Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Crowd Evacuation Simulation in Flowing Fluids†

Xunjin Zou[1,2] | Yunqing Ye[3] | Zhenmin Zhu[1] | Qiang Chen*[4,5]

[1]School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang, China
[2]Jiangxi Provincial Natural Resources Development Center,Nanchang, China
[3]School of Software, East China Jiaotong University, Nanchang, China
[4]School of Information Management, Jiangxi University of Finance and Economics, Nanchang, China
[5]Jiangxi Vocational College of Science and Technology, Nanchang, China

Correspondence
*Qiang Chen, School of intormation Technology, Mailu Campus of Jiangxi University Of Finance and Economics, 665 West Yuping Road, Nanchang, Jiangxi, China. Email: qiangchen@jxufe.edu.cn
*Yunqing Ye, School of Software, East China Jiaotong University, 808 East Shuanggang Street, Nanchang, China. Email: yeyunqingecjtu@163.com

Summary

In this paper, we propose an integrated model for simulating the interaction between crowds and fluid particles. Our focus is on simulating evacuation motion for crowds in the face of sudden floods. Our model treats both the crowd and the water as fluid particles, which allows us to incorporate various forces such as pressure, shear, buoyancy, and active forces to drive the agents. Additionally, we have designed a minimum rotational path-planning algorithm for agents to search for safe destinations during evacuations. To develop practical crowd evacuation strategies, we observed and studied survival techniques from whirlpools and sudden changes in water levels during floods. Our simulated evacuation results provide plausible strategies for crowds to survive dangerous floods.

KEYWORDS:
Crowd Fluid Interaction, Continuum Dynamics, Crowd Simulation, Path Planning

1 | INTRODUCTION

Simulating the interaction between crowds and fluids is a difficult task with numerous potential applications in disaster management, flood control, land planning, emergency response, and more. Unlike interactions with solid ground, humans not only interact with each other, but also have to navigate the impact of floods. Additionally, when faced with sudden floods, they instinctively move towards nearest safe destinations. As a result, simulating the immediate responses of crowds to flooding is both challenging and critical for practical purposes.

The focus of this paper is the simulation of crowd responses to sudden floods. Rapid floods can impede the movements of crowds and even cause them to lose control and be swept away by turbulent fluids. When simulating the evacuation of crowds rushed by floods using agent-based models, two key problems need to be addressed. The first is the range of interactions between floods and agents. The second is the need for agents to instinctively find nearest safe destinations with minimal effort, given the urgency of the situation.

To address these problems, we extend a Smoothed Particle Hydrodynamics (SPH) fluid simulation model to simulate the flood. Using force interactions, we further animate the plausible motions of agents when rushed by the flood. Additionally, we design a minimum-rotational path planning method for agents to reach nearest safe destinations with minimal effort while avoiding collisions.

The main contributions of this work can be summarized as follows:

†Xunjin Zou and Yunqing Ye are co-first authors of the article.

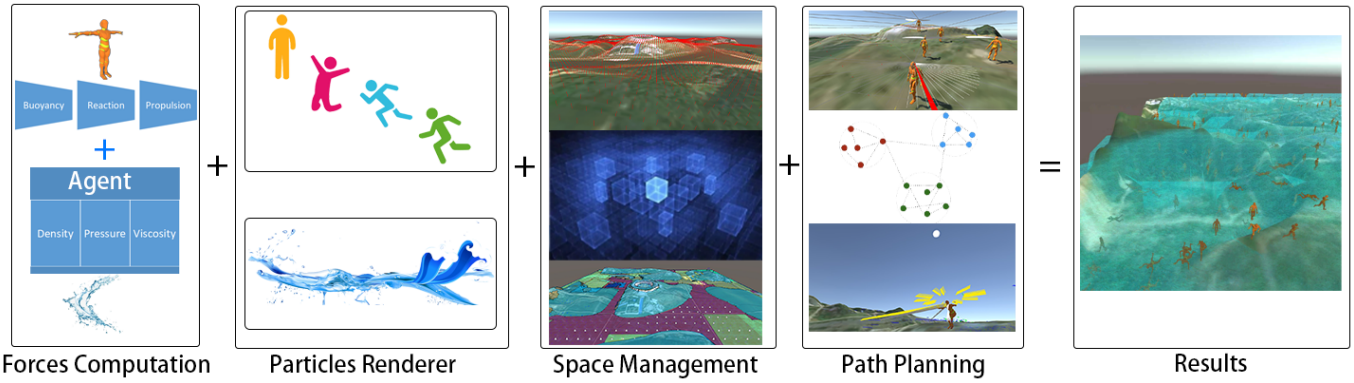

Figure 1 The overview of our model. Our model treats both the crowd and the water as fluid particles, which allows us to incorporate various forces such as pressure, shear, buoyancy, and active forces to drive the agents. The renderer deals different kinds of particle with different ways. A plane partition method is used to divide the 3D scene into a number of zones. The minimal-rotation path finding algorithm ensure the agent moves toward the destination with the total minimal rotation.

- It introduces an integrated model to simulate plausible interactions between crowds and floods for emergency evacuations.
- It introduces a minimum-effort path planning algorithm for crowds to arrive at nearest safe destinations during evacuations.

The remainder of this paper is organized as follows. In Section 2, we review previous related works. Then, in Section 3, we describe our approach. The implementation details of our algorithm are provided in Section 4. In Section 5, we present our experimental results, and in Section 6, we draw conclusions based on our findings.

