

Growing Neural Gas based Traversability Clustering for an Autonomous Robot

Koki Ozasa
Okayama University
Okayama, Japan
pvta6il3@s.okayama-u.ac.jp

Yuichiro Toda
Okayama University
Okayama, Japan
ytoda@okayama-u.ac.jp

Takayuki Matsuno
Okayama University
Okayama, Japan
matsuno@okayama-u.ac.jp

Abstract—One of the most important capabilities of an autonomous robot is to recognize a 3D space of the surrounding environment in real-time from a 3D point cloud measured by a 3D distance sensor. The area in which the robot can travel is limited by a robot embodiment such as a mechanism of the robot. Therefore, the traversability estimation method helps the robot to travel safely and reduces the calculation cost of the path planning. This paper proposes Growing Neural Gas (GNG) based traversability estimation method by utilizing a topological structure learned from the 3D point cloud data. However, the conventional GNG cannot preserve the geometric information of the 3D point cloud if the input vector is composed of the multiple properties. Therefore, this paper apply GNG with Different Topologies (GNG-DT) that learn the multiple topological structures according to the number of properties. This paper proposes a GNG-DT based traversability estimation method by redefining the property of the GNG-DT. We conduct several experiments in both simulation and real environment to verify the effectiveness of our proposed method.

Index Terms—Traversability estimation, Growing Neural Gas, Unsupervised learning

I. INTRODUCTION

Expectations for mobile robots have increased in recent years due to their potential to reduce the heavy burden of work in various fields such as disaster sites, mountains, forests, and more [1]–[3]. Typically, robots in these fields are used for monotonous work via teleoperation systems. However, the teleoperation system requires an operator with operation skills and has problems with time lag, lack of resolution, operation accuracy, and field of view, reducing work efficiency. Therefore, an autonomous mobile robot is required not only to execute monotonous work but also to move to target points and execute tasks. To move to target points, including unknown areas, the robot needs to perceive the surrounding environment in real-time [4]. Furthermore, in environments where the autonomous robot executes tasks, rough terrain may be encountered. Robots must recognize the 3D space of the rough terrain by measuring any sensors installed in the robot [5]. In recent years, various 3D distance measurement sensors such as lidar, stereo vision, and RGB-D cameras have been developed and easily installed in many robots [6]–[8]. However, mobile robots have differences in moving mechanisms, sensors, actuators, and other mounting positions, which determine the traversable area and executable tasks due to the robot embodiment [9]. Therefore, it is crucial for robots operating in rough terrain to estimate the traversability of the rough terrain from the 3D point cloud

data to avoid the risk of operational failure and irrecoverable damage. However, the 3D point cloud is unstructured data, making it difficult to estimate the traversability of the terrain environment directly. This paper focuses on a real-time traversability estimation method from 3D point clouds. Our proposed method is based on Growing Neural Gas (GNG) [10], which can learn the topological structure from unknown point cloud data. GNG can dynamically change the topological structures by adding and removing nodes and edges to preserve the data space of the input vector. Therefore, various GNG-based perceptual systems have been proposed. However, the conventional GNG cannot preserve the 3D environmental space from the 3D point cloud if the input vector is composed of the point cloud and other properties such as RGB-D camera. To solve this problem, Y. Toda et al proposed GNG with Different Topologies (GNG-DT), which builds multiple topological structures of each property and can preserve the 3D environmental space of the point cloud [?]. In this paper, we propose a traversability estimation method based on GNG-DT by utilizing the multiple topological structures. It enables the robot to cluster the 3D environmental space of the traversability in real-time due to the robot's embodiment. To verify the effectiveness of our proposed method, we conduct several experiments, including simulations and real environments.

II. TRAVERSABILITY ESTIMATION FROM 3D POINT CLOUD

In this paper, our proposed method is based on GNG-DT for learning the 3D point cloud since GNG-DT can build the multiple topological structures of each property and preserve the geometric space of the position information [12]. The capability of GNG-DT is useful for estimating traversability from 3D point clouds. Therefore, this section first explains the algorithm of GNG-DT.

