Simulating Collective Transport of Virtual Ants

Abstract

This paper simulates the behaviour of *collective transport* where a group of ants transports an object in a *cooperative* fashion. Different from humans, the task coordination of collective transport, with ants, is not achieved by direct communication between group individuals, but through indirect information transmission via mechanical movements of the object. This paper proposes a *stochastic probability model* to model the decision-making procedure of group individuals and trains a neural network via reinforcement learning to represent the force policy. Our method is scalable to different numbers of individuals and is adaptable to users' input, including transport trajectory, object shape and external intervention etc. Our method can reproduce the characteristic strategies of ants, such as *realign* and *reposition*. The simulations show that with the strategy of *reposition*, the ants can avoid deadlock scenarios during the task of collective transport.

Keywords: Character Animation, Collective Transport

1 Introduction

Collective transport describes the behaviour of a group of ants collectively transporting a heavy prey, a task which would otherwise be impossible for a single individual to complete [1, 2]. This cooperative behaviour saves the effort of dissecting a large prey on site and increases the overall amount of food supplied [2, 3]. Natural-looking animations of this behaviour could greatly enhance the vividness and immersion in interactive applications. However simulating the collective transport of virtual ants is a challenging task since it involves a group of individuals coordinating in an indirect way. It is even more challenging if the animator demands flexible control over the number of individuals, the trajectory, obstacles and other inputs.

In spite of the aforementioned challenges, few attempts have been made to model this behaviour in the field of computer animation. This deficiency is in sharp contrast with the large collection of existing work on simulating the interaction between biped characters [4, 5, 6, 7, 8, 9] and that of swarm behaviour [10, 11, 12]. Collective behaviour in humans normally requires intensive information sharing between individuals, such as in collaborative or adversarial games. Compared to such behaviours in humans, the collective transport of ants is not achieved by direct communication among individuals, but through indirect information exchange via the environment. This process is known as *stigmergy* [13]. Most of the existing work in swarm simulation focuses on navigation and formulation of swarm individuals and does not address the specific problem of force coordination in a decentralised scenario.

In this paper, we present a model for simulating the behaviour of collective transport of virtual ants. The goal of this work is not only to reproduce the phenomenon of collective transport, but also to allow animators to author sophisticated behaviours. The contributions of this work include:

- A novel stochastic probability model is introduced to simulate the strategies of *realign* and *reposition*, as used by ants during prey transport. This stochastic probability model produces the visually-appealing random behaviour by adjusting the ants' body orientation and attachment position during the process of collective transport.

- A *stigmergy*-inspired force policy is proposed and modelled as a neural network.

The policy is further trained with the Q-learning method, a reinforcement learning technique, to optimise the weight parameters of the force policy network. With this force policy, characters can apply force to the object individually and successfully complete the task of collective transport without direct information from the others.

- We developed a complete framework to allow users to author the behaviour of collective transport. Our work is capable of scaling from two to a large number of individuals and can adapt to different scenarios based on user input of trajectories and prey weight etc. In the case of external intervention, individuals can reorganise themselves and restart the transport procedure.

The remainder of this paper is structured as follows. Section 2 surveys the existing work in related topics including multi-character interaction and swarm simulation. Section 3 describes the design of our framework. Section 4 presents the results generated from the proposed framework and discusses the limitations of our existing implementation. The last section, Section 5, concludes this paper by summarising and presenting directions for future research.

2 Related Work

2.1 Multi-character Interaction in Computer Animation

Recently there has been a surge of interest in modelling the interaction between multiple characters, in the field of computer animation [4, 5, 6]. Researchers initially focused on the interaction between two players by editing existing mocap data with an inverted pendulum model for each character [6], or by merging two existing interacting motion samples and automatically detecting the space-time relationship between them [5]. Game theory has been introduced to model the interaction of either collaborative or adversarial goals between two players [7, 8]. Recent work has expanded to scenarios involving more than two characters. Based on written or verbal descriptions of the action scenes, researchers are capable of generating, ranking and recommending a small set of interaction scenarios for multiple characters from a large number of scene candidates [9]. Inspired by language grammars, researchers introduced a symbolic description to represent the interaction amongst individuals [4]. This has successfully generated animations for a group of characters in scenarios such as basketball games, where rules, regulations and planning are critical.

