Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

RFES: a real-time fire evacuation system for Mobile Web3D*

Feng-ting YAN[†1], Yong-hao HU[2], Jin-yuan JIA[†‡2], Qing-hua GUO[3], He-hua ZHU[3], Zhi-geng PAN[4]

[1]*School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China*

[2]*School of Software Engineering, Tongji University, Shanghai 201804, China*

[3]*School of Civil Engineering, Tongji University, Shanghai 200092, China*

[4]*Research Center of Digital Media and Human-Computer Interaction, Hangzhou Normal University, Hangzhou 310013, China*

[†]E-mail: yanfengting2008@163.com; jiajy@tongji.edu.cn

Received Oct. 7, 2017; Revision accepted Mar. 9, 2018; Crosschecked July 3, 2019

Abstract: There are many bottlenecks that limit the computing power of the Mobile Web3D and they need to be solved before implementing a public fire evacuation system on this platform. In this study, we focus on three key problems: (1) The scene data for large-scale building information modeling (BIM) are huge, so it is difficult to transmit the data via the Internet and visualize them on the Web; (2) The raw fire dynamic simulator (FDS) smoke diffusion data are also very large, so it is extremely difficult to transmit the data via the Internet and visualize them on the Web; (3) A smart artificial intelligence fire evacuation app for the public should be accurate and real-time. To address these problems, the following solutions are proposed: (1) The large-scale scene model is made lightweight; (2) The amount of dynamic smoke is also made lightweight; (3) The dynamic obstacle maps established from the scene model and smoke data are used for optimal path planning using a heuristic method. We propose a real-time fire evacuation system based on the ant colony optimization (RFES-ACO) algorithm with reused dynamic pheromones. Simulation results show that the public could use Mobile Web3D devices to experience fire evacuation drills in real time smoothly. The real-time fire evacuation system (RFES) is efficient and the evacuation rate is better than those of the other two algorithms, i.e., the leader-follower fire evacuation algorithm and the random fire evacuation algorithm.

Key words: Fire evacuation drill; Building information modeling (BIM) building space; Mobile Web3D; Real-time fire evacuation system based on ant colony optimization (RFES-ACO) algorithm

https://doi.org/10.1631/FITEE.1700548 **CLC number:** TP391

1 Introduction

Crowd evacuation safety has become a hot topic in the field of urban public safety research (Pluchino et al., 2013; Song et al., 2013; Yan et al., 2019). In subway stations, people are crowded, the number of evacuation exits are insufficient, discharge of exhaust is problematic, and the structure of the scene is complex, all of which can lead to damage from a high temperature and toxic fumes in evacuation scenarios. Once a subway station is on fire, the panic and confusion of the crowd usually worsen the consequences of accidents, or even cause secondary accidents or disasters (Cha et al., 2012). Effective crowd evacuation and hazard avoidance skills play important roles in mitigating dangers in urban public areas. Large-scale fire escape drills require considerable manpower, materials, and financial resources; however, they also present the potential for danger. Thus, virtual reality is an effective approach to conducting fire evacuation drills for large-scale scenes (Kinateder et al., 2014a). The purpose of such drills is to ensure that people can make it to a safety zone within the

‡ Corresponding author

* Project supported by the Key Research Projects of the Central University of Basic Scientific Research Funds for Cross Cooperation, China (No. 201510-02), the Research Fund for the Doctoral Program of Higher Education, China (No. 2013007211-0035), and the Key Project in Science and Technology of Jilin Province, China (No. 20140204088GX)

ORCID: Feng-ting YAN, http://orcid.org/0000-0002-0163-3016

possible evacuation time, thereby reducing casualties. The artificial intelligence (AI) (Martinez-Gil et al., 2015; Wang et al., 2019) technology is well suited to make the best decisions for a crowd (Li et al., 2017). Multi-agent simulations model a crowd as it makes its way along a planned path in a fire evacuation environment. Multi-agent evacuations based on the Web can assist the public in such drills anytime and anywhere. The Web3D technology has been envisioned (Barsoum and Kuester, 2005), especially that on mobile platforms, which forms the Mobile Web3D technology (Athanasis et al., 2015). Research in this domain has broad applications and provides knowledge for computer experts, architectural engineers, fire science experts, social scientists, planning experts, and so on (Chu and Wu, 2011).

The Mobile Web3D evacuation drill system has broad application areas and significant value. The implementation of this system has many benefits for crowd evacuation fire drills: (1) It is a lightweight, high-fidelity escape drill on the mobile terminal Web that consumes a small amount of power and enables people to participate in evacuation training in a three-dimensional (3D) virtual scene through online visualization (Wang et al., 2015); (2) It allows people to participate in evacuation drills and training anytime, anywhere, via a webpage on a mobile terminal device, and gain experience, which results in extensive data for researchers to further study people's behavior during an evacuation; (3) It saves financial and material resources and can also avoid potential casualties and property loss; (4) It can effectively and conveniently assist in crowd evacuation and help command staff develop evacuation procedures.

However, there are numerous challenges in the research and development process for a Mobile Web3D escape drill system, and the main issues are as follows: (1) establishing a large-scale escape scene on the Mobile Web3D; (2) calculating the dynamic diffusion process of real smoke in a fire scenario; (3) planning an optimal path for the escape of multiple agents. This Mobile Web3D escape drill system provides the following solutions for these problems: (1) It creates a lightweight building information modeling (BIM) big scene database, which increases the reuse of the components in structured scenes and spatial structuring, and reduces the number of special components while forming a space-connected graph; (2) It conducts a pretreatment calculation for the dynamic diffusion of flue gas using a fire dynamic simulator (FDS) based on specific scenarios, and applies the calculated results to the Mobile Web3D escape drill system, thus greatly reducing the computational cost related to flue gas on a webpage; (3) Based on a pheromone reuse method, we propose the real-time fire evacuation system based on the ant colony optimization (RFES-ACO) algorithm to plan the best escape route for multiple agents, multiple starting points, and multiple exits, while visualizing people's escape progress during a fire. Through an investigation of evolution algebra, the RFES-ACO algorithm achieves an optimal path planning after convergence. The RFES-ACO algorithm is efficient and the evacuation rate is better than those of the other two types of algorithms, i.e., the leader-follower fire evacuation algorithm (Zheng et al., 2014; Yuan et al. 2017) and the random fire evacuation algorithm (Tian et al., 2014). The notations used in the RFES-ACO algorithm are shown in Table 1.

There are three innovations in this study: (1) The RFES-ACO algorithm is proposed; (2) A lightweight

Table 1 Notations used in the RFES-ACO algorithm

Variable	Description
m	Ant quantity
N_c	Number of path-finding iterations
t=0	Initialization time
α	Relative importance of trajectories
β	Relative importance of visibility
ρ	Persistence of trajectories, $0 \le \rho \le 1$
$\tau_{ij}^{\alpha}(t)$	Information value left after each move
$\eta_{ij}^{\beta}(t)$	Setting of the heuristic parameter, $\eta_{ij}^{\beta}(t)$ = Dd+Sd+Sh
Dd	Reciprocal value of the distance to the escape door
Sd	Reciprocal value of poisonous smoke density per unit area
Sh	Reciprocal value of heat calories per unit area
$\Delta\tau_{ij}$=0	A unit length trajectory pheromone left by ant k on edge (i, j)
P_{ij}^{k}	Probability of choosing the grid point for the next step
(x_0, y_0)	Location of ants equal to the location of escaping people
Δt	Cycle of collecting poisonous smoke data in the 3D space
$\rho_{\min}$	Minimum pheromone concentration
$b_i(t)$	Number of ants located at the feasible point i at time t, and $m = \sum_{i=1}^{n} b_i(t)$
$\upsilon_{ij}^{\gamma}(t)$	Visibility of edge (i, j)

smoke algorithm is proposed; (3) The RFES evacuation system is proposed and completed.

