International Journal of Smart Home
Vol. 10, No. 9 (2016), pp. 69-82
http://dx.doi.org/10.14257/ijsh.2016.10.9.07

Evacuative Path Design due to the Optimal Cost-Function Arithmetic at Compartment Fire

S.C Moon[1], S.G Lee[2*] and I.S Weon[3]

[1]*Core Tech R&D Lab., LIGNex1, SeongNam 13488, Korea*
[2]*Dept of Mechanical Engneering School, KyungHee University, YongIn, 17104, Korea*
sglee@khu.ac.kr
[3]*Dept of Mechanical Engneering Graduate School, KyungHee University, YongIn, 17104, Korea*

Abstract

This paper proposes an optimal fire escape algorithm based on the A pathfinding algorithm in building fire situations. This algorithm uses building map coordinates to generate the escape path while ignoring diagonal directions. A potential function is calculated by considering situational factors, such as fire location, smoke, and heat information, to estimate the effect on the user's escapability based on his or her location within the building. The calculated potential function determines the path with the lowest cost and illuminates the escape route with leading lights to aid the user to escape hazardous situations that might cause loss of orientation. The proposed algorithm suggests safer, shorter paths and aims to safely lead people to exits while minimizing the dangers of smoke and poisonous gases inside buildings. The coordinates of a real-life building environment were obtained to prove the effectiveness of the escape-route algorithm. Simulation results provided different lowest-cost paths based on various building fire situations.*

Keywords*: Refuge path planning, Cost function, Repulsion index, Fire detection, Maximum-Likely matching*

1. Introduction

Downtown areas are becoming more susceptible to damage from disasters, such as fires and earthquakes, due to the continuing overcrowding of bigger and taller buildings. Preparations for such disasters are considered in many architectural designs. [2–4]

There are two approaches to lessening casualties in fires. The direct approach includes installing escape routes, such as fire-proof elevators and descending life-saving devices. [5] Indirect approaches include improving fire evacuation facilities, such as installing sprinklers. [6] To reduce the potential for secondary casualties in fires, users must be able to easily discover exits in hallways or be familiar with the interior of a building. Even firefighters can become casualties; as the USFA study shows, 18.2% of firefighters die yearly due to becoming lost inside buildings during a rescue. [7] This shows that field of vision and orientation are important to the user in addition to the early recognition and extinguishment of fires. Obtaining real-time exit routes inside burning buildings through either location services or the buildings themselves is becoming more important. In previous works on escape-route planning, Boo (2013) [8] developed a simulator that uses a sensor network based on building maps to find the closest exit for the user. JIA (2008) [9] proved a mathematical model for finding escape routes in mines when a fire occurs. Other studies have designed and implemented simulation models for evacuating a building [10, 11] and providing safe routes in narrow and tight spaces, such as ships. [12]

ISSN: 1975-4094 IJSH

When considering buildings with only a small number of exits in an actual fire situation, it is obvious that a bottleneck would occur at the closest exit, delaying the evacuation. Some research is focused on reducing the bottleneck effect through simulation validations [13], although this is limited to ships and other specific spaces that do not have multiple exits.

This paper sets up a coordinate map that expands horizontally and vertically based on the A* pathfinding algorithm. Maximum matching was used to generate all possible routes within a map environment. Limits on duplicate coordinates and how many times one can use the stairs were imposed. To extract the best conditions for the exit path, the potential functions for the exit and fire location were calculated with a repulsion index on the inverse function of the distance; in addition, weights were added on heat and smoke information to find the lowest costs for the best exit route.[1]

2. Building Map environment and Fire detector information

2.1. Building Map Environment

The proposed algorithm is optimized for pathfinding and uses appearance-based nodes, such as hallways, stairs, and exits. Stair nodes are important as they allow passage to multiple levels. Each node becomes an important part of the building's interior map, and a 3D map is generated The 3D coordinates obtained from the path on the X, Y, and Z-axes can be displayed in squares, as shown in Figure. 1. Thus, information nodes, such as rooms and hallways, become a 2D coordinate system, and the stair nodes show the change only in the Z-axis.

In a building map, nodes such as rooms and offices do not have an exit. These dead-end nodes waste time during the pathfinding phase and thus are marked as non-passable. Useful and non-useful nodes are marked as 1 and 0, respectively. As seen in Figure 2, the hallway nodes and stairway nodes are marked as 1 and the walls and room nodes are marked as 0.

