Majority Voting-Based Update of Point-Cloud Map Using LiDAR Information from Multiple Vehicles

Saki Fujisawa
Graduate School of Science and Engineering
Doshisha University
Kyotanabe, Kyoto, Japan
e-mail: ctwn0104@mail4.doshisha.ac.jp

Masafumi Hashimoto, Kazuhiko Takahashi
Faculty of Science and Engineering
Doshisha University
Kyotanabe, Kyoto, Japan
e-mail: {mhashimo, katakaha}@mail.doshisha.ac.jp

***Abstract*—This paper presents a crowd-sensing-based method of updating three-dimensional (3D) point-cloud maps in road environments. This method utilizes data from Light Detection and Ranging (LiDAR) mounted on multiple vehicles. Each vehicle maps its 3D LiDAR point-cloud data onto a two-dimensional (2D) local grid map. The state of each cell in the local grid map—unchanged (occupied or empty), changed (occupied or empty), or undetermined—is classified through majority voting after comparing the LiDAR point-cloud data with the original reference point-cloud map. The 2D local grid map containing the cell states is then sent to a central server, which collects the 2D local grid maps from all vehicles into a 2D global grid map and determines the final cell states through another majority-voting process. Finally, the 3D point-cloud map is updated based on the cell states in the 2D global grid map. This two-stage majority voting method enables simple yet robust map updates. Simulation experiments demonstrate the effectiveness of the map-update method.**

***Keywords*—LiDAR; map update; point-cloud map; crowd-sensing; majority voting.**

I. INTRODUCTION

Vehicles in Intelligent Transportation Systems (ITS) are increasingly equipped with autonomous driving capabilities and advanced driving-assistance systems [1]. In addition, last-mile delivery automation using delivery robots and drones is being actively pursued to support advanced logistics systems [2]. A key technology underlying these systems is environmental mapping, which assists self-localization and environment perception by vehicles and mobile robots [3].

Environmental mapping in ITS typically relies on Mobile Mapping Systems (MMSs) [4], which integrate high-precision sensors, such as Light Detection And Ranging (LiDAR), cameras, inertial navigation systems, and global navigation satellite systems. However, because MMSs are limited in number and cannot cover all road environments nationwide, most maps are built only for major areas, such as expressways, freeways, and primary urban roads. To compensate for the limitations of MMS-based mapping, a mapping method based on Simultaneous Localization and Mapping (SLAM) using cameras and LiDARs mounted on ordinary vehicles has been widely studied [5][6].

Typical SLAM technologies include Normal Distributions Transform (NDT) SLAM [7] and iterative closest point SLAM [8]. Previously, we proposed a method that builds three-dimensional (3D) environmental point-cloud maps based on LiDAR NDT SLAM in dynamic road environments, along with a method that merges multiple environmental maps [9].

Accurate environmental maps are essential for vehicle self-localization and environment perception. When the environment changes—for example, due to new construction or the removal of existing buildings—the accuracy of self-localization and environment perception will deteriorate. Updating the environmental map in response to changes in environmental features, such as buildings and traffic signs, is an important challenge [10].

This paper presents a method that updates 3D environmental point-cloud maps built using LiDAR NDT SLAM. In most conventional map-update methods, the point cloud in the environmental map is matched to the current LiDAR point-cloud data, and points with large matching errors are identified as change points for map update [11][12]. However, because 3D mechanical LiDAR commonly used in automotive applications produces vertically sparse point-cloud data, it is difficult to detect changes in distant environmental features. In addition, in dynamic road environments, environmental features are often occluded by other vehicles or structures, making reliable map updates difficult when relying solely on LiDAR data from a single vehicle.

