

Date of publication xxxx 00, 2021, date of current version xxxx 00, 2021.

Digital Object Identifier xx.xxxx/ACCESS.2021.DOI

Towards Customizable Robotic Disinfection with Structure-Aware Semantic Mapping

ZHENZHEN XIANG AND JIANBO SU, (Senior Member, IEEE)

Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Jianbo Su (e-mail: jbsu@sjtu.edu.cn).

This work was financially supported in part by the National Natural Science Foundation of China under Grants 61533012, 91748120 and 52041502.

ABSTRACT In the current COVID-19 pandemic, people expect to use robots to replace humans to complete the disinfection work in public places. Since different regions in the environment have different risks, in addition to the conventional SLAM capability, the robot also needs to be able to recognize and distinguish different objects in the scene to complete customizable disinfection tasks. In this paper, we propose a LiDAR-based semantic mapping system that can be used for robotic disinfection tasks. By using the prior information about the scene structure, our system can extract different levels of semantic information from the raw point cloud, so as to not only construct an occupancy grid map for navigation, but also construct a hierarchical semantic map that meets the needs of customizable disinfection tasks, including setting the navigation waypoints, disinfection distances and disinfection time. The effectiveness of our proposed system is proved in the real-world metro disinfection applications.

INDEX TERMS Disinfection robot, structure-aware, semantic mapping, prior knowledge

I. INTRODUCTION

With the current global pandemic of COVID-19, the disinfection of public places has gradually attracted people's attention. Since manual disinfection operations are very time-consuming and labor-intensive, researchers are trying to use autonomous robots to complete the disinfection tasks to reduce the burden on related staffs and avoid potential infection risks [1], [2].

Many scenarios that need to be disinfected are relatively standard structured scenes, such as metros and buses. Taking the metro car in Fig. 1 (top) as an example, the disinfection robot needs to autonomously navigate from the start to the end of the metro and perform disinfection task (such as spraying disinfectant or emitting ultraviolet rays) at the same time.

Simultaneous localization and mapping (SLAM) is the core functionality for the robot to complete the disinfection task. Furthermore, since different structures in the scene usually have different risks, disinfection tasks often need to be *customizable*, that is, the disinfection time and disinfection distance of different structures are various [3], [4]. For example, the seats and armrests in metro cars are the regions most

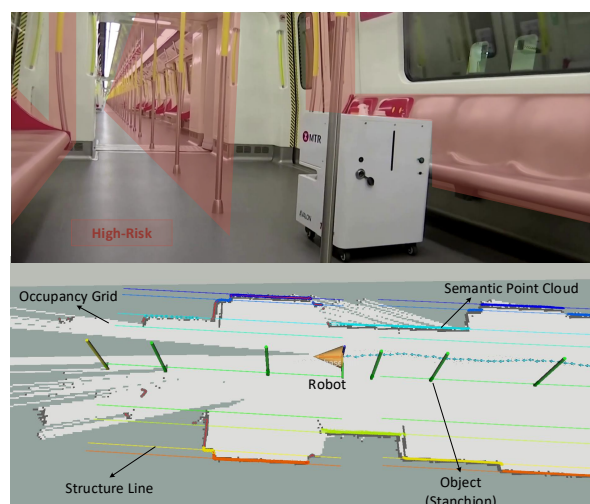


FIGURE 1. The disinfection robot working in the metro car (top) and the semantics in the generated hierarchical semantic map (bottom). Note that different colors of point cloud represent different structures in the metro car which should be treated accordingly in customizable disinfection tasks.

frequently touched by people, so compared to other regions, longer disinfection time and shorter disinfection distance are required to ensure a higher disinfection result.

To be able to fulfill the customizable disinfection requirements for different structures in the environment, unlike normal navigation tasks which only require the robot to know where there are obstacles, disinfection tasks need the robot to understand the structure around it, i.e., to have a *semantic* level understanding.

Therefore, in this paper, we present a LiDAR-based semantic mapping system for robotic disinfection tasks. Our system can use the scene structure as the prior knowledge to segment the point cloud and obtain the semantic information in it, so as to understand the structure around the robot. The semantics as depicted in Fig. 1 (bottom) can be extracted and recorded in the map, and finally used to support the robot to complete customizable disinfection tasks.

II. RELATED WORK

A. LIDAR-BASED SEMANTIC MAPPING

LiDAR-based SLAM system such as *GMapping* [5], *Hector SLAM* [6] and *Cartographer* [7] are widely applied in practical applications where the capability of autonomous navigation for a robot is required. Although these frameworks have made great success, they are usually designed for general navigation tasks which only consider the obstacles and build occupancy grid maps of the environment. Therefore, it is not sufficient to directly apply these frameworks to meet the needs of customizable disinfection tasks.

Compared with traditional SLAM systems, semantic SLAM systems attempt to extract more deeper and higher-level information from the raw data and eventually build a semantic map. Two different levels of semantics have been widely studied and applied: the feature-level semantics and the object-level semantics.

1) Feature-level

The feature-level semantics consists of various geometric primitives, like points, lines and planes detected and extracted from the input point cloud. In [8], Chen et al. divided the point cloud into different categories and used such information to improve the data association. Line features [9] and planar surfaces [10] were also extracted from the point cloud for fast and precise localization in autonomous driving tasks.

2) Object-level

The object-level semantics includes the general objects and application-oriented objects. General objects can be the dense/sparse point cloud models or solid figures. Salas-Moreno et al. matched the objects in the environment to their dense 3D models and perform the SLAM task at the level of objects [11], while Bowman et al. proposed an algorithm to resolve probabilistic data association of 3D objects in semantic SLAM problems [12].

For those application-oriented object models, such as the door signs in the building [13], poles [14], [15] and traf-

fic signs [16] in the driving scenes, crops in the precision agriculture [17], or trees in the forest inventory [18], the performance of SLAM system usually can be further enhanced by integrating such semantic information, and some task-related requirements can only be fulfilled along with the optimization.

In our proposed semantic mapping system, the raw LiDAR point clouds are segmented into different classes in order to provide semantic-level information for customizable disinfection tasks.

B. STRUCTURE AS PRIOR KNOWLEDGE

To boost the performance of general SLAM systems in specific applications, various prior knowledges of the environment and the task are often exploited.

