Springer

KSME

Journal of Mechanical Science and Technology 32 (10) (2018) 4933~4939
www.springerlink.com/content/1738-494x(Print)/1976-3824(Online)
DOI 10.1007/s12206-018-0941-y

A study on the optimal route design considering time of mobile robot using recurrent neural network and reinforcement learning †

Min Hyuk Woo, Soo-Hong Lee* and Hye Min Cha

School of Mechanical Engineering, Yonsei University, Seoul 03722, Korea

(Manuscript Received November 2, 2017; Revised March 14, 2018; Accepted July 2, 2018)

Abstract

Recently, the robots market is growing rapidly, and robots are being applied in various industrial fields. In the future, robots will work in more complex and diverse environments. For example, a robot can perform one or more tasks and collaborate with people or other robots. In this situation, the path planning for the robots to perform their tasks efficiently is an important issue. In this study, we assume that the mobile robot performs one or more tasks, moves various places freely, and works with other robots. In this situation, if the path of the mobile robot is planned with the shortest path algorithm, waiting time may occur because the planned path is blocked by other robots. Sometimes it is possible to complete a task in a shorter time than returning or performing another task first. That is, the shortest path and the shortest path do not coincide with each other. The purpose of this study is to construct a network in which the mobile robot designs the shortest path planning considering shortest time by judging itself based on environment information and path planning information of other robots. For this purpose, a network is constructed using a recurrent neural network and reinforcement learning is used. We established the environment for network learning using the robot simulation program, V-Rep. We compare the effects of various network structures and select network models that meet the purpose. In the future work, we will try to prove the effect of network by comparing existing algorithm and network.

Keywords: Mobile robot; Path planning; Recurrent neural network; Reinforcement learning

1. Introduction

The robots market is growing rapidly in recent years. There are two major factors. One is the development of artificial intelligence that shows the possibility of replacing human role in many fields. Another is that many companies are considering adopting robots as a solution because of the increase in labor costs and the reduction of manpower. In particular, this growth is characterized by a rapid growth in the service robot market compared to the traditional industrial robot's market. In the future, robots need to work in environments that are more complex and various than traditional environment. For example, a robot can perform one or more tasks and collaborate with people or other robots [1]. In such an environment, robots should be able to replace people's roles by ensuring the safety of people while moving freely and by performing tasks effectively.

For this purpose, research on mobile robots is active and it is very important to design efficient path planning. The path planning problem is constantly being studied as a representative NP-hard problem. Many researchers have tried to solve this problem by using algorithms such as Dijkstra algorithm, genetic Algorithm and A* algorithm [2-8]. Although there are many existing algorithms related to path planning, it is difficult to apply to the above situation. This is because applications are limited, and the computation time is very long. Due to these limitations, approaches to solving path planning using machine learning have been actively attempted. However, there are some studies that show the effectiveness in applying the machine learning to specific situations. Cruz and Yu studied reinforcement learning with LSTM, Advantage (λ) Learning to reach the target point in T - shaped maze [9]. Bakker have been studying the path planning of multiple robots in the absence of environment information and have used Kernel smoothing technique and reinforcement learning to solve the problem [10].

In this study, we applied the machine learning method and received environment information from the server, and then tried to construct a path planning network model that can cope with situations where mobile robots solve multiple tasks [11-18]. The environmental information received from the server is the image layers that are collected by the sensor and preprocessed by each object [19]. The network is constructed

*Corresponding author. Tel.: +82 2 2123 2823, Fax.: +82 2 312 2159
E-mail address: shlee@yonsei.ac.kr
† Recommended by Associate Editor Yang Shi

