Multi-scale Batch-Learning Growing Neural Gas for Topological Feature Extraction in Navigation of Mobility Support Robots

Mutsumi Iwasa[*1], Naoyuki Kubota [*2], and Yuichiro Toda [*3]

[*1] Tokyo Metropolitan University, Hino, Tokyo, Japan
E-mail: m-iwasa@tmu.ac.jp
[*2] Tokyo Metropolitan University, Hino, Tokyo, Japan
E-mail: kubota@tmu.ac.jp
[*3] Okayama University, Okayama, Japan
E-mail: ytoda@okayama-u.ac.jp

Abstract. Recently, the concept of digital twin is applied to various research topics. The aim of digital twin is to simulate and analyze the real world in the cyber space. In order to simulate a real-world phenomenon, we often have to extract features and structures based on graph theory and topology. The methodology of growing neural gas (GNG) is useful to extract topological features hidden in big data. In this paper, we propose a method of multi-scale batch learning (MS-BL) to the realize stable learning of GNG. Next, we apply the proposed method to the topological feature extraction in navigation tasks of mobility support robots. Finally, we show experimental results of the proposed method and discuss the effectiveness of the proposed method.

Keywords: Topological Mapping, Growing Neural Gas, Multi-scale Batch Learning, Mobility Support Robots

1. INTRODUCTION

Recently, various approaches to digital transformation [1], cyber-physical systems [2], and digital twin [3] have been proposed and discussed based on the integration of information, intelligence, communication, and robot technologies. The main focus of digital transformation is to realize all areas of optimization from inputs to outputs after the digitization from analogue data to digital data and the digitalization of automation or business processes. The essence of these approaches is to realize super real-time measurement, monitoring, simulation, prediction, search, adaptation, and control integrated mutually from micro-, meso-, and macro-scopic points of view. Especially, the feature extraction from big data is important to realize super real-time information processing. For example, the features extracted in the lower layer are used for the inference and prediction in the higher level. This is called a bottom-up construction. On the other hand, the inference and prediction results are used as the constraints or order parameters for the information processing in the lower layer. This is called a top-down constraint. Furthermore, this kind of information processing cycle is called micro-macro loop based on bottom-up construction and top-down constraint.

The aim of digital twin is to simulate and analyze the real world in the cyber space and this aim is almost similar to that of cyber-physical system. In order to simulate a real-world phenomenon, we often have to extract features and structures based on graph theory and topology. For example, topological mapping methods are used for 3D modeling available for accurate physics simulation from the microscopic point of view. Graph-based methods are used for knowledge representation available for huge-scale rule-based inference from the macroscopic point of view. Furthermore, we can build a topological model and knowledge according to a mesoscopic modeling and simulation approach to integrate microscopic models and macroscopic knowledge, called *Topological Twin*. The aim of topological twin is to (1) extract topological structures hidden implicitly in the real world, (2) reproduce them explicitly in the cyber world, and (3) Simulate and analyze the real world in the cyber world. Furthermore, we must deal with social or mental world in addition to cyber-physical systems. Figure 1 shows topological twin in cyber-physical-social systems. In this paper, we focus on the topological mapping between the macroscopic and mesoscopic levels.

Various types of growing neural gas (GNG) have been proposed to extract topological structures from data [4-7]. Originally GNG was proposed by Fritzke as one of unsupervised learning methods such as self-organized map (SOM) [8], neural gas (NG)[6], growing cell structures (GCS)[7]. Basically, these methods use the competitive learning. The number of nodes and the topological structure of the network in SOM are designed beforehand. In NG, the number of nodes is fixed beforehand, but the topological structure is updated according to the distribution of a sample data. On the other hand, GCS and GNG can dynamically change the topological structure based on the adjacent relation (edge) referring to the ignition frequency of the adjacent node according to the error index. However, while GCS does not delete nodes and edges, GNG can delete nodes and edges based on the concept of age. These methods have been successfully applied to various types of real-world problems [9-12], but we still have problems on the difficulty of setting hyper parameters used in growing neural network.

We have proposed various types of GNG to improve the learning performance. In order to improve the convergence property of GNG, we proposed a method of batch-learning GNG (BL-GNG) [13] and showed the stable optimization of clustering and topological mapping by BL-GNG in the research of intelligent robotics [14]. However, original BL-GNG adds a node to the network after the sampling of all data. Therefore, we propose a method of multi-scale BL-GNG in order to improve the learning speed.

