The Twenty-Seventh International Symposium on Artificial Life and Robotics 2022 (AROB 27th 2022),
The Seventh International Symposium on BioComplexity 2022 (ISBC 7th 2022),
The Fifth International Symposium on Swarm Behavior and Bio-Inspired Robotics 2022 (SWARM 5th 2022)
ONLINE, January 25-27, 2022

PCD map creation from factory design drawing for LiDAR self-localization of autonomous mobile robot

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

[1]Graduate School of Natural Science and Technology, Gifu University, Gifu, Japan
(E-mail: z4525025@edu.gifu-u.ac.jp)
[2]Faculty of Engineering, Gifu University, Gifu, Japan
(E-mail: aa300007@edu.gifu-u.ac.jp, rmorita@gifu-u.ac.jp, satoshi@gifu-u.ac.jp)

Abstract: This paper proposes a method for creating the point cloud data (PCD) map required to the LiDAR localization in autonomous driving. For utilizing an AMR, we must prepare an environmental map in advance. Our method creates the map from the design drawings. For the objects not depicted in the drawings, we introduce tablet scan data, which is merged to the map created from the drawing. Three factors at the map creation affecting the accuracy of self-localization are investigated: the gap size in the PCD map, the presence of the tablet scan data, and the random point alignment at the map creation. The effect of these factors is evaluated by the simulations with the actual scan data. Consequently, the existence of the optimal gap size as well as the accuracy enhancement by both the tablet-scan data and the random point alignment are clarified. In addition, autonomous driving using the PCD map created by our proposal system is successfully achieved.

Keywords: Autonomous mobile robot, map creation, LiDAR-localization, point cloud data.

1. INTRODUCTION

In industrial factories, assembling process necessitates transportation of parts and products. For the transportation, autonomous guided vehicles (AGV) are introduced for the automation to reduce human resources. These AGVs normally utilize "guide," such as mechanical rails or lines drawn on the floor of the factory, to show them the way to the destination. This guide is one of the effective methods to limit the vehicle within their accessible area. In addition, it facilitates the positional recognition and the collision reduction.

However, the guide requires a large-scale rebuild of the factory for its installation. In the factory manufacturing various products, the production lines are forced to rebuild every time the products are updated. Job-shop type factories have to rearrange their layout frequently. Therefore, there are cases that the AGV with guide equipment cannot be easily adopted depending on the industry.

Under this background, our research project aims to develop an autonomous mobile robot (AMR), one of whose advantages is to require no additional rebuilding to the factory itself. To achieve the navigation without guide equipment, we decided to introduce one of the auto-driving technologies, LiDAR localization. 3D LiDARs we are planning to implement will certainly detect current AMR position without any factory rebuilding. However, it requires the current layout information, i.e., a "map" for the localization instead. In particular, our target is the factory whose layout is frequently updated. Thus, how we should obtain the newest map for the rearranged factory layout became a problem.

In the LiDAR localization system that we equipped to our AMR, the 3D maps have a data format called Point Cloud Data (PCD). The simultaneous localization and mapping (SLAM) algorithms are effective methods for a map generation. However, enhancing the map accuracy requires lots of computational cost such as the loop-closure detection or some other sensors like an odometer or an inertial measurement unit (IMU). Compared to SLAM, some other methods [1, 2] prepare the 3D PCD map (we call it PCD map throughout the paper) in advance by some sorts of method, for example, manual driving. This pre-prepared PCD map method has advantages of low computational cost and higher real-time property compared to SLAM.

In this paper, we aim to develop a PCD map creation system, whose input is a design drawing and output is the PCD map of the inside of building, i.e., AMR's environment. To demonstrate its effectiveness, we simulate autonomous moving of an AMR using the PCD map that our proposed system created. The next section describes a method to create a PCD map from a design drawing. In Section 3, to cope with the environmental changes inside the buildings, the scans by a tablet with a LiDAR will be newly introduced. Section 4 discusses the optimal gap size between points of PCDs. In Section 5, we conduct the simulation of the self-localization and evaluate its accuracy. In Section 6, we conclude this paper.