The complexity of the strategies, used in the existing work, far outperforms the intelligence of insects and is computationally unnecessary. Our work specifically focuses on the task of collective transport of ants and develops tools to simulate such behaviour with sufficient control over the group size, movement trajectory and more.

2.2 Swarm Simulation

Swarm simulation deals with the problem of generating the animation of a group of individuals. Researchers introduced the concept of *navigation fields* to direct and control virtual crowds [14]. These *fields* can be generated via user sketches or 2D videos. Researchers have proposed interactive and scalable frameworks which generate freestyle group formations and transitions via natural and flexible sketching interaction [10, 11, 12]. Researchers have also proposed the control of sophisticated group formations via heuristic rules with explicit hard constraints [15]. However, users had to manually specify exact agent distributions, which was time-consuming and labour intensive if the crowd contained many agents. A recent work [12] is capable of generating group behaviours along with coherent and collision free navigation at interactive frame rates. Their method can also dynamically adapt to the environment and the number, shape, and size of the groups.

It is worth noting that there exists little work in the area of swarm simulation which addresses the specific problem as proposed in this work. The majority of existing work focuses on the distribution, navigation and formulation of swarm individuals. However, the main interest of our work is to coordinate the behaviour strategy and force policy of group individuals with indirect information sharing between them.

Another critical application for simulating collective transport is the field of *swarm robotics*. Tasks which are challenging for a single robot, with limited capabilities, can be conducted by a group of robots. This not only allows for the flexibility of adapting to different tasks with different numbers of robots but also increases the system's robustness with sufficient tolerance of individual failures [3]. To implement such a function, robots can be coordinated in either a centralised [16] or decentralised [17] fashion. A centralised structure guarantees the optimal solution but suffers from an exponentially scaling complexity with regards to the number of individuals. A decentralised structure leads to a sub-optimal result but is scalable to a varying number of group individuals. However, compared to animation research, this research in robotics does not consider the synthesis of full-body motion and prioritises stability over other factors.

3 Methodology

The behaviour engine defines the individual's collection of internal states and the rules for switching from one state to another. Intuitively, the behaviour engine is modelled as a Finite State Machine (FSM) (Figure 1a). A character has three states: *search*, *approach* and *transport*.

- **Search**. Characters are initialised at random positions in the scene. They individually search for the prey object by dynamically adjusting their movement direction. Characters can detect the existence of the prey if the distance is within a range of $2cm$ (based on the observations of the species *Pheidole crassinoda* [18]). Once the prey object enters their sensory range, they switch to the state of *approach*.

- **Approach**. The character will approach directly towards the prey object once it is detected. The state of approach will terminate if a collision between the geometric shape of the prey and character is detected. In this case, the character switches to the state of *transport*.

- **Transport**. Once connected, individuals determine how to apply force to move the prey given the mechanical feedback from the prey and other information (such as the desired trajectory). The transport state is subdivided into three strategies: *standard*, *realign* and *reposition*. During the *standard* sub-state, the character doest not adjust its relative position with respect to the object. Inspired by observations of real ants, we propose a *Stochastic Probability Model* to simulate two typical strategies which ants adopt for collective transport: *realign* and *reposition*.

When the character is in the state of either *search* or *approach*, we directly specify the velocity to manipulate the character's locomotion and synthesise the full-body animation based on a *Central Pattern Generator* control framework [19]. The following paragraphs explain the two main components of our work: the *Stochastic Probability Model* as part of the behaviour engine and the *force policy* to determine the dragging force when the individual is attached to the prey object.

