

Novel Integrated Framework for Crowd Simulation

Qingge Ji* and Yugang Han

School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China

issjqg@mail.sysu.edu.cn

Abstract. In this paper, two intuitive and highly efficient solutions are proposed for global planning and local avoidance. We introduce guide and repel vectors to study global planning, which generates a steady and smooth navigation field through a simple and efficient bilinear interpolation method. In addition, this paper proposes a novel velocity-based approach to simulate the local avoidance of agents based on least-effort principle. During the local avoidance phase, humans slightly adjust their motions, so that the energy required to perform a step becomes minimal. The two solutions are integrated into one system, which finally simulates the natural-looking navigation and interaction behavior of agents.

Keywords: Crowd Simulation, Global Planning, Navigation Fields, Local Avoidance, Least effort

1 Introduction

As virtual reality technology develops, crowd simulation technology is paid increasing attention. According to different modeling granularities, existing crowd models can be generally classified into two categories, namely, macroscopic and microscopic approaches. The former models a crowd as continuous flow of fluid [1]. This technology is mainly useful in large and dense crowds but basically neglects the features of individuals. The latter models a crowd as a collective of homogeneous/heterogeneous entities with interactions among them, and the representative approaches include entity-based and agent-based. Individuals are modeled as a set of homogenous entities in the entity-based approach. A typical example of this approach is Helbing's social force model (SFM) [2]. The agent-based approach models each individual in a crowd as an intelligent and autonomous agent [3], in which each agent perceives its own state and reacts to dynamic entities in its neighborhood. The microscopic approach models are flexible, such that adding physical, social, and psychological factors can simulate various interactive behavior. As a result, these models are the most popular ones. However, their computing cost is high. Jiang et al.[4] presented a semantic model for representing the complex environment, where the semantic information is described with a geometric level, a semantic level and an application level. The model promotes the interactions between pedestrians and the environment.

Kraayenbrink et al. [5] proposed semantic crowds that allowed one to re-use the same population for virtual environment.

Main Contribution: Based on previous research, two intuitive and highly efficient solutions are proposed in this paper for global planning and collision avoidance.

We introduce guide and repel vectors to study global planning, which generates a steady and smooth navigation field through a simple and efficient bilinear interpolation method. In addition, we propose a novel velocity-based approach to simulating the collision avoidance of agents through the observation of human behavior in avoiding dynamic obstacles in real life.

2 Related Work

In this section, we briefly discuss prior literature on global planning and local avoidance, which are the two key issues in crowd modeling technology.

Global Planning: To navigate a complex environment, a high-level path planning technology is needed. The most popular crowd navigation technologies include graph search and potential fields. Graph-based algorithms are widely used in global planning [6]. Pettre et al. [7] proposed a graph structure that decomposes a space into multilayered terrains to support fast graph search for multiple characters. Bandi et al. [8] extended A* algorithm to a 3D space and reproduced many interesting navigation behaviors. Roadmaps [9] and Voronoi diagrams [10] are recently introduced to crowd navigation. Potential fields are extensively studied in robot motion planning [11]. Dapper et al. [12] introduced harmonic function to generate potential fields; thus, they would not fall into the local minimum and could simulate various navigation behaviors by adjusting the parameters in the function. Moreover, many researchers have directly attempted to govern navigation by computing velocity fields based on environment description [13], designing velocity fields manually [14], or capturing the velocity fields from videos and user inputs [15]. Our global planning algorithm is inspired by [13]. We introduce two types of vectors, namely, repel and guide vectors. An efficient bilinear interpolation method is used to obtain smooth navigation fields.

Local Avoidance: Collision should be avoided locally by adjusting movements when other agents become sufficiently close. Many local avoidance approaches have been proposed, including particle force interaction [16], geometric [17], and synthetic-vision models [18]. Many researchers have introduced velocity-based methods for collision avoidance recently. Paris and Pettre et al. [19] proposed a predictive approach and resolved potential collisions successfully. Karamouzas and Overmars [20] proposed a velocity-based approach by analyzing experimental data and extended this approach to small groups [21]. Koh and Zhou [22] introduced a collision avoidance framework called relative frame. According to

the duality property of the relative frame and other constraints, they selected a collision-free velocity for an agent. Our local avoidance algorithm is inspired by the work of Koh and Zhou. We use a modified relative frame to predict the potential collision and select an optimal velocity for an agent. However, unlike Koh and Zhou, we adopt the least-effort principle and eventually obtain a realistic and natural-looking result.