2 | RELATED WORK

In this section, we briefly review previous related works. For more comprehensive reviews on crowd simulation, readers can refer to [1,2].

The root of agent-based crowd simulation can be traced back to Reynolds' seminal work in 1999 [3], where he demonstrated that simple local rules can generate common crowd behaviors. Meanwhile, many agent-based crowd simulation models have been proposed, including the well-known Social Force Model (SFM) [4]. In the SFM, a simulated agent moves towards a destination while interacting with other agents through forces. Although the SFM has been used and extended to study evacuation scenarios, it may produce oscillatory crowd behavior. To address this limitation, various other models have been proposed, such as velocity-based models [5], vision-based models [6], and learning-based models [7], among others, to simulate different types of evacuations.

Crowd simulations involve not only path planning but also collision avoidance. Funge et al. [8] considered knowledge learning in addition to stimulus responses, and Shao and Terzopoulos [9] further expanded on it. To prevent collisions, collision detection is only triggered when agents are close enough. While agent-based approaches typically do not consider global planning to avoid local collisions, many collision avoidance methods have been developed, including navigation fields [10], grid-based rules [11], density-dependent techniques [12], particle force interaction models [13], and hybrid long-range collision avoidance model [14].

Crowd evacuation simulation models can be used to analyse the direct and potential risks of flood under dynamical emergency conditions. Shirvani et al. [15,16] use hydrodynamic model to simulate the crowd evacuation with static scenery assumption. These models focus on the computational results thus failed to be applied in practical real-time three-dimensional simulation. In recent years, researchers have borrowed and extended continuum models from fluid dynamics theory for crowd simulation. For example, Hughes [17] developed a model that represents pedestrians as a continuous density field and uses a pair of partial differential equations to describe crowd dynamics. Treuille et al. [18] used a similar potential function to guide pedestrians towards their goals and combined pedestrians into groups using "discomfort fields" to handle geographical preferences. Narain et al. [19] proposed a macroscopic continuum model, called the UIC solver, to simulate dense pedestrian crowds. This model is based on Euler fluid theory and uses potential fields to drive the movements of the agents. Other fluid models, such as the Lagrange fluid model [20,21], have also been proposed to simulate fluid-like crowd behaviors.

Continuum models have proven effective in simulating fluid-like motion, particularly for modeling dense pedestrian movements on land. However, currently, there are no established methods to effectively simulate crowd evacuation in flowing water. This is a crucial gap in our understanding, as there are a wide range of potential applications for such simulations, including geological disaster prevention and control, flood control, land planning, and more. Developing effective models for crowd evacuations in aquatic environments would have significant economic and humanitarian benefits for society, and should be a priority for crowd simulation research.

3 | OUR APPROACH

In this section, we describe our approach including its preliminaries. As illustrated in Figure 1, our approach considers both the crowd and the water as fluid particles, enabling us to incorporate various forces, including pressure, shear, buoyancy, and active forces to drive the agents. The renderer handles different types of particles in different ways. We use a plane partition method to divide the 3D scene into multiple zones, and our minimal-rotation pathfinding algorithm ensures that agents move towards their destination with the least possible rotation.

Based on our observations on real world crowd-flood interactions, we make several assumptions that are used to model the interactions between crowds and floods. Finally, we describe correlation functions and force fields used in our approach. Furthermore, we suggest several escape strategies for the agents to survive dangerous area of flood, such as whirlpools and sudden changes in water levels.

3.1 | Preliminaries

In our approach, we assume crowds and water are affected by the force field anytime and anywhere. Also, in this paper we use the following symbols: $\vec{V}$ is the current velocity of the object, $\vec{F}$ represents the current force applied onto the object, $\vec{X}$ represents the current position of the object, all three physical quantities are expressed by vectors.

In our model, the following assumptions are made.

- When the resultant force is zero, the agents tend to stop with a certain decay rate: $V_{p_i} = V_{p_i 0} + \lambda T$ $(0 < \lambda < 1)$, where p_i is the i-th agent, and T is the decay time. The movement tendency of water is derived by the gravitational potential energy when none of external forces is applied to it.

- When a person collides with water, the water is refracted at a relative speed, i.e. the position of the person and the position of water are incompatible. Mathematically, $\vec{X}p_i \neq \vec{X}_{w_j}$, where w_j is the j-th water particle. Furthermore, we assume all of agents can swim in our model.

- The whole three-dimensional space is divided into continuous adjacent cubes. All of the water and lands are divided by polygonal planes according to costs. The accessible area must be continuous and adjacent.

  We use $vol \subseteq R^3$ and vol_i to denote the 3D space and a divided 3D cube, and use pln_i to denote a plane in the 3D space.

$$vol = \left\{vol_0, ..., vol_i, ..., vol_n\right\};$$
$$vol_i \cap vol_j = pln_{ij}.$$

- Humans always choose their evacuation paths with the least effort.

$$\int_{\text{path}} p \ln_i \cos t_i d_p \tag{1}$$

  In our model, we employ the well-known A* algorithm[22] to compute the set of polygonal planes along the optimal evacuation path.

The basic idea of continuous dynamics is that the movement of people and water are treated as interacting particles. We consider people as solid particles and water as liquid particles. The momentum of both particles follow the most basic Newton's second law ($\vec{F} = m\vec{a}$).

