

Crowd simulation based on constrained and controlled group formation

Peng Zhang · Hong Liu · Yan-hui Ding

Published online: 29 November 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract Freestyle formations appear widely in animation of groups. Most existing algorithms for generating special formations focus on the visualization performances of target formations, while social dynamics factors in the process of crowd motion are ignored. Thus, disregarding those factors will decrease the bionic features and fidelity of the crowd motion. According to this problem, a method based on bionic intelligence algorithm and self-adaptive evaluation to generate special formations is proposed in this paper. Simulation effect with good fluency and lively interaction is generated by means of user interaction, data analysis and crowd motion. In this method, 3D reconstruction is used to repaint characters, graphics or patterns in the 3D modeling system to build the basic virtual scene. Then, station points are generated through interlacing cross sampling. Based on the concentric circles model of fitness, each individual, self-adaptively, chooses a target station point which matches it aptly. Finally, the Artificial Bee Colony algorithm is used for path planning to generate the optimum route to the destination without collision. Visual simulation experiments are also made on the platforms of ACIS/HOOPS and Maya. The results show that this method can generate the optimum target formation with natural motion features and in accordance with users' input. This method is also insensitive to the scale of crowd, exhibiting good performance when the number of individuals is large.

Keywords Crowd motion · Freestyle formations · Artificial bee colony algorithm · 3D simulation

1 Introduction

With the developments of computer graphics and virtual reality, the visual effect of computer simulation is enhanced significantly. Recently, simulation and visualization of the emergence of behaviors among individuals in a crowd as well as between the group and the environment have become a new research focus in the field of crowd simulation.

In natural crowd motion, the group is generally required to abide by special behavioral constraints to get in a particular formation, such as large-scale joint performance of group calisthenics, real-time maneuvers of campaign, etc. Manual analysis and design on the simulation of freestyle formations should appoint a target position for each individual, which results in low efficiency, poor accuracy and weak global controllability. The applications of computer simulation systems provide fresh ideas on how to generate group formations. Both accuracy and efficiency can be improved using a system's automatic analysis and computation. Therefore, this paper proposes a method of crowd simulation based on constrained and controlled group formation. A simulation system is also built to exhibit the processes of generating particular formations by taking advantage of graphic loading, pixel analysis, 3D reconstruction, uniform sampling and intelligent motion. This system can generate the optimum group formation automatically, in accordance with the original characters, graphics or patterns based on users' parametric settings. Our major contributions in this paper are:

1. To propose a method for generating special formations based on swarm intelligence (SI) and self-adaptive

P. Zhang (✉) · H. Liu · Y. Ding
School of Information Science and Engineering,
Shandong Normal University, Jinan 250014, China
e-mail: zhpll@126.com

P. Zhang · H. Liu · Y. Ding
Shandong Provincial Key Laboratory for Distributed
Computer Software Novel Technology, Jinan 250358, China

evaluation, describing the principle and process of this method in detail.

2. To apply the concentric circles model in urban planning for implementing discretization of the characters, graphics or patterns according to density, and search for the global optimum target point for each individual.
3. To generate the optimum solution for organizing the particular formation, and build a 3D simulation experiment system to make microcosmic simulations on the processes of generating group formations based on typical cases.

The rest of the paper is organized as follows. In Sect. 2, we briefly introduce the main research achievements of crowd simulation on group formations. In Sect. 3, we describe the main processes necessary for generating freestyle formations, such as data preparation, 3D reconstruction, self-adaptive matching, and path planning. In Sects. 4 and 5, we show experiments for generating special formations using, respectively, ACIS/HOOPS and Maya. Conclusions and future research focuses are described in Sect. 6.

2 Related work

Crowd simulation originates from the simulation research on crowd motion and collision avoidance in emergency. The mature thoughts of crowd simulation at present include particle systems and artificial life systems based on multi-agent [1–5]. In Refs. [6] and [7], the researchers proposed a simulation model of crowd motion based on particle swarm optimization (PSO) algorithm, in which the best path from the initial position to the destination was computed by means of the PSO algorithm. In Ref. [8], the researchers applied ant colony optimization (ACO) algorithm to evacuation simulation in social emergency. They also built a computational framework of crowd simulation based on the ACO algorithm. In Ref. [9], the authors presented the analysis of a systematic research on crowd evacuation behaviors in panic situations, and established a crowd simulation model for panic conditions. In Refs. [10] and [11], the authors designed a crowd evacuation simulation model based on cellular automata (CA). The simulation's performance was evaluated with respect to autonomy, cooperation and scalability in a confined space. It provided fresh ideas for crowd simulation and research on emergency management.

The constraint and control of formations are commonplace in crowd simulation, computer animation and collaborative robotics, and derive continuous attention of researchers in recent years [12–14]. In Refs. [15] and [16], the authors proposed an interactive dynamic programming framework to simulate freestyle formations, providing a series of interfaces to input free-form data and generating a scheme to update position for each individual automatically after evaluating the

distribution of target points. The research adopted two-level control mechanism to generate crowd path, but the bionics features and the fidelity of path planning were limited. In Ref. [17], the researchers presented a method for editing and controlling crowd motion in team formations. Crowd animation with fluency and high fidelity was synthesized after the processes of matching roles, cross optimization and updating positions. This paper proposed a method of generating special formations, while details of crowd simulation such as path planning and collision avoidance were not described in detail. Ref. [18] combined CA with ACO algorithm to plan path for robots group and built the lifelike controllable scene of crowd motion without collision. This paper only researched on the formation control in the process of robots group motion, while generating final special formations was not studied.

