Development of a Server-Independent Algorithm for Safe Evacuation Systems Utilizing Exit Signs

Jehyun Cho[a], Ghang Lee[a*], Seungjae Lee[a] and Hyunoh Kim[a]

[a]Dept. of Architectural Engineering, Yonsei University, South Korea
E-mail: jkun86@naver.com, glee@yonsei.ac.kr, deeppupils@gmail.com, hyunohkim7@gmail.com

ABSTRACT

A smart exit-sign system is an exit-sign system that can proactively orient the direction of an exit sign toward the shortest safe path in fire situations. In a previous study, an automated direction setting algorithm for a smart exit sign was proposed that assumed the exit signs were controlled by the main central server. This study proposes a new direction setting algorithm for a server-independent smart exit-sign system. The server-independent direction setting algorithm (SIDSA) can be divided into four main steps once the smart exit-sign system is installed and initialized: (1) update of the fire-status information detected by a sensor; (2) sharing of the fire-status information with neighboring exit-sign modules; (3) update of the shortest-path (distance) information according to the fire situation; (4) update of the direction on the display board. SIDSA has been validated through multiple simulations.

Keywords –

Exit sign; Smart exit sign; Evacuation system; Shortest path; Way finding; Server-independent direction setting algorithm (SIDSA)

1 Introduction

When an emergency situation occurs in a building, evacuees not familiar with the building's structure rely primarily on their instincts when looking for evacuation routes. The more complicated and larger a building, the more likely it is that evacuees will get disoriented. Kobe confirmed through experiments that people's dependence on exit-sign lights significantly increases in emergency situations compared to nonemergency situations [1].

Currently, the most commonly used exit sign has a fixed direction sign that is sometimes ambiguous and has the potential to guide evacuees to dangerous areas. Choi's experiment showed that more than 60% of participants failed to find an exit when relying on exit signs in a complete blackout situation, regardless of their age, nationality, and disability. These results suggest that fixed-direction exit signs can be ambiguous [2].

To improve conventional exit signs, the concept of smart exit-sign systems has been proposed [3-6]. These are sensor network systems that detect dangerous areas and orient the direction of exit signs toward the closest safe exit. Some of these systems were proposed as mobile phone–based systems [3].

In a previous study[4], we proposed the automated direction setting algorithm (ADSA), which is based on Dijkstra's shortest-path algorithm. In emergent situations, ADSA calculates the shortest path and orients exit-sign directions toward the shortest safe exit from each exit-sign module. However, ADSA assumed that exit signs would form a sensor network and communicate with each other through a central server. To the contrary, in cases of the loss or failure of a network, or in the case of the failure of the central server due to fire, the system will not function properly. To overcome this limitation, we developed a new direction setting algorithm that does not require a central server.

This paper presents the new server-independent direction setting algorithm (SIDSA) for smart exit-sign systems. Instead of using a central server, modules of a server-less smart exit-sign system communicate only with neighboring modules and update the sign direction based on shortest-distance and direction information sent by neighboring modules. The next section of this paper provides a review of existing smart exit-sign systems, including ADSA. The third section describes SIDSA using illustrated examples, and the fourth section introduces a simulator developed to validate the algorithm. Finally, the fifth section reports the validation results.

2 Literature Review

Several previous studies proposed a smart exit-sign system and an associated algorithm for calculating the shortest path, but these systems were proposed as non-tested conceptual systems.

Kim et al. proposed a smart exit-sign system using Floyd's algorithm [5]. Another research team proposed a similar system using Dijkstra's algorithm [6]. These studies, however, explained neither their choice of algorithms nor how they tackled several problems that occur when Floyd's and Dijkstra's algorithms are deployed in a smart exit-sign system. For example, the system based on Floyd's algorithm calculates paths that do not need to be considered, such as the shortest paths from one door to another, and the system based on Dijkstra's algorithm calculates the shortest safe path only from a specific starting point (exit sign) in a building and is unable to consider every exit sign in the building.

Our previously developed algorithm, ADSA, is a variation of Dijkstra's algorithm, which can calculate the shortest path from every possible starting point (that is, every exit sign) to the nearest exit. The main differences between ADSA and Dijkstra's algorithm are that ADSA classifies node into exit, exit sign, and nodes whereas Dijkstra's algorithm does not distinguish node types, and that ADSA includes additional functions to update the direction of an exit sign after excluding unsafe areas from the shortest-path calculation.