3.1 Stochastic Probability Model

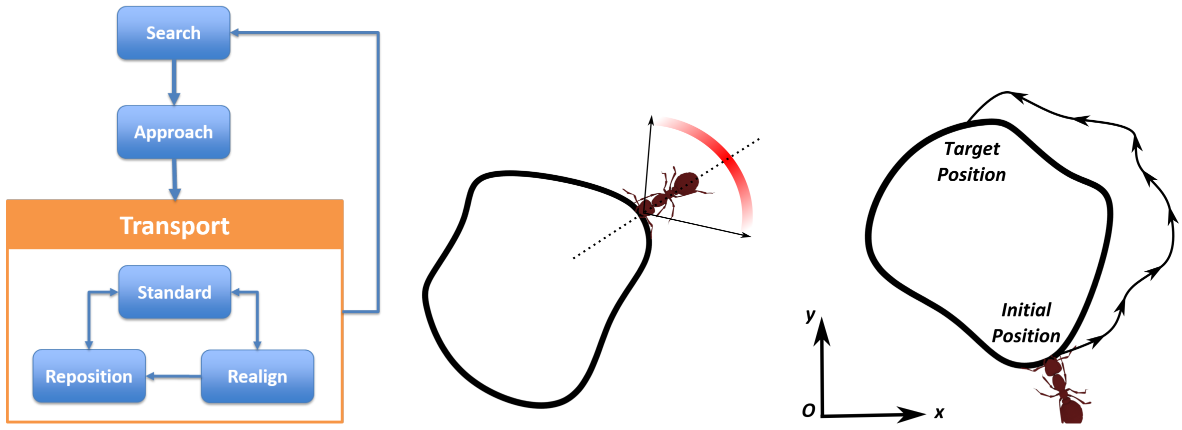
3.1.1 Realign

The strategy of *realign* alters the body orientation of the individual without releasing its hold of the prey object [18] (Figure 1b). The intuition is that the ant will attempt to align the object with its own orientation so that the ant can pull the object while walking backwards [1, 3]. When a single ant experiences difficulty in pulling the prey object, it attempts to pull from varying directions. The strategy of *realign* tends to occur before *reposition* and much more frequently than *reposition* [13].

Various factors, including object weight, surface friction and obstacle obstruction, can all contribute to the resistance which ants experience during transport and thus triggers the strategy of *realign*. Therefore, we choose the term of transport velocity as an abstraction of the prey movement. A score $P_{realign}$ is computed as:

$$P_{realign} = \frac{1}{1 + exp(0.5 - \frac{||\vec{\nu}_o||}{||\vec{\nu}_a||_{max}})} \quad (1)$$

where $\vec{\nu}_o$ is the velocity of the object. $||\vec{\nu}_a||_{max}$ is the maximum moving speed of the virtual ant.



(a) FSM (b) Realign (c) Reposition

Figure 1: (a) Finite State Machine of the behaviour engine. (b) the strategy of realign. (c) the strategy of reposition.

$exp()$ is the exponential function. This representation states that if the prey object moves at a slow speed, the character is more likely to perform the strategy of *realign*, attempting to accelerate the movement of the object by adjusting the force direction.

This score is compared against a stochastic threshold λ_a with a normal distribution $\lambda_a \sim N(\mu_a, \sigma_a)$. Parameters $\mu_a = 0.5, \sigma_a = 0.2$ ensure that the probability distribution between [0, 1] is greater than 98%.

When the character decides to realign its body and pulls from another direction, we compute the target angle θ:

$$\theta = N(\theta_{back}, \sigma_{body}) \tag{2}$$

where θ_{back} is the orientation when pulling backwards and σ_{body} is set to avoid a geometric collision with the prey object.

3.1.2 Reposition

If the individual still fails to move the prey after adjusting the pulling direction, it releases the attachment of the prey object, repositions itself at another attachment point and repeats the pulling process [18]. This process is called *reposition* (Figure 1c). A score $P_{reposition}$ is represented as following:

$$P_{reposition} = \frac{1}{1 + exp(\frac{t}{t_{max}} - \gamma)} \times \frac{1}{1 + exp(\frac{||\vec{\nu}_a||}{||\vec{\nu}_a||_{max}} - 0.5)} \tag{3}$$

where $\vec{\nu}_a$ is the movement velocity of the character. t, t_{max} are the elapsed time and the maximum time since the initialisation of current attachment.

$P_{reposition}$ is also compared against a stochastic threshold $\lambda_p \sim N(\mu_p, \sigma_p)$. Parameters μ_p, σ_p are set to the same values as μ_a, σ_a. If the probability is greater than the threshold, the character chooses to reposition itself, otherwise, it does not.

The target reposition location is computed as a random point along the exterior shape of the object, which is uniformly parameterised between [0, 1]. The movement trajectory $T(t)$ of reposition behaviour is computed as:

$$T(t) = C_o(t) + D(t, D_{min}, D_{max}) + C_c(t) \tag{4}$$

where $C_o(t)$ is the contour of the object shape in the world coordinate, $D(t)$ is a uniform random distribution between $[D_{min}, D_{max}]$, producing a displacement distance between the character and the object contour. $C_c(t)$ is a sub-level trajectory to avoid the potential collision with other individuals. Our current implementation produces $C_c(t)$ as a circular curve with a constant radius and with its centre at the location of the other individual to avoid; although other types of curves would also be suitable.