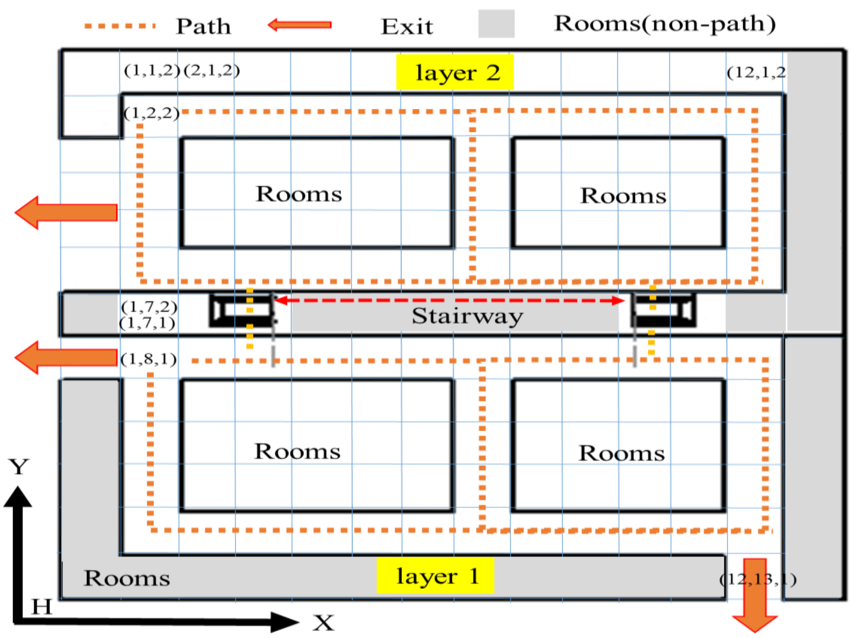


Figure 1. (x,y,h) ID transformed from map building

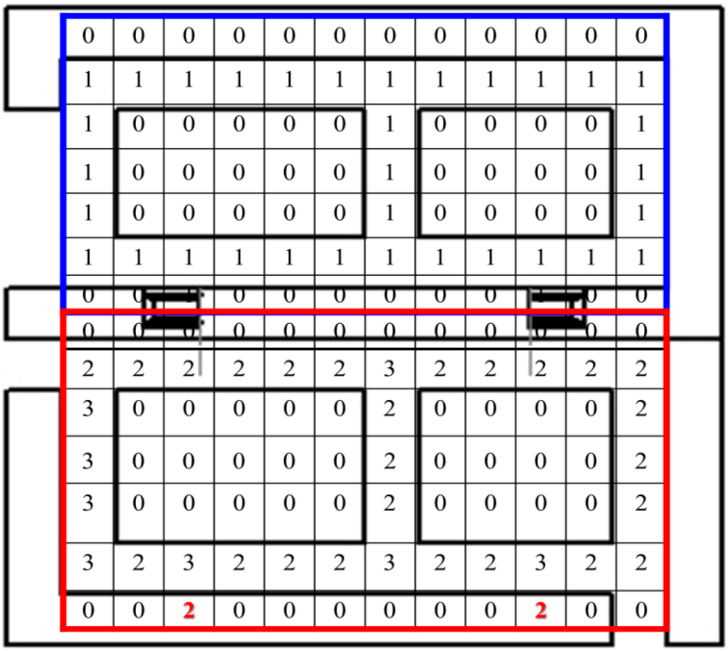


Figure 2. State value for each node ID

2.2. Fire Information

Fire detectors and sensors are placed around the interior of the building based on the Fire Services Act. The sensors used in determining whether a fire has occurred include thermometers, hygrometers, and carbon dioxide/monoxide sensors, which detect the density of gases. Table 1 outlines the types of sensors and their specifications. Each fire detector determines if a fire has started based on probability values and weighted values in their detection ranges.

As shown in Figure 3, there needs to be a step-wise process for fire detection, where there will be different levels of urgency in the escape-route pathfinding process. The different weights are used in the escape-route planning calculations (presented in Chapter 3).

Table 1. Fire – detecting Sensor of Weighting Value

Sensors	Range	Unit	Weight value
Temperature	-20 ~ 80	℃	80(1) ~ 40(0.2)
Humidity	0 ~ 100	%Rh	0(1) ~ 60(0.2)
CO_2	0 ~ 2000	ppm	2000(1) ~ 400(0.2)
CO	0 ~ 300	ppm	300(1) ~ 35(0.2)

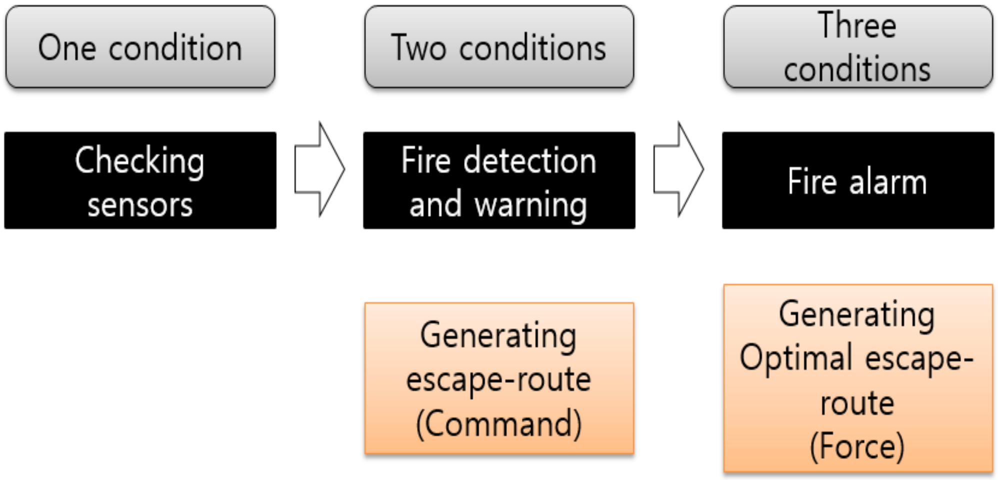


Figure 3. Process for Fire Alarm

3. Escape-Route Exploration Algorithm

3.1. Generating Paths based on Node Conditions

Given a starting node, the availability of the neighboring nodes must be determined before a path can be generated. In a map environment, each node's condition is converted to an index value based on Figure. 3, and some are given as examples in Table 2. Each node's next available destination node angle is set with an azimuth Kθ by selecting values that end up perpendicular to inputs ranging from 0–270° from Equation (1). We only considered perpendicular movements within a map node system and have disregarded diagonal direction movements.

