IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 May 2024, accepted 8 August 2024, date of publication 12 August 2024, date of current version 20 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3442304

RESEARCH ARTICLE

Adaptive Resonance Theory-Based Global Topological Map Building for an Autonomous Mobile Robot

YUICHIRO TODA[1], (Member, IEEE), AND NAOKI MASUYAMA[2], (Member, IEEE)
[1]Faculty of Environmental, Life, Natural Science and Technology, Okayama University, Okayama-shi, Okayama 700-8530, Japan
[2]Graduate School of Informatics, Osaka Metropolitan University, Sakai-shi, Osaka 599-8531, Japan

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

This work was supported in part by the JSPS KAKENHI under Grant JP24K20870 and Grant JP22K12199.

ABSTRACT 3D space perception is one of the key technologies for autonomous mobile robots that perform tasks in unknown environments. Among these, building global topological maps for autonomous mobile robots is a challenging task. In this study, we propose a method for learning topological structures from unknown data distributions based on competitive learning, a type of unsupervised learning. For this purpose, adaptive resonance theory-based Topological Clustering (ATC), which can avoid catastrophic forgetting of previously measured point clouds, is applied as a learning method. Furthermore, by extending ATC with Different Topologies (ATC-DT) with multiple topological structures for extracting the traversable information of terrain environments, a path planning method is realized that can reach target points set in an unknown environment. Path planning experiments in unknown environments show that, compared to other methods, ATC-DT can build a global topology map with high accuracy and stability using only measured 3D point cloud and robot position information.

INDEX TERMS Adaptive resonance theory, autonomous mobile robot, topological map.

I. INTRODUCTION

For robots autonomously moving in unknown environments, the ability to perceive and recognize the surrounding 3D space is important for efficiently performing tasks. In recent years, with the development of 3D distance measurement sensors such as LiDAR and depth cameras, robots have been able to easily measure high-precision 3D point cloud, and research on space recognition using 3D point clouds has been rapidly advancing [1], [2], [3], [4], [5]. In particular, the development of research on 3D SLAM, which simultaneously builds a 3D environmental map and localizes the self-position, which is essential information for autonomous mobile robots, is remarkable, and methods of 3D SLAM that operate in various environments have been realized [6], [7], [8]. Against this background, it is becoming increasingly important to extract necessary information from the built global 3D environmental map and utilize it for the task planning. In particular, an important task plan for autonomous mobile robots is path planning to move from the current point to the target point. In order to plan a path in an unknown environment, it is necessary to extract information such as the objects existing in the environment and the shape of the terrain surface. Also, for the path planning, it is necessary to extract a graph structure (global topological map) described by nodes and edges from 3D point cloud of the environmental map.

In the research of information extraction from 3D point clouds, various methods, including 3D object recognition using deep learning, have been proposed in recent years, enabling high-precision recognition of objects and shapes [9], [10], [11], [12]. However, autonomous mobile robots in unknown environments are expected to constantly encounter unknown objects, and even with methodologies based on supervised learning, there are still problems of false detection for unknown objects and shapes, even when using large-scale datasets. In addition, recognition technology using deep learning is still weak against environmental changes,

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

and when operating autonomous mobile robots in various environments, it is necessary to construct a dataset according to the environment. Furthermore, in order to utilize information for path planning, it is necessary to have a topological structure as mentioned above, so a framework for information extraction including environmental recognition using deep learning and topological map building. In this context, a methodology based on competitive learning, which is one of unsupervised learning and learns the topological structure from unknown data distributions, has been proposed as a research to realize information extraction from 3D point cloud. In particular, the Growing Neural Gas (GNG) [13] proposed by Fritzke is often used among the competitive learning methods for unknown 3D point cloud because GNG can dynamically add and delete nodes and edges contained in the topological structure and learn the 3D space while preserving the geometric features of the 3D point cloud.

In this context, we have proposed GNG with Different Topologises (GNG-DT), a real-time 3D space perception method to extract information from 3D point cloud by learning different topological structures from various attributes such as color and shape [14], [15]. Furthermore, utilizing the topological structures with different attributes for the path planning, our proposed method has realized the path planning in the unknown environment [16]. However, this method has a problem in that all the 3D point cloud contained in the environmental map that the robot has built so far by 3DSLAM is used as the input vector for GNG-DT. As the environmental map expands, the memory usage required for the input vector becomes enormous. In addition, when only the measured 3D point cloud is used as input, the learning rate is fixed in the conventional GNG learning algorithm. Therefore, as the robot gradually moves, the nodes are pulled in the direction of the robot's movement, which is a problem that the geometric structure of the map information cannot be properly preserved.

Generally, the ability to preserve knowledge learned in the past (stability) and the ability to learn new knowledge (plasticity) has a trade-off relationship. This trade-off relationship is called the stability-plasticity dilemma [17], and it is an important problem to deal with, especially in continual learning methods. The GNG-based methods can flexibly learn new knowledge by adaptively generating topological structures (i.e., nodes and edges) according to given data points. On the other hand, because the nodes and edges are excessively and persistently inserted, the GNG-based methods have the potential to cause catastrophic forgetting, where information learned in the past is forgotten when learning new data. Adaptive Resonance Theory (ART) [17] is one of the representative theories to avoid catastrophic forgetting. ART provides a learning mechanism that is inspired by the brain and is defined as a two-layer competitive learning neural network consisting of a feature representation layer and a category representation layer. Representative ART-based clustering methods include Fuzzy ART [18] and Bayes ART [19], and many methods that introduce topological structures have been proposed [20], [21], [22], [23], [24].

The purpose of this study is to adaptively and continuously build a global topological map using only the information from a 3D distance measurement sensor. To achieve this, we apply ART-based Topological Clustering (ATC), which can avoid catastrophic forgetting. Furthermore, we propose ATC with Different Topologies (ATC-DT), which has multiple topological structures, to extract attribute information necessary for autonomous mobile robots to plan paths, and realize path planning on the global topological map. The main contributions of this paper are as follows:

1) We propose a global topological mapping method that uses only measured 3D point cloud, resulting in a small memory consumption of the input vector.
2) We propose an algorithm that can place nodes at any threshold for application to the space perception of autonomous mobile robots.
3) We demonstrate that it is possible to build a global topological map in simulation environments and plan a return path form the trarget point to the starting point by utilizing the global topological map.

The rest of this paper is organized as: Section II summarizes the learning methods for 3D point cloud based on competitive learning related to this study, Section III explains the path planning problem in unknown environments in this study, Section IV proposes a method for building a topological map based on the proposed ATC, Section V explains the path planning method using the global topological map, and Section VI shows the simulation experiments to verify the effectiveness of the proposed method. Finally, Section VII draws the conclusions of the work.

II. RELATED WORKS

As mentioned above, many researches on building topological maps from 3D point cloud using competitive learning methods have proposed the use of GNG based learning methods. The GNG is able to place nodes appropriately to unknown data distribution by gradually increasing the number of nodes. Each edge has the concept of age, and the neighborhood relation can be changed according to the density distribution by deleting edges that exceed a threshold value [13]. Because of these characteristics, applying GNG to 3D point cloud is a learning method that not only enables the topological map building that preserves geometric information, but also enables downsampling and filtering, which are necessary in the 3D point cloud processing. References [27] and [28] have applied GNG to downsampling and filtering methods for extracting local features from 3D point cloud and shown that GNG enables the noise reduction of 3D point cloud and more accurate feature extraction. For learning color information included in the 3D point cloud information, [29] has proposed a method for accurate color assignment by utilizing the neighborhood structure and

adding weights based on the likelihood of the surrounding color information. In addition, as a method for learning normal vectors as attribute information included in input vectors, [30] has proposed a method for learning normal vectors when learning the GNG. However, the proposed feature extraction methods is basically based on point cloud, and GNG is used only for downsampling and filtering, and feature extraction that efficiently utilizes the neighborhood structure is not performed. These method requires the estimation of shape features such as normal vectors from raw 3D point cloud. On the other hand, [31] realized that the shape features, such as normal vectors, can be extracted by building local surfaces from the learned topological structure built by the GNG.

Moreover, when performing such feature extraction, it is essential to have a method to efficiently determine the appropriate number of nodes and neighborhood structure from the acquired point cloud information. References [32], [33], [34] have been proposed that focus on speeding up to enhance the adaptability of the GNG algorithm in order to apply it to dynamic data such as gesture and object tracking. Reference [32] has proposed the real-time 3D object tracking method that speeds up by using a uniform grid structure when selecting the winner node. References [33], [34] have proposed adaptive adjustment methods of criteria for adding and deleting nodes based on the probability density distribution of data and nodes, and the effectiveness has been demonstrated with real data. However, it is difficult to design parameters to achieve node placement at regular intervals, and it is difficult to use it as a methodology for global topological mapping for robots. On the other hand, [35] has proposed based on the euclidian distance between the first winner node and the input data, not the number of learning times, for the condition of adding nodes. This simple addition method based on such a distance base is easy to realize any node position interval. Therefore, in this study, we propose a method to add nodes using a similar method. Next, in the neighborhood structure construction, many methods have been proposed to perform triangulation during learning of the topological structure by GNG, and the effectiveness has been demonstrated in the field of 3D modeling where triangulation is required [30], [36], [37]. However, such a method of performing triangulation adds edges even in places where there is no data. Therefore, there is a possibility that edges will be generated in places where autonomous mobile robots cannot pass in the path planning.

