Hexagon-Based Q-Learning Algorithm and Applications

Hyun-Chang Yang, Ho-Duck Kim, Han-Ul Yoon, In-Hun Jang, and Kwee-Bo Sim*

Abstract: This paper presents a hexagon-based Q-leaning algorithm to find a hidden target object with multiple robots. An experimental environment was designed with five small mobile robots, obstacles, and a target object. Robots went in search of a target object while navigating in a hallway where obstacles were strategically placed. This experiment employed two control algorithms: an area-based action making (ABAM) process to determine the next action of the robots and hexagon-based Q-learning to enhance the area-based action making process.

Keywords: Area-based action making, hexagon-based Q-learning, ROBOSIM simulator.

1. INTRODUCTION

Currently, robots are coping with tasks in dangerous fields previously performed by men, such as rescue missions in buildings damaged by fire or at sites contaminated by gas, information retrieval from deep seas or space, and weather analysis in extremely cold areas like Antarctica. Sometimes, multiple robots are needed to penetrate especially hard-to-access areas, such as underground insect nests in order to collect more reliable and solid data.

Multiple robot control has received much attention since it offers a new flexible and vigorous way to control multiple agents. For instance, Parker used the heuristic approach algorithm for multiple robots and applied it to cleaning tasks [1]. Ogasawara employed distributed autonomous robotic systems to control multiple robots transporting a large object [2]. However, the greater the dependency on communication in a system is, the more difficult a system hierarchy becomes. Therefore, this study proposes an area-based action making (ABAM) process for instinctive intelligence similar to bee behavior in an apiary. This in turn, is incorporated with hexagon-based Q-learning, which is learned intelligence and helps multiple robots to navigate, avoid collision, and search using their own trajectories.

Reinforcement learning through exploring its environment actively enables an agent to determine what the following action should be. During the exploration of an uncertain state space followed with a reward, the agent learns what to do by continuum of its state history and appropriate propagation of rewards through the state space [3]. This research focused on Q-learning as a reinforcement learning technique because Q-learning is a simple way to solve Markovian action problems with incomplete information. In addition, an agent can map state-action pairs onto expected returns based on the action-value function Q [4]. In addition to this simplicity, Q-learning can be adapted to the real world. For example, state space can be harmonized with the physical space of the real world. An action can be regarded as a physical robot maneuver. This paper proposes that the hexagon-based Q-learning can enhance the area-based action making process so that the learning process can be better adapted to real world situations.

The organization of this paper is as follows. Section 2 introduces an area-based action making process. Section 3 presents hexagon-based Q-learning adaptation. Section 4 introduces the design of a small mobile robot. Experimental results from the application of two different searching methods to find a target object are presented in Section 5. Section 6 presents conclusions.

2. AREA-BASED ACTION MAKING PROCESS: INSTINCTIVE INTELLIGENCE

2.1. Area-based action making process

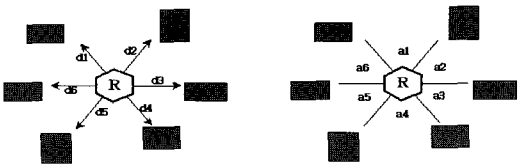
Both Distance-based action making (DBAM) and Area-based action making (ABAM) process are widely used for determining next action of a robot. In the DBAM process is referred to as DBAM, a robot can recognize its surroundings by the distance between itself and an obstacle. But, in the case of

Manuscript received October 11, 2006; revised May 1, 2007 and July 2, 2007; accepted July 30, 2007. Recommended by Editor Jae Weon Choi. This research was supported by the Brain Neuroinformatics Research Program by Ministry Commerce Industry and Energy, Korea.

Hyun-Chang Yang, Ho-Duck Kim, In-Hun Jang, and Kwee-Bo Sim are with the School of Electrical and Electronics Engineering, Chung-Ang University, 221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea (e-mails: {hcyang, hoduck, inhun}@wm.cau.ac.kr, kbsim@cau.ac.kr).

