IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 November 2025, accepted 16 December 2025, date of publication 22 December 2025,
date of current version 29 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3646773

RESEARCH ARTICLE

Adaptive Topological Mapping With Free Area-Based Node Deletion for Autonomous Mobile Robots

HARUKA OZAKI[1], (Student Member, IEEE), YUICHIRO TODA[1], (Member, IEEE), NAOKI MASUYAMA[2], (Member, IEEE), KAI FUJI[1], AND TAKAYUKI MATSUNO[1], (Member, IEEE)

[1]Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan
[2]Graduate School of Informatics, Osaka Metropolitan University, Osaka 599-8531, Japan

Corresponding author: Yuichiro Toda (ytoda@okayama-u.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under Grant JP24K20870 and Grant JP25K03187, and in part by the New Energy and Industrial Technology Development Organization (NEDO) under Grant JPNP24022188.

ABSTRACT This paper proposes an adaptive topological map building method, called Adaptive Resonance Theory-based Topological Clustering with Different Topologies (ATC-DT), for autonomous mobile robots using 3D point cloud data. ATC-DT framework integrates a novel node deletion mechanism that detects layout changes through free area detection. This allows the robot to update topological maps dynamically, removing outdated nodes caused by environmental changes. Experiments in real environments validate the ability of the method to perform global path planning, free area estimation, and adaptive navigation. The approach significantly improves navigation performance by improving map relevance and reducing redundancy of paths.

INDEX TERMS Topological map building, navigation system, autonomous mobile robot.

I. INTRODUCTION

Autonomous mobile robots moving various environments, the ability to perceive the surrounding 3D space is essential for safe and efficient movement [1], [2], [3]. Recently, with the development of precise and affordable 3D distance measurement sensors such as Lidar and RGB-D cameras, the 3D distance measurement sensor has become easier to equip autonomous robots. This has led to a rapid advancement in research related to space perception technology using 3D point cloud in the robots. In particular, the technology for building an environmental map and localizing the self-position is fundamental in the robots, and many high-precision 3DSLAM methods using 3D point clouds have been proposed [4], [5], [6]. While 3DSLAM allows robots to build extensive environmental maps, for mobile robots to execute tasks, there is a need for methods that efficiently extract necessary information from the vast 3D point cloud within the environment maps. Especially for navigation tasks to reach a destination from the current location [7], [8], [9], [10], using a topological structure described with nodes and edges as a map representation to structure point cloud enables direct use in path planning. Consequently, developing efficient techniques for building topological structures from point cloud becomes advantageous.

As a method to efficiently learn topological structures from data, competitive learning methods, including self-organizing maps, have been proposed. In particular, Growing Neural Gas (GNG) [11], the learning method that includes the addition and deletion of nodes and edges in the learning algorithm, can adaptively learn topological structures from unknown 3D point cloud, where the data amount is not constant. Consequently, it has been applied to various 3D environment recognition technologies, including 3D reconstruction and tracking. In this context, the authors have also proposed GNG with Different Topologies (GNG-DT) [12] as a 3D space perception method for autonomous robots, demonstrating efficient information

The associate editor coordinating the review of this manuscript and approving it for publication was Tai Fei.

extraction by learning topological structures for multiple feature attributes, such as color and shape, from 3D point cloud. Moreover, by utilizing GNG-DT for a traversability perception of autonomous mobile robots, an adaptive local planning approach was achieved, allowing the planning of traversable paths based on a topological structure derived in real-time from measured point cloud [13], [14]. This method proved effective in actual environments. In addition, by employing GNG-DT on point cloud from the environmental maps, a strategy for global navigation in unknown environments was introduced, showing promising results in simulation environments. However, because GNG-DT learns with a fixed learning rate, challenges included the inability to continuously build a topological structure from the measured point cloud and the displacement of node positions with each learning epoch.

Therefore, the authors focused on Adaptive Resonance Theory (ART)-based Topological Clustering (ATC), a topological structure learning method based on ART, as an approach to avoid these issues [15], [16]. Unlike GNG-DT, ATC performs node addition and learning based on a distance threshold judgment called a vigilance parameter. Furthermore, the learning rate decays with the number of learning iterations. Therefore, ATC-based learning algorithm realize the continuous topological map building which was impossible with GNG-DT, and the authors proposed ATC-DT as a topological structure learning method for 3D point cloud [17]. Using ATC-DT, it has been shown in simulation environments that the topological structure of environmental maps can be continuously learned from the robot's self-position and the measurement results of the 3D point cloud. However, the effectiveness has not yet been validated in real environments. Although the introduction of ATC-DT has improved the convergence of node learning, there is a problem where node positions become fixed and cannot be adjusted for layout changes in the topological map when layout changes occur in the environment. Therefore, the aim of this paper is to realize a method that allows for the correction of topological maps in environments with layout changes. To achieve this, the authors propose a method to build a topological map that adapts to layout changes by detecting perceptual ranges called free area from point cloud measured by 3D distance sensors and by deleting critical nodes contained in the free area. The main contributions of this paper are as follows:

1) We introduce a novel node deletion mechanism that utilizes a free area detection, enabling the removal of outdated nodes caused by environmental layout changes.
2) To enable real-environment deployment of ATC-DT, we develop a navigation system that integrates ATC-DT with free area detection and LiDAR-Inertial Odometry for real-time topological map adaptation.
3) Validation experiments conducted in real environments confirm that ATC-DT enables the global topological map building and supports reliable path planning, demonstrating its practical applicability to autonomous robot navigation.

The remainder of this paper is organized as follows. Section II reviews related work on topological map construction and environmental adaptation methods. Section III describes the system configuration for real-environment experiments, including sensor integration and localization. Section IV details the proposed ATC-DT algorithm and the integration of the node deletion mechanism. Section V introduces the free area detection method used for adaptive node removal. Section VI presents experimental validations in real environments, demonstrating the feasibility and performance of the proposed approach. Finally, Section VII concludes the paper and discusses future research directions.

II. RELATED WORKS

In this paper, we propose a method for a map construction method that can be used by autonomous mobile robots for action planning. There are various methods for expressing environmental maps, and many methods have been proposed for constructing environmental maps for mobile robots. One of the requirements for autonomous mobile robots is real-time performance, which requires a reduction in the amount of data. To achieve this, it is necessary to abstract the surrounding environmental data. One method for this purpose is the generation of a mesh structure that abstracts input data to reduce the amount of data while constructing an environmental map. Reference [18] proposes a method to estimate a smooth mesh from depth data. This method allows for high-quality mesh generation in real-time but is sensitive to noise and outliers. Reference [19] generates 3D meshes by extracting features not only from depth data but also from images captured by cameras. Although it features less data volume compared to point clouds, it is affected by errors due to camera pose estimation. In [20], a method called the Triangular Mesh Map (TMM) was proposed, where an undirected graph is constructed from the generated nodes and edges, and the shortest path is planned using the Dijkstra algorithm. While the reproducibility of path planning is high if the parameters are fixed, appropriate parameter settings are required. As another method for generating triangular mesh maps, [21] proposed a stochastic triangular mesh (STM) represented as a 2.5D mesh. The map can be updated linearly; therefore, the proposed method can rapidly respond to a changing environment. However, the proposed method cannot represent multi-layer structures such as bridges or overhangs because the structure is composed of the 2.5D model. In [22], [23], a 3D mesh map is built in rough terrain environments, and based on the generated mesh, roughness and normals are estimated to determine traversability. However, there is a challenge that a new mesh needs to be generated in response to the change of viewpoint. In [24], in the literature, Kimera has strengths in metric semantic mapping, but the ‘‘Kimera-Mesher’’ module is used for fast, real-time 3D mesh generation to