3.2 Force Policy

How the ants apply force to the prey object is a challenging task, given the absence of communication. We introduce a feed-forward neural

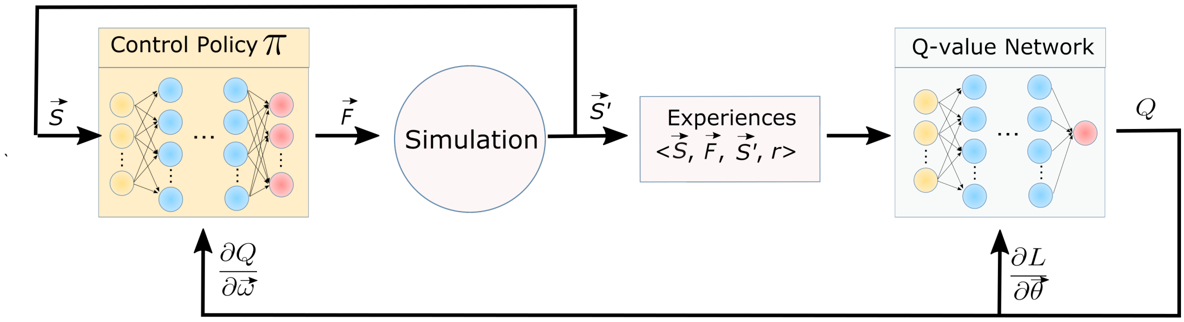


Figure 2: The framework of the force strategy. The control policy π first determines the force $\vec{F}$ based on the current state $\vec{S}$. A Q-value network evaluates the performance of the control policy with a reward function.

network to define a policy π, which determines the force $\vec{F}$ applied on the object to change the current state $\vec{S}$:

$$\pi(\vec{S}, \vec{\omega}) : \vec{S} \rightarrow \vec{F} \tag{5}$$

where $\vec{\omega}$ is the parameters of the decision network. The input $\vec{S} = (\vec{\nu}_a, \vec{\nu}_o, \vec{\nu}_o^*)$ includes the velocity of the individual ($\vec{\nu}_a$) and object ($\vec{\nu}_o$), and the desired transport velocity of object ($\vec{\nu}_o^*$).

We used the Q-learning method, a reinforcement learning algorithm, to optimise the parameters of the neural networks. In Q-learning, we first define a function $Q(\vec{S}, \vec{F}, \vec{\theta}_Q)$ representing the maximum discounted future reward when we choose $\vec{F}$ in state $\vec{S}$. We use a separate neural network to model the representation of the Q function and $\vec{\theta}_Q$ is the parameter of this second neural network.

The total future reward Q is a sum of the rewards r collected at each subsequent time step. The reward r at a specific time is computed using:

$$r = e^{-c_\nu(\vec{\nu}_o^* - \vec{\nu}_o)^2} + e^{-c_\theta(\theta_{back} - \theta_F)^2} \tag{6}$$

The first term minimises the difference between the actual and desired velocity, while the second term prioritises the backwards pulling direction. c_ν, c_θ are positive constants for the respective terms. A data set $<\vec{S}, \vec{F}, r, \vec{S}'>$ is defined as an *experience*, which is collected and stored for latter training processes. $\vec{S}'$ is the simulated state after the force $\vec{F}$ is applied in state $\vec{S}$.

Therefore, the Q value for a specific time can be represented as:

$$\begin{aligned} Q_t(\vec{S}, \vec{F}) &= r_t + \gamma r_{t+1} + \cdots + \gamma^{n-1}\gamma_{t+n} \\ &= r_t + \gamma Q_{t+1}(\vec{S}', \vec{F}') \end{aligned} \tag{7}$$

Layers	Force Policy	Q-value Network
Input	9	12
Hidden #1	16	16
Hidden #2	32	32
Hidden #3	16	16
Output	3	1

Table 1: Architecture of the two neural networks used in this work.

where γ is the discount factor of future reward. r_t is the reward at time t computed by Equation 6. If γ is zero, the policy only considers the instant reward and ignores the future reward. When γ is one, the policy considers the full effect of future rewards even though they are not deterministic. We choose a value (0.9) as a reasonable balance between these two extremes. Equation 7 is a *Bellman* equation, which means that the Q-function can be approximated by iteratively updating this equation until convergence.