1) Building Structure

Karg et al. introduced the similarity of multi-floor layout as global constraints to the graph-based SLAM, which was able to generate consistent maps for multistory buildings [19]. Boniardi et al. took the architecture floor plans as prior knowledges to enhance the robustness of robot localization [20], [21].

2) Aerial Structure

Aerial images captured by satellites or unmanned aerial vehicles are another source of information that are commonly used to assist in the SLAM task. With the extracted line segments [22] or edges [23] from the aerial images, the mapping results could be constrained and closer to the ground-truth. Special application in precision farming by Pretto et al. demonstrated the effectiveness of fusing the aerial and ground information for much robust localization and mapping [24].

3) Geometric Constraints

The orthogonality and parallelism in structured environments have also been extensively investigated and considered as the constraints for both data processing and optimization in SLAM problems. Daoust et al. leveraged the parallelism of tunnel walls to remove unimportant LiDAR measurements for localizing a train in challenging underground environments [25]. The orthogonal and parallel planes in both indoor [26] and underground [27] environments were utilized for better mapping performance.

In our framework, considering that the raw 2D LiDAR point cloud is very sparse and less representative, we also use the prior knowledge of the scene structure for point cloud segmentation.

C. APPLICATIONS OF DISINFECTION ROBOTS

Different type of disinfection robots based on the UV-C light [28] and hydrogen peroxide vapor [29] have been applied to disinfect public utilities including hospitals, schools as well as metros investigated in this paper. As discussed in [30], the

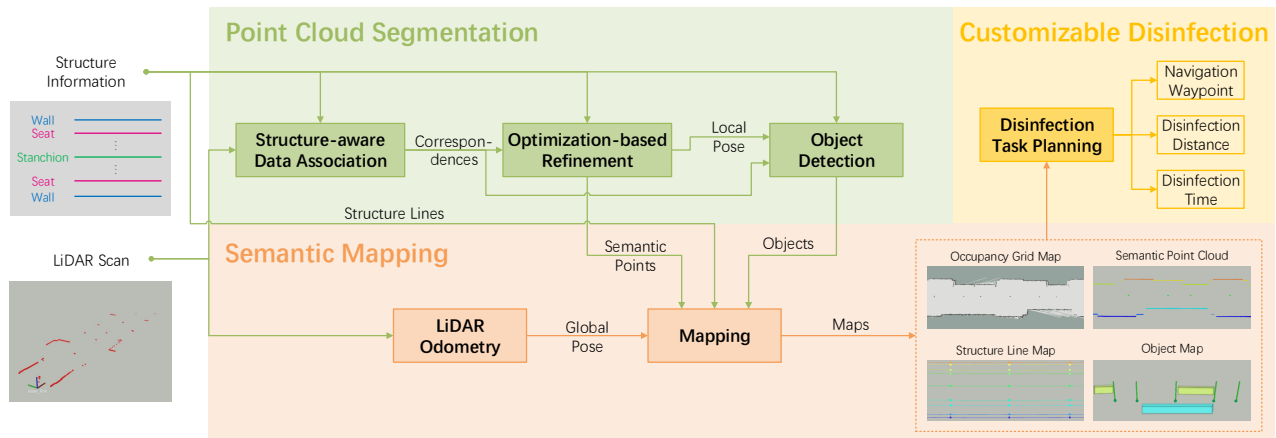


FIGURE 2. The proposed semantic mapping framework for customizable disinfection. The scene structure information is provided to the system as the prior knowledge. The point cloud segmentation subsystem (Sec. IV) first builds correspondences between LiDAR point cloud and the scene structure with a structure-aware method, followed by an optimization-based refinement module and an object detection module to further extract feature-level and object-level semantic information in the scene, respectively. The semantic mapping subsystem (Sec. V) builds and updates a hierarchical semantic map with the poses estimated by a LiDAR odometry module. The constructed semantic maps are further used to support the task planning in customizable disinfection applications.

disinfection performance may be affected by various factors. Due to the shadows caused by occlusions, the UV-C light or chemicals may not always touch the surface of objects in the environment. The effectiveness also reduces as the distance increases. In [31], the authors proposed to identify potential contamination areas for adaptive robotic disinfection in built environment. In [32], Conte et al. designed a disinfection map to evaluate the disinfection performance w.r.t the distance between the robot and targeting structures.

Because some regions of the scene to be disinfected are high-touch and high-risk, the robot needs to take longer disinfection time at a shorter distance to them in order to ensure the complete disinfection. In our system, we mainly rely on the semantic-level understandings of the scenario to fulfill such high-level and customizable requirements.

III. FRAMEWORK

In this section, the pipeline and structure of the proposed semantic mapping system will be described. As shown in Fig. 2, the system can be divided into three parts:

- The *point cloud segmentation* subsystem is designed to segment the point cloud by associating them with the structure of the metro car, followed by an optimization based refinement procedure. To further leverage the object-level information, the stanchions are also detected from the point cloud with the prior knowledge from the scene structure.
- The *semantic mapping* subsystem consists of a LiDAR odometry module and a mapping module. The LiDAR odometry module provides the poses of the robot in the metro car, which can be further used to fuse the extracted semantic points and stanchions and eventually build a hierarchical semantic map.
- The *customizable disinfection* subsystem finally utilizes

the rich semantic information in the scene to plan the robot's behavior, including placing the navigation waypoints, setting different disinfection distances and disinfection time for different regions.

In general, the semantic mapping framework we designed has three characteristics:

- *Semantic*: Semantic information (e.g., different categories of points including the *door*, *wall*, *seat*, *joint*, *stanchion* in the metro car) is extracted from the raw LiDAR point cloud, given the structure of the scene as prior knowledge. Then such semantic information will be merged to construct a hierarchical semantic map of the scene, which is necessary for customizable disinfection tasks, especially for the disinfection distance and time control.
- *Adaptive*: Our framework can be adapted to various scenarios with different structures. On the one hand, to segment the point cloud, the prior knowledge from scene structure can be configured according to different scenarios. On the other hand, the extracted semantic information used by the task planning process can also be selected according to customizable disinfection tasks.
- *Online*: Since the prior knowledge of the scene structure can be integrated, our framework is very lightweight and fast enough to run on the robot in real-time. The construction of the scene semantic map can be completed along with the disinfection process.