3.2 | Fluid Forces Computation

Due to the mass of the fluid is determined by the density of the fluid units, we use the density instead of the mass:

$$\vec{F} = \rho\vec{a}. \tag{2}$$

The forces acting on a fluid particle include three parts:

$$\vec{F} = \vec{F}external + \vec{F}pressure + \vec{F}viscosity. \tag{3}$$

In the above equation, $\vec{F}external$ is the external force, which is generally a gravity ($\vec{F}external = \rho\vec{g}$).

$\vec{F}pressure = -\nabla p$ is the force generated by the pressure difference inside the fluid. This force drives the particles to move from high-pressure zones to low-pressure zones. This force can be computed as the gradient of the pressure field, and its direction points from the region of high pressure to the region of low pressure.

$\vec{F}viscosity = \mu\nabla^2\vec{v}$ is the viscous force caused by the velocity difference between the particles, similar to the shear force, from the fast part to the slow part, and the magnitude of this force is related to the viscosity coefficient μ of the fluid. Based on the SPH fluid model[23], we compute the density, pressure force, and viscous force, described below.

Density. The computation of the density ρ with a kernel function is as follows:

$$\rho(\vec{x}_i) = \sum_j m_j W_\rho(\vec{x}_i - \vec{x}_j, h). \tag{4}$$

The specific form of the smooth kernel function $W_\rho(\cdot)$ is as follows:

$$W_\rho(\vec{x}, h) = \frac{315}{64\pi h^9}(h^2 - x^2)^3, (x = \left|\vec{x}\right|, 0 \le x \le h). \tag{5}$$

Pressure. The pressure p of a single particle can be calculated using the ideal gas equation as follows:

$$p = k(\rho - \rho_0), \tag{6}$$

where ρ_0 is the static density of the fluid and k is the constant associated with the properties of the fluid, usually the temperature. We choose the optical kernel function $W_p(\cdot)$ to compute the pressure as follows:

$$W_p = \frac{15}{\pi \mathrm{h}^6}(h - x)^3, (x = \left|\vec{x}_i - \vec{x}_j\right|, 0 \le x \le h), \tag{7}$$

$$\vec{F}_{pressure,i} = -m\sum_j (\frac{p_i + p_j}{2\rho_i\rho_j}\nabla W_p(\vec{x}, h)), (m \equiv m_j). \tag{8}$$

Viscosity. Similarly, we can also derive the force generated by the viscosity. The viscous force computation is as follows:

$$\vec{F}_{viscosity,i} = \mu\sum_j m\frac{\vec{v}_i - \vec{v}_j}{\rho_j}W_v(\vec{x}, h), \tag{9}$$

where μ is the viscosity coefficient, and the adopted optical kernel function $W_v(\cdot)$ is as follows:

$$W_v(\vec{x}, h) = \frac{15}{2\pi h^3}(-\frac{x^3}{2h^3} + \frac{x^2}{h^2} + \frac{h}{2x} - 1), (x = \left|\vec{x}_i - \vec{x}_j\right|, 0 \le x \le h). \tag{10}$$

3.3 | Agent Forces

In addition to the influence of gravity, pressure, and shear forces, agents can also be affected by other forces of water, i.e., buoyancy. Furthermore, a person may move aimlessly while confronting floods because for the nervousness. In our model, we introduce a buoyancy force and a causal force for the agents. Normally, buoyancy requires calculating the volume of water flowing through the human body per unit of time. In our model, we simplify the velocity calculation of the flow within the radius.

The equation for computing the buoyancy force for the agent p_i is as follows:

$$\vec{F}_{buoyancy,i} = \mu m_i\frac{45}{\pi h^6}\sum_j \frac{\vec{v}_{wj}}{\rho_j}(h - \left|\vec{x}_{pi} - \vec{x}_{w_j}\right|), \tag{11}$$

where m_i is the mass of the agent p_i, and w_j represents the j-th water particle.

The casual force designed for the agent is as follows:

$$\vec{F}_{casual} = \vec{f}, \quad \left|\vec{f}\right| \le \eta, \tag{12}$$

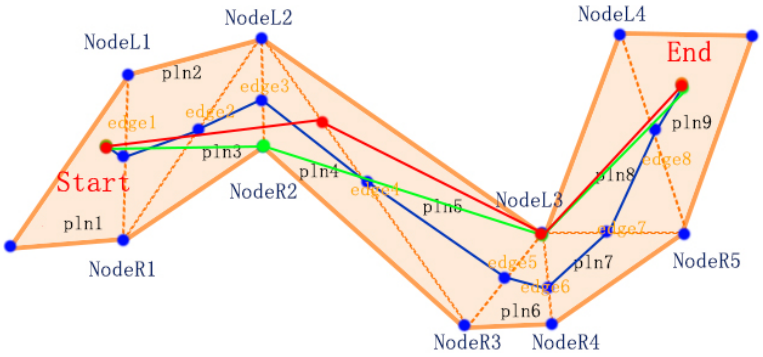

Figure 2 Comparison of three path planning methods: the midpoint method (blue line), the inflection point method[24] (green line), and our minimal-rotation path planning method (red line).

where η is a constant.

In a nutshell, we treat the agent as a fluid particle with an active motion. To this end, the resultant force acting on an agent can be computed as follows:

$$\vec{F}_{agent} = \vec{F}_{external} + \vec{F}_{pressure} + \vec{F}_{viscosity} + \vec{F}_{buoyancy} + \vec{F}_{casual}. \tag{13}$$

3.4 | Global Path Planning

We use a plane partition method to divide the 3D scene into a number of zones. Agents determine their optimal paths according to the zones in which they are positioned. As shown in Figure 2, there exists a set of polygons from the “Start" location to the “End" location while every two adjacent polygons have only one intersecting segment.