3 Global Path Planning

3.1 Environment Decomposition and Organization

To compute a global path to the goal for each agent, we decompose the environment into grids, which have different size and are represented by rectangles. When static obstacles are dense, our method will subdivide the environment until each mixed grid is almost occupied by obstacles; when static obstacles are sparse, our decomposition method roughly divides the environment into several grids, then merges the empty grids, and forms a large empty area.

We use a four-connected graph to organize the empty grids. The connective graph is defined to be the graph that has a vertex for each grid and an edge between two vertices only if the corresponding grids share a segment on their boundaries. A path over this graph is computed, such that following the path from any vertex leads to the vertex corresponding to the grid containing the goal state. The resulting directed graph defines a successor for every grid, except the goal grid. The successor of a grid is the next grid on the path to the goal grid. Each grid with a successor is termed as an intermediate grid, and the intermediate grid has only one successor. Specially, the goal grid has no successor. See Figure 1 for an illustration.

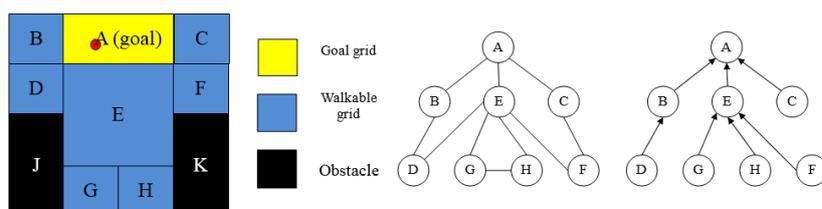


Fig. 1. Environment decomposed into grids and the corresponding connectivity and directed graphs.

3.2 Repel and Guide Vectors

We assume that each grid has four adjacent grids (because the graph is four-connected). A grid must be set as the successor of the grid. The shared boundary is called exit face, and the others are called repel faces. See Figure 2 for an

illustration. To obtain a proper transition from the current grid to successor, we introduce two types of vector fields, i.e., those corresponding to grids in the decomposition, which we call guide vector fields, and those corresponding to faces, which we call repel vector fields. A guide vector field guides an agent through the grid to the exit face, which leads to the successor grid. Repel vector fields prevent an improper grid transition, i.e., a transition from the current grid to a grid that is not the successor is prohibited. For the repel vector on repel faces, its direction is orthogonal to the face and points inward. For the repel vector on the exit face, its direction is orthogonal to the exit face and points outward. The guide vector fields always point toward the exit face. In the case of the goal grid, all repel and guide vector fields point inward to the goal state.

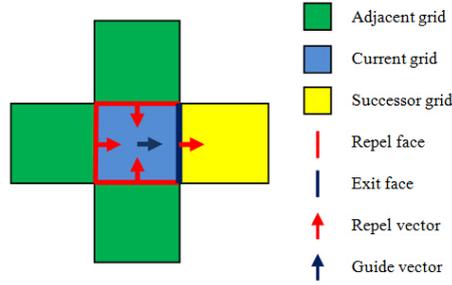


Fig. 2. Illustration for repel face, exit face, repel vector, and guide vector.

The different sizes between adjacent grids pose a difficulty in choosing the appropriate repel or guide vector fields. Zhang et al. [23] proposed a method to resolve this problem.