2. PCD MAP CREATION FROM DESIGN DRAWING

2.1 PCD format

The PCD map is written in Point Cloud Data format [3]. This format consists of a three-dimensional coordinates part as well as a unique header part. Utilizing LiDARs, the PCD file is automatically generated by the software provided from its developer in many cases. Of course, the file we write by ourselves can be also recognized as the PCD file as long as it matches the PCD format including the proper header part. It means that we can intentionally place the points (objects) data at any position we desired in the PCD map.

Based on this fact, we create the PCD map, i.e., the PCD format file by computing the wall positions from the two-

† Ryutaro Kaneko is the presenter of this paper.

dimensional design drawings without the LiDAR scan. At that time, we will make the most use of the geographical information in the drawings by electronically scanning it.

2.2 Problem

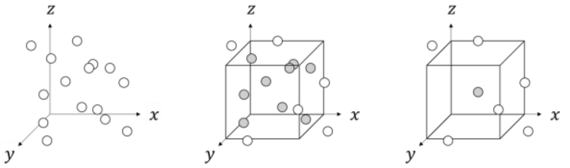
To create a three-dimensional PCD map from a two-dimensional drawing, there are four issues as below.

1. Lack of height information
   3D LiDARs can detect walls as 3D objects in the buildings because they can get the height information (Z-axis coordinates) by several numbers of the lasers directed to the different height. Accordingly, the PCD file should have three-dimensional data. In contrast to it, design drawings are usually two-dimensional and do not have height information.
2. Scale unfitness
   The scale of the design drawings differs one by one, so we must consider the map scale to generate the wall (object) data.
3. Point cloud density
   In PCD maps, an object is expressed as many numbers of point data (a cloud of points) on the surface of the objects. If points are too dense, the computational cost increases, while, if too sparse, the accuracy of self-localization tends to decrease. The density of the point cloud, and to what extent the point in PCD maps should be up-sampled (or down-sampled) are important issues.
4. Not depicted objects in drawings
   Design drawings include only constructional information. However, in the real buildings, there are some furniture in houses, some mechanical tools and inspection equipment composing production lines in factories. The presence or absence of these information affect the accuracy of the self-localization.

2.3 Approach to solve the issue

We consider the approach to solve the issues mentioned in the previous section as follows:

1. Point accumulation to the height direction
   Main place we aim to run our AMR is the inside of buildings. Compared with outside of the buildings, the distance that the LiDAR detects is short. It implies that most of the laser reflects on the walls before it reaches the ceiling. Thus, in the PCD map creation, we accumulate the points in the height direction to the sufficient height regardless of the situation.
2. Scale adjustment
   We scale the PCD map manually at the final stage of the map creation according to the description on the design drawings or actual measurement of the building.
3. Up/Down-sampling
   Voxel Grid filter [4] is applied to down-sample the points when the PCD map is created from the image of the drawings. On the other hand, up-sampling is executed when enlarging the scale.
4. Tablet scan
   We utilize a LiDAR-mounted tablet to scan objects which do not exist in drawings. The data the tablet scanned are converted to the PCD format. Finally, we merge them with the PCD map created from the drawings.



(a) (b) (c)

Fig. 2. Voxel Grid filter Algorithm

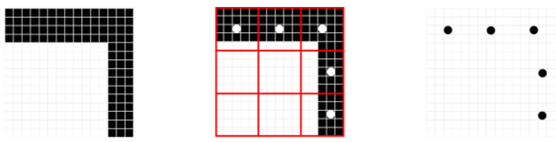
2.4 Up/Down-sampling

2.4.1 Voxel Grid filter

The autonomous driving needs real-time processing for the localization. Too many point cloud data reduce the processing speed. The down-sampling is an advantageous way to save the computational cost.