The force policy π and Q-value network follow a similar architecture design. Each network is composed of 5 fully-connected layers. The first and last layer are the linear-weight neurons. The hidden layers are rectified linear units. The number of neurons for each layer are listed in Table 1.

The parameters $(\vec{\omega}, \vec{\theta})$ of the two neural networks are optimised by the method of Stochastic Gradient Descent (SGD). To iteratively optimise the parameters of the Q-learning network, we compute the loss function (or objective function) using :

$$L = \frac{1}{2}[\underbrace{r + \gamma Q(\vec{S}', \vec{F}', \vec{\theta})}_{target} - \underbrace{Q(\vec{S}, \vec{F}, \vec{\theta})}_{prediction}]^2 \quad (8)$$

Therefore, the optimal gradient direction is:

$$\frac{\partial L}{\partial \vec{\theta}} = [r+\gamma Q(\vec{S}', \vec{F}', \vec{\theta}) - Q(\vec{S}, \vec{F}, \vec{\theta})] \frac{\partial Q(\vec{S}, \vec{F}, \vec{\theta})}{\partial \vec{\theta}} \quad (9)$$

For the control policy π, the optimal parameters of ω would produce the maximum reward Q. Therefore, the gradient of the optimal policy is the direction that most improves Q:

$$\frac{\partial Q}{\partial \vec{\omega}} = \frac{\partial Q}{\partial \vec{F}} \frac{\partial \vec{F}}{\partial \vec{\omega}} = \frac{\partial Q}{\partial \vec{F}} \frac{\partial \pi}{\partial \vec{\omega}} \quad (10)$$

During runtime use, that is, after learning has been completed, the force is determined by forward-feeding the input through the decision network.

The force applied to the object is fundamentally related to the friction forces applied to the ant's stance legs. A double-tripod gait [20] is introduced to switch the legs between stance and swing. The front-left, middle-right and back-left legs are grouped as the *Left Tripod* while the other three legs are grouped as the *Right Tripod*. When the ant moves, the two groups of legs sequentially alternate between stance and swing. The following equations are used to distribute the desired dragging force F_i among the stance legs:

$$\begin{bmatrix} m\ddot{\vec{p}} \\ I\ddot{\vec{\theta}} \end{bmatrix} = \begin{bmatrix} I^D & I^D & I^D \\ [\vec{r}_{i,1}]^T_\times & [\vec{r}_{i,2}]^T_\times & [\vec{r}_{i,3}]^T_\times \end{bmatrix} \begin{bmatrix} \vec{F}_{i,1} \\ \vec{F}_{i,2} \\ \vec{F}_{i,3} \end{bmatrix} + \begin{bmatrix} -\vec{F}_i + \vec{G} \\ -\vec{F}_i \times \vec{r}_i \end{bmatrix} \quad (11)$$

where $\vec{F}_{i,j}$ $(j = 1, 2, 3)$ is the force from the j^{th} stance leg of the i^{th} individual. I^D is the identity matrix. m, I are the mass and inertia of the individual. $\vec{p}, \vec{\theta}$ are the position and orientation of the Center-of-Mass (COM) of each individual. $\vec{G}$ is the gravitational force. $\vec{r}_{i,j} = (r_x, r_y, r_z)$ is the vector connecting the j^{th} footprint to the COM of the individual. $[\vec{r}_{i,j}]^T_\times$ is the corresponding skew-symmetric matrix of $\vec{r}_{i,j}$:

$$[\vec{r}_j]^T_\times = \begin{bmatrix} 0 & r_z & -r_y \\ -r_z & 0 & r_x \\ r_y & -r_x & 0 \end{bmatrix} \quad (12)$$

Demonstration Task	Number of Characters	Frame Rate
Deadlock (Figure 3)	2	20.5
Crowd (Figure 5a)	4	13.6
Crowd (Figure 5b)	8	9.0
Crowd (Figure 5c)	16	3.9
Crowd (Figure 5d)	60	0.9

Table 2: Experiment data of runtime performance of selected demos.

