# Simulating Evacuation Plan of Fire Disaster Based on Cloud Surveillance System

Jong-Yih Kuo, Tai-Yu Lai and Ping-Chi Lee
*Department of Computer Science and Information Engineering,*
*National Taipei University of Technology, Taipei, Taiwan*
*jykuo@ntut.edu.tw*

***Abstract*—In the past, most of the fire detection is temperature sampling, but temperature abnormality may cause a false alarm. Past studies of fire detection are causing the alarm, people who want to exit the building can only escape with the exit signal light. It cannot be based on the situation to build the escape route for the people. Therefore, a simulated evacuation planning of fire disaster base on real-time cloud surveillance system is built in this study. The system monitors whether there is a fire in the video. If the system finds the fire, it makes the file and writes the flame spread information immediately. The users install the positioning application and the simulate application made in this study. First, the positioning application will locate user's device and download the file which contains the information of fire continuously. If the positioning application finds the information, it will send the Intent to turn on the simulation application and simulate the situation immediately. According to the information of fire, the proposed application will recommend the escape route to users in real time, and let users exit as soon as possible. The result is that the time of detecting fire is 0.004 seconds, the time of completing the escape route is 0.004 seconds. Finally, the accuracy is 99.9%.**

***Index Terms*—A* Algorithm; Cloud; Escape Route; Fire Detection; Pathfinding; Sketch UP 3D; Unity 3D; 3D Modeling.**

## I. INTRODUCTION

With the modernisation of the building, there are many buildings in our lives. Fire disaster always occurs suddenly when people work or study. However, in order to escape from the complicated building, people have to familiar with the structure of the building and sure whether the front route is safe. According to the statistics of the National Fire Agency, Ministry of the Interior, the times of fire disaster in different places is shown in Table 1. We found that the times of buildings are more than the others

Table 1
The Times of The Different Places

| | Times | Percentage |
|---|---|---|
| Building | 1242 | 72.9 |
| Vehicle | 234 | 13.7 |
| Forest | 60 | 3.5 |
| Ship | 10 | 0.6 |
| Other | 158 | 9.3 |
| Total | 1704 | 100 |

Past studies are temperature sampling [1]. If the system detects the temperature abnormality, it will cause the alarm, but only the exit signal light can recommend people escaping. If the fire is in the front, people will be in danger. To sure people can escape from the building, monitoring the fire in real time and recommending people how to escape is beneficial to reducing damage.

To solve the problem, a simulate evacuation planning of fire disaster base on real-time cloud surveillance system is built in this study. Setting cloud platform, monitoring the fire, and simulating the scene by using Sketch Up 3D and Unity 3D.

Based on a cloud platform, if the monitor program detects a fire, it will create the information of the fire and notify users to escape from the building immediately. The escape route is made by 3D model and pathfinding technique. The system displays the structure of the buildings and computes the recommended route that helps users escape from the buildings.

The remainder of this paper is organised as follows: Section 2 describes related background and research. Section 3 presents the operation and design approaches of the evacuation system. Section 4 reports the analysis results of the experiments, and Section 5 concludes this paper.

## II. RELATED WORK

### A. Fire Detection

The fire detection method of the past studies is based on the colour, but it cannot apply to every scene [2-7]. The other method is using mask and edge with mask extraction and background subtraction [8]. However, background subtraction cannot apply to the complex background. The fire is a light source so that it will cause the more problem. In the past, it targeted the big fire. It must have clear edge or colour so the system can detect it. The above method is unfavourable to relieve the victims of a disaster.

### B. Modelling Software

Maya is a 3D modelling software. It created by Adobe. It has lots of function of 3D animation creating. It is good at the design of light and material, art designing, and movie effect [9].

3D MAX is also a 3D modelling software. It created by Autodesk. It is good at the transformation of the structure. It applies to the majority of departments [10].

Sketch Up 3D is also a 3D modelling software. It has a simple user interface so everyone can become familiar with it quickly. Qiao's research shows that Sketch Up 3D has six features: design concept, space design, a global design, real-time rendering, usability, and data interchange [11].

Trimble 3D Warehouse provides 3D models for users to download. Users can also upload their 3D models to share other users [12].

Unity is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, mobile devices and websites. It can publish to multiple platforms. It has particle system so that we can make the fire effect quickly [13].