3.3 Vector Interpolation

To obtain a smooth transition from the current grid to successor for an agent, an efficient and simple bilinear interpolation method is used to compute the final repel vector \mathbf{V}_{repel} (Figure 3). We assume that the position of agent (x_i, y_i) is in grid $C = \{(x_1, y_1), (x_2, y_2)\}$, and its successor is $S = \{(x_3, y_3), (x_4, y_4)\}$, where $\{(x_1, y_1), (x_2, y_2)\}$ and $\{(x_3, y_3), (x_4, y_4)\}$ represent the upper left and lower right vertex coordinates of C and S, respectively. The repel vector set of grid C is $\mathbf{F} = \{\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$.

$$\mathbf{V}_{repel} = \frac{x_2 - x_i}{x_2 - x_1} * \mathbf{f}_0 + \frac{x_i - x_1}{x_2 - x_1} * \mathbf{f}_2 + \frac{y_2 - y_i}{y_2 - y_1} * \mathbf{f}_1 + \frac{y_i - y_1}{y_2 - y_1} * \mathbf{f}_3 \quad (1)$$

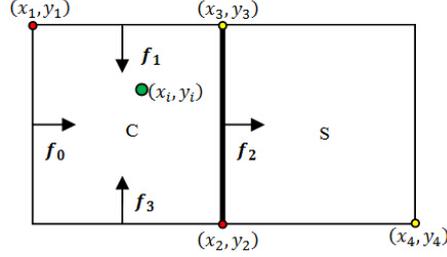


Fig. 3. Computation of the final repel vector V_{repel} .

Considering that the grid size might differ, two cases are considered for f_2 . Figure 4 shows that when the current position of agent (x_i, y_i) locates below the green-dotted line, $f_2 = f_{virtual}$, when (x_i, y_i) locates above the green-dotted line, $f_2 = f_{21}$. $f_{virtual}$ represents the repel vector on the virtual face, and f_{21} represents the repel vector on the exit face.

$$f_2 = \begin{cases} f_{21} & y_i > y_3 \wedge y_i < y_4 \\ f_{virtual} & y_i \geq y_4 \wedge y_i \leq y_2 \end{cases} \quad (2)$$

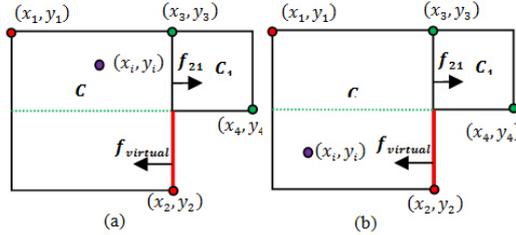


Fig. 4. Selection of f_2 in two cases.

Assuming that the guide vector is V_{guide} , we calculate the linear interpolation between V_{repel} and V_{guide} , and obtain the navigation vector at (x_i, y_i) , denoted as V_{nav} .

$$V_{nav} = \alpha * V_{repel} + \beta * V_{guide} \quad (3)$$

We suppose that $\alpha = 0.5$ and $\beta = 0.5$. We can calculate the navigation vector of each spot in the configuration space using Equation (3). Disregarding other agents, each agent can move step by step along the direction of V_{nav} to the goal state. Figure 5(a) shows an example for the navigation fields, and Figure 5(b) shows the path of an agent moving from the initial point to the goal state. No steep turn exists in the corners, and the whole path is smooth, which vividly simulates the human behavior when turning in our real life.

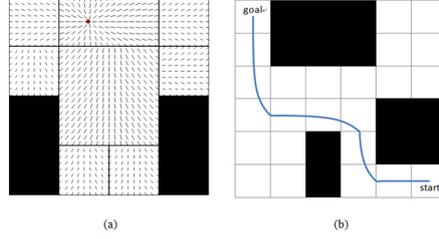


Fig. 5. (a) Example for the navigation fields. The black rectangle denotes obstacle, and the red point denotes the goal state. (b) Path for an agent from the initial position to the goal state.

4 Local Collision Avoidance

Two main challenges occur in local collision avoidance, namely, collision prediction and collision avoidance. In this section, we describe our collision avoidance model.