In this study, the RFES evacuation system, based on Mobile Web3D and supporting HTML5, uses the efficient 3D programming framework of a three.js page, applies multithreading technology, uses lightweight compression technology, and finally realizes this system platform on the Mobile Web3D. Three databases are built in this study: (1) a lightweight scene database; (2) a lightweight smoke database; (3) a multi-morphological role database. The RFES-ACO algorithm is a dynamic adaptive ant colony algorithm that is based on a genetic algorithm. RFES uses the browser/server (B/S) structure, graphics processing unit (GPU) rendering, and a series of lightweight solutions to realize a real-time fire evacuation training system.

2 Related work

Visualization in current mobile device graphics has developed rapidly (Guest et al., 2015), and many studies focus on the basic hardware GPU structure and computing method. In mobile platforms, authenticity and real-time scene rendering have many basic research problems. Humayoun et al. (2015) manipulated the time constraints of the traditional fuzzy algorithm, and proposed a smoothing algorithm based on a summed-area table, improving the overall graphics rendering speed.

Current fire escape systems are based on the present mobile graphics technology, and integrate architectural engineering, fire science, and social psychology in planning escape strategies and visualizing escape behaviors. System software qualified for this function uses EXODUS emergency simulation software and the Legion series of software (Kinateder et al., 2014b; Qin et al., 2014). At present, the main software calculations are carried out on PC terminals, and the webpage application is also on a PC terminal and cannot avoid the restriction of installing plug-ins. Currently, there is no system software on mobile terminal devices and no Mobile Web3D escape software apps have been available to date (Onorati et al., 2014).

Path planning is the core of fire evacuation. Cha et al. (2012) used statistical methods to roughly calculate the flue gas concentration on a fire evacuation path based on the existing pathways, so as to guide people to choose the safest path for evacuation. However, the path planning considerations for smoke were not fine enough. Based on the specific attributes of each agent, Tian et al. (2014) implemented evacuation path planning for a large-scale population. However, flue smoke was not considered a dynamic obstacle in optimal path planning. Lin et al. (2013) used a quick matching method for path planning, but the method was not accurate enough for each agent. Fang et al. (2011) proposed a panicked crowd evacuation process, and the behavior characteristics were blind running and random evacuation. Duo et al. (2016) proposed a greedy evacuation algorithm whereby people tended to escape towards sparsely populated areas.

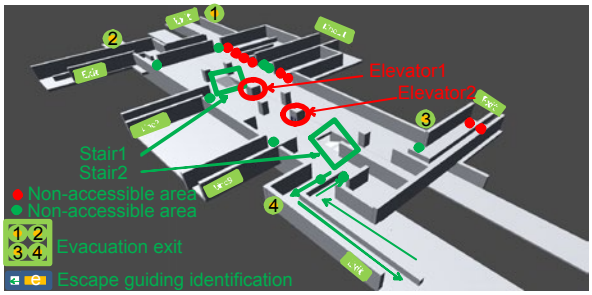
3 Construction of a lightweight scene

The lightweight architectural scene technology is perfect (Yan et al., 2019). In this study, we use a lightweight compression technique in the subway station model.

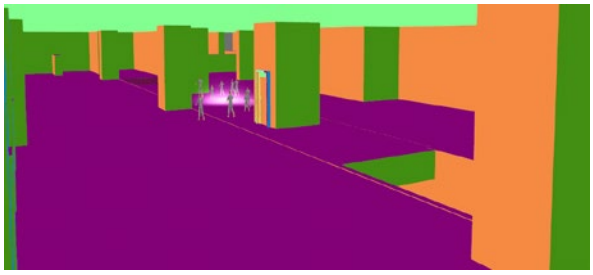
Based on fine-grained scene processing, the scene index structure is reconstructed, and the scene data are compressed. Then interest-driven scheduling, based on the BIM components, completes the on-demand scheduling. The method is called the interest-driven lightweight preprocessing (IDLP) technology. IDLP reduces the size of the BIM raw data by removing redundant and duplicate data, and structures the lightweight data to form a hierarchical model, which meets users' demands and interests. This kind of scheduling is highly relevant and the utilization of the scene module is high. It can meet the needs of real-time rendering on the Mobile Web.

In the whole scene database, three typical metro station scene models are adopted in our simulation to describe RFES for public fire evacuation. Scene 1 is a rectangular metro station model, scene 2 is a curved spatial structure metro station model, and scene 3 is a large-scale, multistory metro station model. The IDLP technology is a divide-and-rule method that can be used in any large-scale complex scene mode, and the renderings of the rendering scene module are according to the interests of the user. As a result of the

IDLP technology, the complex large-scale scene data can be implemented in real-time online transmission and real-time rendering on the Mobile Web. Fig. 1 shows a popular rectangular metro station model, which is called scene 1.



(a)



(b)

Fig. 1 Scene graph of a rectangular metro station (scene 1): (a) aerial view of the metro station; (b) internal perspective of the metro station

This Mobile Web3D escape drill system first builds a lightweight Web3D space model based on the BIM information. After lightweight processing of the BIM scene model, progressive loading of online transmission, and progressive rendering based on frustum movement, the scene is finally shown on Mobile Web3D.

We take the metro station as a case study to create a fire escape drill system. Aerial and internal views of the subway station scene are shown in Fig. 1.

Fig. 2 shows a metro station model with a complex, special structure (scene 2). There are more obstacles and the smoke spreading calculation is more complex than that in scene 1.

In scene 2 (Fig. 2), there are seven gates and one elevator. Gate 7 will be open if a fire starts in the subway station. The seven gates will be used for crowd evacuation.

Fig. 3 is a complex large-scale, multistory metro station. This lightweight scene 3 can be shown on

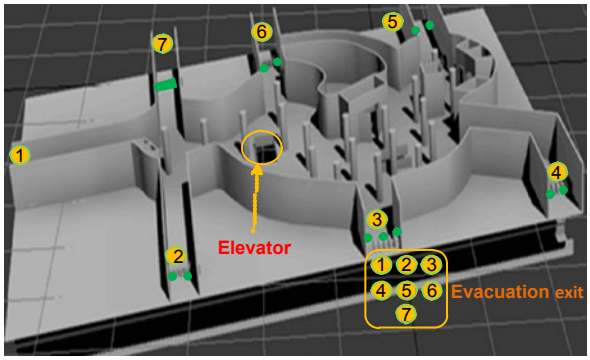


Fig. 2 A curved spatial structure metro station (scene 2)

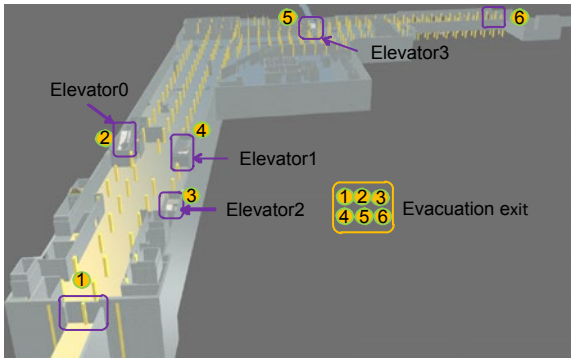


Fig. 3 Large-scale metro station scene (scene 3)

Mobile Web3D. There are six evacuation exits, two big gates, and four elevators.

