IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 October 2024, accepted 9 November 2024, date of publication 15 November 2024, date of current version 26 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3499364

RESEARCH ARTICLE

Passability-Based Local Planner Using Growing Neural Gas for an Autonomous Mobile Robot

KOKI OZASA[1], YUICHIRO TODA[1], (Member, IEEE), YOSHIMASA NAKAMURA[2], TOSHIKI MASUDA[2], HIROHIDE KONISHI[3], AND TAKAYUKI MATSUNO[1], (Member, IEEE)

[1]Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan
[2]Tokyo Metropolitan Industrial Technology Research Institute, Tokyo 135-0064, Japan
[3]NSK Ltd., Tokyo 141-8560, 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) KAKENHI under Grant JP24K20870, and in part by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP24022188.

ABSTRACT 3D spatial perception is one of the most important abilities for autonomous mobile robots. In environments with unknown objects, the ability to perform a local planner, which modifies the global path based on the perception results, is also required as an indispensable capability. In this paper, we propose a method based on Growing Neural Gas with Different Topologies (GNG-DT), which can be applied to unknown data, as a method for 3D spatial perception and local planner in unknown environments. First, we propose a method for extracting travelability perceptions from the features estimated by the topological structure of the GNG-DT. Next, we learn the topological structure of passability information based on the size of the robot from the extracted traversability percepts. Furthermore, we propose a local planner that uses the topological structure of traversability and passability learned from the point cloud currently perceived by the robot. In the experiments, we compared the cases where only traversability was used and where passability information was used in actual environments, and showed that the proposed method can plan a route that determines the area that the robot can actually pass through.

INDEX TERMS Autonomous mobile robot, growing neural gas, local planner.

I. INTRODUCTION

With the development of robotics and networking technologies, autonomous mobile robots have been becoming popular in various locations [1], [2], [3], [4]. In particular, there are growing expectations for autonomous mobile robots in outdoor environments [5], [6], where they can be used for a wide range of applications, from transportation tasks to exploration at disaster sites. In moving in outdoor environments with many dynamic objects and unknown targets, robots need to adaptively detect traversable surfaces, and the ability to perceive 3D space in real time has become an indispensable technology. To achieve such adaptive 3D space perception, many environment recognition techniques based on deep learning using large datasets have been proposed in recent years, and highly accurate space perception is becoming a reality [7], [8]. However, since these methods are basically based on supervised learning, they require labeling work on large data sets and still have poor adaptability to unknown situations. This is due to the fact that 3D point cloud data is unstructured data, which prevents proper recognition of clusters for unknown objects. Moreover, real-time perception is required for autonomous movement, and learning based on deep learning requires parallel processing on GPGPUs. In this study, we propose a method that can be applied to unknown data distributions without the need for a large data set by structuring the data through learning 3D point cloud data based on unsupervised competitive learning. Furthermore, we propose a 3D space perception and movement planning method that utilizes the topological structure constructed by competitive learning to solve the above-mentioned problems.

As such an approach, the authors have studied a 3D space perception method based on Growing Neural Gas (GNG) [9], [10], [11], which is one of the unsupervised learning methods

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

and learns the topological structure described by nodes and edges [12]. GNG includes dynamic addition and deletion of nodes and edges in the learning algorithm, and thus can accurately learn the geometric information of 3D point cloud data measured from LiDARs and RGB-D cameras mounted on a robot while building the topological structure. Learning with GNG not only enables downsampling and noise reduction of huge point cloud data, but also enables feature extraction such as normal vectors and topological clustering necessary for space perception using the neighborhood structure by using the topological structure. On the basis of the above, we believe that a unified framework for 3D space perception systems can be realized by using GNG-based methodologies. However, conventional clustering in GNG involves learning the geometric information of a set of points using GNG and then removing edges using attribute values such as color information, making it impossible to obtain clustering results with different attributes at the same time.

To solve this problem, the authors proposed GNG with Different Topologies (GNG-DT) [13], which generates multiple clustering results with different interpretations by learning the topological structure for each different attribute [14], [15]. GNG-DT has enabled clustering of reference vectors with different attributes such as position, color, and normal information. We have also proposed a traversability clustering system based on GNG-DT as a perception system for autonomous robots, and have verified its effectiveness by global path planning in simulation and real environments. However, the robot needs to adapt to obstacles and dynamic objects that did not exist in the path when moving, and a local planner is required to correct the path for safe movement. A local planner that corrects the path is necessary for the robot to move safely.

What is required of the local planner of an autonomous mobile robot is the ability to perceive unknown objects in real time, to perform traversability perception, and to plan a route that the robot can pass according to the robot's embodiment. Therefore, this paper proposes a control method based on real-time passability node detection and obtained local planner paths using GNG-DT. The contributions of this paper are (1) realization of passability clustering perception by combining multiple feature attributes using GNG-DT, (2) proposal of a local planner based on physicality-based passability determination, and (3) demonstration of flexible route modification in real environments.

II. RELATED WORKS

A. COMPETITIVE LEARNING BASED POINT CLOUD PERCEPTION METHOD

In this study, as a learning method for 3D space perception, we use a competitive learning method that builds a topological structure among unsupervised learning methods. Except for GNG, which is used in this method, other commonly used competitive learning methods include Self-Organizing Map (SOM) [16], Neural Gas with Competitive Hebbian Learning (NG-CHL) [17], Growing Cell Structure (GCS) [18], and so on. SOM is a method that uses high-dimensional feature vectors to construct a low-dimensional visualizable dimensionality. SOM is the most commonly used competitive learning method because it can transform high-dimensional feature vectors into low-dimensional visualizable dimensional reference vectors and is easy to implement. However, SOM retains a fixed topological structure and cannot properly learn the geometric structure of 3D point cloud data. However, in 3D space perception, where the probability density distribution of data and the number of data itself change from moment to moment, NG-CHL is a learning method that does not add or delete nodes, making it difficult to construct topological structures with appropriate granularity. In this context, GCS and GNG are learning algorithms that can add and delete nodes, and are applicable to cases with non-stationary data such as space perception. However, GCS does not remove edges as a process, and has the disadvantage of generating redundant nodes that become dead nodes when learning the data distribution of a point cloud. On the other hand, GNG is an algorithm that can add and remove nodes and edges, and thus can learn and preserve the geometrical feature of the point cloud data appropriately.

The topological structure of GNG can be utilized for clustering, noise reduction, feature extraction, and so on. Many examples of applying GNG to space perception have been done for learning 2D/3D point cloud data [19], [20], [21]. As examples, space perception methods using GNG have been proposed in many fields such as 3D reconstruction and motion recognition [22], [23], [24], [25], [26], [27]. In particular, in the research applied to space perception of mobile robots, there are examples of path planning using 2D maps and 3D space perception. For 2D path planning, a method has been proposed in which a 2D occupied grid map is learned by GNG and utilized for path planning by constructing the topological structure [28], [29]. These methods are one of the methodologies based on the global planner, and it is difficult to deal with the case where a dynamic object enters the map in real time. Therefore, a local planner is also necessary for learning such 2D environmental maps in GNGs. Refs. [30] and [31] have applied GNG to 3D point cloud data for real-time recognition. Ref. [30] proposed GNG-DD, which learns the topological structure of nodes that are dense for the object of interest and sparse for other objects. Ref. [31] proposed Add-if-Silent Rule-Based GNG with Amount of Movement, which applies the Add-if-Silent Rule to GNG as an extension of GNG-DD. This methodology is based on a rule that adds a node at a point for data that does not fire by setting a threshold, instead of only adding a node based on the totalization error. Ref. [32] also proposed a clustering method of feature vectors using GNG for detecting the traversablity perception of a multi-legged robot. This method consists of normal vectors and shape features as feature vectors by performing 3D point cloud analysis based on Principal Component Analysis (PCA) as in the present study. However, these methods focus on perception and global planning, and to

the authors' knowledge, there are no studies that focus on local planners, which can make appropriate path changes in dynamic environments. Therefore, this paper focuses on GNG based 3D space perception and the local planning method utilizing the topological structure of GNG.