Minimal-rotation path finding algorithm. We introduce a minimal-rotation path finding algorithm in our model. Its goal is to ensure the agent moves toward the destination with the total minimal rotation. Its basic idea is as follows: We first initialize a trace queue to be empty. Then, we connect the starting point to the ending point to form a straight line (called the to-target line). Among all the internal edges that are intersected with the to-target line, we identify the internal edge with the minimum distance to the end point. We add this inter-sectional point to the end of the trace queue. Otherwise, if no such internal edges exists (that is, no internal edge intersects with the to-target line), we calculate the two angles between the to-target line and the two end-points of the closest internal edge, respectively, and choose the end point with the smaller angle. This chosen end point is also added to the end of the trace queue. As long as the trace queue is not empty, we will take the point at the first place in the queue to re-calculate the to-target line and repeat the above process until the trace queue is empty. The red line in Figure 2 shows the generated path in this example.

In Figure 2, we compare our path planning method with two well-known path finding methods: the midpoint method, and the inflection point method[24]. As shown in this figure, the path generated by our method has minimal rotations along the path and thus leads to smallest oscillations.

4 | IMPLEMENTATION DETAILS

In this section, we describe some implementation details of our method, including spatial division, target selection, and evacuation strategies.

4.1 | Spatial Division

We use vol_i to a 3D cell in the space (Figure 3), and the 3D space $Vol = \{vol_0, ..., vol_i, ..., vol_n\}$. All the agents and water are stored in the corresponding cells according to their positions. In a single 3D cell, the positions of agents and water are deposited in the adjacent tables according to their adjacent relationships.

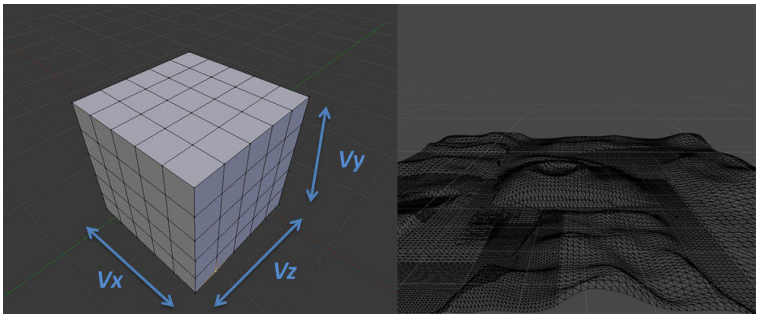

Figure 3 Subdivision of the 3D space into cells.

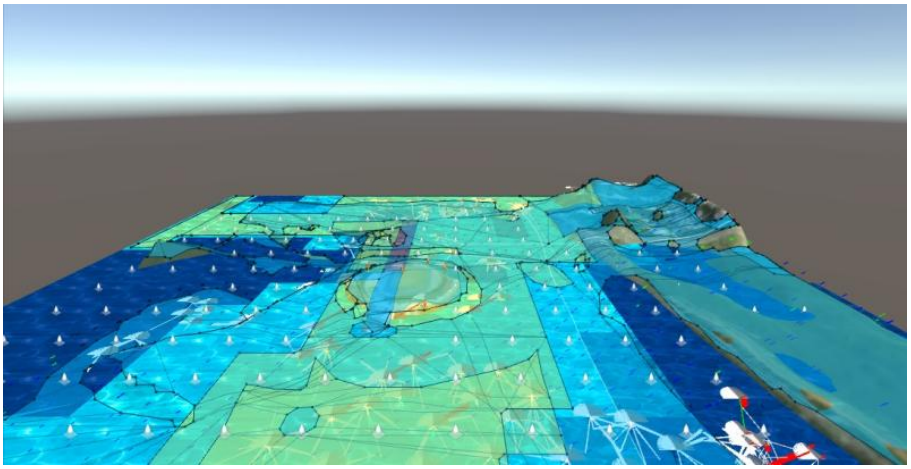

Figure 4 Spatial plane division.

Assuming $p\ln_i$ denotes a plane in the space, the set of planes, $P\ln = \{p\ln_0, ...p\ln_i, ... p\ln_n\}$, is sequentially stored into an array (Figure 4). In our approach we calculate the cost of a plane according to the number of agents and the flow speed of the water within the plane, described below.

$$\text{Cost}\,(pln_i) = \alpha \int pln_i p_k d_p + \beta \int pln_i w_k d_p, \tag{14}$$

where p_k denotes the k-th agent, and w_k denotes the k-th water particle.

4.2 | Target Selection

In the real world, people always search for the accessible safe area through their eyes during evacuation. Our model also emulates this point. In our evacuation simulation, an agent who emits rays from the current position also constantly looks for the target point (Figure 5). The target points will be selected based on the following principles:

- It is close to the current position;
- It has the bearing capacity;
- It is reachable from the current position.