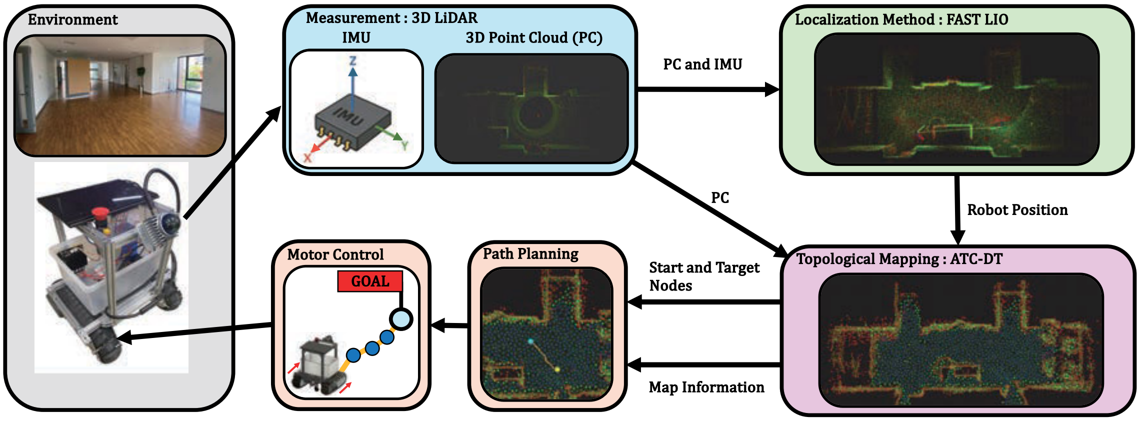


FIGURE 1. System configuration of the proposed method for real-environment navigation.

support obstacle avoidance. Although real-time meshing is possible, additional processing is required for accurate mesh generation. As described above, the mesh map can be abstracted while faithfully reproducing the shape of the environment, and the vertices, edges, etc. of the mesh can be used for path planning.

While environmental representation using mesh structure reduces the input data by dividing the environment into uniform meshes, it consumes a large amount of memory in a large area. However, this method consumes a large amount of memory in arge-scale [25], [26]. Since this research aims to construct environmental maps for large areas, an environmental map construction method that can reduce the amount of data is desirable. A topological map, which is connected by nodes and edges like a mesh map and can be used for path planning, is a more abstract way to represent the environment. Reference [27] proposed a method to build a topological map by extracting a path using edge detection filters from a distance map, followed by skeletonization. The proposed method has low computational cost as the proposed method extracts only important paths, but the proposed method has the drawback of being challenging to maintain the accuracy of the information about the entire environment. Reference [28] proposed a method called PRISM-TopoMap, which uses a learning-based place recognition model that fuses images from multiple cameras and point cloud, and is capable of building more accurate topological maps. However, since 2D features are used for point cloud matching, it may not completely capture the 3D environment. Reference [29] build a topological map using a method that extends the self-organizing map, called Place Cell-Growing When Required (PC-GWR) network, suitable for high-level abstraction for navigation. However, since the learning coefficient converges as learning progresses, which cannot cope with environmental changes. Also, in [13], [30], topological map building using one of the self-organizing maps, GNG, was proposed, allowing 3D space perception and path planning in unknown environments. However, since all measured point cloud is used as input data for large-scale environments, the data volume required for learning becomes enormous. Reference [31] proposed a Voronoi-based method for building topological maps, offering an efficient method for large-scale environments. However, there is a challenge in accurately representing complex environments. Therefore, in this study, a method based on ATC that can be applied even in complex environments and continue to construct a global topological map in real-time only from the measured point cloud is used.

III. SYSTEM CONFIGURATION

In [17], experiments were limited to simulations, but in this study, we realize autonomous navigation in a real environment using topological maps built by ATC-DT. To realize ATC-DT in a real environment, 3D point clouds and self-localization are necessary. Therefore, the robot is equipped with a 3D LiDAR (Mid-360), and the developed robot configuration is shown in Fig. 2 and Table 1. Although the specific system configuration is shown in Fig. 1, in this study, self-localization is executed by FASTLIO, one of the 3D SLAMs, from the point clouds and IMU data measured from the Mid-360. Utilizing self-localization results along with the point clouds recorded at that instance as input for ATC-DT enables topological mapping and navigation in real environments.

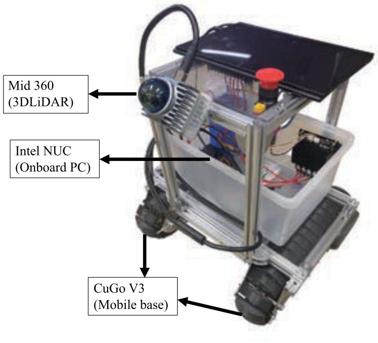


FIGURE 2. Crawler robot developed for autonomous navigation using ATC-DT.

TABLE 1. Specification of robot.

Item	Specification
Width	460 [mm]
Depth	450 [mm]
Height	600 [mm]
Motor output	rated 45 [W]
Sensor	Mid-360
Motor	CuGo V3

IV. TOPOLOGICAL MAP BUILDING USING ATC-DT

Regardless of robots, the stability derived from retaining learned information and the plasticity to learn new information have a trade-off relationship. Such a conflicting relationship is called the stability-plasticity dilemma [32], and it is necessary to effectively adapt to the environment. However, the related research GNG faces challenges due to this relationship, whereby while it can flexibly learn new knowledge from input data, it suffers from catastrophic forgetting, where the newly acquired knowledge causes previously learned information to be forgotten. As an approach to avoid catastrophic forgetting, there is ART [32]. ART is a theory based on biological plausibility and has been devised as an algorithm that self-organizes and learns quickly and stably for any input data. Since the purpose of this research is to create a stable global topological map, we apply ATC that can avoid catastrophic forgetting to the building of a topological map. Additionally, ATC-DT, developed from ATC, supports the construction of multiple topological structures, aiming for global path planning and autonomous movement using this method. The learning algorithm will be explained below.

A. LEARNING ALGORITHM

Firstly, we define the main variables used in ATC-DT. We define the set of attributes in this research as $S = \{$ position information (pos), normal vector (nor), traversability (tra) $\}$. Next, we define the input vector and reference vector respectively as $\mathbf{v} = \{\mathbf{v}^{pos}\}$, $\mathbf{h}_i = \{\mathbf{h}_i^{pos}, \mathbf{h}_i^{nor}, h_i^{tra}\}$. The distance between the input vector in an attribute o and the reference vector of the ith node is defined as d_i^o, as follows,

$$d_i^o = ||\mathbf{v}^o - \mathbf{h}_i^o||. \tag{1}$$

Furthermore, by defining the set of edges for an attribute o $(\in S)$ as $C^o = \{c_{1,2}^o, \cdots c_{i,j}^o, \cdots\}$, a topological structure is built for each of the multiple attributes. If the ith and jth nodes are connected, then $c_{i,j}^o = 1$, and if not connected, $c_{i,j}^o = 0$. Below, the detailed contents of the overall algorithm are explained.

