Search and Rescue Algorithm using a Cooperative Robot System

Jung Kyu Park, Howard Park, Eun Young Park

Abstract*: Cooperative robotics is very different from working on a single robot before. Cooperative robots can perform tasks that were previously impossible with a single robot. This robot research can be applied in computer science, artificial intelligence, and electrical engineering. This research aims to deal with tasks that are difficult or impossible to perform with a single robot. For this purpose, we propose a Se-Re (Search-and-Rescue) algorithm. In addition, the proposed algorithm is designed to allow multiple robots to collaborate in the area of work. Robots use wireless communication to collaborate and are organized in groups, which create environmental maps and share data with each other. The robot uses the shared area map to create a global map. Using this information, the robot does not revisit the previously visited area. The proposed algorithm can be applied both indoors and outdoors. Indoors can be used in general buildings as well as livestock housing, and outdoors can be used for landmine removal, human search, and rescue.

***Keywords* : Cooperative Robot, IoT, Rescue, Search**

I. INTRODUCTION

In the United States in 2001, a 911 attacks occurred, causing a number of casualties. Since the terrorist attacks, mobile robots were first used to search for lives in the remains of buildings [1]. The autonomous mobile robots used in the search were not practical. But rescue workers helped find more than 2% of the victims. This robot is sent in a small gap that a person cannot move and uses sensors such as cameras to examine the buried room. In addition, the measured information is sent to the rescue worker and used in the rescue. This is an epoch-making event in the robot-based rescue field, however more research and development are urgently needed. Completely autonomous and collaborative robots are research projects that must be overcome by robot researchers all over the world [2].

In general, robots used for rescue work are operated by humans or one autonomous robot independently. The rescue robot used at the World Trade Center in 2001 was remotely controlled by people and wired cables were connected to it. The camera image was also transmitted via a wired cable [3]. If autonomous robots are sent alone or uncooperatively, there is a greater risk of missed missions. The most necessary for rescue work in robot research is a robot that operates autonomously without human help. In addition, collaborative robot technology is essential to increase work efficiency and reliability [4].

In this paper, we propose a rescue work algorithm Se-Re (Search-and-Rescue) that performs search and save task. This algorithm is much more efficient than using a single robot by organizing and processing a group of fully automated robots to perform a single task. The robot performs its first environment task and performs an environmental search. In this state, the robots examine the environment in parallel. Robots perform tasks in parallel at different locations. In addition, data sharing between robots does not visit and process the same area again. The first robot that finds an object while performing a task transfers information to other robots working on it. After receiving this information, the robots switch from searching to rescue. The robots move around and surround the target. In order to collide the robot when working in a complicated area, the search and rescue work was set as complete when the distance between the robot and the object was within 1.5~2m.

As mentioned earlier, a group needs a communication algorithm to perform the same task. In addition, location recognition, path planning, and distinction between robots are required. The Se-Re algorithm shares the local maps generated by robots between robots. This local map provides functions such as path planning and movement to perform search and rescue task.

The Se-Re starts with a recognition of the robot's relative position in order to generate a map. This assumption is used in many robotic studies and can be equally applied to robots performing search and rescue operations. Unlike previous studies, Se-Re algorithms provide scalability and robustness. The meaning of robust here is that there is no problem in the algorithm due to robot loss or communication degradation. Each robot uses the same program and can perform the task alone even if team members or communication fail. However, if one of the robots collaborating has a problem, the robot can collaborate. The Se-Re solves this problem by providing the scalability to use different numbers of robots when performing certain missions.

The remainder of this paper is organized as follows. Section II presents the Se-Re algorithm that performs search-and-save task. Finally, Section III shows conclusions, and suggestions for future work.

Revised Manuscript Received on February 5, 2020.
***** Correspondence Author
Jung Kyu Park, Dept. of Computer Software Engineering, Changshin University, Changwon-si, Korea. Email: smartjkpark@gmail.com
Howard Park, Yongsan International School of Seoul, Seoul, Korea. Email: howardpark03@gmail.com
Eun Young Park*, Department of Biomedical Laboratory Science, Shinhan University, Uijeongbu-si, Korea. Email: 71eypark@gmail.com

II. RELATED WORKS

A. Behavior-based robotics

Research on behavior-based robot technology has been around for a long time. Matarić classified behavior-based robotics into two types, depending on the nature of their behavior [5]. The two forms are divided into temporal combinations and direct combinations. The approach included in the time combination category is performed at a specific time. After work, the previous method is changed to a more aggressive one. For example, it is not an aggressive approach for a sensor to show different results depending on the approach. The direct combination category approach runs multiple actions simultaneously in a strict sub-sum style. This method sums up using a mathematical function to increase the accuracy of the lower sum.