Routing strategy. First, based on the minimal-rotation path finding algorithm (described above), agents move toward the target along the planned path with the least rotation. Second, when the agents feel pressure from neighboring agents, their active speeds will slow down, and the active force will be stopped when the neighboring agent density reaches a specified threshold

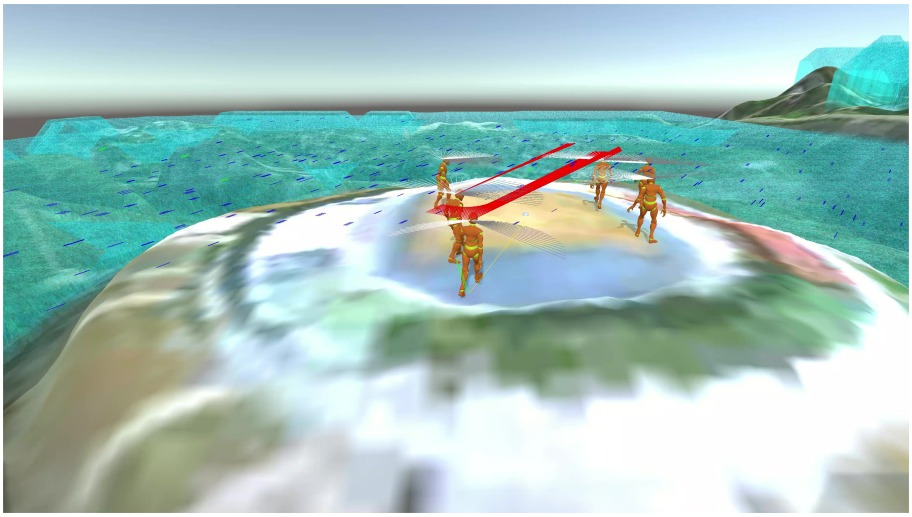

Figure 5 Illustration of the routing strategy

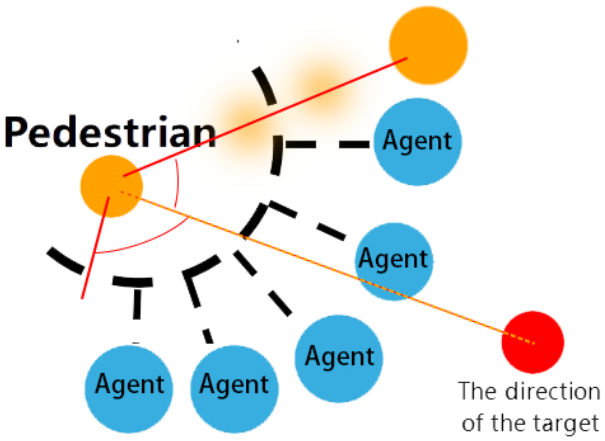

Figure 6 The low-density area is selected with the minimum rotation towards the target.

level. In this case, agents then move with the crowd traction force, i.e, viscous force. Third, when the agent density drops below the threshold level, the agents will re-start the global path planning and resume to move towards the target.

As illustrated in Figure 6, in our model, when agent scan in the front direction towards the target, they will scan from left to right until the occlusion-free direction with the minimum angle is identified.

4.3 | Evacuation Strategies

Our model can be used to simulate the shock of a flood for crowds. In the following, we describe the used evacuation strategies for several different cases.

Evacuation from whirlpools. A typical whirlpool is illustrated in Figure 7. Water is identified as whirlpool areas as follows:

- In a current 3D cell Vol_i, there is a downward trend of water flow;
- In the Vol_i, the dispersion of the current flow velocity is in the warning interval [DivergenceL, DivergenceH];
- In the Vol_i, there are water flow disappearance points within the radius of Detection-R.

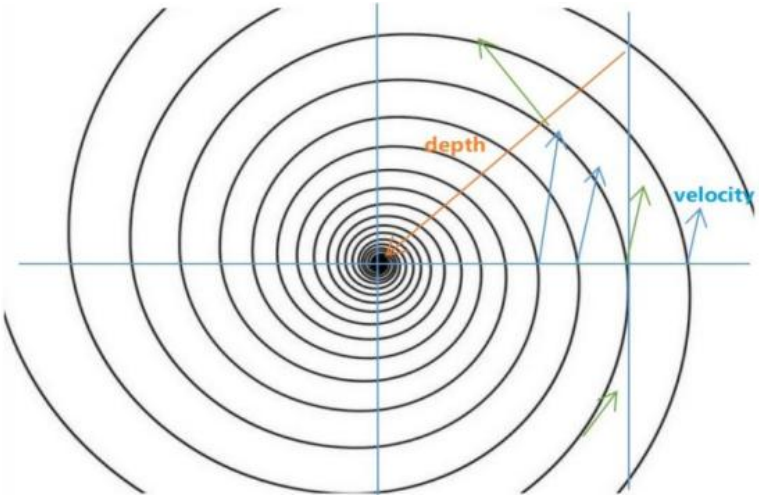

Figure 7 Illustration of a typical whirlpool.

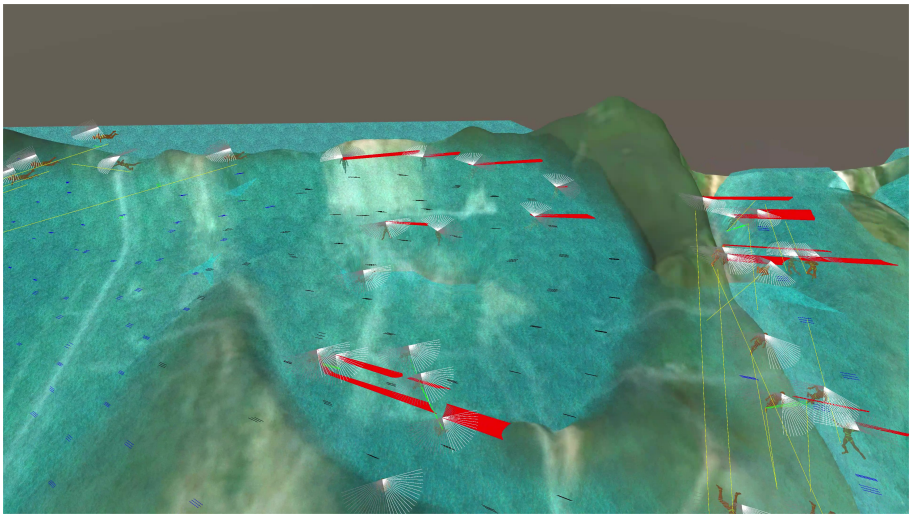

Figure 8 Evacuation simulation from a whirlpool.