Step 0. For initialization, set the number of learning epoch t to 0 and obtain the initial measurement data set V. Randomly generate the reference vectors $\mathbf{h}_1$ and $\mathbf{h}_2$ for the two nodes, and set the winner counts of the nodes to $M_1 = 1$ and $M_2 = 1$.

Step 1. Randomly obtain one input vector $\mathbf{v}$ from the measurement data set V.

Step 2. Select the first winner node s_1 and the second winner node s_2 corresponding to the input vector $\mathbf{v}$. The first and second winner nodes represent the nodes closest to and the second closest to the input vector, respectively. The set A represents the set of node numbers.

$$\begin{aligned} s_1 &= \arg\min_{i\in A} d_i^{pos}, \\ s_2 &= \arg\min_{i\in A\backslash s1} d_i^{pos}, \end{aligned} \tag{2}$$

Step 3. If the distance $d_{s_1}^{pos}$ of the first winner node is smaller than the vigilance parameter V_{thv} $(d_{s_1}^{pos} < V_{thv})$, execute Step 4. Otherwise, add a node according to the following procedure. M_i denotes the number of times the ith node has become the first winner node, and N represents the number of nodes. After adding a node, execute Step 7. The V_{thv} is set slightly larger than the robot's width. In this way, the robot can avoid colliding with walls and other obstacles.

$$\begin{aligned} \mathbf{h}_{N+1}^{pos} &= \mathbf{v}, \\ M_{N+1} &= 1, \\ N &\leftarrow N+1, \end{aligned} \tag{3}$$

Step 4. Update the reference vector of node s_1 using the following equation,

$$\mathbf{h}_{s_1}^{pos} \leftarrow \mathbf{h}_{s_1}^{pos} + \frac{1}{10M_{s_1}}(\mathbf{v} - \mathbf{h}_{s_1}^{pos}). \tag{4}$$

Furthermore, when the distance $d_{s_2}^{pos}$ of the second winning node is smaller than the vigilance parameter V_{thv} $(d_{s_2}^{pos} < V_{thv})$, the reference vector associated with the first winning node is updated using the following equation,

$$\mathbf{h}_k^{pos} \leftarrow \mathbf{h}_k^{pos} + \frac{1}{100M_k}(\mathbf{v} - \mathbf{h}_k^{pos}). \tag{5}$$

Step 5. Increment the age of the edge that has the connection relationship of the position information with the first winner node $(c_{s_1,i}^{pos} = 1)$.

$$g_{s_1,i} \leftarrow g_{s_1,i} + 1. \tag{6}$$

Furthermore, if the distance $d_{s_2}^{pos}$ to the second winning node is smaller than the vigilance parameter V_{thv} $(d_{s_2}^{pos} < V_{thv})$, the age of the edge is reset to 0 $(g_{s1,s2} = 0)$, and if there is no position information edge between nodes s_1 and s_2, a new position information edge is added $(c_{s1,s2}^{pos} = 1)$. The addition of edges regarding attributes other than position information o $(\in S^{pos})$ is performed according to the following formula.

$$c_{s_1,s_2}^o = \begin{cases} 1 & if\ f^o(\mathbf{h}_{s_1}^o, \mathbf{h}_{s_2}^o) = true, \\ 0 & otherwise, \end{cases} \tag{7}$$

where, $f^o(\cdot)$ is a decision function based on each attribute, returning true when the similarity between attributes is high and false otherwise.

Step 6. Update the age threshold of the edge g_{max} and remove all edges of attribute o $(\in S)$ that exceed the threshold $(c_{s1,s2}^o = 0)$.

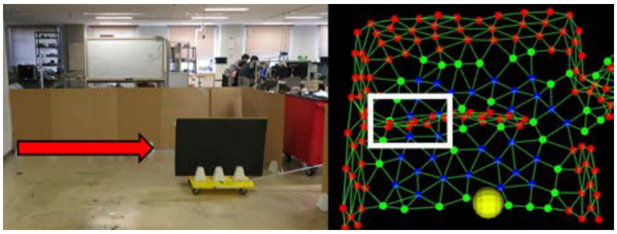


FIGURE 3. Example of remaining nodes: Nodes within the white square could not adapt to environmental changes and were not removed. Blue: traversable nodes; Green: contour and traversable nodes; Red: untraversable nodes; Big yellow: Robot(Start) node.

Step 7. Increment the number of learning epochs t, and update the measurement data set V every λ iterations. Then, return to Step 1.

1) NODE DELETION

In building topological map using ATC-DT, while nodes can preserve geometric information for convergence, there is an issue of unnecessary nodes remaining because nodes positioned on trajectories of dynamic objects also converge (Fig. 3). Therefore, a node deletion function is added to the conventional ATC-DT learning algorithm. Specifically, as shown in the following equation, for a given the ith node, distances between all input data are compared, and if the distance to the closest input data is larger than the threshold thv^{dis}, the ith node is deleted.

$$\min_{j \in A} ||\mathbf{h}_i^{pos} - \mathbf{v}_j|| > thv^{dis}. \tag{8}$$

By adding the node deletion function, it is possible to avoid a situation where redundant nodes remain, as nodes are deleted in places without point cloud due to layout changes.

V. FREE AREA DETECTION FOR NODE DELETION

When performing the node deletion based on equation (8), if the deletion area is not properly set, there is a risk of deleting undesirable nodes. These undesirable nodes correspond to regions that are occluded by dynamic objects. As illustrated in Fig. 4, nodes located in front of the robot within these occluded regions are deleted by the node deletion process. Because the LiDAR sensor cannot capture point clouds in such occluded areas, the corresponding nodes are removed by the deletion algorithm described in Section IV-A1. Therefore, this paper applies free area [33], [34] as the node deletion area. Free area refers to the area where the robot can move freely within the perceived range (perceived by point cloud) as shown in Fig. 5. The details of the free area extraction method are explained.

Step 1. Equally divide the area in front of the robot into N sections.

Step 2. For each area, determine the closest point cloud (proximity point) from the current position of the robot $\mathbf{p}$ using the following equation.

$$\mathbf{r}_i = \min_{j \in V} ||\mathbf{p} - \mathbf{v}_j||. \tag{9}$$

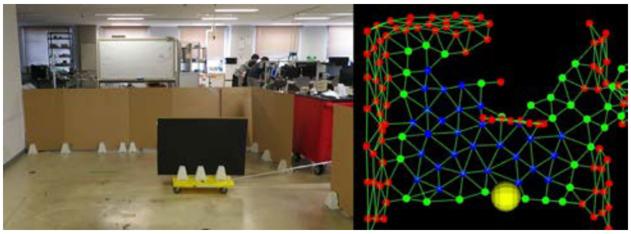


FIGURE 4. Example of node deletion in regions occluded by a dynamic object.

