Manuscript

Visually Smooth Multi-UAV Formation Transformation

Xinyu Zheng[1,*], Chen Zong[1,*], Jingliang Cheng[1], Jian Xu[2], Shiqing Xin[1], Changhe Tu[1], Shuangmin Chen[3,†], Wenping Wang[4]

Abstract

Unmanned airborne vehicles (UAVs) are useful in both military and civilian operations. In this paper, we consider a recreational scenario, i.e., multi-UAV formation transformation show. A visually smooth transformation needs to enforce the following three requirements at the same time: (1) visually pleasing contour morphing - for any intermediate frame, the agents form a meaningful shape and align with the contour, (2) uniform placement - for any intermediate frame, the agents are (isotropically) evenly spaced, and (3) smooth trajectories - the trajectory of each agent is as rigid/smooth as possible and completely collision free. First, we use the technique of 2-Wasserstein distance based interpolation to generate a sequence of intermediate shape contours. Second, we consider the spatio-temporal motion of all the agents altogether, and integrate the uniformity requirement and the spatial coherence into one objective function. Finally, the optimal formation transformation plan can be inferred by collaborative optimization.

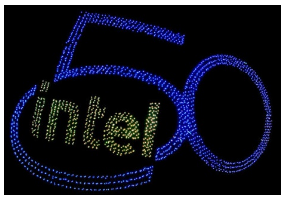
Extensive experimental results show that our algorithm outperforms the existing algorithms in terms of visual smoothness of transformation, boundary alignment, uniformity of agents, and rigidity of trajectories. Furthermore, our algorithm is able to cope with some challenging scenarios including (1) source/target shapes with multiple connected components, (2) source/target shapes with different typology structures, and (3) existence of obstacles. Therefore, it has a great potential in the real multi-UAV light show. We created an animation to demonstrate how our algorithm works; See the demo at `https://1drv.ms/v/s!AheMg5fKdtdugVL0aNFfEt_deTbT?e=le5poN`.

[*]Co-first author.
[†]Corresponding author.
[1]School of Computer Science and Technology, Shandong University, Qingdao, China.
[2]Ningbo Institute of Materials Technology & Engineering, Chinese Academy of Sciences, Ningbo, China.
[3]School of Information and Technology, Qingdao University of Science and Technology, Qingdao, China.
[4]Department of Computer Science, HKU, China.

1. Introduction

"Several years ago, we had an idea of flying drones forming the Intel logo over our corporate headquarters, and here we are doing just that. It really speaks to the innovative spirit that Intel was founded on 50 years ago."

–Anil Nanduri, vice president and general manager, Intel Drone Group



To celebrate Intel's 50th anniversary, the company honored employees and their families by flying 500 Intel® Shooting Star drones over its corporate headquarters in July 2018. Each drone was equipped with LED lights that can create countless color combinations and can easily be programmed for any animation, and the entire fleet of drones was controlled by one pilot.

In fact, multi-UAV formation transformation belongs to flock control that is a hot research topic in diverse fields. Some research works focus on realistic-looking simulation of collective behavior, such as simulation of artificial life [37, 3, 42, 53, 22], simulation of transportation planning [38] and simulation of emergency evacuation [48, 14], while some focus on seeking for a minimum-cost task allocation plan [8, 34] that considers both low-level position control and high-level motion planning [28, 24, 46] from the perspective of either distributed computing or centralized computing. The research theme of this paper, i.e., multi-UAV formation transformation, is to find an optimal motion arrangement such that a flock of UAVs can be smoothly transformed from a source shape to a target shape.

Commonly used criteria include (a) the UAV agents must follow a uniform placement at any intermediate time point, and (b) each UAV agent moves along an as-rigid/smooth-as-possible and collision-free trajectory during the process of formation transformation. In fact, it is observed that whether the intermediate shape is able to define a visually meaningful shape is equally important. Therefore, in this paper, we sum up them into three requirements: (1) visually pleasing contour morphing, (2) uniform displacement of agents, and (3) smooth trajectories.

It is challenging yet fascinating to find an automatic formation transformation plan to satisfy the three requirements at the same time. In this paper, we extend the technique of 2-Wasserstein distance based interpolation [40] to generate the contour of an intermediate shape. We assume that the i-th UAV is positioned at $\mathbf{x}_i^k$ at time t_k and come to infer $\{\mathbf{x}_i^k\}_{i=1,k=1}^{n,K}$, where n is the number of UAVs and K is the number of key frames. On the one hand, we define a CVT-like energy to measure the uniformity between UAVs at time t_k. On the other hand, we use $\sum_k \|\mathbf{x}_i^k - \mathbf{x}_i^{k+1}\|^2$ to measure the rigidity and smoothness of the motion path of the i-th UAV. Finally, we integrate the uniformity requirement and the spatial-coherence requirement into one objective function and then infer the optimal formation transformation by collaborative optimization.

Our contributions are three-fold:

1. We propose a collaborative optimization based framework to formulate the multi-UAV formulation transformation problem, which simultaneously takes into the above-mentioned multiple requirements.
2. Based on the observation that the 2-Wasserstein distance interpolation [40] depends on the domain boundary, we propose to restrict the domain into a more compact region, which can yield a more visually pleasing contour morphing sequence.
3. We give an adaptive parameter tuning mechanism for balancing the uniformity energy and the spatial-coherence energy. We also give a fast swap strategy to strictly guarantee that any two trajectories are collision free.

2. Related Work

2.1. Flock/crowd simulation

Flock is a system consisting of many autonomous units. The connections between the units may be local or global depending on specific occasions, and the interaction between them can be as simple as attraction/repulsion or more complex. The topic of flock/crowd simulation draws wide attention due to both theoretical and application values [37, 3, 42, 53, 22]. The problem has been studied in both normal and emergency situations. Existing approaches on flock/crowd simulation can be roughly categorized into two classes: hypothesis based and data driven.

Most of the hypothesis based approaches are based on biomechanical knowledge such as biological rhythms and psycho-social diagnosis [23, 7]. The existing approaches can be further divided into microscopic, macroscopic and mesoscopic, where macroscopic models mostly simulate overall crowd behavior in a large-scale crowd, microscopic models focus on detailed behavior and unique features of individuals, while mesoscopic models fall in between. Macroscopic models, without respecting realistic biomechanical limitations, are particularly suitable for dense, homogeneous crowds and complex environments but always lead to unrealistic agent behaviors. The underlying fluid dynamics potential fields include continuum fields [18, 44, 10], dynamic aggregation functions [29], navigation fields [31, 5] and so on. Microscopic models, different from macroscopic models, are used to infer complex pedestrian movements while preserving physical, social and psychological features of individuals. They are generally less efficient in the simulation of large-scale crowd. Typical approaches on this side include force-based models [36, 15], velocity-based models [9, 30, 32, 39] and agent-based models [20, 50]. Mesoscopic models aim at combining macroscopic models and microscopic models together and take more aspects into account. It can be divided into at least the following three kinds: zone-based models [51, 2], layer-based models [4, 43] and sequential models [52].