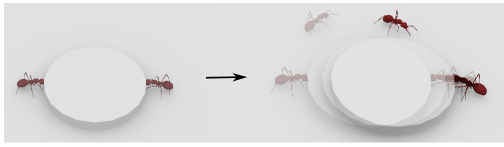


Figure 3: (Left) Using only the strategy of *realign*, two individuals can barely move the object. This creates the effect of deadlock. (Right) By repositioning one of the characters, two individuals apply force from a more consistent direction, thus resolving the issue of deadlock.

where $\vec{r}_i$ is the vector from the attachment point to the COM of the individual. Equation 11 has more than one solution if no further constraints are introduced. We reduce the redundant dimensions of the solution space by assuming that the vertical forces are spread equally over the stance legs.

4 Results and Discussions

The resulting motions from the behaviour engine and the trained force policy, are best seen in the supplemental video. The final force policy was resolved using 150k training iterations, collecting about 1 million tuples. The complete training process took approximately 30 hours on an 8-core computer. We use the open source deep learning framework *Caffe* [21] to build and train the networks. The runtime performance data, after training, is presented in Table 2. The runtime data was collected on a standard laptop with a Core i5-6200U @2.30GHz (CPU) and 8GB (RAM).

4.1 Realign

The strategy of *realign* adjusts the force direction applied by the individuals. In extreme cases (such as in Figure 3), two individuals drag the object from either ends, pulling the object in opposite directions. By adjusting the force directions only, the average translational velocity of the object is close to zero. In observations of real ants, the deadlock resulting from antagonistic pulling is rare and short in duration since real ants would soon reposition themselves [18]. To resolve this *deadlock*, one of the characters would choose to release the object and pick a new attachment point. This is illustrated on the right side of Figure 3.

4.2 Reposition

The strategy of *reposition* adjusts the point from which the individuals apply force. This strategy reduces the possibility of deadlock. This is further verified in the case of collective transport by a group of individuals (Figure 4). Six characters are initialised with even spacing around the object. Since the force policy is trained with the preference of dragging the object in a backwards direction, it is highly likely that forces with similar magnitude are applied from close-to-symmetrical directions. This creates the effect of deadlock similar to the case of two individuals in Figure 3. When one character releases the object, the deadlock is broken and the applied forces become asymmetrical. This re-enforces the probability that individuals who are pulling from opposing direction reposition themselves. The final result is the reorganised formation of the individual spacing. When a character approaches the target position and finds it occupied by another agent, it attempts alternative target locations until it finds an available one.

4.3 Adapting to Different Numbers of Individuals

One of the advantages of the decentralised paradigm is the scalability to different numbers of individuals. This is validated in our work by simulating the task of transport with different group sizes (Figure 5). In the real world, there always exists an optimal group size in order to balance between transport speed and energy efficiency. A larger group would recruit more individuals and thus increase the transport speed. However, the transport speed may not increase linearly with the number of individuals. Figure 6 plots the average transport speed with respect to the number of individuals. The results show that the linear relationship only exists for small team sizes (2$\sim$3 individuals). For greater numbers of individuals, the speed increases at a slower rate. Based on our simulation observations, the reasons for such a nonlinear relationship are two fold. First, when more individuals form a group, the object is generally transported at a higher speed, which in turn increases the probability of individuals repositioning themselves to different attachment points (Equation 3). Second, for an object with a fixed geometric size, an increasing number of individuals would have difficulty in finding an appropriate attachment position and avoiding bodily collisions with existing individuals who are already attached to the object. This leads to the fact that a significant proportion of additional individuals' time would then be spent on looking for an attachment point instead of actually pulling the object.

4.4 Following a Curve

In the previous examples, individuals know the location of the destination (or the nest). Forces are applied as vectors from their current location towards the final destination. In the real world, ants determine their path back to the nest via pheromone trails, which are chemicals laid by nest members and strengthened as the transport continues. Our method allows modelling complex pheromone paths by user-defined curves. The curve is first uniformly parameterised with the value range of [0, 1]. Users can specify the desired transport velocity on different segments of the curve. At time t, the desired location is passed to the controller and our method computes the control inputs for the individuals. The capability of following complex trajectories extends to scenarios such as obstacle avoidance (Figure 7). Although this is not the exactly the same as real ants, it is sufficient and flexible enough to allow artists to reproduce such a behaviour for virtual ants.

