Flock Based Real Time Collision Avoidance Crowd Simulation

Ziqi Wu[1, a *], Xubo Yang[2,b]

[1]Shanghai Jiao Tong University, China

[2]Shanghai Jiao Tong University, China

[a]jiaodapeter@sjtu.edu.cn, [b] yangxubo@sjtu.edu.cn

Keywords: Crowd Simulation, Flock, Collision Avoidance, Continuum

Abstract. Real time simulation is a challenging subject in crowd simulation. Many approaches have been proposed to simulate the crowd behavior. However, few of them[1] focused on solving collision avoidance problem. In this paper, we present a novel real-time collision avoidance method for crowd simulation, which combines advantages of both continuum-based methods and agent-based methods. Also, by integrating the theory of flock, we improve the result of our simulation to be more realistic.

Introduction

With the development of computer graphics, real-time crowd simulation has been applied in various domains such as film effects, virtual reality and city planning. A variety of approaches have been proposed to describe the realistic crowd scenes. For example, the OCEAN model[2] is used to set up the mental model and some other approaches like dynamically evolving flow allows kinetic guidance of agents based on different mathematical equations. Also, some researchers use data-driven method[3] to reconstruct the crowd simulation from the collected data.

However, full-featured crowd simulation is still a challenging task arises from the fact that human behaviors are in highly complexity in nature world[4]. Human behaviors often involve interaction with external environment, obstacles and other people. There are at least three aspects to be considered about, which are route navigation, collision avoidance and individual personality preserving respectively.

In this paper, we present a novel real-time method for crowd simulation. By introducing the theory of flock and swarm[5], our method can predict the crowd behavior in order to improve the simulation of collision avoidance. Meanwhile, our algorithm preserves both heterogeneous personalities and group-based characteristics to make the simulation more realistic.

The remainder of this paper is organized as follows. In section 2, we review the state of art about crowd simulation. In Section 3, we provide a detailed description of our algorithm and how we improve the simulation loop. In section 4, we present our performance analysis, which includes the overall experiment scenes and recorded data. Also, we will give a brief comparison with other existing methods. Finally, section 5 concludes this paper.

Related Works

Crowd simulation has a long history in computer graphics. Many approaches have been proposed and can be divided into two categories, continuum-based methods and agent-based methods.

Continuum-based methods[6][7][8][9] define the model like dynamic fluids. It represents agents as a continuous density field and the field is driven by some external sources. All external sources exert to the crowd and update their properties as a whole to move towards destination. However, this kind of methods pays little attention to individual behaviors, in other words, lack of personalities.

On the contrary, agent-based methods[10][11][12] involves more individual behaviors. They treat individuals as heterogeneous agents whose interactions will result in the feedback to the entire crowd. The agent-based methods allow us to exhibit a variety of interesting phenomenon. However, this kind of methods also introduces two defects. Firstly, the more agents involve in, the more complex the simulation will be. Since every agent has to interact with both static and kinetic obstacles including reciprocal influences[13], it is really sophisticated to model all necessary elements. While some models are presented to define various related factors, little attention is paid to the level of group-based characteristics. For example, relationship like family members does have influence on crowd simulation.

Our model is a hybrid model of continuum crowd and agent-based methods. We compute the potential function to drive the crowd towards their destination in the optimal way and compute individual behaviors with respect to the external environment and other agents. By combining these two steps, we preserve both a smooth path towards crowds' destination and heterogeneous personalities. Notably, we introduce the theory of flock and swarm to compute more characteristics based on the level of similar groups in order to produce more realistic and believable phenomenon.

Overview

In this section, we give a brief overview on how our simulation model and its associated algorithm.

Model of crowd simulation We define the model of crowd simulation from three aspects. (1) Each individual has a specific destination in a two-dimensional scene and this destination point can be reached in a limited time. (2) We define a discomfort field into our algorithm. Area with less density and less obstacles will be assigned a low discomfort value. Individuals prefer to choose path with less discomfort. To produce a more realistic result, we provide a random function to make some of individuals to violate this rule. (3) Individual with similar characteristics prefers to move as a group, for example, members in a family tend to move closer to each other.

Algorithm Our algorithm can be divided into two major steps. The first step is to compute the path towards destination by using continuum methods. We convert the crowd into density field and compute the potential field with respect to both static and kinetic obstacles. By applying this method, we can get an optimal trajectory towards destination with least unit distance cost. Moreover, we can avoid a sharp change of trajectory and the result will be empirical-believable.