In recent years, the data-driven strategy [26, 19, 35] becomes increasingly important within the realm of flock/crowd simulation. Since the real crowd behavior may be influenced by a variety of factors such as distance to neighbors, spatial formation, personal habit and emotion, learning based approaches are proposed to use real captured emergency and evacuation scenarios to obtain prior knowledge. The kind of approaches diminishes the use of hypothetical rules and is able to enhance visual realism. The realistic simulated results can be further used to perform crowd analysis [13, 6], or facilitate direct manipulation on crowd scenes [25, 21].

2.2. Group formation control

Generally speaking, the task of group formation control is to explicitly control spatial adjacency between individuals or their subgroups in order to generate meaningful spatio-temporal transitions. The requirements include at least boundary alignment, stable adjacency structure, uniform in-between gap and reasonable intermediate formation. It's a challenging task to achieve these goals at the same time. Kwon et al. [25] suggested applying the technique of Laplacian mesh editing to deform and concatenate existing crowd formations to synthesize large scale animations. The key idea is to minimize the distortion of relative arrangements among adjacent agents. Takahashi et al. [41] proposed to interpolate two given formations based on spectral analysis. In spite of the ability to generate artistic intermediate formations, the approach doesn't take the uniform placement or boundary alignment into account. Besides, there are many research works on controlling high-level movements by interactive sketch [11, 12], multitouch [16, 17] and so on.

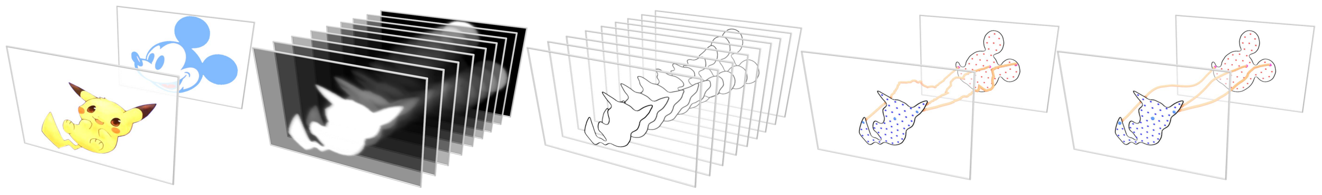

(a) (b) (c) (d) (e)

Figure 1. Algorithm pipeline. (a) The source/target formation shapes. (b) K representative formation shapes. (c) Extract the shape contour. (d) Initialize the positions/trajectories of agents for every frame. (e) Collaboratively optimize the agent positions and trajectories.

2.3. Cost driven formation transform

Alonso-Mora et al. [1] found that centroidal Voronoi tessellation (CVT) is helpful in generating a uniform distribution of agents. They further proposed to guarantee avoidance trajectories by Hungarian algorithm. Zheng et al. [55] observed that the Lloyd descent method of CVT is able to yield a set of collision free trajectories. Xu et al. [54] used the subgroup-based social force model (SFM) for purpose of cohesion such that the intermediate frames look visually pleasing. The reciprocal collision avoidance is guaranteed by minimizing the overall travelling distance. However, these approaches face with a common issue, i.e., how to deal with the multiple challenges simultaneously.

In this paper, we develop a set of systematic strategies to deal with the above-mentioned challenges. First, we extend 2-Wasserstein distance based displacement interpolation [40] to generate intermediate frames; We further restrict the transformation into a more compact domain such that the interpolated shapes are more consistent with human's perception. Second, we enforce the uniformity of agents by utilizing the technique of centroidal Voronoi tessellation (CVT). Third, we give a fast swap based strategy to guarantee that any two trajectories are collision free. Finally, we integrate the uniformity requirement and the trajectory smoothness requirement into a unified algorithm framework, and report the formation transformation plan by collaborative optimization.

3. Algorithm

In this section, we first give the algorithm overview, followed by elaborating the key constituent steps in the following subsections.

3.1. Overview

Given the source/target formation shapes (see Figure 1(a)), our algorithm consists of the following five steps; See Figure 1.

1. Infer $K-2$ intermediate formation shapes that are represented by a density field; See Figure 1(b).
2. Extract the shape contour frame by frame; See Figure 1(c);
3. Initialize the positions of agents at every time point with CVT; See Figure 1(d).
4. Collaboratively optimize the agent positions and trajectories such that the uniformity requirement and the path-smoothness requirement are satisfied at the same time; See Figure 1(e).
5. Identify those agent pairs with possible trajectory intersection and enforce a swap operation until all the trajectories are completely free of collision.

3.2. Wasserstein distance based shape interpolation

2-Wasserstein distance, thanks to its ability of characterizing the distance between two probability distributions $\mu \in \Omega_1$ and $\nu \in \Omega_2$, has the potential to interpolate two images or shapes [33, 47]. Its definition is as follows:

$$W_2(\mu,\nu) \overset{\text{def}}{=} \sqrt{\inf_{\xi \in \Xi(\mu,\nu)} \iint_{\Omega_1 \times \Omega_2} d^2(\mathbf{x},\mathbf{y}) \mathrm{d}\xi(\mathbf{x},\mathbf{y})},$$

where Ξ denotes a transportation plan and $\mathrm{d}\xi(\mathbf{x},\mathbf{y})$ means the amount of mass moving from $\mathbf{x} \in \Omega_1$ towards $\mathbf{y} \in \Omega_2$.

Imagine that we have two 2D contours as the input and each of them is represented by a binary image. After normalized to a unit total mass, one can infer a transition shape (represented by a density field) like barycentric interpolation:

$$\rho(t) = \arg\min_\rho (1-t)W_2^2(\rho_0,\rho) + tW_2^2(\rho_1,\rho),\ t \in [0,1].$$

Solomon et al. [40] proposed an efficient numerical method to solve this problem on a geometric domain. Take Figure 2 for an example. If we view the square as the domain, the interpolated result may be of a scattered shape; See Figure 2(a-b); But if we restrict the domain into a more compact region, it will own higher cohesion; See Figure 2(c-d); The rational behind lies in different distance metrics. Rather than induce the Wasserstein distance by straight-line distance, we use the interior distance instead for the restricted domain, which is able to take the cohesion requirement into account and better manifest how the agents move from one position to the other. We detail the domain restriction technique as follows.

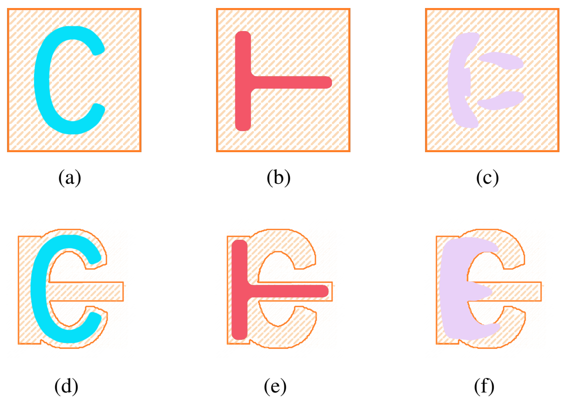


Figure 2. The choice of base domain determines the way of how to measure the distance between two points, and thus leads to different interpolation behaviors. (a-c) Squared domain and the interpolated shapes at $t = 0, t = 1, t = 0.5$. (d-f) Restricted domain and the corresponding interpolated shapes at $t = 0, t = 1, t = 0.5$. Obviously, the restricted domain tends to interpolate a more visually pleasing result.