This paper is organized as follows. Section II explains a simulation example of tracking problems of humans and robots in a public area. Section III proposes a method of multi-scale batch-learning GNG (MS-BL-GNG). First, we explain a learning method of standard GNG, and discuss the setting of hyper parameters through preliminary simulation results. Next, we explain the basic idea of multi-scale batch-learning to enhance the learning property of GNG. Section IV shows numerical simulation results and discuss the convergence property of MS-BL-GNG through the numerical comparison by different settings of hyper-parameters. Finally, Section V discuss the essence of the proposed method and discusses the future direction of this research.

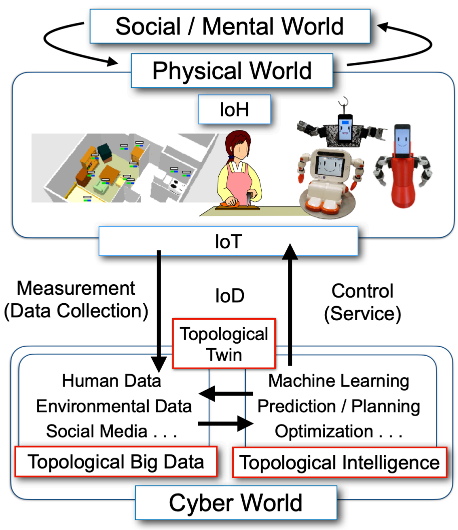
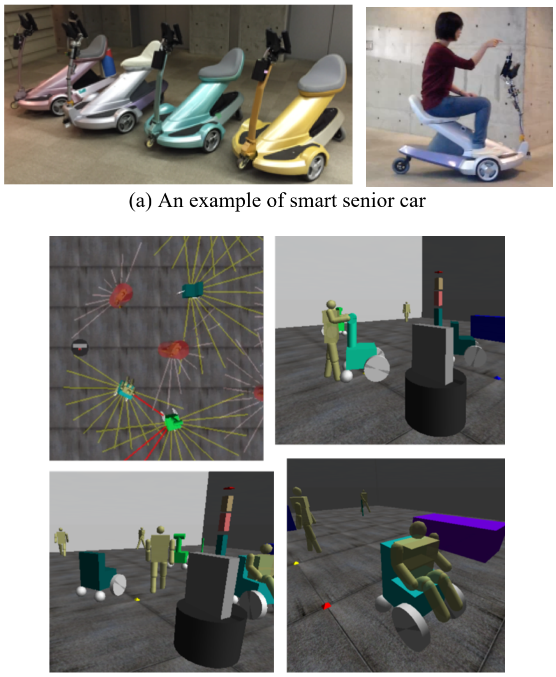


Fig. 1. Topological twin in cyber-physical-social systems

2. Mobility Support Robots

Recently, various types of robot partners have been developed to realize human-robot interaction [15-17]. For example, communication robots are used to realize information support service for elderly care, child nursing and others. Mobile robots are used for delivery service and mobility supports. The mobility support is very important for elderly people to enhance the motivation for going out and to prevent them from locomotive syndrome and dementia. We have developed smart senior cars for elderly people shown in Fig.2 (a) and we will use electric wheelchairs in this research. Therefore, we are developing a human tracking system and navigation system of mobility support robots towards the realization of cyber-physical system.

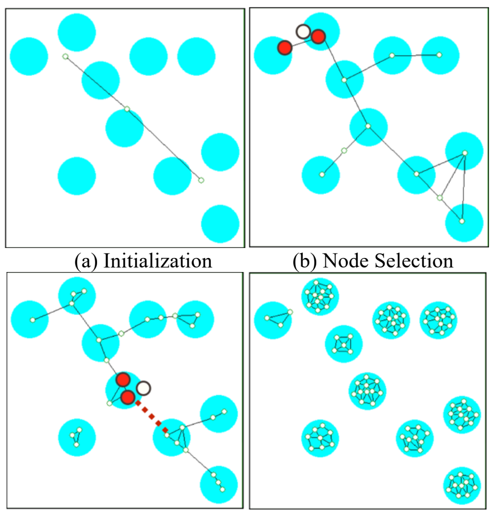
We are developing a mesoscopic simulation [18] in the cyber space (Fig.2 (b)). Each person and robot can take multi-objective behaviors of collision avoidance and target tracing by using fuzzy control [14,19,20]. The numbers of humans and mobility support robots are 10 and 4, respectively, in this simulation. The position of humans and robots is transferred to the macroscopic simulation, and their trajectory patterns are extracted from the movement data by MS-BL GNG. We generated a data set composed of 3200 points as the trajectories of humans and robots by using the mesoscopic simulation.