4.1 Problem Formulation

In our problem setting, we are given a virtual environment where n agents $P_N = \{P_1, \dots, P_n\}$ have to navigate toward their specified goal without colliding with the environment and with one another. For simplicity, we assume that each agent moves on a plane and is modeled as a disc with radius r_i , and its personal space is modeled as a disc with radius ρ_i . At a fixed time t , the agent P_i is at the position $\mathbf{x}_i(t)$, defined by the disc center, and moves with velocity $\mathbf{v}_i(t)$. This velocity is limited by a maximum speed u_i^{max} , i.e., $\|\mathbf{v}_i(t)\| \leq u_i^{max}$. For notational convenience, we will not explicitly indicate the time dependence.

In every simulation step, the agent P_i has a desired velocity $\mathbf{v}_i^{des}(t)$, whose orientation is \mathbf{V}_{nav} , which have been computed in Section 3, and magnitude is u_i^{des} , which is closely related to the crowd density ρ according to Fang et al. [24].

$$\mathbf{v}_i^{des} = u_i^{des} * \frac{\mathbf{V}_{nav}}{\|\mathbf{V}_{nav}\|} \quad (4)$$

$$u_i^{des} = \begin{cases} u_i^{max} & \rho \leq \rho_{min} \\ u_i^{min} + \frac{\rho - \rho_{min}}{\rho_{max} - \rho_{min}} * (u_i^{max} - u_i^{min}) & \rho_{min} < \rho < \rho_{max} \\ \bar{u} & \rho \geq \rho_{max} \end{cases} \quad (5)$$

In the above equations, ρ_{min} and ρ_{max} are the minimum and maximum crowd density thresholds, respectively. \bar{u} is the average speed of all agents, which are in the vision range of P_i 's vision range.

4.2 Collision Prediction

An agent configuration is defined by its position and velocity. Koh and Zhou proposed a relative frame model for collision prediction. Source agent is denoted as the agent that avoids a target agent. Figure 6 shows the relative frame between a source agent and a target agent, where v_r is the relative velocity between the source and target agents; θ_s and θ_g are the orientation of the source and target agents, respectively; θ_r is the relative orientation between the source and target agents. $R_g = r_g + \rho_s$, it means that the target agent should not invade the personal space of the source agent.

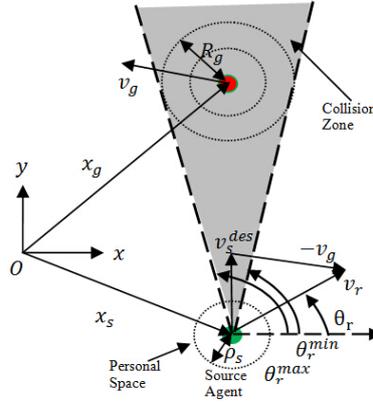


Fig. 6. Relative frame.

The collision zone is defined as a region of space where the source agent should prevent collision with the target agent, i.e., collision is predicted in future if

$$\theta_r^{min} \leq \theta_r \leq \theta_r^{max} \quad (6)$$

and if the two agents do not change their speed and orientation.

When a collision has been predicted, we then compute the time to collision (ttc); if ttc is less than a certain anticipation time t , the target agent is inserted into a set of agents that are on the collision course with the source agent. In real-life, an individual tries to avoid a limited number of other pedestrians, usually those that are on the collision course with him in the coming short time. Similarly, the source agent tries to evade N agents with which will collide first. In our implementation, N is less than 4.

4.3 Collision Avoidance

The least-effort principle originates from the field of psychology and states that given different possibilities of actions, people select the one that requires the

least effort [25]. Based on least-effort theory, many systems for crowd simulation have been proposed [26], [27]. However, all these approaches aim to control the macroscopic (global) behavior of virtual humans, whereas our focus is on the local interactions of individuals. Based on the least-effort principle, we therefore hypothesize that an individual, upon interacting with other individual, tries to resolve potential collisions immediately by slightly adapting his motion. The individual will adjust his trajectory in advance, trying to reduce the interactions with the other walker. We describe our local avoidance algorithm below.