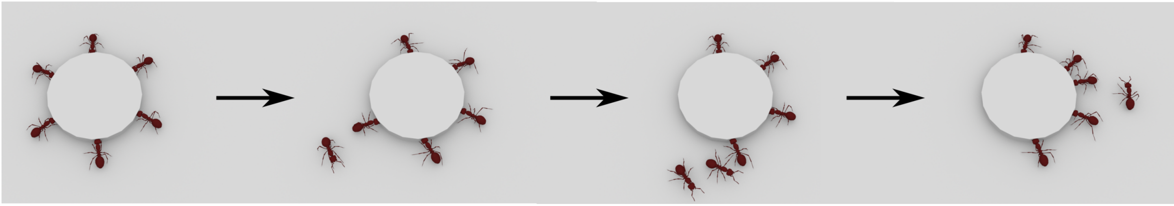


Figure 4: Collective transport by a team of six individuals. Characters are evenly distributed around the object during initialisation. As time proceeds, characters break the deadlock and start to move the object in an *uncoordinated* but *collective* fashion.

(a) (b) (c) (d)

Figure 5: Simulating the task of collective transport with different numbers of individuals (from left to right: 4, 8, 16, 60).

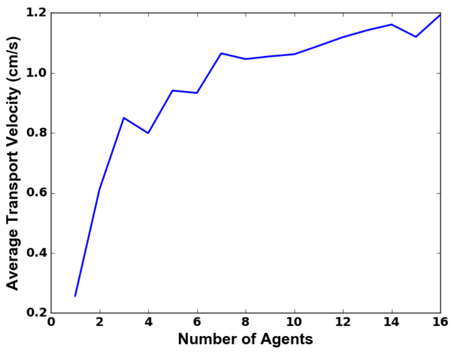


Figure 6: Average transport speed of an object with respect to the number of individuals.

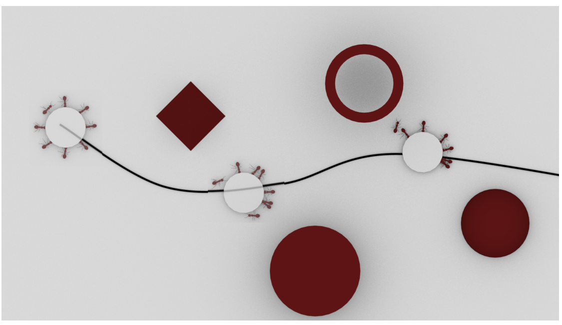


Figure 7: A group of ants are transporting an object along a predefined curve, creating the effect of obstacle avoidance.

4.5 *Adapting to Objects with Different Shapes*

Our method is also capable of simulating a group of ants transporting objects of arbitrary shape. This is validated in the example of the demo of objects with text-shape (Figure 8). The contour of the objects is represented as a set of connected line segments, which are checked for collisions with the geometry of the individual ants.

Figure 8: Ants transport objects of different shapes.

4.6 *External Intervention*

In the real world, the object could be abruptly relocated to another location by wind or even seized by competitors. We categorise such incidences, which cause the sudden relocation of the object, as external intervention. The stability of our method is demonstrated when there exists an external intervention during the process of transport. After the intervention is introduced, all individuals are forced to release the object. They then enter the state of *search* and start looking for the relocated object or an alternative if the original object is not found. Each individual switches to the state of *approach* and then *transport* if an object is detected within their sensory range. With this proposed strategy, the system is capable of accommodating external intervention (see Figure 9).

5 Conclusions

In a classical multi-character system, direct communication exists between individuals. The problem of *stigmery*, like the task of collective transport of ants, differentiates from classical systems because individuals act as if they are alone and do not directly share information with each other. This paper models the limited intelligence of real ants in nature and simulates the behaviour of collective transport which is commonly observed in ant colonies. This model is decentralised, scalable and does not require *a priori* information about the prey object. With no explicit communication but only with individual local sensing, this method is able to scale to scenarios with different numbers of individuals.

One future direction for this work would be to further validate our model by comparing our simulation model with real ants. This would include capturing video footage of real ants involved in the task of collective transport. The relevant information, including the timing and positioning of group members, could then be extracted using techniques from computer vision. The comparison could be used to optimise the parameters used in our behaviour engine model. Another challenge that is not yet fully solved in our work is the design of the *neural networks*. The current architecture is constructed based on empirical knowledge. Since there is no universal guidance on the design of *neural networks*, and compared to the large possibility of network architectures, we can only approach the solution via limited experimentation. How to extend this controller to scenarios other than the task of collective transport is one of the future directions for this research.

