creating a point cloud data (PCD) map required for LiDAR localization in autonomous driving. The proposed method creates PCD maps from paper design drawings. For objects not depicted in the drawings, we introduce a tablet-scan, whose data are merged into the map created from the drawings. Three factors affecting the accuracy of self-localization are investigated during the map creation: the gap size in the PCD map, presence of the tablet-scan data, and random point alignment during the map creation. The effects of these factors on the localization accuracy are evaluated via simulations using actual scan data. Consequently, the existence of the optimal gap size and the accuracy enhancement using both the tablet-scan data and random point alignment are clarified. Moreover, autonomous driving using the PCD map created using the proposed method is successfully achieved.

Keywords Autonomous mobile robot · LiDAR localization · Map creation · Point cloud data

1 Introduction

In factories, assembly processes necessitate the transportation of products and their assembling parts. For this purpose, automated guided vehicles (AGVs) are employed to reduce human resources. "Guides," such as mechanical rails or lines drawn on the floor of factories, are mainly used to show AGVs the way to their destinations. These guides are effective tools for limiting AGVs in their accessible areas. In addition, they facilitate positional recognition and collision reduction.

However, factories are needed to be rebuilt to reinstall these guides. In factories manufacturing various products, the production lines are rebuilt every time the products are updated. Job-shop-type factories have to rearrange their layouts frequently. Therefore, depending upon the factory, there are cases when AGVs with guide equipment cannot be easily adopted, because the reinstall of the guides requires lots of time and costs.

Given the above background, we aim to develop an autonomous mobile robot (AMR) that does not require additional factory rebuilding for guides. We introduce an autonomous driving technology called LiDAR localization to achieve navigation without guide equipment. The three-dimensional (3D) LiDAR localization we implement can certainly detect the current AMR position without any factory rebuilding. However, it requires the latest layout information (i.e., a "map") for the localization instead. Particularly, our targets include factories whose layouts are frequently updated. Thus, obtaining the newest maps for

This work was presented in the joint symposium 27th International Symposium on Artificial Life and Robotics, 7th International Symposium on BioComplexity, and 5th International Symposium on Swarm Behavior and Bio-Inspired Robotics (Online, January 25–27, 2022).

✉ Satoshi Ito
satoshi@gifu-u.ac.jp

Ryutaro Kaneko
z4525025@edu.gifu-u.ac.jp

Yuji Nakamura
aa300007@edu.gifu-u.ac.jp

Ryosuke Morita
rmorita@gifu-u.ac.jp

1 Graduate School of Natural Science and Technology, Gifu University, Tokai National Higher Education and Research System, Yanagido 1-1, Gifu 501-1193, Japan

2 Faculty of Engineering, Gifu University, Tokai National Higher Education and Research System, Yanagido 1-1, Gifu 501-1193, Japan

rearranged factory layouts becomes a problem. Actually, the maintenance of the correct map in a variable environment, i.e., the mapping technology, is crucial in the practical application of self-localization [9, 10, 12]

In the LiDAR localization system that an AMR is equipped with, the 3D maps have a data format called point cloud data (PCD). Simultaneous localization and mapping (SLAM) is effective for map creation. However, enhancing the map accuracy (e.g., by loop-closure detection or using sensors such as odometers or inertial measurement units) requires enormous computation. Compared with SLAM, other methods [7, 13] create 3D PCD maps (we call them PCD maps in this study) in advance using techniques such as manual driving. These PCD maps created in advance require lower computational costs and have better real-time properties for the localization than SLAM.

In this study, we develop a PCD map creation system whose input is a design drawing given as a paper and output is the PCD map of the inside of a building. We simulate the autonomous movement of an AMR using the created PCD map to demonstrate the effectiveness of the developed system. The remainder of this article is organized as follows. Section 2 describes a method for creating a PCD map from design drawings. Scans by a tablet computer with LiDAR are introduced in Sect. 3 to cope with the environmental changes inside buildings. Section 4 demonstrates automatic driving using a PCD map from design drawings by an actual AMR. In Sect. 5, we examine how the accuracy evaluated by the transformation probability (TP) depends upon parameters, such as the gap size or tablet-scan data. Finally, we conclude this study in Sect. 6.

