

# Shape-constrained flying insects animation

Qiang Chen<sup>1</sup> | Guoliang Luo<sup>1</sup> | Yang Tong<sup>1</sup> | Xiaogang Jin<sup>2</sup>  | Zhigang Deng<sup>3</sup> 

<sup>1</sup>VR & Interactive Tech. Inst., East China Jiaotong University, Nanchang, China

<sup>2</sup>State Key Lab of CAD & CG, Zhejiang University, Nanchang, China

<sup>3</sup>Computer Science Department, University of Houston, Houston, Texas

## Correspondence

Guoliang Luo, Virtual Reality and Interactive Techniques Institute, East China Jiaotong University, Nanchang, Jiangxi, China.

Email: luoguoliang@ecjtu.edu.cn

Zhigang Deng, Computer Science Department University of Houston, Houston, Texas.

Email: zdeng4@uh.edu

## Funding information

National Natural Science Foundation of China, Grant/Award Number: 61602222 and 61732015; Natural Science Foundation of Jiangxi Province, Grant/Award Number: 20171BAB212011; Key Research and Development Program of Zhejiang Province, Grant/Award Number: 2018C01090; Key Research and Development Program of Jiangxi Province, Grant/Award Number: 2018BBE50024; Innovation Fund Designated for Graduate Students of Jiangxi Province, Grant/Award Number: YC2018-B072

## Abstract

During the past decades, high-fidelity realistic simulations of various flying insects exhibiting collective behavior have been broadly used in entertainment industries and virtual reality applications. However, due to the intrinsic complexity and high computational cost, shape constrained simulation of collective behaviors remains a challenging topic. In this paper, we present a robust multi-agent model for large-scale controllable shape constrained simulation of flying insects. Specifically, we design an internal force model to biologically mimic an individual insect. We also propose an external force model based on a trade-off mechanic to guide the insects smoothly deforming into a target shape. Our experimental results and comparative studies show our method is able to simulate realistic and dynamic flying insects with various user-specified shape constraints.

## KEYWORDS

curl-noise, distance force, flying insects, multi-agent, shape-constrained

## 1 | INTRODUCTION

The large amount of insects are important living creatures on our earth and many of them exhibit collective behaviors on a regular basis. Normally, these insects fly in a rapid, chaotic manner and behave similarly as a large group. Not surprisingly, flying insects presenting special clustered shapes and deforming in time has been increasingly used as a special effect for entertainment, simulation, and virtual reality applications.

Obviously, motion capture of a large set of insects in the real world is a daunting task, if not technically infeasible. Despite that, researches have attempted noticeable efforts to simulate certain collective behaviors of flying insects, including aggregation,<sup>1,2</sup> migration,<sup>3</sup> predation avoidance,<sup>4</sup> etc. However, to the best of our knowledge, to date relatively few efforts have been done to simulate constrained flying insects in the Computer Graphics community, e.g., imposing 3D shape constraints. To achieve this, the central task would be to design a robust multi-agent model, in which each agent is driven by various intrinsic forces while satisfying user-specified constraints. Due to the control complexity and the high

computational cost, simulating biologically-plausible, shape constrained, collective behaviors of flying insects remains a widely-open research problem to date.

In this paper, we propose a new force-based method to simulate visually-plausible, shape-constrained flying insects. Specifically, we extend a field-based method<sup>5</sup> to simulate the biologically-plausible collective behavior of each insect, and meanwhile the insects can rapidly form into a pre-defined shape through introduced external forces.

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

- a new method to generate flock morphing with pre-defined shape constraints while maximally preserving the biological-plausible dynamics of flying insects;
- a new way to generate user-defined flock shapes by applying distance forces to gather flying insects.

## 2 | RELATED WORK

Constrained animation techniques have been widely studied for the simulation of crowds, fluid and flock. In this section, we briefly summarize recent previous works on constrained animation techniques for crowd simulation, fluid simulation, and flock simulation.

**Constrained crowd simulation.** Guidance shape based methods are one of the most common techniques for constrained crowd simulation.<sup>6,7</sup> For instance, in the work of, Gu et al.<sup>8</sup> first sample a given shape, and then compute the correspondences between the sample points and the agents, finally a user-defined crowd shape can be obtained after each agent have arrived at a specific moving target position. The guidance mesh concept has also been exploited for group manipulation. For example, researchers achieve various crowd deformations using controllable meshes.<sup>9,10</sup> Recent works allow the users to manipulate group motions using an editable mesh.<sup>11,12</sup>

Despite the success of the above methods, they essentially work in 2D and may not be directly applicable for flying insects in the 3D space.

**Constrained fluid simulation.** A significant amount of research efforts have been conducted on constrained fluid simulation using guidance shapes to obtain more believable effects.<sup>13–15</sup> As one of the notable works in this direction, Hong et al. presented a method to control smoke shapes based on geometric potential fields.<sup>16</sup> McNamara et al. introduced an adjoint method to control the free surface of fluid, which can generate special fluid shape motion such as a running man.<sup>17</sup> Shi et al. presented a feedback force based method to dynamically adjust the discrepancy between the fluid and the dynamic target, which can generate fluid animations with user-defined shapes (e.g., a shape of running horse).<sup>18</sup> In the work of, Fattal and Lischinski<sup>19</sup> proposed an attractive force based method to gather smoke around a given target shape.

However, the above methods may share part of key ideas of our work, but can not be directly applicable for flying insect simulation, since flying agents have different dynamic behaviors, which requires adaptable functions to control the flock trajectory and shape.