B. GROWING NEURAL GAS WITH DIFFERENT TOPOLOGIES

In order to perceive the traversability of a road surface, it is necessary to combine not only a single feature but also multiple features to make a decision. The authors have proposed GNG-DT [13] as a GNG method to achieve such clustering. In this study, we will introduce the algorithm of GNG-DT, since the methodology is developed based on a real-time perception system using GNG-DT. To explain the learning algorithm, we define the main variables used in GNG-DT. First, the set of attributes in this study is defined as $S = \{Position(pos), Normalvector(nor), Traversability(tra), Passability(pass)\}$, and the input vector and reference vector are defined as $\mathbf{v} = \{\mathbf{v}^{pos}\}$, and $\mathbf{h}_i = \{\mathbf{h}_i^{pos}, \mathbf{h}_i^{nor}\}$, respectively. Next, we define the distance d_i^o between the input vector and the reference vector of the ith node for an attribute o as follows.

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

GNG-DT learns multiple topological structures, and the edge set of the oth property is defined as $C^o = \{c_{1,2}^o, \cdots c_{i,j}^o, \cdots\}$. The detailed learning algorithm is described as follows.

Step 0. Randomly generate reference vectors $\mathbf{h}_1$ and $\mathbf{h}_2$ for two nodes, initializing the coupling relation $c_{1,2}^o = 1$ $(\in S)$ and the age of the edge $g_{1,2} = 0$.

Step 1. Select one input vector $\mathbf{v}$ at random from the dataset.

Step 2. Determine the first winner node s_1 and the second winner node s_2 from the input vector $\mathbf{v}$ as follows,

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

where A is the set of node numbers.

Step 3. Add the squared error between node s_1 and the input vector $\mathbf{v}$ to the integration error E_{s_1} as the following equation.

$$E_{s_1} \leftarrow E_{s_1} + (d_{s_1}^{pos})^2 \tag{3}$$

Step 4. Update the reference vectors of node s_1 and the nodes that have a connection with node s_1 in each attribute, using the following equation. Note that η_1 and η_2 are learning coefficients, and $\eta_1 > \eta_2$.

$$\begin{aligned} \mathbf{h}_{s1} &\leftarrow \mathbf{h}_{s1} + \eta_1(\mathbf{v} - \mathbf{h}_{s1}) \\ \mathbf{h}_j^o &\leftarrow \mathbf{h}_j^o + \eta_2(\mathbf{v} - \mathbf{h}_j^o) \quad if \quad c_{s1,j}^o = 1 \end{aligned} \tag{4}$$

Step 5. Reset the age of the edge between nodes s_1 and s_2 to 0 and create a new connection of the position information if no connection of the position information exists between nodes s_1 and s_2 $(C_{s_1,s_2}^{pos} = 1)$. The creation of edges for the other attributes $o(\in S^{pos})$ is performed by the following equation.

$$\begin{cases} c_{s_1,s_2}^o = 1 \quad if \ e(\mathbf{h}_{s_1}^o, \mathbf{h}_{s_2}^o) = 1 \\ 0 \quad otherwise \end{cases} \tag{5}$$

where $e(\mathbf{h}_{s_1}^o, \mathbf{h}_{s_2}^o)$ represents the decision function for the attribute, which takes the value 1 if the similarity between attributes is high and 0 otherwise.

Step 6. Increment the age of all the edges that have a connection with the first winner node s_1.

$$g_{s_1,j} \leftarrow g_{s_1,j} + 1 \quad if \ c_{s_1,j}^{pos} = 1 \tag{6}$$

Step 7. Remove edges of all attributes whose age exceeds the set threshold g_{max}. ($c_{s1,s2}^o$=0). As a result, nodes without any connection of the position information are deleted.

Step 8. Perform the following operations for each λ data input.

i. Select the node u with the largest accumulated error.

$$u = \arg\max_{i \in A} E_i \tag{7}$$

ii. The node with the largest accumulated error among the nodes connected to u is f, and insert r between nodes u and f.

$$\mathbf{h}_r = 0.5(\mathbf{h}_u + \mathbf{h}_f) \tag{8}$$

iii. Remove all attribute o($\in S$) edges between nodes u and f $(c_{u,f}^o = 0)$ and add edges of the position information between nodes u, r and r, f $(c_{u,r}^{pos} = 1, c_{r,f}^{pos} = 1)$.

$$c_{i,j}^o = \begin{cases} 1 \quad if \ e(\mathbf{h}_{s_1}^o, \mathbf{h}_{s_2}^o) = 1 \\ 0 \quad otherwise \end{cases} \tag{9}$$

For attributes other than position information $o(\in S^{pos})$, add the edges by considering similarity as in step5.

iv. Update the accumulated error of nodes u and f using the decay rate $\alpha^{dec}(0 \le \alpha \le 1)$ by the following equation.

$$\begin{aligned} E_u &\leftarrow E_u - \alpha^{dec} E_u \\ E_f &\leftarrow E_f - \alpha^{dec} E_f \end{aligned} \tag{10}$$

Step 9. Decay the errors of all nodes by a decay rate $\beta(0 \le \beta \le 1)$.

$$E_i \leftarrow E_i - \beta E_i \quad (\forall i \in A) \tag{11}$$

Step10. If the termination condition is not satisfied, go back to Step 1.

GNG-DT learns topological structures using the above algorithm steps. This paper proposes a local planner that adds and utilizes the topological structures C^{tra} of traversability and C^{pass} of passability as new topological structures.

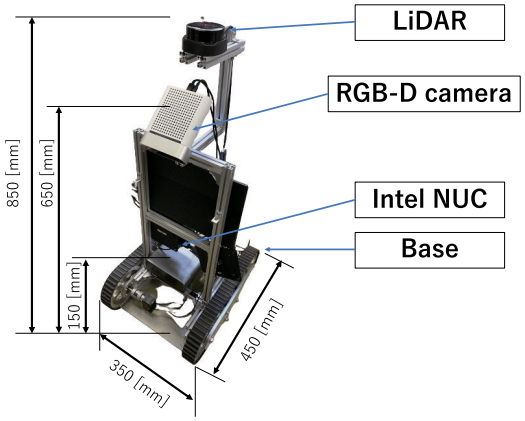


FIGURE 1. Autonomous mobile robot.