Riberio’s research presents a serious game developed using Unity3D to recreate a virtual fire evacuation training tool. Results have shown that users effectively end up learning some evacuation procedures from the activity, even if only to look for emergency signs indicating the best evacuation paths. It was also evidenced that users with higher video game experience had a significantly better performance [14].

### *C. Pathfinding*

Dijkstra algorithm was published by Dijkstra in 1959. It uses Breadth-first search to produce a shortest-path tree [15-20]. Although Dijkstra algorithm can find the shortest path, its efficiency is low. To improve its efficiency, A* algorithm is published.

The number of nodes computed by A* algorithm is less than Dijkstra algorithm, so A* algorithm can find a route earlier. The efficiency of A* algorithm is better because it has a heuristic estimate. It avoids some nodes to have better efficiency.

A* algorithm is based on the current node and uses a formula to compute the points of the neighbour nodes. It puts the neighbour nodes in the open list and selects the node with the smallest points to be the current node until it finds the goal node [19-26]. The formula is shown as follows.

$$F(\mathrm{n}) = \mathrm{G}(\mathrm{n}) + \mathrm{H}(\mathrm{n}) \tag{1}$$

n is the current node, F(n) is the total of the current points and heuristic points. G(n) is the current points. H(n) is the heuristic points.

Heuristic estimate formula is to use Manhattan Distance. Its formula is shown as follows.

$$\mathrm{H}(\mathrm{n}) = | x_1 - x_2 | + | y_1 - y_2 | \tag{2}$$

The $x_1$ is the x value of the first coordination. The $x_2$ is the $x$ value of the second coordination. $y_1$ is the $y$ value of the first coordinates. $y_2$ is the y value of the second coordinate. In other words, the distance is considered by the vertical and horizontal distance.

## III. System Design

### *A. Fire Detection*

The fire detection method does not use all of the videos. It will capture the changing area and do a thumbnail. It gets the edge of the fire by using Sobel filter. It gets the texture of the fire by using Flame texture. It combines the images and input to next step.

It uses YCbCr, RGB, and HSV to filter the image and gets the range of the fire. We can get some candidate flame regions. Finally, the stable regions will be filtered because the fire region is unstable. This study implements the fire detection system is shown in Figure 1.

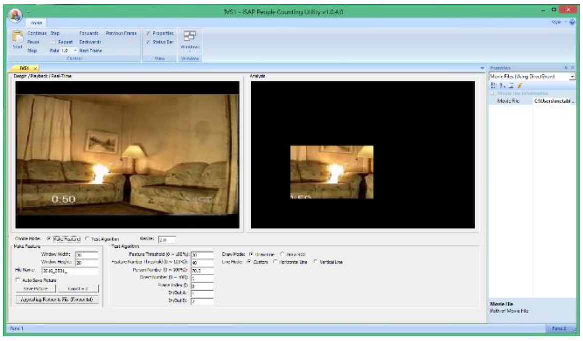

Figure 1: The fire detection program

It can choose how to detect the fire in the right area. The figure is to choose to detect the video. We select one video and start playing; the program will start detecting the fire. If it detects the fire, it will create the information of the fire.

### *B. Cloud Platform*

We choose one computer as a server and execute the fire detection program to surveillance the video. We put the information of the fire in the cloud, too.

Users’ intelligent devices should connect to the network so that can download the information of the fire. It can promote users’ convenience. The program will download the information and give it to the simulation application to use.

### *C. Building Modeling*

This study use Sketch Up 3D to make the building of National Taipei University of Technology. We consider every floor plan and make its structure. It can help users how to escape.

Floors are classified into 1F, 2F-10F, and 11F-16F. There are not classroom on the 1F, so we consider the location of exits and stairways. There is rectangle and semicircle structure on the 2F-10F. There are only rectangle structures on the 11F-16F. After completing every floor, the stairways are created to connect every floor. The stairway is modelled as a slope to reduce the computation time, and the user can move down smoothly. Each stairway contains two slopes and one plane. Two stairways are put in two exits in every floor.

To simplify tasks, we do not create the desks and chairs in the 3D model. The completing 3D model of the building is shown in Figure 2.

