

Forest Lodge Room Navigation Algorithm Based on Beacon

Daesik Ko¹ and Jonghyun Choi²

^{1,2}Mokwon University, Korea
Kds@mokwon.ac.kr
bellstringc@naver.com

Abstract

As the social backbone develops, large cities and complex roads are increasing, and the need for GPS navigation is increasing. However, current GPS navigation is difficult to use in recreational forests. And there are few researches to solve this problem. In this paper, we propose the user's location confirmation and room guidance system through the communication between the beacon running on Bluetooth 4.0 and the user smart phone and main server. The proposed room guidance system includes an algorithm for searching the route based on the direction information to guide simple and fast the user to the room. It is shown that the performance of the proposed direction information based route search algorithm is better than that of DFS / BFS, which are suitable for recreation forest map and the basis of route search algorithm.

Keywords: Beacon, Recreational forest, Path finding Algorithm, IoT network

1. Introduction

Congratulations Most recreational forests utilize guide maps and signposts for room guidance. However, since recreational forests are unlike general environments, it is difficult to use the legacy maps inside recreational forests. There is also a possibility that people might lose their way and get lost because recreational forest maps are usually very different from ordinary maps.

The navigation system that many people are familiar with and frequently used works through GPS [1, 2, 3]. It receives location information from the GPS satellite that includes information such as the longitude, latitude, and height. By interconnecting this information with the electronic map stored in the navigation, the current location is displayed. The software built in the navigation uses the current location information to find the path the user wants. The problem is that there are cases where these GPS-using navigation systems operate abnormally [4].

In order to provide a solution to this problem, studies are being conducted on ways to track location without the use of GPS [5, 6, 7]. Among these solutions, the technology using beacons can also be used in areas that cannot receive GPS signals [8, 9]. The user can conveniently identify his or her location by adjusting coordinates on a general image map without having to use an electronic map. Therefore, in this paper, we propose a beacon-based room guidance system and the applied algorithm.

2. Beacon-based Room Guidance Algorithm

2.1. Beacon-based Recreational Forest Room Guidance System

The room guidance system operating on a smartphone guides user to his room. In order to track user location and to provide path guidance, the user smart phone needs information from the beacon and server installed inside the recreational forest. The

Received (August 9, 2017), Review Result (November 9, 2017), Accepted (November 13, 2017)

communication structure between the beacon, user smart phone, and main server can be represented as Figure 1 below. The main server provides the map information regarding the recreational forest. This map information includes not only a simple map image, but also the ID and location of the installed beacon, and the path information. On the other hand, the beacon operates on Bluetooth 4.0(BLE 4.0), and is installed all around the paths inside the recreational forest that the users use, periodically informing it's ID.

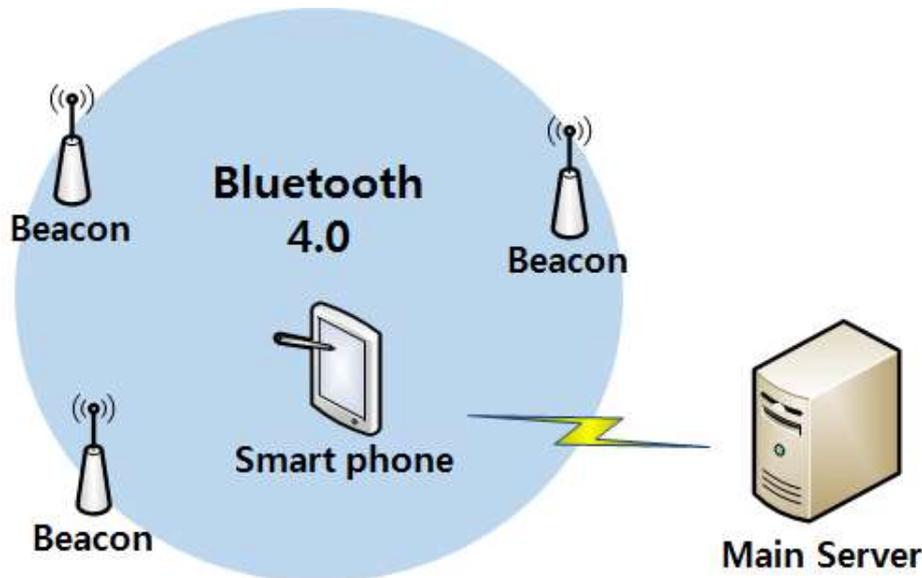


Figure 1. System Network

The current designed room guidance system performs two main functions. The first main function is location identification, and users are guided to their room based on the information obtained from this function.