(a) An example of smart senior car

(b) Mesoscopic simulation of humans and mobility support robots

Fig. 2. Towards cyber-physical system using mobility support robots



(a) Initialization (b) Node Selection

(c) Edge removal (d) Topological mapping

Fig. 3. A learing process of standard GNG.

3. MULTI-SCALE BATCH-LEARNING GROWING NEURAL GAS

3.1. A Methodology of Growing Neural Gas

This section discusses the performance of standard GNG. First, we explain a learning process of standard GNG (in Fig.3). The notations used in GNG are shown as follows.

w_i : the nth dimensional vector of i-th node
A : A current set of nodes
E_i : accumulated error variable
N_i : a set of nodes connected to the i-th node
$c_{i,j}$: edge between the i-th and j-th nodes
$a_{i,j}$: age of edge between the i-th and j-th nodes

Step 1: Generate two or three nodes with the edge connectivity (Fig.3 (a)).
Step 2: Start training with the iteration (it =1) to λ times of weight update
Step 3: Update the network according to a sample data
3-1. Select the nearest node (winner), s_1 and the second-nearest unit, s_2 with a sample data v given as an input to the network according to the probability with $p(v)$ (Fig.3 (b))

$$s_1 = \arg\min_{i\in A} \left\| v - w_i \right\| \tag{1}$$

$$s_2 = \arg\min_{i\in A\setminus\{s_1\}} \left\| v - w_i \right\| \tag{2}$$

3-2. Update the accumulated error by adding the distance between the input and reference vector.

$$E_{s_1} \leftarrow E_{s_1} + \left\| v - w_{s_1} \right\| \tag{3}$$

3-3. Update the reference vectors of the winner and its direct topological neighbors,

$$w_{s_1} \leftarrow w_{s_1} + \eta_1 \left(v - w_{s_1} \right) \tag{4}$$

$$w_j \leftarrow w_j + \eta_2 \left(v - w_j \right) \quad if\ c_{s_1,j} = 1 \tag{5}$$

where η_1 and η_2 are the learning rate.
3-4. If there is no edge between nodes s_1 and s_2, the age is reset to 0 ($a_{s_1,s_2} = 0$). Otherwise, an edge is added to the network ($c_{s_1,s_2} = 1$). The age of edges connecting with the node s_1 is incremented,

$$a_{s_1,j} \leftarrow a_{s_1,j} + 1 \quad if\ c_{s_1,j} = 1. \tag{6}$$

Remove the edge with an age larger than $a_{\max}$ (Fig.3 (c)). As a result, remove the nodes having no edge.
Step 4: Decrease the error variables of all nodes,

$$E_i \leftarrow E_i - \beta E_i \quad (\forall i \in A) \tag{7}$$

Step 5: Increment iteration times, it. Continue with Step 3, if the iteration times is not an integer multiple of a parameter λ.
Step 6: Insert a new node as follows;
6-1. Select the node q with the maximum accumulated error.

$$q = \arg\max_{i\in A} E_i \tag{8}$$

6-2. Select the node f with the maximum accumulated error among the neighbors of q.
6-3. Add a new node r to the network and interpolate its reference vector from q and f.

$$w_r = 0.5 \cdot \left(w_q + w_f \right) \tag{9}$$

6-4. Insert the edges connecting the new node r with nodes q and f, and remove the original edge between q and f.
6-5. Decrease the error variables of q and f by a fraction α.

$$E_q \leftarrow E_q - \alpha E_q \tag{10}$$

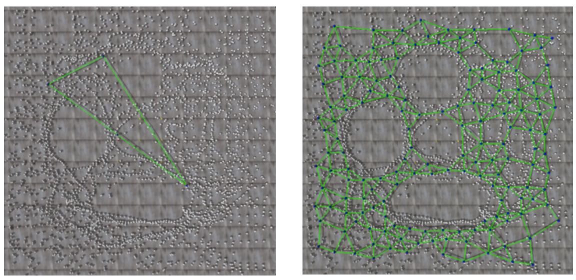
$$E_f \leftarrow E_f - \alpha E_f \tag{11}$$

6-6. Interpolate the error variable of r from q and f

$$E_r = 0.5 \cdot \left(E_q + E_f \right) \tag{12}$$

Step 7: Continue with step 2 if the stopping criterion (*e.g.*, network size or some performance measure) is not yet fulfilled.

