J Inf Process Syst, Vol.16, No.6, pp.1309~1323, December 2020
https://doi.org/10.3745/JIPS.04.0199

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Improved Social Force Model based on Navigation Points for Crowd Emergent Evacuation

Jun Li*, Haoxiang Zhang**, and Zhongrui Ni***

Abstract
Crowd evacuation simulation is an important research issue for designing reasonable building layouts and planning more effective evacuation routes. The social force model (SFM) is an important pedestrian movement model, and is widely used in crowd evacuation simulations. The model can effectively simulate crowd evacuation behaviors in a simple scene, but for a multi-obstacle scene, the model could result in some undesirable problems, such as pedestrian evacuation trajectory oscillation, pedestrian stagnation and poor evacuation routing. This paper analyzes the causes of these problems and proposes an improved SFM for complex multi-obstacle scenes. The new model adds navigation points and walking shortest route principles to the SFM. Based on the proposed model, a crowd evacuation simulation system is developed, and the crowd evacuation simulation was carried out in various scenes, including some with simple obstacles, as well as those with multi-obstacles. Experiments show that the pedestrians in the proposed model can effectively bypass obstacles and plan reasonable evacuation routes.

Keywords
Crowd Evacuation, Navigation Point, Pedestrian Evacuation Route, Social Force Model

1. Introduction

With the rapid progress of urbanization all around the world, the scale of the cities continues to expand, and there are mounting number of places where people are highly concentrated, such as large shopping malls, large stadiums, theaters, schools, hospitals, railway stations, etc. If there is no effective evacuation guidance available in such dense venues, an accident or emergency, such as a fire alarm, an earthquake or a terrorist attack, could cause serious chaos and may even result in injuries and deaths. For example, on March 25, 2018, a fire broke out in a shopping mall in the southern city of Kremlovo, Siberia, Russia. The accident resulted in 64 deaths and 52 injuries owing to the lack of scientific and effective evacuation measures. On August 25, 2018, a fire broke out at the Beilong Hot Spring Hotel in Harbin, a city in northeast China. The hotel had complex building structure, very few evacuation signs, and the accident resulted in 19 deaths and 23 injuries. Therefore, researches on how to design a building layout together with effective evacuation plan under emergencies are of vital importance for public security.

Manuscript received July 31, 2019; first revision September 16, 2019; second revision November 4, 2019; third revision December 26, 2019; accepted January 14, 2020.
Corresponding Author: Haoxiang Zhang (zhanghx@nbut.edu.cn)
* School of Electronic and Information Engineering, Ningbo University of Technology, Ningbo, China (lijun@nbut.edu.cn)
** Robotics Institute, Ningbo University of Technology, Ningbo, China (zhanghx@nbut.edu.cn)
*** Faculty of Information Science and Technology, Ningbo University, Ningbo, China (83524788@qq.com)

Pedestrian motion models have been developed to simulate the pedestrian movement behavior in emergencies and are used in relevant researches. The social force model (SFM) is an important pedestrian movement model, which is widely exploited in crowd evacuation simulations. However, the SFM can only be applied to simple scenes, with single obstacle present. When there are multiple obstacles in the scene, the problems such as pedestrian motion trajectory oscillation, pedestrian stagnation may arise, and the model would be unable to choose a reasonable route. In this paper, we propose a pedestrian evacuation model for multiple obstacles scenarios. The model can plan reasonable pedestrian evacuation routes in such complex scenes, avoiding the problems of pedestrian motion trajectory oscillation, stagnation and poor evacuation routing. Compared with the original SFM, the proposed model suits a much wider range of scenes, and provides better crowd evacuation simulation.

2. Related Work

Many researchers and institutions have conducted extensive research and exploration on crowd evacuation and a lot of remarkable results have been achieved using various crowd evacuation models, such as SFM [1-12], reciprocal velocity obstacles (RVO) model [13,14], cellular automata (CA) model [15-17], and potential field (PF) model [18,19].

SFM is widely used in many pedestrian evacuation simulations. Many scholars have optimized and improved it. For example, Jiang et al. [3] designed an extended SFM with a dynamic navigation field to study bidirectional pedestrian movement. Cao et al. [4] extended SFM by using a discrete grid to allow pedestrians to change their desired speed and directions dynamically. Zhang et al. [5] designed a modified two-layered SFM to simulate and reproduce a group gathering process. Korecki et al. [6] studied skiers' physics and behavior modeled with a modified SFM approach. Anvari et al. [9] presented a three-layered microscopic mathematical model by the modified SFM. In the study of Li et al. [10], the escape panics of classroom evacuation in real-life 2013 Ya'an earthquake in China were simulated and reproduced using SFM. Li et al. [11] developed simulation model on congestion risk during escalator transfers based on a modified SFM. Ji et al. [12] designed a novel approach to detect and locate abnormal events in crowed scenes by SFM. To address the issue of obstacle avoidance, Zhang et al. [20] proposed a novel extended SFM-based mean shift tracking algorithm in which pedestrian environment is taken into full consideration.

In the process of crowd evacuation simulation, the choice of evacuation paths has an important impact on pedestrian evacuation time and efficiency. The shortest path planning is a commonly used approach in crowd evacuation simulation, mainly including Dijkstra algorithm, Floyd algorithm and A* algorithm. Many scholars have proposed new methods in this research field. Hoogendoom and Bovy [21] established a dynamic programming model to determine the path by finding the optimal solution. Yue et al. [22] proposed a pedestrian evacuation model with asymmetrical exits layout. The model performs evacuation simulation by introducing direction-parameter, empty-parameter, cognition coefficient, and imbalance coefficient. Jo et al. [23] presented a microscopic simulation model for assessing pedestrian flow, which uses a two-player game to establish a microscopic model of pedestrian behavior and to determine macroscopic paths based on a given destination. In [24], a simulation model of microscopic pedestrian evacuation with discrete space representation is developed, and the route choice of pedestrians during evacuation under good and zero visibility conditions is studied. Kang et al. [25] found the path for agent by using data from the user and automatically building the navigation grid method. Sud et al. [26]

introduced a new multi-agent navigation graph, combined with the SFM to perform route planning for each agent in real time. In order to evaluate the performance of crowd evacuation, Haghpanah et al. [27] proposed a pedestrian navigation algorithm for city evacuation modeling, which developed a performance evaluation framework consisting of a list of typical obstacles that can be challenging for navigation algorithms to process, as well as a set of performance indices.

