


RESEARCH ARTICLE

Biologically inspired ant colony simulation

Wei Xiang¹ | Jiaping Ren¹ | Kuan Wang¹ | Zhigang Deng² | Xiaogang Jin¹ 

¹State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

²Department of Computer Science, University of Houston, Houston, Texas

Correspondence

Xiaogang Jin, State Key Lab of CAD&CG, University of Houston, Hangzhou 310058, China.

Email: jin@cad.zju.edu.cn

Funding information

National Key R&D Program of China, Grant/Award Number: 2017YFB1002600

Abstract

We present a unified biologically inspired approach to simulate ant colonies inspired by the key observation of collective behaviors of ants in nature. To generate the trajectories of virtual ants, we construct a motion controller to determine the motion states and the paths of virtual ants, considering dynamic internal and external interactions. The motion controller computes a target position for each ant at every time step according to its motion states. The motion states include four states: basic movement, the stop state, and two dynamic interactions (i.e., *internal* and *external*, respectively referring to interaction with neighbors for necessary information transfer about the destination, and interaction with surroundings such as food sources, nests, and obstacles) to represent basic exploration, casual or intentional stop, and purposeful movement, respectively. Based on the motion states, the motion controller plans an optimal path for each virtual ant. Through many simulation experiments, we demonstrate that our method is controllable, scalable, and flexible to simulate hybrid colonies with a large number of ants.

KEYWORDS

ant colony simulation, biologically inspired, information transfer, unified motion controller

1 | INTRODUCTION

Collective behaviors are common for most insect species including ants. In the real world, ants work collectively for foraging, defending, and food conveying. However, due to the tiny body of individual ants, the high density of an ant colony, and the often complicated living environment of ants in nature, it is hard to capture the ground-truth movement data of ants in the real world. As a result, this imposes an additional challenge for the realistic simulation of large-scale ant groups.

Previous works on the simulation or animation of insect groups have been mainly focused on *flying* insects, but it is nontrivial to extend these existing methods for ant colony simulation due to their significant differences in biological mechanism. Researchers have done works on simulating the unidirectional or bidirectional trail pheromone of ants^{1–3} and proposed cellular automaton models to simulate nonblocking traffic flow on ant trails.⁴ Because of the complex living environment of ants, the aforementioned trail models cannot be straightforwardly applied to model collective behaviors of ant groups at scale. In addition, some existing methods have been focused on body control to make ants locomotive^{5,6} and on simulating collective transportation of ants to carry an object.⁷ Despite certain successes of the above existing works focused on the simulation of *specific* behaviors of ants, to date, researchers have not yet proposed a unified, *generalized* motion control model for ant colonies, to the best of our knowledge.

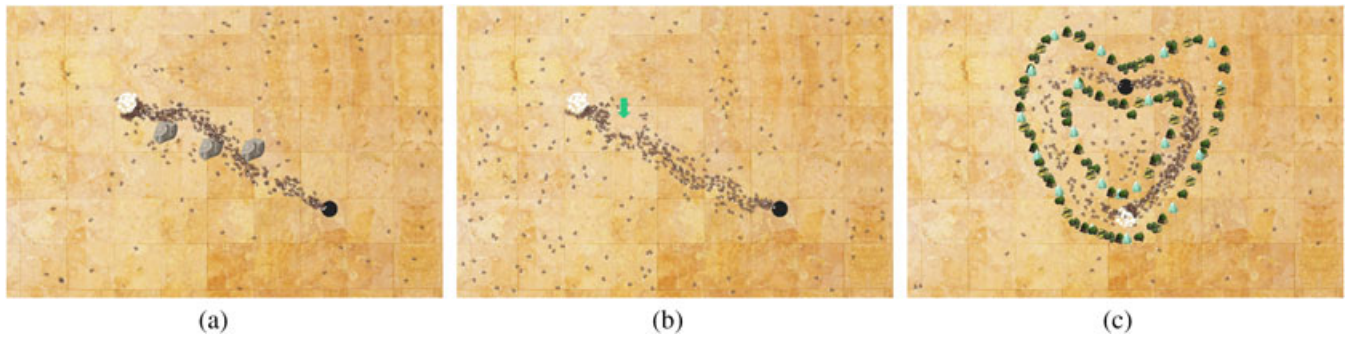


FIGURE 1 (a) An ant colony transports food between a food source and the nest. (b) An ant colony adapts to dynamic external disturbance. (c) An ant colony expresses a heart shape under the constraint of arranged obstacles

In this paper, we propose a unified motion control model for ant colonies. Our approach is inspired by a key biological observation on collective behaviors of ant colonies, that is, ants in a same group share information through secreting pheromone or physical contact.^{8–10} Specifically, our approach can model the probabilistic interactions among ants and with the environment, and further generate a collision-avoidance path for each virtual ant. In addition, our approach offers an intuitive way for users to interactively control the geometric shapes of virtual ant colonies. Figure 1 shows some simulation results of our method.

The main contributions of this work can be summarized below.

- **Unified biologically inspired motion controller for ant colonies.** The motion controller coordinates the collective behavior of ant colonies by managing and switching each individual's motion state as one of the following: basic movement state, stop state, and two different interaction states. With this motion controller, we can effectively simulate various behaviors of ant colonies, including *foraging* behavior, where ants automatically find the food source and generate a bidirectional path between the food source and the nest; *avoiding* behavior, where ants avoid obstacles ahead; and *escaping* behavior, where the colony flees from an external attack.
- **Intuitive and scalable control.** Our method is able to simulate both small and large sizes of ant colonies in either simple or complex environments. The steerable characteristic of our simulation is realized by adjusting some parameters of our algorithm such as the moving speed and the size of the colony, or changing the motion scenario interactively by adding or removing key objects. Our approach also offers interactive animation control for real-time simulation applications.

Not limited to the simulation of ant colonies in this work, our approach can be potentially extended for many other applications that need collaborative coordinations among many entities to achieve certain tasks, such as microrobots or nanorobots, drone groups, and other biological organisms. Our approach can also be used to generate various realistic group formations consisting of a large number of individuals (such as humans, insects, and ants), for special visual effects in entertainment and virtual reality applications.

The remainder of this paper is organized as follows. Section 2 introduces related works that are most relevant to ant simulation. Section 3 gives a pipeline overview of our approach. In Section 4, we explain the details of our method, which contains theoretically basic rules and our algorithm. We show simulation results in Section 5 and discuss the limitations and future research directions of this work in Section 6.

2 | RELATED WORK

Motion control of multiagent groups. Numerous methods have been proposed for behavior simulation or motion control of multiagents in a group, especially focused on human crowds, bird flocks, and recent insect swarms.

Most of existing approaches regard agents in a group as autonomous individuals and execute global path planning together with local navigation and collision avoidance for individuals. As the two most seminal force-based approaches, the boid's model¹¹ generates movements of individuals using heuristic rules: separation, alignment and cohesion, and the social force model¹² combines sociological factors with physical forces to model interactions among pedestrians. Following the above works, many approaches have been proposed, for example, some rule-based methods for crowds¹³ consider

hierarchical rules based on individual behavior, variations of the social force model (e.g., the work of Pelechano et al.¹⁴) control individuals in dense crowds, and the anticipatory model predicts and resolves collisions in crowds.¹⁵ Compared with real video of bird flocks, a self-organized aerial display that combines the three usual rules in the boid's model with starling behavior has been proposed.¹⁶ Though force-based methods appear easy to be formulated and succeed in simulating multiagent groups, the simulation results are affected by the instability numerical values, for example, forces vary quickly and can reach extremely large; meanwhile, researchers often suffer from tedious parameter tuning to obtain better simulation results.