In the following two sections, we will first describe the details of structure-aware point cloud segmentation, followed by the semantic mapping procedure. For the customizable disinfection, we will demonstrate its implementation and application in the experiment section.

IV. STRUCTURE-AWARE POINT CLOUD SEGMENTATION

Since the 2D LiDAR point cloud is not as dense as the 3D LiDAR point cloud or camera images, it is usually difficult to extract the semantic information with a direct segmentation method. As reviewed in Sec. II-B, the information of scene structure can be considered as constraints in the SLAM system. Similarly, we propose to leverage the prior knowledge of scene structure in the semantic perception process. By associating the point cloud with the scene structure which can be manually annotated, we can obtain the corresponding label for each part of the point cloud.

A. REPRESENTATION OF THE SCENE STRUCTURE

The most accurate information of the scene structure is the floor plan, e.g., the layout of a metro car or bus. However, since there are many types of floor plans for different scenes and may not be easy to obtain, in order to enhance the versatility of the method, we suggest to simplify the description of the scene structure as much as possible.

Taking the metro car in Fig. 1 as an example, we can find that the distance from different part of the car to the center axis of the care is stable throughout the scene, although it can be various among different type of cars. Thus, we can model the such scenarios with a group of parallel lines, i.e., *structure reference lines*, as shown in Fig. 2, which can be easily generalized to other similar scenes like the buses and trains.

B. STRUCTURE-AWARE DATA ASSOCIATION

To associate the point cloud with the scene structure, we consider it as a registration or pose estimation process, that is, estimating the robot's pose in the scene. Then after the successful registration, we can extract the corresponding label for each laser point by the nearest neighbor search (NNS).

In the scene like the metro car, a two-stage grid-based searching method is designed to establish the initial correspondences between the point cloud and the scene structure:

- **Rotation Estimation.** We first attempt to rotate the laser point cloud with a sampled angle in a predefined angle range. Then the laser points are transformed to the rotated frame of the robot, projected along the rotated heading of the robot. The histogram of the projected laser points is computed along the lateral direction which is perpendicular to the rotated heading of the robot. Since most of the points fall on the corresponding reference lines, it can be imagined that, the distribution of the projected points will cluster better when the selected rotation angle is closer to the ground-truth orientation. Here we use the entropy of the distribution to find the best rotation angle:

$$\psi^* = \arg \min_{\psi} \sum_{i=1}^N -p_i(\psi) \log(p_i(\psi)), \quad (1)$$

where $\psi \in [-\psi_0, \psi_0]$ is the sampled rotation angle, and $p_i(\psi) \in [0, 1]$ is the corresponding ratio of the projected points located in the i -th lateral bin.

- **Position Estimation.** Based on the rotated laser point cloud, we can further perform the lateral grid searching, since the reference lines can not provide longitudinal constraints. For each given structure reference line and lateral movement step, we can compute the number of points that fall in a specific distance range of each reference line. The lateral position corresponding to the maximum number of matching points will be considered as the best estimation.

C. OPTIMIZATION-BASED REFINEMENT

After the initial data association, the point-line correspondences can be computed by a distance threshold. Then we can perform a further refinement by accurately estimating the robot's pose in the reference frame \mathbf{T}_{robot}^{ref} , which can be obtained by solving the following non-linear least squares problem:

$$\min_{\mathbf{T}_{robot}^{ref}} \sum_{i=1}^N \left(w_i \sum_{j=1}^M d_l(\mathbf{l}_i^{ref}, \mathbf{T}_{robot}^{ref} \mathbf{p}_{ij}^{robot}) \right), \quad (2)$$

where \mathbf{p}_{ij}^{robot} is the j -th laser point corresponding to the i -th reference line \mathbf{l}_i^{ref} , $d_l(\mathbf{l}, \mathbf{p})$ is the point-to-line distance, w_i is the weight for the i -th reference line.

Then we can accurately align the point cloud to the scene structure and refine the initial correspondences. Together with the estimated local pose of the robot in the scene, we can further detect objects in the point cloud to obtain the object-level semantic information.

D. OBJECT DETECTION

As required by the customizable disinfection task, the robot needs to distinguish different objects in the scene, for example, the seats and stanchions in the metro car, so that the disinfection time and distance can be configured accordingly.

For some objects like the seats, because of their relatively large size, they can be easily and accurately identified with the structure-aware segmented results. However, it is not trivial to detect small objects like the stanchions in the metro car due to their small (the diameter is ~ 3 cm) and reflective surfaces. Thus, on the one hand, valid points for these objects may be very few when it is too far away; on the other hand, the shadow effect may influence the detection when the robot is close to them. As depicted in Fig. 3, in the metro car scenario, we design a two-stage method to extract the stanchion points and remove those noisy ones: line fitting & euclidean clustering.

To coarsely select the candidate stanchion points, we first transform the laser points to current reference frame so that the points are aligned to the reference lines. Since the stanchions in a car precisely lie on a straight line, the parameters of this line can be estimated by a random sample consensus (RANSAC) based line fitting method. Most of the noisy

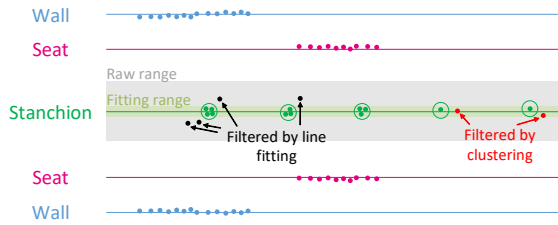


FIGURE 3. Stanchion detection based on line fitting and euclidean clustering. The laser point cloud are first transformed to the reference frame. The region in gray is used for line fitting, while the light green region is used to remove outliers in black caused by the shadow effect. The green circle shows the range threshold for adding points to the same stanchion, while those points in red are filtered because their distances to previous stanchions are not sufficient to be considered as a new stanchion.

points caused by shadow effect will be filtered through a distance threshold to the fitted line.

To further distinguish between different stanchion instances, the stanchion points are examined from near to far along the fitted line. If the point is far enough to the previous point, a new stanchion will be added to the buffer. If the consecutive points are near enough to the previous points, the point will be added to current stanchion. Otherwise it will be considered as an outlier.