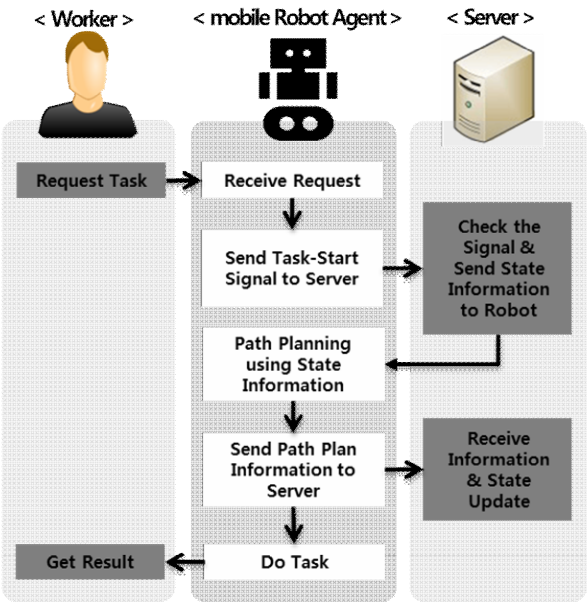

Fig. 1. Scenarios of path planning and task execution of mobile robots.

using a recurrent neural network and reinforcement learning is used [20-23]. We established the environment for network learning using the robot simulation program, V-Rep [24, 25].

In Sec. 2, we introduce a scenario in which mobile robots work. Based on this, target mobile robots for learning network models, environment of mobile robots, vision sensors to obtain inputs to be used in networks will be introduced. In Sec. 3, we introduce the three network models used in this study and introduce the data pre-processing process and reward algorithm for network learning. In Sec. 4, we will show and compare the results of three network models and Sec. 5 presents conclusions and future work.

2. Environment for network learning

2.1 Research scenario

The path plan and task execution scenarios of the shortest time path design mobile robot system proposed in this study are shown in the Fig. 1. The scenario is composed of the worker, the mobile robot agent, and the central server. The worker performs his or her role in the factory line and requests the mobile robot agent for various tasks. The mobile robot agent is a robot equipped with the shortest time path planning network, receives the worker 's request and transmits the task start signal to the central server. The central server receives the signal and transmits information such as the current work environment image and the path planning of the other robot so that the mobile robot agent can plan the path. Based on this, the mobile robot agent plans the path and sends the planned path to the central server. The central server updates the current state information to reflect the added path plan for the other mobile robot agent. The mobile robot agent moves the planned path and provides the result to the worker.

Table 1. Environment properties.

Total width (m)	15
Total length (m)	25
Number of mobile robots (unit)	3
Number of shelves (unit)	120
Width of shelf (m)	0.5
Length of shelf (m)	1
Distance with shelf and wall (m)	2
Distance with each shelves (m)	1.5

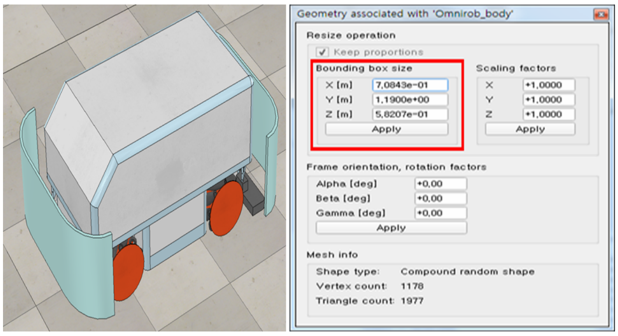

Fig. 2. Mobile robot modeling using V-Rep.

2.2 Mobile robot modeling

In Amazon 's Kiva robot system, the central server controls the entire robot and identifies the mobile robots' location by making whole environments to grid and attaching a QR code to each lattice [26]. In order to apply the Kiva robot systems, the whole environment must be rebuilt. Also, it is difficult to apply it to various environments because it is specialized only in the role of lifting and moving the shelf. The mobile robot, which is the target model of this study, is KUKA 's KMR IIWA [27]. This robot can perform various tasks because manipulator platform and mobile platform are combined. In addition, autonomous navigation software can be installed, so this can move freely.

2.3 Work environment modeling

It shows the work environment modeling of mobile robots in Fig. 3. A total of three mobile robots work in the work environment. Two of mobile robots already have their own path plan and automatically move when the simulation starts. The mobile robot located at the upper and lower part has already been designed for the route. The central mobile robot is the target of network learning. On the right side of the mobile robot, shelves are arranged, and the shelves are composed of 120 pieces.