Velocity-based approaches spring up these years, often using a cost function to select a velocity in the velocity space for each individual at every time step. Researchers have proposed vision-based steering approaches to simulate collision-free crowds by using visual perceptions of individuals on the environment,¹⁷ compute pedestrian trajectories to match real density-dependent avoidance behavior among people,¹⁸ and steer agents by constantly minimizing the risk of collision and improve simulation quality.¹⁹ Recently, an implicit crowd simulation method²⁰ incorporating general dissipation functions in optimization-based backward Euler has been proposed to provide collision-free motions of crowds with high-quality collective behaviors. The aforementioned approaches cannot directly be applied to simulate ant colonies, mainly because individual behaviors (interactions) of ants not only benefit their collision avoidances but also have important responsibilities to transfer information for foraging or other purposes.

From the macroscopic point of view, continuum crowd approaches often use a dynamic potential field to integrate global navigation with local collision avoidances and protect large crowds from explicit collisions,²¹ and deal with interagent collisions through unilateral incompressibility constraints for dense crowds.²² A nonequilibrium continuum dynamical model has also been proposed for the collective motion of large groups like bird flocks.²³ Though the continuum methods can simulate flow-like, smooth and dense crowds or bird flocks without collisions, they cannot be used in ant colony simulation because ants are discrete in the colony, even though sometimes the colony's motion is similar to a track flow (e.g., the track between a food source and their nest). Some researchers also simulate pedestrian dynamics using a two-dimensional cellular automation.²⁴ Furthermore, Wang et al.²⁵ propose a first-of-its-kind hybrid model to simulate various behavior patterns of insect swarms, based on a noise function and potential fields.

Data-driven (example-based) approaches learn features of motion from real data and use the learned features to build statistical models to simulate crowds²⁶ and insect swarms,²⁷ or use the analyzed features to reverse engineer underlying individual rules for bird flocks.²⁸ Based on examples, some approaches blend existing crowd data to generate a new crowd animation,²⁹ clone crowd motions from the existing examples to animate large crowds,³⁰ and diversify context-aware motion for crowd simulation.³¹ Recently, a biologically plausible dynamics model^{32,33} has been proposed to compute collision-free trajectories for flying insect swarms through a hybrid formulation that combines a force-based model with a data-driven noise model to display interactions among insects. However, due to the complex living environment and the extraordinary high density of ants in many situations (e.g., tracks between the food sources and the nest), it is hard to obtain accurate motion data of ant colonies in real world.

Though many methods specialized for motion control of multiagents have been proposed, to the best of our knowledge, none of them can directly deal with the simulation of ant colonies. Arguably, one of the main reasons is the significant distinctions of biological mechanism between ant colonies and other animal groups, such as communication mechanism and different living environments.

Ant simulation. Previous works have been done to compute trails for foraging ants. For example, cellular automation-based models³⁴ can simulate one-dimensional unidirectional pheromone flow in an ant trail by controlling nonmonotonic variation of speed with the trail density and apply to a 2D robotic trail simulation for pedestrians,³⁵ and then study the spatio-temporal organization of ants on the trail to build bidirectional trails for other traffic systems.⁴ In addition, researchers have developed mathematical models to analyze biological mechanisms or motion rules of ants^{36,37} and provide a physics-based model for repulsively interacting self-propelled particles undergoing single-file motion.³⁸ However, directly applying the motion rules derived from the above data analyses often fails to produce realistic visual simulation of ants for CG applications, due to the intrinsic simplicity of the derived rules.

Researchers have also explored the modeling of ants' locomotion. Guo et al.⁶ describe a control pipeline to automate insects' locomotion on a predefined moving path. Later, Guo et al.⁷ further simulate collective transport of virtual ants on a predefined transmission path, by applying forces for the ants around the object to solve the positioning problem. These methods are primarily focused on the body joints' locomotion on a preset path to complete a special task; nevertheless, none of them can be straightforwardly extended to simulate a variety of behaviors of an ant colony at scale and simulate adaptive ant behaviors in dynamically changing environments, due to the hard-coded fixed tasks in their methodologies.

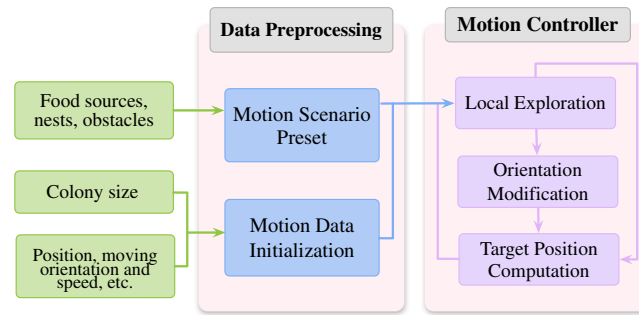


FIGURE 2 The pipeline of our approach

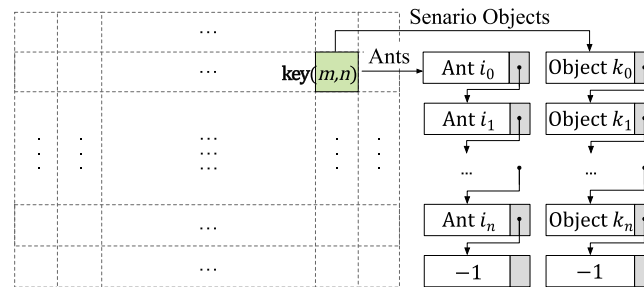


FIGURE 3 Illustration of the grid map used in our approach. Each cell in the m th row of the n th column has a unique key $\mathbf{key}(m, n)$ and links with a series of ants and scenario objects that are located in this cell. Note that the node “-1” denotes empty

3 | APPROACH OVERVIEW

As Figure 2 illustrates, our approach can be conceptually regarded as a two-phase computational process. At the data preprocessing stage, the whole motion area, which refers to the whole space of the scene, is formulated as a 2D grid map (Figure 3). The motion scenario can be preset with food sources, nests, and obstacles, and the motion data are initialized with the colony size, and the primary position, orientation, moving speed, and other motion states of each ant. Then, at the motion control stage, each ant moves to a new position without collisions at every time step. A motion controller is constructed here to have a local exploration process for each ant to sense the surroundings and decide which motion state the ant should turn into. After the local exploration for each ant, the motion controller modulates its orientation to avoid collisions or out-of-bounds and then computes its target position. For simplicity, in all of our experiments, the ants share the same moving speed, although our motion control model can be trivially extended to have time-varying speeds for different ants. Thus, the motion controller can continuously update orientation and position of ants, and generate paths for them.

The motion state of each ant could be (i) the *stop* state where the ant stops moving for a while, (ii) the *interaction with neighbors* state referring to its interaction with neighbors, (iii) the *interaction with environment* referring to its interaction with the environment, and (iv) the *basic movement* state where the ant runs randomly to explore the map. Specifically, Figure 4a illustrates possible motion state transitions for deciding a target moving orientation for an ant (see Section 4 in detail), and the interaction system is illustrated in Figure 4b, where external disturbance can be interactively introduced in real-time simulations. Our approach can simulate various ant behaviors such as *foraging*, *avoiding*, and *escaping* through interactions, and our method has the capability to adapt to a dynamically changing environment in real time.

4 | BIOLOGICALLY INSPIRED MOTION CONTROLLER

In this section, we will describe the details of the biologically inspired motion control model in our approach. As illustrated in Section 3, we define four motion states for an ant: stop, basic movement, interaction with environment, and interaction with neighbors. Figure 4a shows the possible transitions among these motion states. Note that ants prefer interacting with the environment to avoid obstacles or escape from external attacks rather than turning into the other states.