After all stanchion points are detected and clustered, the position of each stanchion is computed by averaging the position of its points. Finally, the detected stanchions are transformed to the map with the robot's odometry and fused with previously recorded ones or inserted as new ones.

V. SEMANTIC MAPPING

With the segmented point cloud and detected objects in robot's frame, we can use a LiDAR-based odometry system (e.g., *Cartographer*) to provide relative motion estimation in the global map. Then with the detected hierarchical semantics (i.e., labeled points, reference lines and objects like stanchions), we can build a hierarchical map of the metro. The semantics contained by this map can be used not only for navigation, but also for the customizable and high-level disinfection tasks. As shown in Fig. 1, our hierarchical semantic map consists of four types of semantics: occupancy grids, semantic points, reference lines and stanchions.

1) Occupancy Grid Map

The occupancy grid map is the standard mapping result of common LiDAR-based SLAM system. It can be obtained by probabilistically fusing the raw laser point cloud with the robot's pose. Although it is useful for path planning and localization in general navigation tasks, it is not sufficient for customizable disinfection tasks, since only the obstacle and free space information are recorded.

2) Semantic Point Cloud

By leveraging the prior knowledge of scene structure, the raw laser point cloud can be classified into different categories

TABLE 1. Stanchion Object Definition

Property	Value	Description
Index	i	The stanchion index in the map
Radius	r	Radius of the stanchion
Centroid	(x, y)	Position of the stanchion centroid
Std.	(σ_x, σ_y)	Rolling standard deviation of the centroid's position
Points	$\{p_1, p_2, \dots, p_n\}$	Laser points of the stanchion
State	$s \in \{Initial, Candidate, Accepted, Fixed, Removed\}$	State of the stanchion
Visibility	$v \in \{True, False\}$	Whether it is visible to the robot

to represent different part of the scene. With the estimated robot's poses given by the LiDAR odometry module, the segmented point clouds are accumulated to generate the global semantic point cloud map.

3) Reference Line Map

The reference lines form a representation of the structure of the scene. With the current segmented point cloud, on the one hand, the local reference frame in which the reference lines are defined is updated to make the reference lines align to the points in a sliding window by a similar optimization as Eq. (2); on the other hand, previous added points are refined and only the inliers are reserved by a distance threshold for the reference lines, which can effectively remove those incorrectly classified points.

4) Stanchion Map

The stanchion map is built with the detected stanchions in each frame of laser point cloud. Each stanchion object has the properties illustrated in Table 1. Given the robot's pose in the map \mathbf{T}_{robot}^{map} , the centroid of the i -th stanchion \mathbf{P}_i^{map} is updated by optimizing the following problem:

$$\min_{\mathbf{P}_i^{map}} \underbrace{\sum_{m=1}^M d_p(\mathbf{P}_i^{map}, \mathbf{T}_{robot}^{map} \mathbf{P}_m^{robot})}_{\text{new-point-to-stanchion distances}} + w \underbrace{\sum_{n=1}^N d_p(\mathbf{P}_i^{map}, \mathbf{P}_n^{map})}_{\text{existing-point-to-stanchion distances}}, \quad (3)$$

where \mathbf{P}_m^{robot} is the m -th newly detected point corresponding to the i -th stanchion, and \mathbf{P}_n^{map} is the n -th existing point. The weight factor w is used to balance the influence of the new data. After the update of the centroid, the existing and newly added points will be refined with the stanchion's radius. Also, the maximum number of points is restricted.

The most essential part of stanchion mapping is the stanchion state management as shown in Fig. 4. The stanchions successfully detected in the first frame are set as *Initial*. A newly detected stanchion will be set as *Candidate* if there are not enough valid points. If the number of valid

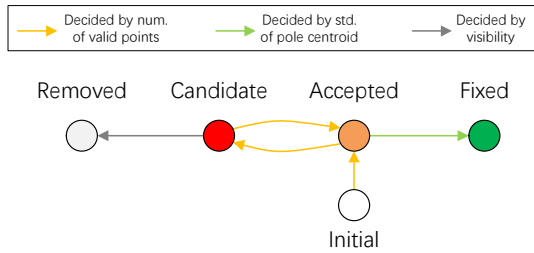


FIGURE 4. The stanchion state transition diagram. The start state of a new stanchion can be *Initial* at the first frame, or *Candidate* after initialized. Different arrow colors represent different conditions to change the stanchion state.

points is over a threshold, the stanchion will be considered as *Accepted*. The standard deviation of the centroid is monitored to change the stanchion’s state from *Accepted* to *Fixed*. If the centroid of the stanchion is stable enough in a sliding window, its position will be fixed, thus the mapping process will become more robust to noisy stanchion points caused by the shadow effect. If the stanchion is invisible to the robot and its state is *Candidate*, it will be considered as an outlier and marked as *Removed*.

VI. EXPERIMENTS

A. DATA COLLECTION & EVALUATION METRICS

The dataset used for the evaluation of our framework was collected on a Metro-Cammell train (M-train) of Hong Kong Mass Transit Railway (MTR), which is the oldest and most widely used trains in Hong Kong. Two sequences of laser scan data were recorded with a SICK TiM561 LiDAR sensor working at 15 Hz of scanning frequency. There are totally four different types of cars in the data sequences. Each of the cars is about 20 meters long. The summarized information of the dataset is shown in Table 2.

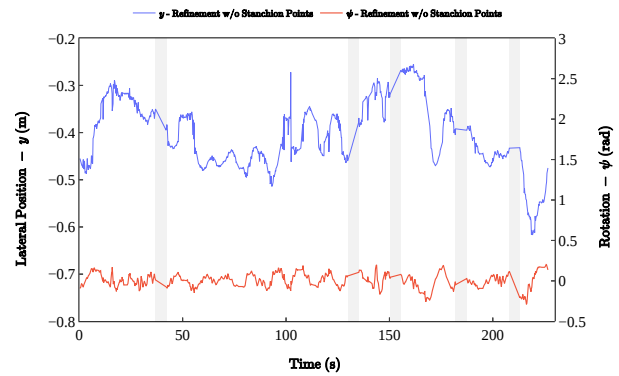
TABLE 2. Statistics of Testing Data

Sequence Num.	Car Num.	Stanchion Num.	Length (m)	Duration (s)	Laser Scans
2	4	63	90.48	538	8098

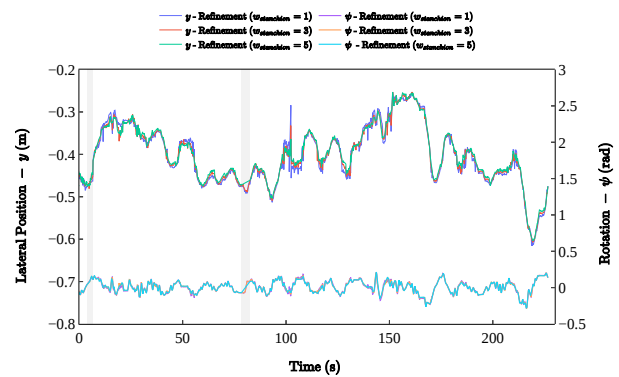
The evaluation metrics for proposed semantic mapping system are mainly based on the the measurements of the metro car structure, such as the distances between stanchions and the length of cars.