On the basis of agent-based simulation algorithm, we integrate the theory of flock into the simulation to improve details. We assign those agents having similar characteristics, e.g. destination, initial position, with the same group id. In each group, those who move faster are regarded as leaders and they perform a dragging forward force to remain agents. On the contrary, the rest of agents perform a pulling back force to leaders to keep the cluster moving as a cluster. Meanwhile, there are two other forces involved, which are the **attractive force** and **repulsive force**. The attractive force is to guarantee the distance among each other agent won't be very dispersed while the repulsive force is to ensure the distance is not too close among each other in case of overlap and collision.

Implementation

Density Conversion The density conversion computation is inspired by Continuum Crowd[6][15]. We convert crowds into density by using a simple linear weighted function to calculate their density contribution for its four neighbor grids based on the distance between their current position and the center point of the grid as shown in figure 2.

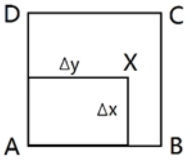

Fig. 2 Density Contribution of an agent

We define the per-agent density ρ_i so that the density field can be calculated from the Eq.1 below.

$$\rho_a = \sum_i \rho_i * (1 - \Delta x) * (1 - \Delta y) \qquad \rho_b = \sum_i \rho_i * \Delta x * (1 - \Delta y) \qquad (1)$$

$$\rho_c = \sum_i \rho_i * (1 - \Delta x) * \Delta y \qquad \rho_d = \sum_i \rho_i * \Delta x * \Delta y$$

By summing up all density contributed by an agent, we can easily construct a density field.

Simultaneously, we calculate the average velocity field which is inversely proportional to the density field. The average velocity works as a lower bound of speed for crowd movement. If the density in a certain grid exceeds the predefined maximum value, all individuals in the grid will move in this average speed. The Eq. 2 shows how we get the average velocity divide the sum of density by the sum of individual velocity.

$$V_{avg} = \frac{\omega * \sum v_i}{\sum \rho_i} \tag{2}$$

Discomfort Field and Unit Cost The computation of our discomfort field is very similar to the method mentioned in Continuum Crowd. However, judging from the observation, sometimes there are two or more reachable paths from individual's position towards their destination at a time. Some of them are congested with shorter distance while some of them are longer but spacious. Compared with previous approached presented in [][], the crowd prefer to choose the spacious path even they have longer distance according to our algorithm, which is a big improvement and are more in line with the reality.

Considering from the above two aspects, we construct a discomfort field which is proportional to the density field. The equations are described as follows.

Potential Field Since we have got the unit cost field, we can construct the potential field in order to drive the crowd from the high potential value to the place with low potential value. In our method, we define the destination point with potential value zero and try to find the optimal path. Compared with the implementation, we utilize the priority queue to back trace from the destination point to construct the path.

Notably, we don't use the potential field to drive the movement of the crowd directly. Instead, we convert the potential field into a steering force[14] by Eq. 3,

$$V_{steer} = \frac{V_{prefer} - V_i}{V_{max}} * F_{max} \tag{3}$$

where V_{prefer} is the product of the direction that agents will go and V_{max}.

Attractive Force, Repulsive Force and Alignment Force Members within the same group will try to get closer to each other when they are far away from each other. In order to satisfy the requirement of real-time, we cannot calculate the reciprocal attractive forces among all members directly. Instead, we firstly create a virtual core member for each group. The position of the virtual core member is the average result of the sum of the position of all individual. The second step is to add a force for each individual to move towards the virtual core member. Notably, in order to make our result more realistic, we define a biggest impact area of attractive force. If distance from agents to the core member's position is out of area, the attractive force will have no impact on them.

$$F_{att} = \frac{\alpha \sum_{l(i,j)<R} m_i(\vec{l_i} - \vec{l_x})}{\sum_{l(x,i)<R} m_i} \tag{4}$$

Repulsive is the force that causes an agent to steer away from all of its neighbors. To calculate the repulsive force, We enforce a pair-wise minimum distance between agents. We iterate all pairs within the distance and symmetrically pushing them apart from each other. The procedure does not guarantee that the minimum distance is always enforces, e.g., a room full of people. But for the vast majority of time, our result is good enough for crowd simulation.

$$F_{Coh} = \frac{\beta \sum_{l(x,i)<r} m_i(\vec{l_0} - \vec{l_i} + \vec{l_x})}{\sum_{l(x,i)<r} m_i} \qquad (5)$$

Alignment is a behavior that causes a particular agent to line up with agents close by. We iterate through all of the agents and find the ones within the neighbor radius - that is, those close enough to be considered neighbors of the specified agent. If an agent is found within the radius, we add its velocity to get the sum of all velocity and then average it by the total number of neighbors. The result will be the aligned direction for influenced agents.