2.4 Vision sensor

In the path planning of the mobile robot, it is indispensable to recognize the current position of the mobile robot, the target

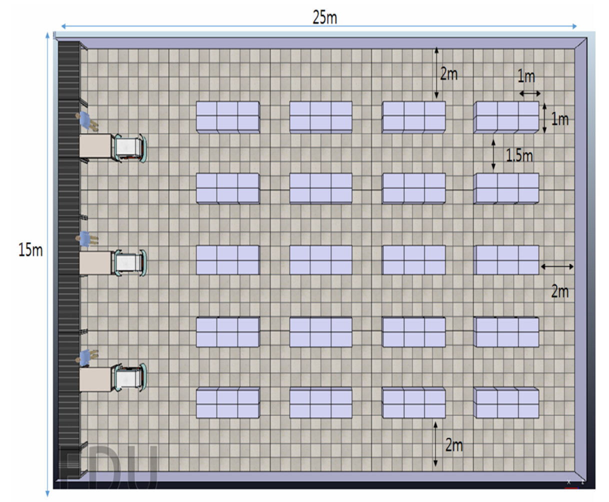

Fig. 3. Work environment modeling of mobile robot modeling using V-Rep.

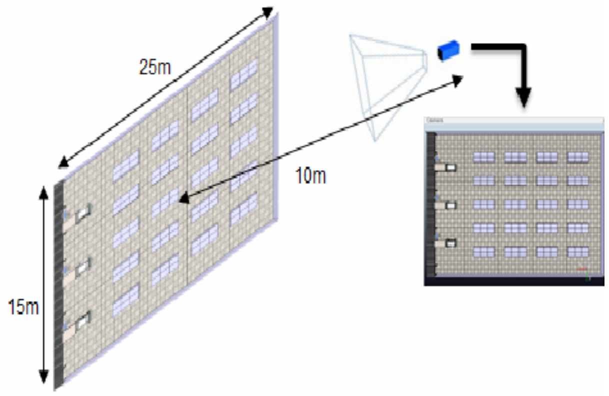

Fig. 4. Vision sensor for getting the current environment information.

point, and the position of the obstacle. Unlike Amazon 's Kiva system, this research uses a vision sensor as a method to obtain current environmental information. A vision sensor is installed 10 m above the working environment and real-time images of the environment are obtained. This image will be used as learning data of the network, and data will be exchanged between the mobile robot agents and the central server. This method is likely to be realized in the near future because of the rapid development of object recognition technology and image processing technology based on the vision. In addition, since it is necessary to install only the vision sensor without installing a new working environment, it is also effective in reducing the initial cost.

3. Learning

3.1 Data pre-processing

In this study, we use image data of work environment as learning data of network. In addition, we assume that the central server is equipped with an image recognition technology capable of distinguishing mobile robots, obstacles and target

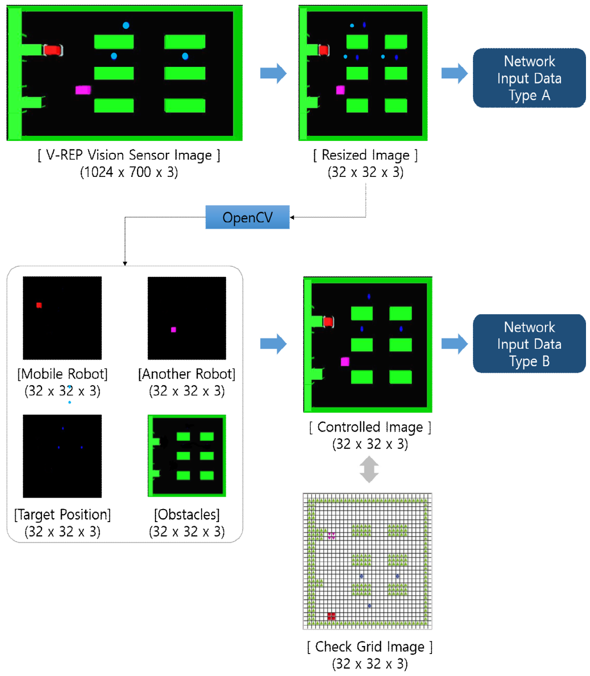

Fig. 5. Image pre-processing flow for learning network.

points from images. In order to realize this, two steps of preprocessing are performed in this study. It is the first preprocessing step in the simulation program stage. Since we need to be able to distinguish objects from images according to the above-mentioned assumptions, we set the colors of objects in the simulation program to be distinguished. The target mobile robot is red, the other mobile robots are pink, the target positions are blue, and the obstacles are green.