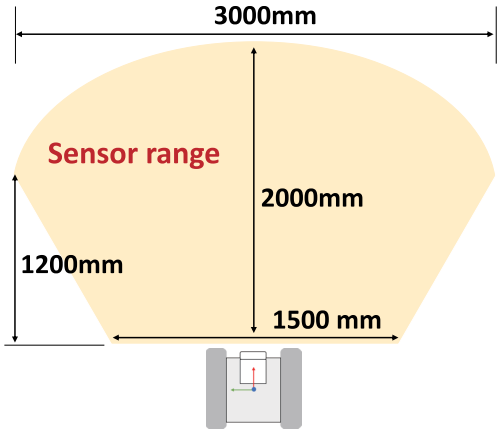


FIGURE 2. Road surface recognition range by RGB-D camera.

III. SYSTEM CONFIGURATION

The autonomous mobile robot used in this study is shown in Fig.1, and the robot specifications are shown in Table 1. The autonomous mobile robot uses Dynamixel XH430-V210-R motors on both wheels. Azure Kinect DK, an RGB-D camera, is used for environment recognition. To detect the road surface, the camera is mounted at a 45° downward angle, 650 mm above the floor. The downward-facing RGB-D camera can acquire information on the road surface immediately in front of the robot, and the measurement range is shown in Fig.2 when the floor surface is perfectly flat. In this research, the 3D point cloud data measured in this range is used as input to learn GNG-DT to perceive the traversability of the road surface.

IV. PASSABILITY PERCEPTION BASED ON GNG-DT

In this section, we propose a perception method for realizing a local planner utilizing the topological structure built by GNG-DT. Specifically, the traversablity perception proposed in Ref.[14] did not take the size of the robot into account, resulting in perceiving nodes that the robot could not actually pass through. In this paper, we propose a passability perception that considers the size of the robot, thereby extending the previous perception method to a method applicable to local planners.

TABLE 1. Specification of autonomous mobile robot.

Width	350 [mm]
Depth	450 [mm]
Height	850 [mm]
Weight	10 [kg]
Wheel diameter	150 [mm]
Maximum output	52 [rev/min]

A. SLOPE ANGLE PERCEPTION

In Step 4 of the GNG-DT learning algorithm, the normal vector is estimated by using a method based on PCA for the first winner node s_1 [33]. Specifically, the normal vectors are calculated using the following equation for the covariance matrix F_{s_1} with the first winner node s_1 and its neighbors as local surface elements.

$$F_{s_1} = \begin{bmatrix} \mathbf{h}_1^{pos} - \mathbf{h}_{s_1}^{pos} i \\ \vdots \\ \mathbf{h}_k^{pos} - \mathbf{h}_{s_1}^{pos} \end{bmatrix}^T \begin{bmatrix} \mathbf{h}_1^{pos} - \mathbf{h}_{s_1}^{pos} \\ \vdots \\ \mathbf{h}_k^{pos} - \mathbf{h}_{s_1}^{pos} \end{bmatrix} \tag{12}$$

Next, eigenvectors and eigenvalues are calculated from the covariance matrix F_{s_1}. The eigenvector with the smallest eigenvalue $\lambda_{s_1}^3$ among the eigenvectors ($\lambda_{s_1}^1 \geq \lambda_{s_1}^2 \geq \lambda_{s_1}^3$) is calculated as the normal vector of the node $\mathbf{h}_{s_1}^{nor}$ is calculated. Using the estimated normal vector of the first winner node s_1 and the current robot posture $\mathbf{p}_t$, the slope angle of the first winner node is calculated as follows.

$$deg_{s_1} = \cos^{-1}\left(\frac{\mathbf{h}_{s1}^{nor} \cdot \mathbf{p}_t}{\|\mathbf{h}_{s1}^{nor}\| \cdot \|\mathbf{p}_t\|}\right) \tag{13}$$

where deg_{s_1} denote the slope angle of the first winner node. Furthermore, if deg^{max} is the maximum slope angle at which the robot can travel, the property of the i-th node with the slopes at which it can travel are determined as follows (p_i^{deg}).

$$p_i^{deg} = \begin{cases} 1 & \text{if } deg_i < deg^{max} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The node attributes of the slope angle will be used as information to calculate the traversability.

B. SURFACE ROUGHNESS PERCEPTION

In this study, the roughness of the road surface is added as another feature for the traversability perception. The surface roughness is calculated using the following equation for the shape feature representing the 3D shape in Ref. [34],

$$h_i^{rou} = \frac{\lambda_i^3}{\lambda_i^1} \tag{15}$$

where λ_i^j represents the eigenvalues calculated during normal vector estimation for the i-th node and j represents the order of magnitude of the eigenvalues. Using these values, the node property for road surface roughness are defined by the following equation,

$$p_i^{rou} = \begin{cases} 1 & \text{if } h_i^{rou} < thv^{rou} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where thv^{rou} represents the threshold of road surface roughness that is traversable, and a node is treated as traversable as long as the threshold is not exceeded.

C. TRAVERSABILITY PERCEPTION

In GNG-DT, each topological structure can be combined. In this study, a topological structure C^{tra} of traversablity is constructed by combining the topological structures of slope angle (p_i^{deg}) and surface roughness (p_i^{rou}) to determine the traversablity of a node, which is then used for local planning. Specifically, the traversablity property of the i-th node is determined as follows (p_i^{tra}),

$$p_i^{tra} = p_i^{deg} \cdot p_i^{rou} \tag{17}$$

where $p^{tra} = 1$ indicates a node that is traversable, and 0 indicates a node that is untraversable. Furthermore, the topological structure representing the traversablity of the i-th node and the j-th node is updated as follows,

$$c_{i,j}^{tra} = \begin{cases} 1 & \text{if } p_i^{tra} = p_j^{tra} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Thus, by taking advantage of the characteristics of GNG-DT and combining clusters related to traversablity, more suitable traversablity clusters can be generated and used for the local planning.

D. PASSABILITY PERCEPTION

The robot can perceive the traversable area by using the above-mentioned perception. However, there are cases in which the area belonging to a plane that is judged to be traversable is actually a narrow path bounded by walls, or where the ceiling is so low that the robot's upper part could collide with it. Therefore, in order to utilize the robot for the local planning, it is necessary to detect areas where the robot can pass through, taking into consideration the robot's body size. Therefore, in order to include passability information that is in accordance with the robot's body size in the traversability clusters, we propose a method to judge whether or not a node in the traversable information is passable, as shown in the following equation,

$$p_i^{pas} = \begin{cases} 1 & if \;\; dis_i < pass^{thv} \\ 0 & otherwise \end{cases} \tag{19}$$

where fis_i is the shortest distance between a travelable node ($p_i^{tra} = 1$) and an element of the set N of untravelable nodes ($p_i^{tra} = 0$), calculated as the following equation.

$$dis_i = \min_{j \in N} d_j^{pos} \tag{20}$$

Specifically, as shown in Fig.3, for a node i, a circular threshold distance is set around the node. If there is a node that cannot be traveled, then node i is judged as an impassable node (that cannot be passed). By performing this process for all possible nodes, a traversability cluster with passable information is created. When p^{pas} is 1, it represents a passable node, and when p^{pas} is 0, it represents an impassable node. A passable node is always a travelable node, but even the traversable node may become the impassable node.

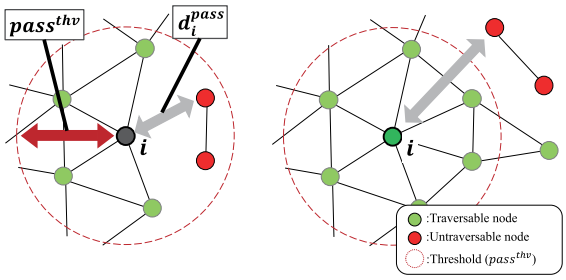


FIGURE 3. Passability node detection.