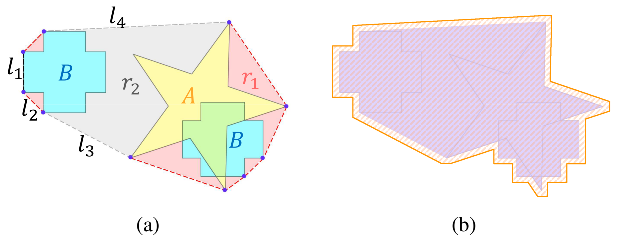


Figure 3. (a) By replacing the supporting segment of each inward-corner subregion (colored in pink) with the counter-part boundary segment of $A \cup B$, we get a compact region that encloses A and B. (b) The final restricted domain can be obtained by enforcing an additional small outward offset. The restricted domain defines the accessible area where an agent is allowed to move.

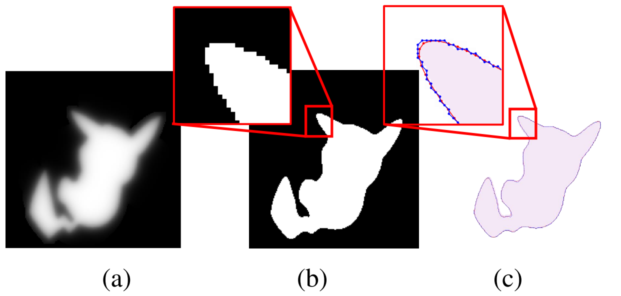


Figure 4. (a) Grayscale image. (b) Distinguish the foreground from the background with the average grayscale value. (c) Before/after boundary smoothing (the smoothed boundary is colored in red).

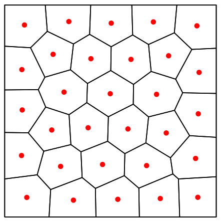
Domain restriction. Suppose that there are two shapes A and B, as shown in Figure 3(a), and we come to define a compact and connected region to enclose A and B, which will be used as the geometric domain to define an interpolated shape. Let ConvexHull(A, B) be the convex hull of A and B. The boundary of ConvexHull(A, B) can be further split into two kinds of curved segments: (a) contacting segments (e.g., l_1) and (b) supporting segments (e.g., l_2, l_3, l_4). The region ConvexHull$(A, B) \backslash (A \cup B)$ consists of separate sub-regions. One type of sub-regions has only one supporting segment, and the other type has at least two supporting segments. They are respectively named *inward corners* (pink) and *penetrating parts* (grey). Finally, we traverse the boundary of ConvexHull(A, B) in a counter-clockwise direction and replace the supporting segment of each inward-corner subregion with the counter-part boundary segment of $A \cup B$. For consideration of numerical robustness, we enlarge the resulting area by a small gap, generally 5% of the bounding box size, and obtain a restricted domain that allows an agent move inside; See Figure 3(b).

Shape extraction. We distinguish the foreground from the background with the average grayscale value c. Imag-ine that we get the histogram of the grayscale value distribution; See Figure 4(a). To guarantee the numerical robustness, we compute the average grayscale value by using only those pixels whose grayscale value is between $c_{\min+1}$ and $c_{\max-1}$, where $c_{\min}$ and $c_{\max}$ are respectively the minimum grayscale value and the maximum grayscale value. The contour can be thus extracted at the level of the average grayscale value; See Figure 4(b). Finally, we perform a simple smoothing operation; See Figure 4(c).

3.3. Collaborative optimization

As mentioned above, we have two kinds of requirements on the position configuration of the agents. On the one hand, we hope that the agents are uniformly placed in each intermediate shape. On the other hand, we hope that each agent trajectory is as rigid/smooth as possible. In the following, we shall integrate the couple of requirements into one objective function:

CVT-like energy. Centroidal Voronoi tessellations (CVTs), as a special type of region decomposition, are proven to be useful in providing an exceedingly uniform point placement. In its essence, CVT is to find the best discrete point set that can well approximate the original continuous density function ρ. After the given domain Ω is paritiononed into a set of sub-regions by a Voronoi diagram, CVT measures the uniformity by summing the second-order moment of each sub-region together. We discretize the whole transform process into K representative frames, where t_1, t_K are respectively the time points of the first frame and the last frame. In our problem, we assume that the i-th agent is positioned at $\mathbf{x}_i^k \in \Omega^k$



at $t_k, k = 1, 2, \cdots, K$. We define the overall uniformity as follows:

$$F_{\text{CVT}} = \sum_{k=1}^{K} \sum_{i=1}^{n} \int_{\Omega_i^k} \rho_k \|\mathbf{x} - \mathbf{x}_i^k\|^2 \mathrm{d}\mathbf{x},$$

where $\rho_k = 1/|\Omega^k|$ is a constant density function defined in the k-th intermediate shape, which is used to equally emphasize the importance of different frames.

Spatial coherence. We can enforce the requirement that the trajectories are as rigid/smooth as possible by reducing the total travelling distance or an equivalent cost. Recall that we discretize the time span into $t_1, t_2, \cdots, t_K$. It's reasonable to assume that each agent moves along a straight-line segment between t_k and t_{k+1}. When the total moving cost is minimized, collision between agents can be effectively avoided (we shall discuss the avoidance problem later). Here we use the sum of squared distances to measure the overall travelling cost:

$$F_{\text{Smoothness}} = \sum_{i=1}^{n} \sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|^2.$$

In fact, according to the Cauchy–Schwarz inequality

$$\sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|^2 \geq \frac{1}{K-1} (\sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|)^2,$$

we have two observations:

- When $\sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|^2$ is completely minimized, all the intermediate positions are collinear and lying on the segment $\mathbf{x}_i^1 \mathbf{x}_i^K$, which explains why the avoidance term $F_{\text{Smoothness}}$ can make the path as rigid/smooth as possible. (It's worth mentioning that generally the resulting path is not necessarily a straight-line segment since we shall balance the term $F_{\text{Smoothness}}$ and the term F_{CVT}.)
- When $\sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|^2$ is completely minimized, all the intermediate positions are equally spaced, which implies that the i-th agent roughly moves in a constant speed throughout the formation transformation.

Initialization. We need to initialize $\{\mathbf{x}_i^k\}_{i=1,k=1}^{n,K}$, totally $n \times K$ positions, at the very beginning of the optimization. First, with the support of CVT, we generate a set of uniformly distributed positions $\{\mathbf{x}_i^1 \in \Omega^1\}_{i=1}^{n}$ at the time point t_1. We take the resulting positions $\{\mathbf{x}_i^1 \in \Omega^1\}_{i=1}^{n}$ as the initial placement of agents in the second frame, and generate n corresponding positions $\{\mathbf{x}_i^2 \in \Omega^2\}_{i=1}^{n}$ with CVT. In this way, $\{\mathbf{x}_i^k\}_{i=1,k=1}^{n,K}$ can be incrementally initialized frame by frame. Such a progressive initialization style enables a rough spatial coherence between successive frames.