**Constrained flock simulation.** Reynolds<sup>20</sup> was the first to develop a particle system to animate flock behaviors. Despite many excellent researching of flock simulation have been presented from then on, to the best of our knowledge, rare researchers have focused on shape-constrained flock animation. Anderson et al. presented several special shape constraints for bird motions in.<sup>21</sup> Moreover, Klotzman and et al recommend a solution to generate bird animations with special shapes (e.g., a V shape).<sup>22</sup> As for flying insects, Xu et al. delivered a flock animation method that can ensure the animation to form specific flock shapes.<sup>23</sup> Wang et al.<sup>24</sup> designed a curl-noise based model to simulate the constrained flock morphing of the global flock trajectory.

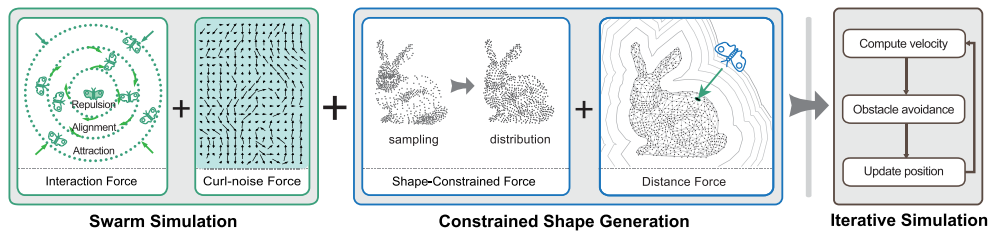
However, it is noteworthy that the insects in the above methods show minimal dynamic behaviors after they have arrived at the target positions, which substantially reduces the quality of the visual effects.

## 3 | OUR APPROACH OVERVIEW

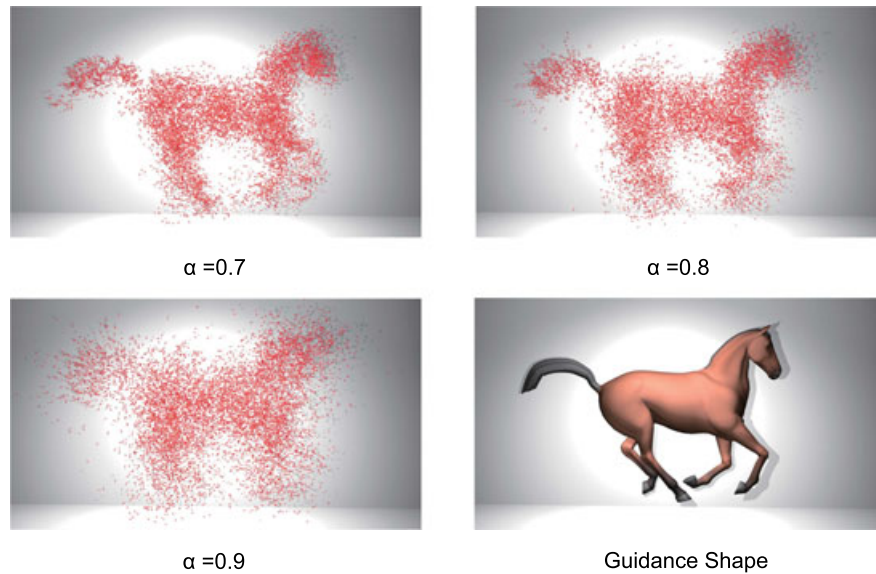
The pipeline of our approach is depicted in Figure 1, which contains the following main parts:

Part I. **Swarms simulation.** We design an *Internal force*,  $\vec{F}_{internal}$ , by combining the *Interaction force* and the *Curl-noise force* to simulate the chaotic behaviors of flying insects.

Part II. **Constrained shape generation.** First, we apply a stratified triangle sampling method to obtain sample points over the guidance target mesh. Then, we present a *shape constrained force* to optimally distribute the insects on the target mesh. Lastly, we design a *distance force* to attract the insects flying towards the sample points. By combining the shape constrained force and the distance force as an *external force*,  $\vec{F}_{external}$ , we can preserve the shapes of special flock.



**FIGURE 1** Overview of our shape-constrained flying insects model. The Swarm simulation is the weighted combination of the Interaction force (including three forces within three zones, i.e., repulsion, alignment and attraction) and the Curl-noise force; The Constrained shape generation is driven by both the Shape constrained force and Distance force. Then we iteratively update the positions of insects at each time step to obtain user-specified flock shapes



**FIGURE 2** Simulation of one key-frame of a running horse (10k agents) with different  $\alpha = 0.7, 0.8, 0.9$  values

Part III. **Constrained animation.** According to Newton's second law, we iteratively compute the acceleration of each insect based on both the internal and the external forces. Specifically, we update the insects' velocities at each time step with the consideration of obstacles when following the user-specified global migration path (see Section 6).

In sum, any agent in our model is driven by a union force  $\vec{F}_i$  and the proposed flying insect simulation model can be described by the following equation:

$$\vec{F}_i = \alpha \vec{F}_{i, \text{internal}} + (1 - \alpha) \vec{F}_{i, \text{external}}. \quad (1)$$

where  $\alpha$  is a controlling parameter used to balance the degree of the dynamics of the flying insects and the constraint of flock shape, through directly weighting the internal force  $\vec{F}_{i, \text{internal}}$  and the external force  $\vec{F}_{i, \text{external}}$ . Figure 2 shows the agents' different dynamics with different value of  $\alpha$  values. In our experiments, we empirically set  $\alpha = 0.7$  to obtain better dynamics for agents within the shape constraint.

## 4 | SWARM SIMULATION

In this section, we describe the internal force, through the combination of an interaction force and a curl-noise force, for swarm simulation, and it can simultaneously achieve the simulation of individual activities and flock behaviors. The internal force  $\vec{F}_{i, \text{internal}}$  of the  $i$ -th agent includes the interaction force  $\vec{F}_{i, \text{int}}$  (see Section 4.1) and the curl-noise force  $\vec{F}_{i, \text{cur}}$  (see Section 4.2).