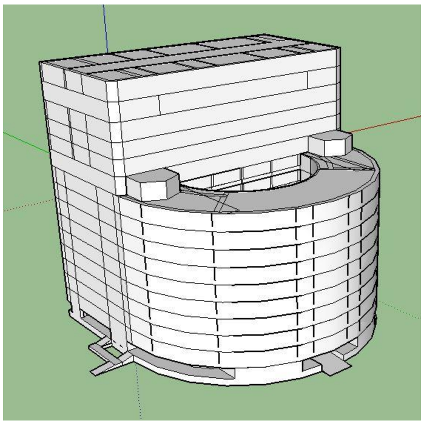

Figure 2: 3D Model of the building

### *D. Pathfinding*

A* algorithm makes the escape route. A* algorithm is conformed to our research. This study applies C# to implement the A* algorithm. The C# and Unity 3D is used to implement the proposed system. Import the 3D model to the project and make the terrain.

To use A* algorithm, we have to define Node, Node List, and Obstacle first. We use the optimisation A* algorithm in this research. We add Dot Product in A* algorithm[26]. The vector in the composition of the start node and the goal node is named SG. The vector in the composition of the start node and the neighbour node is named SN. If the dot product between SG and SN is less than zero, we put the neighbour node in Negative Direction List. Otherwise, if the dot product between SG and SN is greater than zero, the algorithm computes continuously.

The vector in the composition of the current node and the goal node is named CG. The vector in the composition of the current node and the neighbour node is named CN. If the dot product between CG and CN is less than zero, we put the neighbour node in Reverse Move List. Otherwise, if the dot product between CG and CN is greater than zero, the algorithm computes its points. The method can reduce the computes of some nodes. It can improve the effect of A* algorithm. The escape route is shown as the Figure 3.

When the system detects the fire, users may be in the different classroom. The system will provide the buttons for users to choose. The users have to choose the floor first, and then they choose the classroom. Then, they can apply the proposed application to find a path to escape. The deciding position interface is shown in Figure 4.

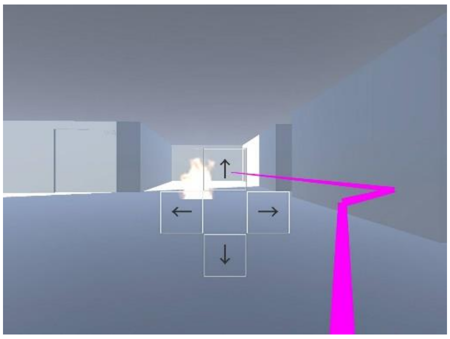

Figure 3: The escape route

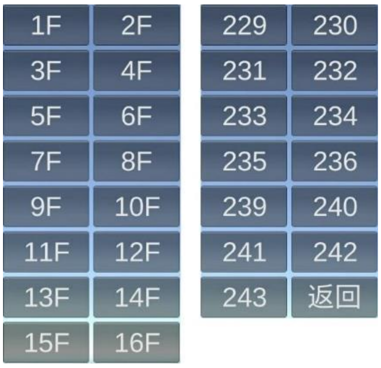

Figure 4: The deciding position interface

An Android application is implemented to create the buttons that users characters can move in this application. The control buttons of deciding position are shown as the Figure 5.

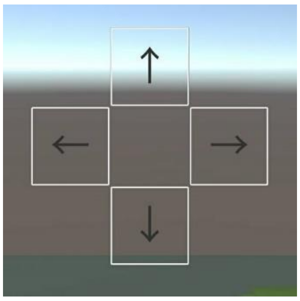

Figure 5: Moving buttons

The system will download the information of the fire continuously. If the file is not the same as previous one, the system will refresh the scene to add new fire. The escape route is made every second because the fire will be found every time. The escape route can avoid the fire the system finds.

An optimisation A* algorithm is not good at the three-dimensional scene. The system will detect the user character’s height and decide the floor number. The system will open an optimisation A* algorithm on the floor. It can reduce its computes.

The system decides the exit position with four detections. First, the system will detect the stairways of the two sides. If the right side is in danger, the system will set the exit on the left side unconditionally. If the two sides are safe, it will enter the next phase.

Second, the system will detect whether there is on fire in the corridor. If the left side of the corridor is on fire, the system will set the exit as the right side unconditionally. If the corridor is safe, it will enter the next phase.