In summary, the research on crowd evacuation simulation technology has made great progress. Many models can effectively simulate crowd evacuation behavior. However, in complex multi-obstacle scenarios, the route selection is generally complicated, and these models often have difficulty to simulate close-to-real crowd evacuation behaviors. Therefore, it is necessary to carry out exploration and develop new models that can simulate the crowd evacuation behavior more effectively.

3. SFM Introduction and Problem Analysis

3.1 SFM Introduction

In an actual escape process, people always determine the escape route according to their destination and their surrounding environment, while keeping a certain distance from obstacles and other people. Based on this principle, Helbing proposed the SFM using Newton's mechanical formula to model pedestrian escape behavior. In this model, pedestrians are described by abstract particles that are attracted by the destination, producing a driving force to them. At the same time, a particle is affected by the repulsion and friction of the obstacle and other particles. The resultant force acts on the particle so that it moves continuously in a two-dimensional space, which is described using the kinetic formula, as described in Eq. (1).

$$m_i \frac{dv_i}{dt} = m_i(v_i^0(t)e_i^0(t) - v_i(t))/\tau_i + \sum_{j(\neq i)} f_{ij} + \sum_W f_{iw} \quad (1)$$

where $m_i(v_i^0(t)e_i^0(t) - v_i(t))/\tau_i$ is the driving force of the pedestrian i to the destination; m_i is the mass, v_i^0 and $v_i(t)$ are its expected speed and actual speed, and $e_i^0(t)$ is the desired direction of motion; τ_i is a certain characteristic time, and its value is set to be 0.5; f_{ij} is the interaction forces between the pedestrian *i* and pedestrian *j*, and f_{iw} is the interaction force between the pedestrian *i* and the obstacle *w*. The resultant force of these forces acts on the pedestrian, producing an acceleration that drives the pedestrian to move continuously in a two-dimension space.

$$f_{ij} = (A_i exp\,[(r_{ij} - d_{ij})/B_i] + kg(r_{ij} - d_{ij}))n_{ij} + \kappa g\left(r_{ij} - d_{ij}\right)\Delta v_{ji}^t t_{ij} \quad (2)$$

Eq. (2) is the formula of the interaction forces between the pedestrians. The interaction forces f_{ij} consists of repulsive interaction force and sliding friction force. A_i, B_i and *k* are constant numbers. A_i is the intensity of the repulsive interaction between pedestrians, and the value is 2,000 N. B_i is the minimum distance at which the repulsive interaction force is generated, and the value is 0.08 m; *k* is 1.2×10^5 kg/s^2; r_{ij} is the sum of the radius of pedestrian i and j, $r_{ij} = r_i + r_j$. d_{ij} indicates the distance between the centroids of pedestrians *i* and *j*; $d_{ij} = ||r_i - r_j||$. n_{ij} represents a normalized vector from pedestrian *j* to i, $n_{ij} = \left(n_{ij}^1, n_{ij}^2\right) = (r_i - r_j)/d_{ij}$. *g*(*x*) represents a function that takes a value of 0 when the pedestrians cannot touch each other ($r_{ij} < d_{ij}$), or *x* otherwise. The value of *k* in the is 2.4×10^5 kg/m/s, Δv_{ji}^t indicates the relative rate of the tangential direction of pedestrians i and j, $\Delta v_{ji}^t = (v_j - v_i).t_{ij}$. t_{ij} represents the tangent direction of pedestrian i and j, $t_{ij} = (-n_{ij}^2, n_{ij}^1)$.

$$f_{iw} = (A_i exp\,[(r_i - d_{iw})/B_i] + kg(r_i - d_{iw}))n_{iw} + kg(r_i - d_{iw})(v_i.t_{iw})t_{iw} \quad (3)$$

Eq. (3) is the formula of the interaction forces between pedestrians and obstacles. The interaction forces f_{iw} consists of repulsive interaction force and sliding friction force. A_i, B_i and k are constant numbers, the same values as above. r_i is the radius of the pedestrian i. d_{iw} is the distance between the pedestrian i and the edge of the obstacle; n_{iw} represents the normalized vector from the edge of the obstacle to the pedestrian i. v_i represents the actual rate of pedestrian i, and t_{iw} represents the tangential direction of pedestrian i and the edge of the obstacle.

3.2 Problem Analysis

In SFM, the movement of a pedestrian is driven by the combined force of self-driving force, the interaction force between pedestrians, and the force between pedestrian and obstacle. If there are obstacles in the scene, using SFM to simulate pedestrian evacuation behavior could cause pedestrian motion oscillation and stagnation, and the resultant evacuation route is distorted.

3.2.1 Pedestrian motion oscillation and stagnation

According to the SFM principle, using the original SFM in the scene with obstacles will cause pedestrian motion oscillations and stagnation. Fig. 1 shows an example of this situation. Suppose *S* is the starting point, *T* is the destination point, and SFM is used for pedestrian evacuation.

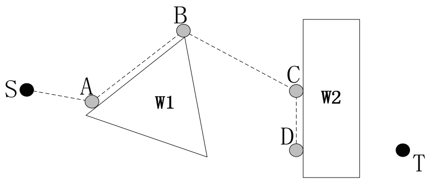


Fig. 1. An example of pedestrian motion oscillation and stagnation.