The next step is to resize and reorganize the image using OpenCV. The input type A in Fig. 5 is image data modified to 84x84 size. The input type A is used for verification only because it takes a long time to learn V-rep simulations in real time to acquire images and to learn. To make the input type B, we extract location information of target mobile robot, other mobile robots, obstacles, and target points that are color-coded in the resized image. The input type B is used for learning because the learning speed is fast. The input type B is the data reconstructed in the form of a grid map by composing each layer. We use input type B for learning network and validates with input type A for every thousand episodes.

3.2 Reward policy algorithm

As a method for learning the network, reinforcement learning is used among the machine learning methods. Reinforcement learning is a way that the network takes action from current state, receives reward from the changed state and maximizes total reward. Unlike supervised learning or unsupervised learning, it does not need prepared data for network learning, so it is a suitable method for problem situations with various environments. For the network to solve the problem

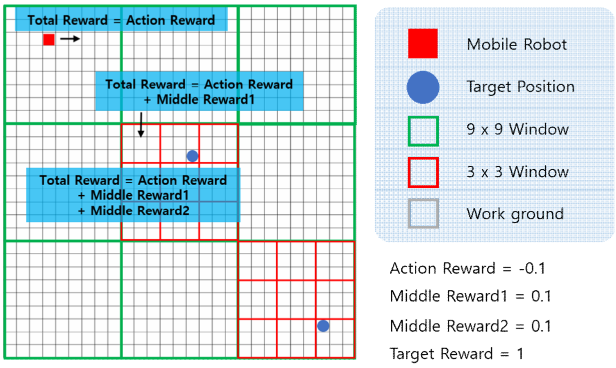

Fig. 6. Image pre-processing flow for learning network.

Algorithm 1 Reward Policy

Total Reward R_t, Action Reward R_a, Middle Reward $R_{m,i}$
Action State Width M_x, Action State Height M_y
Window Unit Size U
Target Positions T
Mobile Robot Positions S

Begin
Get i where $U^i \le \min(M_x, M_y) < U^{i+1}$
$R_{mi} \leftarrow 0.0$
$R_a \leftarrow -0.1$
for every T_j in T
if (round(T_j / U^i) == round(S / U^i)) → $R_{m,i} = 0.1$
if (round(T_j / U^{i-1}) == round(S / U^{i-1})) → $R_{mi-1} = 0.1$
...
if (round(T_j / U^1) == round(S / U^1)) → $R_{m,1} = 0.1$
if (T_j == S) → $R_{m0} = 1$
end
return $R_t = R_a + \sum R_{mi}$
end

Fig. 7. Reward policy algorithm.

situation we assumed, it is important to set up an appropriate reward policy. Since the work environment implemented in this study is wide, it is very difficult for the network to converge in a way that reward is given only when it reaches the target point. Therefore, we implemented a reward policy algorithm that tries to find the target point by layering the whole map from large block to small block and giving intermediate compensation.