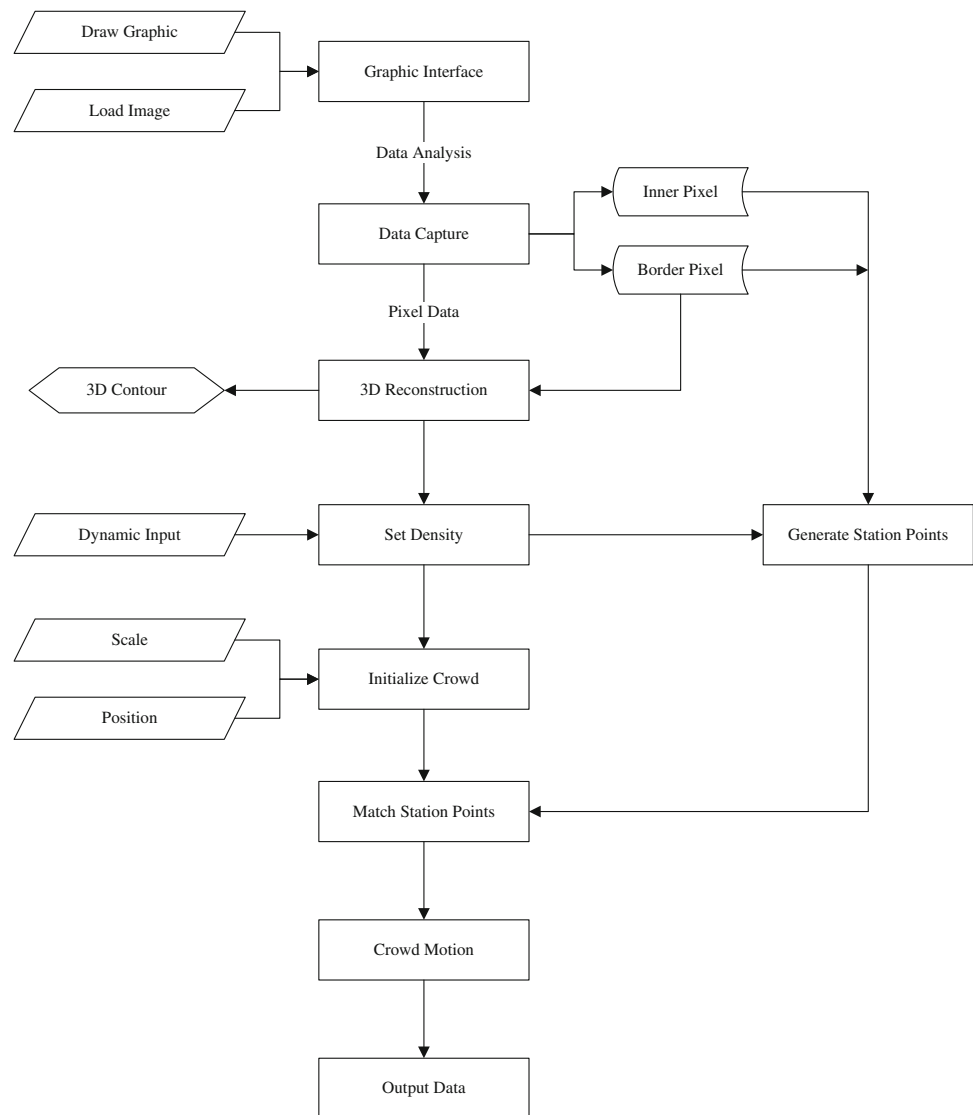
Compared with the related works above, our main innovations are listed below.

- (1) To apply swarm intelligence in generating special formations. The inspiration of SI derives from crowd behaviors in nature. It can improve crowd simulation's intelligence and speed the typical algorithms in the SI.
- (2) To propose a density-centered method for generating formations. In Refs. [15] and [16], the target formations are generated based on the crowd's size input by users in advance. But when designing formations in our real life, the designers always do not realize the accurate number of individuals. Therefore, our method generates formations based on the density input by designers. The number of individuals necessary for the target formation is computed by the system according to the target pattern and density. This thought is not only in accordance with the design process, but also an effective approach to reduce the designer's workload.

Therefore, our paper describes the core algorithm and main process of generating group formations in detail. According to the special characters, graphics, patterns and density input by users, the concentric circles model and bionic intelligence algorithm are adopted to make self-adaptive evaluation and path planning, respectively. It can improve intelligence and fidelity in crowd motion.

3 Flow of generating freestyle formations

In this section, we propose a method for crowd simulation based on constrained and controlled group formation. The 3D simulation system using this method can automatically generate the optimum path from the initial position to the corresponding destination for each individual, with fluency and without collision. The flow of this method is described in Fig. 1.

Fig. 1 The flow chart of generating freestyle formations

In Fig. 1, users can draw or import an image with characters, graphics or patterns through the graphic interface provided by this system. Then the system automatically analyzes the characters, graphics or patterns, making 3D reconstruction to generate the target formation patterns to be organized according to the analysis result. Based on the density input by users, the interlacing cross sampling mechanism is used to generate basic station points covering the original patterns. Finally, the system initializes the crowd according to the result of sampling, matching station points and planning path to generate the optimum routes to the target points without collision. The detailed processes are listed below.

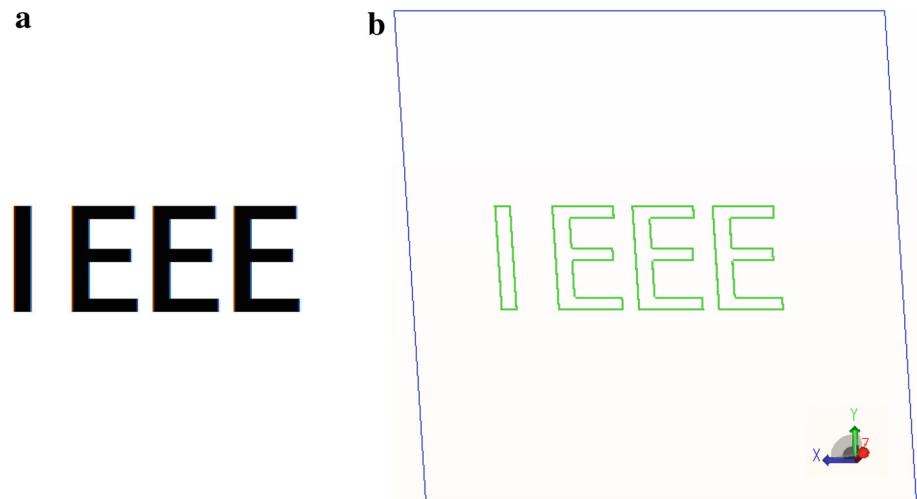
3.1 Data preparation

In the process of data preparation, firstly, users input an image with characters, graphics or patterns through the

interface, and then, the system automatically recognizes and analyzes the image to capture basic data information. This process includes two main sections: users' input and data capture.

According to Fig. 1, the system provides two versions of the "Graphic Interface": Draw Graphic and Load Image. Users can draw basic vector graphics like rectangles and pentagons through the "Draw Graphic" interface. While arbitrary forms of images with characters, graphics or patterns can be imported to this system to be transformed into the binary format through the interface of "Load Image". Because the size of the 3D modeling environment is set to 600×600 px. So the image imported by users is required in size 600×600 px. If the image format is not bitmap, the system will transform it to the standard bitmap format. So, the system is non-sensitive to the image format. But to save processing time, we recommend users to import the bitmap image directly.

Fig. 2 The result of 3D reconstruction. **a** 2D characters, **b** 3D scene



In the “Data Capture” process, the system automatically recognizes and analyzes the pixels of characters, graphics or patterns input by users, extracting the border and inner points, respectively, to store for subsequent use.

3.2 3D reconstruction

Simulation environments are always three-dimensional to improve fidelity of crowd simulation, while the image with characters, graphics or patterns is two-dimensional. Therefore, the system should make 3D reconstruction according to the stored pixel data, as described in Sect. 3.1, to be able to implement transformation from a 2D image to a 3D scene.

In this research, the system automatically reads the stored border data (see Sect. 3.1) and reconstructs the contents in the ACIS/HOOPS modeling engine framework. The result is described in Fig. 2.

3.3 Station points generation

In crowd simulation based on constrained and controlled group formation, the set of station points forms the basic frame of the formation. The positions and distributed character of station points directly determine the final formation and visual effect of the crowd simulation.

To implement global even distribution, this method adopts interlacing cross sampling algorithm based on density. It can generate uniform lattice to cover the original characters, graphics or patterns through processes of dividing grids, crosswise and lengthwise cross sampling. The algorithm flow is listed below.