B. DATA ASSOCIATION & REFINEMENT

We first evaluate the performance of the data association and refinement. Since the data association is directly determined by the pose estimation result in local reference frame (i.e., the frame in which the reference lines are defined), we can plot the local localization results along the robot’s movement. As the main advantage of point cloud segmentation is to assign different weights for the points with different labels, we have specially compared the localization results without or with



(a) Without stanchion points



(b) With different weights for stanchion points

FIGURE 5. The estimated robot poses w.r.t the reference frame in the refinement. The gray regions show the failures of data association due to insufficient point-to-line correspondences. The appropriate addition of stanchion points significantly improves the successful rate of refinement.

stanchion points with different weights in the optimization for refinement, as shown in Fig. 5.

To determine the success of data association, only those refined laser points that fall within 5cm of the reference lines are considered valid. The minimum number of valid points for a successful refinement is 50. In Fig. 5, the short-cuts in gray region mean the failure of refinement, which can demonstrate the robustness of different configurations of the data association and refinement process.

Obviously, the addition of stanchion points in the optimization can effectively improve the robustness of the data association process. As shown in Fig. 5(b), a larger weight ($w = 3$) can result in better and more stable performance than a smaller weight ($w = 1$). This is mainly because that the stanchions are good landmarks in the metro which evenly distribute and maintain a proper distance with each other in the metro cars.

However, from the figure we can also find that the weight for stanchion points should not be set too large ($w = 5$), otherwise it will decrease the stability of the pose estimation. The cause of this situation may be the possible noisy measurements of the stanchions, since they are too thin to be accurately measured by the LiDAR sensor. Moreover,

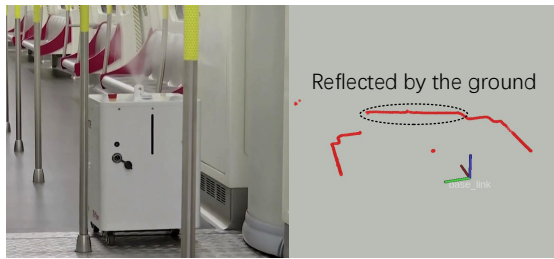


FIGURE 6. The situation of localization failure in the reference frame when only stanchion points are used in the fine pose estimator. The uneven ground of the car joint will cause the invalid measurements of laser scanner.

the situation shown in Fig. 6 can also cause the failure of localization, which can be identified in the data association process.

C. SEMANTIC MAPPING

To compare the performance of different popular LiDAR mapping frameworks in our scene and their influences on our semantic mapping results, we first evaluate the conventional occupancy grid maps generated by different frameworks. Then the semantic maps generated by our proposed framework with different LiDAR odometry modules are compared.

1) Evaluation of the occupancy grid maps

To evaluate the quality of the occupancy grid maps constructed in the our testing scenarios, the *GMapping*¹, *Hector Mapping*² and *Cartographer*³ are adopted for comparisons. Since the *GMapping* additionally needs the odometry of the robot as input, the *laser_scan_matcher*⁴ ROS package was used for it.

In Fig. 7, the mapping results are listed for qualitative comparison. It should be noted that we have chosen the map of *Cartographer* as the reference, and the other two maps are manually aligned to it with the first stanchion in the map. Vertical reference lines in the figure are added for better comparison. As shown by the red arrows, the map of *Hector Mapping* has the largest difference, while the map of *GMapping* remains an almost stable bias. These errors are mainly caused by inaccurate scan matching, for example, the situation in Fig. 6. The map of *Cartographer* achieves the best performance and no obvious drift or error occurs.

2) Evaluation of the semantic maps

To quantitatively evaluate our semantic maps as Fig. 8, we focus on the positions of stanchions in the map, since the distance between two adjacent stanchions can be conveniently and accurately measured.

The distances between adjacent stanchions estimated with different LiDAR odometry modules are demonstrated in

¹<http://wiki.ros.org/gmapping>

²http://wiki.ros.org/hector_mapping

³<https://google-cartographer-ros.readthedocs.io/en/latest/>

⁴http://wiki.ros.org/laser_scan_matcher

Fig. 9. There are three clusters corresponding to 1.00m, 1.75m and ~ 2.25 m respectively. The third cluster is related to the stanchion in the joint of cars, so the distances may be slightly different. As shown in the figure, the distances estimated by *Cartographer* have achieved the best performance compared with the other two methods. It should be noticed that the first two points in the second cluster are related to the first and last stanchion in the data sequence, therefore, their positions are not as accurate as the other stanchions due to lack of measurements.

TABLE 3. The RMSE of estimated distances of adjacent stanchions and the length of metro car with different LiDAR odometry modules in our framework

LiDAR odometry	RMSE of distance		RMSE of
	between adjacent stanchions (m)		car length (m)
	$d = 1.00\text{m}$	$d = 1.75\text{m}$	$L = 18.0\text{m}$
GMapping	0.071	0.075	0.179
Hector Mapping	0.031	0.068	0.099
Cartographer	0.018	0.043	0.064

Table 3 lists the root mean square error (RMSE) of the estimated distances between adjacent stanchions as well as the length of the metro car (which is defined as the distance between the first and the last stanchion in a car). The results consistently demonstrate that *Cartographer* outperforms the other two frameworks in our scene.