The formula for calculating the dispersion of water flow in the current cell vol_i is as follows:

$$Divergence_i = \nabla \vec{V}(\vec{x}_i). \tag{15}$$

Once the whirlpool can be detected, agents can take a simple strategy by following the direction of the current and moving away from the spiral center.

Evacuation from sudden changes in water levels during floods. As illustrated in Figure 9, we use a simple method to detect the water level mutation when meeting the following conditions:

- In the current cell vol_i, the velocity of the current flow in the vertical direction increases over the threshold value of $FallLimit$.

$$\vec{V} \bullet (0, -1, 0) \geq FallLimit. \tag{16}$$

- In the current vol_i, the water flow drop exceeds the threshold value $DropLimit$.

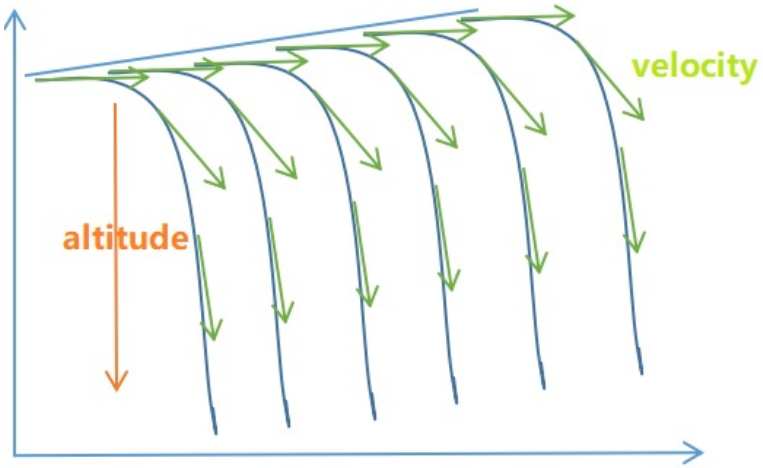

Figure 9 An illustration of the mutation of the water levels of floods.

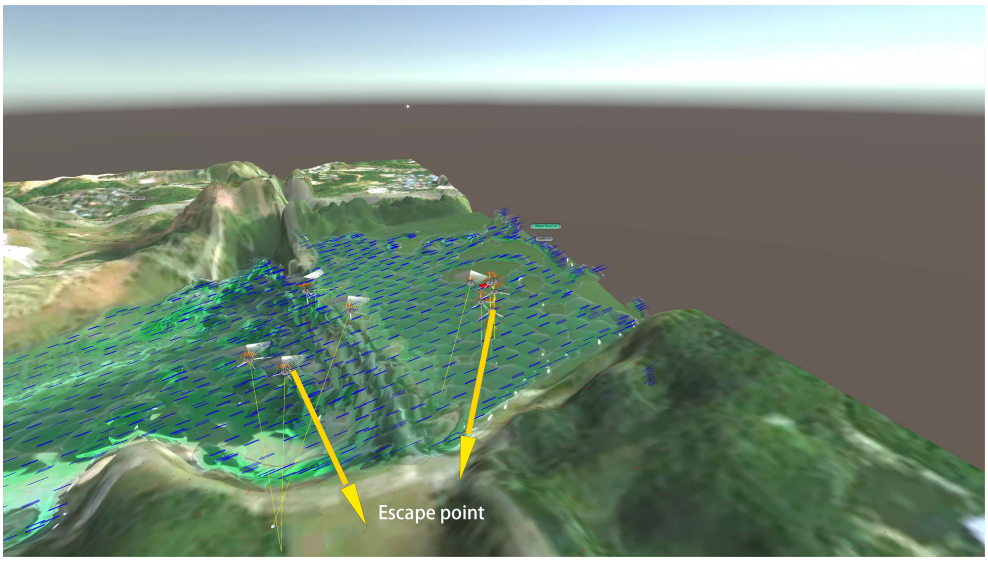

Figure 10 Evacuation from sudden changes of water levels during floods.

We use a simple strategy to evacuate from the mutation of the water levels of floods. When confronting an upstream, agents try to depart from the current and move toward an area with a small flow rate. The agents attempt to move towards an area with a small flow rate along the flow direction, when facing a downstream (refer to Figure 10).

5 | EXPERIMENTAL RESULTS

We implemented our approach in Unity 3D with C# language. The simulation system runs stably at a speed of more than 24 FPS on a PC equipped with Intel (R) i5 3.20GHz CPU and 8G RAM. Our experiments were focused on the demonstration of two aspects. First, we simulated the interaction between crowds and floods. Second, when a dangerous situation occurs, we simulated how agents evacuate from the dangerous area. We describe our experiment below. For more simulation details, please refer to the demo video.