Han-Ul Yoon is with the School of Electrical and Electronics Engineering, the University of Illinois at Urbana Champaign, Urbana, IL 61801 USA (e-mail: huyoon@wm.cau.ac.kr).

* Corresponding author.



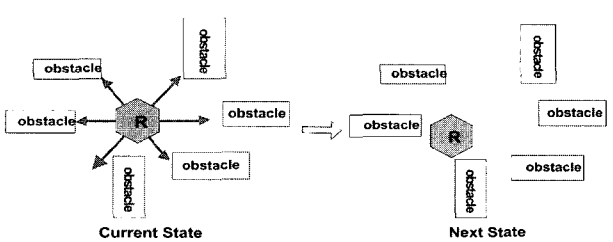
(a) DBAM. (b) ABAM.

Fig. 1. Different actions taken under DBAM and ABAM.

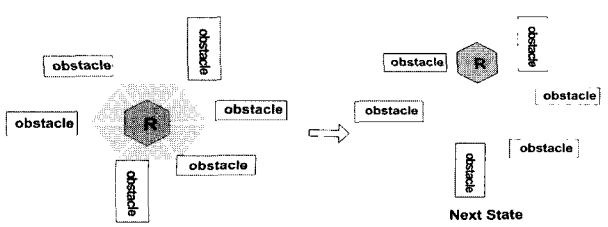
ABAM, a robot uses the circumferential areas for recognizing its surroundings. The key to the ABAM process is that it removes uncertainty regarding its surroundings. It is similar to the behavior-based direction change in regards to controlling robots [5,6]. Under the ABAM process robots recognize the shape of their surroundings and then take action, i.e., turn and move toward the widest guaranteed space. Fig. 1 depicts the different actions in the same situation under DBAM and ABAM, respectively [7,8]. As you can infer by their name, DBAM process selects d4 that is the direction of the longest distance from the robot. Otherwise, ABAM process selects a4 that has the widest area on the neighborhood.

2.2. The advantage of ABAM over DBAM

Fig. 2 illustrates how a robot can avoid both obstacles and collisions and estimate its tracking area. In Fig. 2, the robot is surrounded by 6-obstacles. Under DBAM, the robot perceives that there is no obstacle in the southwest direction. Thus, it will try to proceed toward that direction, which will result in being struck two obstacles. This scenario is shown in Fig. 2(a). Under ABAM, however, the robot calculates the distance between the two obstacle areas and choose the direction that has maximum distances for next movement. Therefore, the robot changes its direction accordingly. This scenario is shown in Fig. 2(b).



(a) DBAM.



(b) ABAM.

Fig. 2. Illustrative example of collision avoidance on DBAM and ABAM.

3. HEXAGON-BASED Q-LEARNING ALGORITHM

3.1. Q-learning algorithm

Q-learning is a well-known algorithm for reinforcement learning. It leads an agent to acquire optimal control strategies from delayed rewards, even when there is no prior knowledge of the effects of its actions on the environment [9,10]. The Q-learning algorithm presented in Table 1 is a possible state or action, indicates an immediate reward value, and is a discount factor. The formula to update the table entry value is:

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a'). \quad (1)$$

Fig. 3 explains the Q-learning algorithm more clearly. Each grid square represents possible states. 'R' stands for a robot or an agent. The values upon the arrows are values relevant to the state transition.

For example, the value $\hat{Q}(s_1, a_{right}) = 72$, a_{right} refers to the action that moves 'R' to its right [9,11].

If the robot takes action to move right, the value is updated, where $r = 0$, $\gamma = 0.9$ are predetermined values. The formula is as follows:

$$\begin{aligned} \hat{Q}(s_1, a_{right}) &\leftarrow r + \gamma \max_{a_2} \hat{Q}(s_2, a_2) \\ &\leftarrow 0 + 0.9 \max_{a_2} \{63, 81, 100\} \\ &\leftarrow 90. \end{aligned} \quad (2)$$

Table 1. Q-learning algorithm.