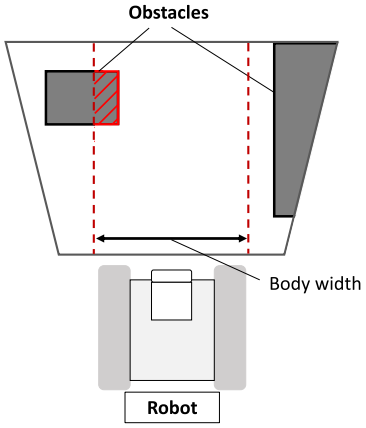


FIGURE 4. Obstacle detection under moving.

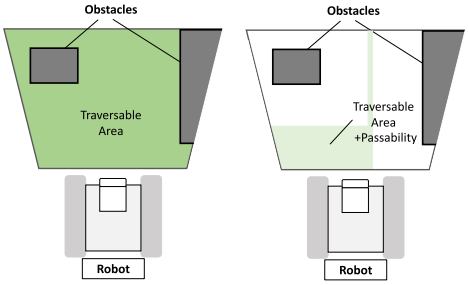


FIGURE 5. Travelable area including passability information.

V. LOCAL PLANNER BASED ON PASSABILITY INFORMATION

In order for a robot to move in a real environment, it must perceive the environment and plan its path using the information it obtains. The local planner plays an important role in this process. It is used to plan a route to a destination while avoiding dynamic obstacles that are not included in the robot's map information. In this study, we propose a local planner that utilizes traversability and passability information perceived by GNG-DT to select appropriate start and target nodes.

A. CONTOUR NODE DETECTION

In this study, we propose a local planner that uses the learned topological structure. The contour information of the

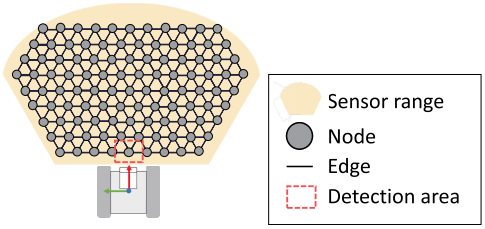


FIGURE 6. Start node detection.

topological structure is very important because the boundary with the unknown environment is a node located at the contour (boundary) of the constructed topological structure. Therefore, in this study, contour nodes are extracted in Step 5 of the GNG-DT learning algorithm, as in the literature. Specifically, we calculate the angle θ_k between the i-th node and the i-th edge of each of its neighbors when the nodes in the neighborhood of the i-th node $\boldsymbol{h}_i^{pos}$ are sorted in counterclockwise order, and perform a threshold judgment as expressed in the following formula to The approximate judgment of the contour node is performed by calculating the angle θ_k of the edge with the i-th node of each neighbor node. Since GNG-DT creates topological structures for multiple attributes, the contour information possessed by a node differs depending on the attribute. When $z_i^o = 1$, the node is a contour node in attribute o.

$$\begin{cases} z_i^o = 1 & if \ \ \theta^{max} > \theta^{thv} \\ z_i^o = 0 & otherwise \end{cases}$$
$$\theta^{max} = \max_k \theta_k \tag{21}$$

B. START NODE DETECTION

Obtain a node that indicates the current position of the robot, which is the starting point for local planning. A conceptual diagram is shown in Fig.6. The light yellow background represents the measurement area of the sensor, and the gray dots and black lines represent the nodes and edges constructed by GNG-DT. A candidate node of the start node is determined by the following formula.

$$s = \arg\min_{i \in A} ||\mathbf{h}_i^{pos}|| \tag{22}$$

If the candidate node s is passable and exists in the area indicated by the red dashed line (Fig.6), the candidate node s is selected as the start node of the local planning. If the start node is successfully selected, the target node described is selected by using the target node detection method described in subsection V-C. If the start node does not exist in the red dashed line area, or if the candidate node s is not passable, the selection of the start node fails and the avoidance action described in subsection V-E is performed.

C. TARGET NODE DETECTION

When the robot searches in the unknown environment, the robot can plan the path only within the current perception. However, it is not possible to plan in detail the path from the current position to the final destination in the unknown

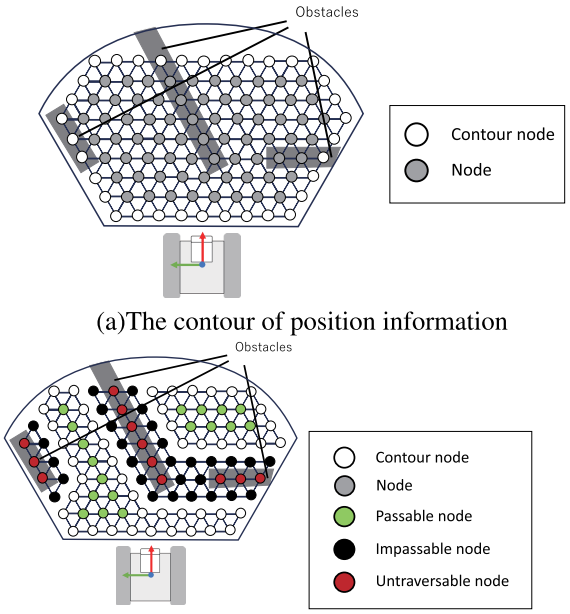


FIGURE 7. The state of obstacle avoidance.

environment. In such an environment, subgoals are important. By setting the subgoal in the current perceptual range, the robot can expand its perceptual range during the search to that point, update the path to the final destination. Therefore, from the topological structure learned by GNG-DT, the robot obtain a node that indicates the target position, which is the end point of path generation by the local planner. In this study, the contour information of the node described above is used to detect the subgoal. Fig.7 shows the contour node according to the position information and passability information.

From the contour nodes in the topological structures, the robot can detect a set D of candidate nodes that contain subgoal nodes as the following equation.

$$D := \{i | z_i^{pos} = 1 \cap z_i^{pass} = 1\} \tag{23}$$

The points and lines in Fig.8 represent the topological structure constructed by GNG-DT. Green dots represent passable nodes, red dots represent impassable nodes, and white dots represent contour nodes. The white dots highlighted on a yellow background represent candidate target nodes; because the topological structure in GNG-DT can be clustered, candidate nodes can only be detected within the cluster where the acquired start node is located. Therefore, the robot does not select a target node that requires the robot to cross an impassable area. When the robot is given the target position outside the current perception range, the candidate node closest to the target $\mathbf{v}^d$ is selected from the set of candidate nodes D as the following equation,

$$g = \arg\min_{i \in D} ||\mathbf{v}^d - \mathbf{h}_i^{pos}|| \tag{24}$$

where g is the target node number.

When the target position is within the perception range of the robot, there are four cases as shown in Fig.9. In Case 1, the

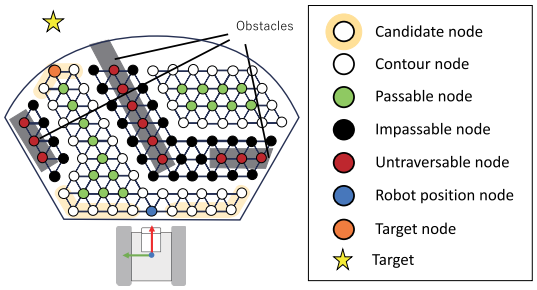


FIGURE 8. Target node detection.