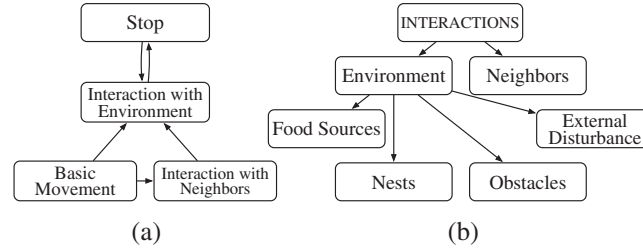


FIGURE 4 The state machine of an ant. (a) illustrates the state transition diagram of an ant when deciding its move at the next time step. (b) illustrates the interaction system in our approach. Specifically, the priorities for the ant to interact with the environment are obstacles > external disturbance > food = nest

We describe the pipeline of the motion control process below, where Step 2 and Step 3 aggregate as the local exploration process and decide a preferred target position $\mathbf{p}^*(t + 1)$ and an orientation $\mathbf{O}^*(t + 1)$.

- (1) Randomly choose to either enter the stop state (in Section 4.1) and quit the movement within a certain period, or turn to Step 2 for active motion updating.
- (2) Enter the basic movement state and calculate a target orientation. Details about the basic movement state are explained in Section 4.1.
- (3) Search in a valid range of distance to check possible interactions. If any interaction happens, enter the corresponding interaction state and recalculate a moving orientation according to the obtained information. Details on the interactions are elaborated in Section 4.2.
- (4) Check the boundary and collision avoidances for each ant, adjust the moving orientation to $\mathbf{O}(t + 1)$, and then calculate the target position $\mathbf{p}(t + 1)$ accordingly.
- (5) Move forward one step for each ant.

Except for the current position and orientation, we assign each ant with a status label $S_c \in \{S_{\text{nothing}}, S_{\text{food}}, S_{\text{nest}}\}$, which denotes where the ant comes from, and S_c represents what information it can transmit to other ants in the colony. Here, $S_c = S_{\text{nothing}}$ if the ant has not visited a food source or a nest; $S_c = S_{\text{food}}$ if the ant has visited a food source and laden with food; and $S_c = S_{\text{nest}}$ if the ant has visited its nest. In our simulation, ants are always unaware of the exact positions of their destinations (food sources or the nest), and the ants transfer information by recognizing the S_c labels of their neighbors to find the destinations. The label of each ant (S_c) will change when the ant arrives at a destination, and then, the ant will decide a new destination (refer to Section 4.2.2). Specifically, the ants with $S_c = S_{\text{nest}}$ (or S_{food}) may come from different nests (or food sources).

As mentioned in Section 3, we divide the motion area into a grid map (refer to Figure 3), which is utilized as a HashMap, and each cell in this map has a unique key. Each ant is associated with a key, and each key links with none or a few ants and/or scenario objects. An ant searches for neighbors or the neighboring objects from its located cell and the neighboring cells within a certain distance threshold.

The motion control module in our approach is described in Algorithm 1, and the interaction (i.e., information transfer) process within the motion control module is described in Algorithm 2. Together with the size of the ant colony M in Algorithm 1, the start position \mathbf{p}_0 , start moving orientation \mathbf{O}_0 , and the initial speed $speed$ are initialized at the beginning of the simulation. Particularly, the initial values of \mathbf{p}_0 , \mathbf{O}_0 , and S_c for an ant may differ in various motion scenarios with different purposes. For example, we set \mathbf{p}_0 as the position of its belonging nest, S_c as S_{nest} , and \mathbf{O}_0 as a random unit vector to simulate the scenario when an ant leaves its nest and goes foraging. For those scenarios when ants are not initialized in the nest, we will select an appropriate \mathbf{p}_0 for each ant to avoid overlapping.

4.1 | Basic motion control

In the basic movement state, an ant chooses a new orientation randomly as we assume it knows nothing about its neighbors and environment. An ant enters the basic movement state first when it decides not to be still during the period of the time step Δt . When an ant enters the basic movement state, a direction is randomly selected as illustrated in Figure 5a. To ensure the smoothness of the basic motion, the ant can only slightly change its moving orientation. That is, the critical

deviation angle θ should not be too large. In our simulation, we empirically set θ to $\pi/6$. Through the basic movements of ants, most of the motion area can be visited sooner or later.

Algorithm 1. Ant motion update at time t

Initialization:

Set \mathbf{p}_0 , \mathbf{O}_0 , speed, and S_c for every ant;

Iteration:

- 1: **for all** ant i ($i \geq 0$ **and** $i < M$) **do**
 - 2: turn into the basic movement state to change the orientation;
 - 3: check the current and neighboring cells to determine possible interactions;
 - 4: check collision avoidances and boundary conditions, determine $\mathbf{O}(t + 1)$; and
 - 5: calculate the target position and move to $\mathbf{p}(t + 1)$;
 - 6: **end for**
-

Algorithm 2. Interactions at time t

Interaction:

for all ant j in the range of ant i 's sense **and** $j \in$ the neighborhood of i **do**

- 2: **if** S_c^j is the destination of ant i **then**
- if** the distance is the shortest **then**
- 4: Change the moving orientation to the destination;
- end if**
- 6: **end if**

end for

- 8: Check for possible interactions with the environment;
- Update orientation.

Output:

$\mathbf{p}^*(t + 1)$, $\mathbf{O}^*(t + 1)$ and S_c of every ant.

For those ants that are not always active in moving, we set a stop state to simulate their stop situations. In the stop state, ants stay at their current positions and hold their orientations for a certain period. Specifically, ants turn into the stop state in the following situations.

- 1) *Casual stop*. Active ants randomly stop to take a rest (the probability is 0.005 in our simulation).
- 2) *Collecting food*. Ants that are not laden with food arrive at a food source and pause there to collect food. The S_c label of the ants will become S_{food} when the stop time runs out, and the ants start to move toward the nest.
- 3) *Dropping food*. Ants with the S_{food} label arrive at the nest and stop moving for a certain period to unload food. As soon as the stop time runs out, the S_c label turns into S_{nest} and the ants start to move back to the food.

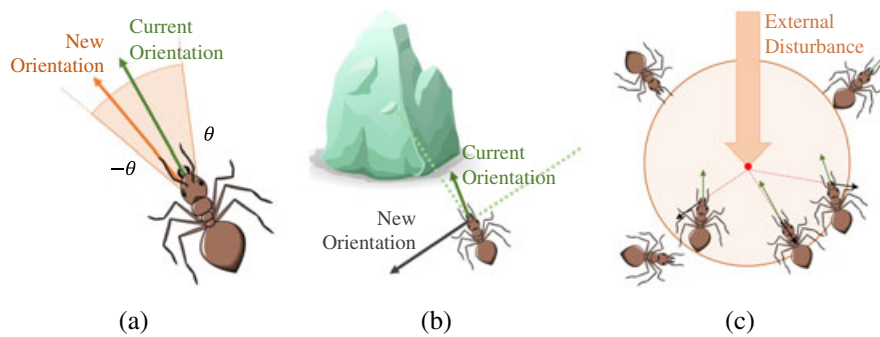


FIGURE 5 Selection of the moving orientations. (a) shows the selection domain in the basic movement state. The new orientation is randomly selected from a certain range $(-\theta, \theta)$ deviating from the current orientation. (b) exhibits the *avoiding* behavior when an ant interacts with an obstacle. (c) shows the *escaping* behavior when facing an external disturbance

TABLE 1 Situations when information transfer happens between two ants, where $j \in N_i$ and the neighbor j comes from ant i 's destination

	S_c^i	S_c^j	Information transfer from ant j to ant i
Situation 1	S_{nothing}	S_{food}	Information about the food source where j comes from.
Situation 2	S_{nest}	S_{food}	Information about the food source where j comes from.
Situation 3	S_{food}	S_{nest}	Information about the nest where j comes from.

Note. Ant j does not know the exact location of its coming place; it can only transfer information to ant i by body contact to lead i closer to the destination. Figure 6 illustrates the possible information transfer processes in detail.