We first retrieve a set of candidate relative orientation O_r , such that the orientation of relative velocity can be selected to resolve the collision with the agents who are on the collision course. According to condition (6), the collision can be avoided if the source agent selects a new relative velocity \mathbf{v}_r^{new} , that satisfies the condition

$$\neg(\theta_r^{min} \leq \theta_r^{new} \leq \theta_r^{max}) \quad (7)$$

To avoid unrealistic orientation deviate, we bound the max angle deviation θ_i^{max} to $\frac{\pi}{2}$. We can compute O_r by combining condition (7) and θ_i^{max} .

We then retrieve the set of candidate relative speed U_r . When O_r is determined, the max relative speed $u_r^{max} = \|v_i^{des}\| + \|v_j\|$ and the min relative speed $u_r^{min} = \left| \|v_i^{des}\| - \|v_j\| \right|$.

Having retrieving O_r and U_r , we select an optimal pair $\mathbf{P}=(u_r, \theta_r)$, where $u_r \in U_r \wedge \theta_r \in O_r$, so that the expenditure energy for the source agent is minimum. See Figure 7(a) for an illustration.

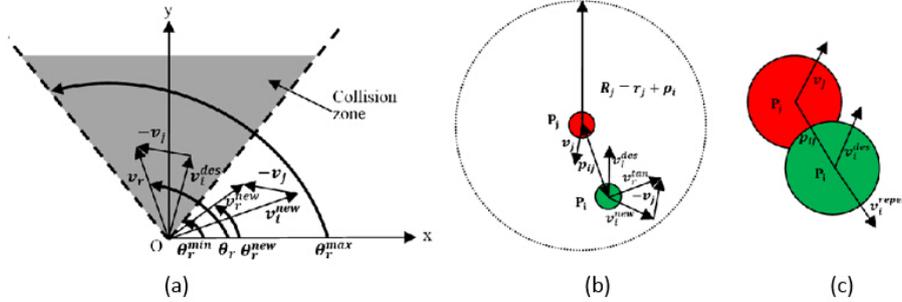


Fig. 7. (a) Selection of an optimal relative velocity for the source agent. (b) (c) Two cases for imminent collision: case (b) One agent enters into the personal space of other but they are not overlapping yet. case (c) Two agents have overlapped

$$\mathbf{v}_i^{new} = \mathbf{v}_r^{new} + \mathbf{v}_j \quad (8)$$

$$\Delta u_i = \left| \|\mathbf{v}_i^{new}\| - \|\mathbf{v}_i^{des}\| \right| \quad (9)$$

$$\Delta \theta_i = \arccos \frac{\mathbf{v}_i^{new} \cdot \mathbf{v}_i^{des}}{\|\mathbf{v}_i^{new}\| * \|\mathbf{v}_i^{des}\|} \quad (10)$$

In the above equations, Δu_i is the value for speed changed, and Δ_i is the angle deviation of the source agent. The cost function is

$$f(u_r, \theta_r) = \alpha * \frac{\Delta u_i}{u^{max}} + \beta * \frac{\Delta \theta_i}{\theta^{max}} \quad (11)$$

where $u^{max}=1.5\text{m/s}$ is the maximal value for speed changed, and $\theta^{max}=\frac{\pi}{2}$ is the maximum angle deviation. The constants α and β define the weights of specific cost terms and can vary among the agents to simulate a wide variety of avoidance behavior.

Computing the minimum value of Equation (11) is time-consuming. Thus, we restrict the domain O_r to a discrete set of orientation samples (the default size of the discretization step is set to 0.01π). Similarly, we discretize the domain U_r into a set of adjacent speed samples (the default distance between adjacent samples is set to 0.05). Assuming that the discretized set of O_r is \mathbf{O}_r and that of U_r is \mathbf{U}_r , then the set of admissible relative velocity is

$$FAV_r = \{u_r \theta_r \mid u_r \in \mathbf{U}_r \wedge \theta_r \in \mathbf{O}_r\} \quad (12)$$

The discretized cost function is

$$v_r^{new} = \underset{\mathbf{V}^{cand} \in FAV_r}{\operatorname{argmin}} \left\{ \alpha * \frac{\left| \|v^{cand} + v_j\| - \|v_i^{des}\| \right|}{u^{max}} + \beta * \arccos \frac{(v^{cand} + v_j) \cdot v_i^{des}}{\|v^{cand} + v_j\| * \|v_i^{des}\|} \right\} \quad (13)$$