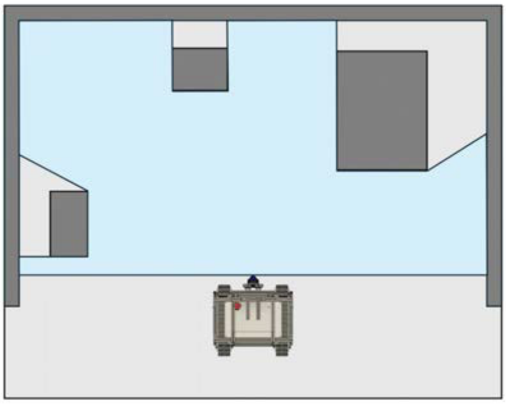


FIGURE 5. Visualization of the free area (shaded in light blue), which represents regions where the robot can move without obstruction.

where, $\mathbf{r}_i$ is the proximity point in the $i = \{0, 1 \ldots N\}$-th area, V is the set of point cloud, and $\mathbf{v}_j$ represents the j-th point cloud.

Step 3. Calculate the unit vector $\mathbf{q}_i$ directed from robot $\mathbf{p}$ to the nearby point $\mathbf{r}_i$.

$$\mathbf{q}_i = \frac{\mathbf{r}_i - \mathbf{p}}{||\mathbf{r}_i - \mathbf{p}||}. \tag{10}$$

Step 4. Calculate the inner product between the vector from a node $\mathbf{h}_k$ to a proximity point $\mathbf{r}_i - \mathbf{h}_k$ and $\mathbf{q}_i$. The condition for the node $\mathbf{h}_k$ to exist on the robot side with respect to the tangent at the proximity point $\mathbf{r}_i$ is given by the following expression. Figure 6 shows how the free area is determined for a certain proximity point. Node $\mathbf{h}_1$ is on the robot side because its inner product is $\boldsymbol{\eta}_i^T(\mathbf{r}_i - \mathbf{h}_k) > 0$ and node $\mathbf{h}_2$ is not on the robot side because its inner product is $\boldsymbol{\eta}_i^T(\mathbf{r}_i - \mathbf{h}_k) < 0$.

$$\boldsymbol{\eta}_i^T(\mathbf{r}_i - \mathbf{h}_k) > 0. \tag{11}$$

Step 5. Perform Step 3 and Step 4 at all adjacent points, and determine the node $\mathbf{h}_k$ that satisfies equation (11) at all adjacent points as a node existing within the free area.

When dividing the front of the robot into six areas in Step 1, the extracted free area is as shown in Fig. 7. By setting the free area, it is possible to extract nodes located within the range where point cloud exists. Indeed, the appearance of the free area extracted in the actual environment is shown in Fig. 8. The purple nodes represent the nodes existing within the

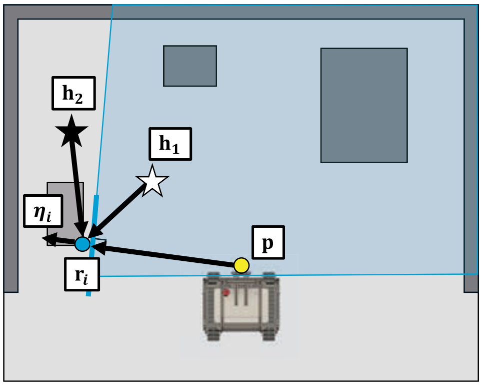


FIGURE 6. Example of free area determination at a specific proximity point using geometric criteria.

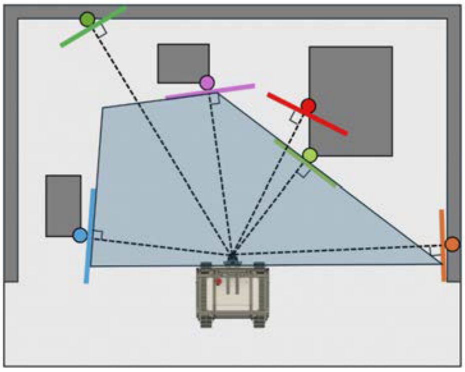


FIGURE 7. Free area extracted by dividing the front region into six sections. Colorful dots represent proximity points in each section.

free area. By performing node deletion only on these nodes, the deletion of undesirable nodes can be avoided, allowing for appropriate node deletion. The entire flowchart of ATC-DT, including free area detection and node deletion, is shown in Fig. 9.

VI. EXPERIMENTAL RESULTS

The purpose of this experiment is to verify the feasibility of global path planning using ATC-DT in real environments, as well as to confirm the detection of free area in dynamic states, the adaptation to changes in the environmental layout, and the ability for path planning and autonomous movement to the starting point. The robot used in the experiment is shown in Fig. 2, and its specifications are listed in Table 1.

A. OBSTACLE REMOVAL VERIFICATION

1) EXPERIMENTAL SETTING

In this section, we verify the proposed method when obstacles are completely removed. The experimental environment is shown in Fig. 10. As for the experimental procedure, initially, the robot autonomously moves to the backside of the obstacle. Next, the obstacle is removed and the robot starts autonomous movement towards the starting point as the destination.

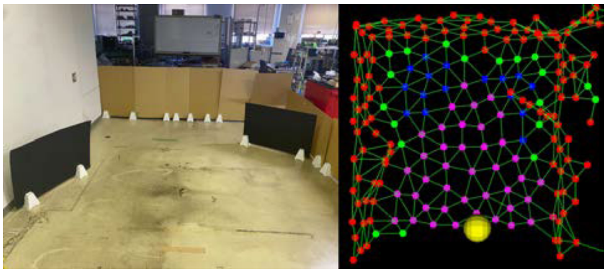


FIGURE 8. Free area detection results in a real environment. Purple dots indicate nodes identified within the free area.

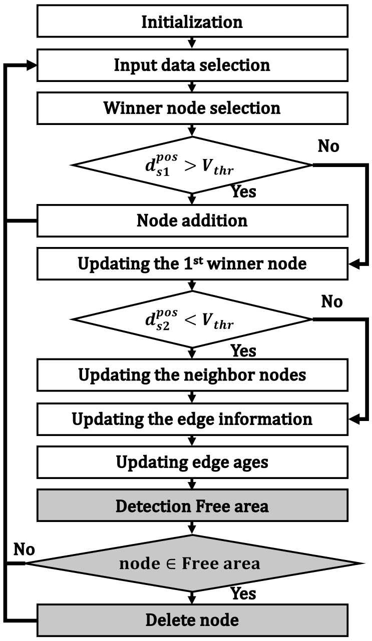


FIGURE 9. Flowchart of the proposed ATC-DT-based topological map building with free area detection and node deletion.