In real road environments, many vehicles travel in different lanes and at different times. To overcome the limitations of single-vehicle LiDAR-based map updates, researchers have proposed crowd-sensing-based methods that aggregate the sensor information from multiple vehicles (crowd-sourcing vehicles) on a central server to update the map [13][14][15]. For example, Kim et al. [15] updated a LiDAR point-cloud environmental map by mapping vehicle-mounted LiDAR point-cloud data onto a voxel map (3D grid map). The current LiDAR point-cloud data were compared with the environmental map and changes were detected based on evidence theory. The map changes collected from the crowd-sourcing vehicles were reproduced on a map-cloud server based on similarity-based clustering, and the environmental maps were updated. Although this method represents a theoretical advancement, its effectiveness seems difficult to demonstrate using vertically sparse point-cloud data from mechanical 3D LiDAR.

Inspired by the work of Kim et al. [15], this paper presents a crowd-sensing-based method for updating 3D environmental point-cloud maps. To address the vertical sparsity of mechanical LiDAR, the 3D LiDAR point-cloud data from each vehicle are mapped onto a two-dimensional (2D) grid map. By comparing

 ISBN: 978-1-68558-399-6

the current LiDAR point-cloud data with the environmental map, changes in the grid-cell states are detected through majority voting. The 2D grid map containing the cell states is sent to a central server, which collects and fuses the 2D grid maps from multiple vehicles and determines the final cell states through another majority-voting process. The 3D environmental point-cloud map is then updated based on the cell states. Compared with the method of Kim et al. [15], the proposed method employs a simpler voting-based framework using 2D grid maps instead of evidence-based theory using 3D voxel maps, thereby reducing computational complexity.

The remainder of this paper is structured as follows. Section II overviews the proposed map-update method. Sections III and IV describe the map-update processes on the vehicle side and the central server side, respectively. Section V presents simulation experiments for a three-vehicle system, and Section VI presents the main conclusions of this study.

II. OVERVIEW OF ENVIRONMENTAL MAP UPDATE

The 3D mechanical LiDAR used in this study is a 32-layer mechanical LiDAR (Velodyne HDL-32E) with horizontal and vertical Fields-Of-Views (FOVs) of 360° and 41.34°, respectively, and angular resolutions of 0.16° and 1.33°, respectively. During each 0.1 s sampling period (or scan), the laser beams complete one full rotation in the horizontal direction. LiDAR acquires 70,000 point-cloud points per scan.

Our method updates the NDT-based environmental map. The 3D LiDAR point-cloud data are stored on a 3D grid map composed of cubic voxels with a side length of 0.3 m. For map-update process, the 3D LiDAR point-cloud data captured in each scan (referred to as the new LiDAR point-cloud data) and the previously built 3D point-cloud map (referred to as the environmental map) are projected onto a 2D grid map on a horizontal plane. Each cell in the 2D grid map is a square with a side length of 0.3 m.

Figure 1 illustrates the map-update flow. The environmental map is updated by comparing it with the new LiDAR point-cloud data obtained on each vehicle. The new LiDAR point-cloud data contain the data from stationary and moving objects, such as cars, two-wheelers, and pedestrians. Because the map-update process requires only stationary-object information, LiDAR data corresponding to moving objects are removed using the occupancy grid-based method [9].

The NDT map-matching method aligns the new LiDAR point-cloud data with the environmental map and estimates the vehicle's self-pose. Based on this estimated pose, both the environmental map and the new LiDAR point-cloud data are mapped onto a 2D local grid map. By comparing the new LiDAR point-cloud data with the environmental map in the 2D local grid map through majority voting, each cell is classified into one of five states: *unchanged* (*occupied* or *empty*), *changed* (*occupied* or *empty*), or *undetermined*. The 2D local grid map, embedded with these cell states, is sent to a central server, which collects the 2D local grid maps from multiple vehicles and determines the final cell states through majority voting. Finally, the 3D environmental point-cloud map is updated based on the cell states.

In this study, the 3D LiDAR point clouds are mapped onto a 2D grid map to mitigate the vertical sparsity inherent in mechanical 3D LiDAR, simplify the map-update process, and reduce computational cost. However, because height information is discarded, the map-update accuracy may decrease for environmental features with significantly different vertical structures.