For each s,a initialize table entry $\hat{Q}(s,a)$
Zero Observe the current state s
Continue to infinity
• Select action a and execute
• Receive immediate reward r
• Observe new state s'
• Select action
• Update table entry for $\hat{Q}(s,a)$
• $s \leftarrow s'$

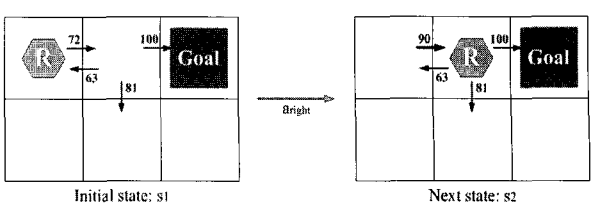


Fig. 3. Illustrative example of Q-learning.

3.2. Hexagon-based Q-learning adaptation

The unique Q-learning type for this robot system was adapted to enhance the ABAM process. The adaptation can be performed with a simple and easy modification, namely, through hexagon-based Q-learning.

Fig. 4 illustrates an example of hexagon-based Q-learning. The well-known standard Q-learning method is based on square state space. But hexagon-based Q-learning uses the different shape of the state space from the ordinary square-based state space.

The reason for changing the shape of state space from a square to a hexagon was that the hexagon is a polygon that can be expanded infinitely by its combination. According to this adaptation, the robot could perform an action in 6-directions and have 6-table entry $\hat{Q}$ values. Moreover, the hexagon-based Q-learning has extra advantages that it has fast responses and many radius of action. In Fig. 4, the robot is in the initial state. Now, if the robot decides that +60 degrees guarantees the widest space after calculating its 6-areas of surroundings, the action of the robot would be $a_{+60°}$. After the action is taken, if Area6' is the widest area, the value of $\hat{Q}(s_1, a_{+60°})$ can be updated using (1) and (2) as

$$\begin{aligned}\hat{Q}(s_1, a_{right}) &\leftarrow r + \gamma \max_{a_2} \hat{Q}(s_2, a'_\theta) \\ &\leftarrow 0 + 0.9 \max_{a_2} \{Area1', \cdots, Area6'\} \quad (3) \\ &\leftarrow \gamma \text{Area6'},\end{aligned}$$

where s is a possible state, a is a possible action, r indicates an immediate reward value, here predetermined as 0, and γ is the discount factor [12,13].

After moving from the initial state to the next state, immediate reward becomes the difference between the sum of the total area after action is taken and the sum of the total area before action is taken. Thus, the reward is

$$r = \sum_{j=1}^{6} Area_j - \sum_{i=1}^{6} Area_i, \quad (4)$$

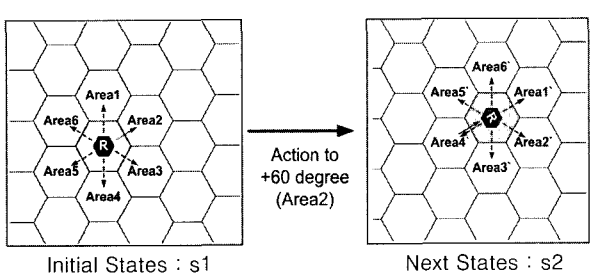


Fig. 4. Illustrative example of hexagon-based Q-learning.

Table 2. Hexagon-based Q-learning algorithm.

For each s, a initialize the table entry $\hat{Q}(s, a_\theta)$ zero calculate 6-areas at the current state s
Do until the task is completed
- Select action a_θ to the widest area and execute
- Receive immediate reward r
- Observe new state s' If $\hat{Q}(s', a'_\theta)$ is greater than or equal to $\hat{Q}(s, a_\theta)$
- Update table entry for $\hat{Q}(s, a_\theta)$
- $s \leftarrow s'$ Otherwise, if $\hat{Q}(s', a'_\theta)$ is far less than $\hat{Q}(s, a_\theta)$
- Move back to previous state
- $s \leftarrow s$

where $Area_i \in s$, and $Area_j \in s'$ respectively. The hexagon-based Q-learning algorithm is presented in Table 2.