4 Construction of the scene map

Path planning for public escape from a building fire requires an accurate scene map, which is constructed from the static scene map and the dynamic smoke-spreading obstacle map. The accurate scene map is dynamically changing according to the fire scenario. In our simulation, the dynamic accurate scene map is called the DAS map. The raster method is adopted for the DAS map, and the dynamic spread of the smoke is described in Section 5, which explains how the voxel smoke data correspond with the DAS map. The DAS map is used for path planning in the RFES-ACO algorithm, which is accurately described in Section 6. In our simulation, the length of a voxel block is 0.25 m, so the length of a planar DAS map mesh is also 0.25 m.

Figs. 4 and 5 show the plane maps of the second and first floors underground of the metro station space for scene 1, respectively. Fire zone appears in the

second floor and the first floor is the transfer layer. Exits 1–4 are for crowd evacuation.

From Fig. 4 we can see that there are walls, stairs, bearing columns, and elevators. Here, we define the walls and bearing column as obstacles on the two-dimensional (2D) map. The elevators will be disabled when a fire starts, so the elevators are all defined as obstacles. The fire regulations are also used in Fig. 5. Fig. 6 shows the legend used in the map.

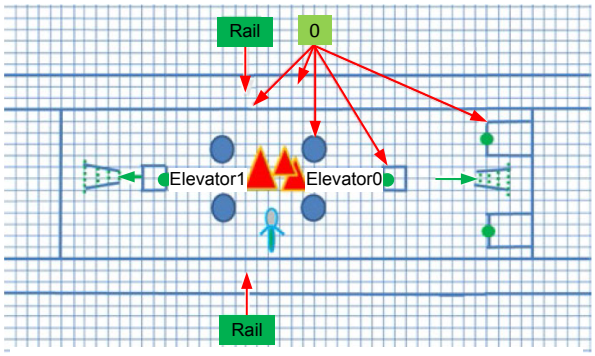


Fig. 4 The fire zone on the platform floor

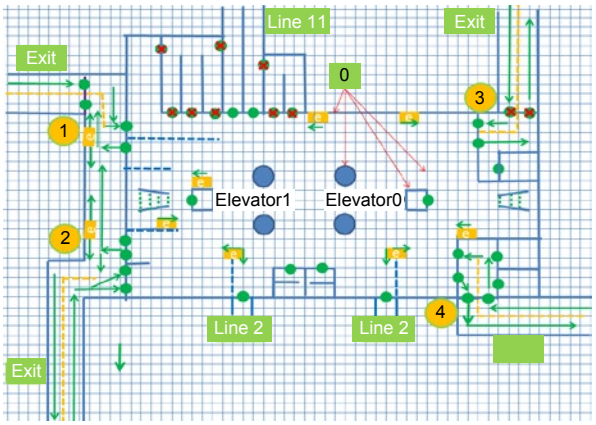


Fig. 5 The transfer layer of the metro station

5 Lightweight FDS smoke data

In this section, we use the FDS to calculate the data on spreading smoke in a building fire scenario (Fig. 7). All the doors are considered open since they will be opened for public evacuation when a fire starts.

The key work in this section is to make the smoke data lightweight. After calculating the smoke data (including smog data, toxic data, and heat data), all the data will be read and made lightweight.

To ensure the high accuracy of the fire escape

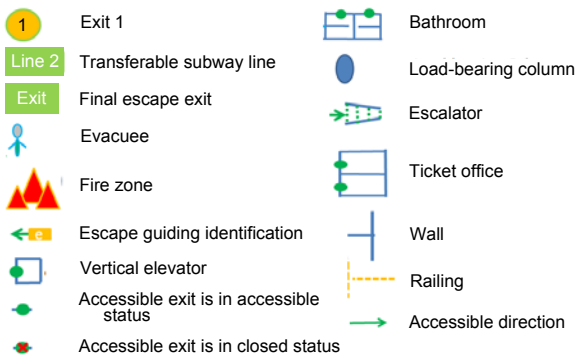


Fig. 6 Map legend

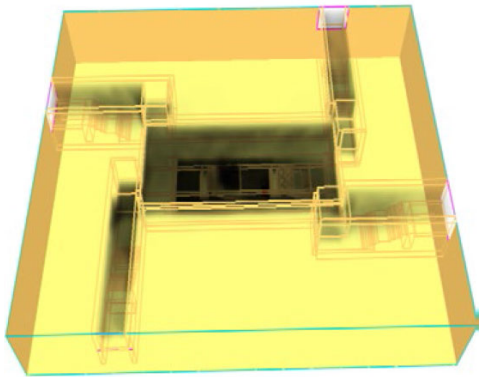


Fig. 7 Dynamic diffusion data for smoke scene voxelization based on the fire dynamic simulator

system, we calculate the flue gas with FDS to guarantee the accuracy of the dynamic diffusion of flue gas (Benkoussas et al., 2016).

After voxelization of the scene space, according to the value of the space voxel, we visualize the smoke in the voxel block. In the vertical direction, we count the number of smoke voxel blocks, and store the information in the corresponding 2D plane network as the heuristic value of path planning.

The toxic fumes and heat in the scene spread constantly and dynamically, which constantly changes the appropriate evacuation path and requires dynamic path planning. Dynamic path planning involves obtaining accurate poisonous smoke and heat data as heuristic information, combining these data with several optimal paths already planned out and then constantly updating the path until the iteration is completed, and proposing the optimal path. When path planning has been completed, the poisonous smoke and heat data are read in new frames, and a new round of path planning calculations is started.

The key technologies involved in making the smoke data lightweight are shown in Fig. 8.

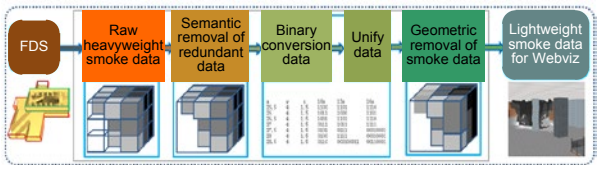


Fig. 8 Making smoke data lightweight

Smoke rises to ceiling and spreads around the ceiling when there is a fire. The dynamic spread of smoke is huge. To visualize a real-time smoke scenario, we address key layer smoke (KLS) driven rendering. KLS-driven rendering is based on a key layer of 1.6 m, which affects the evacuees' sight and breath. If there is smoke in a 1.6-m voxel, then other voxels that are higher than 1.6 m are filled with smoke, for the smoke spreads around the ceiling. Therefore, in the rendering process, the GPU can render one smoke rectangle, which reaches the ceiling. The KLS-driven rendering technology renders one smoke rectangle instead of 14 voxels (there are 14 voxels from ceiling to the 1.6-m voxel position). In this way, the cost of rendering is reduced by 13/14 times.

Fig. 9 shows the smoke-covered first floor underground after the second floor underground is on fire. The scene is a visualized rendering realized on a Mobile Web3D terminal.

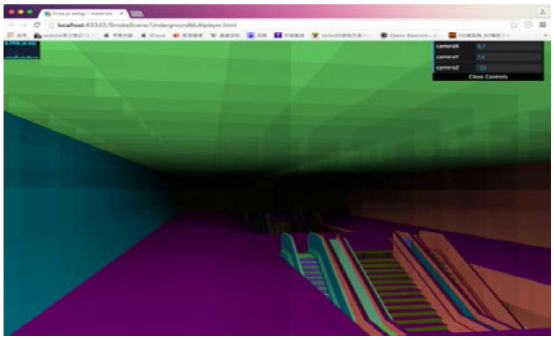


Fig. 9 Subway station filled with smoke

Using the KLS-driven rendering technology, in a 50-frames/s fire evacuation scenario, the time required for smoke rendering is less than 7 ms.