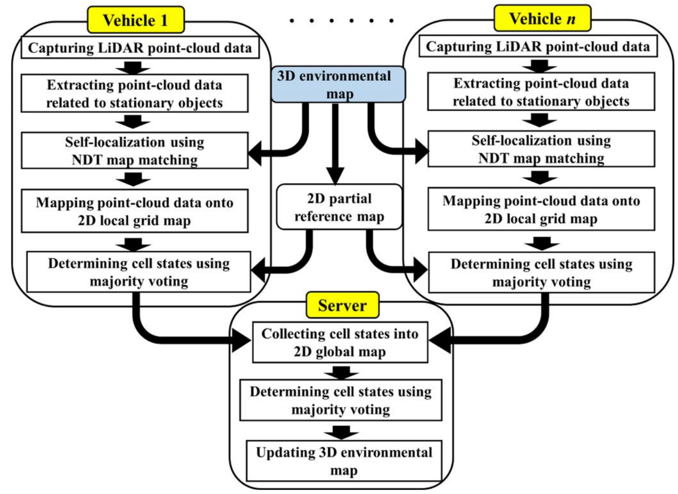


Figure 1. Overview of the map-update procedure.

III. CELL-STATE DETERMINATION OF 2D LOCAL GRID MAP ON VEHICLE

A. Classification of Cell State

As shown in Figure 2 (a), a square area with a side length of 80 m, centered at the self-position (V_i) of the i-th vehicle (i = 1, 2, ...), is extracted from the environmental map in the world coordinate frame. The extracted area is referred to as the partial reference map R_i. The new LiDAR point-cloud data obtained by the i-th vehicle at V_i is denoted as S_i.

As illustrated in Figure 2 (b), both the partial reference map R_i and the new LiDAR point-cloud data S_i are projected onto the same 2D local grid map, which consists of square cells with a side length of 0.3 m. The presence or absence of point-cloud data in R_i and S_i is compared on a cell-by-cell basis to detect environmental changes. Each cell is then assigned to one of five classes, as defined in Figure 3.

・Unchanged (occupied): The cell contains both partial reference-map data and new LiDAR point-cloud data

・Unchanged (empty): The cell contains neither partial reference-map data nor new LiDAR point-cloud data.

・Changed (occupied): The cell contains new LiDAR point-cloud data but no partial reference-map data.

・Changed (empty): The cell contains partial reference-map data but no new LiDAR point-cloud data.

・Undetermined: The new LiDAR point-cloud data in the cell are occluded and therefore unavailable.

Because of the limited vertical FOV of LiDAR, upper positions of tall environmental features may not be observed. In such cases, the states of those regions cannot be reliably determined. Therefore, cells in which the height of objects in

 ISBN: 978-1-68558-399-6

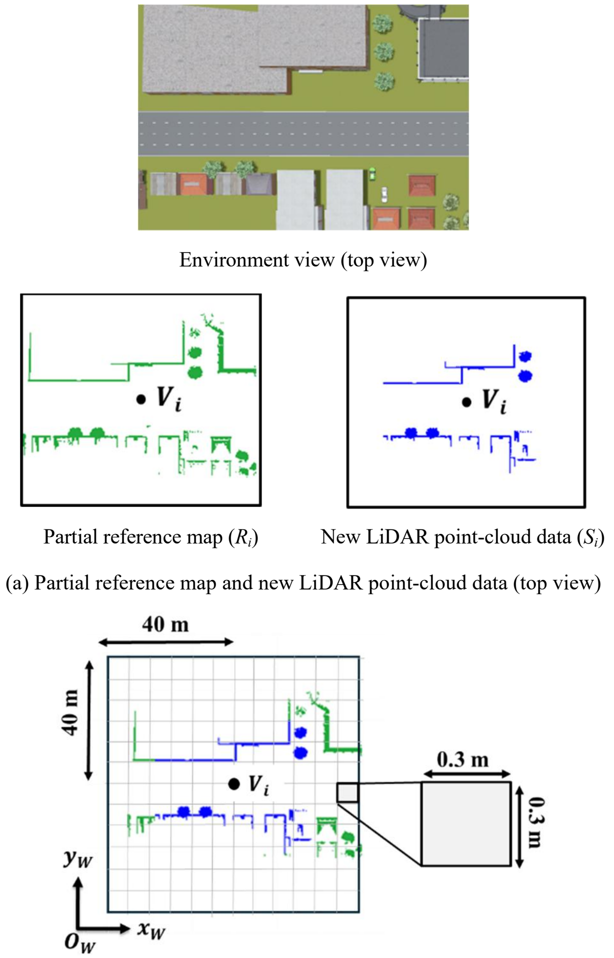


Figure 2. Mapping of a partial reference map and new LiDAR point-cloud data onto a 2D local grid map (top view).