Scenario 1: Water impact. In this experiment, we simulated the crowd was impacted by water flow running at five meters per second. The numbers of simulated agents were 1, 10 and 100, respectively. The agents walked cross a river while their movements were influenced by the water flow. Figure 11 shows a snapshot of the simulated result.

Figure 11 The simulated agents walked cross a river.

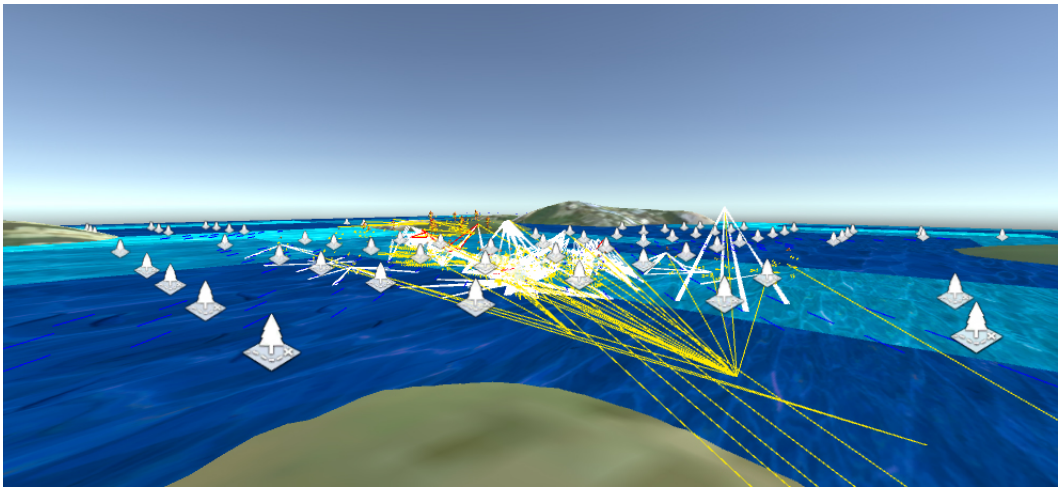

Figure 12 Path planning in the flood.

Scenario 2: Path planning in the flood. In this experiment, we simulated the evacuation of agents when they were flushed by a sudden flood. Based on our model, the agents searched for a possible safe target to evacuate from danger. The agents dynamically adapted the evacuation path according to dynamic environment. Meanwhile, the agents tried to avoid colliding with each other, dangerous water, and other dynamic factors. The simulated result is shown in Figure 12.

Scenario 3: Evacuation from whirlpools. In this experiment, we simulated agents to be carried to dangerous areas by a sudden flood. Due to different distributions of terrains, dynamic water flow is likely to form dangerous areas, such as whirlpools. Based on our model, agents will follow the directions of currents and move away from the spiral center. The simulated result is shown in Figure 8.

Scenario 4: Evacuation from sudden changes of water levels during floods. We simulated crowd evacuation from the mutation of water levels of floods. In our model, agents attempt to depart from the currents and move towards an area with a small flow rate when confronting an upstream. Conversely, the agents attempt to move towards an area with a small flow rate along the flow direction when facing a downstream. Our simulation result shows that the agents with plausible motions can survives when facing the sudden mutation of water levels of floods (Figure 10).

Scenario 5: Short-term decisions to evacuate from a flood. Agents may lose the global evacuation context but desperately obtain guidance from surrounding objects when flushed by a sudden flood. Our model can simulate the short-term decisions for

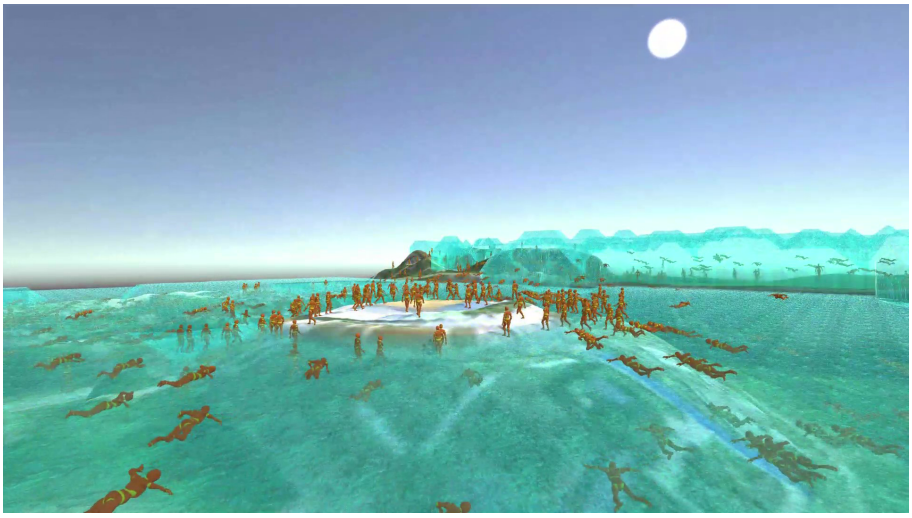

Figure 13 Agents move towards a possible safe destination while flushed by a flood.

the agents to arrive at the nearest possible safe destination. As shown in Figure 13 (also refer to the supplementary demo video), agents tried to move toward a safe destination according to the current situation. The evacuation path information was obtained from neighbors.