6 Lightweight peer-to-peer transmission

As we know, there is a general bottleneck in traditional centralized network strategies, which underuses the computing resources. At present, peer-to-peer (P2P) technology is simulated, and there is no actual software available. However, the Web real-time communication (WebRTC) technology has P2P transmission capability, and can effectively use nodes and bandwidth resources. Thus, it can be developed for virtual scenes.

The technical route for lightweight transmission is shown in Fig. 10.

In this study, the key P2P technology is Lite3DStream file transmission. The following steps cover the entire virtual scene transfer process:

1. The user accesses the resident proxy server via a Web browser. Usually, the user must pass authentication. After that, the metainfo files are uploaded, and the MagnetURL is provided to the user interface of the running proxy server with the files that are needed.

2. The proxy servers are filtered from all the torrent servers in the torrent server index based on the node network and CPU speed to find the server with the strongest computing performance. The proxy server transmits the metafile to the virtual P2P client in the same server or different servers, and then the virtual P2P client contacts the tracking server and requests the active node list.

3. The server is traced and the node list is sent to the proxy server.

4. The proxy server immediately returns the list of nodes containing the active node's IP address and port number to the user's device.

5. The data plane management and active node are running on the user's device, and both initializations are finished.

6. The data plane management requests file blocks from the active node, and the different file data blocks are downloaded onto the user's device. The data plane management can update and download the files at the same time, and the storage management in the user's device tracks the file blocks that have been downloaded.

7. The data plane manages the periodic exchange status messages so that the peers can perceive which chunks of data can be used.

8. The data plane management sends the activity information to the active node to confirm whether the open connection node is still active.

9. The data plane management sends status information to the tracking server, and enables the

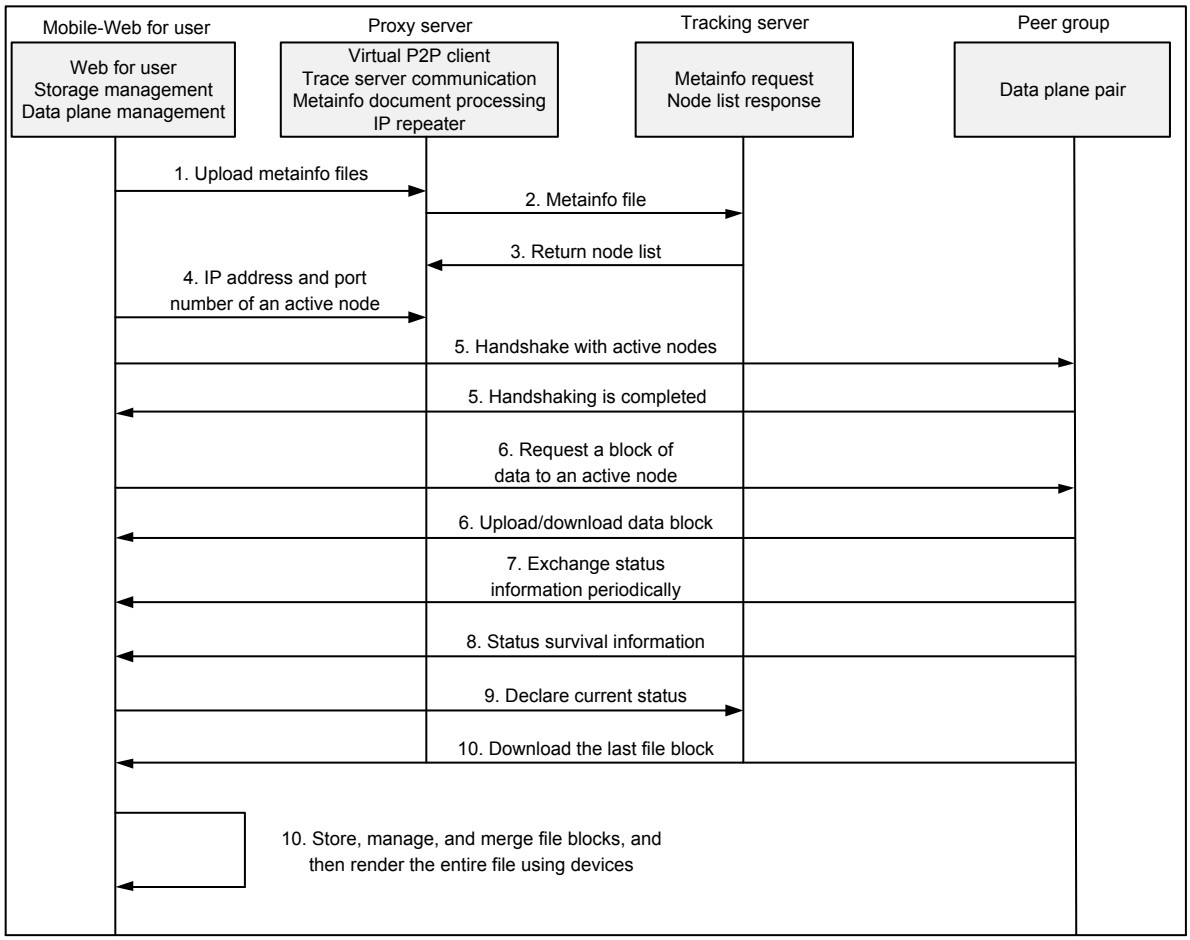


Fig. 10 Lite3DStream module transmission sequence diagram

tracking server to update the status of the user and the population.

10. Finally, after the last file block is downloaded, the storage management combines these blocks of files and generates a complete file, which is written onto a device. After a file is downloaded, if the user's Web browser is maintaining a working status, the user's machine will generate a seed of the file, which is shared with new users.

7 RFES-ACO escape path algorithm

The ant colony optimization algorithm determines path planning based on ant pheromones. Based on the ACO and genetic algorithms, and considering the static scene and the dynamic spreading of the smoke scenario, we propose the RFES-ACO algorithm. The following steps form the algorithm:

1. Map the walls, the pillars in the hall, the escalator obstacles, and the gates to the corresponding grid point positions in a 2D plane grid. If the grid is occupied by an obstacle, the grid is assigned "1;" otherwise, it is assigned "0."

2. Start path planning for the RFES-ACO algorithm. First, initialize constants and variables and set the parameter values: initialization time t=0, number of path-finding iterations N_c=0, the information value left after each move $\tau_{ij}^{\alpha}(t)=C$ (where i and j are the current and next positions of the ant, respectively), heuristic parameter $\eta_{ij}^{\beta}(t)=\mathrm{Dd}+\mathrm{Sd}+\mathrm{Sh}$, the initial value of the pheromone intensity increment $\Delta\tau_{ij}$=0, the ant quantity set to m=20 according to the number of communication points and the number of grids in the accessible area, and the location of ants (x_0, y_0) equal to the locations of escaping individuals.

3. A single ant looks for the path. Calculate the

distance d between the ant and the escape door, and $1/d$ is the heuristic value of the distance. In every Δt, collect poisonous smoke and heat data in the 3D space, map the data onto the corresponding plane grid, form the data for the corresponding concentration and heat, and convert the data into the reciprocals as the heuristic values. In this case, the heuristic pheromone concentration is 0.