However, all the algorithms mentioned above were developed on the basis of a sensor network with a central server, which will encounter communication issues when connections to the central server are compromised [7]. If the connection between exit signs and the central server is disconnected due to, for example, damage of the wire between some exit signs and the central server, the server-dependent smart exit-sign system will not function as designed. This study introduces a new algorithm, SIDSA, for finding the safe path to the closest exit in a smart exit-sign system without a central server. When a central server, which collects the status of each exit sign and recalculates the shortest safe path, does not exist, each exit sign should determine a safe direction depending on the limited information passed from neighbor exit signs. The next section provides a detailed description of SIDSA, which was developed to overcome these issues.

3 The Server-Independent Direction Setting Algorithm

SIDSA includes functions for detecting an exit-sign node on fire, collecting information about neighboring nodes, calculating the distance to the nearest exit, and orienting the direction of exit signs to the shortest safe path to the nearest exit. This algorithm makes the individual exit sign independent of a central server. This section describes SIDSA in detail and provides illustrated examples.

3.1 Characteristics of the Server-Independent Direction Setting Algorithm

The main characteristics of SIDSA are as follows. Firstly, each exit module stores and updates four pieces of data: a unique identifier, the status information (whose default is "safe"), the distances between the exit sign and its neighboring exit signs, and the distance from the exit sign to the nearest exit. The initial distance value is calculated using ADSA before the installation of exit-sign modules in a building.

Secondly, SIDSA is implemented in each exit-sign module. It updates the distance and direction information according to updated neighbor-node information so that each exit-sign module can show the correct direction even if several exit modules are broken down or disconnected from other modules due to a fire.

Thirdly, SIDSA, being implemented in an exit sign, communicates information only with neighboring exit signs directly connected to the exit sign in which it is implemented when determining a safe direction. This makes the communication speed of SIDSA faster and more reliable than that of a server-based smart exit-sign system.

Lastly, with respect to the entire system, directions are updated gradually from one exit sign such as an exit sign on fire to other exit signs. Figure 1 illustrates a case in which people watch the gradual process of direction change in real corridor space.

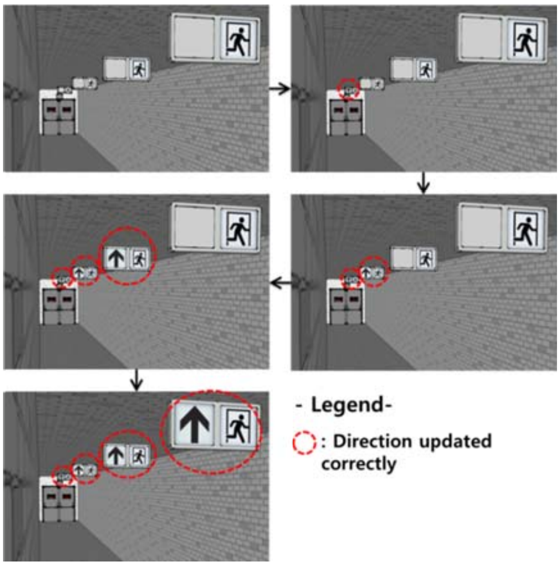

Figure 1. An example of gradual direction changes from an exit

The four major steps of SIDSA are described in detail in the next section.

3.2 Steps of the Server-Independent Algorithm

SIDSA is composed of four steps. Figure 2 is a flowchart of these steps.

Step 1: Update the status data

The first step checks whether a fire around an exit-sign module has occurred. An exit-sign module may detect a fire using sensors for temperature, hazardous gases, and so on. If a fire is detected, the status of the exit-sign module changes from “safe” to “unsafe” and SIDSA moves on to Step 2.

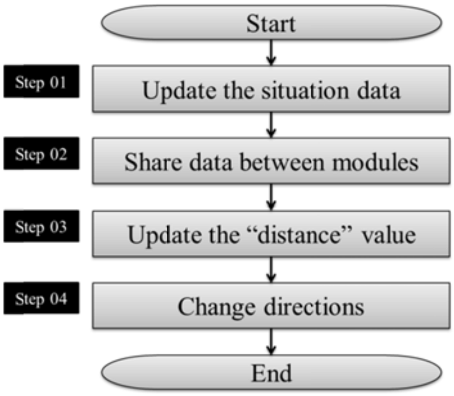

Figure 2. Flowchart of four steps of SIDSA

Step 2: Send data to neighboring exit signs

In this step, the unsafe module sends the new status data collected in Step 1 to its neighboring exit signs.