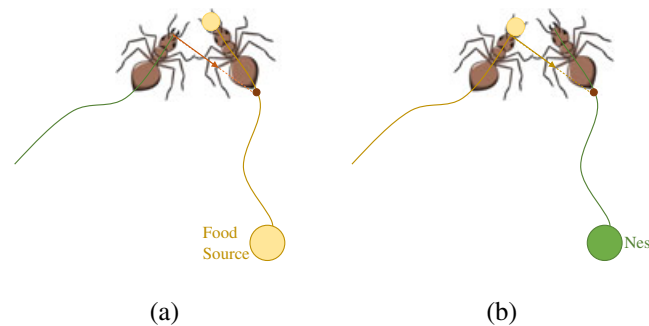


FIGURE 6 Information about a food source or the nest is transferred between two ants. In both (a) and (b), the left ant is the focal ant i and the right one is the neighboring ant j who comes from a food source or i 's belonging nest. Due to physical contact with j , the focal ant i knows the current body rotation of the neighbor; then, i changes its moving orientation to the neighbor's latest arrival position to move closer to the food source or the nest

4.2 | Interactions

Besides the basic movement process, ants also navigate in the current and nearby cells in the grid map, avoid collisions, escape from external disturbances, and develop a trail between a food source and the nest through interactions. Interactions among ants facilitate information transfer and correct the moving orientations of the ants to their destinations. Interactions with the environment enable the ant colonies to adapt to dynamically changing environments.

4.2.1 | Interaction with neighbors

The idea of the interaction with neighbors is inspired by a well-studied biological phenomenon where two ants can communicate with each other through physical contacts.^{8–10} In our method, we assume that physical contact happens once two ants are sufficiently close to each other. This type of information transfer forms information chains in the colony, thus speeding up the generation of the bidirectional foraging-transporting trail between a food source and a nest.

For each ant i , its neighbors N_i are defined as the ants that are close to ant i within a certain distance. An ant will interact with a neighbor if the neighbor can transfer information to it. Information transfer between two ants may happen under the following situations (also refer to Table 1 and Figure 6):

- (1) $S_c^i = S_{\text{nothing}}$ and $S_c^j = S_{\text{food}}$, $j \in N_i$: Specifically, the ant i has not yet visited a food source or the nest, while the neighbor comes from a food source. In this case, the ant i can get information about the food source from the neighboring ant j .
- (2) $S_c^i = S_{\text{nest}}$ and $S_c^j = S_{\text{food}}$, $j \in N_i$: The ant i has visited the nest and attempts to find a food source, while the neighbor comes from a food source and is laden with food. Information about the food source will be transferred from the neighbor j to the ant i . This information can also contribute to the automatic generation of a path from the nest to a food source. The information transfer mechanisms in the above Situations 1 and 2 are illustrated in Figure 6a.

- (3) $S_c^i = S_{\text{food}}$ and $S_c^j = S_{\text{nest}}$, $j \in N_i$: This represents that the ant i comes from a food source, laden with food, and is looking for the nest to unload food. The neighbor j comes from ant i 's belonging nest, and it can transfer information about the nest to the ant i . This situation explains the mechanism of path generation from a food source to the nest, and Figure 6b illustrates this information transfer process.

Through various compositions of the above Situations 2 and 3, an ant colony can generate a bidirectional and connected trail between a food source and the nest. Ants change their moving orientations when any of the aforementioned situations happen.

4.2.2 | Interaction with environment

To simulate the interactions between ants and environment, we consider several external factors that may affect an ant colony in our simulation experiments. These factors are basically composed of food sources, nests, obstacles, and external disturbances, and the corresponding ant behaviors are called avoiding obstacles, collecting food, dropping food, and fleeing from external attacks, respectively.

Interaction with Obstacles: It is instinct for animals to avoid barriers ahead, thus in our approach, we set obstacles in the environment to simulate the *avoiding* behavior of ants. In our method, the avoiding behavior takes action when an obstacle locates in front of an ant, and the distance is in a critical value. The motion controller will change the moving direction of the ant accordingly, similar to the obstacle avoidance policy used in steering behavior.³⁹ As illustrated in Figure 5b, the new orientation of the ant is vertical to the line from the ant's current position to the obstacle.

Interaction with Food Source or Nest: An ant interacts with a food source or the belonging nest to collect food or drop food, and the S_c of the ant will change after the interactions. The premise for an ant to interact with a food source is that it is unladen with food ($S_c = S_{\text{nest}}$ or $S_c = S_{\text{nothing}}$) and arrives at a food source. In this case, the ant switches to the stop state to collect food, and the S_c of the ant is changed into S_{food} after the interaction and then the ant starts moving to look for the nest to drop food. Similarly, an ant interacts with a nest when it is laden with food ($S_c = S_{\text{food}}$) and arrives at the nest. In such a case, the ant will switch to the stop state to drop food, and then, the S_c is changed into S_{nest} and the ant goes again for foraging. The cyclic switches of S_c accelerate the foraging and the transporting processes for the ants even though they never know the accurate positions of their destinations. Moreover, an ant with $S_c = S_{\text{nothing}}$ also interacts with the nest upon arrival, and then, S_c of the ant changes to S_{nest} while the ant keeps moving.

Interaction with External Disturbance: The *escaping* behavior is performed when there is an external interference in the motion area. The dangerous area is a circle centered at the disturbance point, and the ants within this circular area will escape. In our approach, the ants within the range of external attack run away in a direction that is far away from the center of attack (see in Figure 5c).

4.3 | Target position computation

After performing the interactions in local exploration, the motion controller determines the preferred moving orientation $\mathbf{O}^*(t + 1)$ for each ant without checking collisions. The preferred velocity $\mathbf{v}^*(t + 1) = v\mathbf{O}^*(t + 1)$ of each ant at time $t + 1$ is calculated by the predefined speed v and $\mathbf{O}^*(t + 1)$. The predicted position $\mathbf{p}^*(t + 1)$ is the position that an ant moves with its preferred velocity in a time step:

$$\mathbf{p}^*(t + 1) = \mathbf{p}(t) + \mathbf{v}^*(t + 1)\Delta t, \quad (1)$$

where $\Delta t = 1$ s in our simulation.

Boundary check will be performed when the predicted position of an ant has been calculated to constrain it inside the motion area, and then, the motion controller determines the target moving orientation $\mathbf{O}(t + 1)$ to avoid collisions for the ant:

$$\mathbf{O}(t + 1) = w_0\mathbf{O}^*(t + 1) + (1 - w_0)\mathbf{R}_d(t + 1), \quad (2)$$

where w_0 is the weight assigned to the preferred target orientation $\mathbf{O}^*(t + 1)$ to ensure certain continuity, and $\mathbf{R}_d(t + 1)$ represents the sum of the weighted repulsive orientations:

$$\mathbf{R}_d(t + 1) = \sum_{d=1}^{n_c} w_d \mathbf{R}_d^*(t + 1). \quad (3)$$

Here, n_c represents the total number of the ants that could collide with the current ant, and $\mathbf{R}_d^*(t+1)$ is the predicted repulsive orientation, which is the unit vector pointing from the d th colliding ant to the current ant:

$$\mathbf{R}_d^*(t+1) = \frac{\mathbf{p}_d^*(t+1) - \mathbf{p}^*(t+1)}{\|\mathbf{p}_d^*(t+1) - \mathbf{p}^*(t+1)\|_2}. \quad (4)$$

We assign a weight $w_d = \mathbf{C}/\|\mathbf{p}_d^*(t+1) - \mathbf{p}^*(t+1)\|_2^k$ to each repulsive orientation in Equation (3), where \mathbf{C} and k are constants. w_d has a negative correlation with the predicted Euclidean distance between the current ant and the d th colliding ant.