### 4.1 | Interaction Force

As illustrated in the left of Figure 1, the basic idea of our interaction force model is to sort animals by three spheres (similar to<sup>1</sup>), denoted by three radii  $R_{\text{rep}}, R_{\text{ali}}$  and  $R_{\text{att}}$  ( $0 < R_{\text{rep}} < R_{\text{ali}} < R_{\text{att}}$ ). Specifically,  $R_{\text{rep}}$  denotes the repulsion zone which

**TABLE 1** Parameters setting and simulation performances of our experiments

Scenarios	Agents	Max speed	Setting		Simulation FPS (CPU)
			Parameters	Values	
Moon-shape flying insects	5,000	0.75m/s	$R_{rep}$	0.3	71.45
	10,000		$R_{ali}$	10	38.17
Mammoth to shark	5,000		$R_{att}$	30	70.17
	10,000		$\omega_{rep}$	0.2	34.74
Running horse	5,000	-	$\omega_{ali}$	3	70.93
	10,000		$\omega_{att}$	13	
			$\omega_{dis}$	4	33.99
			$\eta$	0.125	
			20,000	$\rho$	0.32
	$\alpha$			0.7	

is the minimized private space for each agent;  $R_{ali}$  denotes the alignment region where the agents incline to align with their neighbors;  $R_{att}$  determines a attractive zone where the agents are aggregated.

The interaction force  $\vec{F}_{i,int}$  includes the repulsion force  $\vec{F}_{i,rep}$ , the alignment force  $\vec{F}_{i,ali}$ , and the attraction force  $\vec{F}_{i,ali}$ . The three forces are computed as follow<sup>5</sup>:

$$\vec{F}_{i,*} = \frac{\omega_*}{N_*} \sum_{j=1}^{N_*} \left( S(|r_{ij}|) \frac{\vec{r}_{ij}}{|r_{ij}|} + (1 - S(|r_{ij}|)^2) \frac{\vec{u}_{ij}}{|u_{ij}|} \right), \quad (2)$$

where  $\vec{F}_{i,*}$ ,  $*$   $\in$  ( $re$ ,  $ali$ ,  $att$ ), are the three types of forces for the  $i$ -th agent; the constant parameters  $\omega_*$  are the weights for the forces. Their values can be found in Table 1. And  $N_*$  denote the numbers of neighbors in the three different cases, respectively.  $\vec{r}_i$  is the  $i$ -th agent's 3D position, and  $\vec{r}_j$  is the 3D position of its  $j$ -th neighbor, and  $\vec{r}_{ij} = \vec{r}_j - \vec{r}_i$ .  $\vec{u}_{ij}$  is the velocity of the  $i$ -th agent,  $\vec{u}_i$ , subtracts that of its  $j$ -th neighbor,  $\vec{u}_j$ . Also, the value of  $S(|r_{ij}|)$  can be computed by below:

$$S(|r_{ij}|) = \begin{cases} 1, & R_{ali} \leq |r_{ij}| \leq R_{att}; \\ 0, & R_{rep} \leq |r_{ij}| < R_{ali}; \\ -1, & 0 \leq |r_{ij}| < R_{rep}. \end{cases} \quad (3)$$

## 4.2 | Curl-noise Force

With the forces defined in Section 4.1, we can simulate relatively low-density insects in 3D space. However, in high-density circumstances, this could lead to violent oscillation phenomena, in which insects may repeatably be pushed away and dragged back by the repulsion force and the attraction force. We found that applied the Curl-noise field can avoid this issue. Curl-noise firstly proposed by Bridson et al.<sup>25</sup> for fluid simulation.

Inspired by the above works, we extend the curl-noise as intrinsic force to drive the insects flow along the mesh surface, through which can avoid the oscillation phenomenon and generate more fluent visual effects. The curl-noise force can be computed using the following formula<sup>3</sup>:

$$\vec{F}_{i,cur} = \nabla \times \left( \left( p(s1) \left( \frac{\vec{r}_i}{\rho} \right), p(s2) \left( \frac{\vec{r}_i}{\rho} \right), p(s3) \left( \frac{\vec{r}_i}{\rho} \right) \right) * \eta \right), \quad (4)$$

where  $\vec{F}_{i,cur}$  is the curl-noise force of the  $i$ -th agent;  $p(s1)$ ,  $p(s2)$  and  $p(s3)$  are the values produced by the Perlin Noise function<sup>26</sup> with different seeds (i.e.,  $s1$ ,  $s2$  and  $s3$ ) at the current position  $\vec{r}_i$ ;  $\rho$  and  $\eta$  are the parameters to alter the density of the grid and the magnitude of the Perlin noise, respectively.

## 5 | CONSTRAINED SHAPE GENERATION

In this section, we describe our algorithm for the generation of user-specified flock shapes. First, we sample the inputted 3D shapes based on a stratified points sampling method (Section 5.1). Then, we design an external force to generate the specified flock shapes. The external force of the  $i$ -th agent,  $\vec{F}_{i,external}$ , includes a shape-constrained force,  $\vec{F}_{i,con}$ , to distribute

agents onto the mesh surface (Section 5.2); and a distance force,  $\vec{F}_{i,dis}$ , to attract insect flying toward the sampling point and flying around the target mesh faithfully (Section 5.3).

## 5.1 | 3D Shape Sampling

Without loss of generality, we assume the target shapes are 3D triangulated meshes. Especially, we defined the sequence of key-frames with the same topology when it is captured from animation motions.

Inspired by the work of,<sup>27</sup> we apply a stratified sampling method to obtain sample points from the guidance meshes. And, the stratified sampling method can better preserve geometric features between guidance sequences.<sup>28</sup>

Considering insects' size information, we need ensure the minimum distance between any two insects during animation. So, we appropriately scale the inputted guidance meshes to meet the minimum distance requirement and generate adequate sample points. The sample points can be defined as  $s_i$  ( $i = 1, 2, 3 \dots N$ ).