As a result, GNG can conduct clustering and topological mapping in each cluster simultaneously (Fig.3 (d)). Figure 4 shows a typical learning result of topological feature extraction by standard GNG where $\eta_1 = 0.05$; $\eta_2 = 1.0$; $a_{\max} = 50$; λ=500; β=0.05. The total number of data sampling is 200,000. These parameters are decided experimentally. Table 1 shows the comparison results of average distance against all data after the learning. Figure 5 shows the comparison results of the average distance against all data every 100 times of learning by the different parameter setting of η_1 and η_2. There is no significant difference of the learning results among them based on the discussion on the tradeoff between the learning rate and its corresponding learning stability (convergence property). However, the parameter setting is very sensitive to its learning performance. Therefore, we still have a problem that we carefully have to decide the setting parameters according to the aim of topological mapping, e.g., online adaptability and offline optimality according to sampling data set.



(a) Data distribution (3200 points) (b) A learning result
Fig. 4. A typical learning result of topological feature extraction by standard GNG.

Table 1. Comparison results of average distance after learning

(η_1, η_2)	(0.02, 0.005)	(0.05, 0.01)	(0.1, 0.05)
Average Distance	0.341754	0.337413	0.36549

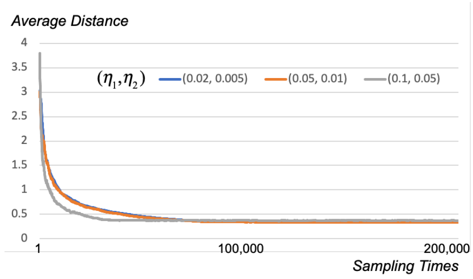


Fig. 5. A typical Comparison results of the different parameter setting of η_1 and η_2.

3.2. A method of multi-scale batch-learning for GNG

In general, we can take one of three training approaches of stochastic gradient decent, Mini-batch gradient decent, and batch gradient decent. The stochastic gradient descent, an iterative method to optimize an objective function, is used in the original GNG. The reference vectors (nodes) are updated when a sample data is given to the network. The mini-batch gradient decent calculates the sum of weight update according to two or more data (less than the full size of data set). The mini-batch gradient decent has been often used in the training of deep learning methods. While the number of weight parameters in the deep learning methods is predefined, the number of nodes is increasing in GNG. Therefore, we should use multi-scale size of data from a small size to full size in order to realize fast and efficient training of GNG. Fig.6 shows an updating strategy of multi-scale batch learning (MS-BL) where the number of data and maximal number of nodes are D and N, respectively.

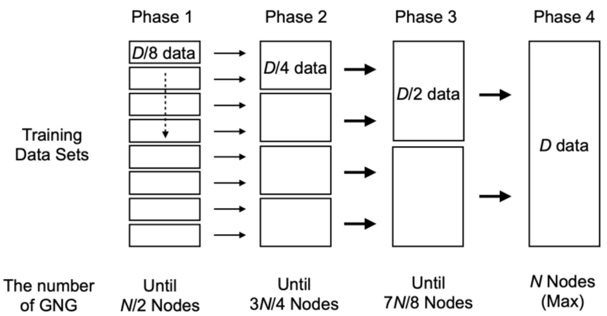


Fig. 6. Learning phase update in MS-BL-GNG (λ_L={D/8, D/4, D/2, D}, μ_L ={N/2, 3N/4, 7N/8, N} where D and N are the number of data and maximal number of nodes, respectively, in the above example)

We divide the full size of data set into 8 mini-batch size of data sets in this example where the number of data in the phase level (L) is represented as λ_L. Individual mini-batch size of data sets are used sequentially to update the reference vector. A new node is added to the network after the weight update by one mini-batch learning. As a result, the number of nodes is increasing faster in the lower phase level. If the number of nodes reaches the parameter (μ_L), the learning phase level to the next level. MS-BL-GNG tries to cover the overall data set in the beginning and can conduct fine topological mapping by using full size of data set in the final phase.

We modify the learning algorithm of a standard GNG in the following.

Step 1: Generate two or three nodes with the edge connectivity. Initialize the phase level (L=1).