Based on the target moving orientation $\mathbf{O}(t+1)$ of an ant, the motion controller calculates the velocity $\mathbf{v}(t+1) = \mathbf{v}\mathbf{O}(t+1)$ of the ant. Finally, the ant moves to the target position:

$$\mathbf{p}(t+1) = \mathbf{p}(t) + \mathbf{v}(t+1)\Delta t. \quad (5)$$

5 | RESULTS AND EVALUATIONS

We have implemented our algorithm in C++ on a desktop computer equipped with Intel[®] Core i7 4.00 GHz, 32-GB RAM, and NVIDIA[®] Geforce GTX 1060 GPU. By regarding the grid map mentioned in Section 3 as a HashMap, the computation cost of neighbors search is reduced to $\mathcal{O}(1)$ because the neighbors search is often limited to the current and neighboring cells no more than a fixed distance. The computational complexity of our algorithm is $\mathcal{O}(M)$, where M is the size of the ant colony.

Figures 7 and 8 show several simulation results with 500 ants. We present our simulation result in terms of ant colony animations and preset a couple single-colony scenarios (Figure 7) and a couple hybrid-colony scenarios (Figure 8). In hybrid-colony scenarios, we divided all the ants into several colonies, thus ants may come from different nests. We

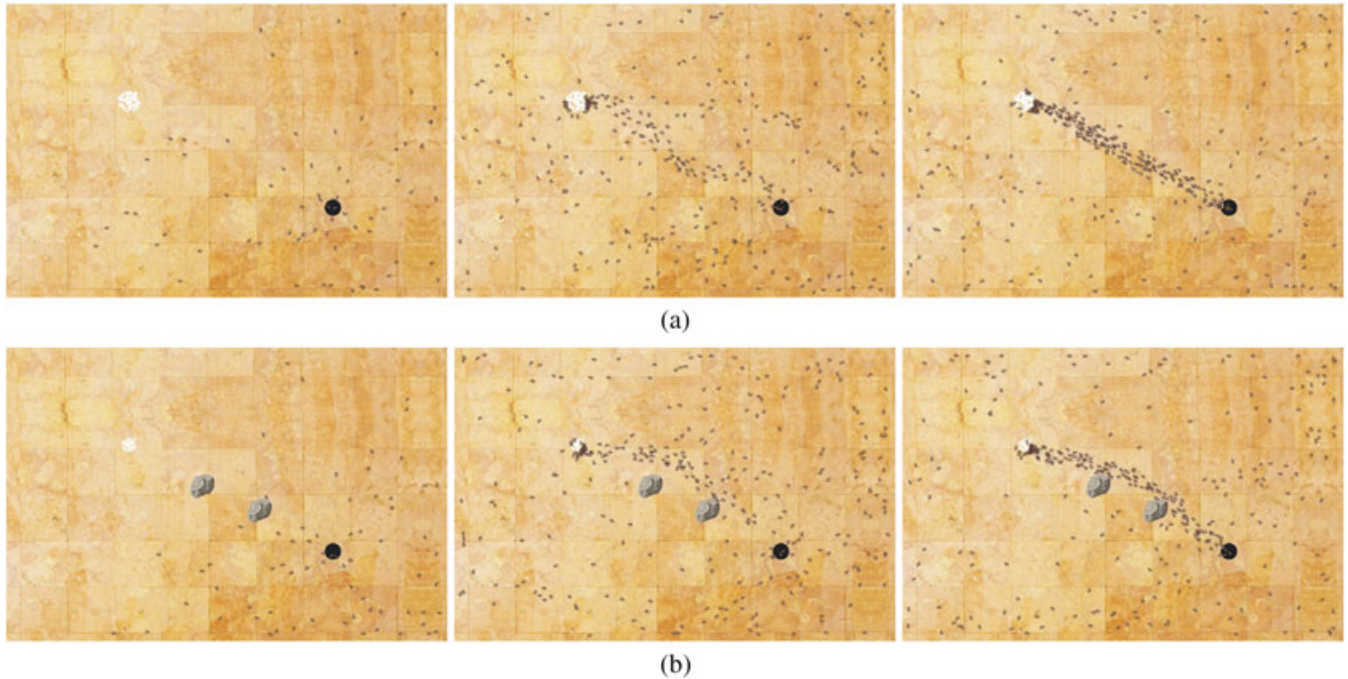


FIGURE 7 Different ant colony simulations using our approach in single-colony motion scenarios. The black hole, the white stack, and the rocks are the nest, the food source, and the obstacles. (a) Single colony with a food source. (b) Single colony with a food source and multiple obstacles

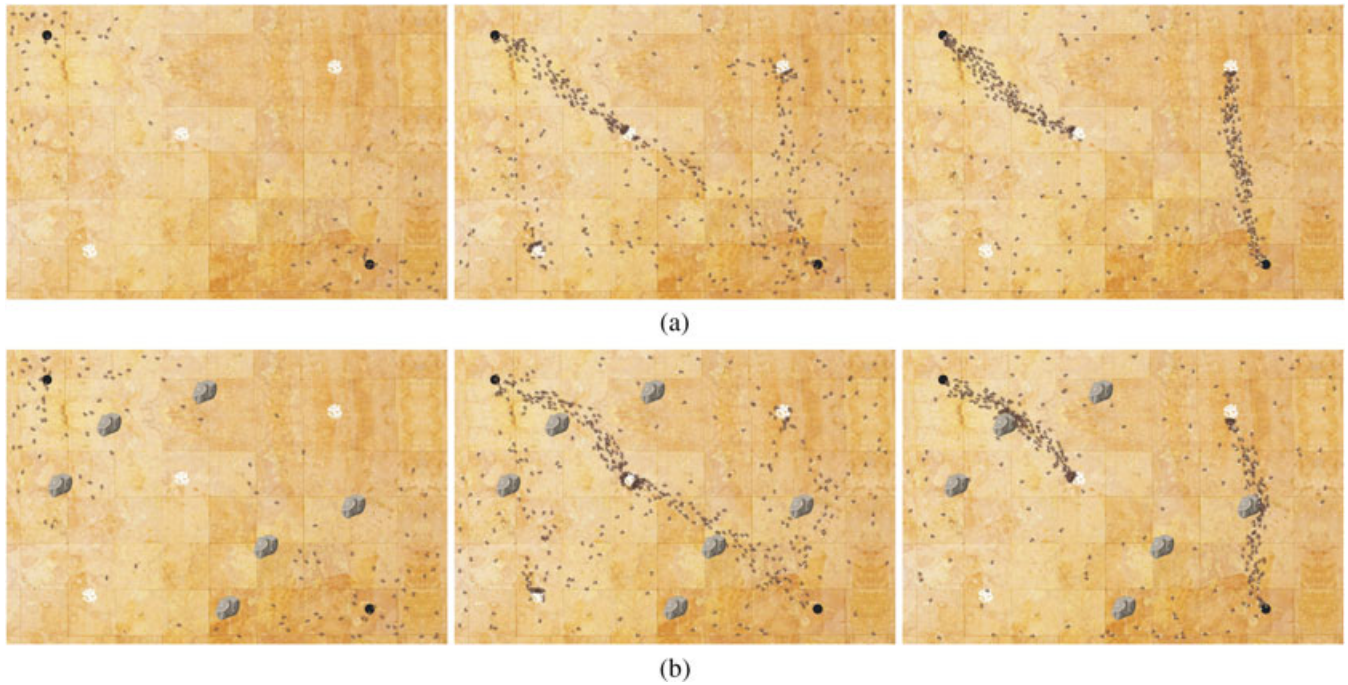


FIGURE 8 Different ant colony simulations using our approach in hybrid-colony motion scenarios. Note that each ant can only leave its belonging nest and return to the belonging nest. (a) Hybrid colonies with multiple food sources. (b) Hybrid colonies with multiple food sources and obstacles