In this way, GNG based topological structure learning methods can extract information included in the point cloud due to the design of the learning algorithm, and can learn any topological structure. Therefore, it is also used as a recognition technology in the field of robots, including autonomous mobile robots. For example, in [38] and [39], the Dynamic Density GNG (DD-GNG) is proposed as a method for detecting terrain surfaces in environmental recognition method for legged robots. This method increases the sampling of the point of interest from the features extracted by the topological structure learned so far to learn a dense topological structure for the point of interest, and is a method that adjusts the density, demonstrating the effectiveness in the real environment. Furthermore, as an extension of DD-GNG, [40] has proposed to detect dynamic objects, increase the node density of dynamic objects by using a node addition method based on Add-if-silent, and utilize it for obstacle avoidance of moving robots. However, in these proposed methods, the topological structure learned by GNG is used for robot perception, and the learning algorithm includes the dead node deletion rule. Such a rule of node deletion plays a very effective role in perception, but in the global topological map building, which is the subject of this study, there is a problem that the topological structure outside the range of measurement data is deleted. In addition, [16], [41], [42] have proposed the global topological map building methods for the path planning. However, these methods use all the point cloud measured so far as input, and as the environment becomes larger, there are problems such as the difficulty of efficient sampling for executing in real time and the memory space for holding 3D point cloud data. Therefore, in this paper, we propose a new methodology to continually learn a global topological map only from point cloud data measured by the 3D distance measurement sensor.

III. PROBLEM SETTING

This section explains the problem of path planning set in this paper. As shown in Fig. 1, this study assumes a robot (red sphere) placed in a 3D rough terrain environment, and the problem is set as the path planning for the robot to reach the set destination. The current position of the robot is assumed to be able to estimate its own position by SLAM. At this time, the point cloud drawn in gray in Fig. 1 (a) represents an unknown environment, and the green point cloud represents 3D point cloud data measured by a 3D distance sensor. The input data used for the the global topological map building is, in conventional research, all 3D point cloud that has become a known environment as the robot moves, as shown in (b). In this study, as shown in (c), only the 3D point cloud data measured at the current time by the 3D distance measurement sensor installed on the robot is used for learning.

IV. LEARNING ALGORITHM

A. ART-BASED TOPOLOGICAL CLUSTERING WITH DIFFERENT TOPOLOIES

In this study, we develop a method based on ATC, an unsupervised learning method applicable to unknown data structures for learning 3D point cloud. ATC is a method that can simultaneously learn the geometric structure of input vectors and preserve the topological structure. In the following, we explain the learning algorithm of ATC-DT.

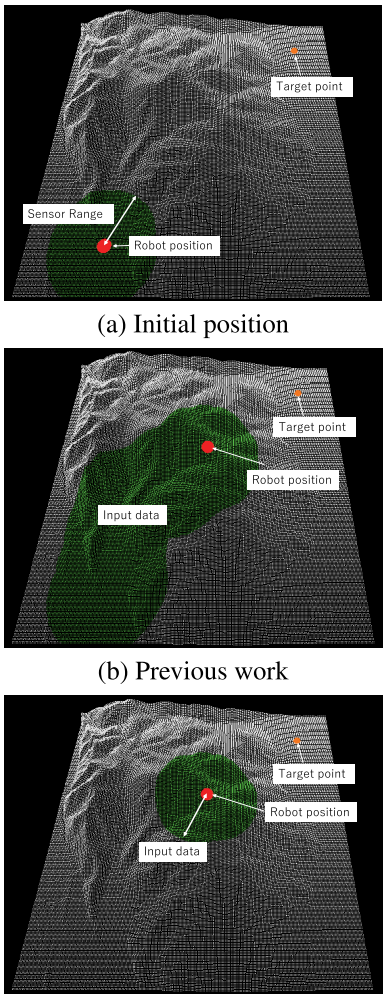


FIGURE 1. Terrain environment.

Firstly, we define the main variables used in ATC-DT. Firstly, we define the set of attributes in this study as $S = \{$position information(pos), normal vector(nor), tarversability(tra)$\}$, and we define the input vector and the reference vector as $\mathbf{p} = \{\mathbf{p}^{pos}\}$, $\mathbf{h}_i = \{\mathbf{h}_i^{pos}, \mathbf{h}_i^{nor}, h_i^{tra}\}$, respectively. Next, we define the distance d_i^o between the input vector and the i-th node's reference vector for a certain attribute o as follows.

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

In addition, by defining the set of edges of each attribute o ($\in S$) as $C^o = \{c_{1,2}^o, \cdots c_{i,j}^o, \cdots\}$, we learn the topological structures for each of multiple attributes. In the following, we explain the detailed contents of the overall algorithm whose flowchart is shown in Fig. 2.

Step 0. For initialization, set the learning times t to 0 and obtain the initial measurement data set P. Randomly generate two node reference vectors $\mathbf{h}_1$ and $\mathbf{h}_2$ and set the number of node wins to $M_1 = 1, M_2 = 1$.

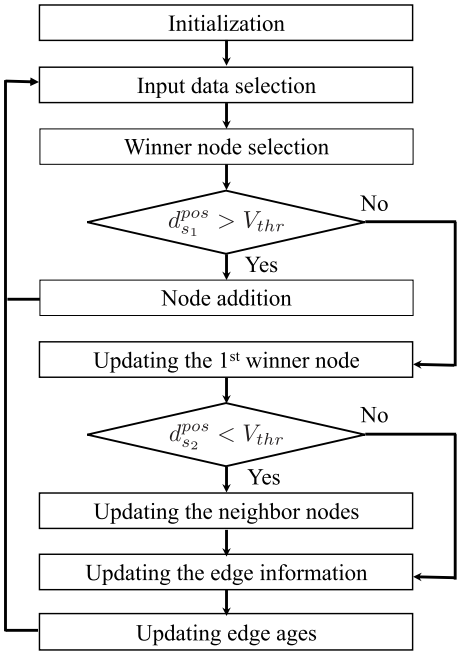


FIGURE 2. Flowchart of the learning algorithm.

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

Step 2. Select the 1st winner node s_1 and the 2nd winner node s_2 for the input vector $\mathbf{v}$.

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

where A represents the set of node numbers.

Step 3. If the distance $d_{s_1}^{pos}$ of the 1st winner node is smaller than the vigilance parameter V_{thr} ($d_{s_1}^{pos} < V_{thr}$), execute Step 4. Otherwise, add a node according to the following procedure,

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

where M_i represents the number of times the ith node became the 1st winner node, and N represents the number of nodes. After adding a node, execute Step 7.

Step 4. Update the reference vector of the 1st node s_1 according to the following equation,

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

Furthermore, when the distance $d_{s_2}^{pos}$ of the 2nd winner node is smaller than the vigilance parameter V_{thr} ($d_{s_2}^{pos} < V_{thr}$), the reference vector that has a connection with the 1st winner node is updated according to the following equation,

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

Step 5. Increment the age of the edge that has a neighborhood relation with the 1st winner node and the position information.

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

Furthermore, if the distance $d_{s_2}^{pos}$ to the 2nd winner node is smaller than the vigilance parameter V_{thr} ($d_{s_2}^{pos} < V_{thr}$), the age of the edge is reset to 1 ($g_{s1,s2} = 1$), 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 creation of edges for attributes o ($\in S^{pos}$) other than position information is performed according to the following equation,

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

where $f^o(\cdot)$ represents the result of the decision function according to the attribute, and the function returns true if the similarity between attributes is high, otherwise the function returns false. In addition, contour node detection is performed.

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

Step 7. Increment the learning count t, and update the measurement data set P every λ times. Then, return to Step 1.

Through this procedure, this study learns the terrain data of 3D point cloud. In the following sections, we will discuss the differences from previous studies in the learning algorithm and the method of extracting perceptual features used in path planning in this study.

B. UPDATING AND ADDING NODES

In the GNG-DT method, which was used in previous study [16] for learning topological structure, the update of nodes was performed according to the following equation,

$$\begin{aligned} \mathbf{h}_{s1} &\leftarrow \mathbf{h}_{s1} + \eta_1(\mathbf{p} - \mathbf{h}_{s1}), \\ \mathbf{h}_j^o &\leftarrow \mathbf{h}_j^o + \eta_2(\mathbf{p} - \mathbf{h}_j^o) \quad if \quad c_{s1,j}^{o} = 1, \end{aligned} \tag{8}$$

where η_1 and η_2 represent the learning rates, which were fixed values. Therefore, in this study, as the robot gradually moves and the density distribution of the data also gradually shifts, the node positions are updated accordingly. This led to a problem where it was not possible to maintain the geometric structure of the global topological map. On the other hand, in the proposed method, as shown in Eqs. (4) and (5), the learning rate decays each time it is selected as the first winner node, leading to a learning algorithm where the node positions converge.