A. Growing Neural Gas Different Topologies

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 $S = \{\text{location}(\text{pos}), \text{normal vector}(\text{nor}), \text{traversability}(\text{tra}) \dots\}$, and the input vector and reference vector are defined as $\mathbf{v} = \{\mathbf{v}^{\text{pos}}\}$, and $\mathbf{h}_i = \{\mathbf{h}_i^{\text{pos}}, \mathbf{h}_i^{\text{nor}}, \mathbf{h}_i^{\text{tra}}\}$, respectively. Next, we define the distance d_i^o between the input vector and the reference vector of the i th node for an attribute o as follows.

$$d_i^o = \|\mathbf{v}^o - \mathbf{h}_i^o\| \quad (1)$$

This work was supported by JSPS KAKENHI Grant Number 20K19894.

GNG-DT builds multiple topological structures, and the edge set of the o th property is defined as $C^o = \{c_{1,2}^o, \dots, c_{i,j}^o, \dots\}$. 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. Obtain one input vector \mathbf{v} at random from the measurement data.

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

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

A is the set of node numbers.

Step 3. Add the squared error between node s_1 and the input data \mathbf{v} to the integration error E_{s1} as shown in the following equation.

$$E_{s1} \leftarrow E_{s1} + (d_{s1}^{pos})^2 \quad (3)$$

Step 4. Step4. Update the reference vectors of node s_1 and the nodes that have a joint relationship with node s_1 in each attribute, using the following formula. Note that η_1 and η_2 are learning coefficients, and $\eta_1 \neq \eta_2$.

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

Step 5. Reset the age of the edge between nodes $S1$ and $S2$ to 0 and create a new location edge if no location edge exists between nodes $S1$ and $S2$ ($C_{S1,S2}^{POS} = 1$). The creation of edges for attributes other than location $o(\in S^{pos})$ is performed using the following equation.

$$\begin{cases} c_{s1,s2}^o = 1 & \text{if } g(\mathbf{h}_{s1}^o, \mathbf{h}_{s2}^o) = 1 \\ c_{s1,s2}^o = 0 & \text{otherwise} \end{cases} \quad (5)$$

$g(\mathbf{h}_{s1}^o, \mathbf{h}_{s2}^o)$ represents the decision formula 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 location binding relationship with the node s_1 .

$$g_{s1,j} \leftarrow g_{s1,j} + 1 \quad \text{if } c_{s1,j}^{pos} = 1 \quad (6)$$

Step 7. Delete edges of all attributes whose age exceeds the set threshold g_{max} . ($c_{s1,s2}^o=0$). As a result, nodes that do not have a joint relationship with other nodes in the location information are deleted.

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

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

$$u = \arg \max_{i \in A} E_i \quad (7)$$

ii. The node with the largest integration error among the nodes connected to u is f , and insert r at the midpoint of the edge joining u and f .

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

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

$$\begin{cases} c_{i,j}^o = 1 & \text{if } \|\mathbf{h}_i^o - \mathbf{h}_j^o\| < \tau^o \\ c_{i,j}^o = 0 & \text{otherwise} \end{cases} \quad (9)$$

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

iv. Update the total error of nodes u and f using the decay rate $\alpha(0 \leq \alpha \leq 1)$ by the following formula.

$$\begin{aligned} E_u &\leftarrow E_u - \alpha E_u \\ E_f &\leftarrow E_f - \alpha E_f \end{aligned} \quad (10)$$

v. The integration error of node r is computed as the average of the integration errors of nodes u and f .

$$\begin{aligned} E_u &\leftarrow E_u - \alpha E_u \\ E_f &\leftarrow E_f - \alpha E_f \end{aligned} \quad (11)$$

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

$$E_i \leftarrow E_i - \beta E_i \quad (\forall i \in A) \quad (12)$$

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

Figure 1 shows the whole procedure of GNG-DT. The GNG-DT algorithm involves building multiple topological structures within the learning framework (Steps 4 and 5). To preserve the geometric space of position information, the distance metric of the position information is utilized in the winner node selection and accumulated error calculation (Steps 2 and 3).