Kube's research summarizes four of the actions required for a box-organizing robot [6]. The four moves are: finding objects, moving targets, avoiding obstacles, and moving slowly. Previously, robots worked together to bite and orient things. However, the proposed four motions are not a cooperative method but a robot alone. The object search operation should not collide with other robots while ensuring the current robot's operation. Slow movement means moving the target box without breaking it when moving the target box. Target movement means keeping the path so that the robot can move the box exactly to the destination. Four behaviors are controlled by signals in the environment in which the robot operates.

B. Exploration and Mapping

In self-driving robots, environmental navigation is a very important function. In order to perform a specific task with mobile robot, navigation is essential and time-consuming part of the task. In most cases, the navigation time of the robot is very important (for example rescue robot). For this reason, it is necessary to reduce the search time by using several robots simultaneously when using one robot [7, 8]. In most cases it is assumed that adding additional robots to the current search task will save time. However, if there is no collaboration function between robots, it is no different from working with one robot because all robots move on the same path and perform the same task. Also, in some cases, collisions between robots can take longer than working with one robot. Therefore, if multiple robots are used to perform the search, it is necessary to perform the procurement so that the search areas do not overlap.

Yamauchi proposed a frontier-based exploration technique for multi-robot navigation and map generation [9]. This technique assumes that there is no environmental information in advance as a global optimization technique. The study uses occupied grid maps to represent the environment. The occupied grid map consists of each cell and the cell has a probability value. According to the probability value of the cell, the occupancy of the object, the opening without the object, and the unknown unknown are expressed. Here, the cell adjacent to the unknown area is defined as the frontier cell. The robot must move to the frontier cell to explore new areas using the grid map. At this point, the robot moves to an open cell that is free to move. In the frontier-based navigation algorithm, the robot moves to a place where many frontier cells are gathered in order to improve the exploration efficiency.

III. SE-RE (SEARCH AND RESCUE) ALGORITHM

A. Structure of Se-Re

The Se-Re algorithm can search for a target in a work area where the young robots are the same. While performing a search, wireless communication is used to share surrounding environment information and target information. Each robot operates with the same program but provides fault tolerance for performing common tasks even when a particular robot is not running. Se-Re is built as standard for multiple collaborative robots but can also be used when working with a single robot.

The C/C ++ language is used to implement the Se-Re algorithm. We also used the behavior-based robot technology mentioned in Section 2 and used the same behavior for various purposes. To implement this method, it consists of a single loop and the robot has a value representing the current search or rescue state. Fig. 1 shows the overall structure of the Se-Re algorithm. In Fig. 1, the gray shape represents the robot's motion, and the dark gray color represents the end of the robot's mission. The dark gray boxes in bold type indicate the acquisition and transmission of information through the sensor. In the overall structure of the Se-Re algorithm, detailed functions have been deleted.