In addition, in the conventional GNG-DT, nodes are added and the topological structure grows every time the number of learning times reaches a fixed number. On the other hand, in the proposed method, ATC-DT, as shown in Fig. 3, nodes are added and updated by dividing the cases into three based on the relationship between the input vector and the distance

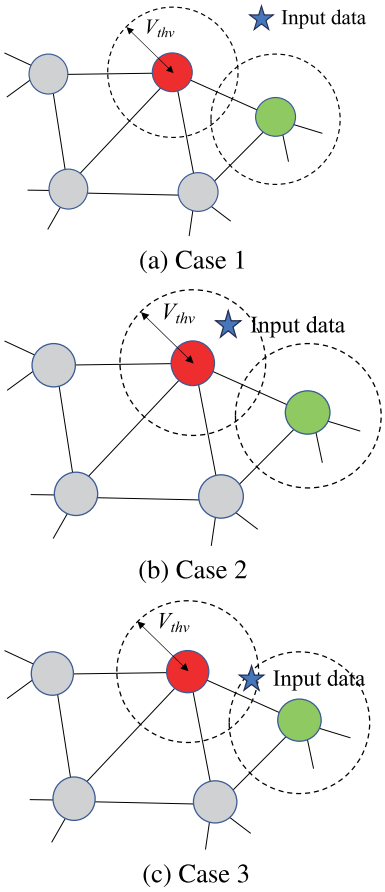


FIGURE 3. Learning case classification based on vigilance parameters. Red and green circles indicate the 1st and 2nd winner nodes, respectively.

of the 1st and 2nd winner nodes and the vigilance parameter. Specifically, when the positional relationship between the input vector and the 1st winner node is outside the range of the vigilance parameter as in Case 1 (Fig. 3(a)), nodes are added as in Eq. (3), and in other cases, Case 2 (Fig. 3 (b)), Case 3 (Fig. 3(c)), the 1st winner node is updated based on Eq. (4). Furthermore, when the positional relationship between the 2nd winner node and the input vector exists within the range of the vigilance parameter (Case 3), the nodes with a neighboring relationship with the first winner are updated based on Eq. (5), making it a learning algorithm that places nodes in appropriate positions. In this way, by using a method of updating and adding nodes based on the vigilance parameter, our proposed method improves the problem of topological structure deviation due to movement of the data distribution, which was a problem in the conventional method [16]. Furthermore, by designing the vigilance parameter V_{thr} in this study, it is possible to place nodes at any euclidean distance. This is a very important element in path planning from the learned topological structure, and it is a method that can build a topological map according to the robot embodiment.

C. UPDATING AGE THRESHOLD

In previous studies using GNG-DT, the age threshold for edge deletion was manually designed. On the other hand, in ATC-DT, the threshold is automatically determined based on the information of existing edges and edges that have been deleted in the past, similar to [25]. Specifically, the age threshold g_{max} of the edge is updated by the following equation,

$$g_{max} = \bar{\gamma}_{del}\frac{|\Gamma_{del}|}{|\Gamma_{del}|+|\Gamma|} + g_{thr}\left(1-\frac{|\Gamma_{del}|}{|\Gamma_{del}|+|\Gamma|}\right), \quad (9)$$

where Γ and Γ_{del} represent the set of edge ages held by the 1st winner node and the set of edge ages that have been deleted so far in learning, respectively. $\bar{\gamma}_{del}$ is the arithmetic mean of the set Γ_{del}. The coefficient g_{thr} is calculated by the following equation,

$$g_{thr} = \Gamma_{0.75} + \mathrm{IQR}(\Gamma), \quad (10)$$

where $\Gamma_{0.75}$ represents the 75th percentile of the set Γ, and $\mathrm{IQR}(\Gamma)$ represents the interquartile range.

D. TRAVERSABILITY ESTIMATION

In this study, as in [16], the slope angle of the learned terrain is used to determine the feasibility of travel for path planning. Specifically, in Step 4 of the ATC-DT learning algorithm, the normal vector is estimated for the 1st winner node s_1 using a method based on principal component analysis [43]. Specifically, the normal vector is calculated using the following equation for the covariance matrix F_i, considering the 1st winner node s_1 and its neighboring nodes as local surface elements.

$$F_{s_1} = \begin{bmatrix} \mathbf{h}_1^{pos} - \mathbf{h}_{s_1}^{pos} \, i \\ \vdots \\ \mathbf{h}_k^{pos} - \mathbf{h}_{s_1}^{pos} \end{bmatrix}^T \begin{bmatrix} \mathbf{h}_1^{pos} - \mathbf{h}_{s_1}^{pos} \\ \vdots \\ \mathbf{h}_k^{pos} - \mathbf{h}_{s_1}^{pos} \end{bmatrix}. \quad (11)$$

Next, compute the eigenvectors and eigenvalues from the covariance matrix F_{s_1}. At this time, the normal vector that becomes the feature is calculated as the eigenvector $\mathbf{h}_{s_1}^{nor}$ of the node, which has the minimum eigenvalue among the obtained eigenvalues.

In this way, the slope angle of the 1st winner node is calculated using the normal vector of the 1st winner node s_1 estimated in this way and the unit vector of the z axis in the global coordinate system $\mathbf{u}_z = (0, 0, 1)$ as follows.

$$deg_{s_1} = \cos^{-1}\left(\frac{\mathbf{h}_{s1}^{nor} \cdot \mathbf{u}_z}{\|\mathbf{h}_{s1}^{nor}\| \cdot \|\mathbf{u}_z\|}\right), \quad (12)$$

where deg_{s_1} represents the slope angle of the 1st winner node. Furthermore, if the maximum slope angle that the robot can travel is deg^{max}, the attribute of the node of the slope angle that can be traveled is determined as follows ($h_{s_1}^{tra}$),

$$h_{s_1}^{tra} = \begin{cases} 1 & \text{if } deg_{s_1} < deg^{max}. \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

Furthermore, using the traversability information, in Step 5, the traversability edge $c_{i,j}^{tra}$ of the ith and jth nodes is updated by the following equation,

$$c_{i,j}^{tra} = \begin{cases} 1 & \text{if } c_i^{tra} = c_j^{tra}, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

In this way, we learn the topological structure related to traversability (C^{tra}) and utilize it to path planning.

E. CONTOUR NODE DETECTION

In this study, we deal with path planning to reach a target point in an unknown environment. When realizing such path planning, it is important to perceive the boundary between the unknown and known environment. This boundary becomes the contour node of the global topological map learned by ATC-DT, so it is necessary to detect contour nodes from the topological structure. In this study, in the learning algorithm Step 5 of ATC-DT, we perform an approximate contour node detection of only the 1st winner node and its neighboring nodes, as in conventional research [16]. Specifically, we start by detecting the direction of gravity from the accelerometer mounted on the robot. In this study, this corresponds to the z-axis direction of the global coordinate system. Next, we project the position information of the i-th node ($\mathbf{h}_i^{pos}$) onto the vector in the direction of gravity. Finally, we place the group of nodes located in the vicinity of the i-th node in counterclockwise order. After that, we calculate the angle θ_k formed by adjacent nodes, and detect the contour nodes by performing threshold determination represented by the following equation,

$$z_i = \begin{cases} 1 & \text{if } \theta^{max} > \theta^{thr}, \\ 0 & \text{otherwise.} \end{cases}$$
$$\theta^{max} = \max_k \theta_k. \quad (15)$$

The contour nodes detected by this equation are considered as sub-goal candidates, and by selecting an appropriate sub-goal from among the sub-goal candidates, a path that can move to the destination is planned.

V. PATH PLANNING IN UNKNOWN ENVIRONMENTS

A. PATH COST

In this study, for path planning, we use the same method as the conventional one mentioned in [16]. As a path planning method, we use Dijkstra's algorithm [44], and use the topological structures C^{pos} and C^{tra}. The path cost in Dijkstra's algorithm not only uses the distance information between nodes, but also calculates and uses the slope angle information R_i using the normal information possessed by node i, as shown in the following equation,

$$R_i = \begin{cases} deg_i/deg^{max} & if \;\; 0 \le deg_i \le deg^{max}, \\ \infty & otherwise, \end{cases} \quad (16)$$

where R_i takes a value of $0 \le R_i \le 1$ as long as it does not exceed the maximum possible slope angle, with

values closer to 0 representing flatter areas and values closer to 1 representing steeper slopes. Furthermore, in order to consider the surrounding environment, we also use the angles of the nodes i, j and the nodes with edge $c_{i,j}^{pos}$ based on position information, and calculate the average value as follows,

$$r_{i,j}^{env} = \frac{\sum_{c_{i,n}^{pos}=1, n\in A} R_n}{N_i} + \frac{\sum_{c_{j,n}^{pos}=1, n\in A} R_n}{N_j}, \tag{17}$$

where N_i and N_j represent the number of nodes that have a neighborhood relationship with nodes i and j, respectively. Utilizing these elements, the slope cost $r_{i,j}$ is defined as follows,

$$r_{i,j} = R_i + R_j + r_{i,j}^{env}. \tag{18}$$

In this way, the slope cost is calculated by the topological structure C^{pos} of the normal vector and position held by the node. The obtained slope cost is used in the Dijkstra's algorithm with the topological structure C^{tra} of the traversability shown below, and determines the path.