Experiments and Analysis

We have run a diverse set of crowd simulation with our algorithm and found that it does produce satisfying results in real time. Our implementation is based on C++ and we make use of Box2D physics engine to simulate the collision effect. All simulations run on a 2.40GHZ Intel Core i5 processor with a Nvidia GeForce GT540M graphics card and the RAM is 8 Gigabytes. To better illustrate our method, we construct several scenes including tunnels and opposite walking to test the result of our algorithm. Meanwhile, we evaluate our method by comparing the performance and visual effect with other methods.

Here we give a brief description of two typical scenes we constructed. Fig. 3 shows a simulation of two groups met in a narrow tunnel. Each group of people walked as a cluster and kept an appropriate distance with each other. From the result we can see that both groups of people anticipated the potential collision in a distance and changed their path in advance.

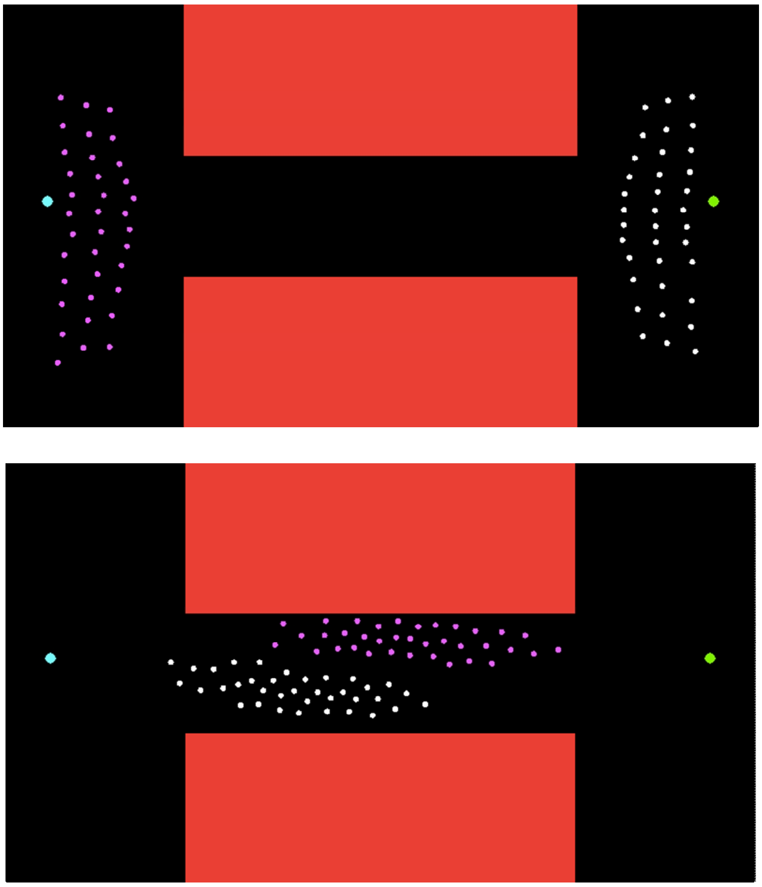

Fig. 3 Simulation of two groups of people met in the tunnel

Fig. 4 shows two groups of people walks towards each other and their distance was getting closer. Majority of people detoured and changed their path to avoid collision while a few still kept the same way towards their destination. This was because we provide the simulation with a random probability for violating the rules and this made the result more realistic.