B. Feature extraction from topological structure

To estimate the traversability from a 3D point cloud, the normal vector is one of the most important features since it allows the shape of the terrain surface to be estimated [13], [14]. The normal vector of a point p_i is generally estimated by using a local surface element (Fig. 2). In GNG-based normal vector estimation, the topological structure can be utilized for constructing the local surface. The normal vector \mathbf{h}_i^{nor} of the i th node is estimated using the following procedure. First, a covariance matrix C_i is calculated using the following equation.

$$C_i = \sum_{j=1}^k (\mathbf{h}_j^{pos} - \mathbf{o}_i)^T (\mathbf{h}_j^{pos} - \mathbf{o}_i) \quad (13)$$

$$\mathbf{o}_i = \frac{\mathbf{h}_i^{pos} + \sum_{j=1}^k w_{i,j} \cdot \mathbf{h}_j^{pos}}{1 + \sum_{j=1}^k w_{i,j}} \quad (14)$$

$$w_{i,j} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\mu^2}\right) \quad (15)$$

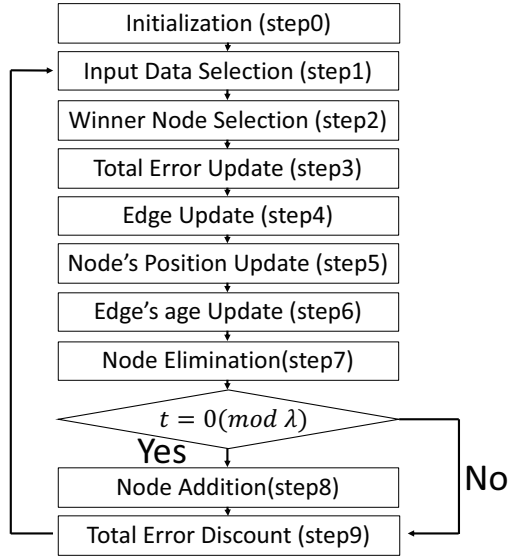


Fig. 1. GNG-DT flowchart.

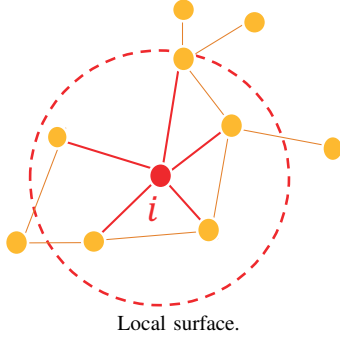


Fig. 2. An example of normal vectors extraction using a topological map.

After calculating the covariance matrix C_i , eigenvector \mathbf{v} and eigenvalue λ of the covariance matrix are calculated, whose eigenvector with the minimum eigenvalue is selected as the normal vector \mathbf{h}_i^{nor} . In addition, the curvature r of the i th node can be calculated by using the following equation,

$$r = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \quad (16)$$

where λ_1 , λ_2 , and λ_3 are the eigenvalues in ascending order ($\lambda_1 \leq \lambda_2 \leq \lambda_3$).

C. Traversability estimation

The traversability of a robot moving in a terrain environment varies depending on its embodiment, such as its actuator, mechanism, and shape. This paper adds the property of traversability to the property set of GNG-DT and builds the topological structure of traversability to enable real-time traversability estimation. Specifically, the traversability of the robot is estimated based on the slope angle of the terrain surface. In particular, the traversability h_i^{tra} of the i th node is calculated as follows,

$$h_i^{mob} = \begin{cases} 1 & \text{if } deg_i < deg^{max} \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

$$(18)$$

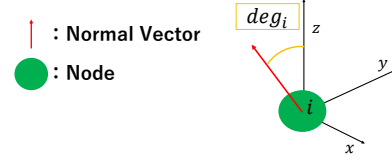


Fig. 3. The state of angel calculation.

TABLE I
EXPERIMENTAL PARAMETER SETTING

Parameters	
g^{max}	88
η_1	0.005
η_2	0.00005
β	0.005
λ	400
deg^{max}	20.0 [deg]

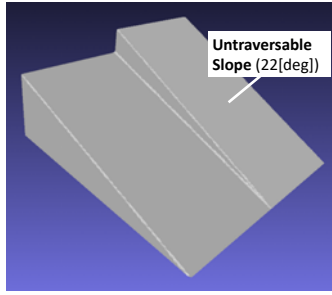
where, deg_i indicates the slope angle between the unit vector of the z axis and the normal vector \mathbf{h}_i^{nor} (Fig. 3) and deg^{max} is the maximum slope angle of the traversable. The traversability is estimated in Step 5, and the robot simultaneously learn and cluster the 3D terrain environment in real-time.