- Step 1: Input the density of station points.
- Step 2: Initialize grids (set pixel as the fundamental unit and construct cross grids).

Step 3: Sample crosswise (run the algorithm of interlacing sampling along Y axis based on the density of station points).

Step 4: Sample lengthwise (run the algorithm of interlacing sampling along X axis based on the density of station points).

Step 5: Adjust the station points.

The station points on the border are generated through sampling crosswise, while the station points in the inner area are generated through sampling lengthwise. In characters, the distance between borderlines is fairly short. Therefore, only transverse sampling is required for basic characters to prevent the algorithm from generating a large number of station points cluttering together to distort the display of the characters’ formation. The set of vertices is the key factor to determine the basic contour of characters, graphics or patterns, which may be ignored in the process of sampling. So, a necessary adjustment to the station points should be made in Step 5. The adjustment formula is listed below.

$$Pos_i = \begin{cases} I_k, & \text{if } \|Pos_i - I_k\| \leq \frac{1}{2}ave_distance, k \in [0, I_n) \\ Pos_i, & \text{otherwise} \end{cases} \quad (1)$$

where Pos_i is the coordinate of the i th station point, I_k is the coordinate of the k th vertex, $ave_distance$ is the density input by users and I_n is the number of vertices. The result after adjustment is described in Fig. 3.

Figure 3a shows that some vertices are ignored in the original sampling process. Figure 3b shows the sampling result after adjustment. In Fig. 3c, we take one point for example to depict the adjustment according to formula (1). As can be seen from the comparison in Fig. 3, the sampling with adjustment can fill vertices effectively, providing accurate data set for matching station points and intelligent motion.

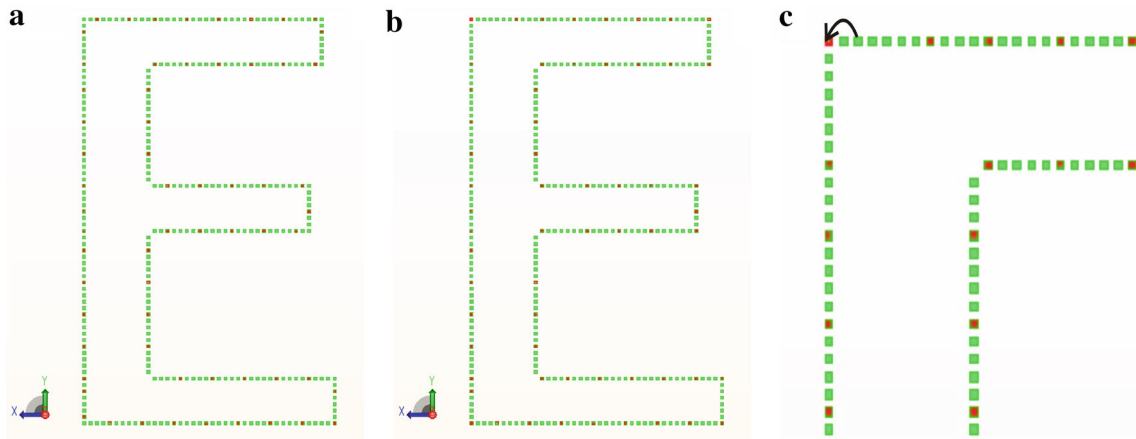


Fig. 3 The comparison of the original sampling and the result after adjustment. **a** The original sampling, **b** the result after adjustment, **c** the adjustment of one point

3.4 Self-adaptive matching

The aim of crowd motion is to generate the optimum path from the initial position to the destination without collision. Before this process, each individual should evaluate the station points and set the optimum one as its destination. In the process of matching station points, there are two types of optimization: individual optimization and global optimization. When using the individual optimization method, each individual will choose the nearest station point as destination. This method is simple to implement and the computational complexity is low. However, a large number of individuals may be jammed and cause a speed decrease in the crowd motion. In the global optimization method, a global optimization algorithm is used to minimize the time overhead of the crowd motion. When choosing station points, each individual evaluates not only the distance but also the accessibility. Therefore, this method can guarantee congestion avoidance by effective in-advance optimization.

In our research, we adopt a global optimization method to match station points. We borrow some urban planning ideas to propose a concentric circles model based on the optimum center. Firstly, the system divides areas automatically on the characters, graphics or patterns according to the crowd density. The result is described in Fig. 4, in which $r_1 = r_2 = ave_distance$. As is shown in Fig. 4, we draw some circles using the image’s center point as the circles’ centers. The radius gradually increases by r_1 from the innermost circle to the outermost one. To generate some sub-areas, we divide the annulus to many sectors using r_2 as the heights along Y -axis.

Each individual chooses a station point as destination based on formula (2):

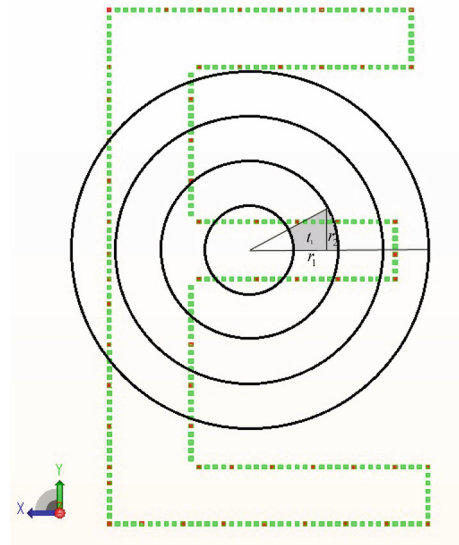
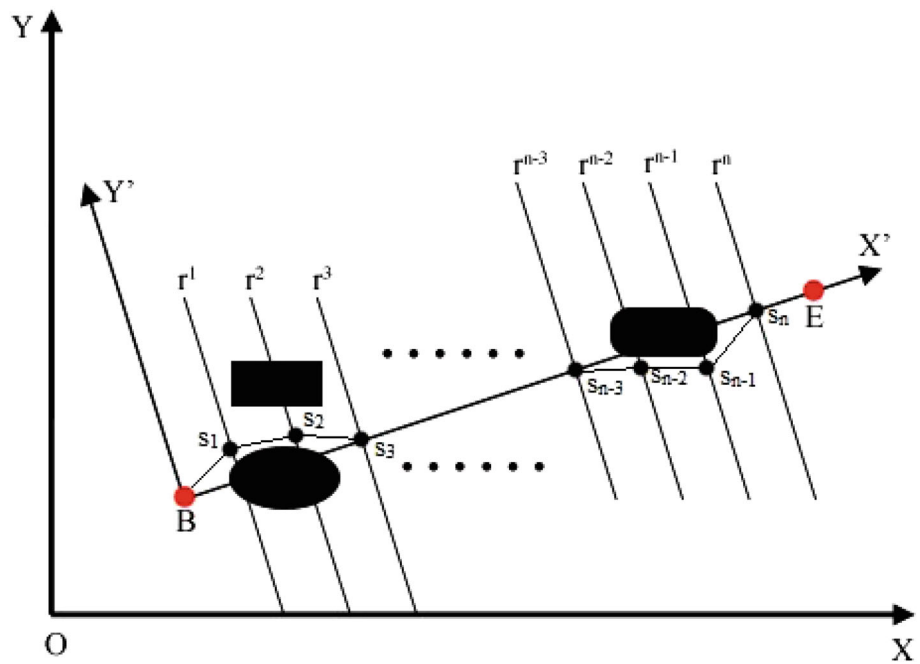