Starting from the initial point, the algorithm determines the accessible grid locations in the four directions, and records the accessible grid in the open table. Each item in the table is the next point at which an ant can choose to walk. Then select one point from these four points according to transition probability.

4. The fourth step of path planning is to determine the value of the transition probability of four position items:

$$P_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\upsilon_{ij}^{\gamma}(t)}{\sum\limits_{s\in\text{allowed}_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)\upsilon_{is}^{\gamma}(t)}, & j\in\text{allowed}_k,\\ 0, & \text{otherwise},\end{cases} \tag{1}$$

where all the next grid points are denoted by k, the current grid points by allowed_k, and a specific grid point in allowed_k by s.

In the first iteration, $\tau_{ij}^{\alpha}(t)$ is not considered, and only the heuristic value of $\eta_{ij}^{\beta}(t)$ is considered, while the numerical value of $\eta_{ij}^{\beta}(t)$ is determined by η_{ij}. $\upsilon_{ij}^{\gamma}(t)$ is the visibility of edge (i, j). Parameters γ, α, and β are set to 2 based on the empirical values, and t refers to time.

The probability sum of the four directions is 1. If the grid is occupied by an obstacle, the corresponding walking probability value is given to other accessible grid points.

According to P_{ij}^{k} of each point, choose the grid point for the next step (using the roulette method to select an accessible point according to the probability of each point), and the selected point is signed as the accessible grid point for the next step.

Record the accessible grid points selected in the closing table to consider whether they can be selected, and then determine whether they are accessible. The grid points stored in this table are taboo grid points; that is, they can no longer be used as new nodes.

According to the grid point selected, continue to determine the accessible points in its surrounding eight directions if accessible, record the information of these points, calculate the accessible probability, then use the roulette method to select an accessible point as the new accessible node according to the probability of each point, and record these nodes in the closing table.

In Fig. 11, there are smoke data which are mapped in grids of $P(i-1, j)$, $P(i-1, j+1)$, $P(i, j+1)$, and $P(i+1, j+1)$. The four grids are obstacle grids.

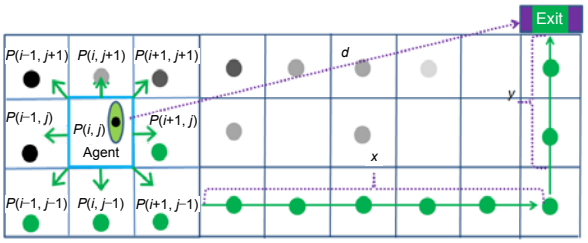


Fig. 11 Path-finding decision making by a panicked crowd

Continue to iterate this method to select accessible points and to find the path to the accessible nodes in the space region.

5. The ant continues to use the path planning methods in steps 6 and 7 until it reaches the escape door (exit) after entering a new spatial area.

6. Other ants repeat the whole process from step 6 to step 8, and each ant finds an accessible path.

7. Each ant can obtain all the grid points through which it passes, according to the data items in its closing table. The length of each path, the total amount of poisonous smoke in the path, and the total damage value of the heat in the path can be calculated according to the walking steps, the poisonous smoke data at each grid point, and the heat data at each grid point, respectively.

Calculate the damage value of the poisonous smoke and heat from which each person suffers in each optimal route, and record the maximum value of the damage. When the damage value becomes greater than the value that a person can tolerate, the path is set as impassable.

8. From the group of accessible paths, select several optimal paths that have the minimum damage found by the first iteration. On each path, each ant releases the same amount of pheromone. Depending

on the length of each path, we calculate the average pheromone value on each node of each path, and add this value into each node of this path. This updates the pheromone track. Next, the genetic algorithm is used for the optimal and suboptimal paths, and then a new optimal path is generated.

9. All ants start the second iteration to find the second path, which is affected by the pheromones left by the first walk. According to Eq. (1), collect the pheromones from node i to node j, and complete the calculation of $\tau_{ij}^{\alpha}(t)$ in the formula.

10. According to the number of simulations, set the iteration number and obtain a credible optimal path.

Complete the second iteration from step 6 to step 10. According to simulation needs, the number of iterations can be set to 50, 100, 200, or 500. Each simulation will create an optimal escape route in step 10. This route is recommended for evacuees.

Steps 1–8 are the completed evolution algebra; after step 10, all of the evolution algebra is the number of iterations.

In our simulation, the number of ants is the same as the number of people, and the large-scale scene 3 is used for path planning. Based on the fire scenario, a comparison examination is executed between the RFES-ACO algorithm and the classic ACO algorithm. Due to the heavy raw smoke data, the classic ACO algorithm cannot complete the optimal path planning in real time, but it can find the optimal path without considering the dynamic smoke. However, the RFES-ACO algorithm can complete optimization path planning while accommodating the data on dynamic spreading smoke.

In the classic ACO algorithm (Table 2), when α, β, and ρ are 1, 4, and 0.5, respectively, an optimal shortest path is 437.1 m and the evolution algebra is 142. However, the RFES-ACO algorithm (Table 3) considers the dynamic data, using the pheromone adaptive path planning to find the first path, and uses the genetic algorithm to vary the path node to speed up the path planning. $\rho_{\min}$ is used as the concentration threshold of the pheromone.

Simulation results are shown in Tables 2 and 3. The RFES-ACO algorithm plans better paths than classic ACO under any condition, and the evolution algebra of the RFES-ACO algorithm is lower than that of ACO.

Table 2 Simulation results for the classic ACO algorithm without dynamic spreading smoke

α	β	ρ	Length of the shortest path (m)	Evolution algebra
1	4	0.5	437.1	142
1	4	0.9	438.4	179
2	2	0.5	452.4	6
2	2	0.9	457.8	8
4	1	0.5	512.1	7

Table 3 Simulation results for the RFES-ACO algorithm

α	β	ρ	Length of the shortest path (m)	Evolution algebra	$\rho_{\min}$
1	4	0.5	427.1	93	0.001
1	4	0.1	425.6	31	0.100
1	4	0.9	423.9	42	0.001
1	4	0.9	425.6	37	0.500
1	4	0.9	425.8	13	0.100
2	2	0.5	448.7	4	0.001
2	2	0.9	448.6	4	0.500
4	1	0.5	456.0	5	0.100
4	1	0.5	466.9	6	0.001

In summary, the optimal results and convergence speeds of the RFES-ACO algorithm are all sharply better compared with those of the classic ACO. Based on the multithreading parallel technology, pheromone reuse, and the information exchange in parallel genetic algorithms, the RFES-ACO algorithm can perform efficient and accurate optimal path planning.

8 Visualization of crowd evacuation

The data can be made lightweight, the BIM scene numerical data can be initialized, dynamic variations on the DFS smoke data can be calculated, and loading can be initialized on the Mobile Web3D. The number of people in the second floor underground in the subway station is initialized at 3586, the number of people in the first floor underground in the subway station is initialized at 730, and the overall evacuation crowd is initialized at 4316 people.

Crowd evacuation in dynamic scenarios requires collision prevention. In the leader-follower traveling process shown in Fig. 12, "1" represents the leader, and "3" and "2" represent the followers.