## 5.2 | Shape Constrained Force

In our method, we assume each sample point,  $s_i$ , on the guidance mesh has a spring-like force to attract and push away its neighboring points along the surface. We denote  $N_i$  as the number of neighbors, retained through a KD-Tree, of the  $i$ -th sample points that are within a Manhattan distance threshold from the point. In our experiments, we set this threshold to  $k\sqrt{\frac{A}{N}}$ , where  $A$  is the surface area of the mesh, and  $k$  is set to 2.0.

Inspired by the work of,<sup>29</sup> the shape constrained force in our work is defined as follows:

$$\vec{F}_{i,con} = \frac{1}{N_i} \sum_{j=1}^{N_i} \left( \frac{\vec{s}_i s_j}{|s_i s_j|} \cdot e^{-M_{ij}^2 / 2\sigma^2} \right), \quad (5)$$

where  $\vec{F}_{i,con}$  is the shape constrained force for the  $i$ -th sample point;  $s_j$  denotes the neighbor's position of the  $i$ -th sample point;  $M_{ij}$  is the Manhattan distance, which requires less computation without performance loss, compared to other distance metrics such as the Euclidean distance. Besides,  $e^{-M_{ij}^2 / 2\sigma^2}$  is the energy between the  $i$ -th sample point and its  $j$ -th neighbor.  $\sigma = \omega \frac{A}{N}$  is only used to control the fall-off of the constrained force.  $\omega$  is the weight of the influence of the neighbors. Increasing  $\omega$  will decrease the influence of the neighbors to the  $i$ -th sample point. In our experiments, we empirically set  $\omega$  to 0.9.

Typically, the distribution of the sample points can be solved at the preprocessing step. However, we observed that in the real-world flying insects tend to aggregate into a dense group, which may easily break the flock shape. Thus, we use  $\vec{F}_{i,con}$  to scatter the insects along the mesh during the constrained animation.

## 5.3 | Distance Force

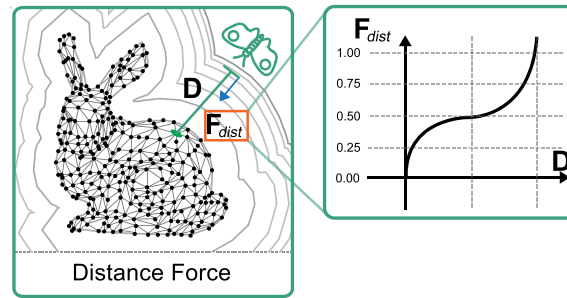
The objective of our method is to achieve certain flock shapes at specified frames while preserving intrinsic dynamic behaviors of the insects. To achieve this, after computing the guidance mesh, we compute the direct correspondences between the insects and the target positions (i.e., the sampled points). Then, we apply the distance force to attract the insects flying towards the target positions.

Theoretically, the distance force will disappear when an insect arrives at its destination, or loses its influence when the insect flies around the target position within a short distance. At this point, the internal force will dominate the dynamic behaviors of the insect. Then, while the escaping distance is increased due to the internal force, the attractive force will also be recovered and continue attracting the insect. Based on this consideration, we design a harmonic attractive force as a function of the distance force  $\vec{F}_{i,dis}$  (illustrated in Figure 3), defined as follows:

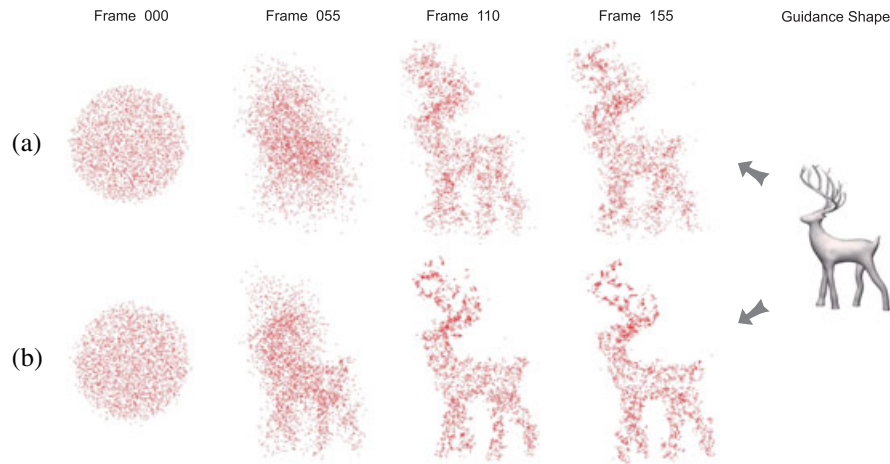
$$\vec{F}_{i,dis} = \omega_{dis} \cdot H(d) \cdot (4 * (d - 0.5)^{1/3} + 0.5) \cdot \left( \frac{\vec{r}_i \vec{s}_i}{d} \right), \quad (6)$$

where  $d$  denotes the Euclidean distance between the  $i$ -th agent's position  $\vec{r}_i$  and its corresponding destination  $\vec{s}_i$ .  $\omega_{dis}$  is the weight to adjust the influence of the distance force. We can obtain more stable flock shapes by increasing the value





**FIGURE 3** Illustration distance field.  $D$  denotes the Euclidean distance between the insect's position and the attractive vertex.  $F_{dist}$  is the distance force, which gradually declines to 0 from the position of the insect to the destination