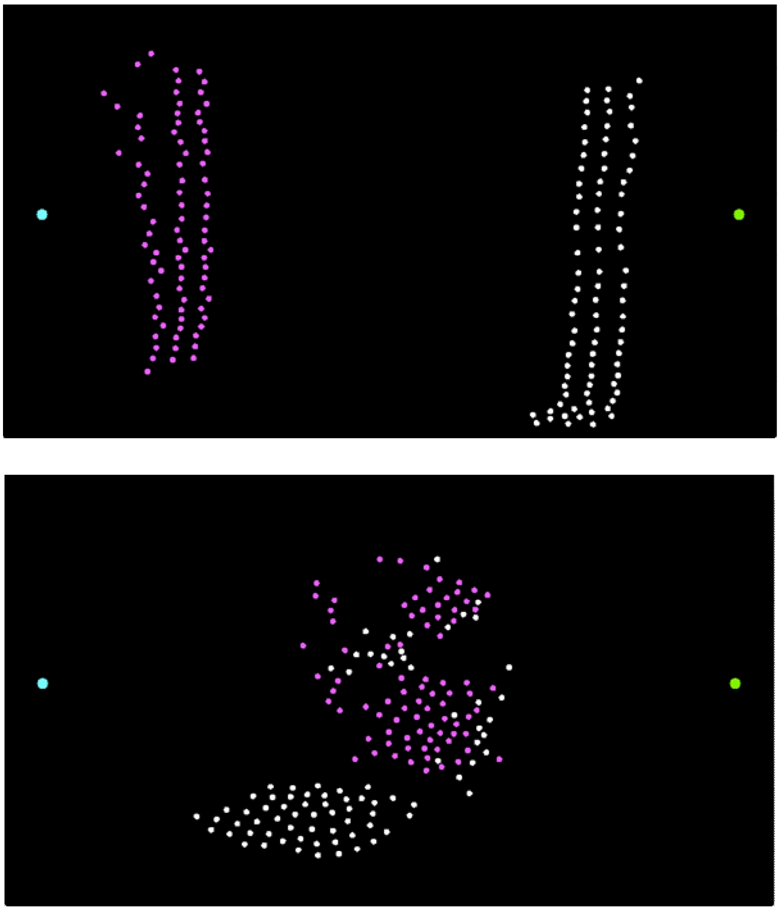

Fig. 4 Simulation of two groups of people moving towards each other

We also list some collected data from our experiments to illustrate the efficiency of our algorithm.

Table.1 data collected from scene of tunnel

Number of agents (1000)	Frame per second
3.5	58.93
6	37.96
8	32.74
12	20.58

Table.2 data collected from scene of moving towards each other

Number of agents (1000)	Frame per second
11.7	81.8
22.6	43.6
29	30.6
36.4	19.4

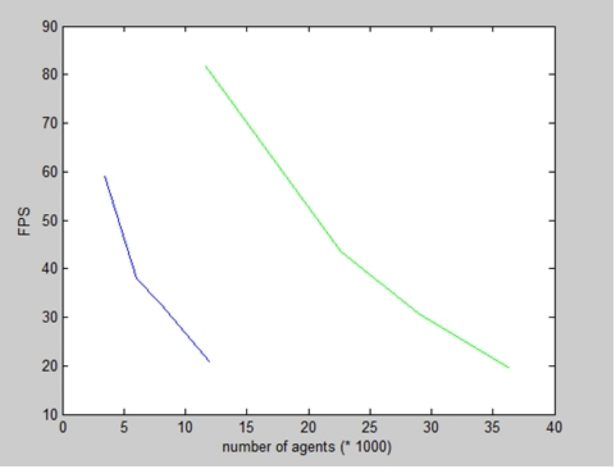

Fig. 5 Frame per second with different number of agents
(a) blue line is the tunnel scene (b) green line is moving toward scene

We also run a comparison with agent-based model[16]. We setup a complex scenario that contains lots of obstacles. Several people in yellow start walking from the one side of road and want to cross the road. Also, there are lots of people in blue wandering on the road work. We can see that those people do not have the preference to avoid the high density area. Instead, they just plan to cross the road with the shortest path, which doesn't match with the reality. The result is shown in Fig. 6.