Having retrieving v_r^{new} , the optimal new velocity for the source agent is easy to compute. We then update the source agent position into

$$\mathbf{x}_i^{new} = \mathbf{x}_i + \mathbf{v}_i^{new} * \Delta t \quad (14)$$

4.4 Resolve Imminent Collision

We divide imminent collision into two cases (Figure 7(b)(c)). In case (b), we introduce the concept of relative tangential velocity, which is equivalent to applying a tangential force to separate the two agents. In case (c), we introduce the concept of repel velocity, which is equivalent to applying a repulsive force to separate the two agents immediately.

4.5 Avoiding Static Obstacles

An agent A_i also needs to avoid colliding with the static obstacles of the environment. In our simulations, such obstacles are modeled as axis aligned boxes. Collisions are resolved by following an approach similar to the one described above.

We first retrieve the nearest obstacles of the agent A_i that are inside the visual field of the agent. We then compute the maximum and minimum orientations among the vectors lined from the current position of A_i 's to each vertex of the convex polygon obstacle. The maximum and minimum orientations are θ_r^{max} and θ_r^{min} , respectively, which have been discussed above. Finally, we use a least-effort criterion to select an optimal velocity for the agent A_i .

5 Experimental Results

We test our approach against a wide range of scenarios. These scenarios range from the simple interactions between pairs of agents to more challenging and large test cases as follows:

- ◆ Squeeze: Two agents have to avoid a head-on collision while walking in an opposite direction (Figure 8(a)).
- ◆ Overtake: An agent moves down a hallway and encounters a slower agent in front (Figure 8(b)).
- ◆ Square: Four agents are placed on the vertex of a square and have to walk toward their diagonal position (Figure 8(c)).
- ◆ Complex environment: Three hundred agents walk through an environment filled with many obstacles and have to evacuate from the exit (Figure 8(d)).

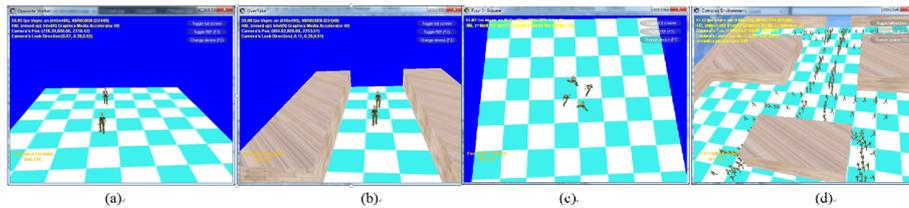


Fig. 8. Scenarios: (a)-(c) interactions in simple environments; (d) three hundred agents evacuate from an obstacle-filled environment.

6 Conclusion

In this paper, we present a novel integrated framework for navigation and interaction behavior. A creative global path planning algorithm and a bilinear interpolation method were used to compute the navigation fields. A least-effort criterion was also employed in the local avoidance to achieve realistic local movements.

7 Acknowledgement

This research was supported by National Natural Science Foundation of China (No. 60473109), NSFC-Guangdong Union Foundation of China (No.U0735001). We would like to thank the anonymous reviewers for their suggestions and comments.