Super-linear optimization. By combining the above mentioned two kinds of requirements, we get a new objective function:

$$F = F_{\text{CVT}} + \lambda F_{\text{Smoothness}}, \tag{1}$$

whose gradients w.r.t. $\mathbf{x}_i^k$ is

$$\nabla_{\mathbf{x}_i^k} F = 2m_i^k(\mathbf{x}_i^k - \mathbf{c}_i^k) + 2\lambda(2\mathbf{x}_i^k - \mathbf{x}_i^{k-1} - \mathbf{x}_i^{k+1}),$$

where λ is a balance parameter (the choice of λ will be discussed later), m_i^k is the mass of Ω_i^k, and $\mathbf{c}_i^k$ is the mass center of Ω_i^k. Like [27], we use the L-BFGS solver to quickly find the solution with super-linear convergence rate. The optimization terminates when $\|\nabla F\| \leq 1e-6$. At the conclusion of the optimization, two kinds of information are reported, including (1) the position of the i-th agent at t_k, i.e, $\mathbf{x}_i^k$, and (2) the motion path of the i-th agent:

$$\Gamma_i : \mathbf{x}_i^1 \rightarrow \mathbf{x}_i^2 \rightarrow \cdots \rightarrow \mathbf{x}_i^K.$$

See Figure 1(e) for an example.

3.4. Adaptive tuning of λ

In our objective function, there are two energy terms that need to balanced. Experimental results show that the choice of λ depends on many factors including the number of agents, the number of intermediate frames, the scale of the input shapes, and the source/target positions. It is not easy to find a fixed λ that owns high suitability. In the following we give an adaptive strategy to automatically tune the value of λ.

Normalization. We assume that a normalized scenario is roughly like this: the source shape and the target shape are scaled into a $[0, 1] \times [0, 1]$ square, and both K and n are set to 50. We rewrite the balancing parameter λ as

$$\lambda = \lambda' \times \frac{s}{1} \times (\frac{K}{50})^2 / (\frac{n}{50})^2, \tag{2}$$

where s is the real scale of the scenario. The above formula enables us to tune λ' instead, without considering the number of intermediate frames, the number of agents, as well as the scenario scale. In the defaulting setting, we set λ' to be 1.

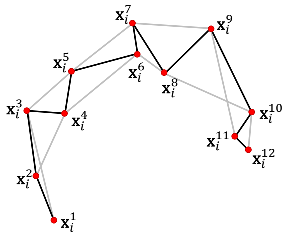
Uniformity measure. As pointed out in [49], one can use the CVT energy to measure the uniformity of the agent distribution. Recall that we initialize the agent positions for each frame with a single CVT optimization (see Figure 1(d)). Suppose that $F_{\text{CVT}}^{\text{init}}$ denotes the total CVT energy at initialization. It's natural that the collaborative optimization improves the path smoothness at the cost of uniformity.

Let $F_{\text{CVT}}^{\text{opt}}$ be the total CVT energy after optimization. We have $F_{\text{CVT}}^{\text{opt}} \geq F_{\text{CVT}}^{\text{init}}$ generally. We define

$$\tau_1 = \sqrt{\frac{F_{\text{CVT}}^{\text{opt}}}{F_{\text{CVT}}^{\text{init}}}},$$

and use $\tau_1 (\geq 1)$ to evaluate the uniformity. The closer to 1 the value of τ_1, the more uniform the agents will be.

Trajectory smoothness measure. Since our algorithm takes the cohesion feature into consideration such that the intermediate formation is more visually pleasing (see Section 3.2), the trajectory of the i-th agent, say Γ_i, is generally not a straight-line segment. Here, we measure how zigzag Γ_i is by checking if Γ_i can be refined to a much shorter path with slight perturbation. Considering that Γ_i is a polyline curve with K vertices, we connect any pair of vertices if they are in a two-ring neighborhood; See the inset figure. Suppose that the augmented graph is $\mathcal{G}$ and the corresponding shortest path is $\Gamma_i^{\mathcal{G}}$. It is proper to define an indicator to measure the trajectory smoothness with $\frac{\|\Gamma_i\|}{\|\Gamma_i^{\mathcal{G}}\|}$, which is obviously equal to or larger than 1. By averaging the indicator over all the agents' trajectories, we obtain the overall smoothness indicator τ_2 ($\geq$1). The closer to 1 the value of τ_2, the smother the trajectories will be.



Adaptive strategy. Without doubt, λ' is more convenient for users to tune than λ. However, experimental results show that the choice of λ' still depends on the source/target shapes, and thus manual adjustment of λ' is very tedious. With the support of τ_1 and τ_2, we have a convenient way to tune λ' by checking if the two measures are sufficiently close to each other. We observe that when the difference between τ_1 and τ_2 is very small, e.g., less than 0.3, the balance between the trajectory smoothness and the agent uniformity can be carefully maintained. Based on the observation, we develop an adaptive strategy to tune λ' automatically. The algorithm pseudo-code is available in Algorithm 1. Experimental results with different λ' are demonstrated in Fig-ure 5. It can be seen that a larger λ' tends to make the paths to be smoother.

3.5. Collision-free trajectories

A basic rule to trajectory planning is to guarantee that different agents are free of collision during the entire formation transformation. Generally speaking, the requirement that no collision occurs between the i-th agent and the j-agent implies (1) Γ_i and Γ_j don't intersect each other at all, or (2) the two trajectories have an intersection point but the two agents pass through that intersection at different time points.

Algorithm 1 Adaptive tuning of λ'

```
Input: ${\{\mathbf{x}_i^k\}}_{i=1,k=1}^{n,K}$,$\tau_1$,$\tau_2$;
Output: $\lambda'$;
 1: $\lambda' = 1$;
 2: step = 1;
 3: $\tau_1$,$\tau_2$ = CollaborativeOptimization($\{\mathbf{x}_i^k\}$,$\lambda'$);
 4: flag = $\tau_1 < \tau_2$;
 5: while $|\tau_1 - \tau_2| > 0.3$ do
 6:     if $(\tau_1 < \tau_2) \neq$ flag then
 7:         step *= 0.5;
 8:     end if
 9:     if $\tau_1 < \tau_2$ then
10:         $\lambda'$ -= step;
11:     else
12:         $\lambda'$ += step;
13:     end if
14:     $\tau_1$,$\tau_2$ = CollaborativeOptimization($\{\mathbf{x}_i^k\}$,$\lambda'$);
15:     flag = $\tau_1 < \tau_2$;
16: end while
```

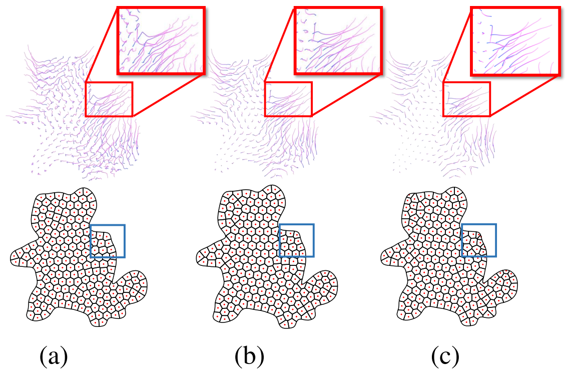


Figure 5. The parameter λ' is used to balance trajectory smoothness and agent uniformity - a larger λ' tends to encourage the trajectory smoothness. The top row shows the trajectories while the bottom row shows the intermediate frame at $t = 0.5$. (a) $\lambda' = 4.17$. (b) $\lambda' = 41.7$. (c) $\lambda' = 417$.