**FIGURE 4** Insects deform into a deer shape with different values of  $\omega_{dist}$  (top:  $\omega_{dist} = 4$ ; bottom:  $\omega_{dist} = 8$ ). Increasing the value of  $\omega_{dist}$  can obtain more stable flock shapes and gradually limit the movements of the insects

of  $\omega_{dis}$ , but a too large  $\omega_{dis}$  will limit the movements of the insects (as shown in the bottom of Figure 4). In most of our experiments,  $\omega_{dis}$  is set to 4. In addition, with the normalized distance, i.e.,  $0 < d \leq 1$ , the value of  $H(d)$  can be computed as follows:

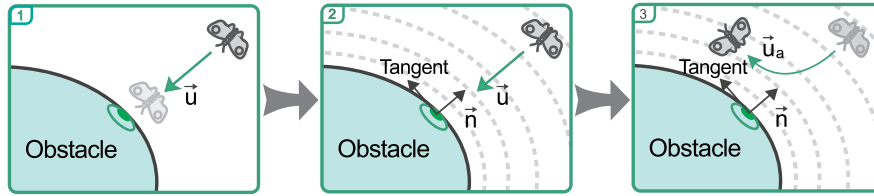
$$H(d) = \begin{cases} -1, & 0 < d \leq 1; \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

## 6 | PATH PLANNING AND COLLISION AVOIDANCE

With the aforementioned internal force and external force, our approach can automatically simulate the constrained flock animation without user interventions, except providing 3D meshes as guidance shapes at key-frames. In specific, to update the velocities of the insects in the process of the constrained flock simulation, we compute their accelerations based on Newton's second law and calculate the updated velocities with the time interval  $d_t$ . Moreover, since different flying insects have different flight speed ranges in the real world,<sup>30</sup> we set the max speed as 0.75 m/s in our experiments, which is approximately the flight speed of flies. Note that the max speed can also be replaced as a viscous drag force in our scheme. Additionally, we design the path planning and collision avoidance parts of our approach as follows.

**Global path planning.** In many applications, global migration paths of the flock will generate more artistic effects of the constrained animation. In our approach, we apply an arbitrary B-spline curve as the global path. Thus, through arc-length parameterization, we align the global positions of the flock with the curve.

**Collision avoidance.** In our method, each insect keeps a private space within the distance  $R_{rep}$ , because intruders will be pushed away by the repulsive force  $F_{rep}$ . However, the insects may tend to aggregate into a dense group under the attractive forces. This especially could happen to some animal guidance shapes with slim legs or horns,



**FIGURE 5** The process of collision avoidance in our model.  $\vec{n}$  is the normal on the boundary,  $\vec{u}$  is the actual velocity of the insect while  $\vec{u}_a$  is the avoidance velocity. The insect flies toward the obstacle with the velocity  $\vec{u}$  (left),  $\vec{u}$  smoothly ramps down towards the tangent (middle), and the actual velocity  $\vec{u}$  is replaced by the avoidance velocity  $\vec{u}_a$  (right)

such as a deer shape. In such cases, we adapt the minimum distance enforcement (MDE) method<sup>31</sup> to separate the two insects with their distance lower than a user-specified threshold. Therefore, our method can ensure local collision avoidance.

Furthermore, we apply the method proposed by Bridson et al.<sup>25</sup> to avoid obstacles for insects when moving along the global path. We assume the obstacles are stationary in the environment. To avoid an obstacle, the velocity vector of the insect,  $\vec{u}$ , needs to be tangent to the normal at the boundary of the obstacle,  $\vec{n}$ . In other words,  $\vec{u} \cdot \vec{n} = 0$ . Thus, the insects should smoothly slow down when approaching the obstacle (as shown in Figure 5). So, we replace the velocity  $\vec{u}$  with the obstacle avoidance velocity  $\vec{u}_a$ , defined below:

$$\vec{u}_a = \text{ramp}(\cdot) \cdot \vec{u} + (1 - \text{ramp}(\cdot))\vec{n}(\vec{n} \cdot \vec{u}), \quad (8)$$

where  $\text{ramp}(\cdot)$  is a ramp function with a variable,  $x = D_{\text{dist}}/L$ ;  $D_{\text{dist}}$  denotes the shortest distance from the insect to the obstacle; and  $L$  is used to alter the influence range. Also,  $\text{ramp}(\cdot)$  is defined as follows:

$$\text{ramp}(x) = \begin{cases} \frac{15}{8}x - \frac{10}{8}x^3 + \frac{3}{8}x^5, & x \leq 1; \\ 1, & \text{otherwise.} \end{cases} \quad (9)$$

## 7 | RESULTS AND COMPARISON

We implemented our method with C++ language and ran many simulations on an off-the-shelf PC with Intel(R) Core(TM) i7-7700 CPU and 16GB memory. We employed an efficient nearest neighbors search algorithm<sup>32</sup> to query the nearest neighbors of insects when computing the attractive force and shape-constrained force. We fixed the parameters for all the simulations and recorded the positions and directions of all the insects for rendering in Cinema 4D. We reported the performances of our simulation experiments in Table 1. As clearly shown in this table, our method is efficient and can achieve a real-time performance with 10,000 insects on an off-the-shelf computer. The animation results are enclosed in the supplemental demo video of this paper. The selected rendering scenarios are described below.