3.3 Network models

The reinforcement learning method basically deals with the Markov decision process problem. The Markov decision process is a problem in which the current state and the action taken in that situation define the reward, and previous or future states and actions can not affect the current reward. However, in the problem situation of this study, time series data such as path planning information and task execution time information of other robots are included in the current state

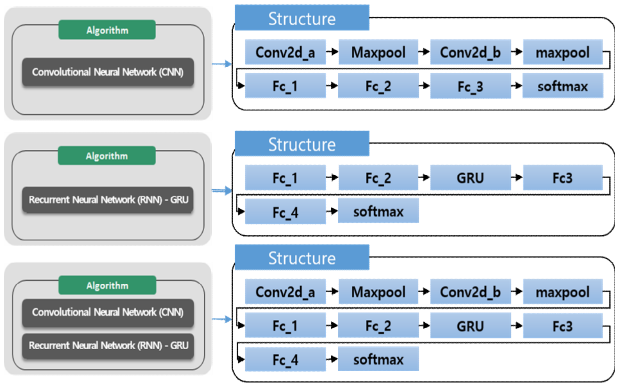

Fig. 8. Network models.

information for the path planning considering the shortest time. In other words, since states information are not independent, additional method are needed to apply the reinforcement learning method efficiently. In this study, we use recurrent neural network to structure network. Recurrent neural networks are widely used in speech recognition, music genre classification, video classification, and string generation. It has an excellent network structure for processing time series data. We conduct learning and verifying three network models and compare the results. One is consisting of convolution and fully connected neural network. Another is consisting of recurrent and fully connected neural network. The other is consisting of convolution, recurrent and fully connected neural network.

4. Results

We compared the learning results of network models with the loss and the score that is sum of total reward in episode. Fig. 8 shows the loss graph of each network model. We use the double Deep Q Network method among the reinforcement learning method. In this method, the loss is calculated by difference between the prediction of reference network and the learning network. The horizontal axis of the graph represents the episode, and the vertical axis represents the loss value [28]. In order to make sure network learning is working, the loss value should decrease as the episode becomes larger. Among the three network models, the model using only the convolution neural network is the only type in which the loss graph converges, and the model including the recurrent neural network is not correctly learning.

Fig. 9 shows the score graph of each network model. The horizontal axis of the graph represents the episode and the vertical axis represents the score that is sum of the reward. The trend of the graph shows that the score is maximized so that the network functions well. But when the simulation is confirmed, the mobile robot does not move. When the mobile robot hits the obstacle, it receives negative reward, so the more the learning progresses, the more the negative experience is accumulated, and the robot converges to a form that does not move.

In the previous experiments, we used input data type B,

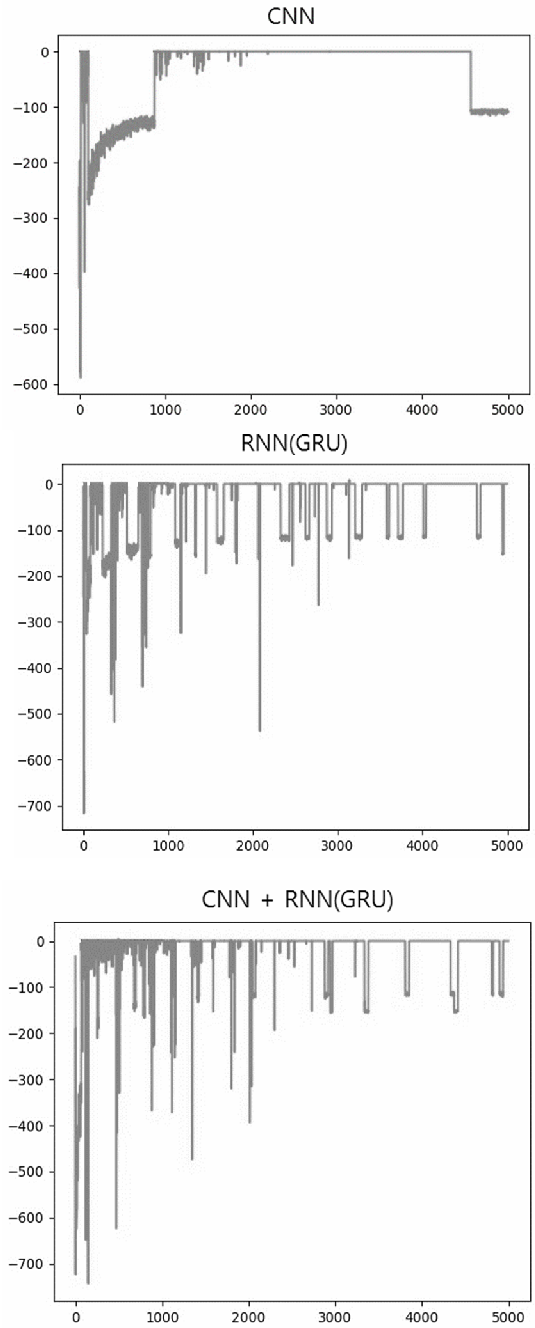

Fig. 9. Loss graphs of network models.