Step 0. Initialize the cost from the start node s to each node to infinity.

Step 1. Set the cost $d_{i,j}^{cos}$ between all pairs of vertices.

$$d_{i,j}^{cos} = \begin{cases} 0 & \text{if } i = j, \\ \alpha^{dan} r_{i,j} + d_{i,j} & \text{if } c_{i,j} = 1, \\ \infty & \text{otherwise.} \end{cases} \tag{19}$$

Step 2. From the nodes with undetermined minimum cost, select the node i with the smallest cost and determine the minimum cost.

Step 3. For the node i confirmed in Step 2 and the unsettled node j that is in a neighboring relationship with it, if the sum of the costs with node i is smaller than the previous cost, update it.

$$d_{s,j}^{cos} \leftarrow d_{s,i}^{cos} + d_{i,j}^{cos} \quad \text{if } d_{s,j}^{cos} > d_{s,i}^{cos} + d_{i,j}^{cos}. \tag{20}$$

Step 4. Continue Steps 2 and 3 until the minimum cost to all nodes is determined, and calculate the minimum cost from node s to each node.

Where the coefficient α^{dan} in Eq. (19) of Step 1 is a coefficient related to the slope cost, and the smaller α^{dan} is, the more it plans the shortest distance path, and the larger it is, the more it can plan a path with a smaller slope.

B. SETTING SUB-GOALS IN UNKNOWN ENVIRONMENT

In path planning, if the target point is included in the global topological map, it is possible to plan the path. However, in an unknown environment, it cannot set the node that becomes the target point. Therefore, it is necessary to set sub-goals. In this study, by selecting the contour nodes of the topological structure constructed by ACT-DT as candidate points for sub-goals, the robot plans the path in an unknown environment. where if the set of candidate points for sub-goals is $A^{sub}(= \{i | z_i = 1 \cap h_i^{tra} = 1\})$, then the node s^{sub} that becomes the sub-goal is selected from the traversable nodes by the following equation,

$$\begin{aligned} s^{sub} &= \arg\min_{i\in A^{sub}} d_i^{sub}, \\ &= \arg\min_{i\in A^{sub}} \|\mathbf{p}^{tar} - \mathbf{h}_i^{pos}\|, \end{aligned} \tag{21}$$

where $\mathbf{p}^{tar}$ represents the position vector of the target point. In this way, by selecting the node closest to the destination from the traversable nodes as the sub-goal, the robot moves to the destination while knowing the environment to the target point. Even if the sub-goal selected is a dead end, the global topological map is updated in real time with the movement of the robot, so it is possible to reselect the best sub-goal and replan the path.

VI. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENTS

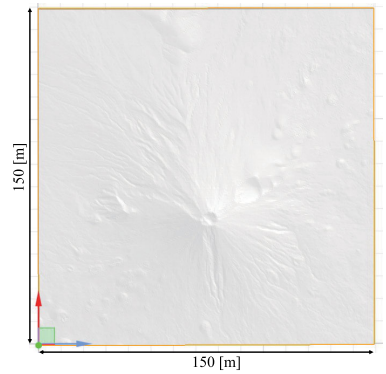
In order to verify the effectiveness of the proposed method, we constructed a simulation environment and conducted experiments. Specifically, we obtained point cloud of a 3D rough terrain environment from STL files of terrain data provided by the Geospatial Information Authority of Japan, and used it for the experiment. The terrain data used in this experiment is shown in Figs.4 and 5. In Fig. 4, except for a mountain in the center, it is a simple flat environment, and the size of the environment was set to 150[m] × 150 [m]. Also, the start point in path planning was set to $(x, y, z) = (5,5,0)$, and the target point was set to $(x, y, z) = (140,140,0)$. Next, in Fig. 5, in order to verify whether it can be applied to complex terrains, we used terrain data with severe undulations. where the size of the environment, the start point, and the target point were the same as in Fig. 4 for the experiment.

In this experiment, we used the conventional method of GNG-DT and Grow When Required (GWR) [45], which incorporates measures against catastrophic forgetting, as a method of comparing topological maps. In order to compare the performance of topological map building at any euclidean distance, we used the euclidean distance calculated by the following equation, which is different from the conventional method, for the calculation of activity a in GWR,

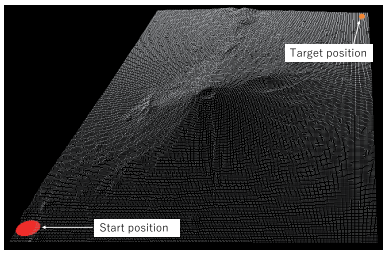
$$a = ||\mathbf{p}^{pos} - \mathbf{h}_{s_1}^{pos}||, \tag{22}$$

where s_1 represents the 1st winner node for the input vector $\mathbf{v}$, and if the activity a exceeds the activity threshold a_T, it was set as the node addition condition related to the activity of GWR. Also, the basic parameters related to ATC-DT used in the experiment are shown in Table 1, and for parameters excluding the node placement interval in the comparison methods GNG-DT and GWR, the ones used in [16] and [45] were used.

In the evaluation of the experiment, the Root Mean Square Error (RMSE), which is a quantization error that shows the

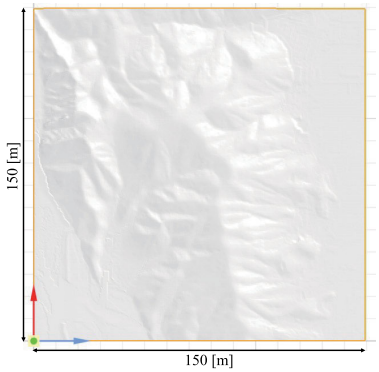


(a) Top view of STL data

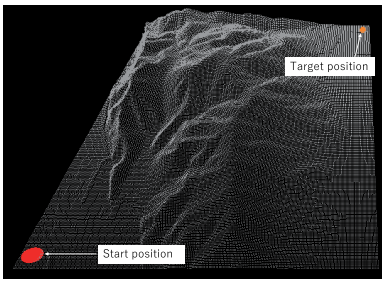


(b) Bird view of 3D point cloud

FIGURE 4. Experimental environment (Data 1).



(a) Top view of STL data



(b) Bird view of 3D point cloud

FIGURE 5. Experimental environment (Data 2).

TABLE 1. Experimental parameter setting.

Parameter	Value
λ	4000
α^{dec}	0.5
θ^{thr}	135.0 $[deg]$
deg^{max}	20.0 $[deg]$

basic performance of competitive learning, is used.

$$Eva_1 = \sqrt{\frac{1}{|V|}\sum_{\mathbf{v}\in V}\|\mathbf{p}-\mathbf{h}_{s_1}^{pos}\|^2}. \tag{23}$$

Furthermore, to verify whether nodes can be placed at any threshold, the average distance of the edges calculated by the following equation is used as an evaluation metric.

$$Eva_2 = \frac{1}{|C^{pos}|}\sum_{c_{i,j}^{pos}=1}\|\mathbf{h}_i^{pos}-\mathbf{h}_j^{pos}\|. \tag{24}$$

Furthermore, as a metric to determine whether learning converges, the maximum update amount of nodes in the node placement at the current time and one time before, calculated by the following equation, is used.

$$Eva_3 = \max_{i\in A(t)}\min_{j\in A(t-1)}\|\mathbf{h}_i^{pos}(t)-\mathbf{h}_j^{pos}(t-1)\|, \tag{25}$$

where $A(t)$ represents the set of node numbers at time t, and $\mathbf{h}_i^{pos}(t)$ represents the position of the ith node at time t. In this study, we used these metrics to verify the effectiveness of the experiment.

B. COMPARATIVE EXPERIMENT IN A KNOWN ENVIRONMENT

1) EXPERIMENTAL CONDITIONS

In this section, we conducted a comparative experiment by inputting all point cloud in each simulation environment to verify the basic learning performance of the proposed method. In addition, the purpose was to verify whether each proposed method can be placed at any euclidean distance, and the experiment was conducted with the threshold for adding and deleting nodes in each method set to 2.5, 5.0, 10.0. The number of point cloud inputs per trial was set to 1000 times. The trial was conducted 10 times and the evaluation was conducted using the average and variance of the evaluation metrics mentioned above.

2) EXPERIMENTAL RESULTS

The experimental results are shown in Fig. 6 and Table 2. The number of nodes of each method and the value of RSME are converging for all data sets and parameters. Also, as the number of nodes increases, the RSME decreases, indicating that nodes are being added appropriately, and Each method is able to learn 3D point cloud. This can also be confirmed from the learning result examples of 3D point cloud and topological map shown in Fig. 7. Furthermore, from the results of the average edge distance, the proposed method, ATC-DT and GWR, can learn with smaller errors compared to the conventional method, GNG-DT, for the set parameters, and GWR has the smallest errors in all results. Next, in the results of the maximum update amount of

nodes from one time before, GNG-DT continues to have a large update amount until the end of learning. The update amounts of ATC-DT and GWR are smaller compared to GNG-DT, and in ATC-DT, except for the results where the threshold is 2.5, the update amount at the end of learning is 0.00, indicating that learning has completely converged.