In fact, the agent trajectories yielded our collaborate optimization are mostly collision free since it takes the travelling cost into account. But there may still exist a few pairs of agents that may collide with each other. In the following, we shall explain why collision possibly occurs and how to re-arrange trajectories based on swap operations.

Spatial coherence. In our current formulation, we use $\sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|^2$ to characterize the spatial coherence of Γ_i. The intention behind is to make the i-th agent have a least moving cost, which is similar to but different from $\sum_{k=1}^{K-1} \|\mathbf{x}_i^k \mathbf{x}_i^{k+1}\|$.

First, if we replace the spatial coherence term by $\sum_{k=1}^{K-1}\|\mathbf{x}_i^k\mathbf{x}_i^{k+1}\|$ in our formulation, it can be verified that Γ_i and Γ_j are naturally free of intersection. We explain this observation as follows. Suppose that the two trajectories Γ_i and Γ_j collide with each other at a time point between t_k and t_{k+1}. By swapping the sub-sequence

$$\{\mathbf{x}_i^{k+1}, \mathbf{x}_i^{k+2}, \cdots, \mathbf{x}_i^K\}$$

and the sub-sequence

$$\{\mathbf{x}_j^{k+1}, \mathbf{x}_j^{k+2}, \cdots, \mathbf{x}_j^K\},$$

$\|\Gamma_i\| + \|\Gamma_j\|$ is decreased but the uniformity term F_{CVT} remains unchanged. This shows that if we define $F_{\text{smoothness}}$ by $\sum_{k=1}^{K-1}\|\mathbf{x}_i^k\mathbf{x}_i^{k+1}\|$, the optimal transformation by our formulation naturally guarantees that any two trajectories cannot collide with each other.

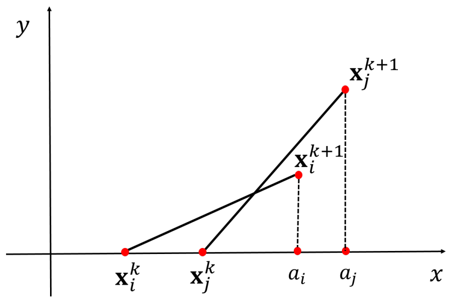


Figure 6. Minimizing $\|\mathbf{x}_i^k\mathbf{x}_i^{k+1}\|^2 + \|\mathbf{x}_j^k\mathbf{x}_j^{k+1}\|^2$ is able to avoid intersection between $\mathbf{x}_i^k\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^k\mathbf{x}_j^{k+1}$ except that $a_i < a_j$ and $\mathbf{x}_i^k \to \mathbf{x}_j^k \to \mathbf{x}_i^{k+1} \to \mathbf{x}_j^{k+1}$ forms a convex polygon, where a_i, a_j are respectively the projection points of $\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^{k+1}$ onto the directional axis $\overrightarrow{\mathbf{x}_i^k\mathbf{x}_j^k}$.

However, $\sum_{k=1}^{K-1}\|\mathbf{x}_i^k\mathbf{x}_i^{k+1}\|$ is not smooth with regard to the positional variables $\{\mathbf{x}_i^k\}$, causing low convergence rate. That's why we use $\sum_{k=1}^{K-1}\|\mathbf{x}_i^k\mathbf{x}_i^{k+1}\|^2$ instead to characterize the spatial coherence of Γ_i. As Figure 6 shows, we create a straight line through $\mathbf{x}_i^k$ and $\mathbf{x}_j^k$, and let a_i and a_j be respectively the projection points of $\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^{k+1}$ onto the directional axis $\overrightarrow{\mathbf{x}_i^k\mathbf{x}_j^k}$. Generally, when $\|\mathbf{x}_i^k\mathbf{x}_i^{k+1}\|^2 + \|\mathbf{x}_j^k\mathbf{x}_j^{k+1}\|^2$ is minimized, it naturally guarantees that there is no intersection between $\mathbf{x}_i^k\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^k\mathbf{x}_j^{k+1}$. Let a_i, a_j be respectively the projection points of $\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^{k+1}$ onto the directional axis $\overrightarrow{\mathbf{x}_i^k\mathbf{x}_j^k}$. The only exception occurs when $a_i < a_j$ and $\mathbf{x}_i^k \to \mathbf{x}_j^k \to \mathbf{x}_i^{k+1} \to \mathbf{x}_j^{k+1}$ forms a convex polygon. We omit the proof for brevity. Empirical observation is that when the optimization terminates, the trajectory collision may occur for only a few agent pairs. Despite the rarity, we still have to enforce an additional step to strictly guarantee the avoidance requirement. In the following, we shall develop a fast swap based strategy to handle this issue.

Swap strategy. Imagine that we have the agent positions $\{\mathbf{x}_i^k\}_{i=1}^n$ at the time point t_k, as well as the positions $\{\mathbf{x}_i^{k+1}\}_{i=1}^n$ at t_{k+1}. A naïve idea for obtaining an intersection free matching between $\{\mathbf{x}_i^k\}_{i=1}^n$ and $\{\mathbf{x}_i^{k+1}\}_{i=1}^n$ (in terms of straight-line distance) is to utilize Hungarian algorithm. However, Hungarian algorithm costs $O(n^3)$ time to find the perfect matching, which is computationally expensive.

A straightforward way is to check all pairs of agents. If $\mathbf{x}_i^k\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^k\mathbf{x}_j^{k+1}$ intersect each other, then we swap $\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^{k+1}$ for the i-th agent and the j-th agent. It costs $O(n^2)$ time for one iteration. Repeat this process until no collision occurs. Empirical observation shows the number of iterations ranges between 3 to 10 generally. To this end, the empirical timing cost can be taken as $O(n^2)$. In our setting, the agents, optimized by the CVT-like energy, are distributed very uniformly at every frame. We observe that it is enough to check every pair of neighboring agents. Inspired by this observation, we visit the Voronoi tessellation by $\{\mathbf{x}_i^k\}_{i=1}^n$ at t_k, and for each pair of neighboring agents at $\mathbf{x}_i^k$ and $\mathbf{x}_j^k$, we swap their descendant positions if $\mathbf{x}_i^k\mathbf{x}_i^{k+1}$ and $\mathbf{x}_j^k\mathbf{x}_j^{k+1}$ intersect each other. In this way, the timing cost can be further reduced from $O(n^2)$ to $O(n)$. (For purpose of being strictly collision free, we also perform similar checking operations for any two second-order neighboring agents in our implementation.)