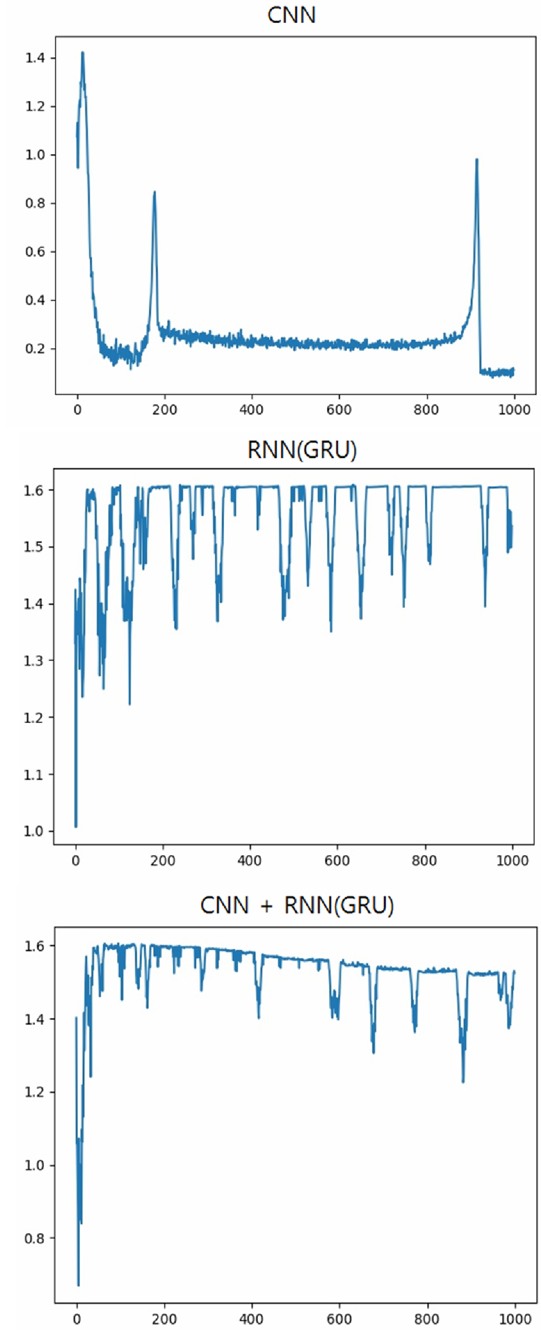

Fig. 10. Score graphs of network models.

which was pre-processed twice, for learning network. But the networks did not learn properly. Therefore, we have experimented with learning network using input type A. The network model used in this experiment is a model using convolution and recurrent neural network. The loss and score results are shown in Fig. 10. Unlike the previous experiments, the loss tendency to be lowered even though the recurrent neural network was used. From this, we confirmed the possibility that the network can solve the problem by modifying the preprocessing process of the input data.

5. Conclusions

In this study, we used machine learning as a way to design efficient path plan of a mobile robot. In addition, image data of working environment is used for location recognition and network learning of mobile robots. This approach is effective because it is a flexible method that can be applied in a variety of complex situations. Vision sensors make it easier to apply the method in existing work environments. It can reduce the initial cost since it does not require the factory to be rebuilt.