Step 3: Update the distance value to the nearest exit

The third step recalculates the distance value from an exit sign module to the nearest exit after excluding an unsafe exit sign. The calculation is conducted using the distance information between an exit sign and its neighboring exit signs and the distance between neighboring exit signs and their nearest exit.

Figure 3 is an illustrated example of finding the exit door nearest to an exit sign A. In this example, exit sign A is linked to the other exit signs, B, C, D, and E. From exit sign A, the values of 15m, 13m, 10m, 7m, are ‘neighbor distances’, which are the distances between exit sign A and the other exit signs. These values are saved in exit sign A during the initialization process, that is, before exit signs are installed.

The other distance values in each data set of an exit sign denote the ‘shortest distance’, which is the distance from each exit sign to the nearest exit. This second distance value may change when an unsafe exit sign is detected. A new distance between exit sign A and the nearest exit is the sum of the two distance values, that is, the distance between exit sign A to neighboring exit signs and the distance between a neighboring exit sign and its nearest exit. Among four distance values, the direction to the nearest exit is the direction toward a neighboring exit sign with the smallest distance value. However, exit sign A stores the shortest safe path distance value, as shown in Table 1.

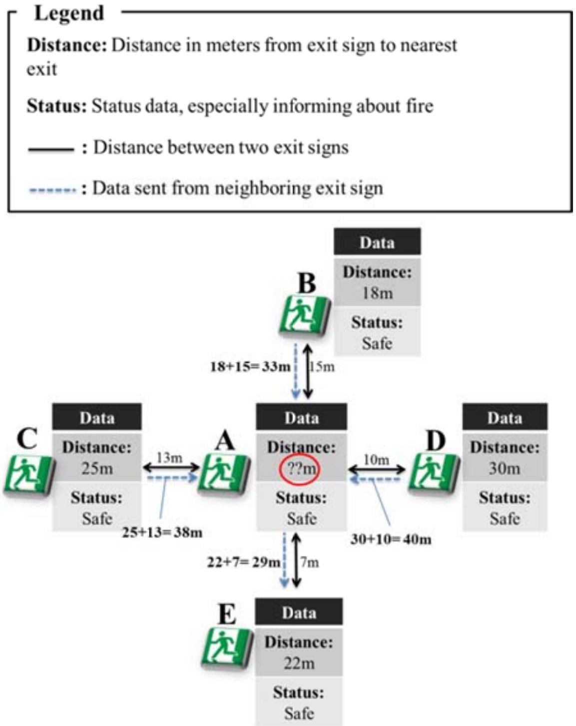

Figure 3. Finding the nearest exit door from exit sign A

In this example, the smallest sum distance value is 29 m (= 22 m + 7 m) toward exit sign E. Thus, exit sign A saves 29 m as the final distance value and chooses exit sign E as the direction toward the nearest safe exit.

Similarly, if the status of exit sign E changes to “unsafe,” exit sign E is excluded from the update process.

Table 1. Data about status of exit sign A after distance-value update

Exit Sign			
Internal Information			
ID		A	
Status		Safe	
Shortest Distance(m)		**29**	
Neighbor Distances(m)			
Exit sign B	15	Exit sign D	10
Exit sign C	13	Exit sign E	7
External Information			
Exit sign B		Exit sign C	
Shortest Distance(m)	18	Shortest Distance(m)	25
Status	Safe	Status	Safe
Exit sign D		Exit sign E	
Shortest Distance(m)	30	Shortest Distance(m)	22
Status	Safe	Status	Safe

Step 4: Orient the direction to the new nearest exit

In the last step, the direction of an exit sign is updated to reflect the new nearest exit according to the results of Step 3. For example, in the case of Figure 3, exit sign A changes the direction sign toward exit sign E, which is the next exit module on the path to the nearest exit from exit sign A.

These four steps are continuously repeated by every exit signs in a network.

4 Implementation of SIDSA in a Simulator

We developed a SIDSA simulator to test the algorithm’s validity. The simulator consisted of several functions and interfaces.