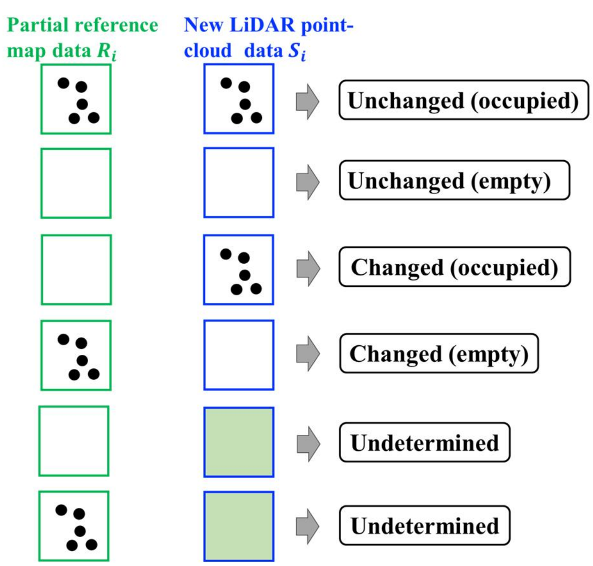


Figure 3. Classification of cell states. Cells with black circles (LiDAR measurement points) indicate occupied cells; the white and green cells indicate empty and occluded cells, respectively.

the environmental map exceeds that of the new LiDAR point-cloud data are treated as invalid cells and remain unclassified.

B. Determination of Cell State

The state of each cell in the 2D local grid map is classified at every LiDAR scan and determined through majority voting. For example, if the LiDAR point-cloud data are mapped onto a cell during ten scans and the cell state is classified as *unchanged* (*empty*) in four scans and *changed* (*empty*) in six scans, the final state is voted as *changed* (*empty*). If the cell state is classified as *unchanged* (*empty*) in five scans and *changed* (*empty*) in five scans, the final state is voted as *undetermined*.

When many measurement points of new LiDAR point-cloud data are mapped onto one cell, that cell is more likely to contain an object than cells with fewer measurement points. Therefore, instead of assigning one vote per LiDAR scan for the five cell classes, occupied cells are assigned multiple votes per LiDAR scan. The number of votes per scan is determined by the following exponential function (see Figure 4):

$$w(n) = a + (b - a)(1 - e^{-k(n - n_{\min})}) \tag{1}$$

where n denotes the number of measurement points of the new LiDAR point-cloud data in an occupied cell, and n_{min} (set to 5 in this study) is the minimum number of measurement points required to classify a cell as occupied. Below this threshold, a cell cannot be classified as occupied. Parameters a and b represent the lower and upper bounds of the vote weight, respectively, while k controls the rate of increase. In the experiments (Section V), we set $a = 2$, $b = 3$, and $k = 0.5$ based on preliminary results.

In summary, majority voting assigns one vote per LiDAR scan for *unchanged* (*empty*), *changed* (*empty*), and undetermined cells, and multiple votes per LiDAR scan—computed by (1)—for *unchanged* (*occupied*) and *changed* (*occupied*) cells.

Table I presents exemplary cell states determined by the three vehicles. For vehicle 1, the cell is classified as *changed* (*occupied*) in 25 scans and *undetermined* in 20 scans. According to (1), the cell is voted as *changed* (*occupied*) in 33 votes and *undetermined* in 20 votes. Thus, the final state is *changed* (*occupied*). Similarly, the final states for vehicles 2 and 3 are *changed* (*occupied*) and *undetermined*, respectively.