$$\begin{bmatrix} K_{0^\circ} \\ K_{90^\circ} \\ K_{180^\circ} \\ K_{270^\circ} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} State1_{Pos} \\ State2_{k} \end{bmatrix}, (k = 1 \sim 4) \quad (1)$$

Table 2. K-state value based on Surrounding Nodes

Example 2-Node		θ	State-2_k	Index(S)	K_θ
State-1_{pos} (x,y,h)	1 (7,3,1)	0	1	[1, 1]	1
		90	1	[-1, 1]	1
		180	1	[-1, -1]	1
		270	1	[1, -1]	1
		Stair	0	-	0
State-1_{pos} (x,y,h)	1 (6,3,1)	0	0	[1 ,0]	0
		90	0	[0, 1]	0
		180	0	[-1 ,0]	0
		270	1	[1 ,-1]	1
		Stair	1	-	1

Each node's condition information is marked as State-1pos, the current node is distinguished by the state value 1 or 0, and State-2k represents the conditions of the neighboring nodes from the current node.

Each node's condition is defined by the angle calculations from Equation (1), and the defined available moves range from 0 to 4 destinations. From each node's availability, an escape route like that in Figure. 4 can be discovered. The exploration directions are analyzed in the X and Y-axis directions. The available destination angle Kθ of Figure. 4 is changed to 1 or 0 depending on the absolute calculations of Equation (2) and Index (S)'s matrix elements (a1, a2), which are obtained via different azimuth values.

$$\text{Index(S)} = \begin{bmatrix} a1 & a2 \end{bmatrix} => |a1 \cdot a2| = K_{\theta} \qquad (2)$$

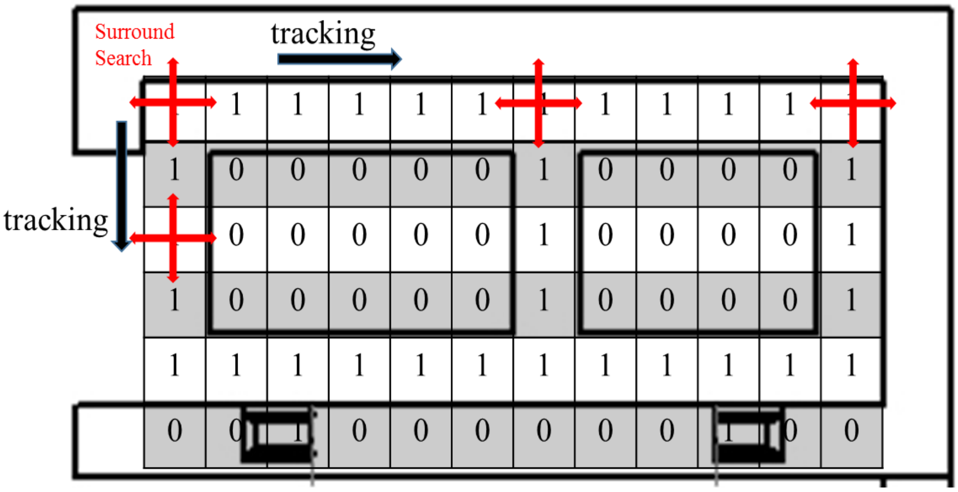


Figure 4. Path State Generation from Surrounding Nodes

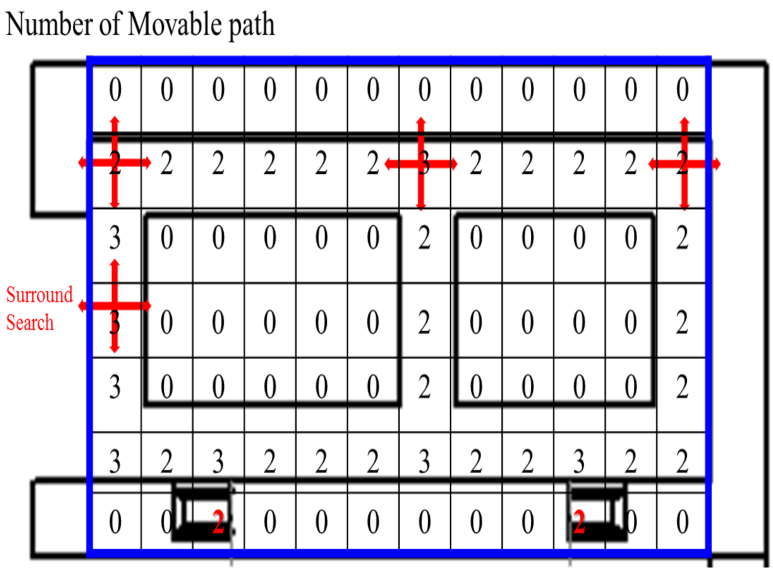


Figure 5. Number of Movable Paths

Table 3. Path-finding Algorithm

Algorithm 1: Finding Path Node	Algorithm 2: Generation Escape path

- Simplicity building Node as ID
- ID change to state value
Path State-1_{pos} = 1
Non-path State-1_{pos} = 0
- Detecting the exits and exit ID
- Start to calculate the 4th azimuth from the exits
For k = 1:4 (k = 0°, 90°, 180°, 270°)
- Calculated range of movement as feature K_θ
If (K_θ = 1)
- Count N & matching ID
- Extract a stairway
End if
End for
	While
	- connecting the evacuative path
	- calculate the number of path planning (num)
	If connected path-ID = exit-ID || path-ID => repeated path
	- evacuative path find unsolved path
	If num(t) = num(t-1)
	- group number will finish when the number does not change
	Break
	End If
	End while