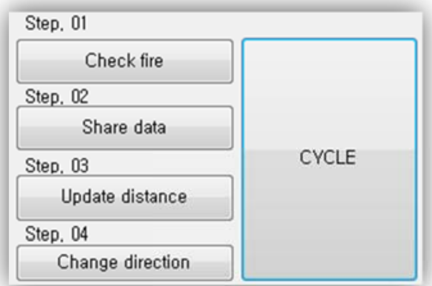

Figure 4. An interface for running SIDSA step-by-step

Figure 4 presents an interface for running the four steps of SIDSA. Users can run each of the four steps individually by pressing the appropriate button on the left side of Figure 4 or run the entire cycle without a break by pressing the cycle button on the right side.

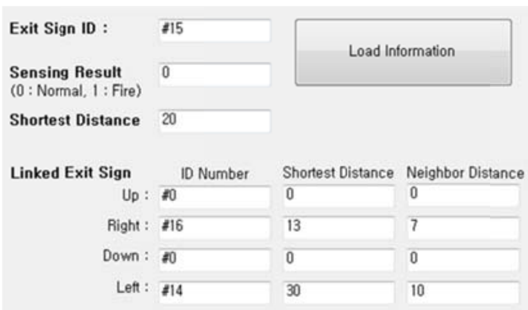

Figure 5. An interface for tracking an exit sign’s internal and external data

Figure 5 presents an interface of the SIDSA simulator for checking the information stored in a particular exit sign. In the example shown in Figure 5, exit sign 15 is connected to exit signs 14 and 16. The shortest distance value to the nearest exit is 20 (= 13 + 7) via exit sign 16. In the same way, the distance value of 40 via exit sign 14, which is placed on the right of exit sign 15, means that there is an exit at a distance of 40 meters from exit sign 15 via exit sign 14.

The third interface allows users to enter the number of a specific exit sign to indicate that fire occurred near that sign. This function is designed to help users check whether the direction of an exit sign is updated correctly under various circumstances.

5 Validation

To validate the applicability of SIDSA, we ran simulation tests for several test cases. Figure 6 illustrates a network of exit-sign modules on the floor plan of a hypothetical building, which is designed to have four exits, three intersections, two corners, and 27 smart exit signs. The distance between two directly connected exit signs is designed to be 1 m to easily check the validity of SIDSA.

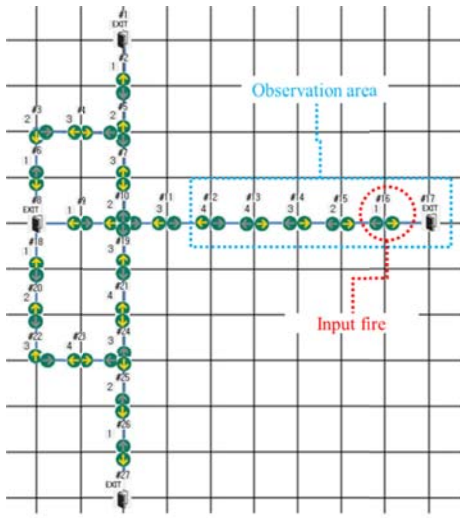

Figure 6. Assumed corridor plan and arrangement of exit signs

We assumed fire situations at various locations with different spatial characteristics, such as a corner, a middle point of a corridor, an intersection, and an exit module close to an exit, and we manually checked the validity of update results step-by-step. Figure 7 presents a step-by-step visualization of SIDSA results.

Through a number of implementation cycles, the bugs in both algorithm and simulator have been corrected.

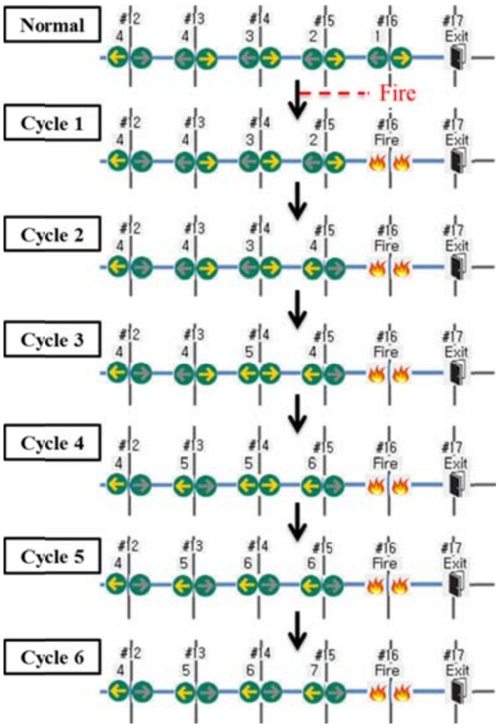

Figure 7. Multiple implementation of SIDSA to reflect a specific situation

Figure 8 illustrates the final simulation results with correct directions.