References

1. Treuille A, Cooper S, Popovi? Z. Continuum crowds[C]. ACM Transactions on Graphics (TOG). ACM, 2006, 25(3): 1160-1168.
2. Helbing D, Molnar P. Social force model for pedestrian dynamics[J]. Physical review E, 1995, 51(5): 4282-4286.)
3. Shao W, Terzopoulos D. Autonomous pedestrians[C]. Proceedings of the 2005 ACM SIGGRAPH/Eurographics, 2005: 19-28.
4. Hao Jiang, et al., A Semantic Environment Model for Crowd Simulation in Multilayered Complex Environment. VRST 2009, Kyoto, Japan, November 18 C 20, 2009. pp191-198
5. Kraayenbrink N, Kessing J, Tutenel T, et al. Semantic crowds. Entertainment Computing[J], 2014(5):297-312
6. Paris S, Donikian S, Bonvalet N. Environmental abstraction and path planning techniques for realistic crowd simulation[J]. Computer Animation and Virtual Worlds, 2006, 17(3-4): 325-335.
7. Pettre J, Laumond J P, Thalmann D. A navigation graph for real-time crowd animation on multilayered and uneven terrain[C]. First International Workshop on Crowd Simulation. 2005, 43(44): 194-202.
8. Bandi S, Thalmann D. Space discretization for efficient human navigation[C]. Computer Graphics Forum. Blackwell Publishers Ltd, 1998, 17(3): 195-206.
9. Bayazit O B, Lien J M, Amato N M. Better Group Behaviors in Complex Environments using Global Roadmaps[J]. Artificial Life 8, 2003: 362-371.
10. Sud A, Andersen E, Curtis S, et al. Real-time path planning in dynamic virtual environments using multiagent navigation graphs[J]. Visualization and Computer Graphics, IEEE Transactions on, 2008, 14(3): 526-538.
11. Philippsen R, Siegwart R. An interpolated dynamic navigation function[C]. Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005: 3782-3789.
12. Dapper F, Prestes E, Idiart M A P, et al. Simulating pedestrian behavior with potential fields[M]. Advances in Computer Graphics. Springer Berlin Heidelberg, 2006: 324-335.
13. Lindemann S R, LaValle S M. Simple and efficient algorithms for computing smooth, collision-free feedback laws over given cell decompositions[J]. The International Journal of Robotics Research, 2009, 28(5): 600-621.
14. Park M J. Guiding flows for controlling crowds[J]. The Visual Computer, 2010, 26(11): 1383-1391.
15. Patil S, Van Den Berg J, Curtis S, et al. Directing crowd simulations using navigation fields[J]. Visualization and Computer Graphics, IEEE Transactions on, 2011, 17(2): 244-254.
16. Karamouzas I, Heil P, van Beek P, et al. A predictive collision avoidance model for pedestrian simulation[M]. Motion in Games. Springer Berlin Heidelberg, 2009: 41-52.
17. Van den Berg J, Lin M, Manocha D. Reciprocal velocity obstacles for real-time multi-agent navigation[C]. Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008: 1928-1935.
18. Ondřej J, Pettr J, Olivier A H, et al. A synthetic-vision based steering approach for crowd simulation[J]. ACM Transactions on Graphics (TOG), 2010, 29(4): 123-131.
19. Paris S, Pettr J, Donikian S. Pedestrian reactive navigation for crowd simulation: a predictive approach[C]. Computer Graphics Forum. Blackwell Publishing Ltd, 2007, 26(3): 665-674.

20. Karamouzas I, Overmars M. Simulating human collision avoidance using a velocity-based approach[C]. Workshop in Virtual Reality Interactions and Physical Simulation. The Eurographics Association, 2010: 125-134.
21. Karamouzas I, Overmars M. Simulating and evaluating the local behavior of small pedestrian groups[J]. Visualization and Computer Graphics, IEEE Transactions on, 2012, 18(3): 394-406.
22. Wee Lit Koh and SuiPing Zhou. Modeling and Simulation of Pedestrian Behavior in Crowded Places. ACM Transactions on Modeling and Computer Simulation, 2011, 21(3): 20-42.
23. Zhang L, LaValle S M, Manocha D. Global vector field computation for feedback motion planning[C]. Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009: 477-482.
24. Fang Z, Lo S M, Lu J A. On the relationship between crowd density and movement velocity[J]. Fire Safety Journal, 2003, 38(3): 271-283.
25. George K. Zipf. Human behavior and the principle of least effort[J]. Addison-Wesley, 1949. 3.
26. Safonova A, Hodgins J K. Construction and optimal search of interpolated motion graphs[C]. ACM Transactions on Graphics (TOG). ACM, 2007, 26(3): 106-116.
27. Guy S J, Curtis S, Lin M C, et al. Least-effort trajectories lead to emergent crowd behaviors[J]. Physical Review E, 2012, 85(1): 016110.