Fig. 4 The concentric circles model

$$\begin{cases} goalpos_{ik} = c_1 \frac{\|Pos_k - center\|}{\sum_{j=1}^{sn} \|Pos_j - center\|} + c_2 \frac{\|x_i - Pos_k\|}{\sum_{j=1}^{sn} \|x_i - Pos_j\|} \\ \quad + c_3 \frac{count[k]}{sn} \\ goalpos_i = \min(goalpos_{ik}) \end{cases} \quad (2)$$

where $goalpos_{ik}$ is the fitness of the k th station point relative to the i th individual, $center$ is the coordinate of the center point, sn is the number of individuals, $count[k]$ is the number of individuals in the ring area t_k that contains the k th station point, $goalpos_i$ is the destination of the i th individual and c_1, c_2, c_3 are weight coefficients ($c_1 + c_2 + c_3 = 1$). The parameter $count[k]$ depends largely on $ave_distance$. As can be seen from formula (2), individuals take factors of distance, accessibility and distributivity synthetically into consideration when choosing station points. Each indi-

Fig. 5 The description of path planning



vidual, self-adaptively, chooses the station point with the global optimum fitness as destination to improve the speed of crowd motion.

3.5 Path planning

When the process of matching station points finishes, the crowd should perform path planning in order to generate the optimum routes with high fluency, good fidelity and without collision [19]. The aim of path planning is to search the optimum position set appearing on the route of an individual's motion. The data set is described as formula (3):

$$R = \{B, s_1, s_2, \dots, s_n, E\} \tag{3}$$

where B is the initial position, E is the destination and s_i is the i th point on the route. The result is described in Fig. 5.

In Fig. 5, rectangles and ellipses represent obstacles. To reduce the computational complexity of path planning, we make transformation on the original coordinate system (XOY). Set the line covering point B and E as X axis and the line perpendicular to X axis as Y axis to generate a new coordinate system ($X'BY'$). The formula to transform coordinate is:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x - x_B \\ y - y_B \end{pmatrix} \tag{4}$$

where (x', y') is the new coordinate, (x, y) is the original coordinate, α is the angle between BX' and OX and (x_B, y_B) is the original coordinate of point B . In Fig. 5, the original solution space is continuous. Therefore, a path should be discretized into $n + 1$ uniform segments. The positions along

a path are limited to the set of lines (r^1, r^2, \dots, r^n) that go through discrete points and are perpendicular to the X' axis. The length of each path can be computed with formula (5).

$$D = \sum_{i=0}^n \sqrt{\left(\frac{|BE|}{n+1}\right)^2 + (s_{i,y'} - s_{i+1,y'})^2} \tag{5}$$

where $s_0 = B, s_{n+1} = E$. Thus, path planing can be formulated as an optimization problem to minimize D .

This method uses the artificial bee colony algorithm [20–22], which was recently proposed in the area of swarm intelligence, to plan a path by taking D as the fitness function. Compared with the other traditional path-planning algorithms such as the A^* algorithm, artificial potential field method and topological method, the ABC algorithm has a simple model and less parameters, which result in low computational complexity and rapid convergence in path planning. Moreover, the ABC algorithm is non-sensitive to the shapes of obstacles. It can be easily transferred from a simple environment to another complex environment without any adjustment. The algorithm flow is listed below [23].

- Step 1: Initialize the scale of bee colony, the maximum iteration number, the solution space, etc.
- Step 2: Calculate fitness for each solution according to formula (5).
- Step 3: Sort the solutions in descending order of fitness and select the top 50% to build an employed matrix.
- Step 4: Compute the selected probability of each employed forager.
- Step 5: The remainders switch roles to scouts or onlookers.

Step 6: The candidate solutions update positions according to their corresponding roles.

Step 7: Go back to Step 2 if the iterative condition is satisfied, otherwise exit the iteration.

While initializing parameters in Step 1, set d as the dimensionality of the solution space, $M CN$ as the maximum iteration number and $limit$ as the threshold of fitness. Generate m particles according to formula (6):

$$G = \{S_1, S_2, \dots, S_m\}, \quad S_i = \{s_{i,0}, s_{i,1}, \dots, s_{i,n+1}\} \quad (6)$$

where $i \in \{1, 2, 3, \dots, m\}$, $s_{i,j}$, are generated randomly on r^j and out of obstacle range.

In Step 4, compute the selected probability of each employed forager according to formula (7) [24]:

$$P_{S_i} = \frac{fit_{S_i}}{\sum_{i=1}^{en} fit_{S_i}} \quad (7)$$

where P_{S_i} is the selected probability of the solution with number S_i , fit_{S_i} , is the fitness of the solution with number S_i , and en is the number of employed foragers.

In Step 6, update positions of the employed foragers according to formula (8):

$$x_{ij} = x_{ij} + rand(-1, 1) \times (x_{kj} - x_{ij}) \quad (8)$$

$k \in \{1, 2, 3, \dots, en\}$, where k is randomly generated and $k \neq i$. Position updating is based on a greedy strategy. Update position if the new fitness is greater than the current value and recalculate position otherwise. Abandon this solution and generate a new one randomly if it remains stagnant for $limit$ times. Go back to Step 2 if the iteration number is less than $M CN$, and exit the algorithm to output the global optimum solution otherwise.

3.6 Output data

When all individuals arrive at station points, the system provides the function of path delineation to trace the experienced route visually, generating a path data set to store and output. The data set can be imported into a design system of 3D animation such as Maya or 3dMax to generate crowd simulation effect with higher visualization quality.

4 Simulation experiments on ACIS/HOOPS

To demonstrate the good performance of the proposed method, we make microcosmic simulation experiments on crowd motion based on constrained and controlled group formation.

4.1 Experimental environment

These experiments adopt the virtual environment modeling method [25] based on component, to build a modeling system using Visual Studio.NET 2003 and ACIS/Hoops [26] as platforms on a DELL Precision T3500 (Xeon W3550/12GB/500GB) graphic workstation. The basic configuration of this system is listed below.

Hardware system: Xeon W3550 quad-core processors, 12G DDR3 RAM, NVIDIA Quadro FX580 HD graphics, Broadcom 5761 gigabit ethernet controller, etc.

Software system: Visual Studio.NET 2003, Xtreme Toolkit Pro, ACIS/HOOPS modeling engine and application framework, etc.

4.2 Description of operation panel

The operation panel is described in Fig. 6.

The functions of this operation panel are listed below.