The trajectory of evacuation in the SFM is SABCD. When a pedestrian does not encounter an obstacle in SA and BC, the direction of motion is determined by $e_i^0(t)$ in Eq. (1), and the direction of motion points to T. When the pedestrian is on AB, the direction of the motion is along AB according to the repulsive force W1 obtained by Eq. (3), and it is similar on CD. When the pedestrian reaches point D, the direction of the self-driving force is to the right. According to Eq. (3), the repulsive force of the pedestrian is to the left, and the friction force is zero. As pedestrians approach obstacles, d_{iw} decrease, leading to an increase in repulsive force. Eventually, the pedestrian stops at point D, where the forces on opposite directions are balanced. Therefore, the pedestrian from point S cannot be evacuated in the simulation using the SFM.

3.2.2 Unreasonable pedestrian evacuation route

The movement of pedestrians in SFM is driven by the combined force of the self-driving force of the pedestrian to the destination, the force between the pedestrians, and the force between the pedestrian and the obstacle. It does not take into account the fact that pedestrians would actively choose the shortest evacuation route, so the simulated evacuation route may be longer than the actual one. For example, the evacuation route driven by the SFM in Fig. 2 is SABCDT, which is longer than the route SET that the pedestrian may actually choose. So, in some scenes, the pedestrian evacuation route using the SFM may be longer than the one actually selected by the pedestrian.

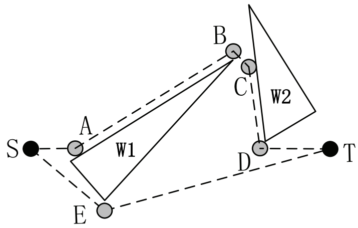


Fig. 2. The example of the pedestrian evacuation route in SFM is longer than the actual pedestrian evacuation rout.

4. Improved SFM Algorithm Based on Navigation Points

In this paper we propose an improved SFM algorithm to solve the above problems. The algorithm consists of three steps. First, generate navigation points, then plan evacuation routes that the pedestrian could possibly choose; finally, optimize the candidate evacuation routes, and find the shortest route as the simulation result.

4.1 Generate Navigation Points

In the proposed algorithm, obstacles in the scene are modeled by convex polygons. To ensure higher generality, any obstacles of non-convex polygon would first go through the following steps to be converted into a convex polygon.

Input: The vertices of the polygon (non-convex).
Output: The vertices of the converted polygon (convex).

Step 1) Traverse the polygon to find out whether there is a reflex angle in the graph. If a reflex angle is found, continue to step 2; otherwise jump to step 3.

2) Connect the two adjacent endpoints of the reflex angle as the new edge of the polygon, and remove the reflex angle and its two edges. Go back to step 1.

3) Output convert polygon vertices.

In the process of evacuation, if there are obstacles between the pedestrian and the destination, in order to enable the pedestrians to bypass obstacles and not to collide with it, a reasonable point near the apex of the obstacle is selected as a navigation point, and the chosen navigation points need to meet two conditions. First, the chosen navigation points cannot intersect with obstacles, otherwise there could be pedestrian movement oscillation and stagnation. For example, in Fig. 3, let’s assume that point D1 is selected as the navigation point at vertex D, and it intersects with the obstacle. When a pedestrian reaches the navigation point D1, the pedestrian will collide with the obstacle, causing motion oscillation and stagnation. Second, the adjacent two navigation points in an obstacle should be connected, for if they are not, pedestrians will collide with obstacle if they are evacuated from one navigation point to another. For example, in Fig. 3, point A1 is selected as the navigation point near vertex A, and point B1 as the navigation point near vertex B. A1 and B1 are NOT connected. If the pedestrian is evacuated from A1 to B1, he will collide with the obstacle. Instead, navigation points A1 and B2 are an example of correct choice of navigation points, which meet the above condition.

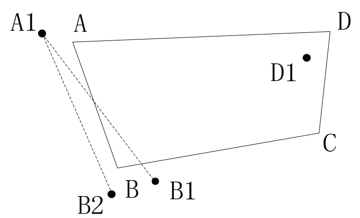


Fig. 3. Example description of the selected navigation point that requires the conditions.

Here we propose a navigation points selection algorithm to keep the selected navigation point and vertex at a certain angle and distance to meet the above two conditions. An angle is first formed by two adjacent edges of the obstacle vertex, and then an angular bisector is made in the angle. On the extension line of the angular bisector, a position twice the pedestrian radius from the vertex is selected as the navigation point. The following is the algorithm description of generating the navigation point near the vertex B, as shown in Fig. 4.

Input: Coordinates of vertices A, B and C.	
Output: The navigation point near the vertex B.	
Step	1) Calculate the vector $\overrightarrow{OA}$, $\overrightarrow{OB}$, $\overrightarrow{OC}$ according to the coordinates of the vertices A, B, and C. 2) Calculate the vector $\overrightarrow{BA} = \overrightarrow{OA} - \overrightarrow{OB}$, and the unit vector $\overrightarrow{BA'}$ of the vector $\overrightarrow{BA}$. 3) Calculate the vector $\overrightarrow{BC} = \overrightarrow{OC} - \overrightarrow{OB}$, and the unit vector $\overrightarrow{BC'}$ of the vector $\overrightarrow{BC}$ 4) Calculate the vector $\overrightarrow{BF} = \overrightarrow{BA'} + \overrightarrow{BC'}$, and the unit vector $\overrightarrow{BF'}$ of vector $\overrightarrow{BF}$. 5) Calculate the vector $\overrightarrow{BB'} = \mathrm{N} \times R \times \overrightarrow{BF'}$ (N is a negative integer and R is the pedestrian radius. In this paper, the value of N is -2, and the value of R is 0.4.), and the vector $\overrightarrow{OB'} = \overrightarrow{OB} + \overrightarrow{BB'}$, get the point B' coordinates.