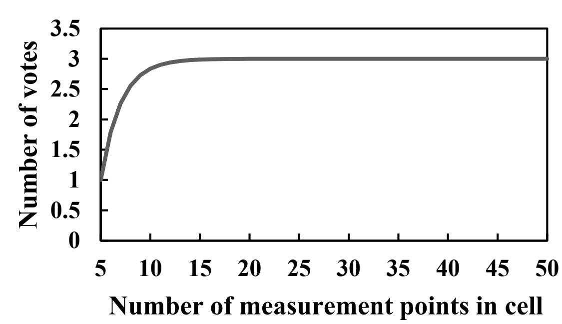


Figure 4. Number of votes versus number of measurement points in an occupied cell

TABLE I. EXAMPLE OF NUMBER OF CELL STATES AND DECISION BY THREE VEHICLES

	Unchanged (occupied)	Unchanged (empty)	Changed (occupied)	Changed (empty)	Undetermined	Decision
Vehicle 1	0	0	25	0	20	Changed (occupied)
Vehicle 2	0	0	25	0	20	Changed (occupied)
Vehicle 3	5	0	18	0	22	Undetermined

IV. CELL-STATE DETERMINATION OF 2D GLOBAL GRID MAP ON CENTRAL SERVER

Each vehicle generates a 2D local grid map containing its estimated cell states. The local grid maps from multiple vehicles are then collected into a 2D global grid map on the central server. Each cell on the global grid map is a 0.3 m × 0.3 m square.

The state of each cell on the 2D global grid map is determined through majority voting based on the "one vote per vehicle" principle. For example, in Table I, two vehicles classify a cell as *changed* (*occupied*), and one vehicle classifies it as *undetermined*. Consequently, the central server determines the final state as *changed* (*occupied*).

For cells classified as *unchanged* (*occupied*) or *undetermined*, the point-cloud data are retained in the reference map. For cells classified as *changed* (*occupied*), the reference map is updated using the new LiDAR point-cloud data. For cells classified as *changed* (*empty*), the LiDAR data occupying the cell are deleted from the reference map. In *undetermined* cells where multiple classes receive equal votes, the point-cloud data are retained in the reference map.

This paper introduces a two-stage majority voting method, in which both the vehicles and the central server perform majority voting. Each vehicle decides its cell states in the 2D local grid map and passes them to the central server, which then examines the cell states reported by all vehicles to finalize the decision. A one-stage majority voting scheme performed solely by the central server is also conceivable. In this scheme, each vehicle would send the vote counts for five classes to the central server. The central server would determine the final cell states by aggregating these votes.

When vehicles travel in different lanes within the same environment, each vehicle decides different cell states in the 2D local grid map. The two-stage majority voting method reduces bias in the number of cell states determined by each vehicle. Furthermore, it reduces communication costs between vehicles and the central server compared to the one-stage majority voting method. For these reasons, this paper applies the two-stage majority voting method. The following section compares the performances of the one-stage and two-stage majority voting methods.

V. SIMULATION EXPERIMENTS

Simcenter Prescan (Siemens) [15] is used as the simulator to generate vehicle motion and the associated LiDAR point-cloud data in a dynamic road environment. The simulation environment includes many moving objects, such as cars and pedestrians. Figure 5 is an example of the LiDAR point-cloud data generated by the simulator.

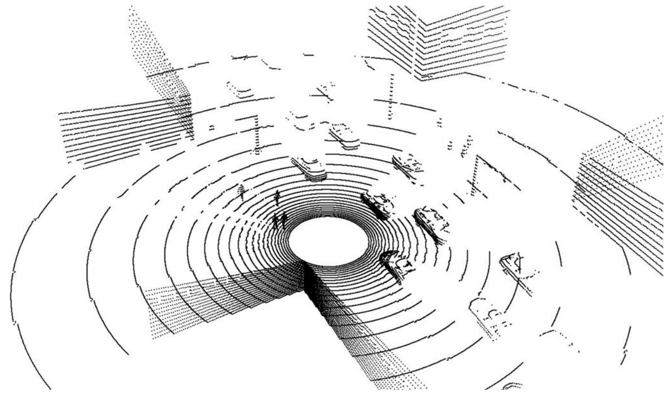


Figure 5. Example of LiDAR point-cloud data in an intersection environment (bird's-eye view).