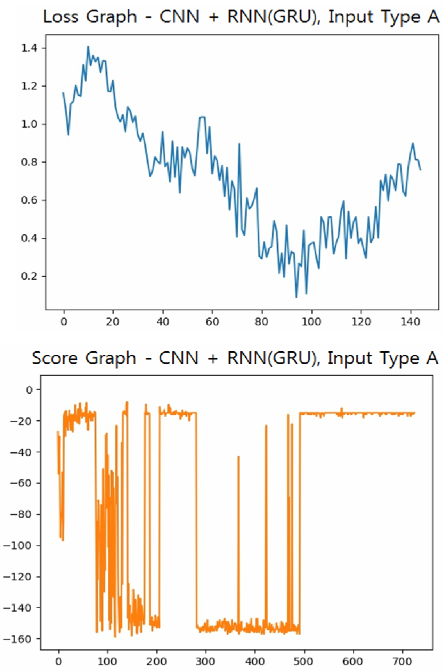

Fig. 11. Loss and score graph using input type A for learning network.

The method used in this study is not how the central server plans the route for the entire robot, but how each robot plans it. Distributed agents are a good way to reduce the load on the central server and minimize the likelihood of an accident when the central server goes down. It also has the benefit of being able to change the number of robots running flexibly. This study will be an effective alternative in the recent trend of expanding demand and application area of robots. However, since the performance of the network is not yet as effective as the target, it is necessary to improve the performance. In the future, we will modify the reward policy and network variables. Also, we improve the network to solve the problem by using various techniques.

Acknowledgements

This work was supported by the Technology Innovation Program (Project Number: 10082577) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

[1] N. Ku, S. Ha and M. I. Roh, Design of controller for mobile robot in welding process of shipbuilding engineering, *Journal of Computational Design and Engineering*, 1 (4) (2014) 243-255.

[2] V. Roberge, M. Tarbouchi and G. Labonte, Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning, *IEEE Transactions on Industrial Informatics,* 9 (1) (2013) 132-141.

[3] Y. Zhang, D. W. Gong and J. H. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neurocomputing,* 103 (2013) 172-185.

[4] M. A. Hossain and I. Ferdous, Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique, *Robotics and Autonomous Systems*, 64 (2015) 137-141.

[5] M. A. Contreras-Cruz, V. Ayala-Ramirez and U. H. Hernandez-Belmonte, Mobile robot path planning using artificial bee colony and evolutionary programming, *Applied Soft Computing,* 30 (2015) 319-328.

[6] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico and L. Jurisica, Path planning with modified a star algorithm for a mobile robot, *Procedia Engineering,* 96 (2014) 59-69.

[7] N. Cao, K. H. Low and J. M. Dolan, Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms, *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (2013) 7-14.

[8] J. W. Lee, D. H. Lee and J. J. Lee, Global path planning using improved ant colony optimization algorithm through bilateral cooperative exploration, *Proc. of the 5th IEEE International Conference on Digital Ecosystems and Technology Conference (DEST)* (2011) 109-113.

[9] D. L. Cruz and W. Yu, Path planning of multi-agent systems in unknown environment with neural kernel smoothing and reinforcement learning, *Neurocomputing,* 232 (2017) 34-42.

[10] B. Bakker, Reinforcement learning with LSTM in non-Markovian tasks with long-term dependencies, *Technical report,* Dept. of Psychology, Leiden University (2001).

[11] S. Y. Fu, L. W. Han, Y. Tian and G. S. Yang, Path planning for unmanned aerial vehicle based on genetic algorithm, *2012 IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing, IEEE* (2012) 140-144.

[12] J. Ni, X. Li, X. Fan and J. Shen, A dynamic risk level based bioinspired neural network approach for robot path planning, *World Automation Congress* (2014) 829-833.

[13] D. Zhu, W. Li, M. Yan and S. X. Yang, The path planning of AUV based on DS information fusion map building and bio-inspired neural network in unknown dynamic environment, *International Journal of Advanced Robotic Systems,* 11 (3) (2014).

[14] S. X. Yang and C. Luo, A neural network approach to complete coverage path planning, *IEEE Transactions on Systems, Man and Cybernetics,* 34 (1) (2004) 718-724.

[15] H. Qu, S. X. Yang, A. R. Willms and Z. Yi, Real-time robot path planning based on a modified pulse-coupled neural network model, *IEEE Transactions on Neural Networks,* 20 (11) (2009) 1724-1739.