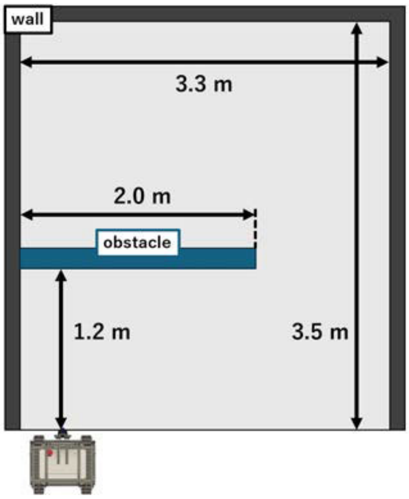


FIGURE 10. Layout of the experimental environment used for obstacle removal verification.

2) RESULTS

The experimental results are shown in Fig. 11. This figure shows that the robot first senses obstacles in front of it and then moves autonomously to the rear of the obstacles. It also shows that when returning to the starting point, the robot deletes red nodes representing removed obstacles by deleting nodes within the free area and returns via the shortest route. The ATC-DT had node convergence, indicating that a global topological map could be built while preserving geometric information. Figure 11 shows the movement to a target position set in an unknown environment. By combining ATC-DT with the self-localization results by FASTLIO, it was possible to build topological mapping in the real environment and reach the target position. In addition, free area could be detected even in dynamic states, and nodes of removed obstacles were deleted without issues, verifying that nodes could be deleted for completely removed obstacles. Table 2 summarizes the average path length, variance, and path reduction rate. This table also demonstrates that node deletion within the free area contributes to path length reduction.

As shown in Fig. 12, there was a situation where not only the node of the remaining obstacle was deleted, but also the surrounding nodes were deleted when removing nodes. This is considered to occur due to node deletion in the robot's blind spot range, as shown in Fig. 13. When detecting the free area, nodes located beyond the x-coordinate of the input data closest to the robot are targeted. In this case, if the sensor detection range was unevenly detected as in the right figure of Fig. 13, nodes within the blind spot range in the free area became a target for deletion, resulting in a situation where even nondesignated nodes are deleted.

To verify the effectiveness of node deletion in a free area, trajectories of robots were compared between the proposed method with deleted nodes and the conventional method without deletions in ATC-DT. Each experiment was conducted 10 times, and the resulting trajectories are shown in Fig. 14. It was observed that by executing node deletion, obstructing nodes were removed, aiming for the shorter path to the goal. Thus, the autonomous movement with node deletion allows for efficient path planning and reaching the destination.

TABLE 2. Comparison of path lengths in the obstacle removal verification.

Condition/Path length	Average	Variance
Without node deletion method	12.4880	0.7199
With node deletion method	9.8536	0.1705
Reduction Rate (%)		**21.10**

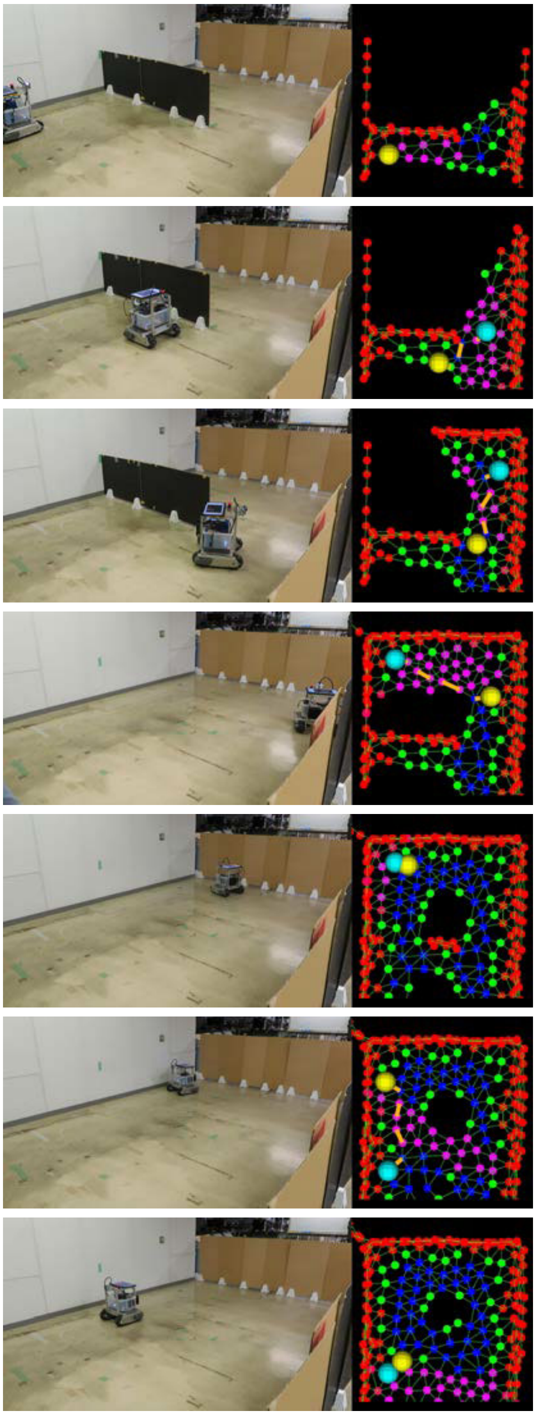


FIGURE 11. Robot trajectories (left) and topological map (right) built by ATC-DT. Blue: traversable nodes; Green: contour and traversable nodes; Red: untraversable nodes; Purple: nodes within free areas; Big yellow: Robot(Start) node; Big light blue; Goal node.

B. LAYOUT CHANGE VERIFICATION

In this section, we verify the behavior when the position of the obstacle is moved without removing it. The experimental environment is shown in Fig. 15. The experimental procedure involves autonomously moving the robot to the back side of the obstacle, similar to the previous experiment. Subsequently, the obstacle is moved diagonally upward to the right, and autonomous movement is initiated with the original start location as the destination.

C. RESULTS

The results of the experiment are shown in Fig. 16. This figure shows that the robot first moves autonomously to

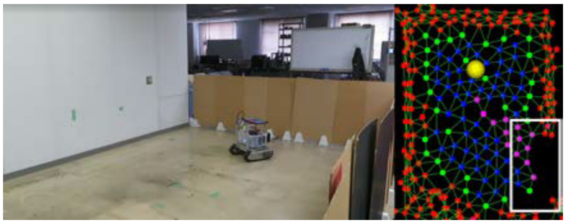


FIGURE 12. Example of excessive node deletion, including unintended surrounding nodes.