3) DISCUSSION

From the results of the distance between edges, it is concluded that GNG-DT cannot place nodes at any distance using the threshold value τ^{del}. This is thought to be because the threshold τ^{del} represents an approximate value of quantization error in learning, not a parameter in node placement. Therefore, when comparing the set threshold value and the value of quantization error in GNG-DT, it is found that the results are almost identical. On the other hand, GWR can place the nodes according to the set distance threshold, and GWR has obtained the result with the smallest error with the set parameter. From this, GWR is a method that can place nodes according to any distance, and has the smallest error result for the set threshold. However, in GWR, as shown in Fig. 8, there are example results where nodes are generated in places where there is no data distribution in the learning process. This is thought to be because the addition of nodes in GWR is a method of adding a new node at the midpoint between the input vector and the 1st winner node, and nodes are actually added where there is no data distribution when the initial nodes are sparse, and the nodes remain as they are. On the other hand, the proposed method, ATC-DT, does not produce any dead nodes in all trials, and although the offset error is large compared to GWR, it is able to place nodes at a value close to the set threshold. From this, it is thought that the proposed method, ATC-DT, is a method that can appropriately learn the geometric structure of 3D point cloud at any granularity.

Next, in the results of the node update amount, it is considered that GNG-DT oscillates with a constant update amount because it uses a fixed learning rate at the time of node update, so the learning about node update does not converge. On the other hand, ATC-DT and GWR use a method that reduces the learning rate based on the input vector to converge the node update, so they take values close to 0 at the end of learning. In particular, in ATC-DT, the update amount is 0 in all results, except for the result with a vigilance parameter value of 2.5. Therefore, it is considered that the proposed method of using the winner count for adjusting the learning rate is a method that can easily converge the node position to a stable position compared to other methods. This convergence property is a very important in the global topological map building, which is the purpose of this study, and the proposed method using ATC-DT is an effective methodology, including the ability to place nodes according to any threshold.

C. COMPARATIVE EXPERIMENT IN UNKNOWN ENVIRONMENT

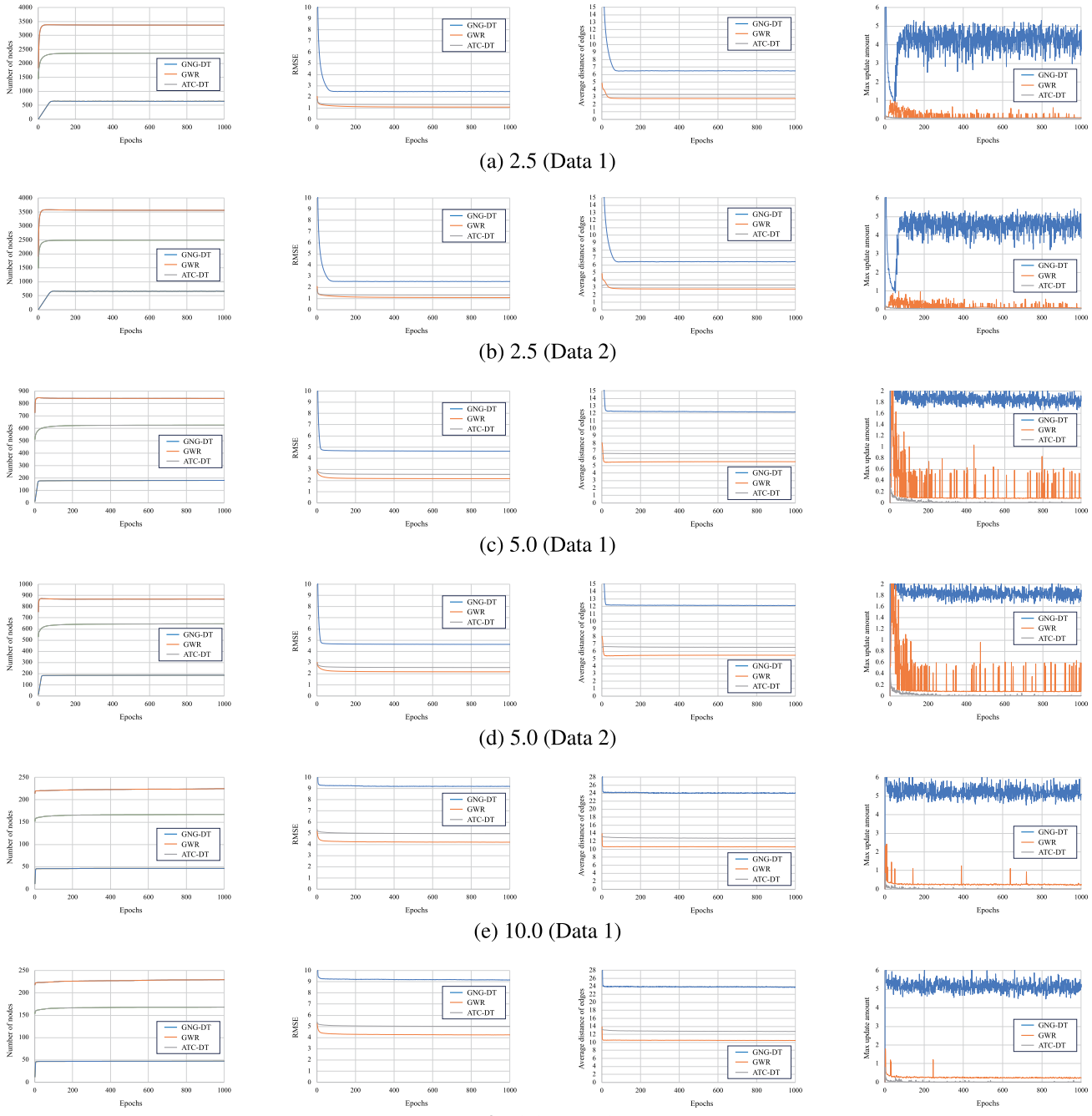
1) EXPERIMENTAL CONDITIONS

Next, to verify the effectiveness of a topological map building using only measurement data, we conducted a path planning experiment in an unknown environment, similar to [16]. In this experiment, we used GNG-DT and GWR as comparative methods for the proposed method. In GNG-DT, we prepared two patterns: one using the current measurement data as the input vector set (GNG-DT1) and the other using all the 3D point clouds measured so far as the input vector set (GNG-DT2), similar to [16]. The parameters of each method related to the node placement interval were set to 5.0. In GNG-DT, since nodes cannot be placed according to the parameters, we used 2.5, which was the closest to 5.0 in the previous experimental results, as the parameter. Each method was tried 10 times, and the evaluation metric used the RMSE, the average distance between edges, the number of nodes, and the number of data in the input vector set, where to verify the learning performance of the global topological map, we calculated the RMSE for all the point cloud sets used as input vectors so far, not the input vector set at the current time, and used it as an indicator. As a new evaluation index, we added the number of successful return path planning and the path length when planning the path from the target point to the start point after reaching the target point.

2) EXPERIMENTAL RESULTS

The experimental results are shown in Table 3, and an example of a global topological map building is shown in Fig. 9. In Table 3, in the path planning experiment in an unknown environment, the length of the planned path changes with each trial and the number of learning times is different, so the results of each index when arriving at the target point are summarized. From the experimental results, the number of input point cloud is about one-tenth of both environments compared to [16] in the proposed method, ATC-DT, GNG-DT1, GWR. Next, when looking at the results of the number of nodes and RMSE, which are indicators of the results of the global topological map building, in GNG-DT1, which learned only with the measured point cloud, the number of nodes is small and RMSE is also a large value, and the global topological map has not been built. On the other hand, other methods including the proposed method have a larger number of nodes compared to GNG-DT1, and RMSE is about 2, and a global topological map has been build. In terms of the average distance of the edges, the proposed method, ATC-DT, has the smallest error in both results compared to the set node, which is one interval of 5.0.

Next, to verify the performance of topological map building, the number of successful return path plans from the target point to the start point after arrival, and the average value of path length are shown in Table 4. In addition, the results of the path moved from the start point to the target point and the return path in each method are shown in Fig. 10.