[16] R. Kala, A. Shukla, R. Tiwari, S. Rungta and R. R. Janghel, Mobile robot navigation control in moving obstacle environment using genetic algorithm, artificial neural networks and A* algorithm, *Computer Science and Information Engineering, 2009 WRI World Congress*, 4 (2009) 705-713.

[17] D. Xin, C. Hua-hua and G. Wei-kang, Neural network and

genetic algorithm based global path planning in a static environment, *Journal of Zhejiang University-Science A*, 6 (6) (2005) 549-554.

[18] L. E. Zarate, M. Becker, B. D. M. Garrido and H. S. C. Rocha, An artificial neural network structure able to obstacle avoidance behavior used in mobile robots, *IECON 02 [IEEE 2002 28th Annual Conference of the Industrial Electronics Society]*, 3 (2002) 2457-2461.

[19] S. J. Huang, S. Liu and C. H. Wu, Intelligent humanoid mobile robot with embedded control and stereo visual feedback, *Journal of Mechanical Science and Technology*, 29 (9) (2015) 3919-3931.

[20] H. Brahmi, B. Ammar and A. M. Alimi, Intelligent path planning algorithm for autonomous robot based on recurrent neural networks, *International Conference Advanced Logistics and Transport (ICALT)* (2013) 199-204.

[21] T. W. Chow and Y. Fang, A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics, *IEEE Transactions on Industrial Electronics,* 45 (1) (1998) 151-161.

[22] Y. Pan and J. Wang, Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, *IEEE Transactions on Industrial Electronics,* 59 (8) (2012) 3089-3101.

[23] B. Zhang, Z. Mao, W. Liu and J. Liu, Geometric reinforcement learning for path planning of UAVs, *Journal of Intelligent & Robotic Systems,* 77 (2) (2015) 391-409.

[24] E. Rohmer, S. P. Singh and M. Freese, V-REP: A versatile and scalable robot simulation framework, *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE* (2013) 1321-1326.

[25] H. T. Hwang, S. H. Lee, J. W. Lee, H. C. Kim, M. H. Woo, K. W. Moon, J. Lu, H. S. Ohk, J. K. Kim and H. W. Suh, Development of a transportability evaluation system using swept path analysis and multi-body dynamic simulation, *Journal of Mechanical Science and Technology,* 31 (11) (2017) 5359-5365.

[26] P. R. Wurman, R. D'Andrea and M. Mountz, Coordinating hundreds of cooperative, autonomous vehicles in warehouses, *AI Magazine,* 29 (1) (2008).

[27] A. Tudico, N. Lau, E. Pedrosa, F. Amaral, C. Mazzotti and M. Carricato, Improving and benchmarking motion planning for a mobile manipulator operating in unstructured environments, *Portuguese Conference on Artificial Intelligence* (2017) 498-509.

[28] V. Mnih et al., Human-level control through deep reinforcement learning, *Nature*, 518 (7540) (2015) 529-533.

Min Hyuk Woo is currently studying his Master's degree at Yonsei University in Seoul, Korea. He received his bachelor's degree in mechanical engineering from Yonsei University in 2013. His current research interests include PLM, Web-based Collaborative Design, CAD/CAM.

Soo-Hong Lee is currently as a Full-time Professor at the Department of Mechanical Engineering, Yonsei University in Seoul, Korea. He received his bachelor's degree in mechanical engineering from Seoul National University in 1981 and his master's degree in mechanical engineering design from Seoul National University in 1983. He completed his Ph.D. from Stanford University, Califor-nia, USA, in 1991. His current research interests include Intelligent CAD, Knowledge-based Engineering Design, Concurrent Engineering, Product Design Management, Product Lifecycle Management, Artificial Intelligence in Design, and Design Automation.

Hye Min Cha is a graduate student at the Department of Integrated Engineering in Yonsei University in Seoul, Korea. She received her bachelor's degree in Mechanical Engineering and Department of Human Environment Design in Yonsei University in 2016. Her current research is related in machine learning, Knowledge-based Engineering Design, CAD/CAM and concurrent design.