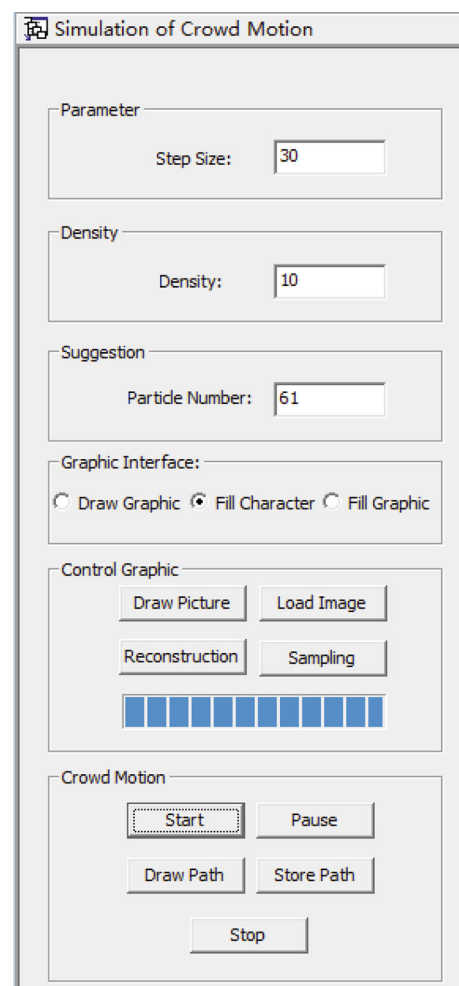


Fig. 6 The operation panel to generate freestyle formations

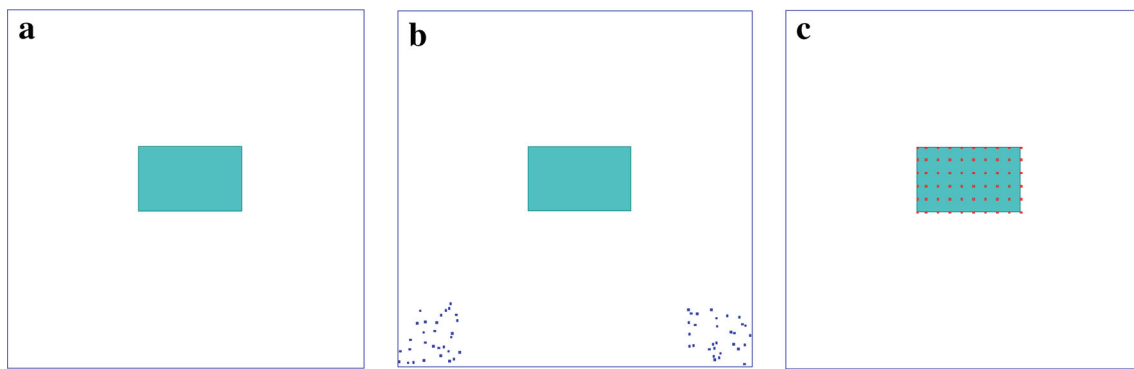


Fig. 7 The simulation experiment based on geometry graphics. **a** Draw a graphic, **b** initialize crowd, **c** the final formation

Step Size: Limits the size of a single step for all individuals, to maintain good visual effects.

Density: Defines the distribution density to limit the distances between individuals.

Particle Number: Displays the number of sample particles.

Draw Graphic: Allows users to draw basic geometry graphics directly in the 3D environment.

Fill Character: Samples and fills characters, graphics or patterns based on the filling principle of characters.

Fill Graphic: Samples and fills graphics or patterns based on the filling principle of graphics.

Draw Picture: Provides the function to draw 2D characters, graphics and patterns.

Load Image: Loads image files with characters, graphics or patterns.

Reconstruction: Reconstructs characters, graphics or patterns in the 3D environment after analysis.

Sampling: Analyzes and samples characters, graphics or patterns based on the density and shows the number of station points in the “Suggestion”.

Start: Starts the process of crowd motion.

Pause: Pauses the process of crowd motion.

Draw Path: Shows the path of each individual.

Store Path: Outputs and stores the path data.

Stop: Stops the process of crowd motion and clears the environment information.

station points based on density. All particles make self-adaptive matching based on the procedure described in Sect. 3.4, and plan their paths to destinations based on the procedure described in Sect. 3.5. The final formation is shown in Fig. 7c. The color of each station point will change to red to indicate that the particle has reached its destination. As can be seen from Fig. 7c, the formation is filled with regular queue, uniform space and high fidelity.

4.3.2 Fill characters

In this experiment, users can firstly draw 2D characters using the function of “Draw Picture”. Then the image with characters can be imported and analyzed in this system by the function of “Load Image”. Finally, the characters will be constructed in the 3D environment by the function of “Reconstruction”. The process is described in Fig. 8.

Figure 8a shows some characters we input in an image and then saved in binary format. Figure 8b shows the system’s 3D reconstruction for the characters in the ACIS/HOOPS environment. Figure 8c depicts the result of sampling. Rectangular particles represent the crowd in Fig. 8d, e. In Fig. 8e, the particles, self-adaptively, choose station points and plan the paths to their destinations intelligently. The red station points in Fig. 8f represent that all particles have arrived at their destinations.

4.3.3 Fill border of patterns

In this experiment, users can draw images with graphics or patterns using the function of “Draw Picture” and import the images to the system. In the sampling process, the system analyzes the graphics or patterns and only samples on the borderline to generate station points filling the contour. The special formation is finally generated through path planning and intelligent motion. The process is described in Fig. 9.

Figure 9a shows the peace dove we draw in an image and then saved in binary format. Figure 9b shows the system’s 3D

4.3 Simulation experiments based on typical cases

4.3.1 Draw graphics

In this experiment, users can draw a basic geometry graphic in the 3D environment by choosing a special graphic from the toolbox. The process is described in Fig. 7.

Figure 7a shows a rectangle we draw with the ACIS/HOOPS modeling system. In Fig. 7b, rectangular particles represent the initial crowd. The system automatically analyzes and samples on the rectangle graphic and generates