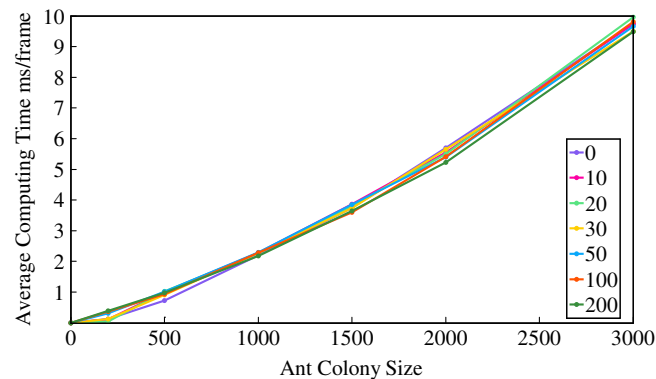


FIGURE 9 Comparison of the average computing costs with different numbers of obstacles and different colony sizes. The legends under the diagram represent different numbers of the obstacles

compare the simulation results without obstacle (Figures 7a and 8a) and with obstacles (Figures 7b and 8b) in the single-colony and hybrid-colony motion scenarios. In Figures 7 and 8, each example of the two groups of simulation results abstractly shows the same moving process: Ants start from their belonging nest and gradually find a food source through the combination of random exploration and information transfer, and finally build a connected trail between the food source and their belonging nest for food finding and transporting. Comparing the generated path between the food source and the nest in our result with the (shortest) straight line segment between the two, we found that the ants in our algorithm can autonomously generate a relatively short foraging-transporting path between the food source and the nest. We set a nest, a food source, and different numbers of obstacles in the simulation scenario and calculated the average computing costs for different scales of ant colonies. As Figure 9 shows, the computational complexity of our algorithm is approximately linear to the size of the simulated ant colony. From Figure 9, we can also find that our method is very efficient because it can update a frame of simulating 3000 ants in 10 ms. From the same figure, we also find that the computing time is not seriously affected by different setups of motion scenarios because the average computing time of the simulations with the same colony size but different numbers of obstacles is similar.



FIGURE 10 Comparison on the trails between our method and the traffic trail model³⁸ with 300 ants. The first row is the trail generated by our method and the second row is the trail generated from the traffic trail model

We also compared our method with the traffic trail model,³⁸ as shown in Figure 10. The traffic trail model regards ant individuals as self-propelled particles and simulates an ant trail with the absence of jamming, and the trail is constructed through the physical forces for acceleration. By contrast, our motion control model considers virtual ants as autonomous agents and they can autonomously construct trails through information transfer. Besides, the trails in our simulation are curvy and bidirectional, compared with the predefined straight and unidirectional trail in the traffic trail model. Obviously, the trails resulted from our method are more realistic and closer to what can be observed in the real world.

5.1 | Outspread experiments

We also did some outspread experiments to test the controllability and flexibility of our motion control model in a variety of scenarios. To test whether our method can effectively adapt to dynamic changes in an environment, we add in-time changes on the connected ant trail between the nest and the food source.

Adaptation to dynamic changes in an environment. The tested dynamic changes in an environment include adding and removing obstacles, repositioning the food source, and adding external disturbances. Without the loss of generality, in our experiments, we only set one food source and one nest in the motion scenarios, and an ant colony with 500 ants was tested in our experiments.

We performed a series of experiments on dynamic obstacle changes. As shown in Figure 11, we interactively added three rocks on the trail and ants responded by constructing a new trail that is relatively near to the old one; then, we gradually removed the obstacles and the trail gradually went back to the original one. We also show the simulation results from the dynamic replacement of the food source (Figure 12). In this experiment, we replaced the food source; as a response, the ants dispersed to seek for the food source; and finally, the ants rebuilt a trail between the nest and the food source again. Furthermore, we have experimented to add external disturbances in real-time simulation. In Figure 13, we added an external disturbance on the trail, and then, the ants escaped to rebuild a new trail. The original trail was recovered after the removal of the external disturbance.

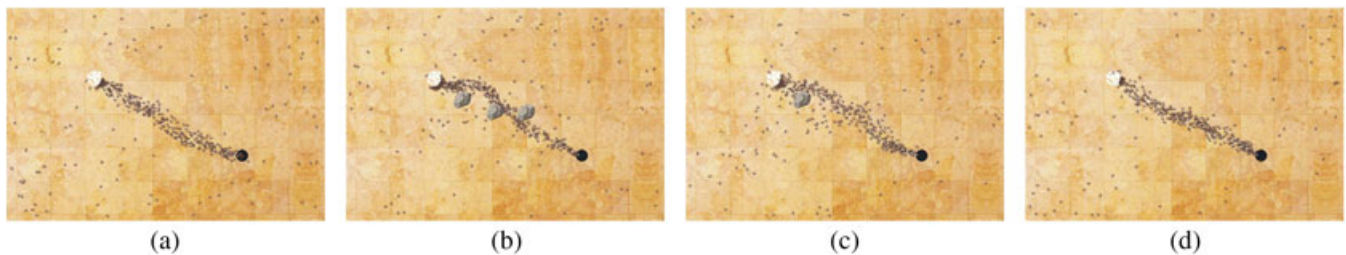


FIGURE 11 Results of adaptation to the dynamic changes of obstacles in environment. The size of the ant colony is 500. The figure illustrates the changes of the trail before and after adding obstacles and removing obstacles. (a) Original trail. (b) The reconstructed trail after putting three obstacles. (c) The reconstructed trail after removing two obstacles. (d) The reconstructed trail after removing all obstacles

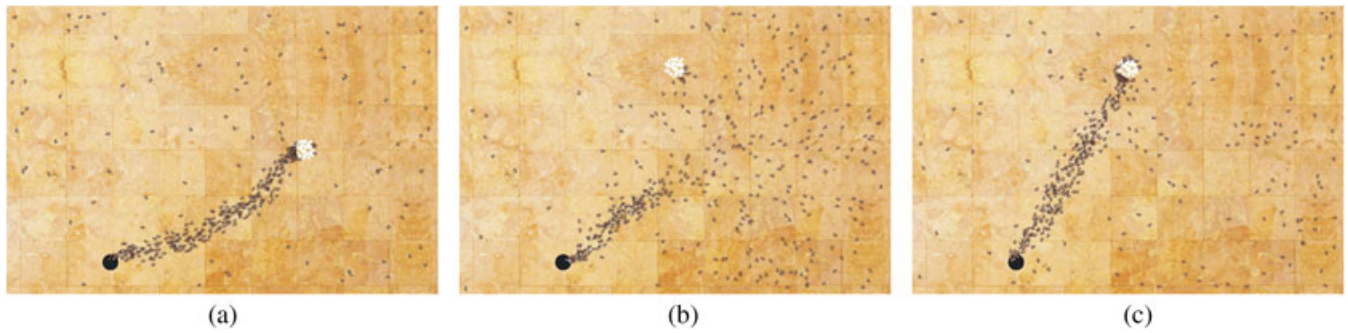


FIGURE 12 Adaptation to the repositioning of the food source. In this experiment, the colony has 500 ants. (a) Original trail. (b) Repositioning the food source. (c) The reconstructed trail