2 PCD map creation from design drawings

2.1 PCD format

A PCD map is written in the PCD format [3]. This format comprises a 3D-coordinate data part and a unique header part. Using LiDAR, a PCD file is automatically generated by the software provided by its developer in many cases. The file we wrote ourselves can also be recognized as a PCD file provided it matches the PCD format, including the proper header part. This means that we can intentionally place point (object) data at any position we desire in a PCD map. On this basis, we create a PCD map (i.e., a PCD format file) by computing wall positions from two-dimensional (2D) paper design drawings without a LiDAR-scan. Then, we make the most use of the geographical information in the drawings by electronically scanning them.

2.2 Problem

Four problems are encountered in creating a 3D PCD map from 2D paper design drawings.

1. Lack of height information
   3D LiDARs can detect walls as 3D objects in buildings, because they can get the height information (Z-axis coordinates) using several lasers directed at different heights. Accordingly, the PCD file should have 3D data. By contrast, design drawings are usually 2D and do not have height information.
2. Scale unfitness
   The scales of design drawings differ one by one. Thus, we must consider the map scale to generate the wall (object) data.
3. Point cloud density
   In PCD maps, an object is expressed as numerous point data (a cloud of points) on the surface of the object. If the points are too dense, the computational cost increases. Meanwhile, if they are too sparse, the accuracy of the self-localization decreases. The density of the point cloud and to what extent the points in PCD maps should be upsampled (or downsampled) are critical issues.
4. Objects not depicted in drawings
   Design drawings include only constructional information. However, in real buildings, there may be some types of furniture in houses or some mechanical tools and inspection equipment composing production lines in factories. The presence or absence of such information affects the self-localization accuracy.

2.3 Approaches to solving the problems

We consider the following approaches to solving the aforementioned problems.

1. Point accumulation to the height direction
   The main place we aim to run an AMR is the inside of buildings, where the distance that the LiDAR detects is short compared with that outside of buildings. This implies that most of the laser reflects on the walls before it reaches the ceiling. Thus, in the PCD map creation, we accumulate points in the height direction to a sufficient height regardless of the situation.
2. Scale adjustment
   We scale the PCD map manually at the final map creation stage according to the description on the design drawings or the actual measurements of a building.
3. Upsampling or downsampling

The voxel grid filter (VGF) [6] is applied to downsample points when the PCD map is created from 2D paper design drawings. Conversely, upsampling is executed when enlarging the scale.

4. Tablet-scan

   We use a LiDAR-mounted tablet computer to scan objects that are not in the drawings. The scan data are converted to the PCD format. Finally, we merge them with the PCD map created from the drawings.

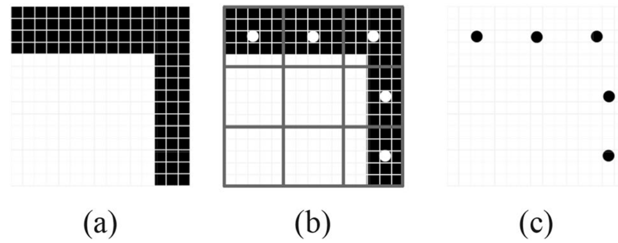


Fig. 2 VGF in 2D space

2.4 Upsampling and downsampling

2.4.1 VGF

Autonomous driving requires real-time processing for localization. Numerous PCD points reduce the processing speed. In this regard, downsampling is an advantageous means for reducing computational costs.

VGF [6] downsamples points in a PCD map. It divides a space comprising several points in Fig. 1a into several voxels (Fig. 1b). Then, it calculates the centroid in the voxels as representative points and reconfigures the points (Fig. 1c).

VGF is generally applied for 3D PCD. However, we apply it to 2D drawings at the early stage before creating 3D PCD. Therefore, we divide the drawings as pixel units and not as voxel units. For instance, Fig. 2a is divided into several desired size cells (e.g., the side length (gap size) is 5 pixels in this example; Fig. 2b). Then, the centroid of black pixels in each cell is calculated, and all points in a cell are replaced with one representative point at this centroid position. As a result, a 2D image with downsampled points is obtained (Fig. 2c).