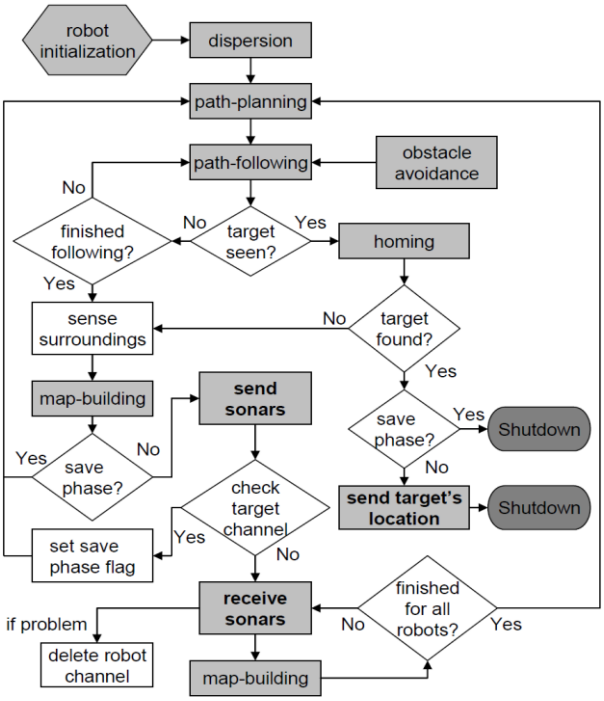


Fig. 1.Structure of Se-Re algorithm

When executing a program with Se-Re algorithm, various options are prepared for fine function control. By specifying this option value, you can operate the robot without recompiling the code and passing it to the robot.

Table 1 describes the options used in the program and shows the default values.

B. Obstacle Avoidance

Obstacle avoidance is a key part of autonomous driving. The implementation of this task is very complex and consists of control and integrated control of the sensor. However, the task of acquiring and calculating sensor data is frequently called, which proves to be too much of a computational burden on the system. The algorithm was implemented to consider accuracy and complexity while implementing and repeating the test process. The newly created simplified behavior has inaccuracies, but the inaccuracy could be reduced by using multiple behavior combinations. It's better to combine simple results and combine the results than to do something very complex but accurate. Inaccuracies occurring in simple operations can be compensated for by other operations.

Parameter	Description	Initial value
-i <address>	ip address of Player server	'local host'
-p <port>	port number of Player server	6655
-a <value>	area of operating environment (m^2)	25 m2
-x <position>	x coordinate in the operating environment (m)	0 m
-y <position >	y coordinate in the operating environment (m)	0 m
-r <resolution>	resolution of global map (m)	0.4 m
-s <speed>	speed of robot moving (m/sec)	0.6 m/s
-d <distance>	Minimum distance between robot and obstacle for obstacle avoidance (m)	0.6 m

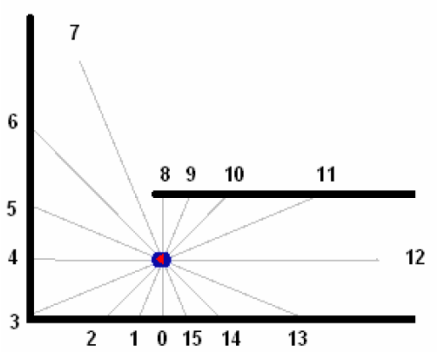


Fig. 2.Traveling from right to left with sonar sensors

Each time an obstacle avoidance operation is called on the robot, the distance values from the front (0 to 8 in Fig. 2) are collected using sensors mounted in front of the robot. Check that the sonar predecessor was used to measure the distance value and that the distance is the minimum distance (default 0.6m and passed as an option when running the program). If there is no value less than the minimum value in this range, it returns the "full speed ahead" command. If any sensor value is smaller than the minimum distance, calculate the sum of the robot's left front distance (from 0 to 3 in Fig. 2) and the right front distance (from 5 to 8 in Fig. 2). The smallest of the calculated sums is identified, and the command is rotated to move the robot in the opposite direction.

Fig. 2 illustrates the robot moving. The robot is in the middle of the hallway and simultaneously performs obstacle avoidance while moving from left to right. Or if the moving area is wide, move up and down repeatedly. As the robot moves and approaches the left corner, the sensor's distance value is smaller on the left front (from 0 to 3 in Figure 2) than on the right front. Some of the sensors on the right front of the robot (numbers 5 to 8 in Fig. 2) also get smaller, but we compare the sum of the sensors for error and accuracy. If the sum of sensor readings on the front left side of the robot is less than the minimum moving range, it cannot move to the left. In this case, turn to the right to move to the new area. If there is no space to move to the right and the sum of the sensors on the front left side is the same as the sum of the sensors on the front right side, the default value moves to the left. Occasionally, deadlocks occur where the robot is trapped in an unmovable space. In this case, the robot performs a backup operation.