The algorithm called the Voxel Grid filter [4] down-samples the points in a PCD map. It divides a space consisting of many points in Fig. 2 (a) into many voxels, as illustrated in Fig. 2 (b). Then it calculates the centroid in voxels as representative points, and reconfigures the points as shown in Fig. 2 (c).

The Voxel Grid filter is generally applied for three-dimensional PCD. In this paper, however, we apply it for two-dimensional images at the early stage before making three-dimensional PCD. Therefore, we divide images as pixel unit, not as voxel unit. Actually, a picture in Fig. 3 (a) is divided into several desired-size cells (e.g., side length is 5 pixels in this example), as illustrated in Fig. 3 (b). Then, the centroid of black pixels in each cell is calculated, and all the points in the cell are replaced with one representative point at this centroid position. As a result, a two-dimensional image with down-sampled points, as depicted in Fig. 3 (c), is obtained.



(a) (b) (c)

Fig. 3. Voxel Grid filter in two-dimensional space

2.4.2 Up-sampling

After applying the two-dimensional Voxel Grid filter, the outputted two-dimensional points should be rescaled to adjust to actual size. This operation is usually enlargement, which expands the gap size at the same time. Sometimes, the gap size exceeds the value that is larger than planning to be utilized in the LiDAR localization: up-sampling is required to increase the density of the points at the rescale process.

Now, all the gap sizes between points in the PCD are almost the same thanks to Voxel Grid filter. Applying this advantage, points can be easily interpolated to reduce the gap size to the desired one.

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

Selecting a certain facility in Gifu University as an experimental place of the autonomous driving, the PCD map in Fig. 4 was created from its design drawing. At first, this map was created without Voxel Grid filter: it was down-sampled in the equal intervals, i.e., by picking up the rows and columns every constant value from the pixel data. Although the autonomous driving was achieved, the self-localization sometimes provided the wrong information due to the matching error between the LiDAR scan data and the created maps. Investigating the maps, we found that the points roughness and fitness are uneven. In addition, there were some blanks at the position where the wall should have existed, as depicted in the right enlarged map in Fig. 4. We considered this must be the main reason of the matching errors.

Then, we applied the method we proposed in the previous section. The cell side length was set to 0.2 m when applying two-dimensional Voxel Grid filter. Because of the effect of Voxel Grid filter, the gap became constant and the walls were connected without the blank, as shown in Fig. 5. This map effectively reduced the matching error between the scan data and map data during the autonomous driving.

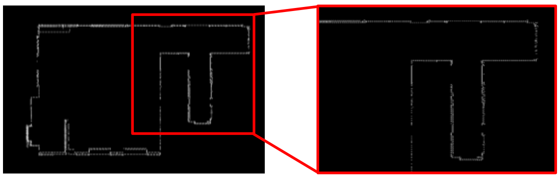


Fig. 4. PCD map down-sampled in constant interval.

Fig. 5. PCD map down-sampled by 2D Voxel Grid filter.

2.5 Outline of application software

We developed a software application that enables us to sequentially apply all of the above processes.

The processes the users must conduct are as below.

1. Scan a factory design drawing data
   A design drawing (we assume it is given as a paper) must be converted to image data format (jpg., png. and so on) using an electronic device like a scanner or a camera.
2. Pre-processing including binarization
   After the binarization, the black pixels are regarded as the walls: they are stored into an array, i.e., two-dimensional point data, and are utilized to create three-dimensional point cloud based on its geographical information. This is the reason why the figures other than the wall (e.g., dimension lines) should be deleted in the pre-process by some image editors.
3. Run our application
   When our application started, the file to which the data generated in the previous process 2 are saved must be designated. Then, the processes described in Section 2.4.1 are executed with some necessary parameter assignments. This process outputs a temporary PCD map.
4. Check the scale of the temporary PCD map
   The end of task 3 automatically starts a PCD viewer. This viewer allows us to evaluate the distance between certain points on the current PCD map. The temporary PCD map created in the previous task 3 has the correct phasic relation between walls, but the scale is not correct from the actual buildings. It is because the image scanned in the task 1 depends on not only the scale of the original drawing but also the scanner resolution. Distances on the PCD map is evaluated between the points where actual length is known.
5. Rescale
   The end of the process 4 automatically opens a pop-up window, in which the scale parameter can be inputted. After that, the rescaling and the up-sampling described in Section 2.4.2 is carried out. Finally, the desired PCD is outputted.