D. RUNTIME ANALYSIS

The runtimes for the main components in our framework are shown in Fig. 10, which were tested on a laptop with an Intel Core i7-8750H CPU @ 2.20 GHz. The average runtimes of the data association and refinement are 0.493ms and 1.102ms, which makes the average rate of the point cloud segmentation subsystem over 600Hz. The stanchion detection process is very efficient by leveraging the prior knowledge of the structure, which only spends less than 0.3ms in average.

It should be noted that the runtime of updating the map will burst when the local point cloud and reference lines are refined, or when the local map is merged with the global map before the robot enters into a new car.

E. CUSTOMIZABLE DISINFECTON TASK

To support the customizable disinfection task, the robot needs to understand the structure of the metro car around it. With the segmented point cloud in our map, the structure label can be decided by accumulating the point cloud in a local moving window around the robot. Fig. 11 shows the extracted structure labels on the left and right side of the robot when it is moving in a the metro car.

Based on the semantic level understanding of the scene, the navigation waypoints, the disinfection distance and the disinfection time can be determined according to different structures of the metro car in the customizable disinfection task, as shown in Fig. 12:

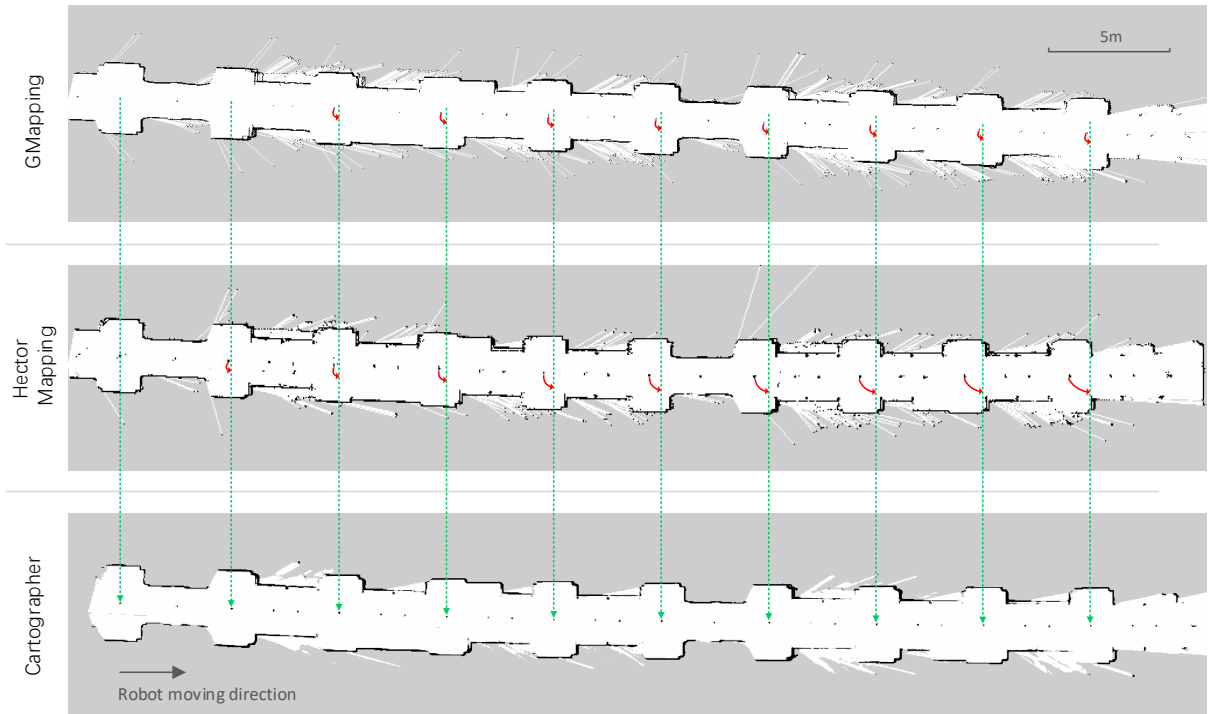


FIGURE 7. Comparisons of the grid maps generated by different LiDAR odometry modules. The grid maps are aligned by the first common stanchion (at the left of the figure). The green dotted vertical lines take our map as the reference to compare the differences between each map. The red arrows point out the obvious differences of GMapping and Hector Mapping compared with Cartographer.

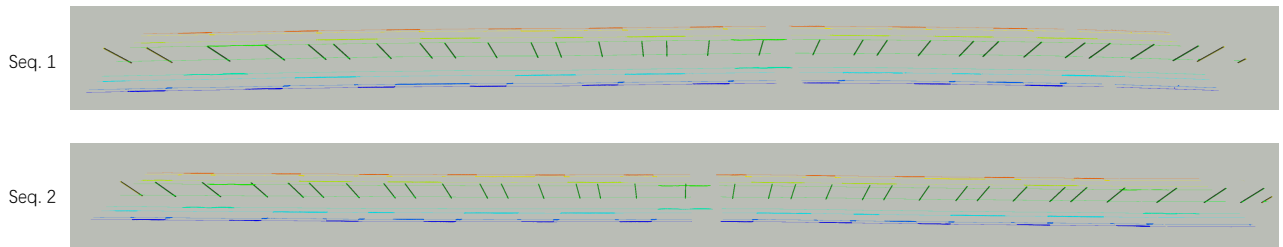


FIGURE 8. The hierarchical semantic map generated with our framework. The legends are the same as those in Fig. 2.

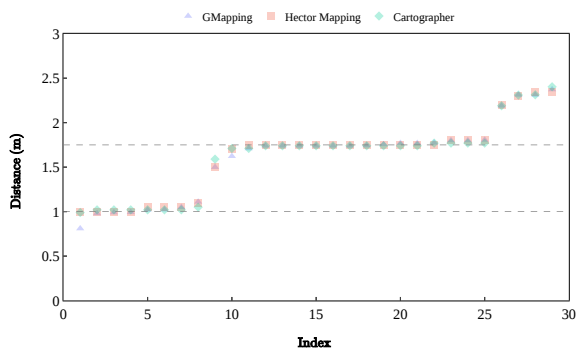


FIGURE 9. The distances between adjacent stanchions in the maps estimated with different LiDAR odometry modules.