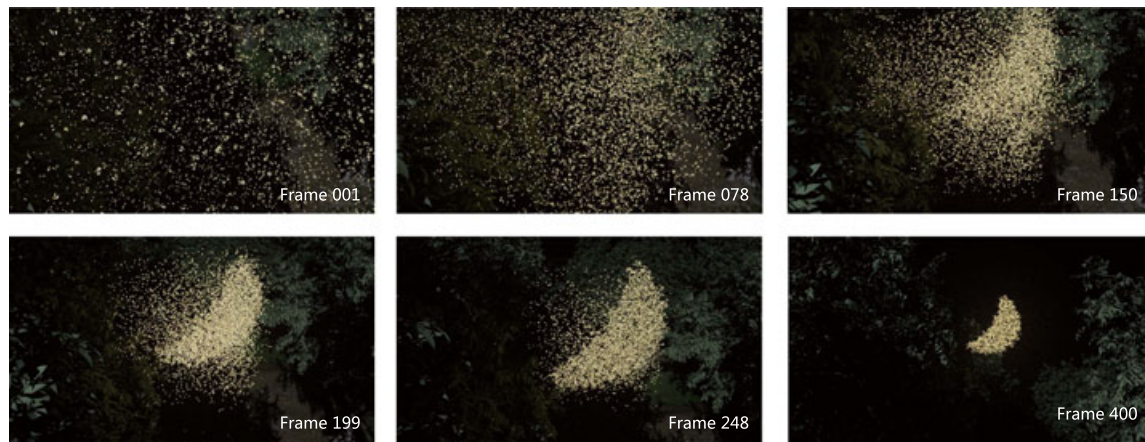
**Moon-shape flying insects:** In Figure 6, we show an animation where 10,000 flying insects are morphed into a moon shape with biologically-plausible motion in the night. At the start of the animation, the insects gradually fly towards the boundary of the pre-defined target and form a user-specified flock shape at the end. Each insect can fly chaotically and faithfully follow its target position during the animation, producing believable visual effects as a flock.

**From a mammoth shape to a shark shape:** In this experiment, we show the flock morphing from a mammoth shape to a shark shape in Figure 7. The results of our experiment indicate that the flock can fluently change their flock shape from an arbitrary shape to another. Also, our approach can obtain real-time performances when the total number of insects is no more than 10,000 (refer to Table 1).

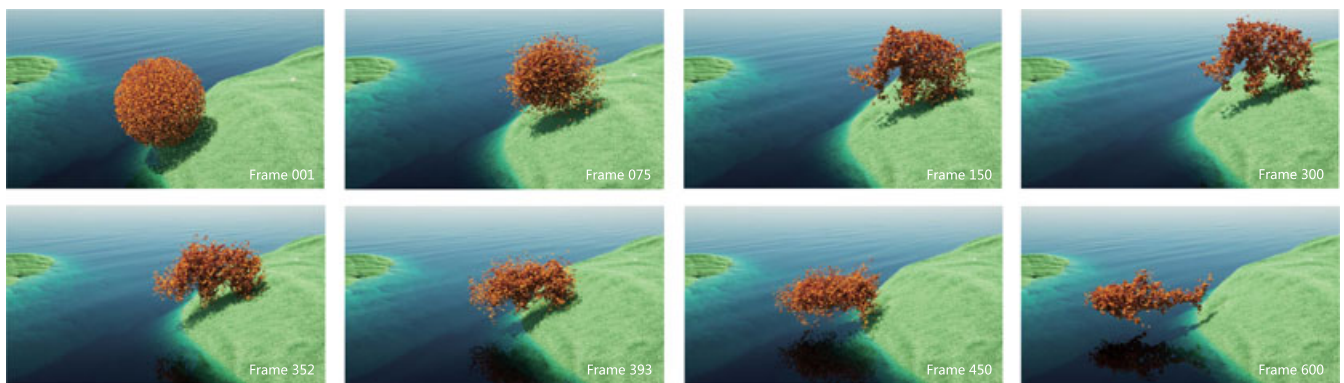
**A flock with a running horse shape:** Figure 8 shows a flock morphing along with a rapidly-changing motion target. A running horse, formed with 10,000 insects, follows a user-specified global trajectory. When the insects confront an obstacle, the flock shape will be affected temporarily, e.g., the insects avoid the obstacle like fluid. Such a morphing feature can be practically useful for certain visual effects applications.

Also, we simulated the simplified scenarios with different numbers of insects to compare their visual results and computational efficiencies (refer to Table 1). For more animation results, please refer to the supplemental demo video.

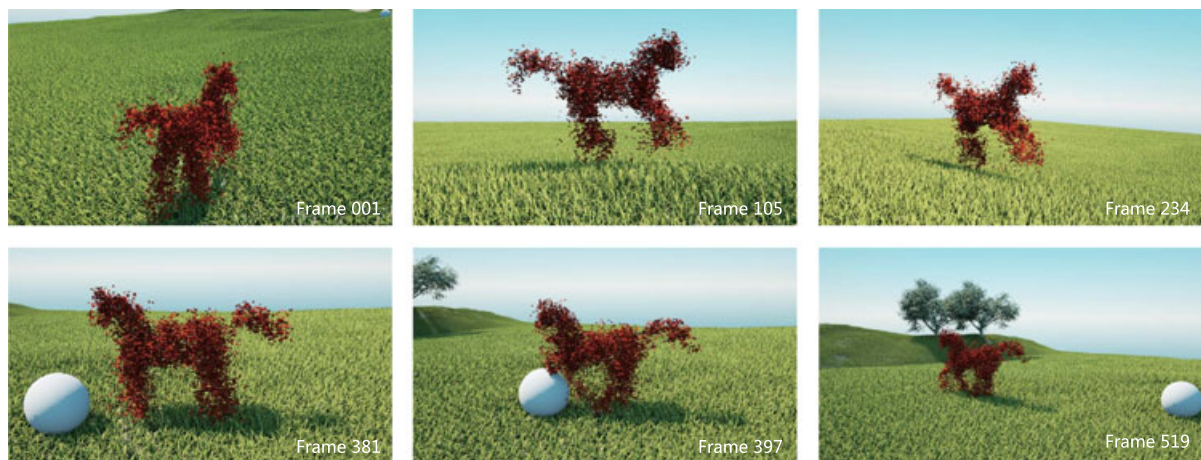
**Comparison with the baseline method:** We also compared our method with a baseline method.<sup>23</sup> The method in<sup>23</sup> can also obtain stable flock shapes, but the simulated insects are stucked at the boundary of the guidance mesh, without