Figure 6 (a) illustrates an urban road environment with two lanes in each side (four lanes in total), and Figure 6 (b) shows the built reference map. Three vehicles travel in different lanes and directions at different times; specifically, vehicles 1 and 2 travel from west to east at 40 km/h, and vehicle 3 travels from east to west at 40 km/h. Four areas have changed since the reference map was built. The changes include the construction of one building in area 1, the demolishment of two buildings in area 2, the construction of two buildings in area 3, and the demolishment of six buildings in area 4.

First, the effectiveness of the proposed map-update method is qualitatively demonstrated in areas 1 and 2 in Figure 6 (a). Figures 7 (a) and (b) show views of area 1 when the environmental map was built and at the time the three vehicles passed, respectively. Building 1, which does not exist in Figure 7 (a), appears in Figure 7 (b). The LiDAR point-cloud data of building 1, absent from the environmental map (Figure 7 (c)), are successfully added to the updated map (red dots in Figure 7 (d)).

In area 2, buildings 2 and 3 had been demolished by the time the three vehicles passed the site (Figure 8 (b)). Although both buildings exist in the environmental map (red dots in Figure 8 (c)), their LiDAR point-cloud data are removed in the updated map (Figure 8 (d)).

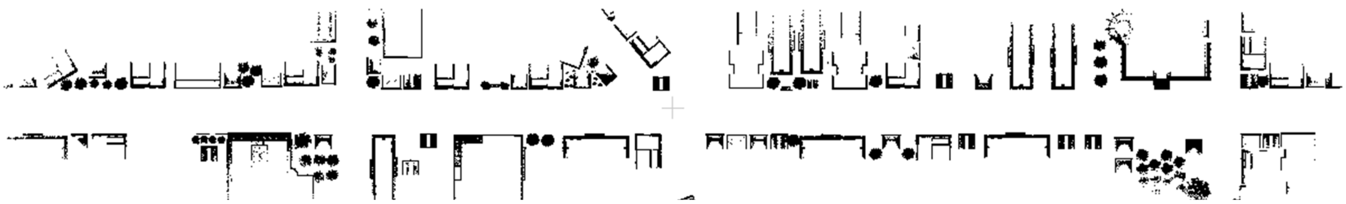
The map-update accuracy over the entire environment shown in Figure 6 is quantitatively evaluated in the following five cases.

・Case 1: Two-stage majority voting, applying the "multiple votes per LiDAR scan" principle on each vehicle and the "one vote per vehicle" principle on the central server.

・Case 2: Two-stage majority voting, applying the "one vote per LiDAR scan" principle on each vehicle and the "one vote per vehicle" principle on the central server.

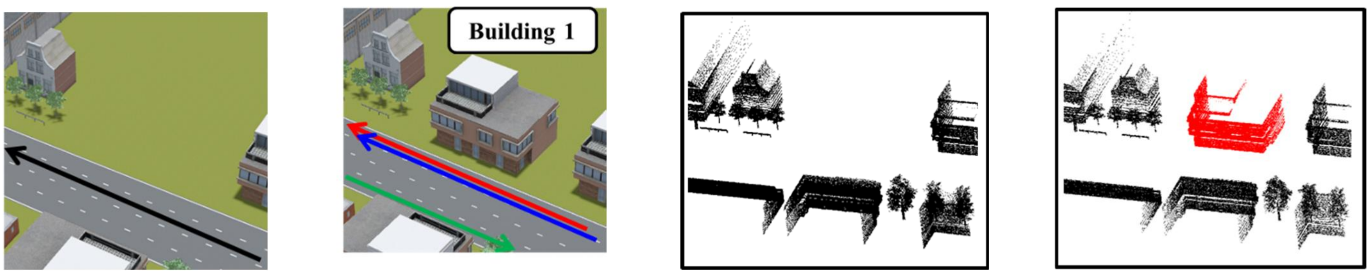
・Case 3: One-stage majority voting on the central server under the "multiple votes per LiDAR scan" principle.