4. Evaluation

4.1. Timing

We implemented our formation transformation algorithm in Matlab and C++. The experiments were conducted on a PC with Intel(R) Core(TM) i7-8700 CPU 3.20 GHz and 16 GB memory. The computational cost mainly consists of three parts: intermediate contour generation (Matlab), initialization with CVT (C++) and collaborative optimization (C++). Two main factors, i.e., the number of agents and the number of intermediate frames, are central to the overall timing cost. From the timing statistics in Table 1, we can see that the total cost climbs with K and n but not dramatically. In real formation transformation show, it is sufficient to set $K = 100$ and $n = 50$. Our algorithm requires totally 4 minutes to report the transformation plan for such a parameter setting, and nearly half of the timing cost is spent on interpolating the intermediate shapes (represented by a binary image). We shall further boost the performance in the future.

4.2. Comparison

Here we compare our algorithm with the following approaches:

- ORCA [45]: Based on velocity obstacles (VO), it fo-

Model	K	n	Intermediate contours (s)	Agent position initialization (s)	Optimization (s)	Total (s)
Mickey-Pikachu	50	50	68.6	19.0	44.5	133.4
Figure 1	50	100	68.6	29.2	107.8	207.0
	100	50	135.7	23.8	63.0	226.3
Circle-Pentacle	50	50	60.8	20.4	24.6	107.3
Figure 7	50	100	60.8	36.2	51.3	149.7
	100	50	121.2	31.4	66.3	221.7

Table 1. Time statistics with different K's and different n's.

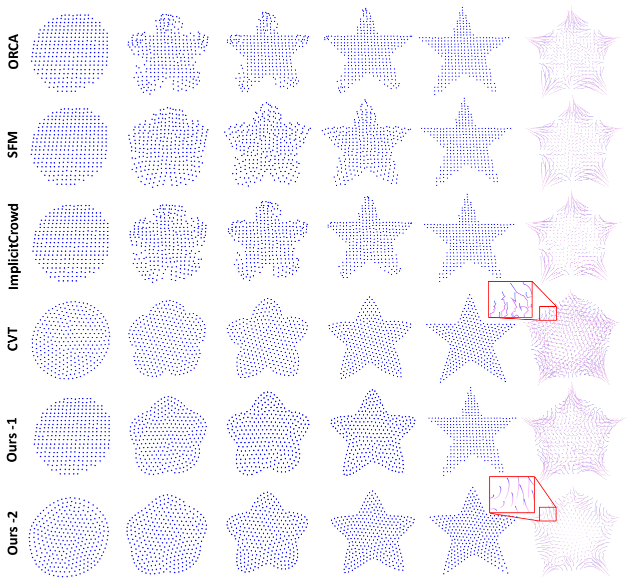


Figure 7. Transforming a disk to a star. From top to bottom: ORCA [45]; SFM [54]; ImplicitCrowd [20]; CVT [55]; Ours-1 is our result obtained by inputting user specified agent placement in the source/target shapes; Ours-2 is our result obtained by automatic computation of the agent placement in the source/target shapes. The last column shows the corresponding trajectories.

cuses on finding a strictly collision-free motion plan by solving a low-dimensional linear program. The approach doesn't take visually pleasing intermediate shapes into account and thus not suitable for the formation transformation purpose.

- SFM [54]: It adapts the original social force model (SFM) by adding a local attraction term to maintain the local stability. However, the approach lacks explicit control of intermediate shapes, which makes it fail to precisely guarantee the uniformity of agents as well as the boundary alignment property.
- ImplicitCrowd [20]: Based on the assumption that the interaction energy between any given pair of pedestrians follows a power law as a function of their projected time to collision, it defines an anticipatory potential function and uses an optimization-based implicit integration scheme to deal with the crowd simulation problem. By contrast, ORCA is overly conservative while ImplicitCrowd can give more realistic behavior. However, it cannot satisfy either the agent uniformity requirement or the boundary alignment requirement.
- CVT [55]: It uses the Lloyd descent method of CVT to yield a set of collision free trajectories. Although it enables both the uniformity of agents and the trajectory avoidance requirement, the approach doesn't enforce the trajectory smoothness. Experimental results show that the motion paths are zigzag; See the close-up window. In addition, it includes a step of pre-computing intermediate contours. In our implementation, we also use the 2-Wasserstein distance based displacement interpolation technique to generate the contour morphing sequence.

In Figure 7, we show an example of transforming a disk to a star, where the last column shows the corresponding trajectories. It's worth mentioning that our algorithm not only supports user specified agent placement in the source/target shapes (see Ours-1), but also can infer a more uniform placement automatically (see Ours-2). It can be seen from Figure 7 that our algorithm can achieve the three goals at the same time, i.e., (1) visually pleasing intermediate formation, (2) uniform agent placement, and (3) smooth/rigid trajectories.

4.3. More challenging scenarios

Variable number of agents. Even if the source/target shapes are given, it is challenging yet necessary to accomplish the formation transformation with a limited number of agents. What is the most important is that the formation must follow a visually pleasing/meaningful shape during the transformation. Besides, it is also important to make the agents evenly spaced so that the region of interest has a balanced coverage (e.g., the region can be evenly lit with enough light). In Figure 8, we use 50, 200 and 500 agents to generate the transformation and observe that even if there are only 50 agents, they can still well manifest the underlying interpolated shape.

Source/target shapes with unequal areas. The original 2-Wasserstein distance based shape interpolation requires that the source shape and the target shape must be of the

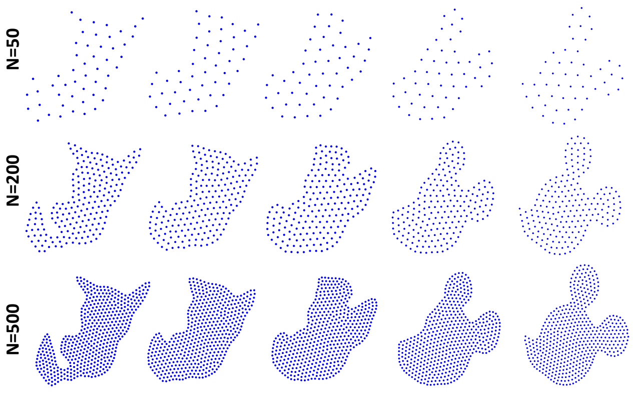


Figure 8. When the source/target shapes are given, users can arbitrarily specify the number of agents, and our algorithm can always report a visually smooth formation transformation.

same area. This requirement cannot be satisfied in a general formation transformation. Recall that we define a density function $\rho_k = 1/|\Omega_k|$ for the k-th intermediate shape. In this way, the area of the k-th frame, weighted by ρ_k, is exactly equal to 1, which guarantees that the 2-Wasserstein distance based barycentric interpolation can be well defined. In Figure 9(a), we show an example of unequal-size source and target shapes. Our algorithm can produce a visually pleasing transformation for this situation.

Multiple components, different topology structures, and highly non-convex contours. Since the source/target shapes vary greatly, the situation of formation transformation can be arbitrarily complex. It is highly desirable for a transformation algorithm to handle various challenging situations. We divide the challenging scenarios into three kinds including (a) source/target shapes with multiple components, (b) source/target shapes with different topology structures, and (c) source/target shapes with highly non-convex contours. It can be seen from Figure 9(b-e) that even under the challenging situations, our algorithm still has the ability to generate a visually pleasing transformation animation with uniform agent placement and smooth/rigid trajectories.