The location identification tracks the user's current location based on the beacon ID. This function can operate concurrently or separately with the route calculation and guidance, and can be operated at all times. If the user's location information is periodically transmitted to the main server, the recreational forest manager can identify users' current location and route.

The room guidance presents the route to the destination (room) using the user's current location obtained from the location identification and map information.

2.2. Recreational Forest Room arrangement and Road Status

The room guidance system requires a route calculation algorithm. The algorithm must be able to guide the path to the user's room inside the recreational forest, and should be able to provide the shortest route guidance with actual existing paths. In Korea, there are about 40 national recreational forests managed by the Forest Service, and each recreational forest vary in the number of rooms, room location, road location, *etc.*, [10].

Figure 2 shows *the National Jiri-Mountain natural recreation forest* layout schematized into the form of a graph, in order to run it through the route guidance algorithm. It is displaying only important node.

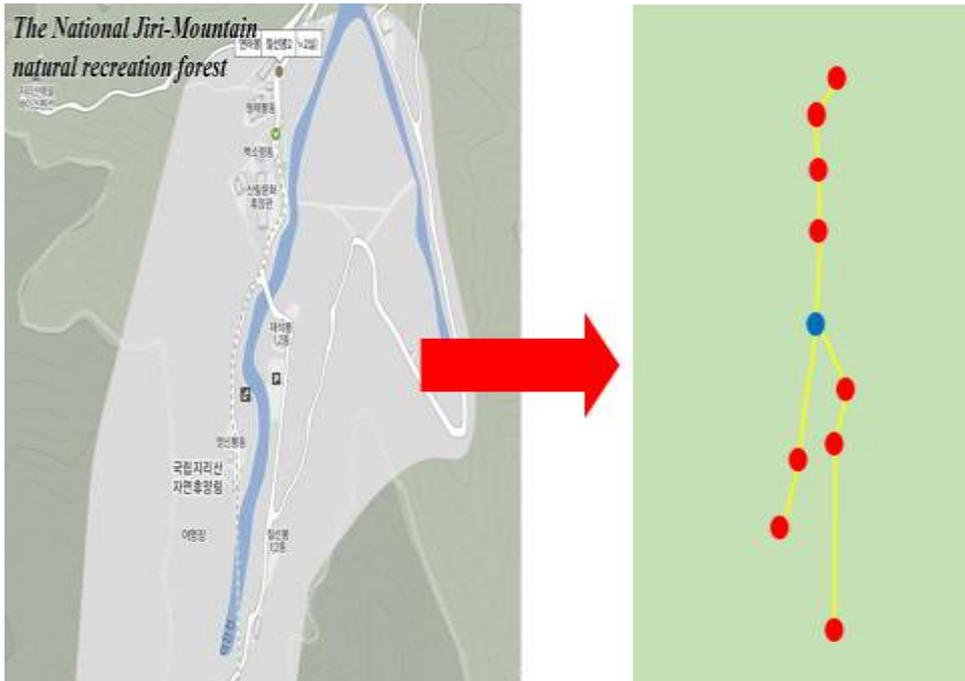


Figure 2. Schematized the National Jiri-Mountain Natural Recreation Forest

The recreation forest can be displayed in a relatively simple structure by display of the destination and significant roads that user require. As another example, the graphical representation of *the National Deogyu-Mountain natural recreation forest* is shown in Figure 3, in which the layout relatively seems more complex. Likewise, once you schematize the facility layout, you can see that this is also produced into a relatively simple structure.