Each node's available number of destinations *N* is categorized as shown in Fig. 6. Structures such as rooms and walls have a State-1_{pos} *N* of 0, while exit nodes and intersections have a State-1_{pos} *N* of 3. Corner nodes have a State-1_{pos} *N* of 2. Stair nodes have an *N* value of 2, and depending on the number of available destinations near the stair node, the *N* value increases by 1. The escape route ends when the pathfinding process encounters a duplicate node or an exit is successfully found.

Figure 5 represents the number of movable paths in all four directions based on each node's State-1_{pos}.The algorithm for the repetition technique is applied, as in Table 3. The algorithm is divided into two phases: the first phase is where each node's path combinations are formed, and the second phase is the selection of successful exit paths.

3.2 Determining the Potential Function based on Fire detection

Data from thermometers, hydrometers, and gas sensors are used as conditions to define when a fire has been detected. As mentioned in Figure. 4, when two conditions are met, this is defined as a "fire warning." When three conditions are fulfilled, a fire is considered detected and calculations of the repulsion index take place.

Depending on the fire's location, each node's potential function is calculated with the reciprocal of the distance between the fire and itself, and a repulsion [14] Urep value is assigned to each node. The relative repulsion index of the fire's location is the inverse of the distance between each node and the location of the fire, and its calculations are presented in Equation (3).

$$U_{rep} = \frac{1}{(dx+dy)^2} \tag{3}$$

$$F = U_{rep} \times \omega_f \tag{4}$$

$$Path_k = \sum_{1}^{n} F \qquad (k, n \geq 1) \tag{5}$$

Here, Equation (3)'s distance values are the sum of the perpendicular distances left and right of the node, with dx representing the number of nodes in the X-axis and dy representing the number of nodes in the Y-axis.

To give the directional information of the exit in the simulations, normal paths, in relation to the distance to the fire, have a weight difference of 0.5 in a "fire warning" situation and 0.9 in a fire detection situation. The generated fire information is given as a factor for the repulsion index and is considered in the calculations of each node's cost function F in Equation (4).

When a matching procedure via Procrustes analysis is completed, a set of new coordinates are generated at each location instance. In order to obtain the coordinates with the least amount of errors, a recursive least-squares method was applied. The optimal approximation equations of each n location value were obtained by using this method.

As shown in Figure. 6, in order to obtain the optimal approximation equation, the next n locational information sets were overlapped to the previous sets.

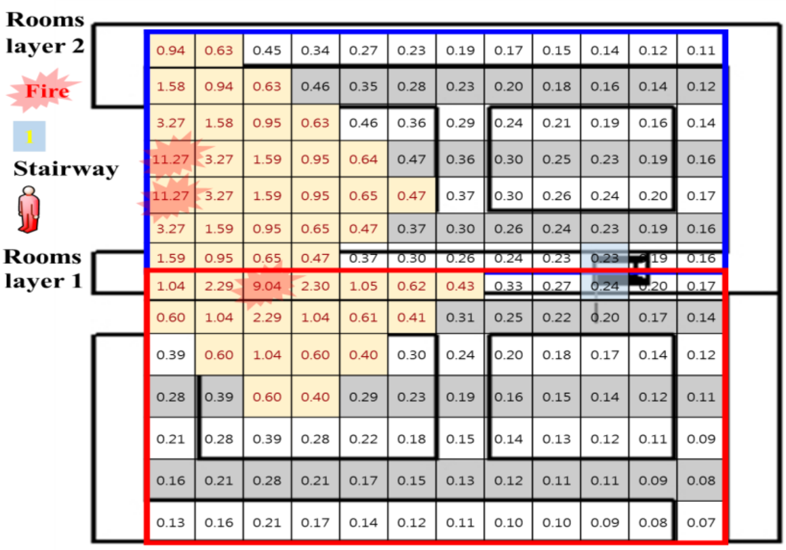


Figure 6. Cost function for Each Node when the Fire Occurred

Based on each exit route explored from the user's location, the sum of each node's cost function is presented in Equation (5).

Here, k is the discovered number of exit routes and n is the number of nodes each path takes to get to the exit. The cost-function distribution classified by equation (5) helps in determining the direction of the evacuation. The proposed algorithm's exit route has a cost twice the repulsion value when a stair node is present, and this allows for the decrease of stair node usages in the algorithm where movement by stairs delays evacuation times.

4. Experiment Setup and Results

A virtual escape-route experimentation was carried out as shown in Figure. 1 with a two-story building with all hallways, rooms, and walls converted to coordinate nodes. The first and second floors are connected to a stairway at one end of the building map, which makes the seventh column the Y-axis. The floor plan's size ranges from (1,1,1) to (12,13,2). The exits on the first floor are at coordinates (1,8,1) and (12,12,1), with the second floor's exits at locations (1,3,2), (1,4,2), (1,5,2), and (1,6,2). Fig. 7 shows the locations of the fires in the simulation at (1,4,2), (1,5,2), and (7,3,1) along with each node's cost-function calculations and repulsion indexes depending on the proposed exit route

The cost function is calculated with the location of the fires only without considering the location of the user. Nodes shaded in yellow are considered dangerous, as their cost function is within the top 10th percentile. It also shows how a fire affects the region around it a similar manner.