2.4.2 Upsampling

After applying the 2D VGF, the output 2D points should be rescaled to adjust them to the actual size. This operation usually involves enlargement, which simultaneously expands the gap size. Sometimes, the gap size is larger than desired for the LiDAR localization. Thus, upsampling is required to increase the density of points during the rescaling process.

Now, all gap sizes between the PCD points are almost the same due to the VGF. Given this advantage, points can be easily interpolated to reduce the gap size to the desired value. Finally, the points are accumulated in the height direction, and the header part is added to complete the PCD map.

2.4.3 Example of the system

We selected a certain facility at Gifu University as the experimental site for autonomous driving. The PCD map in Fig. 3 was created from its design drawing. First, the map was created without VGF. It was downsampled in equal intervals, i.e., by picking up the rows and columns of every constant value from the pixel data. Although autonomous driving was achieved, the self-localization sometimes provided the wrong information because of the matching error between the LiDAR-scan data and the created map. Investigating the map, we found that the points' distribution was uneven: dense in some places and sparse in others. In addition, there were some blanks at the position where the wall should have existed, as shown in the enlarged map in Fig. 3 (right). We considered that this must be the main reason for the matching errors. Then, we applied VGF. The cell side length was set to 0.2 m when applying the 2D VGF. Owing to the VGF effect, the gap became constant and the walls were connected without blanks (Fig. 4). This effectively reduced the matching error between the scan data and the created PCD map during autonomous driving.

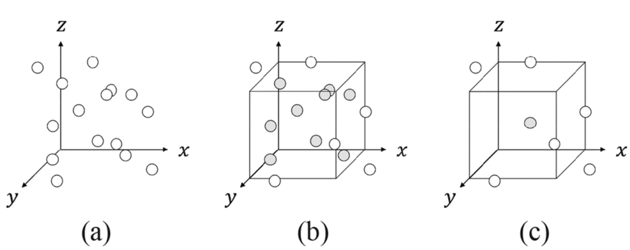


Fig. 1 VGF algorithm

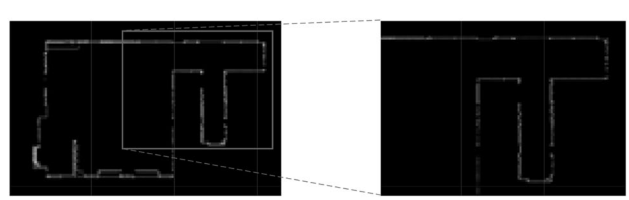


Fig. 3 PCD map downsampled in constant intervals

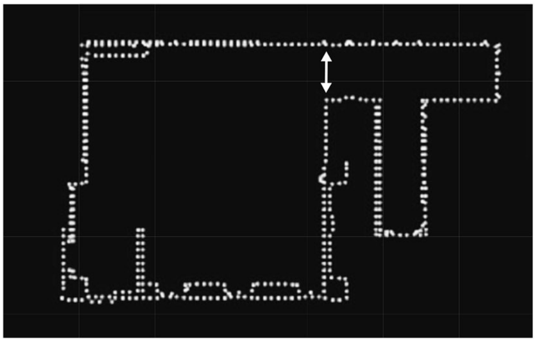


Fig. 4 PCD map downsampled by the 2D VGF

2.5 Outline of the application software

We developed a software application that enabled us to sequentially apply the aforementioned processes. The processes that users must perform are as follows.