The travel directions of all followers point to the leader, and in each frame of collision prevention, the followers cannot walk into the same grid. The specific process is as follows:

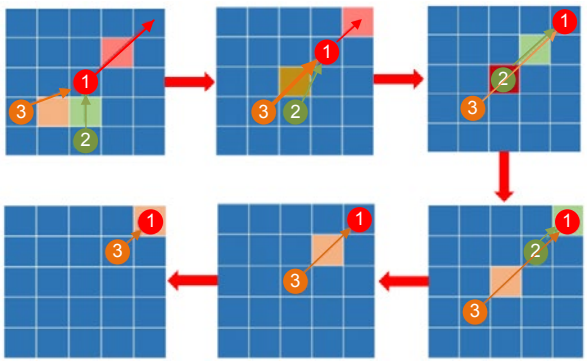


Fig. 12 Collision prevention for moving people based on the leader-follower model

1. Initialize the position of the smart agent randomly in the grid, and identify the smart agent and the team to which it belongs.

2. The leader of each team plans the optimal route according to the RFES-ACO algorithm, which is the walking path of the leader. If there is only one agent in the team, the agent itself is the leader.

3. If a team is walking on the platform, the team member characteristics are determined, and the speed of the slowest member in this group is used as the maximum speed of the team. Specific data based on this are shown in Table 4.

Table 4 Movement velocities of different crowd densities

Number of test teams	Crowd density (agent/m^2)	Average speed (m/s)
1	2.0	1.408
2	3.0	1.429
3	4.0	1.361
4	5.0	1.005
5	6.0	0.769
6	7.0	0.554
7	8.0	0.423

4. If the members of the team are in other locations, determine the position of the leader and the density of people ahead within the radius R of the scene, and determine the traveling speed of the next step according to Table 4.

5. Other people move based on their own speeds in each frame, their positions, and attribute information while following the leader.

6. All people escape successfully.

The average speed based on the crowd density derived from Xie et al. (2016) and the specific information is shown in Table 4.

9 Simulation results

The RFES system was based on the browser-server mode. The server end reduced the size of the scene and smoke data, and planned the optimal path. All the data were transferred to the client, and visualized on the Mobile Web, and the user could interact with the evacuation scene in fire evacuation.

Server end test environment: The operating system (OS) was Win-Server 2008, the memory was 4 GB, and the CPU was an Intel Xeon 2.39 GB.

Client end test environment: A common Hongmi Note 2 (only 899 CNY from JD Worldwide) with a 32-GB ROM and a 2-GB RAM was equipped with Google Chrome 52.0.2743.

Network bandwidth test environment: The laboratory bandwidth was 100 Mb/s, and the campus export bandwidth was 4.5 Gb/s.

9.1 Test results for size reduction of the scene model

We chose scenes from our BIM model database, where we had many scene models that range from several megabytes to several gigabytes. In the simulation, scenes 1–3 corresponded to the scene data numbered 1–3 here.

The IDLP technology was applied to the scene data from Table 5. In Fig. 13, the test results showed that the IDLP method was significantly better than the solibri IFC optimizer (SIO) method and the industry foundation classes files compressor (IFCCompressor) method. Because the raw IFC data were larger, the lightweight results were more significant. The SIO and IFCCompressor methods could not make the large-scale scene data lightweight, but the IDLP could quickly achieve lightweight data with a lightweight effect of about 78%, showing perfect robustness.

Table 5 Scene models

Number	Name	Amount of data (MB)
1	MetroStation-Scene.No.1.ifc	369.151
2	MetroStation-Scene.No.2.ifc	484.438
3	MetroStation-Scene.No.3.ifc	5429.433

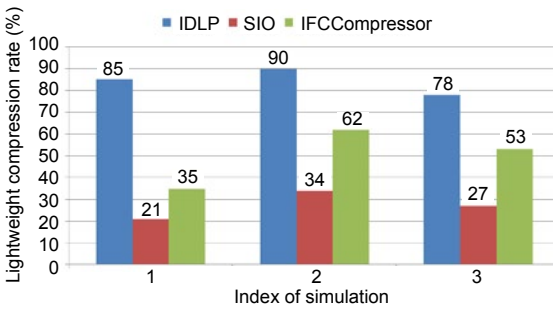


Fig. 13 Results of reducing the size of the scene model

9.2 Test results for network transmission

A total of 105 people visited the metro station when taking part in the fire evacuation training online. Increasing the number of users would increase the computer resource cost when using traditional methods. However, the P2P method was different from traditional methods. The loading time at initialization was slightly longer, but as the user number increased, the loading time constantly decreased and finally became steady. This reduced the access pressure on the server, and could even be loaded when the server was unable to connect. The test results are shown in Fig. 14.

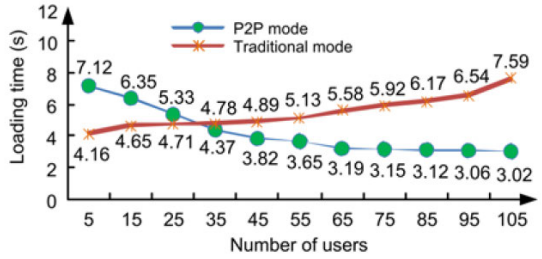


Fig. 14 Loading time for the metro station of scene 3

9.3 Path-planning time based on RFES-ACO

Based on the parameters in Table 3, nine sets of simulations were performed. Considering the number of iterations and the length of the optimal path, we chose the best and the worst results (Figs. 15 and 16). In the best solution for the RFES-ACO algorithm, the evolution algebra was 45 and the distance was 424. However, the evolution algebra was 6 and the distance was 450 in the worst solution for the RFES-ACO algorithm. This means that the RFES-ACO algorithm can be used for optimal path planning.

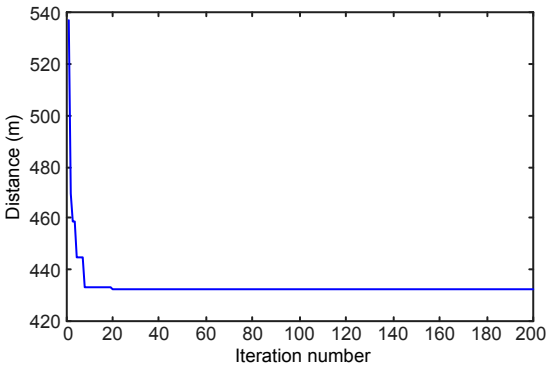


Fig. 15 Best solution for the RFES-ACO algorithm

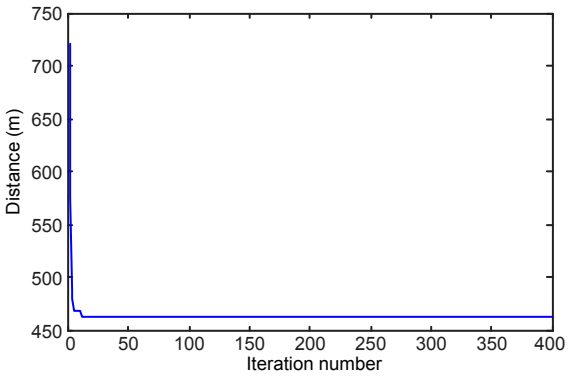


Fig. 16 Worst solution for the RFES-ACO algorithm

The RFES-ACO algorithm can be used to find the optimal path. The mean path-planning time of scene 1 (the raster number: 80×68/0.25=21 760) was 12 s, scene 2 (the raster number: 96×78/0.25=29 952) 16 s, and scene 3 (the raster number of every floor: 450×60/0.25=108 000) 18 s.

In Fig. 17, the mean time spent in path planning was relatively close among the three scenes. This is because there are more escape exits in a large-scale scenario than in other corresponding scenes. In scenes 1–3, the time spent in path planning was less than 20 s. The path was pushed into an array, which was transmitted to the Mobile Web3D device in real time and could be visualized by the user.