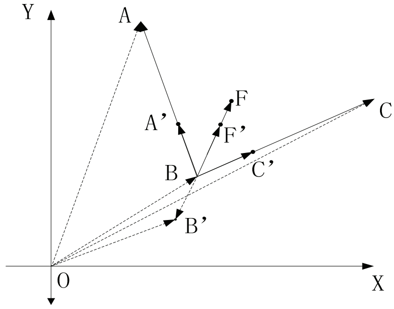


Fig. 4. Algorithm description of generating the navigation point near the vertex B.

4.2 Explore the Candidate Evacuation Routes

In order to evacuate to the destination as quickly as possible, the pedestrian would choose the shortest evacuation route among the various candidate routes. If there are obstacles between the pedestrian point and the destination point, pedestrians need to bypass obstacles and there are often multiple routes that pedestrians can choose from. The method we use to explore the candidate evacuation routes is that each time the pedestrian bypasses obstacles, he only needs to consider the nearest obstacle and find the navigation point that bypasses the obstacle from the left and right sides of the nearest obstacle, respectively, and use these two navigation points as pedestrian positions. Look for the next nearest obstacle for pedestrians, find the navigation points where pedestrians bypass obstacles from the left and right sides, and proceed until the pedestrian position and the destination point are connected to get the evacuation route.

We use a binary tree to construct possible evacuation routes. A node of the binary tree represents a navigation point of pedestrian evacuation. The node has three properties: navigation point state, navigation point coordinate, and distance. The navigation point state has three values: invalid, incomplete, and completed. Invalid indicates that the pedestrian has passed this navigation point; incomplete means that the pedestrian has not passed this navigation point; completed indicates that the navigation point is the destination point. In the binary tree, the route from the root node to the leaf node whose navigation point state is completed can be used as an evacuation route that the pedestrian can choose.

The algorithm is described below.

Input: Starting point of the pedestrian, destination point, navigation points.
Output: the evacuation routes that the pedestrian probably chooses.

Step 1) Set the pedestrian starting point as the root node of the binary tree. The state of the root node is incomplete. The coordinate is the starting position coordinate of pedestrian. The distance is 0.

2) If the pedestrian is inside the newly converted polygon, then select the navigation point that is closest and connected to the pedestrian as the child node of the root node. The state of the child node is set to incomplete, the coordinate is the navigation point, and the distance attribute is the distance between the pedestrian and the navigation point. Otherwise continue to step 3.

3) Traversing the binary tree, if there is a node with state is incomplete, treat it as the current node and continue to step 4. Otherwise, jump to step 7 to continue.

4) In the scene, determine whether the current node and the destination point are connected. If not, continue to step 5. Otherwise, set the destination point as the child of the current node in the binary tree, and set its state to complete. The coordinate value is the coordinate value of the destination point. The distance attribute is the distance between the destination point and the current node. Jump to step 3 to continue.

5) Find the obstacle closest to the current node between the current node and the destination, and then find the two navigation points that bypass the obstacle from the left and right sides of this obstacle respectively.

6) Set these two navigation points to be the children of the current node. Set the properties of these two child nodes separately according to the following rules. If the child node does not appear in its ancestor, set the state to incomplete, the coordinate value is the coordinates of the navigation point in the scene, and the distance property is the distance between the child node and its parent node. If the child node appears in its ancestor, set the state to invalid. Jump to step 3 to continue.

7) Traversing the binary tree, each route from the root node to the leaf node whose state is completed is an evacuation route that the pedestrian possible choose.

Fig. 5 shows an example of the implementation process of the algorithm. According to the algorithm, five available routes are found, they are SACIT, SACHJT, SBDHJT, SBDJT, SBGFT.

4.3 Choose the Shortest Evacuation Route

In each of the evacuation routes obtained by the above algorithm, there may be two non-adjacent navigation points that are connected. In order to make the evacuation route shorter, pedestrians generally do not pass through the navigation point between the two connected navigation points during the evacuation process. For example, in Fig. 5(d), SACIT is one of the routes that the point S pedestrian might choose. However, in the actual evacuation process, pedestrians tend to choose the route SAIT instead. Point A and point I are connected, the pedestrian would prefer to take route AI rather than route ACI. Therefore, in order to make the evacuation route more realistic, it is necessary to optimize the evacuation routes obtained above.

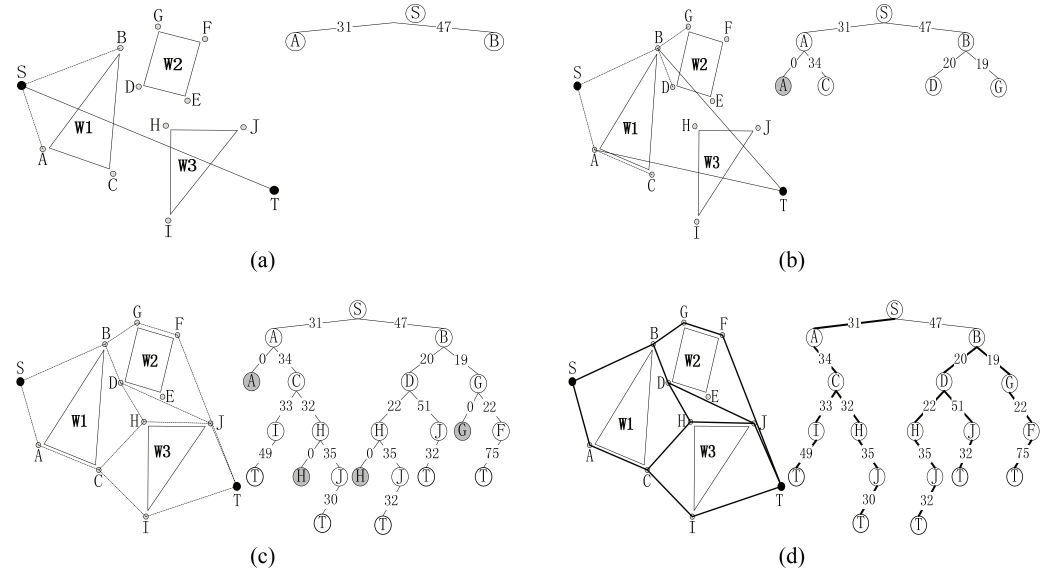


Fig. 5. Example of using the binary tree to explore the evacuation route that pedestrians probably choose: (a) calculate the route of node S, (b) calculate the route of node A & B, (c) all nodes are completed, and (d) find five available routes.