1. Scanning factory design drawing data
   A design drawing (we assume it is given as a paper) must be converted to an image data format (jpg., png., etc.) using an electronic device such as a scanner or a camera.
2. Preprocessing including binarization
   After binarization, black pixels are regarded as walls; they are stored in an array (i.e., 2D point data) and used to create a 3D point cloud based on its geographical information. This is why, figures other than the wall (e.g., dimension lines) should be deleted in the preprocessing using image editors.
3. Running the application
   Upon starting our application, the file where the data generated in Process 2 are saved must be designated. Then, the processes described in Sect. 2.4.1 are executed with some necessary parameter assignments. This process outputs a temporary PCD map.
4. Checking the scale of the temporary PCD map
   The end of Process 3 automatically starts a PCD viewer, which allows us to evaluate the distance between certain points on the current PCD map. The temporary PCD map created in Process 3 has the correct geographical relation between walls, but the scale is not correct from the actual buildings. This is because the image scanned in Process 1 depends on not only the scale of the original drawing but also the scanner resolution. The distances on the PCD map are evaluated between the points where the actual lengths are known.
5. Rescaling
   The end of Process 4 automatically opens a pop-up window where the scale parameter can be input. The scale parameter should be obtained from the scale description on the design drawings or the actual measurement of a part of the site. Actually, in Fig. 4, the corridor width denoted by the arrow was measured for rescaling the PCD map. Afterward, the rescaling and upsampling described in Fig. 4 are performed. Finally, the desired PCD map is output. A few seconds is required to obtain the PCD map from the design drawings. Python was used as the programming language and included an image processing library, OpenCV.

3 Layout change process

We can obtain the PCD map from the design drawings so far. However, the PCD map does not contain some objects that are not depicted in the design drawings, such as furniture, containers, and mechanical tools. If these objects are not reflected in the PCD map, the LiDAR localization refers to data different from the current environment in practical driving scenarios, resulting in critical matching errors.

This problem might be solved by adding the missing objects to the design drawings and then employing the proposed map creation system. However, the objects have various heights unlike the wall and the proposed method cannot handle them. Moreover, as is often done, if we scan the inside of a building using an AMR with LiDAR, we can create a map that includes all objects. Nevertheless, our design policy avoids this method, because it is not easy for factory workers to operate AMRs for map creation.

In this study, we propose another method, "tablet scan," where a LiDAR-mounted tablet is used to scan the missing objects. As the result of the tablet-scan, the real-scale PCD of the object is obtained. Tablets are easier to handle because of their user-friendly interface than LiDAR alone on an AMR. Many people might already have tablets or consider them worth purchasing, because they work as computers. Afterward, the obtained data are merged with the PCD map created using the proposed system. This merging process is still a manual task on the software "Cloud Compare" [2] placing the scan data using a mouse. It is important to scan together the extra "key frame" such as the walls that surely exist in the design drawings. This key frame allows us to place the object data in an accurate position easily.

Figure 5a depicts the PCD map where the stairway data shown in Fig. 5c scanned by the tablet are merged with the PCD map we created from the design drawing. Figure 5b shows a picture of the stairway. In fact, the original design drawing contains the stairway, but the space that the stairway occupies in the building does not have a constant height (i.e., there is an empty space below or above the stairway). A special process that our map creation system cannot perform

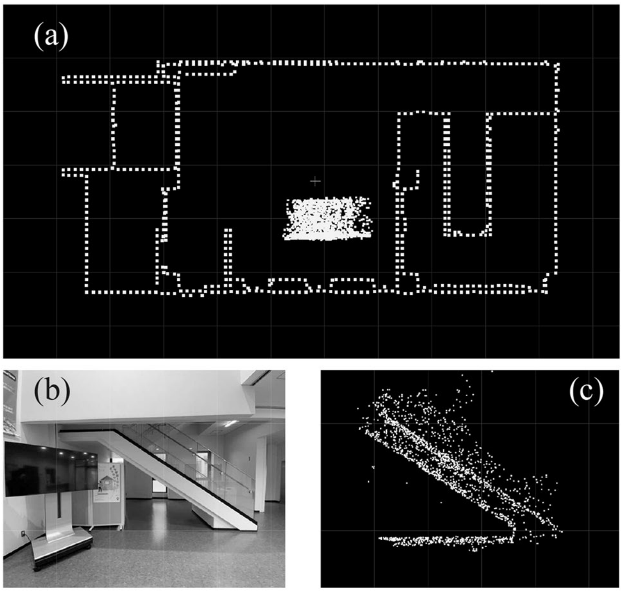


Fig. 5 PCD map where an obstacle PCD is overlaid

Fig. 6 Tablet-scan. left: Tablet used in the experiment. Right: a scene of a tablet-scan

is required to reflect this space in the PCD map. Thus, the tablet-scan method is used to express such a spatial property correctly.