(a) 2.5 (Data 1)

(b) 2.5 (Data 2)

(c) 5.0 (Data 1)

(d) 5.0 (Data 2)

(e) 10.0 (Data 1)

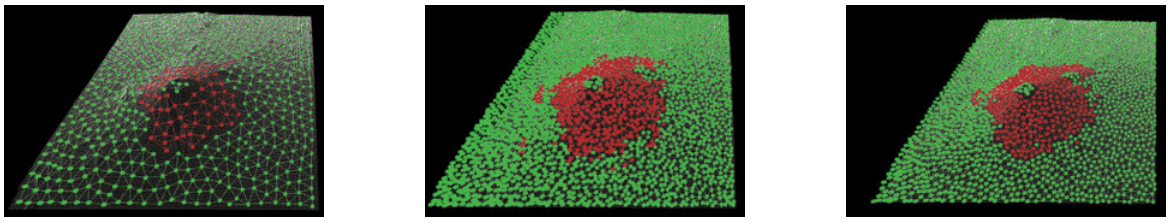
(f) 10.0 (Data 2)

FIGURE 6. **Experimental results on the transition of each evaluation. The subfigure results represent, from left to right, the evaluation metrics: Number of nodes, RMSE, Edge distance, and Maximum update amount. The values in the subcaptions represent the node interval thresholds set for each method.**

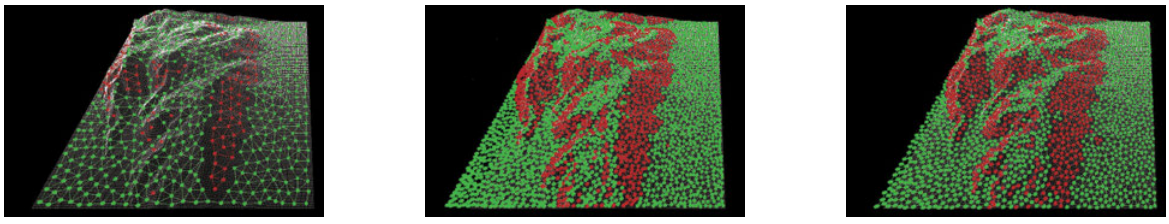
As a result, in GNG-DT1, it was not possible to plan a return path in all trials. Next, in GWR, although it was successful 8 times in the Data 1 environment, it was only successful 3 times in the Data 2 environment with more complex terrain. On the other hand, in the proposed method ATC-DT and the conventional method GNG-DT2, it was possible to plan a return path in all trials in both environments. In addition, when comparing the average value of the path length, in the Data 1 environment, ATC-DT had the shortest result, and in the Data 2 environment, GNG-DT2 had the shortest result.

3) DISCUSSION

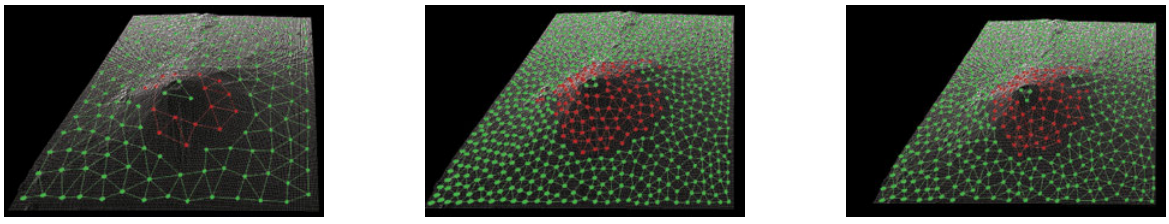
The results show that GNG-DT1 could build only a global topological structure using the measured point cloud. This is because in GNG-DT, a rule is incorporated to delete nodes



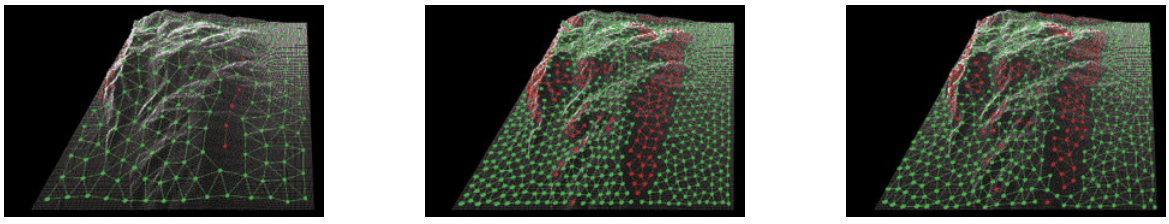
(a) 2.5 (Data 1)



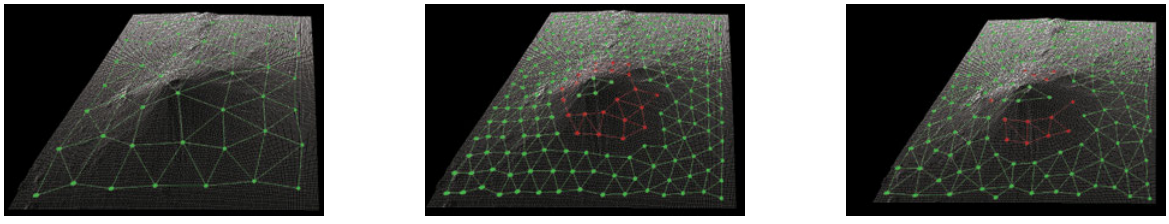
(b) 2.5 (Data 2)



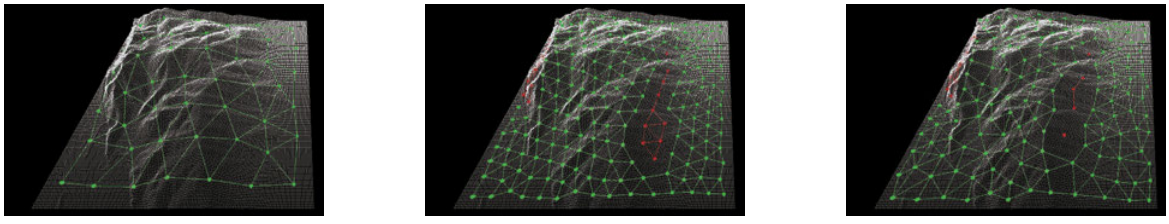
(c) 5.0 (Data 1)



(d) 5.0 (Data 2)



(e) 10.0 (Data 1)



(f) 10.0 (Data 2)

FIGURE 7. Learning examples. The green and red spheres indicate traversable and untraversable nodes, respectively.The results for each method show the cases with the smallest number of nodes among 10 trials. The results in the subfigure represent the topological structures built using all point cloud data by GNG-DT, GWR and ATC-DT, from left to right. The values in the sub-captions represent the node interval thresholds set by each method.

TABLE 2. Learning results at the end of each method.

Dataset	Method	Parameter	Number of Nodes	Eva_1	Eva_2	Eva_3
Data 1	GNG-DT	2.5	647.4±7.1	2.50±0.02	6.47±0.03	3.65±2.15
	GWR		3376.4±49.5	1.08±0.01	2.78±0.02	0.07±0.02
	ATC-DT		2373.1±12.3	1.31±0.00	3.34±0.01	0.04±0.01
	GNG-DT	5.0	181.5±1.43	4.62±0.02	12.19±0.05	1.82±0.21
	GWR		840.5±13.4	2.16±0.02	5.52±0.04	0.08±0.01
	ATC-DT		625.9±5.97	2.55±0.01	6.56±0.03	0.00±0.00
	GNG-DT	10.0	46.2±0.4	9.20±0.05	24.07±0.19	5.09±0.89
	GWR		225.0±5.52	4.21±0.05	10.58±0.13	0.25±0.07
	ATC-DT		167.2±3.08	4.95±0.04	12.75±0.13	0.00±0.00
Data 2	GNG-DT	2.5	665.0±8.01	2.52±0.02	6.48±0.03	3.83±1.76
	GWR		3358.2±15.1	1.08±0.00	2.78±0.03	0.08±0.01
	ATC-DT		2486.4±10.27	1.31±0.00	3.32±0.01	0.06±0.01
	GNG-DT	5.0	186.3±1.2	4.63±0.02	12.11±0.04	1.78±0.31
	GWR		868.4±12.8	2.17±0.01	5.48±0.05	0.59±1.60
	ATC-DT		647.3±6.7	2.56±0.01	6.53±0.03	0.00±0.00
	GNG-DT	10.0	47.4±0.70	9.14±0.06	23.86±0.21	5.11±0.68
	GWR		230.0±3.3	4.23±0.03	10.49±0.08	0.23±0.05
	ATC-DT		168.2±3.6	5.00±0.06	12.77±0.13	0.00±0.00

TABLE 3. Learning results of the topological mapping.

Dataset	Method	Number of Inputs	Number of Nodes	Eva_1	Eva_2
Data 1	GNG-DT1	3230.2±254.5	45.0±3.9	86.86±3.07	5.60±0.09
	GNG-DT2	305515.5±765.6	334.1±9.4	2.36±0.01	6.11±0.03
	GWR	3349.1±187.6	603.9±23.9	1.99±0.02	4.24±0.02
	ATC-DT	3357.8±221.5	445.4±13.4	2.08±0.01	5.41±0.03
Data 2	GNG-DT1	3147.1±192.4	39.3±3.1	86.36±2.25	5.74±0.10
	GNG-DT2	31679.9±508.0	358.1±6.5	2.37±0.01	6.11±0.01
	GWR	3162.3.5±202.8	632.7±27.4	2.08±0.12	4.15±0.12
	ATC-DT	3220.6.9±180.9	475.1±7.0	2.10±0.01	5.38±0.02

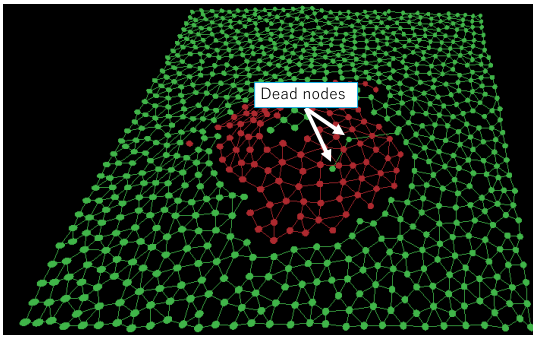


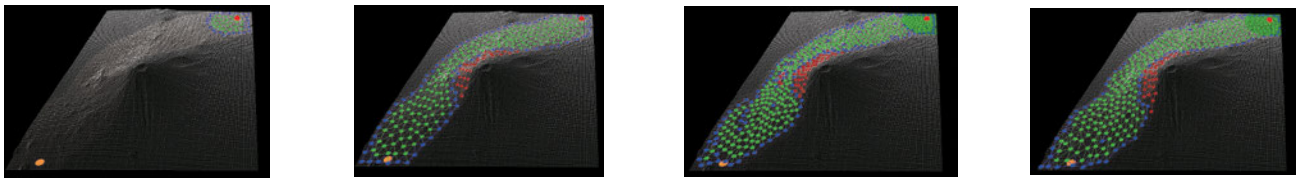
FIGURE 8. An example of dead node occurrence in GWR.