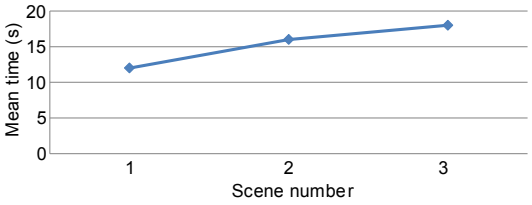


Fig. 17 Mean time of the three algorithms

9.4 Real-time evacuation frame rate on Web3D

We chose a large-scale metro station from our BIM model database, and the original metro station was 102.7 MB. After lightweight processing, the metro station scene was only 4.87 MB. Every original agent was 75.5 MB, which was downloaded from the official website of three.js. Yet, the lightweight agent was 0.5 MB, which was constructed by us.

In our scene, 20, 40, 80, and 200 agents were used to show the evacuation process on our website. The agents were located randomly, and the young people accounted for 70% of the total, while the proportions of elderly and minors were both 15%. In this study, a cloning technology was used to reduce the model and texture data. The whole crowd evacuation system occupied 94 MB of the mobile phone memory when it was running.

During the evacuation drills, the occupants could use the Mobile Web and take part in the fire evacuation drill anytime and anywhere. In the Web3D environment, every occupant had the power to set his/her agent including the gender and physiological properties (Fig. 18).

We also developed another kind of experience. Wearing virtual reality (VR) goggles, a human subject entered a Web3D fire scene and planned a real evacuation route. Using our Mobile Web3D RFES evacuation prototype system, we performed hundreds of real human escape exercises (Fig. 19).

Fig. 18 Fire drill on Mobile Web3D

Fig. 19 Fire drill with virtual reality goggles

From Fig. 20, there were 40 agents preparing for evacuation, and the refresh rate of the page was between 4 and 42 frames/s. When the system was in a stable state, the refresh rate was 37 frames/s.

In Fig. 21, there were 20 agents performing path planning from different starting points to two exits. The refresh rate of the page was between 4 and 42 frames/s in Mobile Web3D. When the system is in a stable state, the refresh rate was 33 frames/s.

There were 40 agents in the scene of Fig. 22, and the path was planned using the RFES-ACO algorithm. The refresh rate in Mobile Web3D was between 4 and 20 frames/s, and the stable refresh rate was 17 frames/s in the Web application on the cellphone of Hongmi Note 2 (the agent model data downloaded from the official three.js website were very large).

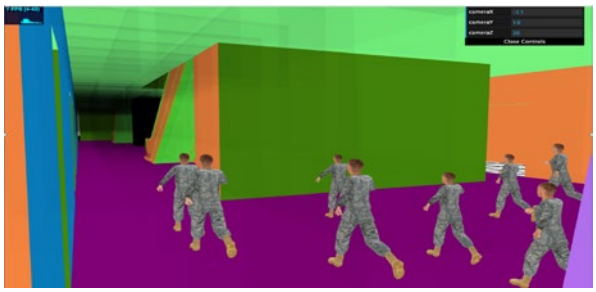


Fig. 20 The fire scenario and agents preparing for evacuation

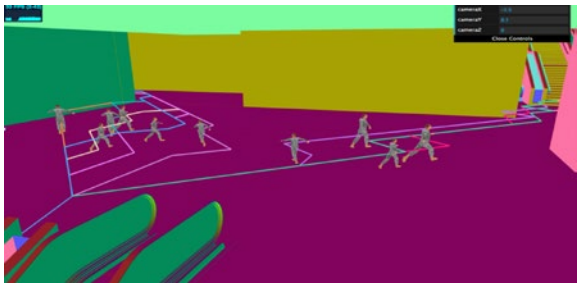


Fig. 21 Optimal path planning using RFES-ACO for 20 agents

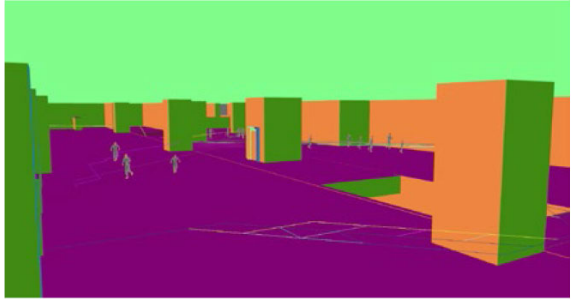


Fig. 22 Evacuation path planning for 40 agents in a complex metro station

There were 80 agents in the scene of Fig. 23, and the path was planned using the RFES-ACO algorithm. The refresh rate in Mobile Web3D was between 0 and 58 frames/s, and the stable refresh rate was 30 frames/s.

In a large-scale metro station, 200 people engaged in path planning. Fig. 24 shows an escape scenario. The refresh rate was stable at 26 frames/s.

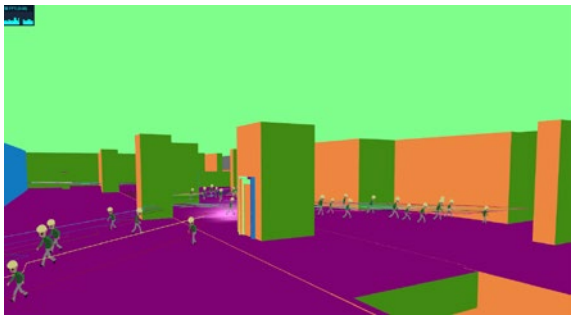


Fig. 23 Evacuation path planning for 80 agents in a complex subway station

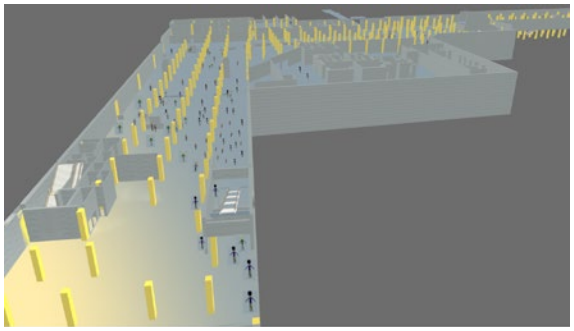


Fig. 24 A large-scale metro station, with 200 people engaging in path planning

9.5 RFES-ACO evacuation success rates

To evaluate the evacuation success rate (i.e., the ratio of survivors to the original number of agents) achieved by the RFES-ACO algorithm, two benchmarking algorithms were used for comparison: the leader-follower algorithm and the random algorithm. The scene for the test is scene 1.

The number of passengers in the test was 200. For each of the three algorithms, five simulations were conducted. The simulation data are listed in Table 6 and the evacuation success rate is shown in Fig. 25, clearly showing that the RFES-ACO algorithm has the highest fire evacuation success rate, whereas the random algorithm has the lowest.