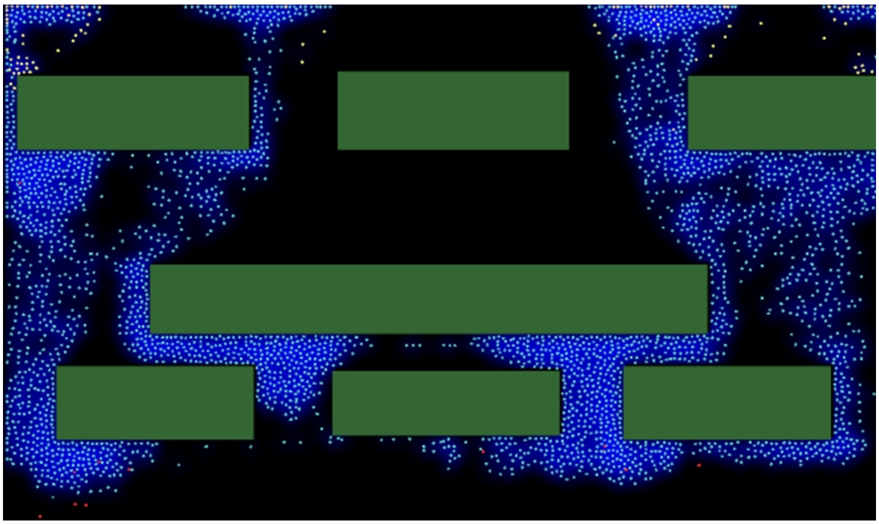

Fig. 6 Agent-based model simulation

Conclusion

In this paper, a novel method of collision avoidance crowd simulation is presented. Several enhancements are integrated into our method to produce more realistic crowd behaviors of collision avoidance. Our algorithm makes use of the continuum-based method to compute the procedure of the global navigation. Meanwhile, we also take advantage of the theory of flock to add more group characteristics into the simulation and we manage to preserve individual personalities on the basis of agent-based method.

By setting up a variety of experiment, our approach has been tested under different challenging scenarios. Also, by comparing our work with other existing algorithm, we have proved that our method is feasible and plausible, which shows great potential to fit into graphic application such as virtual reality and city planning.

However, our algorithm still has some limits. In real life, people tend to have a preference for detour directions. Under most circumstances, people prefer to choose the right side to walk. At present, the path evaluation function of our algorithm does not take this phenomenon into consideration and in the follow-up we will try to optimize our path evaluation function and improve the result.

Acknowledgement

This work is supported and funded by Digital Art Lab of Shanghai Jiao Tong University. We would also like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots[J]. The international journal of robotics research, 1986, 5(1): 90-98.

[2] Durupinar F, Allbeck J, Pelechano N, et al. Creating crowd variation with the ocean personality model[C]//Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3. International Foundation for Autonomous Agents and Multiagent Systems, 2008: 1217-1220.

[3] Lee K H, Choi M G, Hong Q, et al. Group behavior from video: a data-driven approach to crowd simulation[C]//Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2007: 109-118.

[4] Golas A, Narain R, Curtis S, et al. Hybrid long-range collision avoidance for crowd simulation[J]. Visualization and Computer Graphics, IEEE Transactions on, 2014, 20(7): 1022-1034.

[5] Berrondo M, Sandoval M. Defining emergence: Learning from flock behavior[J]. Complexity, 2015.

[6] Treuille A, Cooper S, Popović Z. Continuum crowds[J]. ACM Transactions on Graphics (TOG), 2006, 25(3): 1160-1168.

[7] Narain R, Golas A, Curtis S, et al. Aggregate dynamics for dense crowd simulation[C]//ACM Transactions on Graphics (TOG). ACM, 2009, 28(5): 122.

[8] Hughes R L. A continuum theory for the flow of pedestrians[J]. Transportation Research Part B: Methodological, 2002, 36(6): 507-535.

[9] Jiang H, Xu W, Mao T, et al. Continuum crowd simulation in complex environments[J]. Computers & Graphics, 2010, 34(5): 537-544.

[10] Luo L, Zhou S, Cai W, et al. Agent-based human behavior modeling for crowd simulation[J]. Computer Animation and Virtual Worlds, 2008, 19(3-4): 271-281.

[11] Pan X, Han C S, Dauber K, et al. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations[J]. Ai & Society, 2007, 22(2): 113-132.

[12] Murakami Y, Minami K, Kawasoe T, et al. Multi-agent simulation for crisis management[C]//Knowledge Media Networking, 2002. Proceedings. IEEE Workshop on. IEEE, 2002: 135-139.

[13] Van Den Berg J, Guy S J, Lin M, et al. Reciprocal n-body collision avoidance[M]//Robotics research. Springer Berlin Heidelberg, 2011: 3-19.

[14] Ondřej J, Pettré J, Olivier A H, et al. A synthetic-vision based steering approach for crowd simulation[C]//ACM Transactions on Graphics (TOG). ACM, 2010, 29(4): 123.

[15] Pelechano N, Allbeck J M, Badler N I. Controlling individual agents in high-density crowd simulation[C]//Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2007: 99-108.

[16] Braun A, Musse S R, de Oliveira L P L, et al. Modeling individual behaviors in crowd simulation[C]//Computer Animation and Social Agents, 2003. 16th International Conference on. IEEE, 2003: 143-148.