C. Mapping

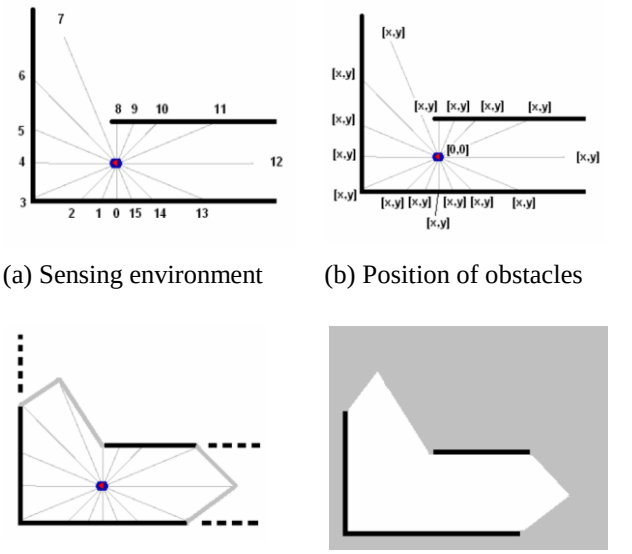
Mapping is the most important part of Se-Re. The robot measures the sensor value of the current location and uses that value to create a local map. Local maps generated by individual robots are shared by other robots. Integrate into a global map that displays the entire area based on the received local map. The generated global map is delivered to all robots so that all robots use the same global map. Robots can use the global map to efficiently search without having to revisit the area they once visited. If one of the robots finds the target during the search using the global map, the other robots will move to the target location.

To test the Se-Re algorithm, a total of 16 sonar sensors were installed in front of the robot and eight behind the robot. These sensors were used to identify obstacles in the environment around the robot. Currently, a large number of sonar sensors are used, but one leisure distance scanner can be used to identify obstacles in an area of 180 degrees. However, laser distance scanners range from $1500 to $ 2000, 800 times more expensive than sonar sensors, which cost about $2 each. Using sonar sensors in an indoor environment can cause specular reflection problems.

Specular reflection is a problem in which the sonar of the sonar sensor, such as mirror reflection, is reflected off an object and does not enter the sensor. This problem can be solved simply by measuring the distance with the sonar sensor and then discarding the maximum value that the sonar sensor can handle. In this study, several sensors were bundled and placed forward or downward to measure the vertical angle of incidence of the sensors [10, 11]. However, these methods can detect in three-dimensional planes, but the problem of horizontal specular reflection is not solved. Alternatively, the laser distance scanner and sonar sensor can be used simultaneously to reduce specular reflection [12]. However, using a laser distance scanner and sonar sensors at the same time can be expensive to manufacture robot hardware.

Fig. 3 shows how the robot uses a sonar sensor to create a local map. As shown in Fig. 3(a), the robot is equipped with 16 sonar sensors in its body to move around and acquire environmental information. The acquired environmental information is distance information from obstacles measured using sonar sensors. As shown in Fig. 3(b), the distance value measured based on the position of the robot is converted to (x, y) coordinates by trigonometric calculation.

The transformed coordinates represent the obstacles around them and are usually represented by lines. At this time, *draw-line()* is used to tie the empty space between sonar sensors. If two adjacent sonar sensor values are smaller than the specified distance (predefined to 1m), a line is drawn between the coordinates in consideration of the obstacle. In addition, if two adjacent sonar sensor values are greater than the specified distance value or exceed the measurement range of the center, it is considered as an unknown area and a line is drawn between the coordinates. Fig. 3(c), the black lines represent obstacles and the gray lines represent unknown areas. The black line around the robot is an unobstructed area so the robot can move freely. Outside the gray line is an unknown area where the robot searches. Fig. 3(d) shows the completed local map.



(a) Sensing environment (b) Position of obstacles

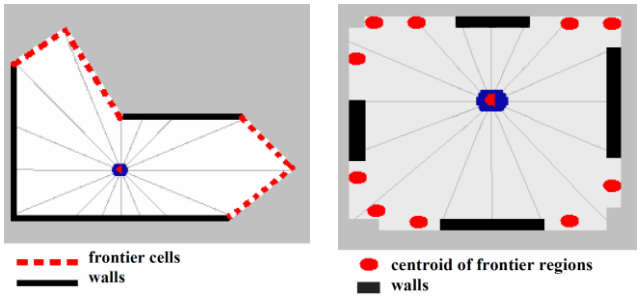
(c) Outline of environment (d) Local grid map

Fig. 3.Traveling from right to left with sonar sensors

D. Path Planning

After creating a local map, the robot shares it with other robots and creates a global map based on it. The generated global map is also shared between robots and the navigation is performed based on the map. To save time in searching, robots need to explore for new areas that haven't visited before. In this study, we used the frontier cell technique to perform the search based on the grid map. The robot moves to the frontier cell to search for a new area and performs the search. Fig. 4(a), the black line represents the wall and the red dotted line represents the frontier cell.