- The navigation waypoints can be set according to the positions of stanchions in the map. For example, in Fig. 12(a), the waypoints are placed between the seats

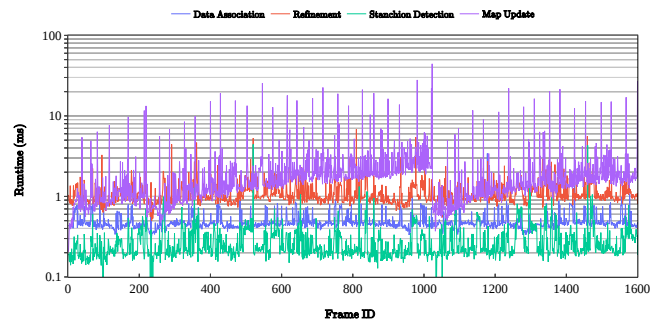


FIGURE 10. The runtime of each component in our framework.

and the stanchions and aligned with the position of the stanchions. Additional waypoints are also inserted into the adjacent stanchions with a spacing of 1.75m. Other possible options for setting the waypoints could choose the seats as the reference.

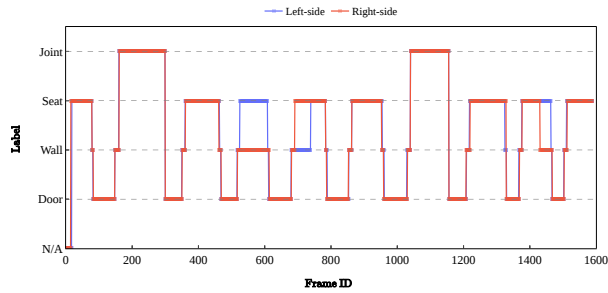


FIGURE 11. The structure labels identified by the robot when moving in the metro car, given the semantic point cloud constructed by the mapping module.

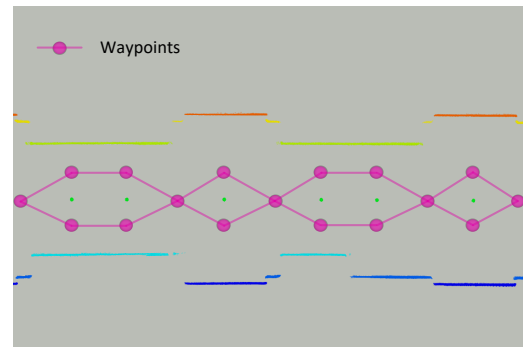
- The effective disinfection distance for different structures such as the seats and stanchions can be configured accordingly. Taking Fig. 12(b) as an example, to ensure the effectiveness of disinfection, the robot must enter a specific range of the object to be disinfected. These ranges will affect the robot's local path planning together with the navigation waypoints.
- The disinfection time can be adjusted by setting the maximum velocity of the robot in different region. As shown in Fig. 12(c), when passing by the stanchions or seats, the robot should slow down or even stop for a while to enhance the disinfection results in these high-risk regions. For other regions, the robot can move faster to reduce the total time consumption of the disinfection task.

VII. CONCLUSIONS

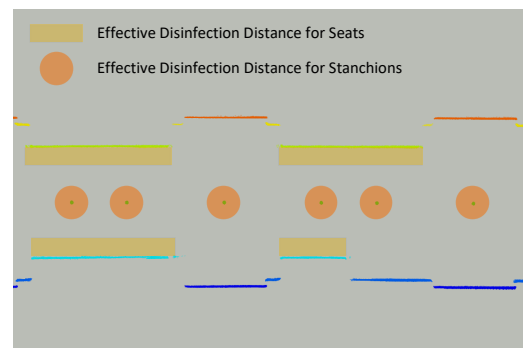
In this paper, we presented a semantic mapping framework for customizable disinfection tasks which can be used in structured scenarios. The structure information about the scene is used in the point cloud segmentation as the prior knowledges. A hierarchical semantic map, including the segmented point cloud, structure reference lines and objects, can be built to provide a semantic level understanding of the scene for the robot. By leveraging such semantic information, our framework can be effectively deployed to customizable disinfection applications in the metro and efficiently run at real-time. In the future work, we will try to deploy our system to other similar scenarios, such as buses and trains, and further explore the potential usage of visual sensors to capture more semantic information from the 3D structure, which can be an alternative to our LiDAR-based perception system.

REFERENCES

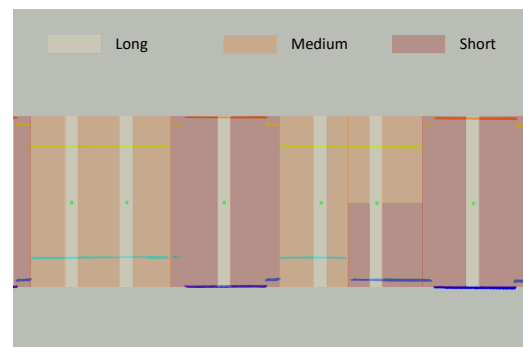
- [1] G.-Z. Yang, B. J. Nelson, R. R. Murphy, H. Choset, H. Christensen, S. H. Collins, P. Dario, K. Goldberg, K. Ikuta, N. Jacobstein, D. Kragic, R. H. Taylor, and M. McNutt, "Combating COVID-19—The role of robotics in managing public health and infectious diseases," *Sci. Robot.*, vol. 5, no. 40, Mar. 2020.
- [2] R. R. Murphy, "Robots and pandemics in science fiction," *Sci. Robot.*, vol. 5, no. 42, May 2020.
- [3] M. Lindblad, E. Tano, C. Lindahl, and F. Huss, "Ultraviolet-C decontamination of a hospital room: Amount of UV light needed," *Burns*, vol. 46, no. 4, pp. 842–849, Jun. 2020.
- [4] B. McEvoy and N. J. Rowan, "Terminal sterilization of medical devices using vaporized hydrogen peroxide: A review of current methods and



(a) Customizable waypoints



(b) Customizable disinfection distance



(c) Customizable disinfection time

FIGURE 12. The customizable disinfection task based on the semantic level understanding of the scene.