Figure 6 depicts the tablet and a scene of the tablet-scan. An iPad Pro (Apple Inc.) was used for the tablet-scan.

4 Autonomous movement experiment

We experimented to verify whether the PCD map created using the proposed system allows an AMR to travel autonomously executing self-localization.

Figure 7 shows the AMR used in the experiment. It mounts a 3D LiDAR (Velodyne-16: Velodyne LiDAR

Fig. 7 AMR used in the experiment

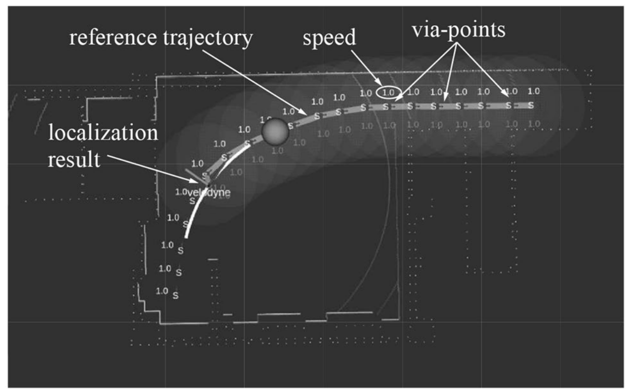


Fig. 8 Reference trajectory (a bold dark line) and localization result on the created PCD map. About 20 plotted via points designate the reference trajectory by interpolating them. The numbers near the via points denote the speed there

[5]). Autoware [1] works as a control system in the Robot Operation System [4]. Normal distribution transform (NDT) matching was operated as the self-localization algorithm [11].

Figure 8 depicts the LiDAR-scan data and AMR's self-position. The reference trajectory, which was planned on the PCD map created by the proposed system, is also depicted. As shown in every-second snapshots in Fig. 9, though the experiment differs from that in Fig. 8, the map created from the drawing enabled an AMR to self-localize and drive autonomously.

In the future, the proposed map creation method should be tested at different places and improved further to ensure a robust map-creating process.

5 Accuracy verification

5.1 TP

In this section, we verify the self-localization accuracy to prove the effectiveness of the proposed system . We adopted TP for the assessment [8]. TP is an index of scan matching

Fig. 9 Snapshots of the AMR movement

that expresses the matching level between a reference PCD map (environment PCD map) and the current LiDAR-scan data. When the TP value is zero, the matching is unstable: the self-localization does not converge to a constant value. According to the reference [8], a TP value greater than 2 is preferable for sufficiently stable computation.

5.2 Method

We can specify the gap size between points when making PCD maps. In Fig. 5, for example, the gap size of the PCD map is selectable in the proposed system and is set to 0.2 m in this case because of the following reason. Autoware also contains a function that creates a PCD map based on LiDAR-scan data, and its default value is 0.2 m.

However, we should adjust the gap size according to the environment. For example, the average distance between the LiDAR and walls differs when an AMR runs in a small space, such as a classroom, or a wide space, such as a factory. Thus, we can guess that the suitable gap size changes depending on the situation. In addition, the variance of the point distribution in the created PCD map or the presence/absence of tablet-scanned objects certainly affects the matching accuracy. In the following section, we compare TP values among various PCD maps created using different combinations of these parameters.

The experiment was performed in the same building shown in Fig. 4 or Fig. 5. As shown in Fig. 10, six points,

Table 1 Parameter combinations

Map name	Gap size (m)	Randomness	Object PCD (stairway)
NonStairway0.5	0.5	No	No
NonStairway0.4	0.4	No	No
NonStairway0.3	0.3	No	No
NonStairway0.2	0.2	No	No
NonStairway0.1	0.1	No	No
NonStairway0.05	0.05	No	No
NonStairway0.025	0.025	No	No
Stairway0.4	0.4	No	Yes
Stairway0.025	0.025	No	Yes
rdm_NonStairway0.4	0.4	Yes	No
rdm_Stairway0.4	0.4	Yes	Yes
LiDARmap	Using LiDAR on AMR		