4. DESIGN OF A SMALL MOBILE ROBOT

The robot system, designed in a laboratory, is consisted of two main micro-controller parts and three sub-parts. The sub-parts are camera vision, sensor, and motor. Each sub-part has its own controller in order to perform its unique function more efficiently. One main micro-controller part controls three sub-parts to avoid process collision and make decisions based on data from its sub-parts. The other main part controls the Bluetooth module and processes event handling [14]. Section 4.1 introduces the design and implementation of three subordinate parts. Two main parts employed in the hierarchical upper control layer are presented in Section 4.2.

4.1. Three subordinate parts design and implementation

1) Object recognition with TMS320LF2407A DSP controller: This robot used the *Movicam II* made by Kyosera, a CCD camera used by SKY cellular phone. The dimensions are 30×16.4×53.7mm(width×thickness ×height) and its weight is approximately 12g. The frame consists of a header, image data, and end maker. The data from the camera is only used to recognize object. Fig. 5 shows the camera and the data components of the frame in detail. When a clock is applied to the clock-port (port 2), it starts to slowly send the data rising clock from the header to the end maker. The data-out port (port 1) of the camera is attached to a DSP (Digital Signal Processor) TMS320LF2407A, which is programmed to perform signal processing. Image data for an entire image is too large to wait for the end of the process (153,600

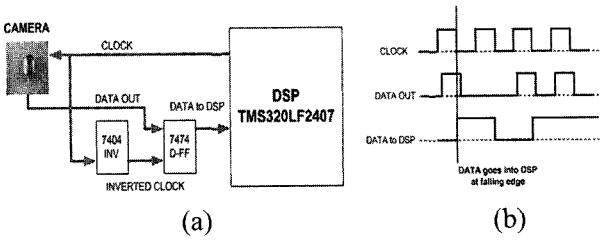


Fig. 5. Connection between camera and DSP (a), data transfer timing (b).

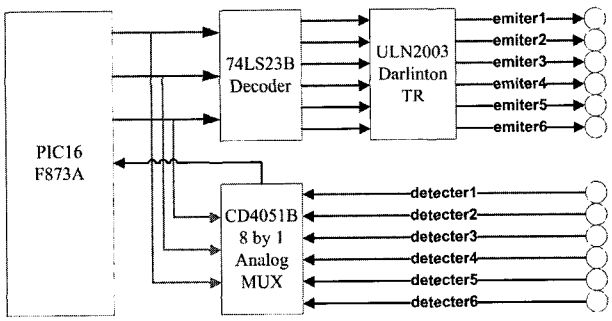


Fig. 6. Functional block diagram of sensor part.

byte). Accordingly, the program was optimized to reduce the image data to within 26,134 bytes. The connection between the DSP and the camera is shown in Fig. 5(a). The data transfer diagram is presented in Fig. 5(b).

2) Six-area calculation using infrared sensor emitter-detector pair: The robot has six infrared (IR) sensors to measure the distance between itself and its surroundings. Each sensor consists of a pair of emitters and detectors. The emitter is a Kodenshi EL-1kl3, high-power GaAs IR and the detector is an ST-1kla, high-sensitivity NPN silicon phototransistor. They are mounted in a durable and hermetically sealed TO-18 metal package.

The six IR sensors are placed at 60 degrees angles to one another so they can cover an entire 360 degrees. To make six IR sensors cover 360 degree, we just placed each IR sensor at 60 degrees angles to one another. But we chose each IR sensor with 17 degrees spec. to avoid interference. An emitter with a narrow beam angle (about 17 degree) is chosen to avoid interference. Fig. 6 illustrates the block diagram showing the arrangement and the area covered by the six sensors.