**FIGURE 6** A flock deforms into a moon shape with biologically-plausible motion



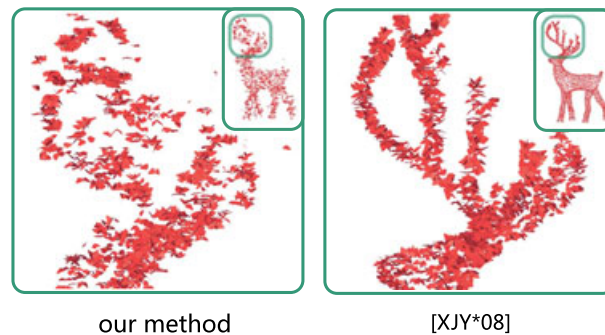
**FIGURE 7** A flock of 10,000 insects first deforms into a mammoth shape from a sphere(top), and then morphs into a shark shape (bottom)



**FIGURE 8** A running horse follows a global migrant path (top row) and avoids an obstacle in a “insects-fluid” manner (bottom row)

preserving their intrinsic motion dynamics. As shown in Figure 9, 4,000 agents form into a deer shape and the deer's horn was zoomed in for clearly presentation. Each insect simulated by our approach can still behave like a flying insect; on the contrary, some insects (in particular, those along the boundary) simulated by<sup>23</sup> fail to exhibit their dynamics.





**FIGURE 9** In the result by our approach (left), the insects exhibit dynamic behaviors. By contrast, in the result by,<sup>23</sup> the insects are stuck on the boundary of the guidance mesh (right), while the insects still flying chaotically without destroying the flock shape according our approach

## 8 | DISCUSSION AND CONCLUSION

In this paper, we present a new force-based method to simulate shape-constrained flying insects animation. Essentially, we apply internal forces to simulate the biologically-plausible collective behavior of each insect, and the insects can rapidly form into a pre-defined shape by applying external forces. Through various simulation experiments, we demonstrate that our method can guide large-scale flying insects so that they deform into arbitrary target shapes, and morph from one shape to another. Especially, our method can ensure flying insects to naturally follow the targets with rapid motion, while maximally preserving the dynamics of the insects. The computational cost of our method mainly depends on the number of the simulated insects. Our method is highly efficient; the performances of our experiments show that our method can be used for real-time simulations even for large-scale insects (e.g., more than 10k insects).

Our current method has two limitations: First, the morphing of flock shapes relies on user-provided guidance 3D models. An more intuitive way for specifying the guidance meshes would be preferred, such as a sketching interface. Second, there are many parameters in our multi-agent and force-based model. At present, we fix the appropriate setting values based on heuristic optimization methods. In the future, we tend to develop a benchmark to evaluate the visual quality of simulated insect animations, which will enable to explore an automatic parameters optimization method.

## ACKNOWLEDGEMENTS

This work has been in part supported by the National Natural Science Foundation of China (No.61602222, 61732015), the Natural Science Foundation of Jiangxi Province (No.20171B AB212011), the Key Research and Development Program of Zhejiang Province (No.2018 C01090), the Key Research and Development Program of Jiangxi Province (No.2018BBE5 0024), the Innovation Fund Designated for Graduate Students of Jiangxi Province (NO.YC 2018-B072).

## ORCID

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

Zhigang Deng  <https://orcid.org/0000-0003-2571-5865>

## REFERENCES

1. Couzin ID, Krause J, James R, Ruxton GD, Franks NR. Collective memory and spatial sorting in animal groups. *J Theor Biol.* 2002;218(1):1–11.
2. Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O. Novel type of phase transition in a system of self-driven particles. *Phys Rev Lett.* 1995;75(6):1226.
3. Wang X, Jin X, Deng Z, Zhou L. Inherent noise-aware insect swarm simulation. *Comput Graph Forum.* 2014;33(6):51–62.
4. Demšar J, Lebar Bajec I. Simulated predator attacks on flocks: a comparison of tactics. *Artificial Life.* 2014;20(3):343–359.
5. Wang X, Ren J, Jin X, Manocha D. BSwarm: biologically-plausible dynamics model of insect swarms. *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*; 2015 Aug 7–9; Los Angeles, CA. New York, NY: ACM; 2015. p. 111–118.
6. Takahashi S, Yoshida K, Kwon T, Lee KH, Lee J, Shin SY. Spectral-based group formation control. *Comput Graph Forum.* 2009;28(2):639–648.