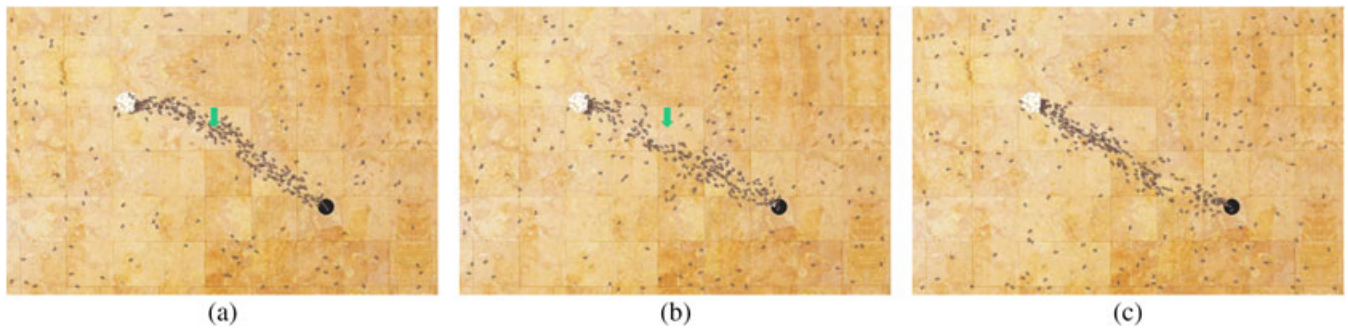


FIGURE 13 Adaptation to an external disturbance. 500 ants are simulated in this experiment, and the green vertical arrow here directs to the center of the dangerous area. (a) Put an external disturbance on the original trail. (b) The reconstructed trail under the disturbance. (c) The reconstructed trail after the disturbance

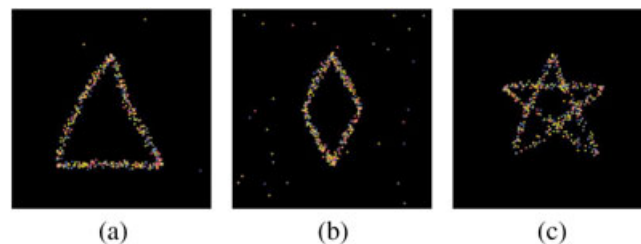


FIGURE 14 Results of group shape constraint by setting up way points with 500 individuals. (a) Three way points. (b) Four way points. (c) Five way points

In sum, our experiments show that our method is effective to handle dynamic changes in environment, and the connected path between the food source and the nest can be rebuilt as a relatively optimal path for foraging.

Group shape constraint. Group shape constraints have been studied before,^{40–42} but these models control the movements of individuals mechanically, besides collision avoidances. Directly applying such models for ants would fail to produce realistic ant formation animations. Taking advantage of the variety of mechanisms in our motion controller, we can extend our approach to handle user-specified group shape constraint and produce natural group shape formations. Two examples are described below.

In Figure 1, obstacles are used to constrain the shape of an ant colony with 500 ants. Two ways can be used to realize this work: the first one is to collocate the obstacles before the ants run out of the nest; hence, the ants can only move in the fixed zone. The other is to add rocks on the generated trail to decentralize the ants and restrain them in a fixed area.

Not limited to constrain shapes of ant colonies, Figure 14 shows the results by constraining the shape of a group through way points. The group changes its shape under the control of our motion controller, and the way points act as intermediate transit destinations. The individuals in the group tend to move toward a target way point through information transfer.



FIGURE 15 An interactive simulator built upon our motion controller. The ant colony is controlled in real time by changing the motion scenario or modifying parameters of the ant colony

Real-time simulator. Because our method can adapt to efficiently dynamic changes in environment, it can work in real time. We develop an interactive simulator to simulate collective ant colonies, as shown in Figure 15. In this simulator, users can change the environment by adding or removing food sources, obstacles, and nests in real time, and the users can also disturb the moving ants. Users are also allowed to adjust various parameters of our approach, including the colony size and the moving speed of the ants. For animation results, please refer to the supplementary demo video.

6 | DISCUSSION AND CONCLUSIONS

In this paper, we present a unified biologically inspired approach to simulate ant colonies. Specifically, we utilize motion states to distinguish different motion situations, including basic movement, interactions, and stop. Besides, we also set several constraint rules in the motion states. Thus, through flexible combinations of states and constraints, we can simulate a variety of ant colony behaviors, including seeking food, going back to the nest, avoiding collisions, escaping from external disturbances, and information transfer. In addition, our method offers flexible controls to users through real-time parameter changes or environment dynamic changes.

In our experiments, we simulated *foraging*, *avoiding*, and *escaping* behaviors of ants in a relatively simple environment. However, the actual environment can be more complex and the ants may conduct much more varieties of behaviors, for example, collectively transforming food, adapting to terrain changes, and resisting natural enemies. These behaviors can be simulated by straightforwardly appending new rules into our motion controller framework.

Our method can be potentially applied in other research fields. The motion control framework can be used to manipulate microrobots or nanorobots, drone groups, and other biological organisms that perform collaborative tasks. For example, the information transfer mechanism may facilitate collective transportations for microrobots or nanorobots, and the motion control framework may manipulate groups of drones into various formations for entertainment or military use.

Due to the complex living environment and the large size and high density of some ant colonies, it is very difficult to obtain comprehensive, accurate movement data of ants in the real world. As one of our future works, we plan to collect some data sets to develop data-driven schemes to simulate more realistic ant colonies. Such data sets would allow us to quantitatively evaluate the realism and accuracy of our motion controller. In addition, our motion controller can be potentially extended to simulate group behaviors and visual effects of other animals or insects, not limited to ants. For example, group shape morphing is another future research direction, building upon our current work.

ACKNOWLEDGEMENT

Xiaogang Jin was supported by the National Key R&D Program of China (Grant 2017YFB1002600).