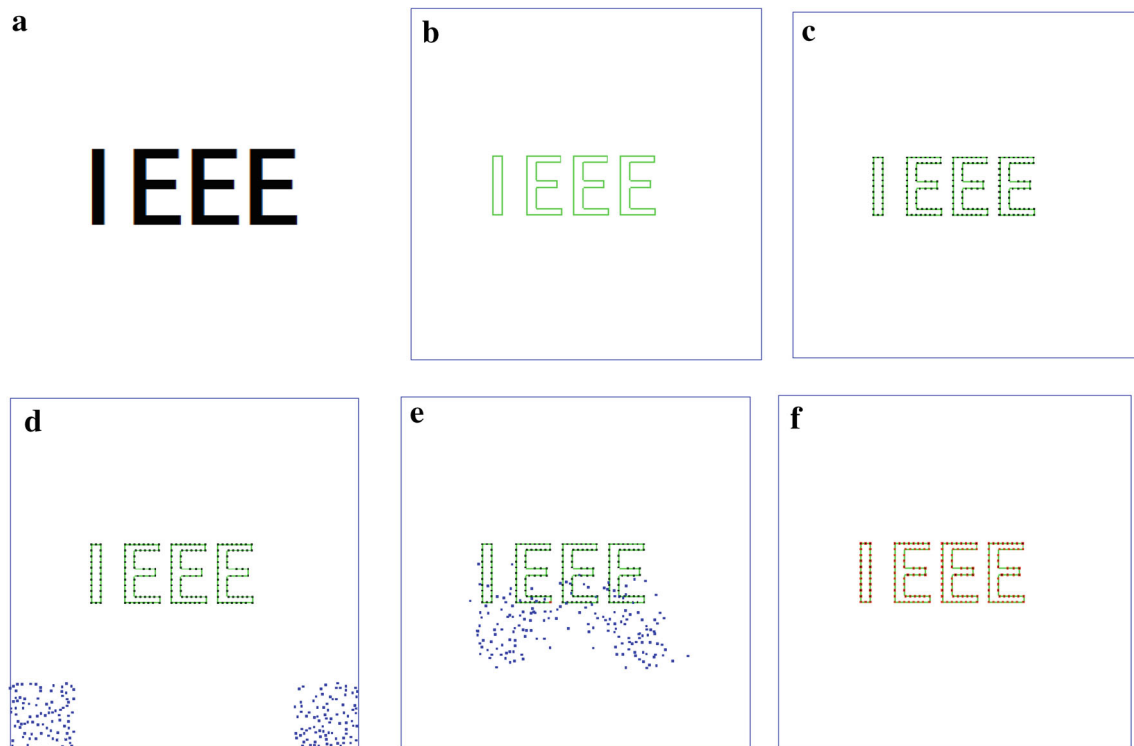


Fig. 8 The simulation experiment based on characters. **a** 2D characters, **b** 3D reconstruction, **c** the result of sampling, **d** initialize crowd, **e** a screenshot in crowd motion, **f** the final formation

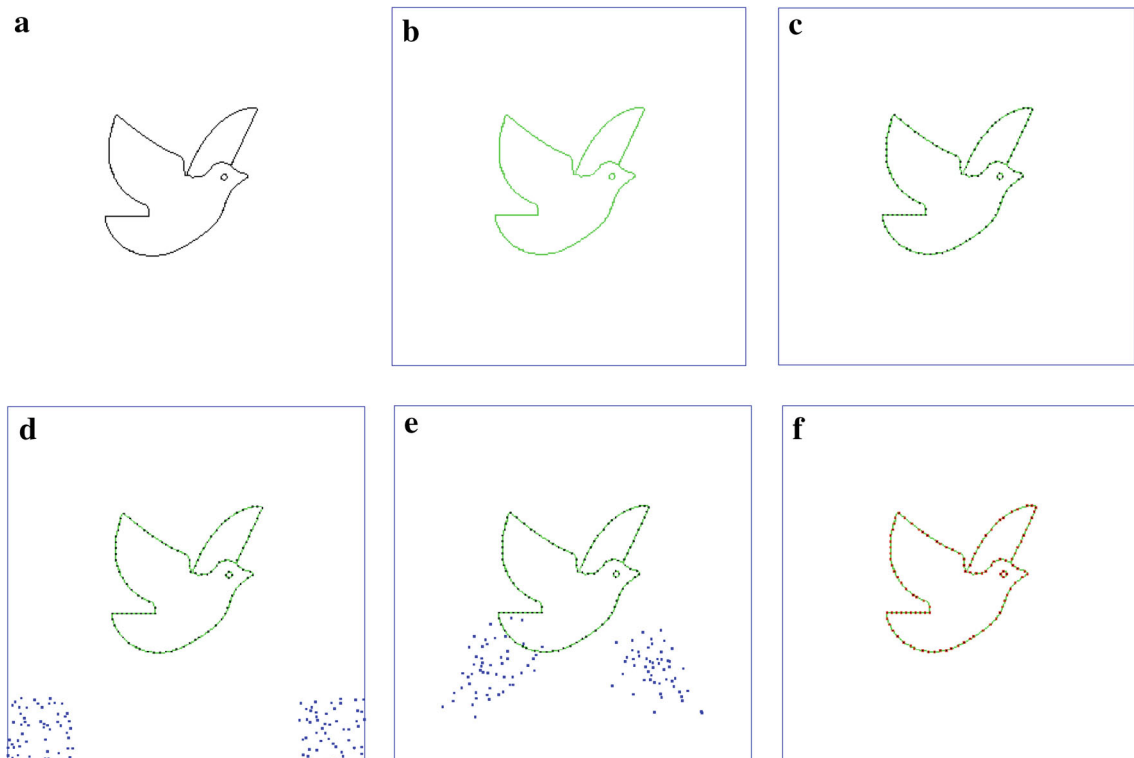


Fig. 9 The simulation experiment based on the border of patterns. **a** The 2D pattern, **b** 3D reconstruction, **c** the result of sampling, **d** initialize crowd, **e** a screenshot in crowd motion, **f** the final formation

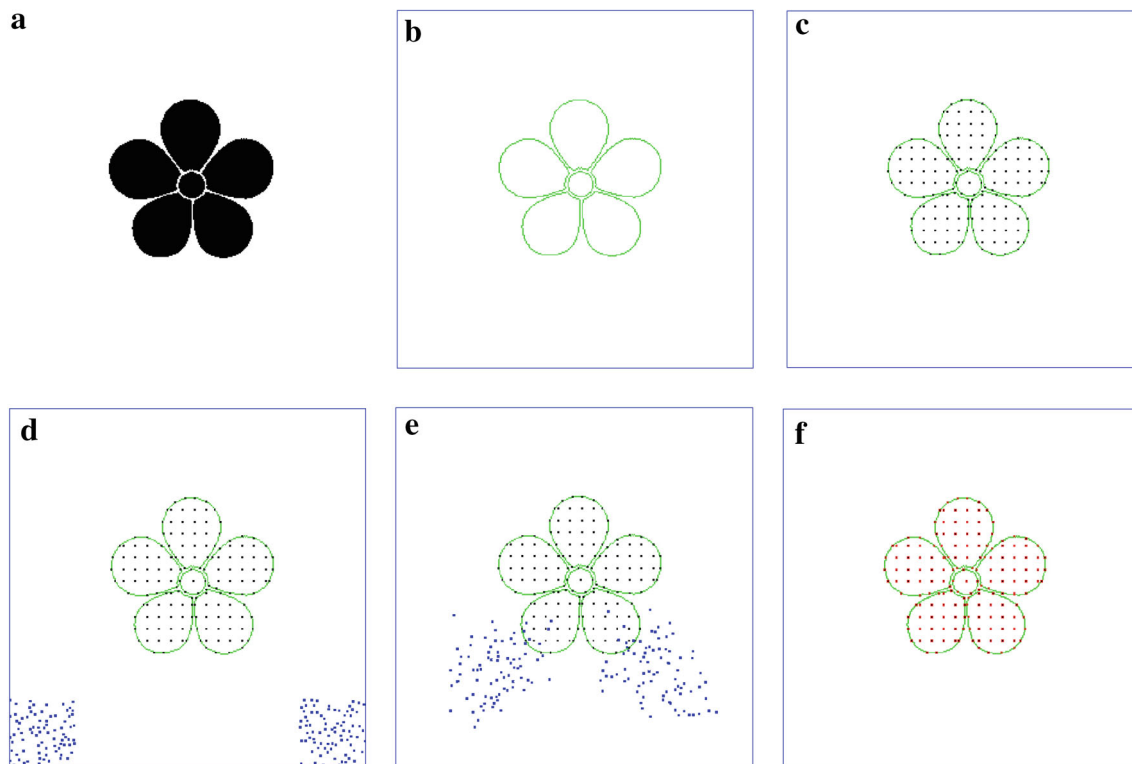


Fig. 10 The simulation experiment based on patterns. **a** The 2D pattern, **b** 3D reconstruction, **c** the result of sampling, **d** initialize crowd, **e** a screenshot in crowd motion, **f** the final formation