Step 2: Initialize the temporal weigh update ($\Delta w_i = 0, i \in A$), selection times ($x_i = 0, i \in A$) and the temporal edge connectivity ($c'_{i,j} = 0,\ i,j \in A$). Start the multi-scale training with the iteration (it =1).

Step 3: Update the temporal weigh update and the temporal edge connection according to a sample data

3-1. Select the nearest node (winner), s_1 and the second-nearest unit, s_2 with a sample data v given as an input to the network according to the probability with $p(v)$

3-2. Update the accumulated error by adding the distance between the input and reference vector.

3-3. Calculate weight updates of the winner and its direct topological neighbors

$$\Delta w_{s_1} \leftarrow \Delta w_{s_1} + \eta_1 \left(v - w_{s_1} \right) \tag{4'}$$

$$\Delta w_j \leftarrow \Delta w_j + \eta_2 \left(v - w_j \right) \quad if\ c_{s_1,j} = 1 \tag{5'}$$

Increment the selection times ($x_{s_1}++,\ x_j++ (j \in N_{s_1})$).

3-4. Update the temporal edge connectivity ($c'_{s_1,s_2} = 1$).

Step 4: Decrease the error variables of all nodes.

Step 5: Increment iteration times, it. Continue with Step 3, if the iteration times is not an integer multiple of λ_L.

Step 6: Update the weights by the MS-BL, update the edge connectivity, and remove the nodes with $x_i = 0$.

$$w_i \leftarrow w_i + \Delta w_i / x_i, \quad if\ x_i > 0 \tag{13}$$

$$c_{i,j} = \begin{cases} c'_{i,j} & if\ c'_{i,j} = 1 \\ 0 & otherwise \end{cases} \tag{14}$$

Step 7: Insert a new node. If the number of nodes (A) reaches the parameter (μ_L), increment the learning phase level.

Step 8: Continue with step 2 if the stopping criterion (*e.g.*, network size or some performance measure) is not yet fulfilled.

Although a standard GNG needs the age of edges, MS-BL GNG does not need to consider it. Furthermore, we have to pay much attention to the setting of learning rates in the standard GNG in order to realize the stable topological mapping, but we can use η_1=1 because of batch-learning.

4. SIMULATION RESULTS

This section shows simulation results of the proposed method. The number of data (D) is 3200; the maximal number of nodes (N) is 160; the number of phases is 5. Therefore, we use the parameter setting of λ_L={200, 400, 800, 1600, 3200}, μ_L ={80, 120, 140, 150, 160} in this simulation example. The learning rate of the nearest node (winner) and second nearest node are η_1 = 1.0 and η_2 = 0.05, respectively. The number of overall sampling times is 200,000 (= 62.5 $\cdot$ D). If we use a standard BL-GNG, we can conduct only 62 times of weight updates. Therefore, we don't show simulation results of a standard BL-GNG in this paper.

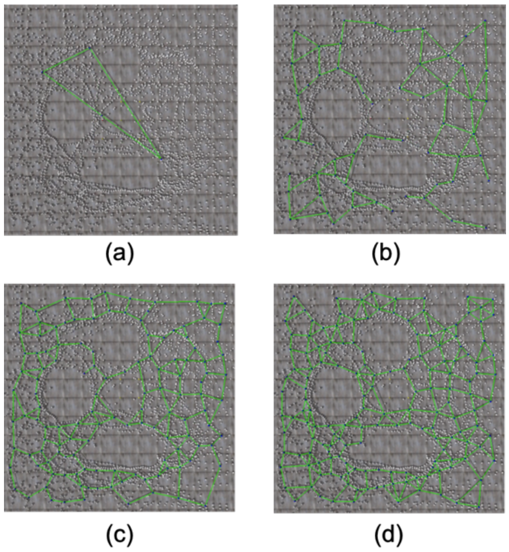
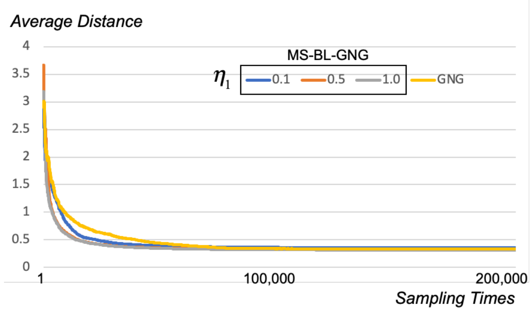
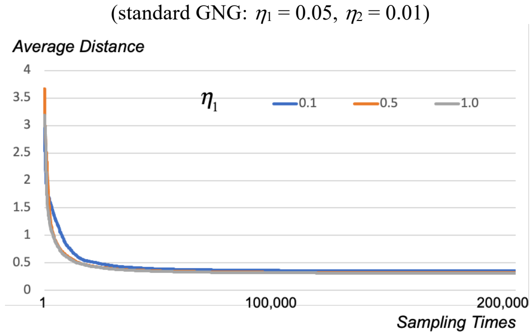


Fig. 7. Examples of the topological stucure for the time series data.