III. EXPERIMENTAL RESULTS

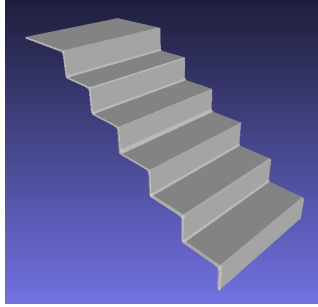
This section presents several experiments that verify our proposed method. The datasets used in these experiments consist of simulation environments and real environments measured by Azure Kinect DK. For all experiments, the experimental parameters of GNG-DT are shown in Table I. The processor of the PC used for the experiments is Intel(R) Core(TM) i7-1165G7 @ 2.80GHz.

A. Simulation environmental data

Figure 4 shows the experimental datasets, and the 3D point clouds were generated from these STL files. In Fig. (a), the slope angles of the left and right sides are 10 [deg] and 22 [deg], respectively. Figure 5 shows examples of the learning results using our proposed methods. In Fig. 5, the blue and red nodes indicate the traversable nodes and the untraversable nodes, respectively. In the slope dataset, the left slope with an angle of 10[deg] was detected as the traversable area, while the right slope with an angle of 22 [deg] was detected as the untraversable area. In the stairs dataset, all of the treads could be detected as the traversable area while all of the risers could be detected as the untraversable area. Therefore, our proposed method could appropriately learn the traversal topological structure (C^{tra}) from the 3D point cloud data, which enables the robot to perceive the traversability in the unknown rough terrain environment. In addition, Fig. 6 shows the transition results of the number of nodes and the calculation time. In both results, the maximum calculation times were less than 28[msec]. Therefore, our proposed method can detect the traversable area from the 3D point cloud in real-time.

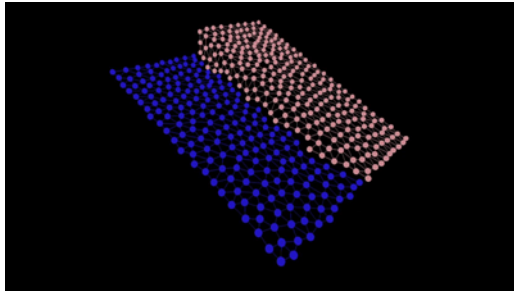


(a) Slope

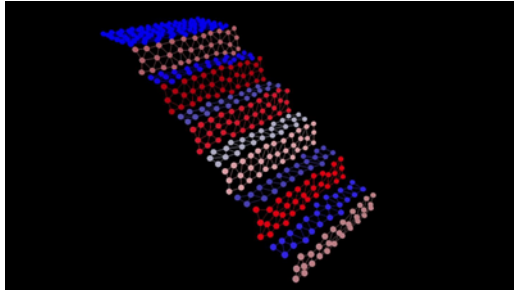


(b) Stair

Fig. 4. STL file.



(a) Cluster of traversability (slope)

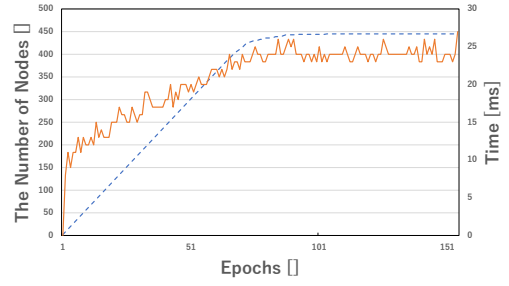


(b) Cluster of traversability (stair)

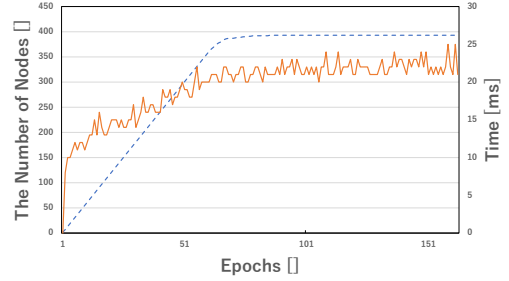
Fig. 5. Learning examples of our proposed method in simulation environments (threshold 20[deg]).