reconstruction for the pattern in the ACIS/HOOPS environment. Figure 9c shows the result of sampling. Rectangular particles represent the crowd in Fig. 9d, e. Figure 9e shows that the particles, self-adaptively, choose station points and plan the paths to their destinations intelligently. The red station points in Fig. 9f represent that all particles have arrived at their destinations.

4.3.4 Fill patterns

In this experiment, users can draw images with graphics or patterns using the function of “Draw Picture”s and import the images to the system. In the sampling process, the system analyzes the graphics or patterns and samples on the whole area (including the border and inner region). The special formation is finally generated through path planning and intelligent motion. The process is described in Fig. 10.

Figure 10a shows the flower we draw in an image and then saved in binary format. Figure 10b shows the system’s 3D reconstruction for the pattern in the ACIS/HOOPS environment. Figure 10c shows the system’s uniform sampling of the border and inner region. The crowd is initialized at two entrances, as shown in Fig. 10d. Figure 10e shows that the particles choose the border or inner station points adaptively and move to their destinations intelligently to generate the final formation, as shown in Fig. 10f.

4.4 Performance comparison with other related method

In this section, we make performance experiment and analysis on our proposed method (PM) compared with the method proposed in Ref. [15] (RM). In this experiment, a heart-shaped pattern is used as the target formation shape. The crowd’s distribution and the time overhead for generating the specified formation are the core evaluation indices in different groups of experiments. The representative results obtained are described in Fig. 11.

In this experiment, the separation distance between every two particles is set to 10 pixels. Figure 11a shows a screenshot in crowd motion based on the method proposed in Ref. [15]. This method does not evaluate the crowd’s density when matching station points, which results in the congestion around the optimum station points and a speed decrease of crowd motion. Figure 11b shows a screenshot in crowd motion based on our proposed method. In this method, the distance and density are all taken into consideration when matching station points, which can guarantee avoidance of congestion and improve the speed of crowd motion. Table 1 shows the experimental results under different constraints of the separation distance.

In the No. 1 experiment, the separation distance is big, which results in the low number of particles and low probability to be congested. Therefore, the difference of time over-

Fig. 11 The experimental results of these two methods. **a** A screenshot in crowd motion based on the RM, **b** a screenshot in crowd motion based on the PM

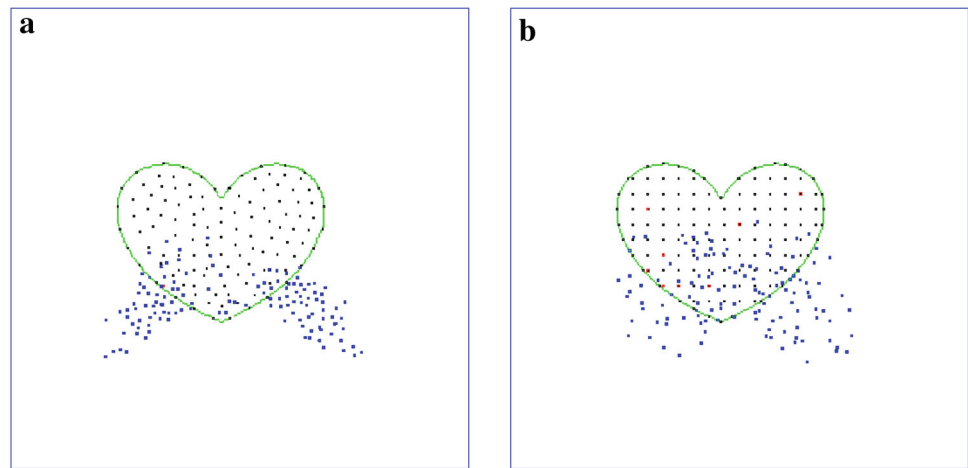


Table 1 The experimental results under different constraints of the separation distance

Number	Separation distance /pixel	The number of particles	PM's time overhead/s	RM's time overhead/s	Speed-up ratio (%)
1	15	36	9.7	10.2	5.2
2	10	129	20.6	26.5	28.6
3	5	456	53.4	70.8	32.6

head between the two methods is not obvious. However, our method's performance superiority becomes more and more obvious along with the decrease of the separation distance. As is shown in the No. 2 and the No. 3 experiments, when using the RM, the probability of resulting in congestion becomes much bigger because of the increase of the number of particles. It results in the speed decrease of crowd motion and increase of the time overhead. While our method can guarantee avoidance of congestion effectively and maintain good performance when the crowd's size is large. The reason for our method's rapid speed exists in two points. First, the ABC algorithm, which is used to plan path, has a simple model and few parameters. It can reduce the computational complexity and improve the convergence speed in a way. Moreover, our method takes the distance as well as the density into consideration when matching station points, which provides an effective approach to increase the speed of crowd motion.

We point out that our proposed method and built system can deal with characters, patterns and graphics effectively, generating the final formation in accordance with users' input and with uniform distribution, good visualization effects and high biofidelity. The main processes, including 3D reconstruction, cross sampling, self-adaptive matching and intelligent motion play important roles in the crowd simulation of freestyle formations. An interactive interface to set density is also provided in this system to control formations dynamically to improve the system's intelligence.

5 Simulation experiments on Maya

To enhance the visualization quality of simulation experiment and exhibit the practical value of our research, we make simulation experiment on group formations using Maya platform. We develop a plug-in module with built-in scrip language to read the environment information and import the data obtained from the ACIS/HOOPS modeling system to Maya. A model base is also set to render and update scene with high fidelity. This experiment takes the large-scale joint performance of group calisthenics as simulation object, simulating the processes of cross sampling, self-adaptive matching and intelligent motion. The simulation results are depicted in Fig. 12.

In Fig. 12a, a city square constitutes the basic scene. In Fig. 12b, the formation pattern and crowd are initialized, respectively, at the center and at the entrances of the square. Figure 12c shows the individuals in the crowd, self-adaptively, setting the target station points and moving to their destinations intelligently, generating the final formation as depicted in Fig. 12d. This experiment demonstrates that it is possible to visually define and simulate the process of group formation with high quality, exhibiting good accuracy, adaptability and global controllability. The research also provides fundamental theoretical and experimental references to other crowd simulation experiments and related researches of the control of natural crowd motion.