(a) Case 1: Comparison results of the different η_2 where η_1 is fix at 1.0 (standard GNG: $\eta_1 = 0.05$, $\eta_2 = 0.01$)



(b) Case 2: Comparison results of the different η_1 where η_2 is fix at 0.05.

Fig. 8. Change of average distance against all data.

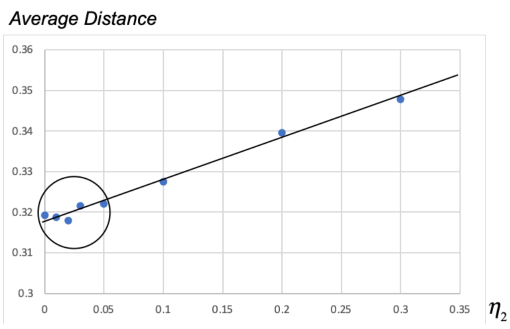


Fig. 9. Comparison of average distance against all data in the learning results obtained by using the different learning rates (η_2) in MS-BL-GNG

Figure 5 shows an example of learning in MS-BL-GNG. The topological structure is updated gradually as the increase of nodes. Figure 6 shows the change of average distance by all data of (a) case 1 where the learning rate (η_2) is 0, 0.05, 0.1, 0.2 and 0.3 and $\eta_1 = 1.0$ and (b) case 2 where the learning rate (η_1) is 0.1, 0.5, and 1.0 and $\eta_2 = 0.05$. The result of case 1 shows that the effect of η_2 to the learning speed is very low. It is obvious that the weight update of reference vectors is mainly done by batch-learning where $\eta_1 = 1.0$. The result of case 2 shows there is no significant difference among parameter settings, but there is a residual error in the learning results in case of $\eta_1<1.0$. Furthermore, we compare the learning result between MS-BL-GNG and standard GNG where $\eta_1 = 0.05$ and $\eta_2 = 0.01$. The standard GNG with the optimal parameter setting is almost same with that of MS-BL-GNG. Figure 7 shows the comparison results of the average distance against all data in the learning results (after 200,000 times of data sampling) obtained by using the different learning rates (η_2). The average distance is increasing approximately linearly as the increase of the learning rate (η_2). The original aim of GNG is to optimize the following function,

$$J_{GNG}=\sum_{i=1}^{D}\sum_{j=1}^{N} r_{i,j}\left\|v_i-w_v\right\|^2+\sum_{i=1}^{D}\sum_{j=1}^{N} u_{i,j}\left\|v_i-w_v\right\|^2$$

$$r_{i,j}=\begin{cases}1 & \text{if } j=\arg\min_{k\in A}\left\|v_i-w_k\right\| \\ 0 & \text{otherwise}\end{cases} \tag{15}$$

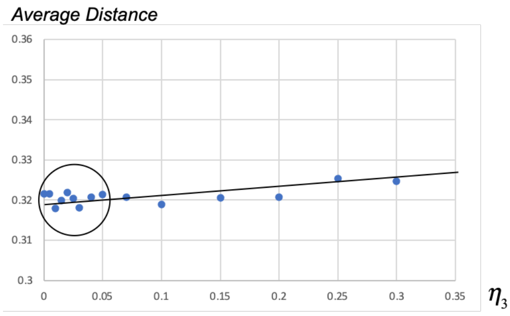
$$u_{i,j}=\begin{cases}\gamma & \text{if } c_{h,j}=1\,(h=\arg\min_{k\in A}\left\|v_i-w_k\right\|) \\ 0 & \text{otherwise}\end{cases}$$

where γ is the membership degree to the h-th node connected with the j-th node. Therefore, the average distance is longer as the increase of the learning rate on the second nearest node. If we use $\eta_2 = 0$, its corresponding learning process is the same with k-means algorithm, because the learning is done according to the average of data selected by the nearest node. The advantage of GNG is in the learning of topological mapping based on the nearest and second nearest node, but we should consider the effect of distance to the learning. Therefore, we use a distance-based learning method used in ART and SOM,