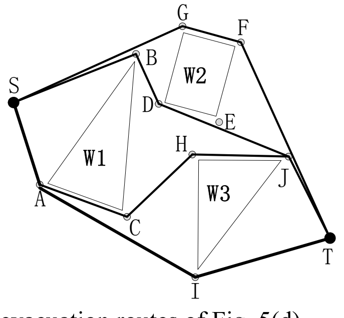


Fig. 6. Result of optimization of all evacuation routes of Fig. 5(d).

The optimization method is to remove the navigation point between two connected navigation points along the evacuation route. Assume that P_M and P_N are two navigation points in an evacuation route obtained in the above algorithm. If P_M and P_N are connected, the pedestrian could directly evacuate from the P_M to the P_N without passing any navigation point in between. So, the navigation point between the P_M and the P_N is deleted in the evacuation route.

After optimizing all evacuation routes, we can calculate the length of each route and then find the shortest one as the pedestrian evacuation route. For example, Fig. 6 is a result of optimization of all evacuation routes of Fig. 5(d), and it can be found that the shortest route is SAIT, which is an evacuation route more likely to be selected by the pedestrian.

5. Experiments and Results

The simulation system was developed with Visual Studio 2017. In the scene of different obstacles, the motion trajectory is drawn according to the position of each pedestrian recorded every time.

5.1 Pedestrian Evacuation Experiment in Obstacle Scene

The simulation scene adopts a rectangular space, where the destination is on the right side, and pedestrians are randomly positioned on the left side. Fig. 7 shows experimental results with obstacles of four different shapes. It can be seen that when the pedestrian's starting position is directly connected to the destination point, the walking route appears as a straight line without passing through the navigation point. If there is an obstacle between the pedestrian and the obstacle, the pedestrian can effectively bypass obstacles and choose reasonable routes to evacuate to destinations.

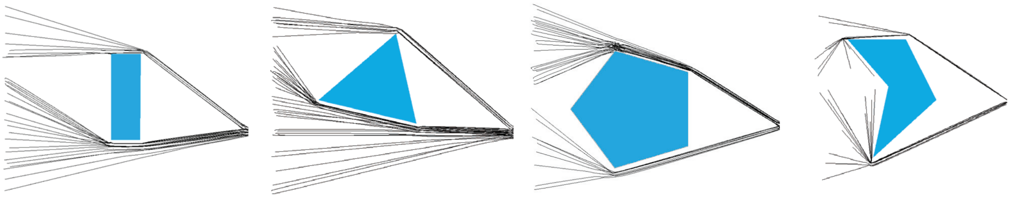

(a) (b) (c) (d)

Fig. 7. Pedestrian evacuation experiments in various simple obstacle scenes: (a) rectangular obstacle, (b) triangle obstacle, (c) irregular convex polygons obstacle, and (d) irregular non-convex polygons obstacle.

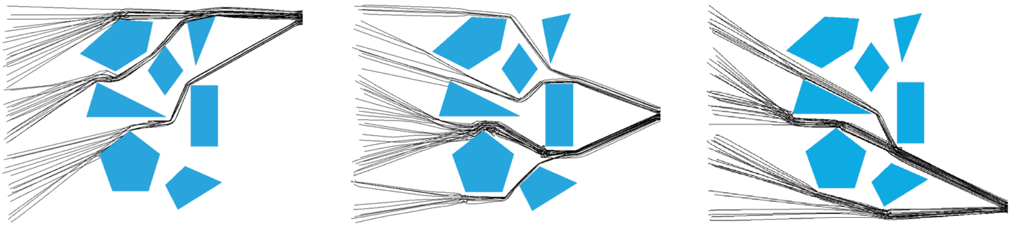

(a) (b) (c)

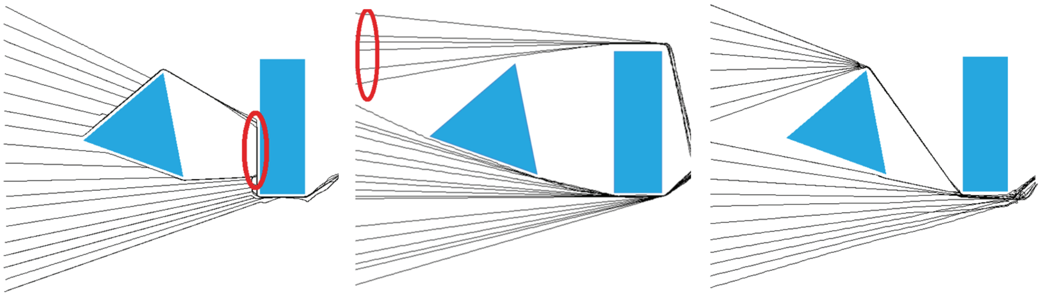
Fig. 8. Experiments on crowd evacuation with different destinations in complex multi-obstacle scene: (a) the destination point is on the upper right side of the scene, (b) the destination point is in the middle of the right side of the scene, and (c) the destination point is at the bottom right of the scene.

Fig. 8 is the experiments on crowd evacuation with different destination in a complex multi-obstacle scene. Pedestrians are randomly located on the left-hand side of each scene. As show from Fig. 8, pedestrians from different starting points plan reasonable evacuation routes leading to different destinations. There is no collision between pedestrians and obstacles, nor evacuate routes longer than the actual ones. It shows that the proposed model can effectively simulate crowd evacuation in multi-obstacle scenes.