3) Maneuver driver with a NMB PG25L-024 stepping motor: NMB PG25L-024 stepping motor is used as a driving part. Its characteristics are the following: drive voltage-12V, drive method 2-2 phase and 0.495° step angle.

Fig. 7 illustrates a torque-frequency-current characteristic curve and maximum self-operating frequency of 600pps. The motor driver is comprised from the serial combination L297 and SLA7024A to control two motors. Table 3 presents the relationship

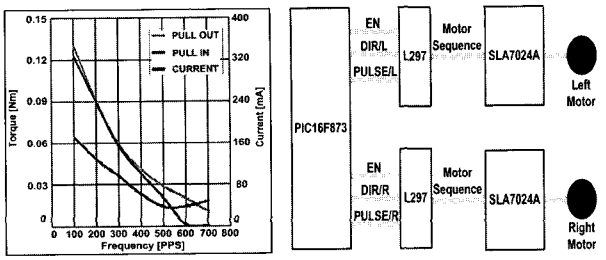


Fig. 7. Characteristic curves of motor (left), functional block diagram of driving part (right).

Table 3. Relationship between signals and actions.

• 0x01 → *Forward* (North)
• 0x02 → *Right 60°* (North East)
• 0x03 → *Right 120°* (South East)
• 0x04 → *Turn around* (South)
• 0x05 → *Left 120°* (South West)
• 0x06 → *Left 60°* (North West)

between the signals and the actions of robots.

4.2. Two main parts as a hierarchical upper control layer

1) First main controller for camera, sensors, and motors: The system is designed so that the main controller may have little over-head. The functions of the main controller are as follows: a) controls the UART Tx/Rx communication between the main controller and sub-controllers, b) generates the rules for following actions, c) changes the direction d) operates the camera. The abstract algorithm is summarized in Table 4.

2) Second main controller for Bluetooth communication: As the first goal is to send out multiple robots into an unknown area, a manual control technique is needed after the robot found the appropriate trajectory.

Table 4. Algorithm for moving the mobile robot.

• Initialization
• Wait until all three subordinate parts are ready.
Continue until the given task is complete
• Send the sensor a signal ID.
†Receive directions for the widest area.
• Store the current state and direction.
• Send the motor part the direction.
†Move to the next state.
• Send the vision part a signal ID.
†Receive the acquired image.
• Feed the image to direction and robot ID [base_ address].
†Produce ACL packet format data and send it to a desktop via Bluetooth communication.
• Run the hexagon-based Q-learning.

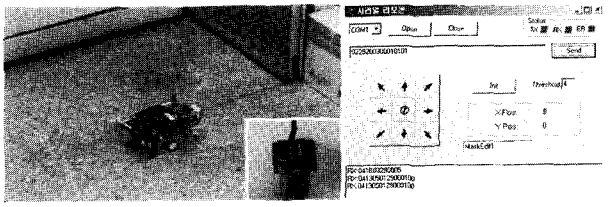


Fig. 8. Prototype robot for remote control test (left), Bluetooth host module (center bottom), and control panel (right).

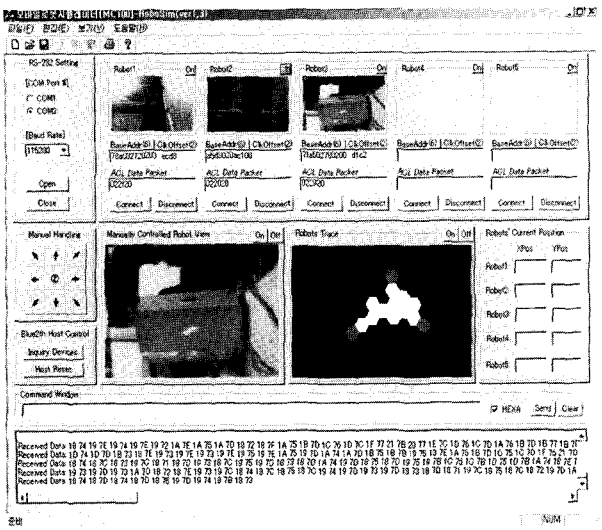


Fig. 9. ROBOSIM, URIS Laboratory mobile robot simulator.