Users were placed in locations (2,8,2) and (3,7,2). The exit-route algorithm used these starting points to find a correct exit path based on each node's cost generated by the cost-function algorithm. The cost-function distribution results of the exit-route algorithm simulation when the user is placed at (2,8,2) can be seen in Figure. 7.

A blue-dotted line is marked on Figure 6. for exit paths in the order of increasing cost functions. The red line represents the number of nodes for each path. Figure 7 shows how the travel distance and cost function are not proportional, how the fire affects the weights, and thus the cost function of each path. Figure 8 shows exit paths with particular characteristics, as seven exit paths, paths A through C-8, share a certain length of the escape route with each other. To obtain the optimal path, we compared the values of the travel length (node count) L and cost value P.

Route A has a short distance L of 9, but has the eighth highest cost function at 0.79. Routes B and C have three different cases of different cost values P. Route B-1 has the fourth lowest P of 0.54 and L of 25. B2 has an L of 19, and P is the second lowest: 0.41. C-1 and C-2 have p values of 0.48 and 0.54, which are respectively the third lowest and the lowest P values. Their respective L values are 27 and 21. B-3 and C-4 have respective P values of 0.75 and 0.67, with L values of 29 and 31. The user in Figure 8 would have to follow route C-2. Despite the long distance of the suggested exit route, it is the safest path that one could take.

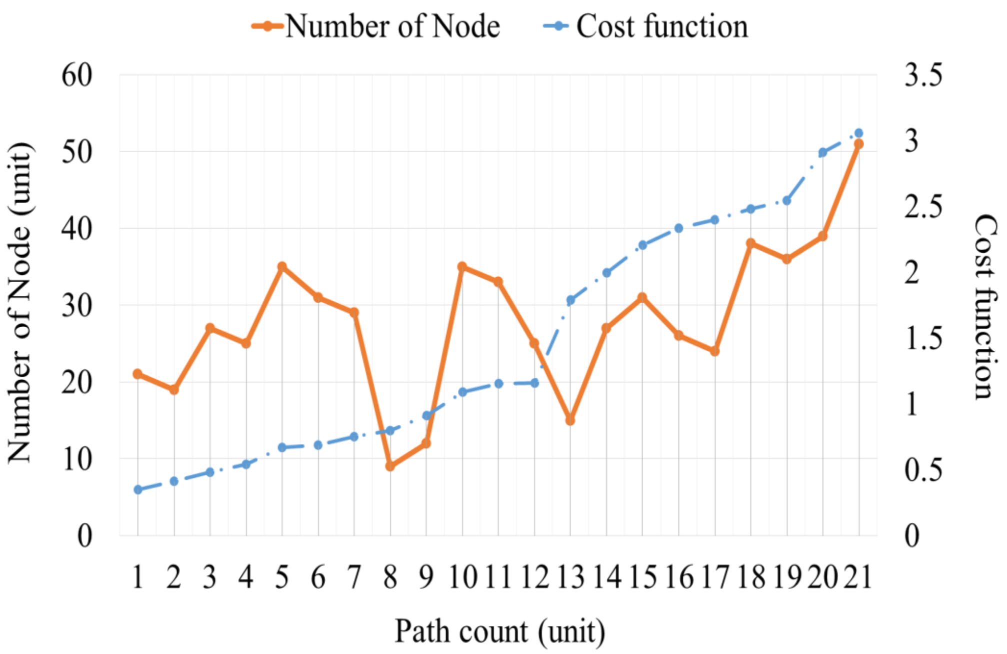


Figure 7. Comparison of Cost Function and the Number of Nodes about All Paths (Position: {2,8,2})