References

[1] J. H. Sudd, "The transport of prey by ants," *Behaviour*, vol. 25, no. 3/4, pp. 234–271, Jan. 1965, ArticleType: research-article / Full publication date: 1965 / Copyright 1965 BRILL.

[2] N. R. Franks, "Teams in social insects: group retrieval of prey by army ants (eciton burchelli, hymenoptera: Formicidae)," *Behavioral Ecology and Sociobiology*, vol. 18, no. 6, pp. 425–429, May 1986.

[3] S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar, and S. Pratt, "Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1470–1481, 2011.

[4] K. Hyun, K. Lee, and J. Lee, "Motion grammars for character animation," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 103–113.

[5] J. C. Chan, J. K. Tang, and H. Leung, "Synthesizing two-character interactions by merging captured interaction samples with their spacetime relationships," in *Computer Graphics Forum*, vol. 32, no. 7. Wiley Online Library, 2013, pp. 41–50.

[6] J. Hwang, I. Suh, and T. Kwon, "Editing and synthesizing two-character motions using a coupled inverted pendulum model," in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 21–30.

[7] K. Wampler, E. Andersen, E. Herbst, Y. Lee, and Z. Popovic, "Character animation in two-player adversarial games," *Acm Transactions on Graphics*, vol. 29, no. 3, pp. 483–496, 2010.

[8] H. P. Shum, T. Komura, and S. Yamazaki, "Simulating multiple character interactions with collaborative and adversarial goals," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 741–752, 2012.

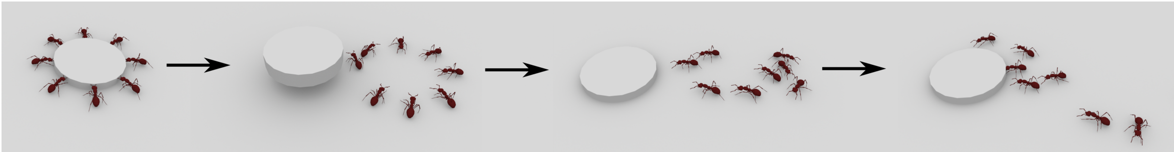


Figure 9: During the process of transport, user intervention can be introduced to abruptly move the object to another location. In such cases, group members return to searching for the object and rearrange themselves to form a new collective system to continue the transport task.

[9] J. Won, K. Lee, C. O'Sullivan, J. K. Hodgins, and J. Lee, "Generating and ranking diverse multi-character interactions," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 219, 2014.

[10] Q. Gu and Z. Deng, "Generating freestyle group formations in agent-based crowd simulations," *IEEE computer graphics and applications*, vol. 33, no. 1, pp. 20–31, 2013.

[11] J. Henry, H. P. Shum, and T. Komura, "Interactive formation control in complex environments," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 2, pp. 211–222, 2014.

[12] L. He, J. Pan, S. Narang, and D. Manocha, "Dynamic group behaviors for interactive crowd simulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '16. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2016, pp. 139–147.

[13] C. R. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robotics and autonomous systems*, vol. 30, p. 85101, 2000.

[14] S. Patil, J. Van Den Berg, S. Curtis, M. C. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 244–254, 2011.

[15] S. Takahashi, K. Yoshida, T. Kwon, K. H. Lee, J. Lee, and S. Y. Shin, "Spectral-based group formation control," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 639–648.

[16] A. Loria, J. Dasdemir, and N. A. Jarquin, "Leader–follower formation and tracking control of mobile robots along straight paths," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 727–732, 2016.

[17] M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal, "Collective transport of complex objects by simple robots: theory and experiments," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, ser. AAMAS '13. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013, p. 4754.

[18] J. H. Sudd, "The transport of prey by an ant, pheidole crassinoda em." *Behaviour*, vol. 16, no. 3/4, pp. 295–308, Jan. 1960, ArticleType: research-article / Full publication date: 1960 / Copyright 1960 BRILL.

[19] S. Guo, J. Chang, X. Yang, W. Wang, and J. Zhang, "Locomotion skills for insects with sample-based controller," *Computer Graphics Forum*, vol. 33, no. 7, pp. 31–40, 2014.

[20] S. Guo, J. Chang, Y. Cao, and J. Zhang, "A novel locomotion synthesis and optimisation framework for insects," *Computers & Graphics*, vol. 38, pp. 78–85, 2014.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.