ORCID

Xiaogang Jin  <https://orcid.org/0000-0001-7339-2920>

REFERENCES

1. Collins RJ, Jefferson D. Antfarm: Towards simulated evolution. Los Angeles, CA: Computer Science Department, University of California; 1990.
2. Sauter JA, Matthews R, van Dyke PH, Brueckner S. Evolving adaptive pheromone path planning mechanisms. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1; 2002 Jul 15–19; Bologna, Italy. New York, NY: ACM; 2002. p. 434–440.
3. Panait L, Luke S. A pheromone-based utility model for collaborative foraging. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems; 2004 Jul 23; New York, NY. Los Alamitos, CA: IEEE Computer Society; 2004. p. 36–43.
4. John A, Schadschneider A, Chowdhury D, Nishinari K. Characteristics of ant-inspired traffic flow. *Swarm Intell.* 2008;2(1):25–41.
5. Guo S, Chang J, Cao Y, Zhang J. A novel locomotion synthesis and optimisation framework for insects. *Comput Graph.* 2014;38:78–85.
6. Guo S, Chang J, Yang X, Wang W, Zhang J. Locomotion skills for insects with sample-based controller. *Comput Graph Forum.* 2014;33:31–40.
7. Guo S, Wang M, Notman G, Chang J, Zhang J, Liao M. Simulating collective transport of virtual ants. *Comput Anim Virtual Worlds.* 2017;28(3-4).
8. Hölldobler B, Wilson EO. The ants. Cambridge, MA: Harvard University Press; 1990.
9. Jackson DE, Ratnieks FLW. Communication in ants. *Current Biology.* 2006;16(15):R570–R574.
10. Fourcassié V, Dussutour A, Deneubourg J-L. Ant traffic rules. *J Exp Biol.* 2010;213(14):2357–2363.
11. Reynolds CW. Flocks, herds and schools: a distributed behavioral model. *ACM SIGGR Comput Graph.* 1987;21(4):25–34.
12. Helbing D, Molnár P. Social force model for pedestrian dynamics. *Phys Rev E.* 1995;51(5):4282.
13. Musse SR, Thalmann D. Hierarchical model for real time simulation of virtual human crowds. *IEEE Trans Vis Comput Graph.* 2001;7(2):152–164.
14. Pelechano N, Allbeck JM, Badler NI. Controlling individual agents in high-density crowd simulation. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 2007 Aug 2–4; San Diego, CA. Aire-la-Ville, Switzerland: Eurographics Association; 2007. p. 99–108.
15. Karamouzas I, Heil P, van Beek P, Overmars MH. A predictive collision avoidance model for pedestrian simulation. In: Motion in games: Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21–24, 2009 Proceedings. Vol 9; Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2009. p. 41–52.
16. Hildenbrandt H, Carere C, Hemelrijk CK. Self-organized aerial displays of thousands of starlings: a model. *Behav Ecol.* 2010;21(6):1349–1359.
17. Ondřej J, Pettré J, Olivier A-H, Donikian S. A synthetic-vision based steering approach for crowd simulation. *ACM Trans Graph.* 2010;29. Article No. 123.
18. Hughes R, Ondřej J, Dingliana J. Davis: density-adaptive synthetic-vision based steering for virtual crowds. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games; 2015; Paris, France. New York, NY: ACM; 2015.
19. Dutra TB, Marques R, Cavalcante-Neto JB, Vidal CA, Pettré J. Gradient-based steering for vision-based crowd simulation algorithms. *Comput Graph Forum.* 2017;36:337–348.
20. Karamouzas I, Sohre N, Narain R, Guy SJ. Implicit crowds: optimization integrator for robust crowd simulation. *ACM Trans Graph.* 2017;36(4). Article No. 136.
21. Treuille A, Cooper S, Popović Z. Continuum crowds. *ACM Trans Graph.* 2006;25:1160–1168.
22. Narain R, Golas A, Curtis S, Lin MC. Aggregate dynamics for dense crowd simulation. *ACM Trans Graph.* 2009;28. Article No. 122.
23. Toner J, Tu Y. Long-range order in a two-dimensional dynamical XY model: how birds fly together. *Phys Rev Lett.* 1995;75(23):4326.
24. Burstedde C, Klauck K, Schadschneider A, Zittartz J. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Phys A: Stat Mech Appl.* 2001;295(3):507–525.
25. Wang X, Jin X, Deng Z, Zhou L. Inherent noise-aware insect swarm simulation. *Comput Graph Forum.* 2014;33:51–62.
26. Lee KH, Choi MG, Hong Q, Lee J. Group behavior from video: a data-driven approach to crowd simulation. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 2007 Aug 2–4; San Diego, CA. Aire-la-Ville, Switzerland: Eurographics Association; 2007. p. 109–118.
27. Li W, Wolinski D, Pettré J, Lin MC. Biologically-inspired visual simulation of insect swarms. *Comput Graph Forum.* 2015;34:425–434.
28. Lukeman R, Li Y-X, Edelstein-Keshet L. Inferring individual rules from collective behavior. *Proc Natl Acad Sci.* 2010;107(28):12576–12580.
29. Ju E, Choi MG, Park M, Lee J, Lee KH, Takahashi S. Morphable crowds. *ACM Trans Graph.* 2010;29. Article No. 140.
30. Li Y, Christie M, Siret O, Kulpa R, Pettré J. Cloning crowd motions. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 2012 Jul 29–31; Lausanne, Switzerland. Aire-la-Ville, Switzerland: Eurographics Association; 2012. p. 201–210.
31. Gu Q, Deng Z. Context-aware motion diversification for crowd simulation. *IEEE Comput Graph Appl.* 2011;31(5):54–65.
32. Wang X, Ren J, Jin X, Manocha D. BSwarm: biologically-plausible dynamics model of insect swarms. In: Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 2015; Los Angeles, CA. New York, NY: ACM; 2015. p. 111–118.
33. Ren J, Wang X, Jin X, Manocha D. Simulating flying insects using dynamics and data-driven noise modeling to generate diverse collective behaviors. *PloS one.* 2016;11(5):e0155698.
34. Chowdhury D, Guttal V, Nishinari K, Schadschneider A. A cellular-automata model of flow in ant trails: non-monotonic variation of speed with density. *J Phys A: Math Gen.* 2002;35(41):L573.

35. Nishinari K, Sugawara K, Kazama T, Schadschneider A, Chowdhury D. Modelling of self-driven particles: foraging ants and pedestrians. *Phys A: Stat Mech Appl*. 2006;372(1):132–141.
36. Li L, Peng H, Kurths J, Yang Y, Schellnhuber HJ. Chaos–order transition in foraging behavior of ants. *Proc Natl Acad Sci*. 2014;111(23):8392–8397.
37. Amorim P. Modeling ant foraging: a chemotaxis approach with pheromones and trail formation. *J Theor Biol*. 2015;385:160–173.
38. Chaudhuri D, Nagar A. Absence of jamming in ant trails: feedback control of self-propulsion and noise. *Phys Rev E*. 2015;91(1):012706.
39. Reynolds CW. Steering behaviors for autonomous characters. In: *Proceedings of the Game Developers Conference*; 1999 Mar 23–27; San Francisco, CA. p. 763–782.
40. Gu Q, Deng Z. Formation sketching: an approach to stylize groups in crowd simulation. In: *Proceedings of Graphics Interface*; 2011; St. John's, Canada. Waterloo, Canada: University of Waterloo; 2011. p. 1–8.
41. Gu Q, Deng Z. Generating freestyle group formations in agent-based crowd simulations. *IEEE Comput Graph Appl*. 2013;33(1):20–31.
42. Xu M, Wu Y, Ye Y, Farkas I, Jiang H, Deng Z. Collective crowd formation transform with mutual information based runtime feedback. *Comput Graph Forum*. 2015;34(1):60–73.

AUTHOR BIOGRAPHIES



Wei Xiang is a Ph.D candidate in the State Key Laboratory of CAD&CG, Zhejiang University, China. Prior that, she received her BSc degree in computer science and technology in 2016 from Shandong University, China. Her main research interests include insect swarm animation and crowd animation.



Jiaping Ren is a Ph.D candidate in the State Key Laboratory of CAD&CG, Zhejiang University, China. She received her BSc degree from the Department of Mathematics at Zhejiang University of Technology, China, in 2013. Her main research interests include crowd animation, insect swarm animation, and collective behaviors of bird flocks.



Kuan Wang received her BSc degree from Zhejiang University, China, in 2016, majoring in digital media technology. Currently, she is in the State Key Laboratory of CAD&CG, Zhejiang University, as a Master's student. Her main research interests include virtual try-on and real-time rendering.



Zhigang Deng is a full professor at the Department of Computer Science at the University of Houston. His research interests include computer graphics, computer animation, virtual human modeling and animation, and human–computer interaction. He earned his PhD in computer science at the Department of Computer Science at the University of Southern California in 2006. Prior that, he also completed his BS degree in mathematics from Xiamen University (China), and his MSc in computer science from Peking University (China). Besides the CASA 2014 general co-chair and SCA 2015 general co-chair, he currently serves as an associate editor of *Computer Graphics Forum* and *Computer Animation and Virtual Worlds Journal*.



Xiaogang Jin received his BSc degree in computer science and his MSc and PhD degrees in applied mathematics from Zhejiang University, China, in 1989, 1992, and 1995, respectively. He is a professor in the State Key Laboratory of CAD&CG, Zhejiang University. His current research interests include digital geometry processing, geometric modeling, 3D printing, virtual try-on, insect swarm simulation, traffic simulation, implicit surface modeling and applications, creative modeling, sketch-based modeling, and image processing. He received an ACM Recognition of Service Award in 2015. He is a member of the IEEE and ACM.

SUPPORTING INFORMATION

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

How to cite this article: Xiang W, Ren J, Wang K, Deng Z, Jin X. Biologically inspired ant colony simulation. *Comput Anim Virtual Worlds*. 2018;e1867. <https://doi.org/10.1002/cav.1867>