(a) Environment



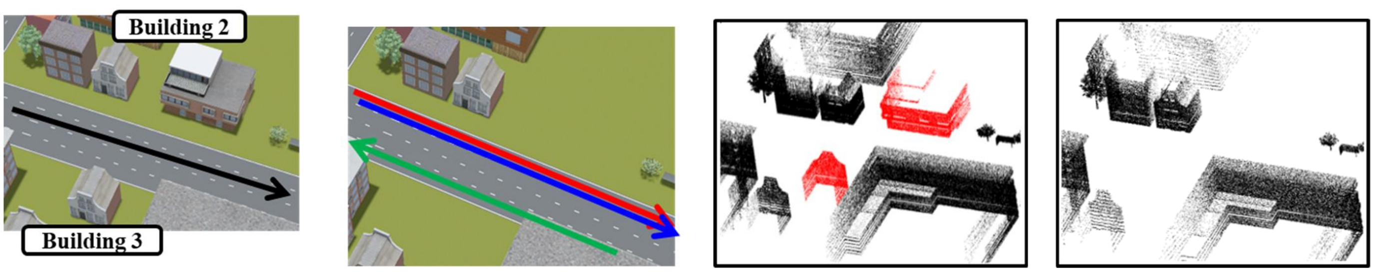
(b) Environmental map

Figure 6. Simulation environment and environmental map (top view).



(a) Previous environment (b) Current environment (c) Environmental map (d) Updated map

Figure 7. Simulation results of area 1 (bird's-eye view). In (a) and (b), the black, blue, and red lines indicate the movement paths of vehicles.



(a) Previous environment (b) Current environment (c) Environmental map (d) Updated map

Figure 8. Simulation results in area 2 (bird's-eye view). In (a) and (b), the black, blue, and red lines indicate the movement paths of vehicles.

・Case 4: One-stage majority voting on the central server under the "one vote per LiDAR scan" principle.

・Case 5 (proposed method): Two-stage majority voting, applying the "multiple votes per LiDAR scan" principle on each vehicle and the "one vote per vehicle" principle on the central server. In addition, vehicles incorporate invalid cells caused by the vertical FOV limitation of the LiDAR (see the last sentence of Section III-A).

Tables II and III summarize the precision and recall results, respectively, of all 514,900 cells in the environment shown in Figure 6. Table IV presents the corresponding F1-score (harmonic mean of precision and recall). Larger values of all three metrics indicate higher cell-state detection performance.

As shown by the F1-scores in Table IV, the proposed method (case 5) achieves the highest cell-state detection

TABLE II. PRECISION

	Unchanged	Changed (occupied)	Changed (empty)
Case 1	0.996	0.748	0.550
Case 2	0.996	0.787	0.536
Case 3	0.996	0.663	0.556
Case 4	0.995	0.768	0.499
Case 5	0.996	0.748	0.645

 ISBN: 978-1-68558-399-6

TABLE III. RECALL

	Unchanged	Changed (occupied)	Changed (empty)
Case 1	0.994	0.998	0.767
Case 2	0.994	0.954	0.767
Case 3	0.994	0.987	0.767
Case 4	0.993	0.882	0.767
Case 5	0.996	0.998	0.767

TABLE IV. F1-SCORE

Case 1	0.830
Case 2	0.829
Case 3	0.811
Case 4	0.807
Case 5	0.850

performance among all cases. In addition, the two-stage majority voting method (cases 1 and 2) outperforms the one-stage majority voting method (cases 3 and 4). Finally, majority voting based on the "multiple votes per LiDAR scan" principle (cases 1 and 3) yields better performance than majority voting based on the "one vote per LiDAR scan" principle (cases 2 and 4).