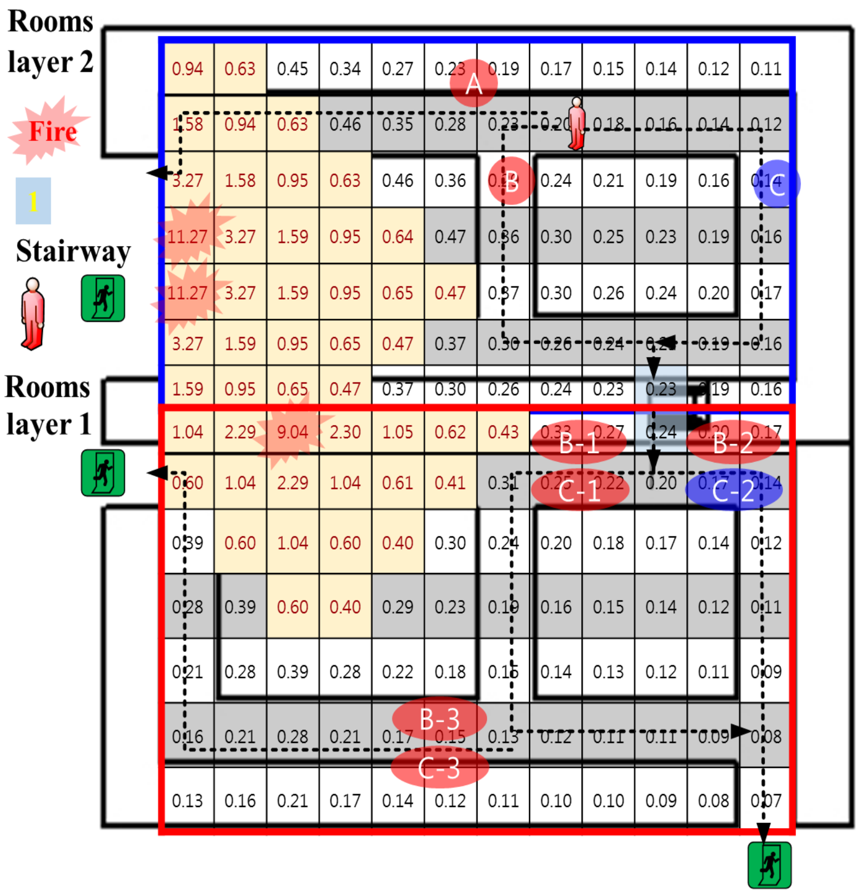


Figure 8. Samples for Escape Path by Cost Function (Position: {2,8,2})

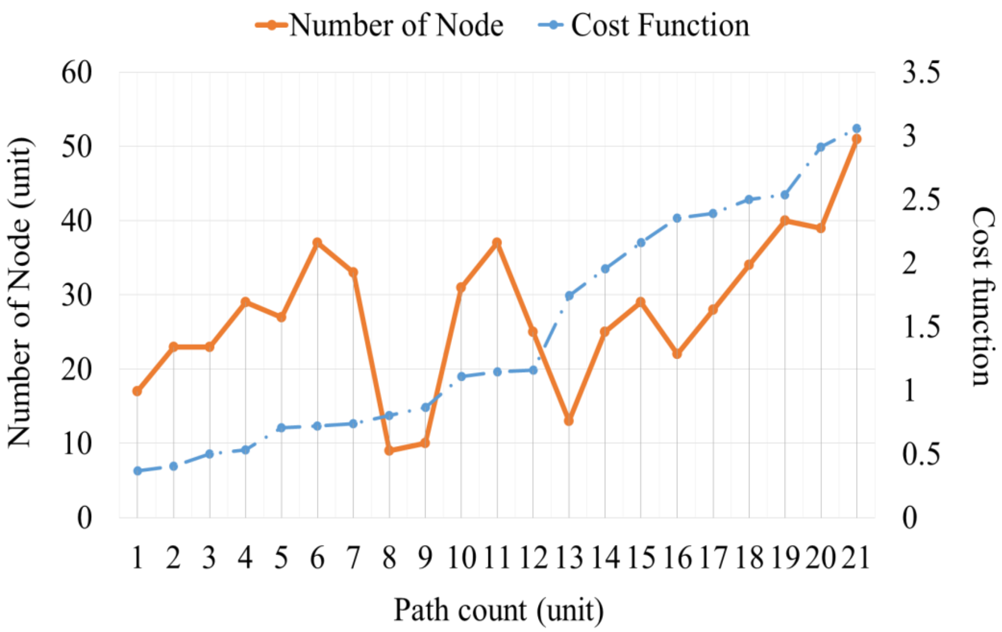


Figure 9. Comparison of Cost Function and the Number of Nodes about All Paths (Position: {3,7,2})

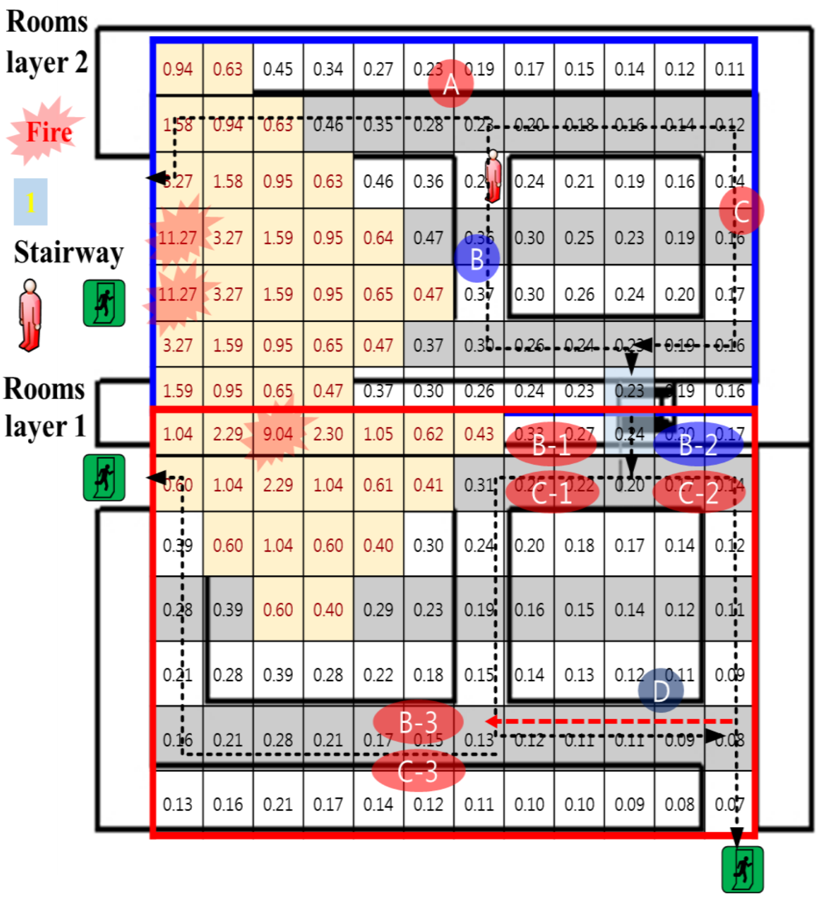


Figure 10. Samples for Escape Path by Cost Function (Position: {3,7,2})