A few seconds are needed to obtain the PCD map from the design drawings. Python is used as a programming language including an image processing library, OpenCV.

3. PROCESS FOR LAYOUT CHANGE

We can obtain the PCD map from the design drawings so far. Unfortunately, however, this PCD map does not contain some objects that are not depicted on the design drawings, such as furniture, containers and mechanical tools. If these objects are not reflected to the PCD map, the LiDAR localization recalls different data from the current environment at the actual driving, which results in crucial matching errors.

This problem will be solved, if we add the missing objects to the design drawing and then apply the map creation system we proposed in the previous section. However, it will be difficult to add them at the precise position. Of course, as often utilized, if we scan the inside of the building using the AMR with the LiDAR, we can get the map including all the objects. However, we have been avoiding this method from the beginning because it is not easy for the factory workers to operate AMR for the map creation: this is why we are aiming to create the PCD map from the drawings.

Here, we propose another method, “tablet scan,” in which a LiDAR-mounted tablet is used to scan the missing objects. Afterwards, their data are merged with the PCD map we created in the previous section. Tablets are easier to handle than AMRs and many people might have them, implying that we don’t have to buy them newly.

Fig. 6 (a) is the PCD map where the stairway data shown in Fig. 6 (c) scanned by the tablet are merged to the PCD map we created from the design drawing. Fig. 6 (b) is the picture of the stairway. In fact, the original design drawing contains the stairway, but the space where the stairway occupies in the building does not have a constant height. i.e., there is empty space under or over the stairway. To reflect it in the PCD map, a special process that our map creation system cannot conduct is required. The tablet scan is available to express such a spatial property correctly.

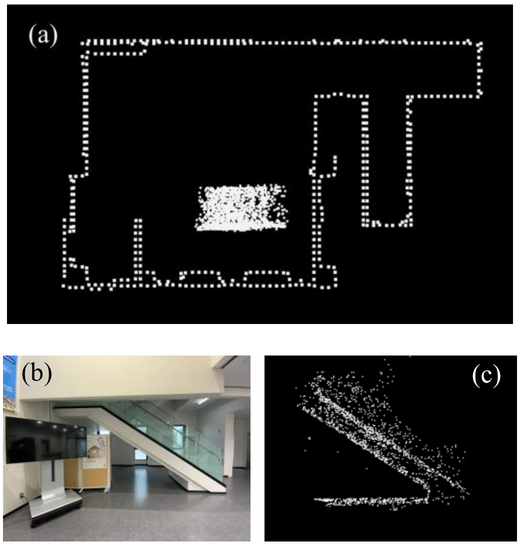


Fig. 6. PCD map where an obstacle PCD is overlaid.

3. AUTONOMOUS MOIVNG EXPERIMENT

We conducted an experiment to verify whether the map created with the system we proposed in this paper allows the AMR to travel autonomously with executing self-localization.

Fig. 7 shows the AMR we used in this experiment. It mounts the three-dimensional LiDAR (Velodyne-16: Velodyne Lidar [5]). Autoware [6] is working as a control system on Robot Operation System [7].

Fig. 8 illustrates the LiDAR scan data as well as the AMR's self-position. The desired path, which was planned on the PCD map we created by our system, is also depicted there. As shown in every second snapshots of Fig. 9, the map created from the drawing enables the AMR to localize itself and to drive autonomously.

Fig. 7. AMR utilized in the experiment.