Table 6 Crowd evacuation success rate

Algorithm	Crowd evacuation success rate (%)					
	Index of simulation					Mean value
	1	2	3	4	5	
RFES-ACO	100	99	100	100	100	99.8
Leader-follower	99	97	96	61	100	90.6
Random	53	32	46	57	36	44.8

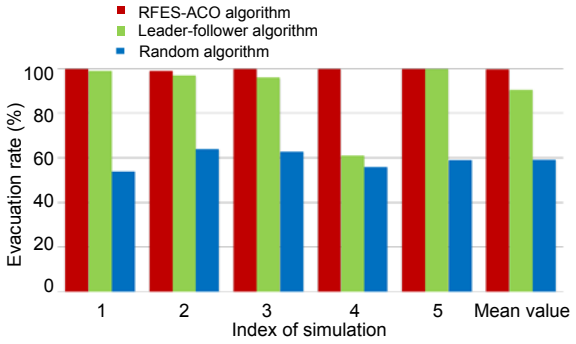


Fig. 25 Evacuation success rates of the three algorithms

10 Conclusions and future studies

In this paper, we have proposed a series of lightweight solutions: (1) A large-scale scene model has been made lightweight. (2) The amount of dynamic smoke has been made lightweight. (3) Dynamic obstacle maps established from a static scene model and dynamic smoke obstacle data have been used for optimal path planning with a heuristic method and an improved ACO algorithm—a real-time fire evacuation system based on ant colony optimization with reused dynamic pheromones. Simulations proved that RFES-ACO is effective with fast convergence. (4) We have proposed and implemented a method to visualize the lightweight building model and lightweight smoke data for Web3D. The simulation results showed that the public could use Mobile Web3D devices to experience fire evacuation drills in real time and smoothly. RFES was efficient and the evacuation rate was better than those of the leader-follower fire evacuation algorithm and the random fire evacuation algorithm.

There is considerable work to be done in the future. The most important task is modeling lightweight building scenes and agents. Then a popular system platform could support more people to take part in Mobile Web3D fire evacuation. Second, an evacuation simulation for disabled people should be

completed in the system. Third, crowd congestion situations, and the bearing capacity of elevators and stairs should be taken into account in Mobile Web3D. At the same time, the system should include rescue workers in a complex escaping crowd situation.

Compliance with ethics guidelines

Feng-ting YAN, Yong-hao HU, Jin-yuan JIA, Qing-hua GUO, He-hua ZHU, and Zhi-geng PAN declare that they have no conflict of interest.

References

Athanasis N, Karagiannis F, Palaiologou P, et al., 2015. AEGIS app: wildfire information management for windows phone devices. *Proc Comput Sci*, 56:544-549. https://doi.org/10.1016/j.procs.2015.07.249

Barsoum E, Kuester F, 2005. WebVR: an interactive web browser for virtual environments. *SPIE*, 5664:540-547. https://doi.org/10.1117/12.582624

Benkoussas B, Djedjig R, Vauquelin O, 2016. Numerical assessment of conventional regulation effectiveness for smoke removal from a two level underground station. *J Fundam Appl Sci*, 8(2):401-425. https://doi.org/10.4314/jfas.v8i2.16

Cha M, Han S, Lee J, et al., 2012. A virtual reality based fire training simulator integrated with fire dynamics data. *Fire Saf J*, 50:12-24. https://doi.org/10.1016/j.firesaf.2012.01.004

Chu L, Wu S, 2011. A real-time fire evacuation system with cloud computing. *J Converg Inform Technol*, 7(7):208-215. https://doi.org/10.4156/jcit.vol7.issue7.26

Duo Q, Shen H, Zhao J, et al., 2016. Conformity behavior during a fire disaster. *Soc Behav Pers*, 44(2):313-324. https://doi.org/10.2224/sbp.2016.44.2.313

Fang X, Huang P, Huo L, 2011. Progress in the research on crowd's emergency behaviors in large-scale events. *China Saf Sci J*, 21(11):22-28 (in Chinese). https://doi.org/10.16265/j.cnki.issn1003-3033.2011.11.023

Guest J, Eaglin T, Subramanian K, et al., 2015. Interactive analysis and visualization of situationally aware building evacuations. *Inform Vis*, 14(3):204-222. https://doi.org/10.1177/1473871613516292

Humayoun SR, Ebert A, Hess S, et al., 2015. Workshop on designing interaction and visualization for mobile applications (DIViM 2015). *LNCS*, 9299:675-676. https://doi.org/10.1007/978-3-319-22723-8_97

Kinateder M, Ronchi E, Gromer D, et al., 2014a. Social influence on route choice in a virtual reality tunnel fire. *Transp Res Part F*, 26(Part A):116-125. https://doi.org/10.1016/j.trf.2014.06.003

Kinateder M, Ronchi E, Nilsson D, et al., 2014b. Virtual reality for fire evacuation research. Federated Conf on Computer Science and Information Systems, p.313-321. https://doi.org/10.15439/2014F94

Li W, Wu WJ, Wang HM, et al., 2017. Crowd intelligence in AI 2.0 era. *Front Inform Technol Electron Eng*, 18(1):19-47. https://doi.org/10.1631/FITEE.1601859

Lin Y, Liu Y, Gao G, et al., 2013. The IFC-based path planning for 3D indoor spaces. *Adv Eng Inform*, 27(2):189-205. https://doi.org/10.1016/j.aei.2012.10.001

Martinez-Gil F, Lozano M, Fernández F, 2015. Strategies for simulating pedestrian navigation with multiple reinforcement learning agents. *Auton Agents Multi-Agent Syst*, 29(1):98-130. https://doi.org/10.1007/s10458-014-9252-6

Onorati T, Aedo I, Romano M, et al., 2014. EmergenSYS: mobile technologies as support for emergency management. *LNISO*, 7:37-45. https://doi.org/10.1007/978-3-319-07040-7_5

Pluchino A, Garofalo C, Inturri G, et al., 2013. Agent-based simulation of pedestrian behaviour in closed spaces: a museum case study. *J Artif Soc Soc Simul*, 17(1):16-29. https://doi.org/10.18564/jasss.2336

Qin K, Hu C, Jia D, et al., 2014. Subway fire evacuation simulation model. Int Conf on Identification, Information and Knowledge in the Internet of Things, p.233-236. https://doi.org/10.1109/IIKI.2014.54

Song Y, Gong J, Li Y, et al., 2013. Crowd evacuation simulation for bioterrorism in micro-spatial environments based on virtual geographic environments. *Saf Sci*, 53:105-113. https://doi.org/10.1016/j.ssci.2012.08.011

Tian Y, Zhou TS, Yao Q, et al., 2014. Use of an agent-based simulation model to evaluate a mobile-based system for supporting emergency evacuation decision making. *J Med Syst*, 38(12):149. https://doi.org/10.1007/s10916-014-0149-3

Wang S, Wang W, Wang K, et al., 2015. Applying building information modeling to support fire safety management. *Autom Constr*, 59:158-167. https://doi.org/10.1016/j.autcon.2015.02.001

Wang Z, Zheng L, Du W, 2019. A novel method for intelligent fault diagnosis of bearing based on capsule neural network. *Complexity*, Article 6943234. https://doi.org/10.1155/2019/6943234

Xie XL, Ji JW, Wang ZH, et al., 2016. Experimental study on the influence of the crowd density on walking speed and stride length. *J Saf Environ*, 16(14):232-235. https://doi.org/10.13637/j.issn.1009-6094.2016.04.047

Yan F, Jia J, Hu Y, et al., 2019. Smart fire evacuation service based on Internet of Things computing for Web3D. *J Int Technol*, 20(2):521-532.

Yuan Z, Jia H, Liao M, et al., 2017. Simulation model of self-organizing pedestrian movement considering following behavior. *Front Inform Technol Electron Eng*, 18(8):1142-1150. https://doi.org/10.1631/FITEE.1601592

Zheng Y, Ling H, Xue J, et al., 2014. Population classification in fire evacuation: a multiobjective particle swarm optimization approach. *IEEE Trans Evol Comput*, 18(1):70-81. https://doi.org/10.1109/TEVC.2013.2281396