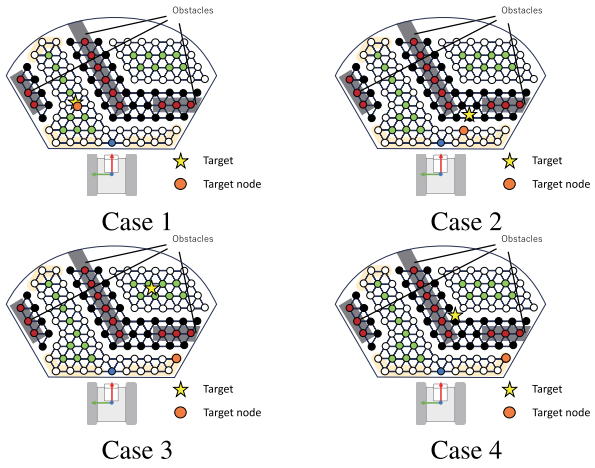


FIGURE 9. Cases of target node detection when the target position is in the perceived range.

target position is located in the same cluster as the start node s; in Case 2, the target position is located in an impassable area and is completely unreachable. In these cases, the closest node to the target position is found from the set of passable nodes with the same cluster, P, as shown in the equation below.

$$g = \arg\min_{i \in P} ||\mathbf{v}^d - \mathbf{h}_i^{pos}|| \tag{25}$$

Case 3 is a case where the target location is in a passable region and can be reached, but there is no path to it. Case 4 is a case similar to Case 2, where the target location is in an impassable area, but the location is closer to other passable clusters than the cluster to which the start node belongs. In these cases, as in the case where the target position is outside the current recognition range, the target node is searched for as in Eq. (24).

D. LOCAL PLANNING

The local planner applies the Dijkstra method to the topological structure learned in GNG-DT. The topological structure used is the traversability cluster with the passability information shown on the right side of Fig.5. Although the traversability cluster detects the road surfaces on which the robot can travel, which does not take into account the robot's body size. If a path is planned within only the traversbaility cluster, the path will include areas that are actually impassable. On the other hand, by using the travel possibility clusters including the passability information for local planning, the path can be planned that does not contact any obstacles. The specific algorithm of Dijkstra's method is described below.

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

Step 1. Set the cost $d_{i,j}^{cos}$ between all two vertices connected by C^{pass}. In the following equation (26), $d_{i,j}^{pos}$ is the distance between nodes i and j,

$$d_{i,j}^{cos} = \begin{cases} \alpha(z_i^{pass} + z_j^{pass}) + d_{i,j} & \text{if } c_{i,j}^{pass} = 1 \\ \infty & \text{otherwise} \end{cases} \tag{26}$$

where z_i^{pass} represents the contour information in the passability information of the i-th node. The α is a weight coefficient, which is set for the reason described below.

Step 2. Among the nodes with undetermined minimum cost, select the node i with the lowest cost to determine the minimum cost.

Step 3. For a node j that has an edge with node i determined in Step 2 but is not yet determined, if the sum of the cost with node i is less than the previous cost, the cost is updated.

$$d_{s,j}^{cos} \leftarrow d_{s,i}^{cos} + d_{i,j}^{cos} \quad if \quad d_{s,j}^{cos} > d_{s,i}^{cos} + d_{i,j}^{cos} \tag{27}$$

Step.4 Step. 2 and Step. 3 are performed until the minimum cost to all nodes is determined, and the minimum cost from node s to each node is calculated.

In this way, the robot can plan the path within the passable area. However, when the robot passes through the passable area, the start node detection may be unstable because the passability information of candidate nodes is not stable and frequently switches between passable and impassable states. The topological structure learned by the GNG-DT is not fixed at a certain position but changes dynamically. Since the passability information is based on the distance between nodes, it is very sensitive to this effect, which is more pronounced for contour nodes. For this reason, in Eq. (26), weight is added to the edges of the contour nodes to avoid using the contour nodes for the path.

E. AVOIDANCE BEHAVIOR

In order to perform local planning, it is essential to determine the start and target nodes. Therefore, if the nodes cannot be detected, the local planner fails. For the robot to continue the search, not only avoidance but also the direction of avoidance is important. Therefore, in this study, the topological structure is used to determine the direction of the robot's avoidance. Specifically, when the start node detection fails, the robot is given an input to avoid the direction in which the nearest passable cluster exists, as shown in Fig.10. In this example, the robot performs avoidance by turning counterclockwise.

The above avoidance algorithm cannot handle situations where there are no passable areas near the robot, such as when obstacles are scattered around. In such a case, the robot performs an avoidance behavior using passability information. First, as shown in Fig.11(a), a detection area is

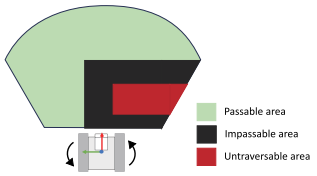
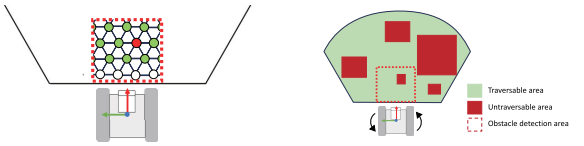


FIGURE 10. An example of avoidance using passability information.



(a) Detection area (b) Avoidance behavior

FIGURE 11. Avoidance behavior using only traversability information.

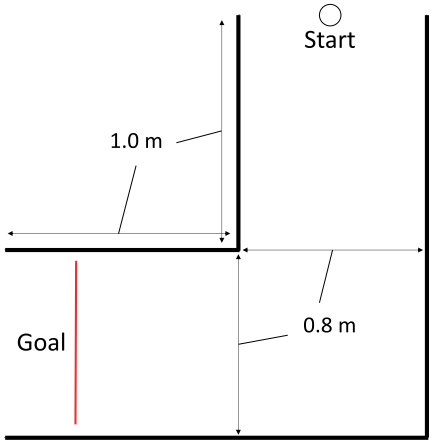


FIGURE 12. Experimental environment 1.

set in front of the robot. If an untraversability node is detected within the area, indicated in red, the robot moves to avoidance behavior. The direction of avoidance is the direction in which there are many traversability nodes, which in this example is counterclockwise. (For comparison, the experiment using only the traversability information used only this algorithm to perform avoidance.)

VI. EXPERIMENTAL RESULTS

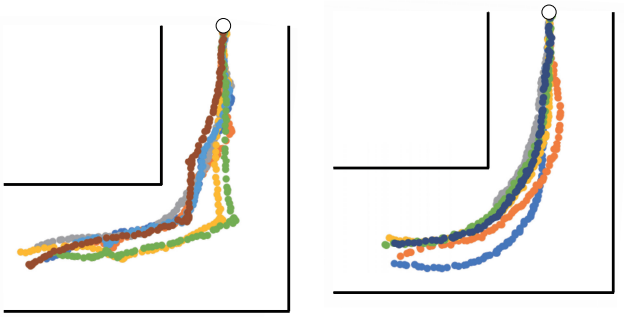
A. VERIFICATION OF EFFECTIVENESS IN L-SHAPED CORRIDOR

To verify the effectiveness of the proposed method, eight runs were conducted in the environment shown in Fig.12. In this experiment, a comparison was made between the local planner using the proposed method with passability information and the local planner using only the traversability information. The parameters used in this experiment are shown in Table 2, and the parameter $pass^{thv}$ of the passability information was set to 0.55m. The robot starts running from the start position shown in Fig.12 and runs through the environment to the goal line using only the local planner. The width between the walls is 0.8m.