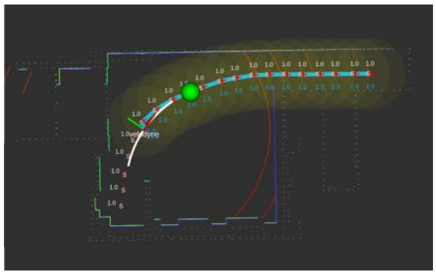


Fig. 8. The reference trajectory and localization result on the created map.

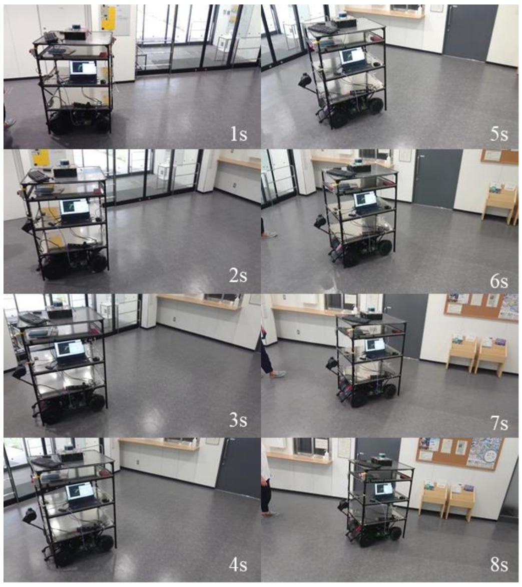


Fig. 9. Snapshots of the AMR movement

5. VERIFICATION OF ACCURACY

5.1 Transformation Probability

This section verifies the accuracy during the autonomous driving to prove the effectiveness of our proposal system. For the assessment, we adopt Transformation Probability (TP) [8]. TP is an index of scan matchings which expresses matching level between reference PCD map (environment PCD map) and current LiDAR scan data. According to the reference [8], TP value is approximately more than two if the matching is stable. On the other hand, this value is zero if the matching is not observed.

5.2 Method

We can specify a gap size between points when making PCD maps. In Fig. 5, for example, a gap size of the PCD map is selectable in our proposed system, and set as 0.2 m in this case because of the following reason: Autoware has a function of creating the PCD map based on the LiDAR scan data so far. Its default value is 0.2m.

However, we should adjust this gap with respect to the environment. For example, the average distance between the LiDAR and the walls is different when an AMR runs in small space like a classroom or in open space like a factory. So, we can guess the suitable gap size that will be exist depending on the situation. In addition, the variance of the point distribution in the created map, or the presence/absence of the tablet-scanned objects will certainly affect the matching accuracy. In the following section, we compare TP among various PCD maps created with the different combination of these parameters.

5.3 Parameters

We created some maps for the autonomous driving simulations by setting the different values to three parameters below. The combination of the parameters is shown in Table 1.

Table 1. Parameters Combination.

Map name	Gap size [m]	randomness	Object PCD (Step)
Nonstep0.025	0.025	No	No
Step0.025			Yes
rdm_Nonstep0.025		Yes	No
rdm_Step0.025			Yes

1. Gap size
We expected that there was an optimal the gap size in the PCD maps. For the matching computation between the scan data and the PCD map, we are utilizing NDT matching algorithm [9]. There, the distribution of the points in each voxel is transformed as the Gauss function. We guess that the shape of the Gauss function does not change so much even if we set too small gap size, indicating that the smaller gap size does not always improve the matching accuracy. On the other hand, too large gap size will not express the existence of the walls correctly. In this section, we seek optimal gap sizes by changing the gap size in 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 Section 3. Actually, AMR will detect the objects that does not exist in the design drawing. If these objects are small, the influence for the matching accuracy is small. However, some large objects such as machining tools reflect the laser for scan in the wide range like the wall, which certainly influences the scan data for matching. So, we will here confirm whether the tablet-scan data improves the accuracy of the self-localization or not.