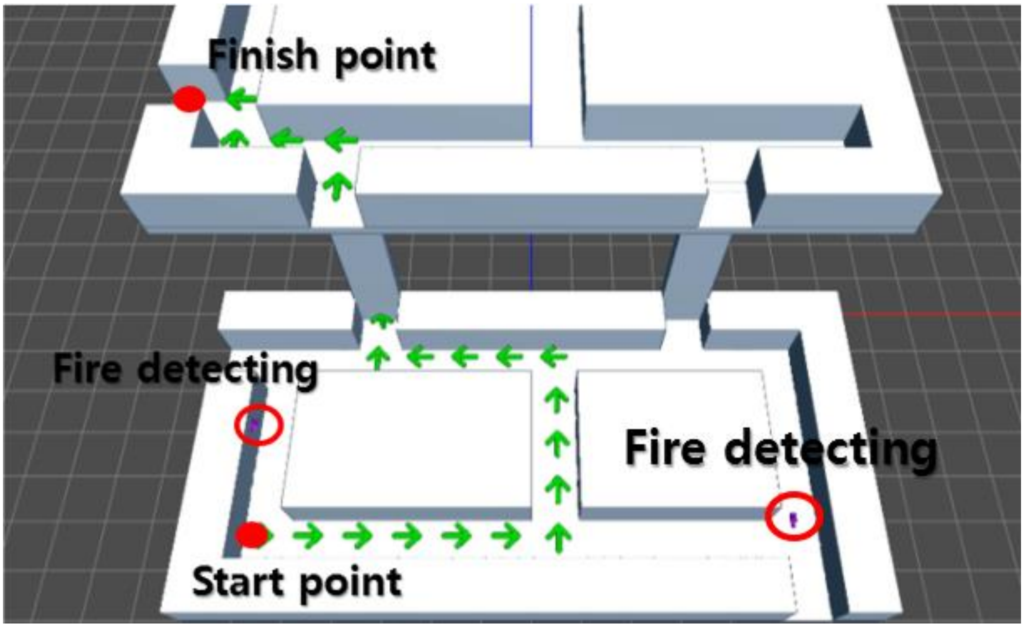


Figure 11. 3D Simulation of Generating Fire Escape Route (Fire: 2-points)

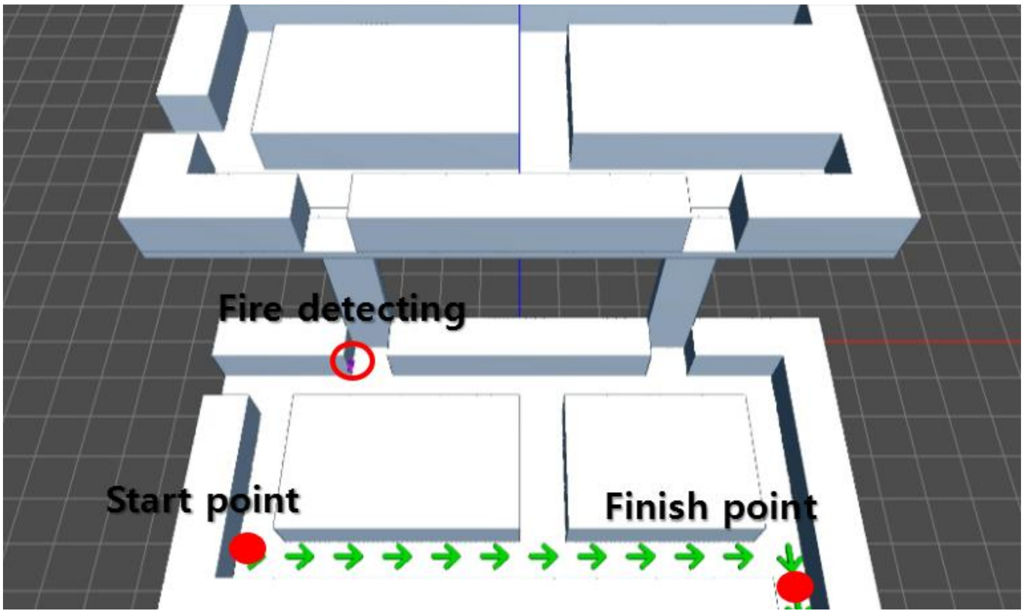


Figure 12. 3D Simulation of Generating Fire Escape Route (Fire: 1-point)

Next, Figure. 9 and Figure. 10 show the simulation results when the user starts from the location coordinates (3,7,2). As the cost-function distribution shows in Fig. 10, the eighth-best route, A, has an L of 9 with a P of 0.8, similar to the first simulation. However, when we check routes B and C, the recommended routes and the rankings have changed. B-1 and B-2 have respective L values of 23 and 17, with P being 0.5 and 0.37, respectively. These values are the third best and best exit routes.