Fig. 12 The simulation experiment on Maya. **a** The scene of a square, **b** initialize the pattern and crowd, **c** a screenshot in crowd motion, **d** the final formation

6 Conclusion

This research proposes a crowd simulation method based on constrained and controlled group formation. We also make microcosmic simulation experiments both in the ACIS/HOOPS and Maya environments, providing fresh ideas for the research of group formations. The interfaces of drawing graphics, loading images and setting density guarantee good interactivity of this system. The cross sampling implements uniform distribution of the crowd, while self-adaptive matching and path planning, based on Swarm Intelligence, improve the system's intelligence and visualization effect. The data set output from the system provides data template and reference to implement realistic simulation behaviors with much higher quality.

The computational efficiency and visualization effect are key factors in crowd simulation. Therefore, our future research will focus on the three points below.

1. Optimize existing sampling algorithms and improve the ability and accuracy to deal with complex characters, graphics and patterns. Make sure that the sampling can fill key points accurately, exhibiting good distribution and regular appearance.
2. Combine the existing computational model with parallel or distributed computing modes. Propose crowd simulation methods based on parallelization to improve efficiency and accuracy.
3. The sampling with uneven distribution is always required in many real simulation scenarios. So, we should improve our method to make it much more universal to implement control of uneven distribution or local distribution.

Acknowledgments This research is supported by: the Natural Science Foundation of China under Grant Nos. 61272094, 61202225, 61303007, 61303157; the Natural Science Foundation of Shandong Province under Grant No. ZR2010QL01; and the Project of Shandong

Province Higher Educational Science and Technology Program under Grant Nos. J11LG32, J13LN13.

References

- Braun, A., Musse, S.R., de Oliveira, L.P.L., et al.: Modeling individual behaviors in crowd simulation. In: 16th International Conference on IEEE Computer Animation and Social Agents, pp. 143–148 (2003)
- Pelechano, N., O'Brien, K., Silverman, B., et al.: Crowd simulation incorporating agent psychological models, roles and communication. Pennsylvania Univ Philadelphia Center For Human Modeling And, Simulation (2005)
- Thalmann, D.: Crowd Simulation. Wiley, New York (2007)
- Paris, S., Donikian, S., Bonvalet, N.: Environmental abstraction and path panning techniques for realistic crowd simulation. *Comput. Animat. Virtual Worlds* **17**(3–4), 325–335 (2006)
- Luo, L., Zhou, S., Cai, W., et al.: Agent-based human behavior modeling for crowd simulation. *Comput. Animat. Virtual Worlds* **19**(3–4), 271–281 (2008)
- Chen, Y., Lin, Y.: Controlling the movement of crowds in computer graphics by using the mechanism of particle swarm optimization. *Appl. Soft Comput.* **9**(3), 1170–1176 (2009)
- Bo, Y., Cheng, W., Hua, H., et al.: A multi-agent and PSO based simulation for human behavior in emergency evacuation. In: 2007 International Conference on IEEE Computational Intelligence and Security, pp. 296–300 (2007)
- Banarjee, S., Grosan, C., Abraham, A.: Emotional ant based modeling of crowd dynamics. In: Seventh International Symposium on IEEE Symbolic and Numeric Algorithms for Scientific Computing, 2005 SYNASC 2005 (2005)
- Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. *Lett. Nat.* **407**(6803), 487–490 (2000)
- Yang, L., Li, J., Zhao, D.: Evacuation microcosmic disperse model based on individual's behavior. *Sci. China Ser. E: Eng. Mater. Sci.* **34**(11), 487–490 (2000)
- Zhao, S., Su, G., He, Y.: Research of emergency evacuation system simulation based on cellular automata. *J. Chin. Comput. Syst.* **28**(12), 2220–2224 (2007)
- Takahashi, S., Yoshida, K., Kwon, T., et al.: Spectral-based group formation control. In: *Computer Graphics Forum*, vol. 28, no. 2, pp. 639–648. Blackwell Publishing, Oxford (2009)
- Pelechano, N., Allbeck, J.M., Badler, N.I.: Controlling individual agents in high-density crowd simulation. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, pp. 99–108 (2007)
- Silveira, R., Prestes, E., Nedel, L.P.: Managing coherent groups. *Comput. Animat. Virtual Worlds* **19**(3–4), 295–305 (2008)
- Gu, Q., Deng, Z.: Generating freestyle group formations in agent-based crowd simulations. *Comput. Graphics Appl.* **33**(1), 20–31 (2013)
- Gu, Q., Deng, Z.: Formation sketching: an approach to stylize groups in crowd simulation. In: Proceedings of Graphics Interface 2011. Canadian Human-Computer Communications Society, pp. 1–8 (2011)
- Li, J., Mao, T., Jiang, H.: Shape constraining and control in crowd animation. *J. Comput. Aided Design Comput. Graphics* **22**(7), 1158–1165 (2010)
- Ioannidis, K., Sirakoulis, G.C., Andreadis, I.: Cellular ants: a method to create collision free trajectories for a cooperative robot team. *Robot. Auton. Syst.* **59**(2), 113–127 (2011)
- Zhao, W., Tan, R., Li, Y.: Real time collision detection algorithm in complex virtual environment. *J. Syst. Simul.* **22**(1), 125–129 (2010)
- Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University (2005)
- Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Global Optim* **39**(3), 459–471 (2007)
- Sun, Y., Liu, H., Cao, J.: Path generation method for swarm animation based on artificial bee colony algorithm. *Comput. Eng.* **37**(22), 131–133 (2011)
- Karaboga, D., Basturk, B.: On the performance of artificial bee colony algorithm. *Appl. Soft Comput.* **8**(1), 687–697 (2008)
- Bao, L., Zeng, J.: Comparison and analysis of the selection mechanism in the artificial bee colony algorithm. In: 2009 Ninth International Conference on IEEE Hybrid Intelligent Systems, vol. 1, pp. 411–416 (2009)
- Li, Z., Si, G., Hu, X.: Research on modeling&simulation of crowd behavior based on smart environment. *J. Syst. Simul.* **20**(s), 195–201 (2008)
- Zheng, X., Liu, H.: A multi-objective evaluation based cooperative character modeling system. In: 2011 15th International Conference on IEEE Computer Supported Cooperative Work in Design (CSCWD), pp. 168–174 (2011)

Author Biographies



Peng Zhang is a postgraduate of School of Information Science and Engineering, Shandong Normal University, Jinan, China. His current research interests include Swarm Intelligence, Optimization Algorithm and Computer Aided Design.



Hong Liu is currently the president of School of Information Science and Engineering, Shandong Normal University, Jinan, China. She is also the director of Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology. She has contributed over 100 articles to professional journals, such as *Computer Science and Information Systems*, *International Journal of Computer Applications in Technology*, *International Journal of Advances in Engineering Software*, etc. Her current research interests include Distributed Artificial Intelligence, Software Engineering and Computer Aided Design.



Yan-hui Ding is an associate professor of School of Information Science and Engineering, Shandong Normal University, Jinan, China. He received a Ph.D. degree in Computer Software and Theory from Shandong University in 2010. His current research interests include Artificial Intelligence and Data Integration.