- emerging opportunities," *J. Appl. Microbiol.*, vol. 127, no. 5, pp. 1403–1420, 2019.
- [5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
- [6] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Nov. 2011, pp. 155–160.
- [7] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1271–1278.
- [8] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based Semantic SLAM," in *Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 4530–4537.
- [9] J.-H. Im, S.-H. Im, and G.-I. Jee, "Extended Line Map-Based Precise

- Vehicle Localization Using 3D LIDAR,” *Sensors*, vol. 18, no. 10, p. 3179, Oct. 2018.
- [10] K. Lenac, A. Kitanov, R. Cupec, and I. Petrović, “Fast planar surface 3D SLAM using LIDAR,” *Robotics and Autonomous Systems*, vol. 92, pp. 197–220, Jun. 2017.
- [11] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, “SLAM++: Simultaneous Localisation and Mapping at the Level of Objects,” in *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013, pp. 1352–1359.
- [12] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, “Probabilistic data association for semantic SLAM,” in *Proceedings of 2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1722–1729.
- [13] J. G. Rogers, A. J. B. Trevor, C. Nieto-Granda, and H. I. Christensen, “Simultaneous localization and mapping with learned object recognition and semantic data association,” in *Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2011, pp. 1264–1270.
- [14] R. Spangenberg, D. Goehring, and R. Rojas, “Pole-based localization for autonomous vehicles in urban scenarios,” in *Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 2161–2166.
- [15] M. Sefati, M. Daum, B. Sundermann, K. D. Kreisköther, and A. Kampker, “Improving vehicle localization using semantic and pole-like landmarks,” in *Proceedings of 2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 13–19.
- [16] C. You, C. Wen, C. Wang, J. Li, and A. Habib, “Joint 2-D–3-D Traffic Sign Landmark Data Set for Geo-Localization Using Mobile Laser Scanning Data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2550–2565, Jul. 2019.
- [17] U. Weiss and P. Biber, “Plant detection and mapping for agricultural robots using a 3D LIDAR sensor,” *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 265–273, May 2011.
- [18] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, “SLOAM: Semantic Lidar Odometry and Mapping for Forest Inventory,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 612–619, Apr. 2020.
- [19] M. Karg, K. M. Wurm, C. Stachniss, K. Dietmayer, and W. Burgard, “Consistent mapping of multistory buildings by introducing global constraints to graph-based SLAM,” in *Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 5383–5388.
- [20] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, “Robust LiDAR-based localization in architectural floor plans,” in *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 3318–3324.
- [21] F. Boniardi, A. Valada, R. Mohan, T. Caselitz, and W. Burgard, “Robot Localization in Floor Plans Using a Room Layout Edge Extraction Network,” in *Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 5291–5297.
- [22] M. P. Parsley and S. J. Julier, “Towards the exploitation of prior information in SLAM,” in *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2010, pp. 2991–2996.
- [23] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, “Large scale graph-based SLAM using aerial images as prior information,” *Auton. Robots*, vol. 30, no. 1, pp. 25–39, Jan. 2011.
- [24] A. Pretto, S. Aravecchia, W. Burgard, N. Chebrou, C. Dornhege, T. Falck, F. V. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, F. Liebis, P. Lottes, A. Milioto, D. Nardi, S. Nardi, J. Pfeifer, M. Popovic, C. Potena, C. Pradalier, E. Rothacker-Feder, I. Sa, A. Schaefer, R. Siegwart, C. Stachniss, A. Walter, W. Winterhalter, X. Wu, and J. Nieto, “Building an Aerial-Ground Robotics System for Precision Farming: An Adaptable Solution,” *IEEE Robot. Autom. Mag.*, 2020.
- [25] T. Daoust, F. Pomerleau, and T. D. Barfoot, “Light at the End of the Tunnel: High-Speed LiDAR-Based Train Localization in Challenging Underground Environments,” in *Proceedings of 2016 Conference on Computer and Robot Vision (CRV)*, Jun. 2016, pp. 93–100.
- [26] M. Hsiao, E. Westman, and M. Kaess, “Dense Planar-Inertial SLAM with Structural Constraints,” in *Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6521–6528.
- [27] Z. Ren, L. Wang, and L. Bi, “Robust GICP-Based 3D LiDAR SLAM for Underground Mining Environment,” *Sensors*, vol. 19, no. 13, p. 2915, Jan. 2019.
- [28] M. Guettari, I. Gharbi, and S. Hamza, “UVC disinfection robot,” *Environ Sci Pollut Res*, Oct. 2020.
- [29] M. Hui, “Hong Kong’s subway is sending robots to disinfect trains of coronavirus,” <https://qz.com/1816762/coronavirus-hong-kongs-mtr-subway-uses-robot-to-disinfect-trains/>, Mar. 2020.
- [30] K. Cresswell and A. Sheikh, “Can Disinfection Robots Reduce the Risk of Transmission of SARS-CoV-2 in Health Care and Educational Settings?” *J. Med. Internet Res.*, vol. 22, no. 9, p. e20896, 2020.
- [31] D. Hu, H. Zhong, S. Li, J. Tan, and Q. He, “Segmenting areas of potential contamination for adaptive robotic disinfection in built environments,” *Building and Environment*, vol. 184, p. 107226, Oct. 2020.
- [32] D. Conte, S. Leamy, and T. Furukawa, “Design and Map-based Tele-operation of a Robot for Disinfection of COVID-19 in Complex Indoor Environments,” in *Proceedings of 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Nov. 2020, pp. 276–282.



ZHENZHEN XIANG received the B.S. and M.S. degree from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with Shanghai Jiao Tong University, Shanghai, China. His research interests include semantic scene understanding, vision and LiDAR based SLAM, and mobile robot navigation.



JIANBO SU (SM'04) received the B.S. degree in automatic control from Shanghai Jiao Tong University, Shanghai, China, in 1989, the M.S. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Science, Beijing, China, in 1992, and the Ph.D. degree in control science and engineering from Southeast University, Nanjing, China, in 1995. He joined the Faculty of the Department of Automation, Shanghai Jiao Tong University, in 1997,

where he has been a Full Professor, since 2000.

His research interests include robotics, pattern recognition, and human-machine interaction. Dr. Su is a member of the Technical Committee of Networked Robots, IEEE Robotics and Automation Society, a member of the Technical Committee on Human Machine Interactions, IEEE System, Man, and Cybernetics Society, and a Standing Committee Member of the Chinese Association of Automation. He has served as an Associate Editor for the IEEE Transactions on Cybernetics, since 2005, with which he received the Best Associate Editor Award from IEEE SMC society, in 2014.

...