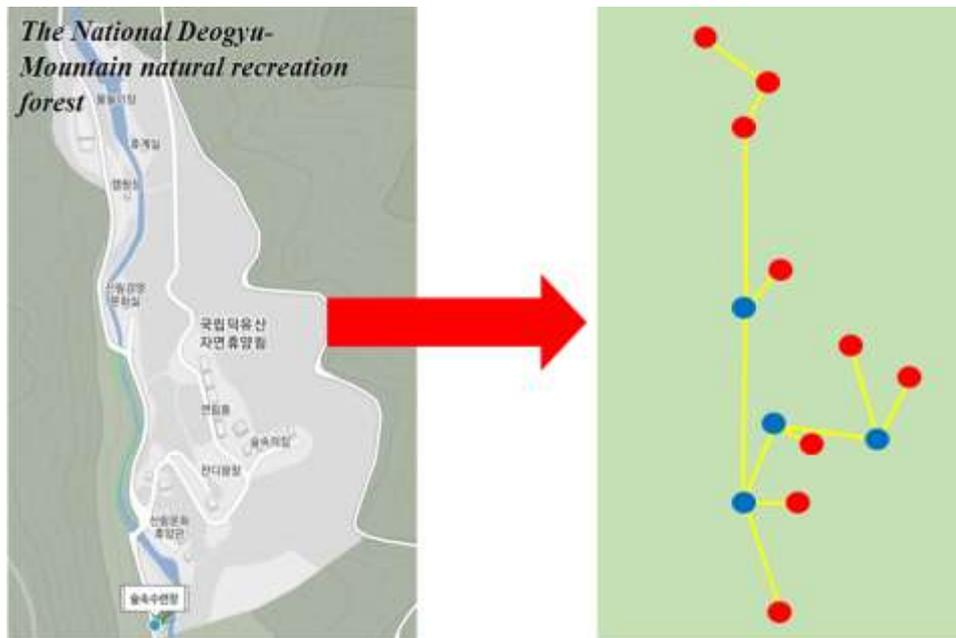


Figure 3. Schematized the National Deogyu-Mountain Natural Recreation Forest

2.3. Comparison and Analysis of Legacy Route Search Algorithms

As described above, although the current existing recreational forests in Korea have various arrangements, the recreational forests can be presented in relatively uncomplicated forms once they are schematized into the form of a graph. We analyze the features of route calculation algorithms that could be used appropriately in recreational forests of such scale.

In route calculation algorithms, there are the most basic algorithms such as the DFS (Depth-First Search) and BFS (Breadth-first Search)[11, 12, 13], and there are also the Dijkstra or A* that searches the most optimal route through various weights or other factors[14]. However, since the proposed room guidance system currently is not in need of weights, the Dijkstra or A* algorithm will not be used in this system, because the performance of these algorithms are determined by the weights and route characteristics. In this paper, we compare and analyze the most basic algorithms BFS/DFS with the presented algorithm, and propose a much advanced algorithm.

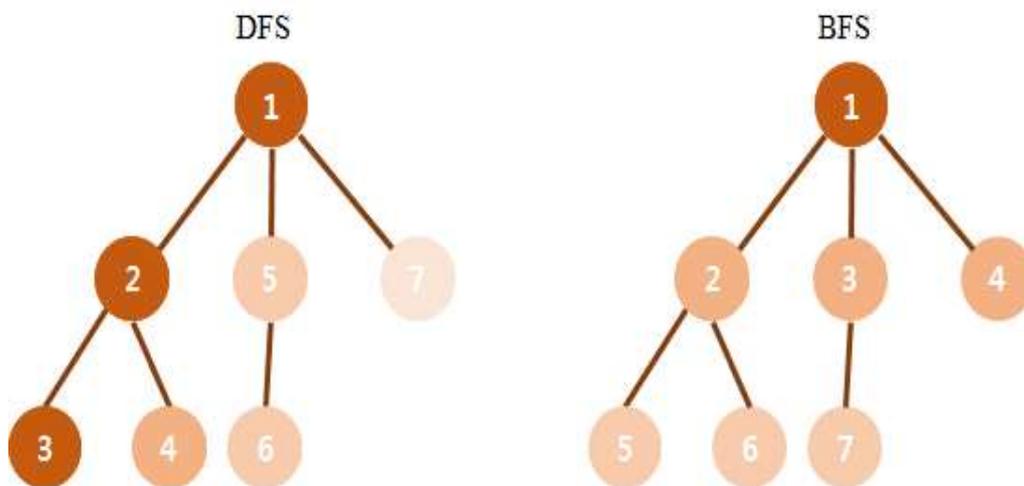


Figure 4. DFS / BFS

DFS first completes search in one direction, and when the destination is not there, it goes back to the intersection and performs search in the other direction. BFS, on the other hand, looks around all the neighboring nodes. With the use of weight or counting hop, it is possible find the shortest route.