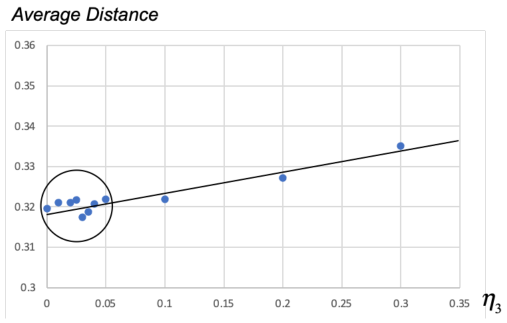
$$\eta_2=\eta_3\exp\left(-\left\|v-w_j\right\|^2/\eta_4\right) \tag{16}$$

where η_3 is the learning rate and η_4 is a parameter related to the variance. The effect of distance is more sensitive as the increase of η_4. This means the learning tendency of the distance-based MS-BL-GNG in case of higher learning rate (η_4) is similar to that of the standard MS-BL-GNG.

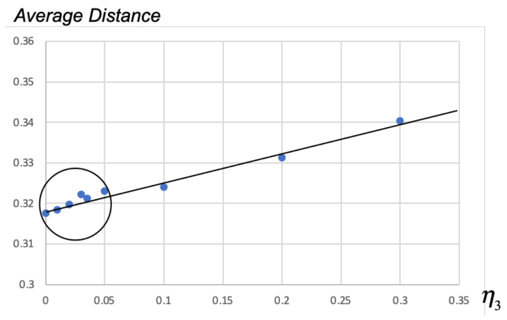
Finally, Fig. 11 shows an example of topological feature extraction in a navigation task of mobile support robots. The obtained topological map represents the features of trajectories of human and robot movements. The density of nodes shows the crowdedness when many humans and robots encounter each other. We can apply the proposed method to extract human movement patterns in public areas such as departments or airports and can use the navigation tasks of mobile robots. Furthermore, the topological map can be applied to improve the service to humans.



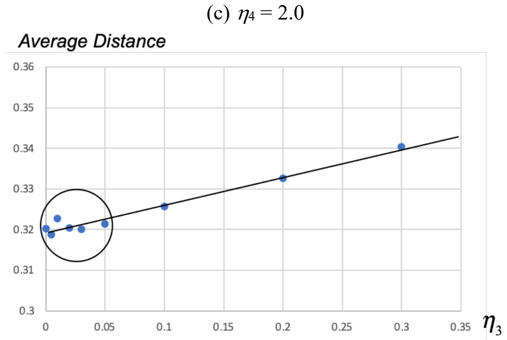
(a) $\eta_4 = 0.5$



(b) $\eta_4 = 1.0$



(c) $\eta_4 = 2.0$



(d) $\eta_4 = 3.0$

Fig. 10. Comparison of average distance against all data obtained by using the different learning rates (η_3) in distance-based MS-BL-GNG.

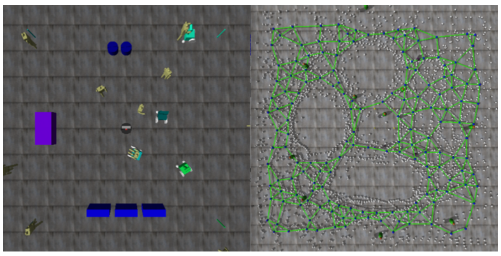


Fig. 11. Topological feature extraction in a tracking task (left: a simulation of human-robot interaction, right: the tracking of humans and robots)

5. Conclusion

In this paper, we proposed the multi-scale batch-learning growing neural gas (MS-BL GNG) to extract topological features in navigation tasks of mobility support robots. We first proposed the MS-BL algorithm based on the property of GNG. Since the number of nodes is increasing in GNG, we can use the small size of data set in the beginning of the learning. Therefore, we proposed a multi-scale approach from a small size to full size in order to realize fast and efficient training of GNG. Next, we show the preliminary experimental results of a simple multi-scale extension of a standard GNG. We discussed the sensitivity of parameter settings to learning results related to the aim of topological mapping in GNG. Next, we applied the distance-based learning method to the weight update of the second nearest node and show the sensitivity of the distance-based learning method in MS-BL-GNG. Furthermore, we discussed the learning performance of MS-BL-GNG by using different hyper parameters. The comparison results show that we can realize various types of topological mapping by considering the effect of learning sensitivity.