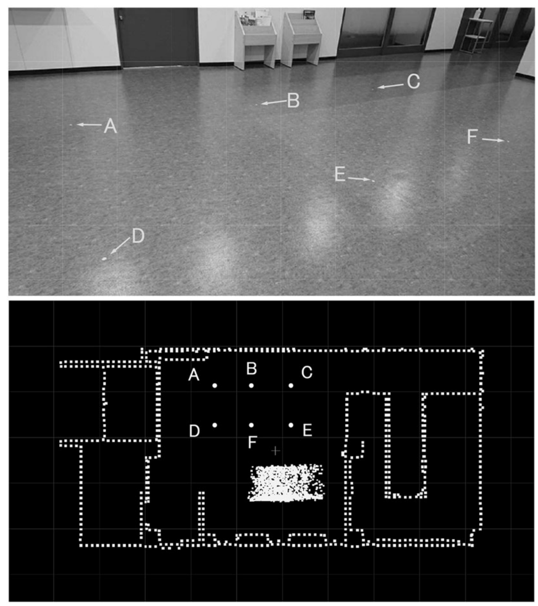


Fig. 10 Six test points, A–F, in the experiment. Top: six points' position on the photo and bottom: their position on the PCD map

Points A–F, were selected as the test points of accuracy. At these test points, the 3D LiDAR was left to stay at least 10 s one by one in several orders. All the LiDAR data were recorded as ROSBAG data with the ROS function. Using the same ROSBAG data, the PCD maps created from different parameters were evaluated using the TP index. The results obtained in the order C→F→B→E→A→D are presented in the next section, although experiments were performed in the six different orders.

5.3 Parameters

We created some PCD maps for autonomous driving simulations by setting different values for the following three parameters. The combinations of these parameters are shown in Table 1.

1. Gap size
   We expected an optimal gap size in the PCD maps. For the matching computation between the LiDAR-scan data and PCD map, we used the NDT matching algorithm [11]. There, the distribution of points in each voxel was approximated as the Gaussian distribution. We assumed that the shape of the Gaussian function did not change so much even if we set a too-small gap size, indicating that small gap sizes do not always improve the matching accuracy. Conversely, too-large gap sizes do not express the existence of walls correctly. In this section, we sought the optimal gap sizes by setting the gap size to 0.025, 0.05, 0.1, 0.2, 0.3, 0.4, and 0.5 m.
2. Tablet-scan
   Next, we examined the effect of the tablet-scan explained in Sect. 3. Actually, an AMR can detect objects that do not exist in design drawings. If these objects are small, their influence on the matching accuracy is minute. However, some large objects, such as machining tools, reflect the laser for scanning in a wide area such as the wall, which certainly influences the scan data for matching. Thus, we confirm whether the tablet-scan data improved the self-localization accuracy. The staircase was selected as a large object to scan by the tablet.
3. Random point alignment
   In Fig. 4, many points align on a straight line, because the walls were drawn as straight lines in the original drawings. If such a distribution is expressed by the Gaussian function, the variance to the wall becomes zero. In the NDT matching, the gradient of the Gaussian function is used to estimate the current position. Steep slopes such as zero variance will make this estimation difficult if the initial position for the detection is far from the optimal position. Thus, we intentionally placed the points of the wall randomly instead of in a straight line (Fig. 11).

Figure 12 shows the time course of TP during the experiment. At the six points, the 3D LiDAR is fixed and the TP value stabilizes at a constant value. The TP value sometimes varies sharply between the points, because the 3D LiDAR moves from one test point to another.

"LiDARmap" used the map created from the PCD scanned by the LiDAR on the AMR using the Autoware function. We expected the "LiDARmap" to produce the better self-localization result. As expected, "LiDARmap" had the highest TP value among all the maps. Notably, the highest TP value of the "LiDARmap" is just a reference to indicate that accuracy can be enhanced to this value.