TABLE 4. Experimental results of the return path planning.

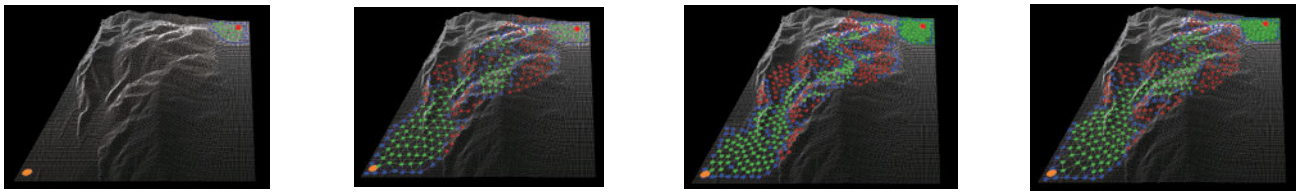
Dataset	Method	Number of Successes	Path Length
Data 1	GNG-DT1	0	-
	GNG-DT2	10	224.3±4.9
	GWR	8	231.8±6.2
	ATC-DT	10	220.1±3.5
Data 2	GNG-DT1	0	-
	GNG-DT2	10	257.4±7.5
	GWR	3	271.0±1.4
	ATC-DT	10	263.8±7.2

that are not selected as winner nodes in order to adapt to dynamic data, resulting in the topological structure within the measurement range. On the other hand, GWR could preserve nodes outside the measurement range, and to some extent, GWR could build a global topological map. However, there were cases where it is not possible to plan a return path, especially in Data 2, where it was only successful three times. This is because in GWR, the learning rate is converged by the number of learning times to converge the node update, and finally, GWR becomes a learning rule that converges to a certain fixed value. Therefore, if the robot is stagnant near a certain point, the structure of the global topological map will be disrupted due to being dragged by the learning of the input measurement point cloud. Fig. 11 shows an example of a topological map in GWR, but the start point and the target point were topologically clustered as different clusters, and it can be confirmed that a path from the start point to the target point was not generated.

On the other hand, in both ATC-DT and GNG-DT2, a return path could be planned in all trials in both

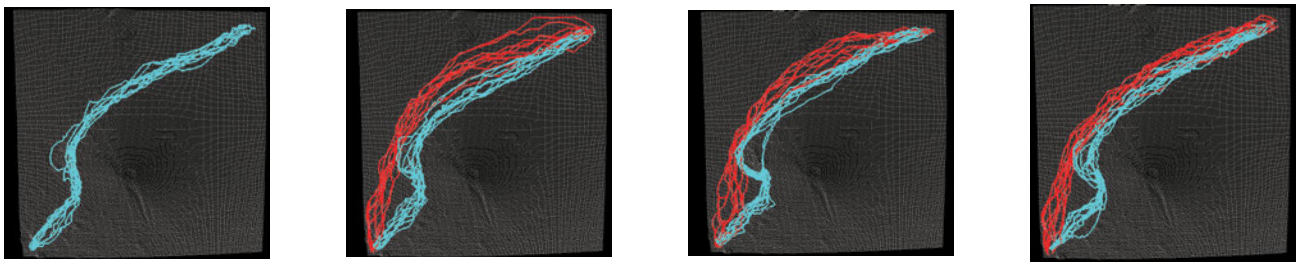


(a) Data 1

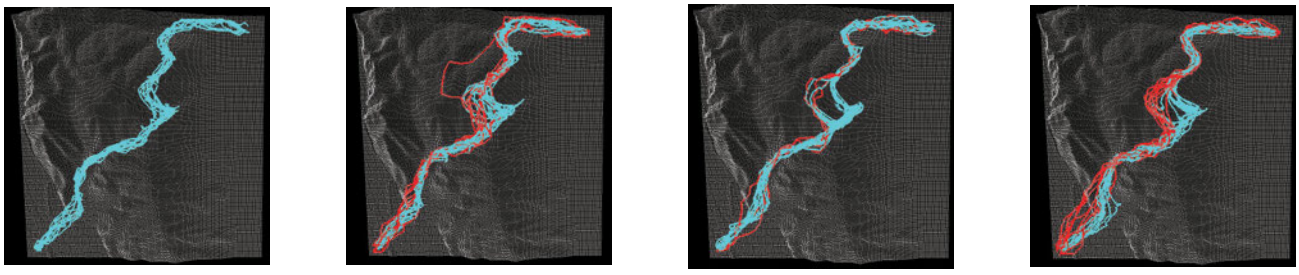


(b) Data 2

FIGURE 9. Examples of the learnign result. Green, red, and blue spheres indicate the traversable, untraversable, and contour nodes, respectively.The results for each method show the cases with the smallest travel distance among 10 trials. The results in the subfigure represent the topological structures built using only point cloud data measured by GNG-DT1, GNG-DT2, GWR and ATC-DT, from left to right.



(a) Data 1



(b) Data 2

FIGURE 10. Experimental results of the return path planning. Red and blue lines indicate the return path and robot trajectory from starting point to end point, respectively. The results in the subfigure represent the paths planned using the topological structures built by GNG-DT1, GNG-DT2, GWR and ATC-DT, from left to right.

environments. However, in GNG-DT2, since learning is performed using all point cloud, a global topological map is built, so GNG-DT2 is natural that a return map can be planned. However, in the proposed method, ATC-DT, it is possible to build a global topological map only from the measured point cloud and the robot's position information. This is because the learning rate in ATC-DT converges to 0 as shown in Eq. (4) due to the winner count, and the update amount of the node has converged because ATC-DT could be placed in the appropriate node position as learning progresses. This can also be confirmed from the fact that the error between the average value of the distances between the edges and the target value is the smallest in both environments. Furthermore, in path planning in unknown environments, even if the robot encounters a dead end, the robot plans the shortest path until the robot perceives an untraversable area, due to the unknown ahead, in both Data 1 and Data 2. After that, when the robot perceives the untraverable area, the robot avoids that area and becomes a travel path to reach the target point. On the other hand, the return path can plan a path to avoid that area from the beginning. In fact, the average travel distance of the robot in each environment in ATC-DT is 239.1 ± 16.4 for Data 1 and 321.2 ± 74.6 for Data 2. Therefore, an efficient path could be planned because the travel distances are longer distances compared to the results in Table 4. From the above, the

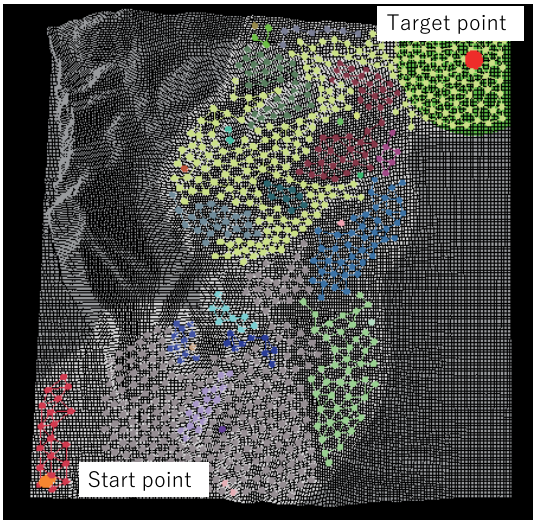


FIGURE 11. An example of topological clustering results using GWR. Different colors indicate different clusters.

proposed method is a method for building an efficient global topological map for autonomous mobile robots.

VII. CONCLUSION

In this paper, we conducted research with the aim of building a global topological map from only the measured 3D point cloud and the robot's position information. Specifically, in the conventional learning method based on GNG, it was impossible to preserve a global geometric structure when only the measured data was used for learning because the nodes that become dead nodes were deleted. Therefore, we proposed a learning method that can learn without forgetting the point cloud measured in the past by using the topological clustering method based on the ART, which is a learning approach to avoid catastrophic forgetting, as a learning method. Furthermore, by extending to ATC-DT, a learning method with a topological structure of multiple attributes used in conventional methods, we proposed a method that allows for perception of traversability against terrain data from unknown 3D point cloud and path planning in unknown environments, just like conventional methods. In the experiment, we showed that in simulation environments using the terrain data of the 3D point cloud, the arbitrarily set nodes can be placed at appropriate intervals, although there is an offset error for the threshold of the position interval. In addition, in the path planning experiment in an unknown environment, we showed that a global topological map can be built accurately and stably using only the measured data and the robot's position information compared to other competitive learning methods.