Obstacle avoidance. In some special scenarios, one may hope that the agents avoid an obstacle or pass through specified accessible area. Our algorithm explicitly controls the geometric domain for every intermediate frame (excluding the in-accessible area from the original domain) and thus is able to naturally guarantee that the agents cannot collide with obstacles. What's more, the agents keep a uniform placement when they move around the obstacles, and all the trajectories are guaranteed to be collision free; See Figure 10.

4.4. Simulation of real formation transformation show

The drone light show becomes a fashionable form of public entertainment showing up at sports competition, festival celebration and some other events. On the one hand, different lighting combinations can give different visual patterns. On the other hand, formation transformation can also offer visual pleasure to audience. To our best knowledge, the intelligent design/manipulation of an aesthetic formation transformation still remains a challenging problem so far. In fact, the study of this paper is exactly motivated by the three requirements of drone formation transformation, i.e., (1) visually pleasing intermediate formation, (2) uniform agent placement, and (3) smooth/rigid trajectories. We simulated the Pokémon drone-light/formation transformation show based on the proposed algorithm; See Figure 11 and the attached video. In spite of the complex geometric shapes, our algorithm can still satisfy these requirements at the same time with 500 UAVs or even fewer. Furthermore, our algorithm is very flexible for users to control:

- Users only need to specify the source/target shapes, without spending efforts in the initial placement of agents.
- Our algorithm owns the boundary alignment property and thus can still manifest the underlying shape with a quite limited number of agents, e.g., 30.
- Although our algorithm is able to automatically infer a sequence of visually pleasing intermediate shapes, we also allow users to edit them.

The above mentioned nice properties distinguish itself from the existing algorithm and show that it has a great potential of generating an automatic drone-light/formation transformation plan.

5. Conclusion and Future Work

In this article, we propose a collaborative optimization algorithm to generate a visually pleasing formation transform. Our transform plan has many nice features including (1) visually pleasing intermediate shapes, (2) strict boundary alignment, (3) high uniformity of agents, and (4) smooth trajectories, which is validated by extensive experimental results. Besides, we test our algorithm in some challenging scenarios and witness its superior performance. These scenarios include (1) source/target shapes with multiple connected components, (2) source/target shapes with different typology structures, and (3) existence of obstacles.

However, our algorithm, in its current form, still has some disadvantages. First, when users specify a large number of agents and a large number of intermediate frames, our algorithm requires a huge computational cost to yield the final transform plan. Secondly, we use the Wasserstein distance based barycentric interpolation [40] to infer the inter-

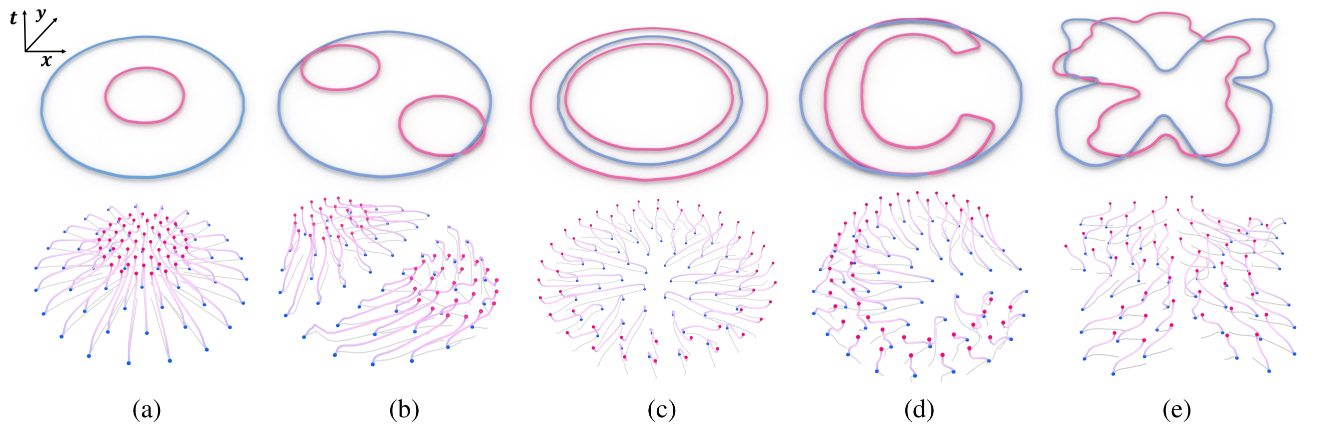


Figure 9. Source/target shapes with different areas (a), multiple components (b), different topology structures (c) and highly non-convex contours (d,e). The bottom row shows the spatio-temporal trajectory by lifting the 2D path (colored in grey) to the 3D space (colored in pink), where the vertical axis denotes the time dimension.

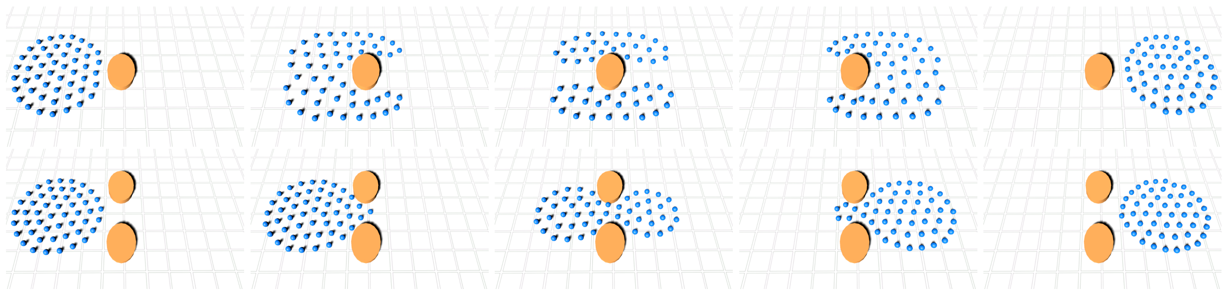


Figure 10. The first row shows how a group of agents avoids an obstacle during the formation transformation, while the second row shows how they run through the middle of two obstacles. All the snapshots are captured from the top view.

mediate formations frame by frame. However, the equally spaced time parameter $t \in [0, 1]$ cannot guarantee that the intermediate formations are also (visually) evenly spaced. In addition, our swapping based strategy for guaranteeing collision free trajectories, in its current form, only works in 2D. Therefore the algorithm cannot be directly extended to 3D space. We shall develop more techniques to cope with the difficulties in the future.