Fig. 8 shows the prototype robot and the control panel that were used for the development of robot manual control via Bluetooth communication. The Bluetooth controller first sets up its connected client module as a Discoverable mode; then, classifies received data by its packet and stores it in the flash memory [15].

Fig. 9 shows the interface of a ROBOSIM simulator that can unify and play the role of a bridge to connect software simulations under intelligent algorithms, i.e., Hexagon-Based Q-Learning, GA (Genetic Algorithm), ANN(Artificial Neural Network), XCS(eXtended Classification System), and so on, with a physical hardware simulation. It can monitor the situation of each robot by displaying its sight, trace and position.

5. EXPERIMENTS WITH TWO DIFFERENT CONTROL METHODS

The task of the robots is a follows: "Find the hidden object while tracking through an unknown hallway." We set up the color of the object as green and that of 5-robots as orange. The object was a stationary robot having the same shape. It was a located at a hidden place near the obstacle. The 5-robots, which try to search the object, recognize the object by the object's color and shape. The 5-robots will decide whether they have finished the task by detecting the object after each action is taken.

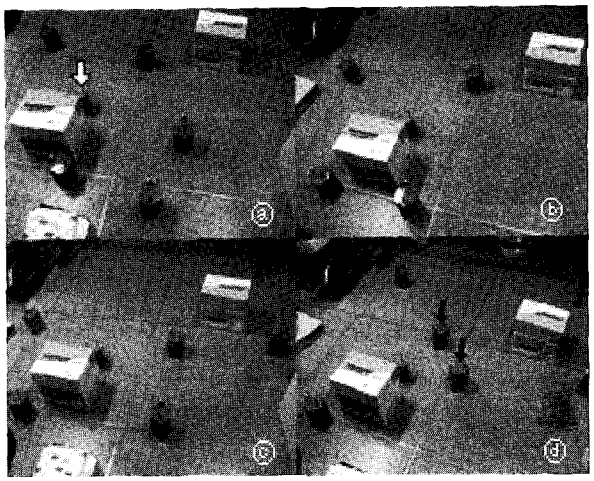


Fig. 10. Five robots searching for object using ABAM.

First, we applied ABAM to all robots. With the feature of ABAM, the robots sense their environment by 6-infrared sensors and calculate 6-area with these values. When the calculation is done, each robot tries to move to where the widest area will be guaranteed. In our 2nd experiment, after the robots started to move, each robot spread out into the environment. Consequently, the ABAM performed better than random search. Fig. 10 shows that the two robots, which are located the right side of the object, succeeded to complete the task. These two robots are designated by black arrow in Fig. 10.

Second, we adopted the hexagon-based Q-learning to ABAM as a modified control method. This method allowed the robots to reduce the probability of a wrong judgment and compensated wrong judgment by reinforcement learning. Each robot tried to search its own area as in the 2nd experiment, however, it canceled the decided action if the action caused negative (or bad) immediate reward value. By using the hexagon-based Q-learning adaptation to ABAM, more than 2 robots completed the task during the 10-trials. The search with hexagon-based Q-learning is presented in Fig. 11.

The results of our experiment are presented in Fig. 12. With random search, one robot found the object at the 2nd trial and 6th trial, although these detections can be considered as just coincidence. Therefore, we can say the random search has no remarkable meaning. With ABAM, the robots performed better than with random search, with the average performance above 1 during the all trial. Finally, with the adaptation of hexagon-based Q-learning to ABAM, the results were remarkable. Especially, 3-robots succeeded to find the object at the 4th, 6th, 8th, 9th, and 10th trial.