However, in the experiments in this paper, we have only stayed in the simulation environment and have not verified the effectiveness using autonomous mobile robots in the real environment. Therefore, as a future task, we will verify the effectiveness of the proposed method in the real environment integrated with the self-localization method using 3D LiDAR. In addition, the proposed method is expected to be difficult to apply in large-scale environments because the computational cost increases as the number of nodes increases. Therefore, we also plan to construct a learning algorithm that suppresses the amount of computation for a large number of nodes.

REFERENCES

[1] M. Shu, G. Chen, and Z. Zhang, "3D point cloud-based indoor mobile robot in 6-DoF pose localization using a Wi-Fi-aided localization system," *IEEE Access*, vol. 9, pp. 38636–38648, 2021.
[2] C.-Y. Tsai, H. Nisar, and Y.-C. Hu, "Mapless LiDAR navigation control of wheeled mobile robots based on deep imitation learning," *IEEE Access*, vol. 9, pp. 117527–117541, 2021.
[3] Y. Lin, Y. Wang, S. Wang, S. Li, M. Wang, H. Cai, and F. Teng, "Noise point detection from airborne LiDAR point cloud based on spatial hierarchical directional relationship," *IEEE Access*, vol. 10, pp. 82076–82091, 2022.
[4] L. Sun, "Practical, fast and robust point cloud registration for scene stitching and object localization," *IEEE Access*, vol. 10, pp. 3962–3978, 2022.
[5] J. H. Park, Y. E. Lim, J. H. Choi, and M. J. Hwang, "Trajectory-based 3D point cloud ROI determination methods for autonomous mobile robot," *IEEE Access*, vol. 11, pp. 8504–8522, 2023.
[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] S. Niijima, R. Umeyama, Y. Sasaki, and H. Mizoguchi, "Semi-automatic town-scale 3D mapping using building information from publicly available maps," *IEEE Access*, vol. 10, pp. 32244–32254, 2022.
[8] Y. Zheng, S. Chen, and H. Cheng, "Real-time cloud visual simultaneous localization and mapping for indoor service robots," *IEEE Access*, vol. 8, pp. 16816–16829, 2020.
[9] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, "A review of deep learning-based semantic segmentation for point cloud," *IEEE Access*, vol. 7, pp. 179118–179133, 2019.
[10] M. Ibrahim, N. Akhtar, M. Wise, and A. Mian, "Annotation tool and urban dataset for 3D point cloud semantic segmentation," *IEEE Access*, vol. 9, pp. 35984–35996, 2021.
[11] J. Seol, J. Kim, and H. I. Son, "Spray drift segmentation for intelligent spraying system using 3D point cloud deep learning framework," *IEEE Access*, vol. 10, pp. 77263–77271, 2022.
[12] J. Zhang, H. Jiang, H. Shao, Q. Song, X. Wang, and D. Zong, "Semantic segmentation of in-vehicle point cloud with improved RangeNet++ loss function," *IEEE Access*, vol. 11, pp. 8569–8580, 2023.
[13] B. Fritzke, "A growing neural gas network learns topologies," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 7, 1995, pp. 625–632.
[14] 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.
[15] K. Ozasa, Y. Toda, and T. Matsuno, "Growing neural gas based traversability clustering for an autonomous robot," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–6.
[16] 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.
[17] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis., Graph., Image Process.*, vol. 36, nos. 2–3, p. 396, Nov. 1986.
[18] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Netw.*, vol. 4, no. 6, pp. 759–771, Jan. 1991.
[19] B. Vigdor and B. Lerner, "The Bayesian ARTMAP," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1628–1644, Nov. 2007.
[20] M. Tscherepanow, "The Bayesian ARTMAP," in *Proc. Int. Conf. Artif. Neural Netw.*, 2010, pp. 157–167.
[21] 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*.

[22] 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.
[23] N. Masuyama, N. Amako, Y. Nojima, Y. Liu, C. K. Loo, and H. Ishibuchi, "Fast topological adaptive resonance theory based on correntropy induced metric," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 2215–2221.
[24] N. Masuyama, C. K. Loo, H. Ishibuchi, N. Kubota, Y. Nojima, and Y. Liu, "Topological clustering via adaptive resonance theory with information theoretic learning," *IEEE Access*, vol. 7, pp. 76920–76936, 2019.
[25] C. Wiwatcharakoses and D. Berrar, "SOINN+, a self-organizing incremental neural network for unsupervised learning from noisy data streams," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113069.
[26] D. Viejo, J. Garcia-Rodriguez, and M. Cazorla, "Combining visual features and growing neural gas networks for robotic 3D SLAM," *Inf. Sci.*, vol. 276, pp. 174–185, Aug. 2014.
[27] J. García-Rodríguez, M. Cazorla, S. Orts-Escolano, and V. Morell, "Improving 3D keypoint detection from noisy data using growing neural gas," *Adv. Comput. Intell.*, vol. 1, pp. 1–20, Aug. 2013.
[28] J. C. Rangel, V. Morell, M. Cazorla, S. Orts-Escolano, and J. García-Rodríguez, "Object recognition in noisy RGB-D data using GNG," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1061–1076, Nov. 2017.
[29] M. Saval-Calvo, J. Azorin-Lopez, A. Fuster-Guillo, J. Garcia-Rodriguez, S. Orts-Escolano, and A. Garcia-Garcia, "Evaluation of sampling method effects in 3D non-rigid registration," *Neural Comput. Appl.*, vol. 28, no. 5, pp. 953–967, May 2017.
[30] S. Orts-Escolano, J. Garcia-Rodriguez, V. Morell, M. Cazorla, J. A. S. Perez, and A. Garcia-Garcia, "3D surface reconstruction of noisy point clouds using growing neural gas: 3D object/scene reconstruction," *Neural Process. Lett.*, vol. 43, no. 2, pp. 401–423, Apr. 2016.
[31] Y. Toda, Z. Ju, H. Yu, N. Takesue, K. Wada, and N. Kubota, "Real-time 3D point cloud segmentation using growing neural gas with utility," in *Proc. 9th Int. Conf. Human Syst. Interact. (HSI)*, Jul. 2016, pp. 418–422.
[32] S. Orts-Escolano, J. Garcia-Rodriguez, M. Cazorla, V. Morell, J. Azorin, M. Saval, A. Garcia-Garcia, and V. Villena, "Bioinspired point cloud representation: 3D object tracking," *Neural Comput. Appl.*, vol. 29, no. 9, pp. 663–672, May 2018.
[33] A. Angelopoulou, J. Garcia Rodriguez, S. Orts-Escolano, G. Gupta, and A. Psarrou, "Fast 2D/3D object representation with growing neural gas," *Neural Comput. Appl.*, vol. 29, no. 10, pp. 903–919, May 2018.
[34] H. Frezza-Buet, "Online computing of non-stationary distributions velocity fields by an accuracy controlled growing neural gas," *Neural Netw.*, vol. 60, pp. 203–221, Dec. 2014.
[35] C. A. Mueller, N. Hochgeschwender, and P. G. Ploeger, "Surface reconstruction with growing neural gas," in *Proc. Act. Semantic Perception Object Search Real World Workshop Intell. Robots Systems*, 2011, pp. 1–24.
[36] R. L. M. E. do Rego, A. F. R. Araujo, and F. B. de Lima Neto, "Growing self-organizing maps for surface reconstruction from unstructured point clouds," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2007, pp. 1900–1905.
[37] N. Kubota, T. Narita, and B. H. Lee, "3D topological reconstruction based on Hough transform and growing neural gas for informationally structured space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 3459–3464.
[38] A. A. Saputra, N. Takesue, K. Wada, A. J. Ijspeert, and N. Kubota, "AQuRo: A cat-like adaptive quadruped robot with novel bio-inspired capabilities," *Frontiers Robot. AI*, vol. 8, pp. 1–24, Apr. 2021.
[39] A. A. Saputra, J. Botzheim, and N. Kubota, "Neuro-cognitive locomotion with dynamic attention on topological structure," *Machines*, vol. 11, no. 6, p. 619, Jun. 2023.
[40] M. Shoji, T. Obo, and N. Kubota, "Add-if-Silent rule-based growing neural gas with amount of movement for high-density topological structure generation of dynamic object," in *Proc. IEEE Int. Conf. Syst. Man, Cybern. (SMC)*, Oct. 2023, pp. 3040–3047.
[41] A. A. Saputra, Y. Toda, J. Botzheim, and N. Kubota, "Neuro-activity-based dynamic path planner for 3-D rough terrain," *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 2, pp. 138–150, Jun. 2018.
[42] 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.
[43] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proc. IEEE Visualizat., VIS.*, Jun. 2002, pp. 163–170.
[44] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
[45] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Netw.*, vol. 15, nos. 8–9, pp. 1041–1058, Oct. 2002.

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 robot.

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-shi, Japan. His current research interests include clustering, data mining, and continual learning.