Third, the system will detect the stairways of the other floors. If the right side of the other floors is in danger, the system will set the exit on the left side unconditionally. If the other floors are safe, it will enter the next phase.

Finally, the system will detect the distance between the user character and the floor exit. The system will set the exit as the side which has lower distance. The deciding exits diagram is shown in Figure 6.

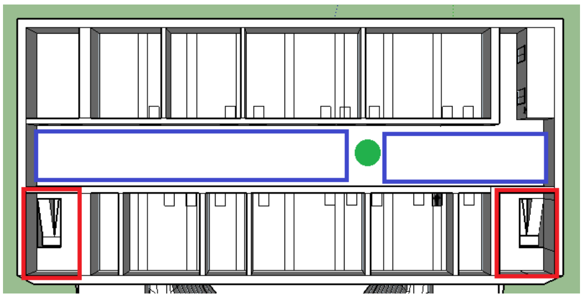

Figure 6: Deciding exits

### *E. Positioning application*

The positioning application is made in this research. When the users open the application, it will display the current location with google map. The application will download the information of the fire continuously. If the file has the information of the fire, it will send the Intent to turn on the simulation application. The image of the positioning application is shown in Figure 7.

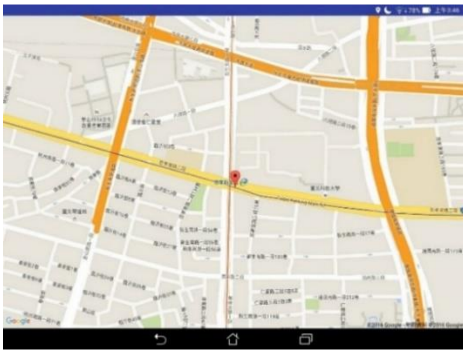

Figure 7: Positioning application

## IV. Experiment Results

This section will introduce the experiment about working time test, offline test, and pressure test. ZenPad S 8.0 is chosen as the device. Working time test is about that we record the time of detecting the fire, getting the file, and completing the escape route. The offline test is about that the device is offline and whether the application still completes the escape route. The pressure test is about simulating 1000 clients to download the file and compare whether its information is correct.

### *A. Working Time Test*

We choose the video about the room on fire as the image (shown in Figure 1). Its fps is 30.00, and its resolution is 720x480. The result is that the time of detecting the fire is 0.004 seconds, the time of getting the file is 0.001 seconds, and the time of completing the escape route is 0.004 seconds. We test it 100 times and the average total time is 0.020 seconds.

### *B. Offline Test*

The network in the fireplace may be unstable. We disconnect the network after getting the file to test whether the system still completes the escape route. We test it 100 times. The system still completes the escape route in the application.

### *C. Pressure Test*

We simulate 1000 clients to download the file in 0.020 seconds. If the file is same as its information, it is defined correctly. Otherwise, it is defined false. As a result, the number of correct is 999, and the number of false is 1.

## V. Conclusion

A simulate evacuation planning of fire disaster base on real-time cloud surveillance system is built in this study. We set cloud platform on the server and establish the monitor program. If the program detects the fire, it will create the information of the fire and notify users to escape from the building immediately. 3D model and pathfinding technique make the escape route. The system displays the structure of the buildings and computes the recommended route that helps users escape from the buildings.

## Acknowledgement

This work was supported by the Ministry of Science and Technology under grant numbers MOST 104-2221-E-027-116.

## References

[1] B. C. Arme, A. Ollero, J. R. Matinez de Dios, “An intelligent system for false alarm reduction in infrared forest-fire detection”, *IEEE Intelligent Systems and their Applications*, vol. 15, pp. 64-73, 2000.

[2] S. Wang, D. Jeng, M. Tsai, “Early fire detection method in video for vessels”, *IEEE Conference on Software Engineering Education and Training*, vol. 82, pp. 656-667, 2009.

[3] M. Stula, D. Krstinic, L. Seric, “Intelligent forest fire monitoring system*”, Information Systems Frontiers*, vol. 14, pp. 725-739, 2012.

[4] S. Ham, B. Ko, J. Nam, “Fire-Flame Detection Based on Fuzzy Finite Automation”, 20th International Conference on Pattern Recognition, pp. 3919-3922, 2010.