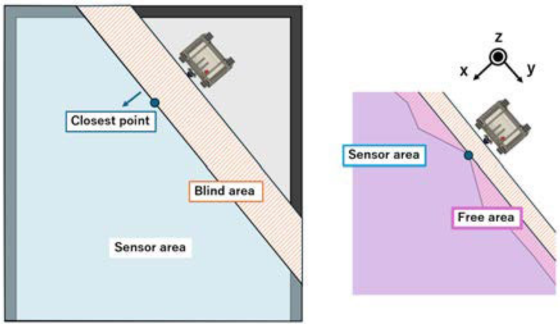
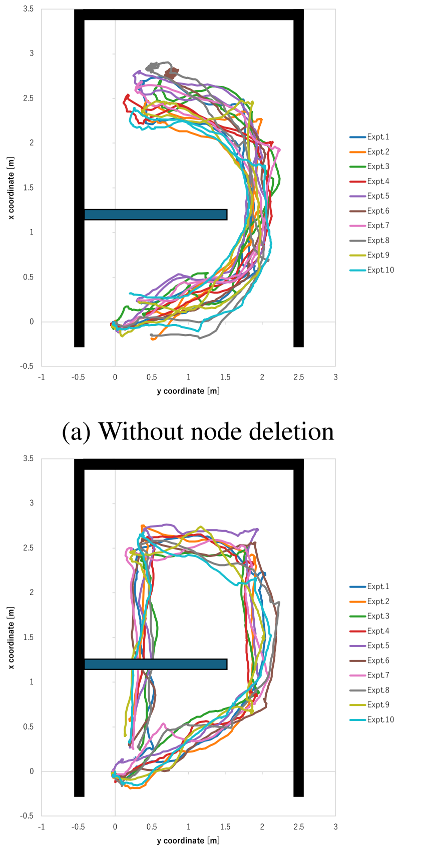


FIGURE 13. Illustration of the cause of over-deletion shown in Fig. 12. Nodes within sensor blind spots are incorrectly included.

the wall while perceiving the obstacle in front of it. Then, when returning to the starting point, the robot perceives a new obstacle that has moved backward and performs path planning recognizing that there are two obstacles until the pre-movement obstacle node enters the free area. Finally, when the remaining obstacle node before the move enters the free area, it is deleted, and the shortest path to the starting point is recalculated. Table 3 summarizes the average path length, variance, and path reduction rate. This table also demonstrates that node deletion within the free area contributes to path length reduction. Also, in this experiment, the trajectories of the robot were compared. Each experiment was conducted 10 times, and the graph showing the trajectories is presented in Fig. 17. By executing the deletion of nodes, it can be seen that the nodes of the remaining obstacles are deleted, aiming for a shorter path distance to the goal.

As shown in Fig. 18, when the obstacle was moved slightly to the upper right, there were instances where the nodes related to the obstacle before moving could not be deleted, and a path to the destination could not be planned. This is because, as shown in Fig. 19, when the obstacle was moved directly sideways or to the rear side relative to the robot's direction of movement, the nodes related to the obstacle before moving exist within the free area and could be deleted. However, when moved to the front side in relation to the robot's direction of movement, the remaining obstacle nodes do not enter the free area, preventing node deletion. To verify this, we moved an obstacle toward the robot and built a topological map using the proposed method. The results are shown in Fig. 20. As discussed, the node deletion method was not performed for the occluded region behind the obstacle, resulting in non-existent nodes remaining. This is a limitation of the proposed method and represents a challenge that needs to be addressed in future work.



(a) Without node deletion

(b) With node deletion (Proposed method)

FIGURE 14. Comparison of robot trajectories in the obstacle removal verification experiment.

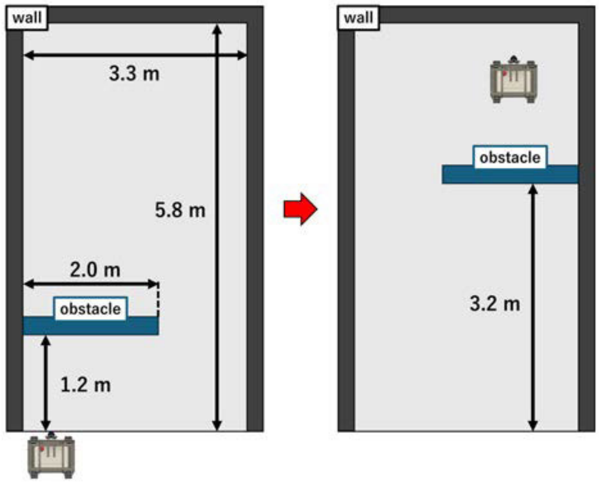


FIGURE 15. Experimental environment used to evaluate behavior under obstacle layout changes.

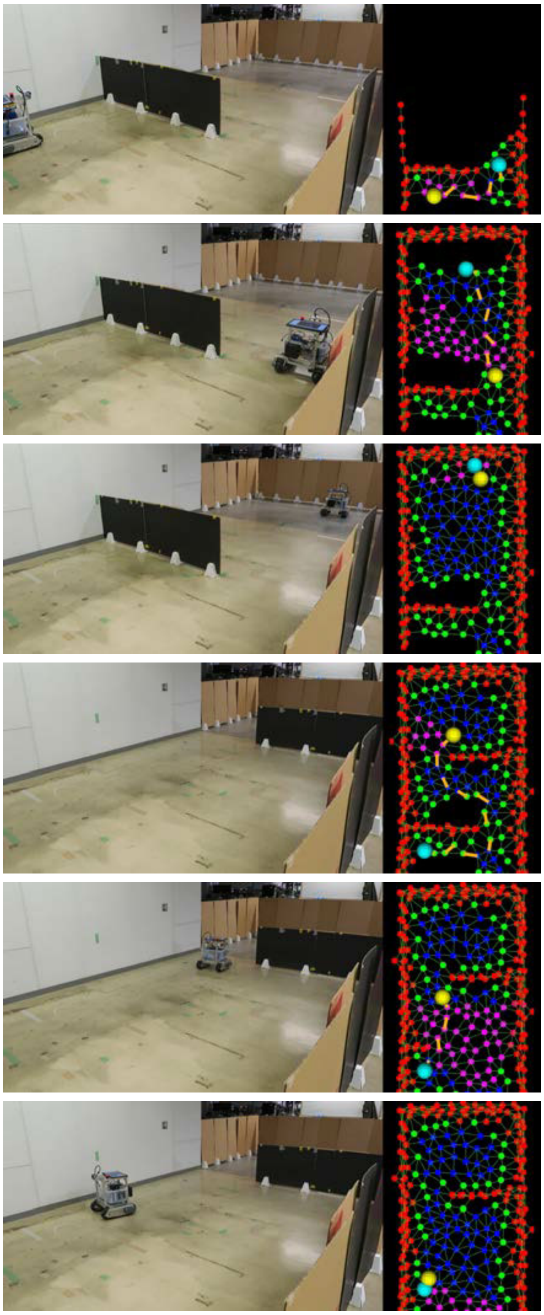
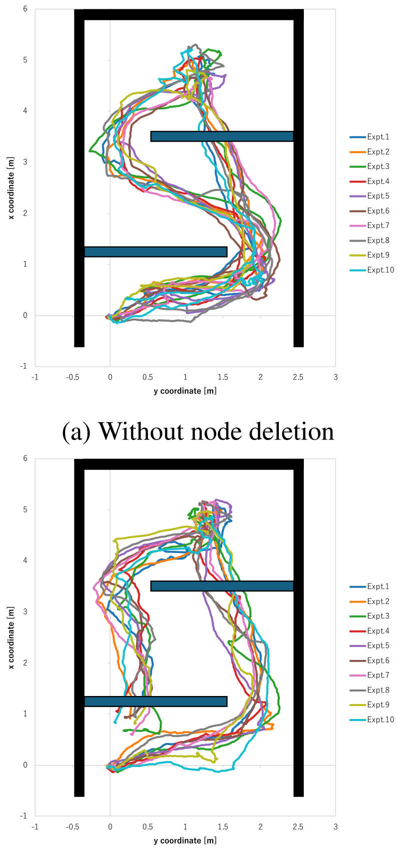


FIGURE 16. Topological mapping result after moving the obstacle diagonally. Color coding as in Fig. 11.