5.2 Comparative Experiment

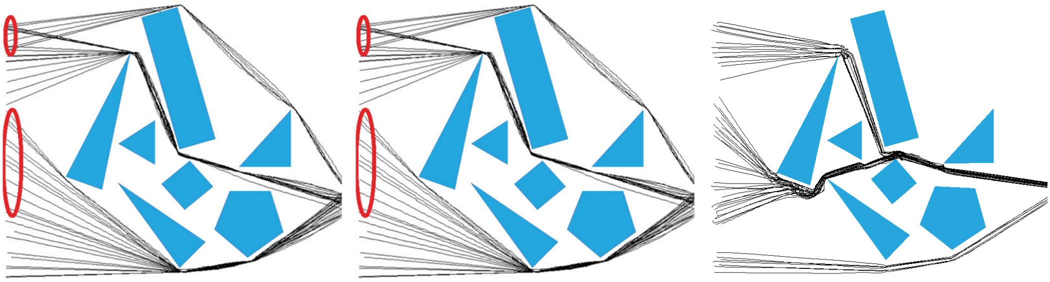
In order to verify the effectiveness of the algorithm, experiments are performed to compare the proposed model with the original SFM and the AnyLogic software. AnyLogic implements a pedestrian library model based on SFM and the shortest route optimization, and it is widely used for pedestrian simulation in complex-scene spaces.

For the experiments, we designed two scenes, with the destination on the right-hand side, and pedestrians at different random positions on the left-hand side, shown in Figs. 9 and 10. From Figs. 9(a) and 10(a), it can be seen that, with SFM, the pedestrians move along the edge of the obstacle, making the path too long and unreasonable. In Fig. 9(a) only some pedestrians can reach the destination point, while most pedestrians stop at the elliptical mark of the obstacle. From Figs. 9(b) and 10(b), we can see that, in AnyLogic, all pedestrians arrive at the destination, but the pedestrians in the elliptical marker area have chosen a longer path, especially in Fig. 10(b). As can be seen in Figs. 9(c) and 10(c), with the proposed model, all pedestrians arrive at the destination and all pedestrians choose the shortest path.



(a) (b) (c)

Fig. 9. Simple scenario results: (a) SFM, (b) AnyLogic, and (c) improved SFM of this paper.



(a) (b) (c)

Fig. 10. Complicated scenario results: (a) SFM, (b) AnyLogic, and (c) improved SFM of this paper.

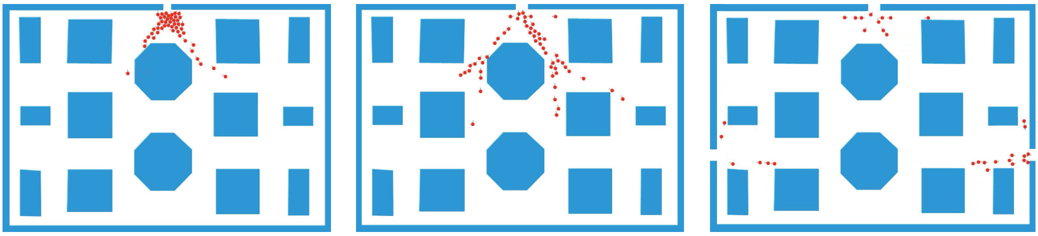
The experimental results show that the proposed model has a great improvement over the original SFM model. It solves the problem of pedestrian movement oscillation, stagnation and unreasonable path of the original SFM model. The experimental results show that the trajectories of the proposed model are very similar to AnyLogic, which indicates the effectiveness of the proposed model. Pedestrians in the AnyLogic model show some randomness when selecting a path, so that the trajectories of some pedestrians are not the shortest path. Assuming that a pedestrian is familiar with the evacuation path and the evacuation scenario has no extra obstacles or blockage, the pedestrian will choose the shortest path for evacuation. The proposed model finds more reasonable paths than the AnyLogic model.

It is worth of adding a few more lines of explanation about this assumption of no extra obstacles. In the scenario, some blocks of various shapes are used to represent obstacles. In a real situation of accident, there might be random fire or blockage caused by some other factors, such as some fell down people. Such situations can be modeled with some more blocks of specific shapes added to the scene. Once the scene is set properly, all such extra random situations could be taken into account, and we can still assume ideal evacuation routes conditions.

5.3 The Proposed Model Application Experiment

The proposed model application experiments were performed on a scene with a size of 75 m×50 m. And the evacuation speed of pedestrians is 4 m/s.

An application experiment was conducted on the rationality of the design of the width and number of exits of the building. The width and number of the exits in the scene are shown in Fig. 11. The number of pedestrians in the experiment was set to 100. Fig. 11 is an effect diagram during the simulation process, and Fig. 12 is an evacuation time comparison diagram of three different designs. It can be seen from Fig. 11(a) that a large number of pedestrians gathered at the exit due to the narrow width of the exit. The pedestrian distribution is arched. At this time, the concentration of pedestrians at the exit is high, and the dangerous accidents such as trampling are extremely likely to occur. It can be seen from Fig. 12 that the evacuation of a scene with only one exit of 2 m takes the longest time. It can be seen from Fig. 11(b) that when the exit width is 3 m, the exit will not be crowded, which will effectively avoid the stampede during pedestrian escape. Therefore, a reasonable exit width can avoid the danger of pedestrian congestion in the evacuation process. Fig. 12 shows that the scene with only one 3 m exit has a longer evacuation time because pedestrians who are farther away from the exit need more time to evacuate. It can be seen from Fig. 11(c), for the scene with three exits, there are no pedestrians crowding at the exit and the evacuation time is the shortest, which design is the most reliable for emergencies.