3. Random point alignment
In Fig .5, many points align on the straight line since the walls were drawn as the straight lines in the original drawings. If such the distribution is expressed by the Gaussian function, the variance to the wall become zero. In the NDT matching, the gradient of the Gaussian function is utilized to estimate the current position. The steep slope like zero variance will make this estimation difficult if the initial position for the detection is far from the optimal position. To avoid this situation, we have an idea such that the points of the wall is intentionally placed with the randomness instead of aligning on the straight line, as shown in Fig. 10. We examined this effect with the different amount of the randomness.

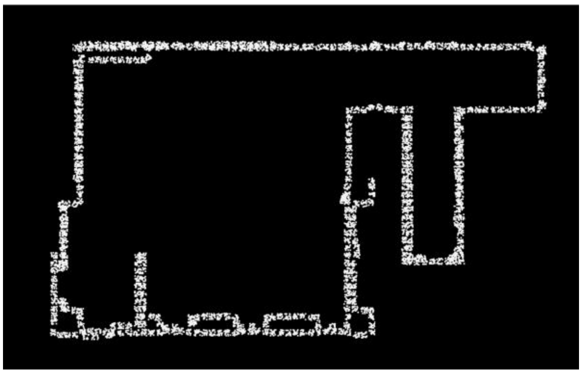


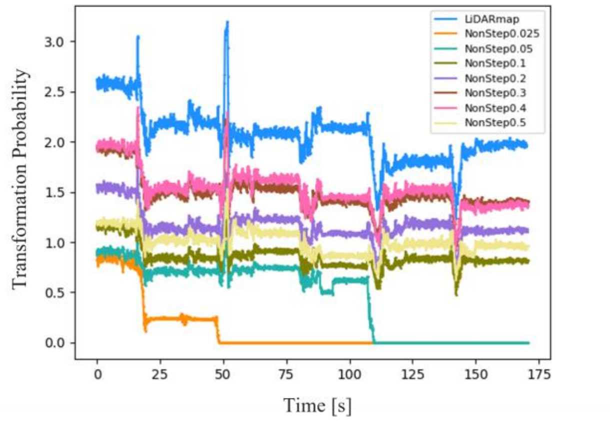
Fig. 10. Down-sampled and randomized PCD map.

Fig. 11 shows the simulation results of the self-localization. Here, “LiDARmap” utilized the map that was created from the point cloud data scanned by the LiDAR on the AMR using the Autoware function. We considered that the “LiDARmap” will produce the best self-localization result. As we expected, “LiDARmap” has the highest TP value compared with the other maps. Note that TP of the “LiDARmap” is just a reference as the best performance to indicate that the accuracy can be enhanced to this extend.

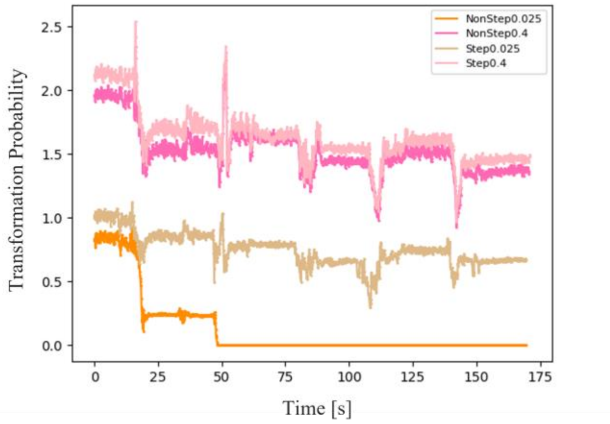
First, we examined the effect of the gap size. Fig. 11 (a) indicates that Nonstep0.4 has the highest TP value in comparison with the other gap size. It also says some matching errors occurred in Nonstep0.025. This result shows that the smaller gap size does not always improve the matching accuracy, rather decreases it.