TABLE 3. Comparison of path lengths in the layout change verification.

Condition/Path Length	Average	Variance
Without node deletion	16.8163	1.1334
With node delete	14.3462	0.4755
Reduction Rate (%)		**14.69**



(a) Without node deletion

(b) With node deletion (Proposed method)

FIGURE 17. Comparison of robot trajectories in the layout change verification experiment.

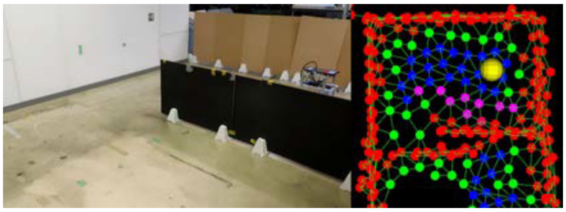


FIGURE 18. Failure case of node deletion due to insufficient inclusion of old obstacle nodes in the free area.

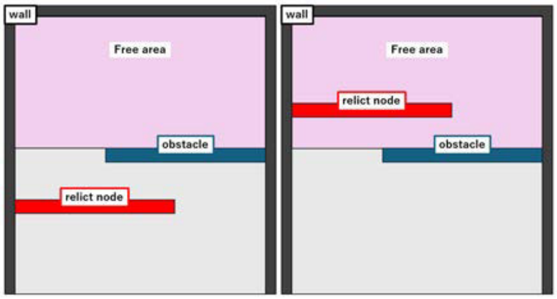


FIGURE 19. Explanation of node deletion failure in Fig. 18. Nodes in front of the robot's motion direction may remain outside the free area.

Figure 21 shows an example of processing time measurement results in this experiment. The ATC-DT used in the proposed method exhibits increased processing time as the number of nodes increases. Therefore, applying this method in large-scale outdoor environments requires improvements to reduce processing time. This is also one limitation of the

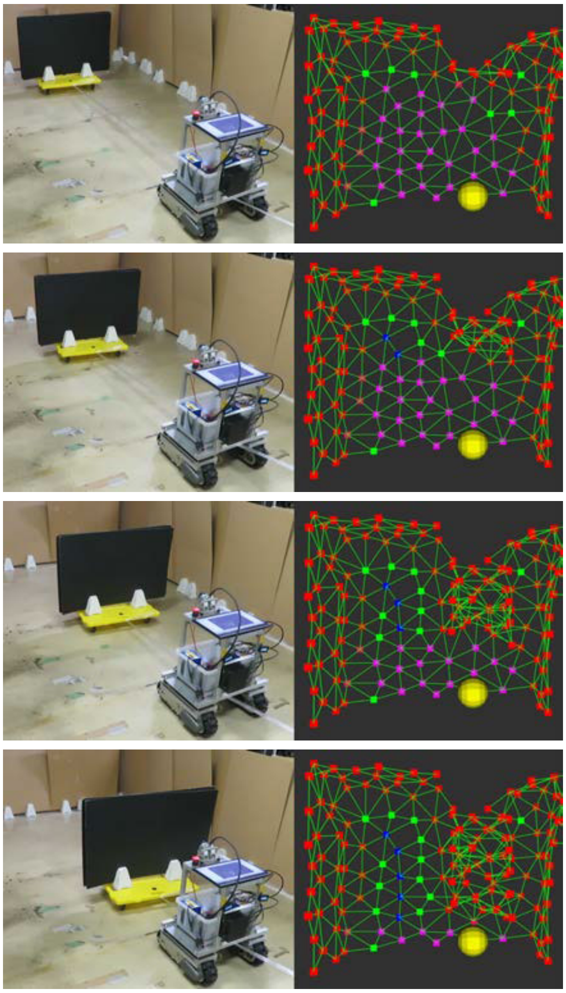


FIGURE 20. An example of node deletion failure when an obstacle is moved toward the robot.

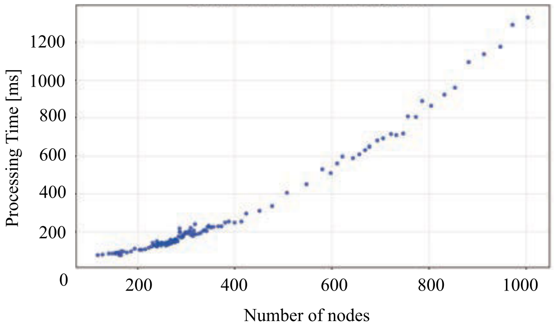


FIGURE 21. Relationship Between Node Number and Processing Time in ATC.

proposed method and represents one of the challenges that need to be addressed in the future.

VII. CONCLUSION

In this paper, we extended the ATC-DT-based topological map building method by incorporating a node deletion mechanism based on free area detection from 3D point cloud measurements. This approach enables autonomous mobile robots to continuously adapt global topological maps to changes in environmental layout. Real-environment experiments confirmed the feasibility of global path planning, free area detection, and adaptive navigation using the proposed method. Comparative evaluations demonstrated that the node deletion mechanism significantly improves path efficiency by removing obstructive nodes.

As future work, the current ATC-DT algorithm exhibits increasing processing time as the number of nodes grows in large-scale environments. Therefore, to achieve real-time operation in large-scale environments with more complex scenarios, we will improve the computational efficiency of the ATC-DT algorithm and free area detection. In addition, we aim to conduct verification experiments in large-scale outdoor environments and with various 3D distance measurement sensors to further evaluate the scalability and robustness of the proposed method under diverse and unstructured conditions.

REFERENCES

[1] H. Teng, Y. Wang, D. Chatziparaschis, and K. Karydis, "Adaptive LiDAR odometry and mapping for autonomous agricultural mobile robots in unmanned farms," *Comput. Electron. Agricult.*, vol. 232, May 2025, Art. no. 110023.

[2] G. Cao, B. Zhang, Y. Li, Z. Wang, Z. Diao, Q. Zhu, and Z. Liang, "Environmental mapping and path planning for robots in orchard based on traversability analysis, improved LeGO-LOAM and RRT algorithms," *Comput. Electron. Agric.*, vol. 230, Jan. 2025, Art. no. 109889.

[3] Y. Jin, X. Xia, Q. Gao, Y. Yue, E. G. Lim, P. Wong, W. Ding, and X. Zhu, "Deep learning in produce perception of harvesting robots: A comprehensive review," *Appl. Soft Comput.*, vol. 174, Apr. 2025, Art. no. 112971.

[4] D. Selvaratnam and D. Bazazian, "3D reconstruction in robotics: A comprehensive review," *Comput. Graph.*, vol. 130, Aug. 2025, Art. no. 104256.

[5] H. Mao and J. Luo, "Approach to 3D SLAM for mobile robots based on point-line features and superpixel segmentation in dynamic environments," *Measurement*, vol. 247, Apr. 2025, Art. no. 116742.