B. Real environmental data

Next, the experiments in the real environment were conducted. Figure 7 shows experimental environments measured by Azure Kinect DK, the slope angles of left and right sides in Fig. 7 (a) set are 10 [deg] and 22 [deg], respectively. The step height of the stair is 20[cm] in Fig. 7 (b). The parameters of GNG-DT were the same as previous experiment. Figure 8 shows examples of the learning results and the blue and red nodes indicate the traversable and untraversable nodes, respectively. In Fig. 8 (a), the slope whose angle is 10 [deg] was detected as the traversable area while the slope whose angle



(a) slope



(b) stairs

Fig. 6. Experimental results of the number of nodes and calculation time in the simulation environments. The blue dot line and orange line indicate the number of nodes and the calculation time, respectively

is 22 [deg] was detected as the untraversable area, which is almost the same result of the simulation environment. In Fig. 8 (b), the learning result was different from the simulation result of the stair since the 3D point cloud of the stair did not include the risers of the stair. However, our proposed method could detect the floor plane and the risers as the traversable and untraversable area, respectively. Therefore, our proposed method can apply to the real environment. Figure 9 shows the transition of the number of nodes and the calculation time. The maximum calculation times of both results were less than 18 [msec] and the sampling time of the Azure Kinect DK is 33 [msec]. Therefore, our proposed method realize the real-time traversability estimation method from the 3D point cloud in the real environment.

C. Dynamic environmental data

The experiment using the dynamic data measured by Azure Kinect DK was conducted. In this experiment, the dynamic data is gathered by moving the Azure Kinect DK through a distance of one meter (Fig. 10). Examples of the learning result are shown in Fig. 11 and the transitions of the number of nodes are shown in Fig. 12. In Fig. 12, (a), (b), and (c) correspond to Fig. 11. After starting to move the Azure Kinect DK, the calculation time increased while the number of nodes decreased since the data distributions measured by Azure Kinect DK were drastically changed, and topological structure of GNG-DT is not stable. However, the maximum calculation time was less than 25[msec] and our proposed method could estimate the traversable area correctly from Fig. 11. From these results, our proposed method can cluster the 3D point cloud from the viewpoint of traversability in real-time.

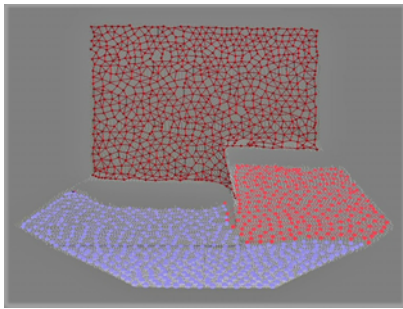


(a) Slope

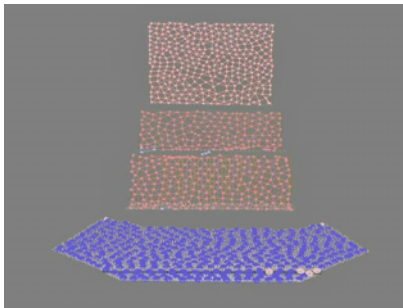


(b) Stair

Fig. 7. Experimental environment measured by Azure Kienct DK.



(a) Slope

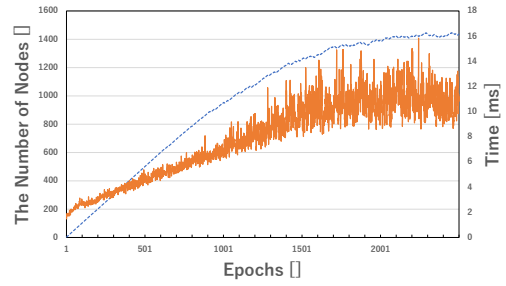


(b) Stairs

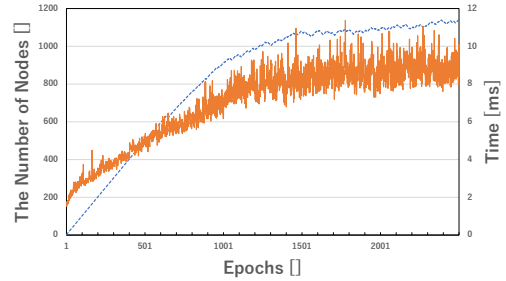
Fig. 8. Experiment result of cluster of traversability.