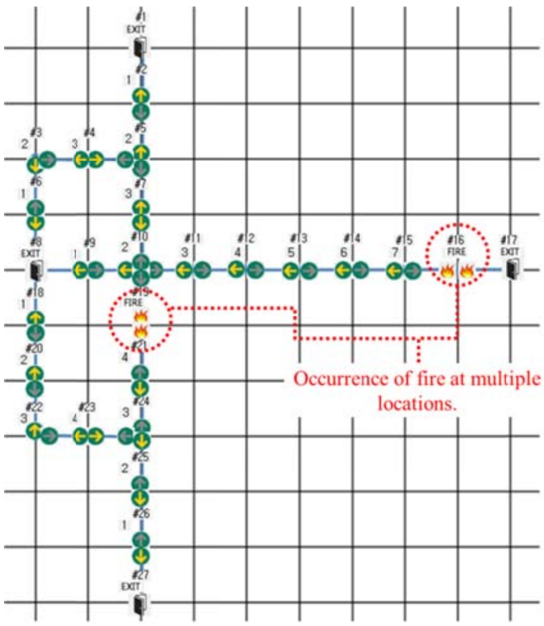

Figure 8. Occurrence of fire at multiple locations

6 Conclusion

Existing smart exit-sign systems have the critical limitation of not being able to perform when the connection from exit-sign modules to the main server is lost. We developed SIDSA to overcome the limitation of server-based smart exit-sign systems. The algorithm is composed of four major steps. It first updates the status of an exit sign when a fire breaks out near that sign. Then, it shares the new status information with the directly neighboring exit signs. In the third step, it recalculates and saves the shortest distance value from the exit sign to the nearest exit. Finally, it orients the direction of the exit sign toward the directly neighboring exit sign on the shortest path. The validity of SIDSA has been tested by developing and running multiple tests using the simulator.

We are currently developing a physical smart exit module with SIDSA installed on it. SIDSA is expected to work reliably in a fire situation without creating concerns about the loss of a connection to the main server. However, more work is required to make the SIDSA-based smart exit-sign system practically useful. The scenarios for fire detection and the communication strategy or protocol for a fire situation need to be strengthened. Also, conformity of the system to local fire codes and regulations needs to be checked.

References

[1] Kobes M., Helsloot I., de Vries B., Post J. G., Oberijé N., and Groenewegen K., Way finding during fire evacuation; an analysis of unannounced fire drills in a hotel at night, *Building and Environment*, 45 (3):537-548, 2010.

[2] Choi J., "A study of the intelligent evacuee guidance of pictogram," Master, Graduate School of Urban Sciences, University of Seoul, 2013.

[3] Lee T.-K., "Development of a Mobile-based Fire Evacuation System using a Wireless Network," Master's Degree, Architectural engineering, Yonsei University, *Yonsei University*, 2013.

[4] Cho J., Lee G., Won J., and Ryu E., Application of Dijkstra's Algorithm in the Evacuation System Utilizing Exit Signs, *The International Symposium on Automation and Robotics in Construction and Mining (ISARC 2014)*, pages 224-229, Sydney, 2014.

[5] Kim D.-O., Mun B.-R., Lee K.-Y., Kim D.-W., Gil H.-J., Kim H.-K., and Chung Y.-S., The development of the escape light control system, *Journal of the Korean Institute of IIluminating and Electrical Installation Engineers*, 23 (6):52-58, 6 2009.

[6] Kim Y.-W., Boo J.-P., and Kim D.-O., A Scheme of Active Refuge Guide Based on

Dijkstra"s Algorithm, *Proceedings of Korea Multimedia Society*, pages 341-345, 2008.
[7] Shen X., Yu H., Buford J., and Akon M., *Handbook of Peer-to-Peer Networking* volume, Springer-Verlag, 2010.