[5] M. Wirth, R. Zaremba, “Flame Region Detection Based on Histogram Backprojection”, Canadian Conference on Computer and Robot Vision, pp. 167-174, 2010.

[6] N. Yang, H. Wang, Q. Zhang, “Adaptive flame image detection algorithm”, *3rd International Congress on Image and Signal Processing*, vol. 4, pp. 1787-1791, 2010.

[7] S. M. Razmi, N. Saad, V. S. Asirvadam, “Vision-based flame detection: Motion detection & fire analysis”,*IEEE Student Conference on Research and Development*, pp. 187-191, 2010.

[8] J. Zhu, Y. Lao, Y. F. Zheng, “Object Tracking in Structured Environments for Video Surveillance Applications”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 223-235, 2009.

[9] Maya, http://www.autodesk.com.tw/products/maya/features/all, Accessed May 3 2016.

[10] 3DS MAX, http://www.autodesk.com.tw/products/3ds-max/features/all , Accessed May 3 2016.

[11] L. Qiao, S. Zheng, Y. Wang, L. Zhang, “Application of Sketchup in the Teaching of Landscape Design of Undergraduate College”, *International Conference on Optics, Photonics and Energy Engineering*, vol. 2, pp. 306-308, 2010.

[12] 3D Warehouse, https://3dwarehouse.sketchup.com/, Accessed April 25 2016.

[13] Unity, http://unity3d.com/unity, Accessed May 6 2016.

[14] J. Ribeiro, J. E. Almedia, R. J. F. Rossetti, A. Coelho, A. L. Coelho, “Using Serious Games to Train Evacuation Behaviour”, *7th Iberian Conference on Information Systems and Technologies*, vol. 7, pp. 1-6, 2012.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms* (Third ed.), Cambridge: MIT Press, pp. 658-664, 2009.

[16] N. Jasika, N. Alispahic, A. Elma, K. Ilvana, L. Elma, N. Nosovic, “Dijkstra's shortest path algorithm serial and parallel execution performance analysis”, *Proceedings of the 35th International Convention*, pp. 1811-1815, 2012.

[17] M. Wei, Y. Meng, “Research on the optimal route choice based on improved Dijkstra*”, IEEE Workshop on Advanced Research and Technology in Industry Applications*, pp. 303-306, 2014.

[18] D. Fan, P. Shi, “Improvement of Dijkstra's algorithm and its application in route planning”, *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4, pp. 1901-1904, 2010.

[19] K. Khantanapoka, K. Chinnasarn, “Pathfinding of 2D & 3D game real-time strategy with depth direction A∗ algorithm for multi-layer”, *Eighth International Symposium on Natural Language Processing*, pp. 184-188, 2009.

[20] R. Anbuselvi, “PathFindingSolutionsforGrid Based Graph”, *Advanced Computing*, vol. 4, pp. 51-60, 2013.

[21] X. Cui, H. Shi, “A*-based Pathfinding in Modern Computer Games”, *International Journal of Computer Science and Network Security*, vol. 11, pp. 125-130, 2011.

[22] J. Hu, W. Wan, X. Yu, “A Pathfinding Algorithm in Real-time Strategy Game based on Unity3D”, *International Conference on Audio, Language and Image Processing*, pp. 1159-1162, 2012.

[23] G. E. Mathew, G. Malathy, “Direction based heuristic for pathfinding in video games”, *2nd International Conference on Electronics and Communication Systems*, pp. 1651-1657, 2015.

[24] J. Stamford, A. S. Khuman, J. Carter, S. Ahmadi, “Pathfinding in partially explored games environments: The application of the A∗ Algorithm with occupancy grids in Unity3D”, *14th UK Workshop on Computational Intelligence*, pp. 1-6, 2014.

[25] M. Wang, H. Lu, “Research on Algorithm of Intelligent 3D Path Finding in Game Development”, *International Conference on Industrial Control and Electronics Engineering*, pp. 1738-1742, 2012.

[26] Z. Zhao, R. Liu, “A optimization of A* algorithm to make it close to human pathfinding behavior*”, International Conference on Electrical, Computer Engineering and Electronics*, vol. 2, pp. 708-714, 2015.