(a) (b) (c)

Fig. 11. Evacuation simulation renderings. (a) There is a 2-m wide exit in the scene. (b) There is a 3-m wide exit in the scene. (c) There is an exit of 3 m in each of the three directions in the scene.

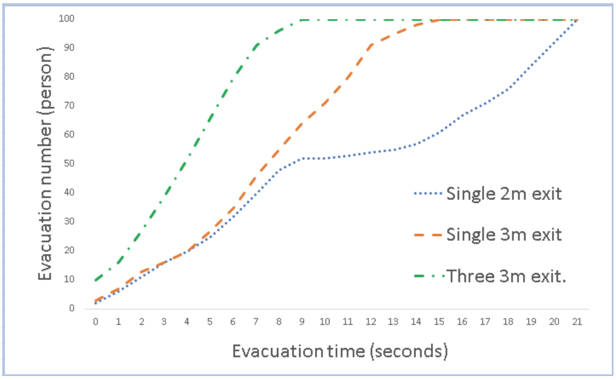


Fig. 12. Evacuation time comparison for different exit type scenarios.

The algorithm can also simulate the evacuation process and the required evacuation time of different numbers of pedestrians in emergencies to provide effective reference information for emergency preparedness plan. Experiments were performed on a scene with an exit of 3 m. The number of pedestrians in the scene is set to be 100, 200, and 300, respectively. Fig. 13 is evacuation simulation renderings of different number of pedestrians, and Fig. 14 is an evacuation time comparison for different number of pedestrians. It can be seen that when the number of pedestrians is 100, there is no crowd gathering at the exit, and the evacuation time is 17 seconds. When the number of pedestrians is 200, pedestrian gathering occurs at the exit, and the evacuation time is 32 seconds. When the number of pedestrians is 300, the pedestrian gathering occurs at the exit, and the area of the gathering is enlarged and prone to a stamped

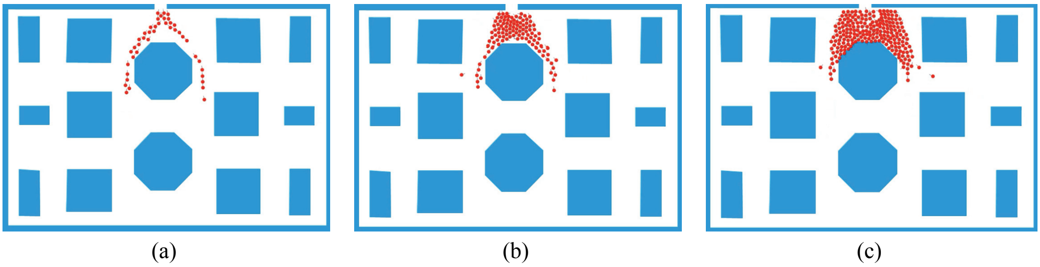


Fig. 13. Evacuation simulation renderings: (a) 100 pedestrians, (b) 200 pedestrians, and (c) 300 pedestrians.

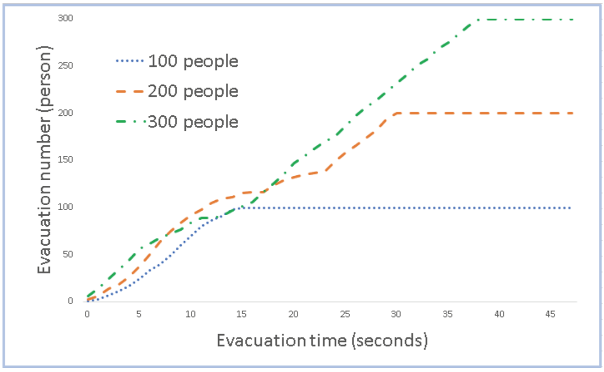


Fig. 14. Evacuation time comparison for different number of pedestrians.

accident, and the evacuation time is 40 seconds. It shows that the scene is not suitable for accommodating more than 300 pedestrians. Therefore, by predicting the evacuation process of different number of pedestrians and the required evacuation time, the reasonable number of pedestrians in buildings can be evaluated.

6. Conclusion

This paper studies the limitations of SFM in crowd simulation and proposes a pedestrian evacuation model for complex multi-obstacle scene. Firstly, the algorithm obtains all the evacuation routes that the pedestrians would possibly choose by adding navigation points and using geometry-based method to construct the evacuation route binary tree. Then it optimizes all the routes, and finally selects the shortest one as pedestrian evacuation route. Experiments show that the algorithm can effectively plan the evacuation route under various scenes, and solve some of the problems with the original SFM, including pedestrian motion trajectory oscillation and stagnation, and poor evacuation route selection.

Crowd simulation is a challenging task, and there are many factors to consider to make the crowd simulation model effective. For example, security factors in the environment can affect the choice of pedestrian evacuation route, the familiarity of pedestrians with the environment also affects the choice, and the scene structure can be very complicated, etc. In the future work, we will take these factors into consideration to improve the proposed model and make the crowd evacuation simulation more accurate.

Acknowledgement

This paper is funded by Zhejiang Provincial Natural Science Foundation of China (No. LY18F010020), and Zhejiang Provincial Natural Science Foundation of China (No. LY19F010001), and Zhejiang Provincial Education Department Foundation of China (No. Y201224456).

References

[1] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282-4286, 1995.

[2] D. Helbing, A. Johansson, and H. Z. Al-Abideen, "Dynamics of crowd disasters: an empirical study," *Physical Review E*, vol. 75, no. 4, article no. 046109, 2007.

[3] Y. Q. Jiang, B. K. Chen, B. H. Wang, W. F. Wong, and B. Y. Cao, "Extended social force model with a dynamic navigation field for bidirectional pedestrian flow," *Frontiers of Physics*, vol. 12, no. 5, article no. 124502, 2017.