The operations of DFS and BFS are roughly shown in Figure 4. First, DFS performs node search until it reaches the end of the route. If the destination node is not found, once again it performs search on the next route until it reaches the end. According to the figure, search is performed in 1-2-3 order, and search is performed on 4 when 3 is not the destination. If 4 is not the destination node, the search goes back to 1 and restarts in 1-5-6 order. If 6 is also not the destination, it once again goes back to 1 and performs search on 7. If 7 is the destination node, this creates a problem that search needs to be performed on all the nodes in the graph.

BFS starts search from the neighboring nodes, and if the destination is not found, it moves on to the next level of neighboring nodes. Starting at 1, the search is performed on the connected 2, 3, 4 node. If the destination node is not among these nodes, node 5, 6 connected to 2 and node 7 connected to 3 are searched. The worst case in BFS is when the destination node has a lot of parent nodes.

2.4. Proposed Direction Information Algorithm

In this proposed algorithm, the nodes located at intersections have the destination information of all the destinations, and the route is determined according to this information. All the nodes have all the information of the destination node, and they are classified by the direction. In other words, if the search is performed in a specific direction, we can know which destination node will eventually be reached.

In Figure 5, for example, P01 beacon has the route for three directions, and they are represented as direction01, direction02, direction03. It also has information on which destinations it can reach for each directions. You can see that destination D1 exists in direction 1, D2, D3, D4 exists in direction 2, and non-exists in direction 3. P01, P02, P03, P04 located in intersections, all contain similar information.

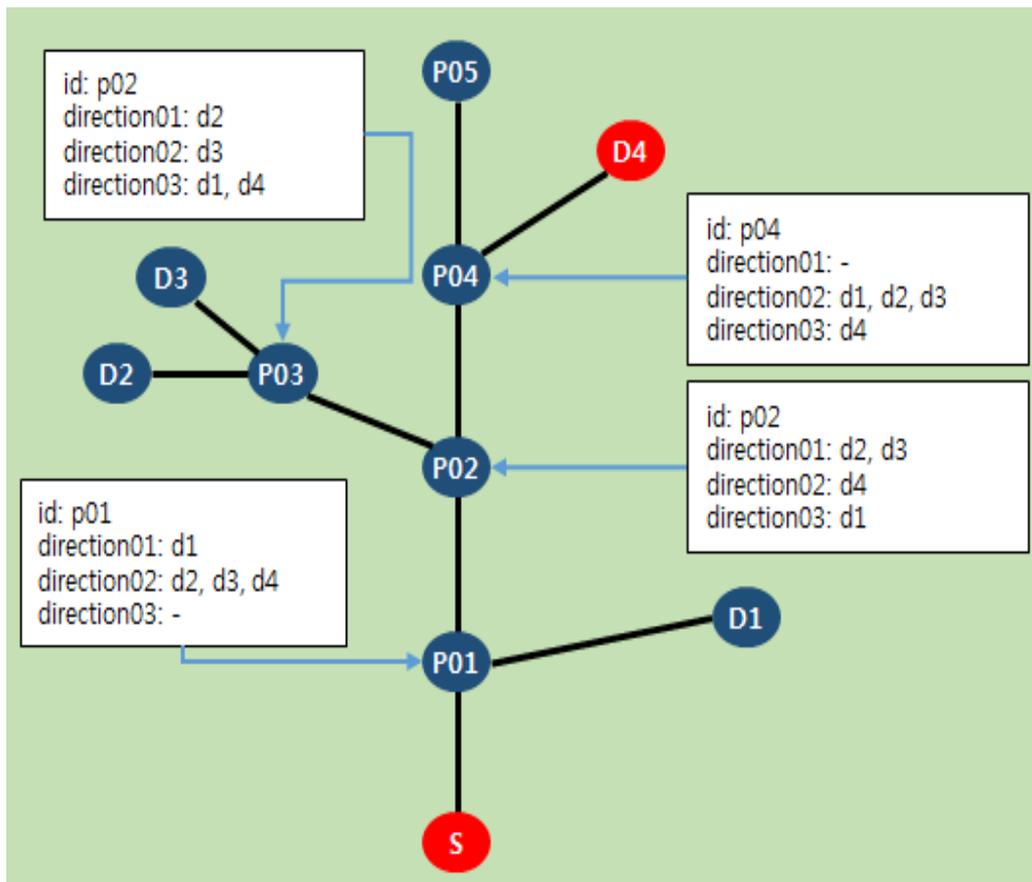


Figure 5. Example of Direction Information

Starting from S, it first checks the destination information from P01. If you check the direction information of direction 1 in P01, you can see that no information exists on D4. Once you check the direction 2, the information regarding D4 exists, thus it heads in the direction of direction 2. In P02, the direction is determined and proceeded in the same manner. Direction 1 has D2 and D3, but since it does not contain the wanted destination D4, the search is proceeded in the direction 2 that contains D4. When we arrive at P04 and check the information for each direction, it is then headed toward direction 3 because it has information regarding D4.