7. Zhang P, Liu H, Ding Y-H. Crowd simulation based on constrained and controlled group formation. *Vis Comput.* 2015;31(1):5–18.
8. Gu Q, Deng Z. Formation sketching: an approach to stylize groups in crowd simulation. *Proceedings of Graphics Interface 2011*; 2011 May 25–27; St. John's, Canada. Waterloo, Canada: Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo; 2011. p. 1–8.
9. Henry J, Shum HPH, Komura T. Interactive formation control in complex environments. *IEEE Trans Vis Comput Graph.* 2014;20(2):211–222.
10. 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.
11. Kim J, Seol Y, Kwon T, Lee J. Interactive manipulation of large-scale crowd animation. *ACM Trans Graph.* 2014;33(4). Article No. 83.
12. Kwon T, Lee KH, Lee J, Takahashi S. Group motion editing. *ACM Trans Graph.* 2008;27(3). Article No. 80.
13. Shi L, Yu Y. Taming liquids for rapidly changing targets. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*; 2005 Jul 29–31; Los Angeles, CA. New York, NY: ACM; 2005. p. 229–236.
14. Madill J, Mould D. Target particle control of smoke simulation. *Proceedings of Graphics Interface 2013*; 2013 May 29–31; Regina, Canada. Toronto, Canada: Canadian Information Processing Society; 2013. p. 125–132.
15. Feng G, Liu S. Detail-preserving SPH fluid control with deformation constraints. *Comput Animat Virtual Worlds.* 2018;29(1):e1781.
16. Hong J, Kim C. Controlling fluid animation with geometric potential. *Comput Animat Virtual Worlds.* 2004;15(3-4):147–157.
17. McNamara A, Treuille A, Popović Z, Stam J. Fluid control using the adjoint method. *ACM Trans Graph.* 2004;23(3):449–456.
18. Shi L, Yu Y. Controllable smoke animation with guiding objects. *ACM Trans Graph.* 2005;24(1):140–164.
19. Fattal R, Lischinski D. Target-driven smoke animation. *ACM Trans Graph.* 2004;23(3):441–448.
20. Reynolds CW. Flocks, herds and schools: a distributed behavioral model. *ACM SIGGRAPH Comput Graph.* 1987;21(4):25–34.
21. Anderson M, McDaniel E, Cheney S. Constrained animation of flocks. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*; 2003 Jul 26–27; San Diego, CA. Aire-la-Ville, Switzerland: Eurographics Association; 2003. p. 286–297.
22. Klotzman M, Tal A. Animation of flocks flying in line formations. *Artificial Life.* 2011;18(1):91–105.
23. Xu J, Jin X, Yu Y, Shen T, Zhou M. Shape-constrained flock animation. *Comput Animat Virtual Worlds.* 2008;19(3-4):319–330.
24. Wang X, Zhou L, Deng Z, Jin X. Flock morphing animation. *Comput Animat Virtual Worlds.* 2014;25(3-4):351–360.
25. Bridson R, Hourihane J, Nordenstam M. Curl-noise for procedural fluid flow. *ACM Trans Graph.* 2007;26(3). Article No. 46.
26. Perlin K. Improving noise. *ACM Trans Graph.* 2002;21(3):681–682.
27. Alexa M, Rusinkiewicz S, Nehab D, Shilane P. Stratified point sampling of 3D models. *Proceedings of the First Eurographics Conference on Point-Based Graphics*; 2004 Jun 2–4; Zürich, Switzerland. Aire-la-Ville, Switzerland: Eurographics Association; 2004. p. 49–56.
28. Wand M, Straßer W. Multi-resolution rendering of complex animated scenes. *Comput Graph Forum.* 2002;21(3):483–491.
29. Lai Y-K, Hu S-M, Martin RR. Surface mosaics. *Vis Comput.* 2006;22(9-11):604–611.
30. Dudley R, Srygley RB, Oliveira EG, DeVries PJ. Flight speeds, lipid reserves, and predation of the migratory neotropical moth *Urania fulgens* (Uraniidae). *Biotropica.* 2002;34(3):452–458.
31. Treuille A, Cooper S, Popović Z. Continuum crowds. *ACM Trans Graph.* 2006;25(3):1160–1168.
32. Behley J, Steinhage V, Cremers AB. Efficient radius neighbor search in three-dimensional point clouds. Paper presented at: 2015 IEEE International Conference on Robotics and Automation (ICRA); 2015 May 26–30; Seattle, WA. Piscataway, NJ: IEEE; 2015. p. 3625–3630.

## AUTHOR BIOGRAPHIES



**Qiang Chen** is currently a PhD candidate at the Virtual Reality and Interactive Techniques Institute, East China Jiaotong University. His main research interests include insect swarm animation and dense crowd simulation.



**Guoliang Luo** earned his PhD in Computer Science at University of Strasbourg in 2015. His research interests include the Computer Graphics, Artificial Intelligence. He is currently an associate professor at East China Jiaotong University. He was enrolled in the Ganjiang Outstanding Youth Talent Program in 2018.



**Tong Yang** received her Master's degree in computer animation from Beijing University, China, in 2014. She is currently an experimenter at Virtual Reality and Interactive Techniques Institute, East China Jiaotong University. Her main research interests include creative modeling and computer animation.



**Xiaogang Jin** is a Professor of the State Key Lab of CAD&CG, Zhejiang University, China. He received his BSc degree in computer science in 1989 and his MSc and PhD degrees in applied mathematics in 1992 and 1995, respectively, all from Zhejiang University. His current research interests include traffic simulation, insect swarm simulation, physically based animation, cloth animation, special effects simulation, implicit surface computing, non-photo realistic rendering, computer-generated marbling, and digital geometry processing. He received an ACM Recognition of Service Award in 2015 and the Best Paper Award from CASA 2017 and CASA2018. He is a member of the IEEE and the ACM.



**Zhigang Deng** is currently a Full Professor of Computer Science at the University of Houston(UH) and the Founding Director of the UH Computer Graphics and Interactive Media (CGIM)Lab. His research interests include computer graphics, computer animation, virtual human modeling and animation, and human-computer interaction. He earned his PhD degree in computer science from 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 MS degree in computer science from Peking University, China. He was the recipient of a number of awards, including the ACM ICMI Ten-Year Technical Impact Award, the UH Teaching Excellence Award, the Google Faculty Research Award, the UHCS Faculty Academic Excellence Award, and the NSFC Overseas and Hong Kong/Macau Young Scholars Collaborative Research Award. Besides the CASA 2014 Conference General Co-chair and SCA 2015 Conference General Co-chair, he currently serves as an Associate Editor for several journals, including Computer Graphics Forum and the Computer Animation and Virtual Worlds Journal. He is a Senior Member of the ACM and a Senior Member of the IEEE.

## 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:** Chen Q, Luo G, Tong Y, Jin X, Deng Z. Shape-constrained flying insects animation. *Comput Anim Virtual Worlds*. 2019;30:e1902. <https://doi.org/10.1002/cav.1902>