Fig.13 shows (a) the results of all trajectories using only the traversability information and (b) the results of all trajectories including the passability information.

TABLE 2. Experimental parameter setting.

	Parameters
g_{max}	88
η_1	0.05
η_2	0.0005
α^{dec}	0.5
β	0.005
λ	200
θ^{thv}	135.0 [deg]
deg^{max}	20.0 [deg]
α^{thv}	0.1
thv^{rou}	5.0×10^{-6}
$pass^{thv}$	0.55 [m]



(a) Traversability (b) Traversability and Passability

FIGURE 13. Experimental results of travel trajectory.

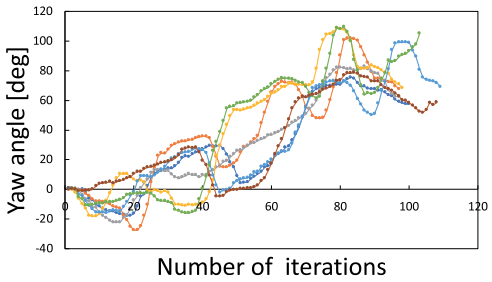


FIGURE 14. Experimental results of yaw angle transition (Traversability).

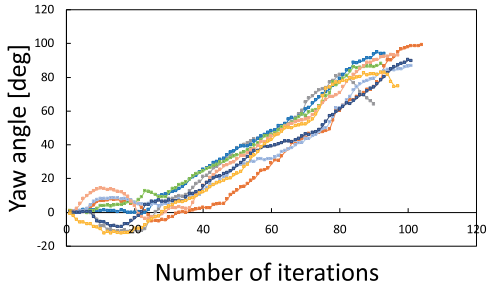


FIGURE 15. Experimental result of yaw angle transition (traversability and passability).

TABLE 3. Average and maximum of processing time.

	Average [ms]	Maximum [ms]
Traversability only	43.49	59.73
Proposed method	45.30	56.98

From Table 3, it can be seen that there was no significant difference in the computation processing time for each of the

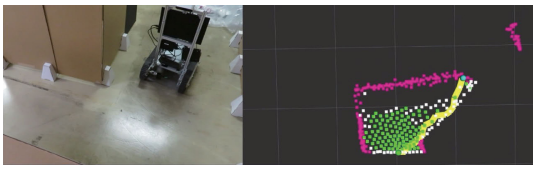


FIGURE 16. The state of movement (Traversability). The left side shows the robot in motion, and the right side shows the learning process in GNG-DT. The green nodes represent traversable node, the black nodes represent untraversable, the white nodes are candidate subgoal nodes, and the red nodes are obstacles. The green circle is the current node and the light blue circle is the destination node, and the yellow line is the result of local planning connecting these two points.

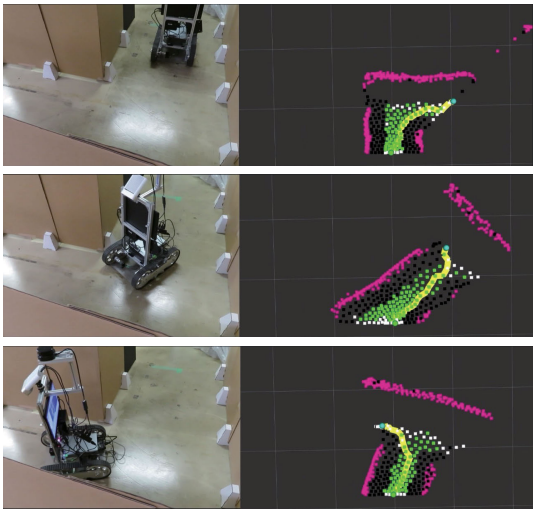


FIGURE 17. The state of robot motion (proposed method).

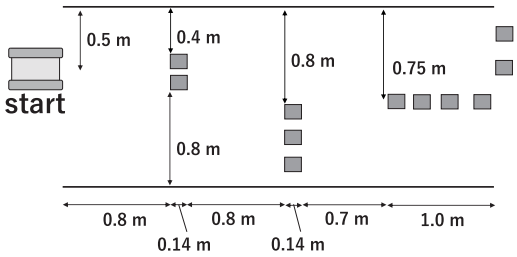


FIGURE 18. Experimental environment 2.

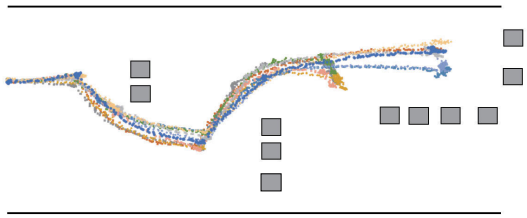


FIGURE 19. Experimental results of travel trajectory (Traversability information only).

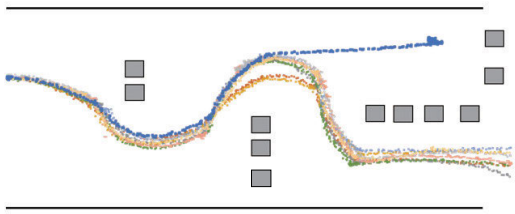


FIGURE 20. Experimental results of proposed method (Traversability and passability information).

runs. However, as shown in Fig.13, the trajectory is angular for the run using only the traversability information.

In our proposed method, a super-clear turn is incorporated to avoidance behavior, and the posture of the robot body changes abruptly during this behavior. Figure 14 shows that the behavior occurs multiple times. The robot basically moves forward by adjusting the output of both wheels based on the results of path planning and avoiding obstacles in advance. Figure 16 shows how the robot behaves when traveling using only the traversability information. As shown in the figure, a path is planned without considering the robot's body size, and it is considered that the robot needs to take actions to avoid obstacles when it actually follows the path. Fig.17 shows the situation in the run with passability information. From the above comparison, the proposed method detects a traversable area including passability information and performs local planning within the area, enabling smoother travel.

B. VERIFICATION OF EFFECTIVENESS IN ENVIRONMENTS INCLUDING DEAD-ENDS

Next, to verify the effectiveness of the proposed method in a more complex environment, we conducted 10 runs each in the environment shown in Fig.18. The parameters of the experiment were the same as in the experiment described above. The robot started from the start position shown in Fig.18, and a run in which the robot was able to escape from the lower right corner of the figure was recorded as a success. The width between the walls is 1.5m, and the minimum width of the passage is 0.7m.

1) RESULT

Figure19 shows the results of all trajectories using only the traversability information, and Fig.20 shows the results of all trajectories including passability information. Figure 21 shows the trajectories of successful and unsuccessful cases in each condition, and Table 4 shows the results of the computation time of the perceptual processing by GNG-DT. Table 5 shows the number of successes and failures for each run. Table 4 shows that there was little difference in the maximum processing time for each run. The average processing time was about 2 or 3 ms faster for the runs using only the traversavility information. The number of successes was 9 out of 10 runs with the passability information, while the number of successes was 0 for the run with only the traversavbility information, and there were 5 contacts with obstacles as shown in Fig. 22.