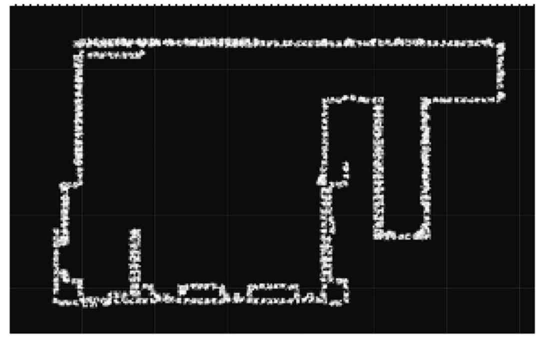


Fig. 11 Downsampled and randomized PCD map

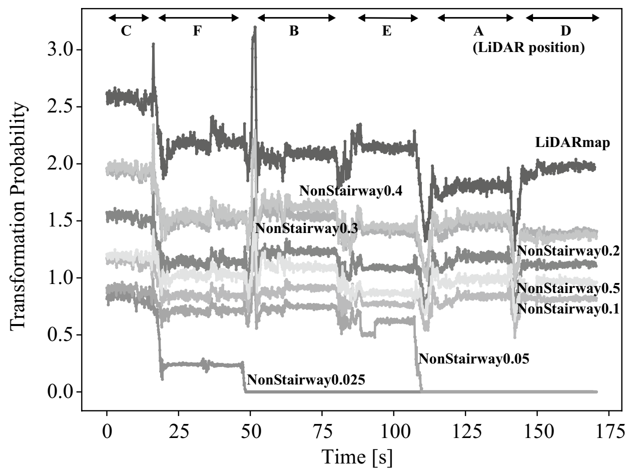
First, we examined the effect of gap size. Figure 12a shows that NonStairway0.4 had the highest TP value. It also shows that some matching errors occurred in NonStairway0.025 and NonStairway0.05. The results show that smaller gap sizes do not always improve the matching accuracy but rather decrease it.

Next, the effect of tablet-scan data was examined for 0.4 and 0.025-m gap sizes, the best and worst sizes, respectively, in Fig. 12a. The results are shown in Fig. 12b. The map with the 0.025-m gap size failed in the self-localization in around 40 s without the tablet-scan data. Nevertheless, the addition of the tablet-scan data enabled us to keep the self-localization throughout the simulation period. Comparatively, the tablet-scan data improved the TP in the map with the 0.4-m gap size. These results indicate that the tablet-scan data surely increase the matching accuracy.

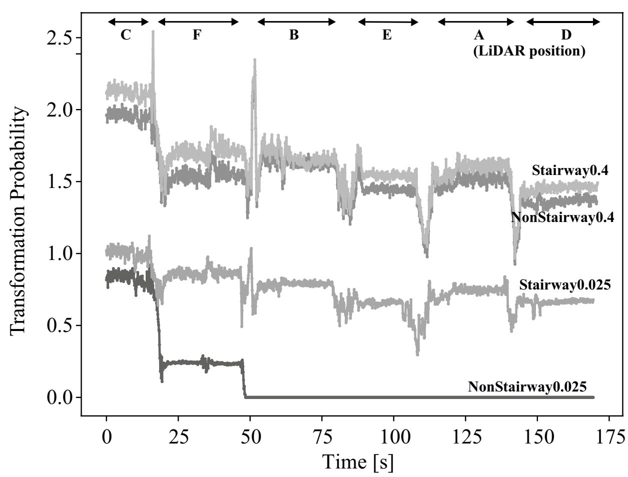
Finally, the effect of the random point alignment was investigated by introducing this process to the map with the best gap size (0.4 m). As shown in Fig. 12c, the matching accuracy increased in both maps with and without the tablet-scan data. Overall, the map with the 0.4-m gap size and with both the tablet-scan data and random point alignment was the best combination. The map with the best combination never made any matching errors in five consecutive self-localization simulations. These results demonstrate that the proposed system is applicable to map creation.