Next, the effect of the tablet-scan data was examined to the gap size 0.4 and 0.025, the best and the worst one in the Fig. 11 (a). The result is shown in Fig. 11 (b). The map with the gap size 0.025 failed in the self-localization in around 40 s without the tablet-scan data. Nevertheless, the addition of the tablet-scan data enables us to keep the self-localization throughout the simulation period. In addition, the tablet-scan data improve TP in the map with the gap size 0.4. These results mean that the tablet-scan data surely increases matching accuracy.

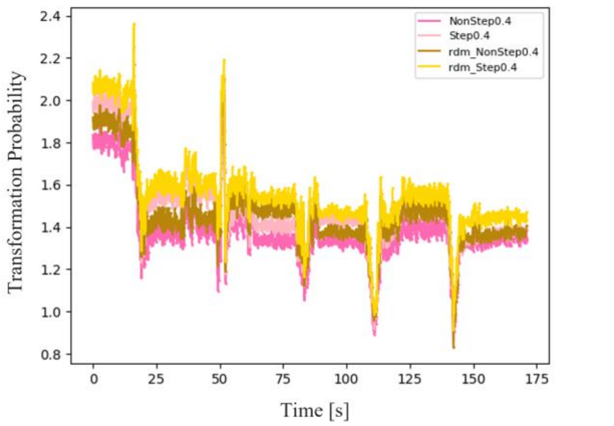
Finally, the effect of the random point alignment was investigated by introducing this process to the map with the best gap size 0.4m. As shown in Fig. 11 (c), the matching accuracy has increased in both maps with and without the tablet-scan data. After all, the map with 0.4 gap size, and with both the tablet-scan data and the random point alignment was the best combination. The map with the best combination had never made any matching errors in five consecutive simulations of self-localization. These results demonstrate that our proposed system is applicable as the map creation



(a) Effect of gap size



(b) Effect of the tablet-scan data



(c) Effect of the random point alignment

Fig.11. TP in the simulations of self-localization without the LiDAR.

6. CONCLUSION

This paper proposed a method for the PCD map creation from the design drawings, and developed a software application to easily apply our method in the map creation process. Our method does not need the LiDAR scan in advance, i.e., beforehand AMR operations for the manual scan that is widely utilized in the ordinal methods. So, it will be possible to shorten the map creating time and to reduce labors for AMR manipulation. Our application program works so quickly as to complete the map creation in a few seconds from inputting an image. Additionally, our system allows us to partially update the PCD map easily without recreating the whole PCD maps, implying that we can cope with the layout rearrangement flexibly.

REFERENCES

[1] Ganesh Iyer, R. Karnik Ram., J. Krishna Murthy, K. Madhava Krishna, “CalibNet: Geometrically Supervised Extrinsic Calibration using 3D Spatial Transformer Networks”, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[2] Zhang h. and Singh S., “Laser-visual-inertial Odometry and Mapping with High Robustness and Low Drift”, *Journal of Field Robotics, Vol. 35, No. 8, pp. 1242 - 1264,* 2018.

[3] Point Cloud Library Website, “https://pointclouds.org/”

[4] Voxel Grid filter, “https://pointclouds.org/documentation/classpcl_1_1_voxel_grid_3_01pcl_1_1_p_c_l_point_cloud2_01_4.html”, Point Cloud Library Website.

[5] Velodyne Lidar Website, “https://velodynelidar.com/”

[6] Autoware developer Website, “https://gitgub.com/CPFL/Autoware”

[7] ROS Website, “https://www.ros.org”

[8] Kitsukawa Y., Kato S., Akai N., Takeuchi E., Edahiro M., “Field Testing of Self-Driving Vehicles: Lessons Learned on Localization”, *International Association of Traffic and Safety Sciences (IATSS)*, Vol. 42, No.2, pp.130-135, 2017.

[9] Takeuchi E., Tsubouchi T., “A 3-D Scan Matching using Improved 3-D Normal Distributions Transform for Mobile Robotic Mapping”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3068-3073, 2006.