2) DISCUSSION

In the case of using only the traversability information, the robot performs avoidance action when the robot detects an

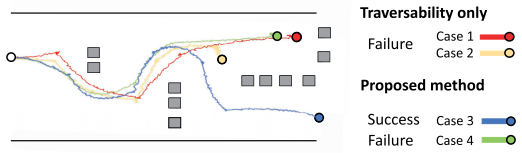


FIGURE 21. Experimental results of travel trajectories.

TABLE 4. Average and maximum of processing time.

	Average [ms]	Maximum [ms]
Case1	38.19	54.71
Case2	38.14	55.48
Case3	41.81	56.34
Case4	40.19	53.70

TABLE 5. Experimental results of success rate.

	Success	Failure	Collision
Traversability only	0	10	5 / 10
Proposed method	9	1	0 / 10

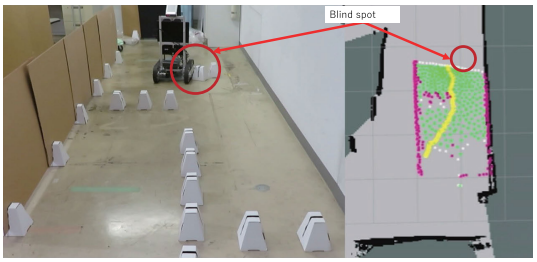


FIGURE 22. An example of collision case in the case of using only the tarversability information.

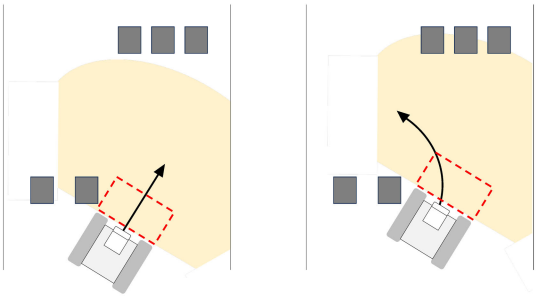


FIGURE 23. The sensor range of the robot and avoidance judgement.

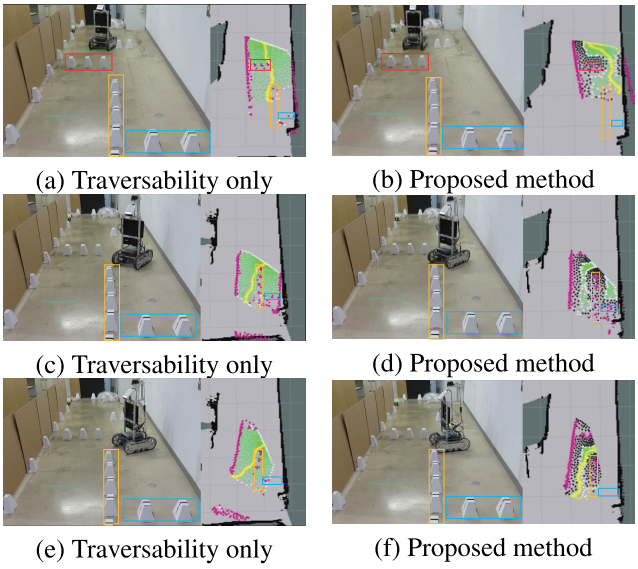


FIGURE 24. Robot motions of avoidance behavior. The detailed experimental video is uploaded as Supplemental items.

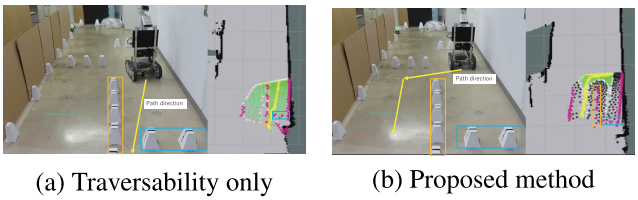


FIGURE 25. Examples of perception result before entering the dead-end.

obstacle within the area indicated by the red frame, as shown in Fig.23. When an obstacle that did not fall within the area in the straight-line state is out of the perceptual range, the robot is given the output to make a large left turn, which is considered to be contact with the side of the robot's body (red circle in Fig.22). Even when the robot did not contact the obstacle, it passed very close to the obstacles in all paths depicted in Fig.24(a), (c), and (d). On the other hand, in the proposed method, the robot passed by the obstacle with a margin as shown in Fig.24 (b), and did not contact the obstacle.

In Case 2 of Fig.21, the robot found a way out of the environment, but failed because it could not pass through a narrow space. The failure is shown in Fig.24 (c) and (e), respectively. In Fig.24 (c), the robot performs the avoidance behavior by turning when it finds an obstacle indicated by the red dots in front of it. However, the local path planned immediately after the avoidance behavior was completed passed right next to the obstacle, as shown in Fig.24 (e). When the robot follows the path, an obstacle appears in front of the robot as before. The robot tries to avoid it and performs avoidance again.

In the successful case of running with passability information shown in Fig.24, the robot fails to acquire the current location node due to the presence of an obstacle in front, as shown in (d), and performs avoidance behavior by turning in the same way. The path planned immediately after the successful detection of the start node and the end of the avoidance behavior is based on the size of the robot, as shown in (f), so that the same obstacle will not cause the avoidance behavior even if the path is followed. Comparing the direction of the robot's body after the avoidance behavior by turning, it can be seen that the robot turns to an angle at which it can pass almost straight through in the case of the one including the passability information.

In Case 1 of Fig.21, the robot failed because it entered a dead-end. Fig.25 (a) and (b) shows the state of each run before the robot entered the dead-end. The target point is set at a position that the robot cannot reach, and the path

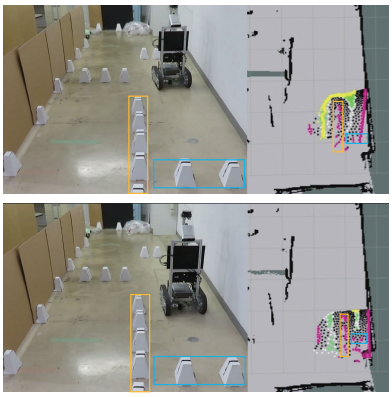


FIGURE 26. Failure example in the proposed method.

is planned through an area between obstacles that the robot cannot pass through. The robot followed this path and entered the dead-end. On the other hand, in 25 (b), the robot does not plan a path through the dead-end, but instead follows an open path on the right side of the path from the robot. This is because the use of passability information allows the robot to perceive the dead-end, and thus is not a candidate for setting a target point. These results indicate that the proposed method can be used for appropriate local planning even when there are dead-end in the environment that are not included in the global information.

Figure 21 shows a case in which a robot entered a dead-end even though it included passability information, as shown in Case 4. Figure 26 shows the situation. The reason is that the path was actually planned to avoid the dead-end, but the output control could not follow it in time. The average and maximum processing times of our local planner in Case 4 are 40.19 [ms] and 53.70 [ms], respectively, according to the Table 4. This exceeds the sampling time of 33 [ms] of the RGB-D camera used in this experiment, and as a result, the robot could not follow the output control and continued to move forward.