Since the algorithm only searches the route that must be taken according to the direction information, the algorithm complexity is reduced. However, the information is increased on nodes, so nodes must be distinguished by the role they perform.

3. Performance Analysis of the Proposed Room Guidance Algorithm

3.1. Algorithm Performance Analysis

In order to compare the performance of each algorithm, we simply compare the algorithm execution time. The execution time is measured from immediately before the route guidance function execution to after the function completion in nanoseconds (ns).

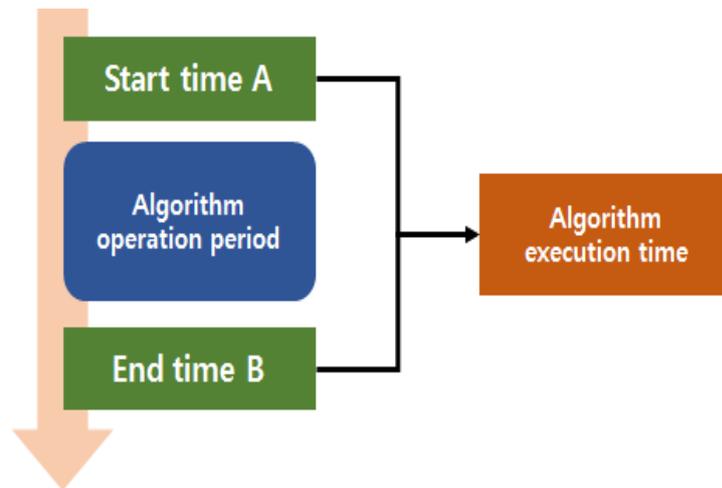


Figure 6. To Measure Execution Time

For the algorithm execution, two maps with different structures has been made, presented in Figure 7. Map 1 uses a total of 10 nodes and has 5 destinations. Map 2 uses a total of 9 nodes, and also has 5 destinations. Each map uses a completely different structure, and they were created to include as much various structures as possible in no cycle environments. This is to judge whether the proposed algorithm operates normally under any circumstances, and whether there is a large change in performance. S is the starting point to start the route search, the node indicated in blue is the passing node, and the node indicated in brown is the destination node. From Map 1, we measure the execution time of the route to each destination D2, D4, D6, D7, D9. Next, from Map 2, we measure the execution time of the route to each destination D3, D5, D6, D7, and D8.



Figure 7. Test Map 1, 2

3.2. Algorithm Performance Test Results

In Figure 8, 9 shows the measurement results of the execution time taken in searching the route from the departure to each destination in Map 1 and Map 2. In Map 1, the execution time is measured when searching from the departure point to each destination D2, D4, D6, D7, D9. In Map 2, the execution time is measured when searching from the departure point to each destination D3, D5, D6, D7, D8. The test results show that BFS relatively takes more time. Although DFS requires relatively less time than BFS, we could identify that the performance is still lower than the proposed algorithm.



Figure 8. Algorithm Runtime Results of Map 1



Figure 9. Algorithm Runtime Results of Map 2

The proposed algorithm is mostly completed within 10 ~ 40 μ s, whereas BFS, which required the longest time, takes at least 600 μ s of time. An even larger difference is expected if a relatively low-performance mobile device is used or if the map size becomes bigger.

4. Conclusion and Future Work

Although it is true that the proposed algorithm has shown a somewhat better performance compared to the existing BFS, there is one problem. It takes up more memory than BFS or DFS. Implemented as a linked list BFS or DFS, only requires enough space for the number of nodes and enough memory space to operate the queue or stack. Memory needs for Node List and all the nodes. Furthermore, memory to operate the queue and stack for the route search will also be required. However, the proposed algorithm stores direction information for all destinations in each node, this will require more memory with the use of mobile devices or larger map sizes.