IV. CONCLUSION

In this paper, we proposed a GNG-DT based learning method to build topological structures from 3D point clouds and a traversability estimation method utilizing these structures, in order to realize autonomous robots in rough terrain environments. We conducted several experiments in simulation and real environments, demonstrating real-time traversability estimation. However, in some real environments, the sensor could not measure the 3D point cloud of the entire area due to lighting conditions. Additionally, in experiments using dynamic data distribution, the number of nodes decreased and the topological structure became unstable. Therefore, one of our future works is to develop



(a) Slope



(b) Stairs

Fig. 9. Experimental results of the number of nodes and calculation time in the real environments. The blue dot line and orange line indicate the number of nodes and the calculation time, respectively

a stable learning method for dynamic data distribution by modifying a batch learning based GNG method proposed in Ref. [15].

Furthermore, our current method only uses slope angle to estimate robot traversability. To apply this method to real rough terrain environments, it is necessary to consider the road surface's unevenness and condition. Therefore, we plan to redefine the robot's detailed traversability based on its embodiment.

REFERENCES

- [1] Liu, M. Robotic Online Path Planning on Point Cloud. *IEEE Trans. Cybern.* **2016**, 46, 1217–1228.
- [2] Seraji, Homayoun. "New traversability indices and traversability grid for integrated sensor/map-based navigation." *Journal of Robotic Systems* 20.3 (2003): 121-134.
- [3] Visser, A., Schmits, T., Roebert, S., & De Hoog, J. (2008, July). Amsterdam Oxford Joint Rescue Forces-Team Description Paper-Virtual Robot competition-Rescue Simulation League-RoboCup 2008. In *Proceedings CD of the 12th RoboCup Symposium*.
- [4] Lu, Feng, and Evangelos Milios. "Robot pose estimation in unknown environments by matching 2d range scans." *Journal of Intelligent and Robotic systems* 18.3 (1997): 249-275.
- [5] Raibert, Marc, et al. "Bigdog, the rough-terrain quadruped robot." *IFAC Proceedings Volumes* 41.2 (2008): 10822-10825.
- [6] Douillard, Bertrand, et al. "On the segmentation of 3D LIDAR point clouds." *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.
- [7] Faugeras, Olivier, and Renaud Keriven. "Complete dense stereovision using level set methods." *European conference on computer vision*. Springer, Berlin, Heidelberg, 1998.
- [8] Kerl, Christian, Jürgen Sturm, and Daniel Cremers. "Dense visual SLAM for RGB-D cameras." *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013.
- [9] A.A.Saputra, W.H.Chin, Y.Toda, N.Takesue, and N.Kubota, "Dynamic density topological structure generation for real-time ladder affordance detection", *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS)*, pp.3439-3444, (2019)
- [10] B. Fritzke, "A Growing Neural Gas Network Learns Topologies.", *Advances in Neural Information Processing Systems 7*. MIT Press, (1995), pp.625–632.

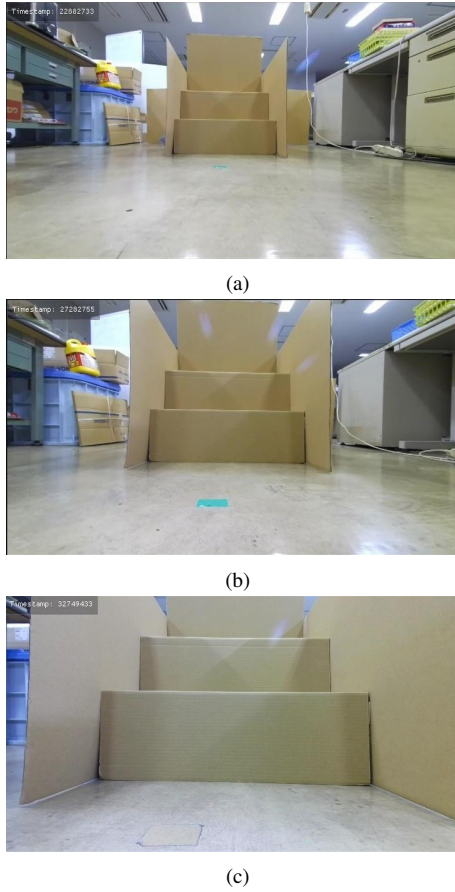


Fig. 10. Experimental environment in the dynamic environmental data.