VII. CONCLUSION

In this paper, we proposed a 3D space perception method and a local planning method for an autonomous mobile robot in an environment that includes unknown objects. Specifically, we proposed a method of extracting the traversable area by learning the topological structure from 3D point cloud data using GNG-DT and perceiving traversability clusters based on the topological structure for 3D space perception of an autonomous mobile robot. Furthermore, we proposed a method for local planning based on the topological structure, which is based on the passability information.

In the experiments, we compared the cases where only traversability was used and where passability information was used in an actual environment, and showed that the proposed method can plan a route that determines the area that the robot can actually pass through.

However, since the experiment was conducted on a flat surface only, future work will include verification of the effectiveness of the proposed method on slopes and terrain with complex shapes. In addition, the limitations of our proposed local planner include the inability to recover once it falls into a dead end and the high possibility of taking a longer path distance compared to the global planning method. Therefore, integration with the global planner is one of our future tasks.

REFERENCES

[1] Y. Liu, "A laser intensity based autonomous docking approach for mobile robot recharging in unstructured environments," *IEEE Access*, vol. 10, pp. 71165–71176, 2022.
[2] D. Jin, Z. Fang, and J. Zeng, "A robust autonomous following method for mobile robots in dynamic environments," *IEEE Access*, vol. 8, pp. 150311–150325, 2020.
[3] M. Kamezaki, A. Kobayashi, R. Kono, M. Hirayama, and S. Sugano, "Dynamic waypoint navigation: Model-based adaptive trajectory planner for human-symbiotic mobile robots," *IEEE Access*, vol. 10, pp. 81546–81555, 2022.
[4] B. Zhang, R. Sengoku, and H.-O. Lim, "Adaptive motion control for an autonomous mobile robot based on space risk map," *IEEE Access*, vol. 11, pp. 69553–69562, 2023.
[5] J. Pak, J. Kim, Y. Park, and H. I. Son, "Field evaluation of path-planning algorithms for autonomous mobile robot in smart farms," *IEEE Access*, vol. 10, pp. 60253–60266, 2022.
[6] J. Lee, G. Park, I. Cho, K. Kang, D. Pyo, S. Cho, M. Cho, and W. Chung, "ODS-bot: Mobile robot navigation for outdoor delivery services," *IEEE Access*, vol. 10, pp. 107250–107258, 2022.
[7] D. C. Guastella and G. Muscato, "Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review," *Sensors*, vol. 21, no. 1, p. 73, Dec. 2020.
[8] S. Chiodini, L. Torresin, M. Pertile, and S. Debei, "Evaluation of 3D CNN semantic mapping for rover navigation," in *Proc. IEEE 7th Int. Workshop Metrology Aerosp. (MetroAeroSpace)*, Jun. 2020, pp. 32–36.
[9] Y. Toda, T. Matsuno, and M. Minami, "Multilayer batch learning growing neural gas for learning multiscale topologies," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 25, no. 6, pp. 1011–1023, Nov. 2021.
[10] 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.
[11] Y. Toda, X. Li, T. Matsuno, and M. Minami, "Region of interest growing neural gas for real-time point cloud processing," in *Proc. 12th Int. Conf. (ICIRA)*, 2019, pp. 82–91.
[12] B. Fritzke, "A growing neural gas network learns topologies," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 7, 1995, pp. 625–632.
[13] 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.
[14] 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.
[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] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer, 2000.
[17] T. M. Martinetz and K. J. Schulten, "A 'neural-gas' network learns topologies," *Artif. Neural Netw.*, vol. 1, pp. 397–402, Jan. 1991.
[18] B. Fritzke, "Unsupervised clustering with growing cell structures," *Neural Netw.*, vol. 2, pp. 531–536, Jul. 1991.
[19] C. Zhong, B. Zhang, and J. Wang, "Scale-adaptive growing neural network based on distortion error stability and its application in image topological feature extraction," *IEEE Access*, vol. 9, pp. 767–776, 2021.
[20] J. Tünnermann, C. Born, and B. Mertsching, "Saliency from growing neural gas: Learning pre-attentional structures for a flexible attention system," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5296–5307, Nov. 2019.
[21] N. Kubota, T. Narita, and B. Hee 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.

[22] J. Garcia-Rodriguez, M. Cazorla, S. Orts-Escolano, and V. Morell, "Improving 3D keypoint detection from noisy data using growing neural gas," in *Advances in Computational Intelligence*, vol. 7903. Springer, 2013.
[23] 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.
[24] S. Orts-Escolano, J. Garcia-Rodriguez, J. A. Serra-Perez, A. Jimeno-Morenilla, A. Garcia-Garcia, V. Morell, and M. Cazorla, "3D model reconstruction using neural gas accelerated on GPU," *Appl. Soft Comput.*, vol. 32, pp. 87–100, Jul. 2015.
[25] 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.
[26] 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.
[27] D. Fišer, J. Faigl, and M. Kulich, "Growing neural gas efficiently," *Neurocomputing*, vol. 104, pp. 72–82, Mar. 2013.
[28] 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.
[29] C. Hahn, S. Feld, M. Zierl, and C. Linnhoff-Popien, "Dynamic path planning with stable growing neural gas," in *Proc. 11th Int. Conf. Agents Artif. Intell.*, 2019, pp. 138–145.
[30] 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, p. 35, Apr. 2021.
[31] 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.
[32] J. Faigl and P. Miloš, "Incremental traversability assessment learning using growing neural gas algorithm," in *Advances in Self-Organizing Maps, Learning Vector Quantization, Clustering and Data Visualization*. Berlin, Germany: Springer, 2020.
[33] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proc. Conf. Vis.*, 2002, pp. 163–170.
[34] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based scale selection in 3D LiDAR point clouds," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. HAL, 2011, pp. 97–102.

KOKI OZASA received the B.E. degree from the Division of Industrial Innovation Sciences, Graduate School of Natural Science and Technology, Okayama University, Okayama, Japan, in 2023, with a focus on topological clustering. His research interests include soft computing and intelligent robots.

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.

YOSHIMASA NAKAMURA received the B.E. degree from Tokyo Metropolitan University, Tokyo, Japan, in 2015, where he is currently pursuing the Ph.D. degree with the Graduate School, Department of Mechanical Systems Engineering. He is an Associate Senior Researcher with the Robot Technology Group, Tokyo Metropolitan Industrial Technology Research Institute, Tokyo. His research interests include self-localization and navigation for autonomous mobile robots.

TOSHIKI MASUDA received the Ph.D. degree from the University of Electro-Communications, in 2010. He is currently the Section Chief of the Project Planning Section, Planning Department, Tokyo Metropolitan Industrial Technology Research Institute. His research interests include hardware and the safety of autonomous mobile robots.

HIROHIDE KONISHI received the M.E. degree in mechanical engineering from Ibaraki University, Hitachi, Japan, in 2000, and the M.E. degree from the Graduate School of Engineering Management (MOT), Shibaura Institute of Technology, Tokyo, Japan, in 2014. In 2000, he joined NSK Ltd., Tokyo, where he developed industrial direct drive motors. He is involved in technology research and business development of autonomous mobile robots.

TAKAYUKI MATSUNO (Member, IEEE) received 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 has been a Professor with Okayama University, since 2022. His research interests include the intelligent control of industrial manipulators and medical robots for surgical assistance.