In addition, this study needs various methods in performance measurement. Therefore, the future work of this paper will be to conduct studies to overcome these limitations. In order to implement a complete room guidance system, the beacon-based user location tracking system will also have to implement.

Acknowledgements

This study was carried out with the support of R&D Program for Forest Science Technology (2017063A00_1719-AB01) provided by Korea Forest Service(Korea Forestry Promotion Institute).

References

- [1] S. Sukkarieh, E. Nebot, H. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications", IEEE Transactions on Robotics and Automation, vol. 15, no. 3, (1999), pp.572-578.
- [2] X. Yun, E. Bachmann, R. McGhee, R. Whalen, R. Roberts, R. Knapp, A. Healey and M. zyda, "Testing and evaluation of an integrated GPS/INS system for small auv navigation", IEEE Journal of Oceanic Engineering, vol. 24, no. 3, (1999), pp. 396-404.
- [3] D. Sanchez-Crespo Dalmau, "Core Techniques and Algorithms in Game Programming", NEWRIDERS, (2003), pp. 228-236.
- [4] B. Vik and T. Fossen, "A nonlinear observer for GPS and INS integration", The 40th IEEE Conference on Decision and Control, vol. 3, (2001), pp. 2956-2961.
- [5] B. Gum, "User Location Determination Using Wireless AP Access Information", JOURNAL OF PLATFORM TECHNOLOGY, vol. 3, no. 3, (2015), pp. 7-11.
- [6] B. Gum, "IP & SF: Indoor Positioning and Service Framework by using Information about BLE-based Beacons in Indoor Spaces", JOURNAL OF PLATFORM TECHNOLOGY, vol. 4, no. 2, (2016), pp. 6-13.
- [7] D. W. Kim, "Performance Analysis of Nonlinear Filters in Indoor Human Localization Based on Wireless Sensor Networks", JOURNAL OF PLATFORM TECHNOLOGY, vol. 3, no. 2, (2014), pp. 15-21.
- [8] Y.-K. Lee, G.-S. Ha and W.-D. Cho, "A study of positioning compensation by using 2 Kalman Filters in GPS signal unavailability area", vol. 10, no. 6, (2014), pp. 59-68
- [9] Y. S. G. Karaisat, M. A. Z. Al-Khateeb, Y. K. Abu-Alreesh, A. A. Ayyash and O. S. Lahlouh, "GPS Navigation and Tracking Device", International Journal of Interactive Mobile Technologies, vol. 5, no. 4, (2011), pp 39-41.
- [10] NATIONAL RECREATION FOREST MANAGEMENT OFFICE, <http://www.huyang.go.kr/main.action>.
- [11] L. Arge, U. Meyer, L. Toma and N. Zeh, "On External-Memory Planar Depth-First Search", Journal of Graph Algorithms and Applications, vol. 7, no. 2, (2003), pp. 105-129
- [12] C.-M. Park, "Location Based Service trend and paradigm shift", Internet & Security Focus, (2013), pp. 24-30.
- [13] A. Kaur, P. Sharma and A. Verma, "A appraisal paper on Breadth-first search, Depth-first search and Red black tree", International Journal of Scientific and Research Publications, vol. 4, no. 3, (2014).
- [14] S.-I. Lee, "Unit's Path-finding Method Proposal for A* Algorithms in the Tilemap", KSCI, vol. 10, no. 11, pp. 71-77.

Authors



Daesik Ko, he received B.S. at KyungHee University (Korea) in 1982 and obtained M.S. and Ph. D. of Computer Engineering KyungHee University (Korea) in 1998. He worked as Post-Doc in University of California, Santa Barbara in 1994. He was the Mokwon University Director Library and Computer Center, Dean of College of Engineering. Also during 2011-2012, he was the president of KIIT. Recently he is a professor of the Mokwon University. He is interested in Multimedia communication, IT convergence, IoT networking.



Jonghyun Choi, he earned his Master of Computer Engineering major in Mobile Embedded System at Kyung Hee University-Suwon (Korea) in 2015 and received Bachelor of Computer Engineering Kyung Hee University-Suwon (Korea) in 2013. Recently he has worked as a researcher at Mokwon University. His main research interests are in fields of mobile, network system and now actively researches about algorithms for navigation (or path-finding).