[6] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[7] A. Meysami, J.-C. Cuillière, V. François, and S. Kelouwani, "Investigating the impact of triangle and quadrangle mesh representations on AGV path planning for various indoor environments: With or without inflation," *Robotics*, vol. 11, no. 2, p. 50, Apr. 2022.

[8] D. Connell and H. M. La, "Dynamic path planning and replanning for mobile robots using RRT," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1429–1434.

[9] C.-B. Moon and W. Chung, "Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1080–1090, Feb. 2015.

[10] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT: An improved path planning algorithm for mobile robots," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113425.

[11] B. Fritzke, "A growing neural gas network learns topologies," in *Proc. Adv. Neural Inf. Process. Syst.*, 1995, pp. 625–632.

[12] Y. Toda, A. Wada, H. Miyase, K. Ozasa, T. Matsuno, and M. Minami, "Growing neural gas with different topologies for 3D space perception," *Appl. Sci.*, vol. 12, no. 3, p. 1705, Feb. 2022.

[13] Y. Toda, K. Ozasa, and T. Matsuno, "Growing neural gas based navigation system in unknown terrain environment for an autonomous mobile robot," *Artif. Life Robot.*, vol. 28, no. 1, pp. 76–88, Feb. 2023.

[14] K. Ozasa, Y. Toda, Y. Nakamura, T. Masuda, H. Konishi, and T. Matsuno, "Passability-based local planner using growing neural gas for an autonomous mobile robot," *IEEE Access*, vol. 12, pp. 171824–171835, 2024.

[15] N. Masuyama, T. Takebayashi, Y. Nojima, C. Kiong Loo, H. Ishibuchi, and S. Wermter, "A parameter-free adaptive resonance theory-based topological clustering algorithm capable of continual learning," 2023, *arXiv:2305.01507*.

[16] N. Masuyama, N. Amako, Y. Yamada, Y. Nojima, and H. Ishibuchi, "Adaptive resonance theory-based topological clustering with a divisive hierarchical structure capable of continual learning," *IEEE Access*, vol. 10, pp. 68042–68056, 2022.

[17] Y. Toda and N. Masuyama, "Adaptive resonance theory-based global topological map building for an autonomous mobile robot," *IEEE Access*, vol. 12, pp. 111371–111385, 2024.

[18] A. Rosinol and L. Carlone, "Smooth mesh estimation from depth data using non-smooth convex optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 6633–6640.

[19] J. Lee and Y. Cho, "Mesh-based photorealistic and real-time 3D mapping for robust visual perception of autonomous underwater vehicle," in *Proc. IEEE ICRA Workshop Field Robot.*, May 2024, p. 7.

[20] Y. Liu and Y. Jiang, "Robotic path planning based on a triangular mesh map," *Int. J. Control, Autom. Syst.*, vol. 18, no. 10, pp. 2658–2666, Oct. 2020.

[21] C. D. Lombard and C. E. van Daalen, "Stochastic triangular mesh mapping: A terrain mapping technique for autonomous mobile robots," *Robot. Auto. Syst.*, vol. 127, May 2020, Art. no. 103449.

[22] S. Pütz, T. Wiemann, J. Sprickerhof, and J. Hertzberg, "3D navigation mesh generation for path planning in uneven terrain," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 212–217, 2016.

[23] F. Ruetz, E. Hernández, M. Pfeiffer, H. Oleynikova, M. Cox, T. Lowe, and P. Borges, "OVPC mesh: 3D free-space representation for local ground vehicle navigation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8648–8654.

[24] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1689–1696.

[25] S. McCammon and G. A. Hollinger, "Topological path planning for autonomous information gathering," *Auto. Robots*, vol. 45, no. 6, pp. 821–842, Sep. 2021.

[26] C. Wang, W. Chi, Y. Sun, and M. Q.-H. Meng, "Autonomous robotic exploration by incremental road map construction," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1720–1731, Oct. 2019.

[27] Y. Yuan and S. Schwertfeger, "Incrementally building topology graphs via distance maps," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2019, pp. 468–474.

[28] K. Muravyev, A. Melekhin, D. Yudin, and K. Yakovlev, "PRISM-TopoMap: Online topological mapping with place recognition and scan matching," 2024, *arXiv:2404.01674*.

[29] X. Zhou, C. Weber, and S. Wermter, "A self-organizing method for robot navigation based on learned place and head-direction cells," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[30] M. Saroya, G. Best, and G. A. Hollinger, "Roadmap learning for probabilistic occupancy maps with topology-informed growing neural gas," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4805–4812, Jul. 2021.

[31] Y. Qi, R. Wang, B. He, F. Lu, and Y. Xu, "Compact and efficient topological mapping for large-scale environment with pruned Voronoi diagram," *Drones*, vol. 6, no. 7, p. 183, Jul. 2022.

[32] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis., Graph., Image Process.*, vol. 37, nos. 2–3, pp. 54–115, Nov. 1987.

[33] Y. Tazaki and Y. Yokokohji, "Outdoor autonomous navigation utilizing proximity points of 3D pointcloud," *J. Robot. Mechatronics*, vol. 32, no. 6, pp. 1183–1192, Dec. 2020.

[34] R. Nonomura, Y. Tazaki, H. Nagano, and Y. Yokokohji, "Free area extraction and 2D graph map construction based on a geometric property of proximity points," in *Proc. JSME Conf. Robot. Mechatron.*, 2024, pp. 1–5.

HARUKA OZAKI (Student Member, IEEE) received the B.E. degree from the Department of Mechanical and Systems Engineering, School of Engineering, Okayama University, Okayama, Japan, in 2023. He is currently pursuing the M.S. degree in mechanical systems and urban innovation science with Okayama University. His research interests include soft computing and intelligent robot.

YUICHIRO TODA (Member, IEEE) received the M.E. and Ph.D. degrees from Tokyo Metropolitan University, Tokyo, Japan, in 2013 and 2017, respectively. He is currently an Associate Professor with the Faculty of Environmental, Life, Natural Science and Technology, Okayama University, Okayama, Japan. His research interests include computational intelligence and autonomous mobile robots.

NAOKI MASUYAMA (Member, IEEE) received the B.Eng. degree from Nihon University, Funabashi, Japan, in 2010, the M.E. degree from Tokyo Metropolitan University, Hino, Japan, in 2012, and the Ph.D. degree from the Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia, in 2016.

He is currently an Associate Professor with the Department of Core Informatics, Graduate School of Informatics, Osaka Metropolitan University, Sakai, Japan. His current research interests include clustering, data mining, and continual learning.

KAI FUJI is currently pursuing the B.E. degree with the Department of Mechanical and Systems Engineering, Okayama University. His research interests include soft computing and intelligent robot.

TAKAYUKI MATSUNO (Member, IEEE) is currently pursuing the Ph.D. degree in engineering. In 2004, he was a Research Assistant with the Department of Engineering, Nagoya University. In 2006, he was an Assistant Professor with the Faculty of Engineering, Toyama Prefectural University. In 2011, he was a Lecturer with Okayama University. In 2017, he was an Associate Professor. He was a Professor with Okayama University, in 2022. He engages in research on the intelligent control of industrial manipulators and the medical robots for surgical assistance.