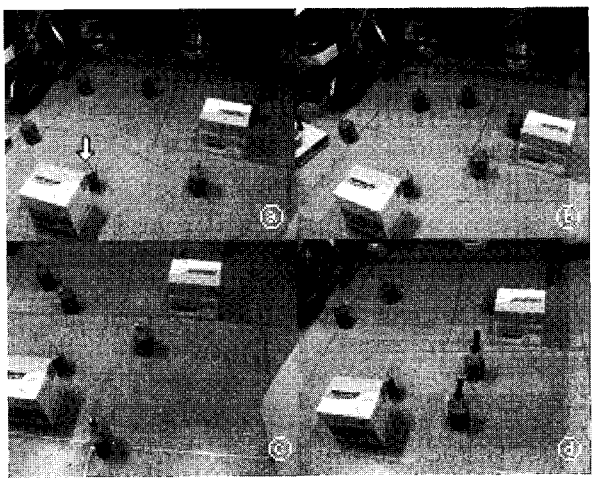


Fig. 11. Five robots searching for object using hexagon-based Q-learning.

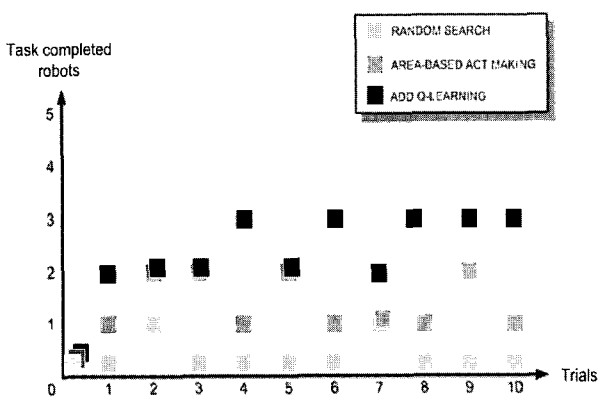


Fig. 12. Experimental result with two different control methods.

6. CONCLUSIONS

This paper introduced an area-based action making (ABAM) process and hexagon-based Q-learning. Five small mobile robots were used to search for a hidden object in an unknown location. The results are presented from the experimental application of two different control methods under the same conditions. The area-based action making process and hexagon-based Q-learning were new ways for robots to search for an object in an unknown space. This algorithm also enabled agents to avoid obstacles during their search. For future research, we first need to clarify the problem of accessing an object. In other words, if multiple robots are to carry out a task such as object transporting or block stacking, they need to recognize the object first and then proceed to approach it. Second, these robot systems require improvement so that the main parts and the subparts adhere more strongly. In addition, stronger complex algorithms, such as Bayesian learning or TD (λ) methods need to be adapted. Third, a self-organizing Bluetooth communication network should be built so that robots can dynamically communicate with one another even if one or more robots are lost. Finally, the total system unification using a ROBOSIM simulator needs to be refined to obtain better results and offer a stronger platform for mobile robot research.

REFERENCES

[1] L. Parker, "Adaptive action selection for cooperative agent teams," *Proc. of the 2nd International Conference on Simulation of Adaptive Behavior*, pp. 15-64, 1992.

[2] G. Ogasawara, T. Omata, and T. Sato, "Multiple movers using distributed, decision-theoretic control," *Proc. of Japan-USA Symposium on Flexible Automation*, vol. 1, pp. 623-630, 1992.

[3] D. Ballard, *An Introduction to Natural Computation*, The MIT Press, Cambridge, 1997.

[4] J. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, New Jersey, 1997.

[5] W. Ashley and T. Balch, "Value-based observation with robot teams (VBORT) using probabilistic techniques," *Proc. of International Conference on Advanced Robotics*, 2003.

[6] J. B. Park, B. H. Lee, and M. S. Kim, "Remote control of a mobile robot using distance-based reflective force," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3415-3420, 2003.

[7] D. Patterson and J. Hennessy, *Computer Organization and Design,* 3rd ed., Morgan-Kaufmann, Korea, 2005.

[8] T. Mitchell, *Machine Learning*, McGraw-Hill, Singapore, 1997.