C-1 and C-2 have L values of 29 and 23, respectively, with P values of 0.53 and 0.40, which are the fourth and second-best exit routes. Compared with the first simulation, route B ended up having a much lower cost function in this experiment. For routes B-3 and C-3, their respective L values are 27 and 33, with respective P values of 0.71 and 0.74. D' has an L value of 37 with a P value of 0.72, which shows that the distance between the user and the fire is an important factor in calculating escape routes.

5. Conclusion

This paper presents an exit-route pathfinding algorithm for fire situations in a multiple-story building. The A* pathfinding algorithm, a pathfinding algorithm for perpendicular node maps, was used to obtain the optimal escape route. Weighted fire information and a repulsion index in relation to the relative distance were used to calculate the cost function between each node in a path.

The path with the lowest cost is recommended depending on the location of the user and the fire, and simulation experiments have been carried out. Future works include calculating the node's cost function depending on the node structure and varying the location and type of fires, which needs to be studied thoroughly to apply correct weights in simulations.

Acknowledgments

This research was supported by Convergence Technology Development funded by the Small Business Administration (No.S2176537), and supported by 100-year-old elderly-friendly society corresponding product research and development project funded by the Department of Health and Human Services (No.HI15C1027)

References

[1] S. C. Moon, S. G. Lee and I. S. Weon, “Cost-Function Algorithm based on Repulsion Index by Fire Detection and Refuge Path Planning”, Asia-pacific Proceedings of Applied Science and Engineering for Better Human Life, Vol. 8, **(2016)**, pp. 30-33

[2] S. W. Moon and H. J. Seong, “u-Disaster Prevention System based Real-Time Fire Monitoring in a Building Facility,” Journal of construction engineering and project management, Vol. 12, No. 1, **(2011)**, pp. 107-114, 2011.

[3] B. S. Han, “Skyscraper structure system & case study,” Daewoo engineering technical issue, pp. 3-21,

[4] The University of Hong Kong faculty of architecture, “Visit : Central Structure of Skyscraper”, Hong Kong Institute of Architects, **(2012)**.

[5] J. P. Choi, B. J. Kang, Y. S. Part and Y. J. Lee, “Schematic Development of ‘Refuge Floors’ and ‘Areas of Refuge’ for High-Rise Buildings,” The Architectural Institute of Korea, Vol. 21, No. 11, **(2005)**, pp. 147-154.

[6] Y. S. Kim and D. S. Sug, “A Study on the Development of Intelligent Guiding Exit Sign System,” Fire Science and Engineering, Vol. 20, No. 6, **(2006)**, pp. 131-134.

[7] Fire engineering, “The time required to escape the scene of firefighters,” Fire & Rescue, **(2004)**.

[8] J. P. Boo, S.C. Kim, D.H. Park, H.C. Bang and D. H. Kim, “ A Study of Fire Refuge Guide Simulator Based on Sensor Networks” Lecture notes in computer science v.7861, **(2013)**, pp.914 – 919.

[9] Jia and J. Zhang, “Mathematic models for mine fire simulation and determination of refuge route”, Journal of natural disasters, Vol. 17, No. 1, **(2008)**, pp. 163-168.

[10] Y. W. Kim, C. Y. Lee and D. H. Kim, “A Design and Implementation of A Scheme of Three-dimensional Refuge Guide”, Proceedings of the KASI fall conference, **(2008)**, pp.282-285.

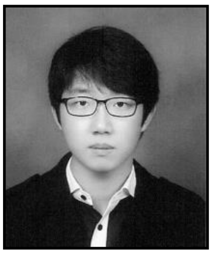
[11] B. O. Jang, “Design and Implementation of Evacuation Simulation of Indoor Environment Fire”, Journal of the Korea Society for Simulation, Vol. 19, No. 2, **(2010)**, pp. 1-8.

[12] S. Ha, Y. O. Cho, N. K. Ku, K. P. Park, K. Y. Lee and M. I. Roh, “Passenger Ship Evacuation Simulation using Algorithm for Determination of Evacuating Direction based on Walking Direction Potential Function,” Journal of the Society of Naval Architects of Korea, Vol. 50, No. 5, **(2013)**, pp. 307-313.

[13] “Fire Protection Contractor Law Enforcement Rules,” The National Emergency Management Agency, **(2013)**.

[14] K. S. Woo, J. H. Park, J. H. Kim and U. Y. Huh, “Path Planning for Mobile Robots Using Fuzzy Potential Field Method,” Proceedings of the KIEE summer annual conference, **(2001)**.

Authors



Ihn Sik Weon, He is Master and Doctor unify course in Department of Mechanical Engineering, KyungHee University, Korea. He has done his B.S in Control and instrument engineering from Korea University. Presently he is carrying his research work as development of position accuracy for autonomous driving and control vehicle, mobility system.

Soon Geul Lee, He is a professor in Department of Mechanical Engineering, KyungHee University, Korea. He has done his Ph.D in Human Engineering from University of Michigan. His areas of specialization are Control mobility system, balancing robot and mobility system by CMG and Detection environment indoor and outdoor for autonomous driving.

Sang Chan Moon, He is a Research Engineer in Core Tech R&D Lab., LIGNex1, Korea. He has done his Ph.D in Mechanical Engineering from Kyung Hee University. He is in interested in autonomous driving, Precision positioning of vehicle, recognition behavior and learning algorithm. Presently his areas of specialization are Inertial navigation system, hybrid navigation for precision guidance system