Finally, we show the feature extraction results in the navigation tasks of mobility supports robots. The extracted features will be used in the path planning and navigation of mobility supports robots. As a future work, we have to discuss the applicability of multi-scale batch learning to dynamically changing data in online adaptation. Especially, we will discuss how to generate multi-scale data set in real time. Furthermore, we will propose a path planning method of mobile robots based on topological mapping.

Acknowledgements

This work was partially supported by JST [Moonshot RnD][Grant Number JP- MJMS2034].

References:

[1] P.Verhoefa, T.Broekhuizena, et al, Digital transformation: A multidisciplinary reflection and research agenda, Journal of Business Research, Vol.122, 2021, pp.889-901.

[2] C.Greer, M.Burns, D.Wollman, E.Griffor, Cyber-Physical Systems and Internet of Things, NIST Special Publication 1900-202, 2019.

[3] T.Banyai, A.Petrilloand, F. De Felice, Industry 4.0: Impact on Intelligent Logistics and Manufacturing, IntechOpen, 2020.

[4] Bernd Fritzke. "A self-organizing network that can follow non-stationary distributions." Artificial Neural Networks—ICANN'97. Springer Berlin Heidelberg, 1997. 613-618.

[5] B. Fritzke: A growing neural gas network learns topologies; Advances in Neural Information Processing Systems, Vol. 7, 1995, pp. 625-632.

[6] T. M. Martinetz and K. J. Schulten: A "neural-gas" network learns topologies; Artificial Neural Networks, Vol. 1, 1991, pp. 397-402.

[7] B. Fritzke: Unsupervised clustering with growing cell structures; Neural Networks, Vol. 2, 1991, pp 531–536.

[8] T. Kohonen: Self-Organizing Maps; Springer, 2000.

[9] N.Mirehi, M.Tahmasbi ,and A.T.Targhi, "Hand gesture recognition using topological features," Multimed Tools Appl, Vol.78, 2019, pp.13361-13386.

[10] P. M. Yanik et al., "Use of kinect depth data and Growing Neural Gas for gesture based robot control," 2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops, 2012, pp. 283-290.

[11] 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," 2007 International Joint Conference on Neural Networks, 2007, pp. 1900-1905.

[12] A.Angelopoulou, J.Garcia-Rodriguez, S.Orts-Escolano, E.Kapetanios, X.Liang, B.Woll, A.Psarrou, "Evaluation of different chrominance models in the detection and reconstruction of faces and hands using the growing neural gas network," Pattern Analysis and Applications, Vol.22, No.4, 2019, pp.1667-1685.

[13] Y.Toda, W.Chin, N.Kubota, Unsupervised neural network based topological learning from point clouds for map building, 2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS), 2017.

[14] Y.Toda, N.Kubota, Topological Structure Learning Based Enclosing Formation Behavior for Monitoring System, 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp.831-836.

[15] C.-S.Lee, M.-H.Wang, Y.-L.Tsai, W.-S.Chang, M.Reformat, G.Acampora, N.Kubota, FML-Based Reinforcement Learning Agent with Fuzzy Ontology for Human-Robot Cooperative Edutainment, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 28 (6), 2020, pp.1023-1060.

[16] J.Woo, Y.Ohyama, N.Kubota, Robot partner development platform for human-robot interaction based on a user-centered design approach, Applied Sciences, 10 (22) , 2020, pp.1-20.

[17] S.Yamamoto, J.Woo, W.H.Chin, K.Matsumura, N.Kubota, Interactive information support by robot partners based on informationally structured space, Journal of Robotics and Mechatronics, 32 (1) , 2020, pp. 236-243.

[18] S.Oshio, K.Kaneko, N.Kubota Multi-scopic Simulation for Human-robot Interactions Based on Multi-objective Behavior Coordination, The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2021), 2021 (accepted).

[19] T.Fukuda, N.Kubota, An Intelligent Robotic System Based on A Fuzzy Approach, Proceedings of The IEEE, Vol.87, No.9, 1999, pp.1448-1470.

[20] N.Kubota, N.Aizawa, Intelligent Control of Multi-Agent System based on Multi-Objective Behavior Coordination, Proc. of the 2008 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008), pp.1458-1463, Hong Kong, China, June 1-6, 2008.