6 Conclusion

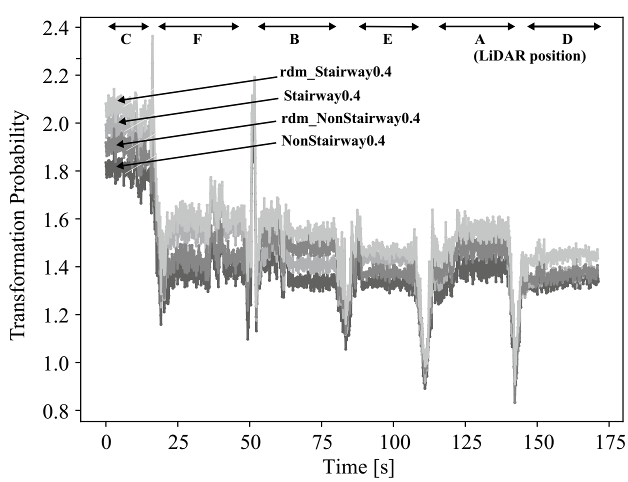
In this study, we proposed a method for PCD map creation from paper design drawings and developed a software application to easily apply this method to the map creation process. The proposed method has the following advantages. It can provide a PCD map without visiting the



(a) Effect of the gap size



(b) Effect of the tablet-scan data



(c) Effect of the random point alignment

Fig. 12 TP in the self-localization simulations without the LiDAR

workspace of AMR. Particularly, even if the workspace is vast, no scan is required. Thanks to this property, the route of an AMR can be planned based on the created PCD map in advance before the AMR is actually introduced. Such an AMR can operate immediately at the site. This merit cannot be obtained if the PCD map is created simultaneously with the self-localization. Indeed, objects not shown in the design drawings might be many, requiring an actual environment scan. If so, the proposed method employs a tablet with LiDAR, which has a friendly graphical user interface, works as a laptop, and above all, is less expensive than a 3D LiDAR sensor. The ease of scanning the workspace is another advantage of the proposed method. Moreover, the skip of the LiDAR-scan (i.e., beforehand AMR operations for the manual scan widely used in the traditional methods) shortens the PCD map creation time and reduces the required labor for AMR manipulation. Comparatively, our application software runs so quickly, completing the PCD map creation in a few seconds after inputting an image. Further, our system allows us to partially update the PCD map easily without recreating the entire PCD map, implying that we can flexibly incorporate layout rearrangements.

Acknowledgements A part of this study is supported by "Human resource development and research project on production technology for aerospace industry : Subsidy from Gifu Prefecture".

References

1. Autoware Developer Website. https://gitgub.com/CPFL/Autoware
2. Cloud Compare home. https://www.danielgm.net/cc/
3. Point Cloud Library Website. https://pointclouds.org
4. ROS Website. https://www.ros.org
5. Velodyne Lidar Website. https://velodynelidar.com
6. Voxel Grid Filter. https://pointclouds.org/documentation/classpcl_1_1_voxel_grid_3_01pcl_1_1_p_c_l_point_cloud2_01_4.html
7. Iyer G, Ram RK, Murthy JK, Krishna KM (2018) Calibnet: geometrically supervised extrinsic calibration using 3d spatial transformer networks. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, p 1110–1117
8. Kitsukawa Y, Kato S, Akai N, Takeuchi E, Edahiro M (2017) Field testing of self-driving vehicles: lessons learned on localization. Int Assoc Traffic Saf Sci 42(2):48–53
9. Otero R, Lagüela S, Garrido I, Arias P (2020) Mobile indoor mapping technologies: a review. Autom Constr 120:103399
10. Puente I, González-Jorge H, Martínez-Sánchez J, Arias P (2013) Review of mobile mapping and surveying technologies. Measurements 46(7):2127–2145
11. Takeuchi E, Tsubouchi T (2006) A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping. In: 2006 IEEE/RSJ international conference on intelligent robots and systems, IEEE, p 3068–3073
12. Wang R, Peethambaran J, Chen D (2018) Lidar point clouds to 3-D urban models: a review. IEEE J Select Top Appl Earth Observ Remote Sens 11(2):606–627

13. Zhang J, Singh S (2018) Laser-visual-inertial odometry and mapping with high robustness and low drift. J Field Robot 35(8):1242–1264