Global maps consist of occupied grid maps, and resolution and size may vary depending on the surrounding environment. In addition, the contents of the grid map may vary according to the environment information, and the frontier cell may increase rapidly. However, if the number of frontier cells is large, the robot does not have to explore through all the cells. In this case, if the robot is larger than the moving area (depending on the size of the robot), the exploration is performed. In previous studies, the frontier domain was defined as a group of frontier cells that the robot could process [7]. Fig. 4(b) shows the frontier area that the robot finds in one area. Allowing the robot to navigate only the frontier area prevents the robot from locking up in a specific area.



(a) Grid map with frontier cells (b) Grid map with frontier regions

Fig. 4.Grid map with frontier cells and regions

IV. EXPERIMENTS

In this study, player/stage simulation software is used to evaluate the proposed algorithm [x]. Stage provides environment with two-dimensional bitmap and can operate virtual robot applying algorithm in this environment. At this time, the virtual robot can use various kinds of robots from pioneer robots used for research. In addition, various sensors such as ultrasonic sensors, vision camera sensors, and laser distance scanners can be applied to the robot. Algorithms developed in the simulation environment can be directly applied to the hardware that the player operates, so that the robot software can proceed quickly. Fig. 5 shows s Plyer/Stage simulation software.

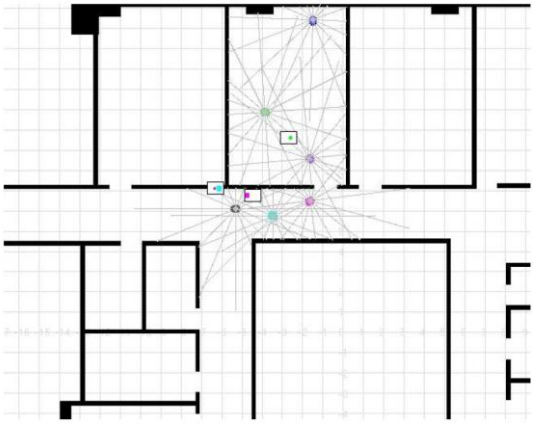
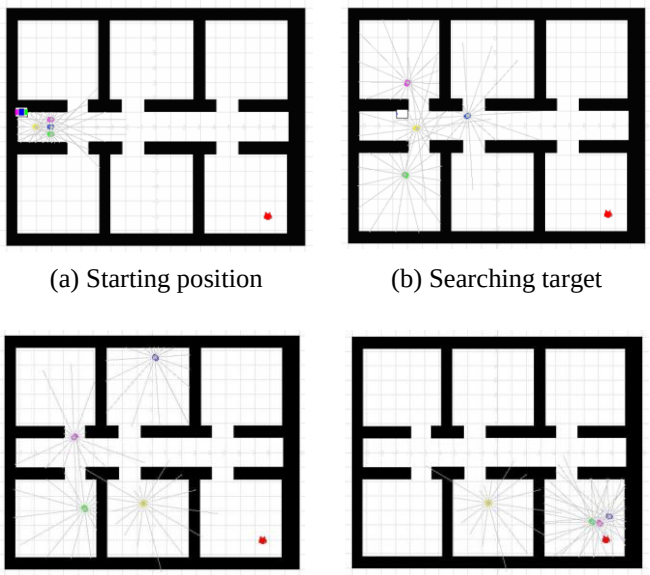


Fig. 5.Player/Stage simulation environment

In 75 simulation experiments, the communication error between robots was about 5% or less. Although it is a simulation, a communication error occurred depending on the distance and obstacles of the robot. We have assumed that a communication error occurs when doing work with multiple robots. That is, one of the robots performs one common task, and one robot is set to end suddenly.