- [11] Y. Toda, A. Wada, H. Miyase, K. Ozasa, T. Matsuno, M. Minami, "Growing Neural Gas with Different Topologies for 3D Space Perception," *Applied Sciences*, Vol. 12, No. 3, 1705, (2022).
- [12] Y. Toda, H. Miyase, K. Naogao, Q. Li, T. Matsuno, M. Minami, "Growing Neural Gas based Space Perception for Semi-autonomous Teleoperation System," *27th International Symposium on Artificial Life and Robotics*, pp.1338-1343, (2022).
- [13] D.Viejo, J.Garcia, M.Cazorla, D.Gill, M.Johnsson, "Using GNG to improve 3D feature extraction - Application to 6DoF egomotion," *Neural Networks*, Vo.32, pp.138-146, (2012)
- [14] Mark Pauly, Markus Gross, Leif P. Kobbelt, ETH Zürich, ETH Zurich, RWTH Aachen, "Efficient simplification of point-sampled surfaces", *Proceedings of the 2002 IEEE International Conference on Visualization*, (2002).
- [15] Toda, Y.; Matsuno, T.; Minami, M. Multilayer Batch Learning Growing Neural Gas for Learning Multiscale Topologies. *J. Adv. Comput. Intell. Intell. Inform.* **2021**, 25, 6, 1011–1023.

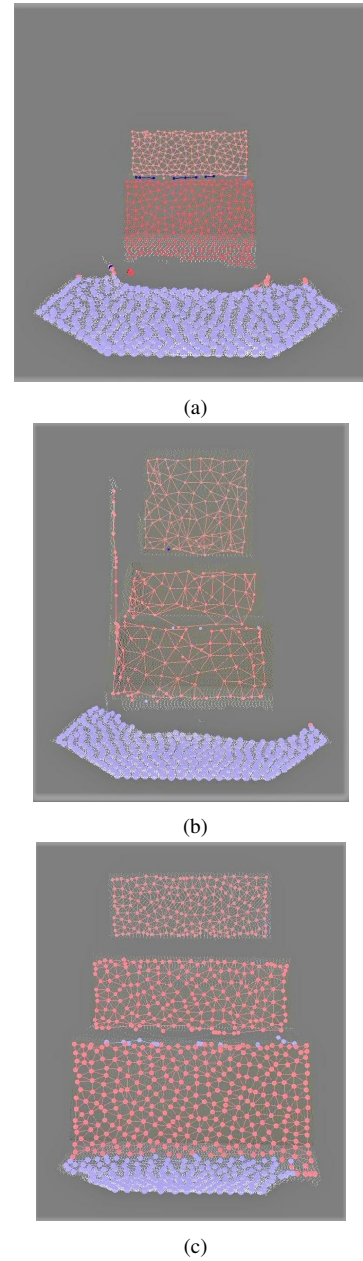


Fig. 11. Examples of the learning result in the dynamic environmental data.

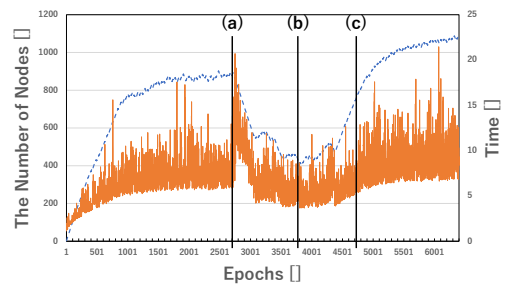


Fig. 12. Experimental results of the number of nodes and calculation time in the dynamic environmental data. The blue dot line and orange line indicate the number of nodes and the calculation time, respectively