[9] C. Clausen and H. Wechsler, "Quad Q-learning," *IEEE Trans. on Neural Network*, vol. 11, pp. 279-294, 2000.

[10] S. Russel and P. Norbig, *Artificial Intelligence: A Modern Approach*, 2nd ed., Prentice-Hall, New Jersey, 2003.

[11] H. U. Yoon and K. B. Sim, "Hexagon-based Q-learning for object search with multiple robots," *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 3612, pp. 713-722, 2005.

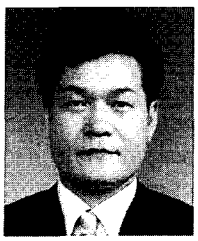
[12] H. U. Yoon and K. B. Sim, "Hexagon-based Q-learning to find a hidden target object," *Lecture Notes in Artificial Intelligence (LNAI)*, Springer-Verlag, vol. 3801, pp. 429-434, 2005.

[13] S. H. Whang, K. B. Sim, I. C. Jeong, *et al.*, "Design of efficient strategies for distributed multi-agent robot soccer system," *Proc. of the FIRA Robot Congress*, 2004.

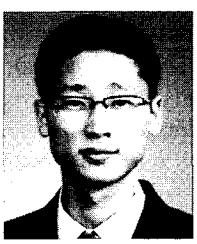
[14] Bluetooth Co., *Specification of the Bluetooth System*, vol. 1, pp. 537-828, 2001.

[15] H. U. Yoon, S. H. Hwang, D. W. Kim, D. H. Lee, and K. B. Sim, "Robotic agent design and application in the ubiquitous intelligent space,"

Journal of Control, Automation, and Systems Engineering (Korean), vol. 11, no. 12, pp. 1039-1044, 2005.

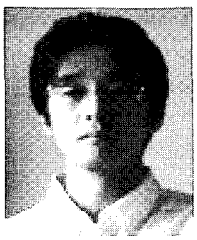


Hyun-Chang Yang received the M.S. degree from the Department of Industrial Engineering of Soongsil University, Korea in 2002. He is currently pursuing a Ph.D. degree at the Chung-Ang University. His research interests include intelligent robot, home network, smart home, ubiquitous sensor network, soft computing, etc.



Ho-Duck Kim received the B.S. degree from the School of Electrical and Electronic Engineering from Chung-Ang University, Korea, in 2005. He is currently pursuing a M.S. degree at the Chung-Ang University. His research fields are evolvable h/w, processor design, embedded linux system, genetic algorithm, DARS, etc.

Han-Ul Yoon received the B.S. and M.S. degrees from the School of Electrical and Electronic Engineering Chung-Ang University, Korea in 2004 and 2006, respectively. He is currently pursuing a Ph.D. degree at the University of Illinois in Urbana-Champaign. Research interests include evolvable h/w, processor design, genetic algorithm, DARS, etc.



In-Hun Jang received the B.S. and M.S degrees in the Department of Control and Instrumentation Engineering from Chung-Ang University, Seoul, Korea, in 1993 and 1999, respectively and he is currently pursuing a Ph.D. degree at the same university.

Kwee-Bo Sim received the B.S. and M.S. degrees from the Department of Electronic Engineering Chung-Ang University, Korea, in 1984 and 1986, respectively and the Ph.D. degree from the Department of Electrical Engineering at the University of Tokyo, Japan, in 1990. Since 1991, he has been a Faculty Member of the School of Electrical and Electronics Engineering, Chung-Ang University, where he is currently a Professor. His research fields are artificial life, emotion recognition, ubiquitous robots, intelligent systems, computational intelligence, intelligent home and home networks, ubiquitous computing and sense networks, adaptation and machine learning algorithms, neural networks, fuzzy systems, evolutionary computation, multi-agent distributed autonomous robotic systems, artificial immune systems, evolvable hardware, and embedded systems etc. He is a Member of the IEEE, SICE, RSJ, KITE, KIEE, KFIS, and an ICASE Fellow. He is currently President of the KFIS.