[4] N. Cao, Z. Qu, Y. Chen, L. Zhao, X. Song, and Q. Bai, "Destination and route choice models for bidirectional pedestrian flow based on the social force model," *IET Intelligent Transport Systems*, vol. 11, no. 9, pp. 537-545, 2017.

[5] H. Zhang, H. Liu, X. Qin, and B. Liu, "Modified two-layer social force model for emergency earthquake evacuation," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 1107-1119, 2018.

[6] T. Korecki, D. Pałka, and J. Was, "Adaptation of social force model for simulation of downhill skiing," *Journal of Computational Science*, vol. 16, pp. 29-42, 2016.

[7] W. Zeng, P. Chen, H. Nakamura, and M. Iryo-Asano, "Application of social force model to pedestrian behavior analysis at signalized crosswalk," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 143-159, 2014.

[8] L. Hou, J. G. Liu, X. Pan, and B. H. Wang, "A social force evacuation model with the leadership effect," *Physica A: Statistical Mechanics and its Applications*, vol. 400, pp. 93-99, 2014.

[9] B. Anvari, M. G. Bell, A. Sivakumar, and W. Y. Ochieng, "Modelling shared space users via rule-based social force model," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 83-103, 2015.

[10] M. Li, Y. Zhao, L. He, W. Chen, and X. Xu, "The parameter calibration and optimization of social force model for the real-life 2013 Ya'an earthquake evacuation in China," *Safety Science*, vol. 79, pp. 243-253, 2015.

[11] W. Li, J. Gong, P. Yu, S. Shen, R. Li, and Q. Duan, "Simulation and analysis of congestion risk during escalator transfers using a modified social force model," *Physica A: Statistical Mechanics and its Applications*, vol. 420, pp. 28-40, 2015.

[12] Q. G. Ji, R. Chi, and Z. M. Lu, "Anomaly detection and localisation in the crowd scenes using a block-based social force model," *IET Image Processing*, vol. 12, no. 1, pp. 133-137, 2018.

[13] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Heidelberg, Germany: Springer, 2011, pp. 3-19.

[14] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760-772, 1998.

[15] T. Nagatani, "Jamming transition in the traffic-flow model with two-level crossings," *Physical Review E*, vol. 48, no. 5, pp. 3290-3294, 1993.

[16] K. Takimoto and T. Nagatani, "Spatio-temporal distribution of escape time in evacuation process," *Physica A: Statistical Mechanics and its Applications*, vol. 320, pp. 611-621, 2003.

[17] Y. Tajima and T. Nagatani, "Clogging transition of pedestrian flow in T-shaped channel," *Physica A: Statistical Mechanics and its Applications*, vol. 303, no. 1-2, pp. 239-250, 2002.

[18] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90-98, 1986.

[19] O. Khatib, "The potential field approach and operational space formulation in robot control," in *Proceedings of the 4th Yale Workshop on Applications of Adaptive System Theory*, New Haven, CT, 1985, pp. 208-214.

[20] X. Zhang, X. Zhang, Y. Wang, and H. Yu, "Extended social force model-based mean shift for pedestrian tracking under obstacle avoidance," *IET Computer Vision*, vol. 11, no. 1, pp. 1-9, 2017.

[21] S. P. Hoogendoorn and P. H. Bovy, "Pedestrian route-choice and activity scheduling theory and models," *Transportation Research Part B: Methodological*, vol. 38, no. 2, pp. 169-190, 2004.

[22] H. Yue, H. Guan, C. Shao, and X. Zhang, "Simulation of pedestrian evacuation with asymmetrical exits layout," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 2, pp. 198-207, 2011.

[23] A. Jo, T. Sano, Y. Ikehata, and Y. Ohmiya, "Analysis of crowd flow capacity through a door connected to a crowded corridor," *Transportation Research Procedia*, vol. 2, pp. 10-18, 2014.

[24] R. Y. Guo, H. J. Huang, and S. C. Wong, "Route choice in pedestrian evacuation under conditions of good and zero visibility: experimental and simulation results," *Transportation Research Part B: Methodological*, vol. 46, no. 6, pp. 669-686, 2012.

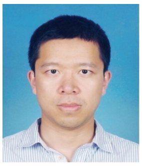
[25] S. J. Kang, Y. Kim, and C. H. Kim, "Live path: adaptive agent navigation in the interactive virtual world," *The Visual Computer*, vol. 26, no. 6-8, pp. 467-476, 2010.

[26] A. Sud, E. Andersen, S. Curtis, M. C. Lin, and D. Manocha, "Real-time path planning in dynamic virtual environments using multiagent navigation graphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 526-538, 2008.

[27] F. Haghpanah, J. Mitrani-Reiser, and B. W. Schafer, "Performance evaluation of pedestrian navigation algorithms for city evacuation modeling," in *Proceedings of the 11th National Conference in Earthquake Engineering*, Los Angeles, CA, 2018.

Jun Li https://orcid.org/0000-0001-8421-3051

She is a lecturer in the School of Electronic and Information Engineering at Ningbo University of Technology. Her research interests include virtual reality, machine learning and multimedia information processing. Li has a Ph.D. in Computer Application Technology from Shanghai University, China. In 2011, she joined Ningbo University of Technology.



Haoxiang Zhang https://orcid.org/0000-0002-5293-9824

He received his Bachelor's degree in Electronic Engineering from Hunan University, China in 2000, and his Ph.D. in Electronic Engineering from University of Bath, UK. His research interests include digital signal and image processing, video coding, and machine learning. He now serves as a lecturer in the Robotics Institute at Ningbo University of Technology.

Zhongrui Ni https://orcid.org/0000-0002-1412-7139

He received the M.S. degree in Computer Application Technology from Ningbo University, Ningbo, China, in 2019. He now serves as a software development manager in the Shanghai Taiyuan Information Technology Co. Ltd. His research interests include virtual reality, machine learning and image processing.