VI. CONCLUSIONS

This paper presented a crowd-sensing-based method for updating 3D point-cloud maps of road environments using LiDAR data collected from multiple vehicles. Each vehicle mapped its 3D LiDAR point-cloud data onto a 2D local grid map, and the state of each grid cell was classified using a majority voting method under the "multiple votes per LiDAR scan" principle. The 2D local grid map comprising the cell states was sent to a central server that aggregated the 2D local grid maps from all vehicles and determined the final cell states through majority voting under the "one vote per vehicle" principle. Simulation experiments demonstrated the effectiveness of the map-update method.

In this study, the 3D LiDAR point clouds were mapped onto a 2D grid map to mitigate the vertical sparsity inherent in mechanical LiDAR, simplify the map-update process, and reduce computational cost. However, because height information is discarded, the map-update accuracy may degrade for environmental features with substantial vertical variation. As future work, the impact of height-information loss on map-update accuracy will be further investigated. In addition, real-world experiments using a test vehicle will be conducted to validate the effectiveness and limitations of the proposed method.

ACKNOWLEDGMENT

This study was partially supported by the KAKENHI Grant #23K03781, the Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, "Autonomous Driving, Technical, Legal and Social Aspects," Springer Open, 2016.

[2] T. Hoffmann and G. Prause, "On the Regulatory Framework for Last-Mile Delivery Robots," Machines 2018, 6, 33, pp. 1–16, 2018.

[3] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," IEEE Access, vol. 8, pp. 58443–58469, 2020.

[4] M. Elhashash, H. Albanwan, and R. Qin, "A Review of Mobile Mapping Systems: From Sensors to Applications," Sensors 2022, 22, 4262, pp. 1–26, 2022.

[5] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual SLAM: What are the Current Trends and what to Expect?," Sensors 2022, 22, 9297, pp. 1–29, 2022.

[6] Y. Zhang, P. Shi, and J. Li, "3D LiDAR SLAM: A Survey," The Photogrammetric Record, vol. 39, pp. 457–517, 2024.

[7] P. Biber and W. Strasser, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2743–2748, 2003.

[8] P. J. Besl and N. D. McKay, "A Method of Registration of 3-D Shapes," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, pp. 239–256, 1992.

[9] S. Tanaka, C. Koshiro, M. Yamaji, M. Hashimoto, and K. Takahashi, "Point Cloud Mapping and Merging in GNSS-Denied and Dynamic Environments Using Onboard Scanning LiDAR," Int. J. Advances in Systems and Measurements, vol. 13, pp.275–288, 2020.

[10] B. Wijaya et al., "High Definition Map Mapping and Update: A General Overview and Future Directions," arXiv:2409.09726, pp. 1–30, 2024.

[11] I. Jinno, Y. Sasaki, and H. Mizoguchi, "3D Map Update in Human Environment Using Change Detection from LIDAR Equipped Mobile Robot," Proc. 2019 IEEE/SICE Int. Symp. System Integration, pp. 330–335, 2019.

[12] D. Fukutomi, K. Maeda, T. Azumi, S. Kato, and N. Nishio, "The Integration of Two Independent Point Cloud Maps Acquired at Different Times," Proc. Embedded System Symp., Information Proc. Society of Japan, pp. 54–61, 2019 (in Japanese).

[13] K. Kim, S. Cho, and W. Chung, "HD Map Update for Autonomous Driving with Crowdsourced Data," IEEE Robotics and Automation Letters, vol. 6, pp. 1895–1901, 2021.

[14] C. Kim et al., "Updating Point Cloud Layer of High Definition (HD) Map Based on Crowd-Sourcing of Multiple Vehicles Installed LiDAR," IEEE Access, vol. 9, pp. 8028–8046, 2021.

[15] C. Kim et al., "Cloud Update of Geodetic Normal Distribution Map Based on Crowd-Sourcing Detection Against Road Environment Changes," J. Advanced Transportation, vol. 2022, pp. 1–26, 2022.

[16] Simcenter Prescan, https://resources.sw.siemens.com/en-US/article-introducing-simcenter-prescan360/ [retrieved: Dec, 2025].