6 | CONCLUSION AND FUTURE WORK

In real-life scenarios, crowds often have little opportunity to avoid dangerous floods, especially when facing whirlpools or sudden changes in water levels without early warning. However, appropriate decision-making can increase the chances of successfully evacuating from the flood. In this paper, we introduce a practical model to simulate both fluid particles and agents, and propose a minimum-effort evacuation path planning algorithm to mitigate the impact of sudden floods. Our experiments demonstrate that our model provides a plausible simulator for the behavior of crowds in response to sudden floods, with a focus on both global aspects such as dynamic force field construction, global path planning, and crowd distribution, as well as local aspects such as short-term path decision-making, water detection, and collision avoidance.

Villages worldwide are often built near rivers, embankments, reservoirs, and potential geological disaster areas. Our model has significant implications for natural disaster early warning and forecasting, flood management, emergency response, land planning, and other potential applications. However, our model do not deal with the decay of tangent velocity and do not analyse the influence of the body size of the agent and the versatile magnitude of the floods. In our future work, we aim to conduct more comprehensive experiments that explore the following aspects in greater detail:

Firstly, we will incorporate more complex scenarios into our simulations, which will involve analyzing the impact of various factors such as buildings, ships, and other obstacles on the model.

Secondly, we intend to extend our model to cover a broader range of fields such as debris flow, landslide, tsunami, and other natural phenomena.

Lastly, we plan to expand our particle types and incorporate more diverse elements in the same scene to achieve a more realistic simulation.

ACKNOWLEDGMENTS

Zhenmin Zhu was supported by the NSFC (Grant No.52065024), Jiangxi Province Key R&D Program (Grant No. 20202BBE53022, Grant No.20223BBE51010), Jiangxi Province 03 Special Project (Grant No.20212ABC03A20). Qiang Chen was supported in part by the China NSFC (Grant No. 62262024), Scientific Research Project of Education Department of Jiangxi Province (Grant Nos. GJJ210511, GJJ207109).

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

References

1. Thalmann D, Musse SR. *Crowd simulation*. Springer Science & Business Media . 2012.
2. Yang S, Li T, Gong X, Peng B, Hu J. A review on crowd simulation and modeling. *Graphical Models* 2020; 111: 101081.
3. Reynolds CW, others . Steering behaviors for autonomous characters. In: . 1999. Citeseer. ; 1999: 763–782.
4. Helbing D, Molnar P. Social force model for pedestrian dynamics. *Physical review E* 1995; 51(5): 4282.
5. Berg V. dJ, Lin M, Manocha D. Reciprocal velocity obstacles for real-time multi-agent navigation. In: Ieee. ; 2008: 1928–1935.
6. Ondřej J, Pettré J, Olivier AH, Donikian S. A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics (TOG)* 2010; 29(4): 1–9.
7. Hu K, Haworth B, Berseth G, Pavlovic V, Faloutsos P, Kapadia M. Heterogeneous crowd simulation using parametric reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics* 2021.
8. Funge J, Tu X, Terzopoulos D. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In: ; 1999: 29–38.
9. Shao W, Terzopoulos D. Autonomous pedestrians. In: ; 2005: 19–28.
10. Patil S, Van Den Berg J, Curtis S, Lin MC, Manocha D. Directing crowd simulations using navigation fields. *IEEE transactions on visualization and computer graphics* 2010; 17(2): 244–254.
11. Loscos C, Marchal D, Meyer A. Intuitive crowd behavior in dense urban environments using local laws. In: IEEE. ; 2003: 122–129.
12. Best A, Narang S, Curtis S, Manocha D. DenseSense: Interactive Crowd Simulation using Density-Dependent Filters.. In: ; 2014: 97–102.
13. Heigeas L, Luciani A, Thollot J, Castagné N. A Physically-Based Particle Model of Emergent Crowd Behaviors. In: ; 2003: 1–9.
14. Golas A, Narain R, Lin M. Hybrid long-range collision avoidance for crowd simulation. In: ; 2013: 29–36.
15. Shirvani M, Kesserwani G, Richmond P. Agent-based simulator of dynamic flood-people interactions. 2019.
16. Shirvani M, Kesserwani G, Richmond P. Agent-based simulator of dynamic floodeople interactions. *Journal of Flood Risk Management* 2021(1).
17. Clements RR, Hughes RL. Mathematical modelling of a mediaeval battle: the Battle of Agincourt, 1415. *Mathematics Computers in Simulation* 2004; 64(2): 259-269.
18. Treuille A, Cooper S, Popovic Z. Continuum crowds. *Acm Transactions on Graphics* 2006; 25(3): 1160-1168.
19. Narain R, Golas A, Curtis S, Lin MC. Aggregate dynamics for dense crowd simulation. In: 2009 (pp. 1–8).
20. Chen Q, Luo G, Tong Y, Jin X, Deng Z. A linear wave propagation-based simulation model for dense and polarized crowds. *Computer Animation and Virtual Worlds* 2021; 32(1): e1977.
21. Toll vW, Chatagnon T, Braga C, Solenthaler B, Pettré J. SPH crowds: Agent-based crowd simulation up to extreme densities using fluid dynamics. *Computers & Graphics* 2021; 98: 306–321.

22. Bayazit OB, Lien JM, Amato NM. Better group behaviors in complex environments using global roadmaps. *Artif. Life* 2002: 362–370.

23. Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications. In: Citeseer. ; 2003: 154–159.

24. Demyen D, Buro M. Efficient triangulation-based pathfinding. In: ; 2006: 942–947.