The experimental results are shown in Fig 6. In the left position in Fig. 6(a), four robots start searching for a target. In Fig. 6(b), four robots are searching in their respective positions. The yellow robot at the bottom center of Fig. 6(c) is shut down. In this state, three other robots, except for the yellow robot, wait for data transmission from the yellow robot. If there is no data transmission of the yellow robot for a predefined period, the yellow robot is removed from the collaboration group and the work continues. In Fig. 6(d), three robots in the lower right corner have found the target and the yellow robot remains in the shutdown state.



(a) Starting position (b) Searching target

(c) Shutdown yellow robot (d) Serch completed

Fig. 6.Target sesrching using 4 robots with Se-Re algoritim

V.CONCLUSIONS

Previously, research was conducted to assume search and rescue operations assuming a robot. In this paper, we proposed an algorithm that not only one robot but also several robots can perform search/rescue task through the setup. The proposed Se-Re (Search-and-Rescue) algorithm supports multiple robots to provide scalability and robustness. Multiple robots share the local map and target information during the search and rescue. In addition, even if a problem occurs in one robot in the work, the problem does not occur in the whole work.

In this paper, various options are added to evaluate the performance of the proposed algorithm. The options used are the number of robots, the complexity of the operating environment, and the communication cycle between the robots. The number of robots in each set of experiments varied, and we found that there are several robots that are optimal for minimizing work completion time. When performing a roll-up operation using multiple robots, communication load may occur, but this is a characteristic of multi-robot operation.

ACKNOWLEDGMENT

This work was supported by Changshin University Research Fund of 2019-86).

REFERENCES

1. C. Xin, D. Qiao, S. Hongjie, L. Chunhe, and Z. Haikuan, “Design and Implementation of Debris Search and Rescue Robot System Based on Internet of Things,” in Proc. of 2018 Inter. Conf. on Smart Grid and Electrical Automation (ICSGEA), 2018 Jun. pp. 303-307.
2. R. Käslin, and et al, “Collaborative localization of aerial and ground robots through elevation maps,” in Proc. of the IEEE Inter. Symp. on Safety, Security, and Rescue Robotics (SSRR), 2016 Oct. pp. 284-290.
3. J. Casper and R. R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center,” IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 22, no 3, May 2003, pp. 367-385.
4. S. Okano, N. Matsuhira, E. Shimokawara, T. Yamaguchi and M. Narita, “Employing Robots in a Museum Environment: Design and Implementation of Collaborative Robot Network,” in Proc. of 16th Inter. Conf. on Ubiquitous Robots (UR), 2019 July, pp. 224-227.
5. M. J. Matarić, “Issues and approaches in the design of collective autonomous agents,” Robotics and Autonomous Systems, vol. 16, no. 2-4, 1995 Dec. pp 321-331.
6. C. R. Kube, H. Zhang, and X. Wang, “Controlling collective tasks with an ALN,” in Proc. of IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS ’93), 1993 Jul. pp. 289-293.
7. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration,” in Proc. of the IEEE Inter. Conf. on Robotics and Automation (ICRA), 2000 Apr. pp. 476-481.
8. W. Tian, “The research into methods of map building and path planning on mobile robots,” in Proc. of IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC), 2017 Dec. pp. 1087-1090.
9. B. Yamauchi, “Frontier-based exploration using multiple robots,” in Proc. of the Second Inter. Conf. on Autonomous Agents, 1998 May. pp. 47-53.
10. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, “Human Interaction With Robot Swarms: A Survey,” IEEE Trans. on Human-Machine Systems, 2015 Oct. vol. 46, no. 1, pp. 9-26.
11. T. Machado, T. Malheiro, S. Monteiro, W. Erlhagen, and E. Bicho, “Multi-constrained joint transportation tasks by teams of autonomous mobile robots using a dynamical systems approach,” in Proc. of the IEEE Inter. Conf. on Robotics and Automation (ICRA), 2016 May. pp. 3111--3117.
12. J. Kim, and W. Chung, "Localization of a Mobile Robot Using a Laser Range Finder in a Glass-Walled Environment," IEEE Trans. on Industrial Electronics, 2016 Jun. vol. 63, no. 6, pp. 3616-3627.

AUTHORS PROFILE

Jung Kyu Park, Assistant Professor, Department of Computer Software Engineering, Changshin University, Korea

Howard Park, Yongsan International School of Seoul, Seoul, Korea

Eun Young Park, Research Professor, Department of Biomedical Laboratory Science, Shinhan University, Korea