References

[1] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley. Multi-robot system for artistic pattern formation. In *2011 IEEE international conference on robotics and automation*, pages 4512–4517. IEEE, 2011. 3

[2] N. T. N. Anh, Z. J. Daniel, N. H. Du, A. Drogoul, and V. D. An. A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 371–383. Springer, 2011. 2

[3] I. L. Bajec, N. Zimic, and M. Mraz. Simulating flocks on the wing: the fuzzy approach. *Journal of Theoretical Biology*, 233(2):199–220, 2005. 1, 2

[4] B. Banerjee, A. Abukmail, and L. Kraemer. Advancing the layered approach to agent-based crowd simulation. In *2008 22nd Workshop on Principles of Advanced and Distributed Simulation*, pages 185–192. IEEE, 2008. 2

[5] A. Barnett, H. P. Shum, and T. Komura. Coordinated crowd simulation with topological scene analysis. In *Computer Graphics Forum*, volume 35, pages 120–132. Wiley Online Library, 2016. 2

[6] P. Charalambous and Y. Chrysanthou. The pag crowd: A graph based approach for efficient data-driven crowd simulation. In *Computer Graphics Forum*, volume 33, pages 95–108. Wiley Online Library, 2014. 2

[7] L. Conradt and C. List. Group decisions in humans and animals: a survey. *Philosophical transactions of The Royal Society B: biological sciences*, 364(1518):719–742, 2009. 2

[8] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004. 1

[9] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998. 2

[10] A. Golas, R. Narain, and M. C. Lin. Continuum modeling of crowd turbulence. *Physical review E*, 90(4):042816, 2014. 2

[11] Q. Gu and Z. Deng. Formation sketching: an approach to stylize groups in crowd simulation. In *Graphics Interface*, pages 1–8. Citeseer, 2011. 2

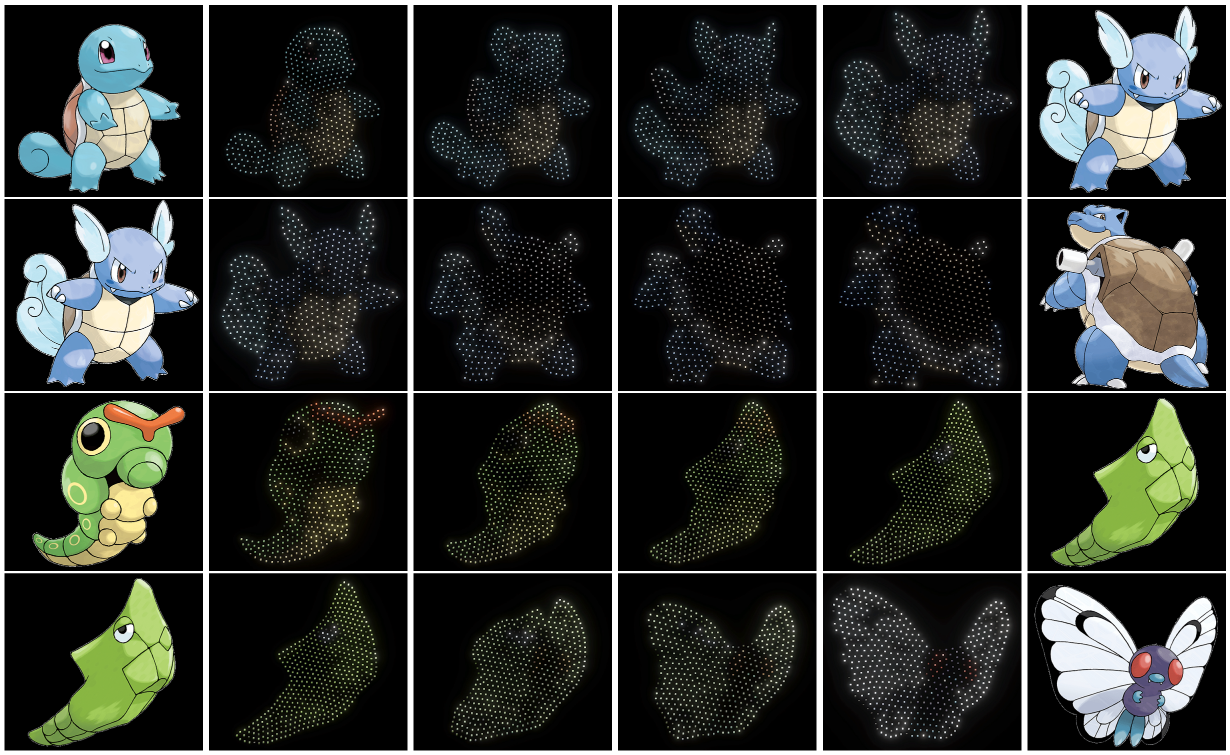


Figure 11. Simulation of Pokémon drone-light/formation transformation show with 500 UAVs; See the attached video for demo. It's worth mentioning that the first two rows can be synthesized into a complete two-stage deformation. The last two rows are in the same situation. In addition, each drone is equipped with a colored light whose color at an intermediate frame is interpolated by referring to the corresponding color setting at the source/target frames.

[12] Q. Gu and Z. Deng. Generating freestyle group formations in agent-based crowd simulations. *IEEE Computer Graphics and Applications*, 33:20–31, 2013. 2

[13] S. J. Guy, J. Van Den Berg, W. Liu, R. Lau, M. C. Lin, and D. Manocha. A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012. 2

[14] G. I. Hawe, G. Coates, D. T. Wilson, and R. S. Crouch. Agent-based simulation for large-scale emergency response: A survey of usage and implementation. *ACM Computing Surveys (CSUR)*, 45(1):1–51, 2012. 1

[15] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000. 2

[16] J. Henry, H. P. H. Shum, and T. Komura. Environment-aware real-time crowd control. In *SCA '12*, 2012. 2

[17] J. Henry, H. P. H. Shum, and T. Komura. Interactive formation control in complex environments. *IEEE Transactions on Visualization and Computer Graphics*, 20:211–222, 2014. 2

[18] R. L. Hughes. A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological*, 36(6):507–535, 2002. 2

[19] E. Ju, M. G. Choi, M. Park, J. Lee, K. H. Lee, and S. Takahashi. Morphable crowds. *ACM Transactions on Graphics (TOG)*, 29(6):1–10, 2010. 2

[20] I. Karamouzas, N. Sohre, R. Narain, and S. J. Guy. Implicit crowds: Optimization integrator for robust crowd simulation. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2, 8

[21] J. Kim, Y. Seol, T. Kwon, and J. Lee. Interactive manipulation of large-scale crowd animation. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014. 2

[22] M. Klotsman and A. Tal. Animation of flocks flying in line formations. *Artificial life*, 18(1):91–105, 2011. 1, 2

[23] J. Krause, G. D. Ruxton, and S. Krause. Swarm intelligence in animals and humans. *Trends in ecology & evolution*, 25(1):28–34, 2010. 2

[24] A. Krontiris, S. Louis, and K. E. Bekris. Simulating formations of non-holonomic systems with control limits along curvilinear coordinates. In *International conference on motion in games*, pages 121–133. Springer, 2010. 1

[25] T. Kwon, K. H. Lee, J. Lee, and S. Takahashi. Group motion editing. *ACM Transactions on Graphics (TOG)*, 27(3):1–8, 2008. 2

[26] K. H. Lee, M. G. Choi, and J. Lee. Motion patches: building blocks for virtual environments annotated with motion data. In *ACM SIGGRAPH 2006 Papers*, pages 898–906. 2006. 2

[27] B. Lévy and Y. Liu. L p centroidal Voronoi tessellation and its applications. In *SIGGRAPH 2010*, 2010. 5

[28] H. Mehrjerdi, J. Ghommam, and M. Saad. Nonlinear coordination control for a group of mobile robots using a virtual

structure. *Mechatronics*, 21(7):1147–1155, 2011. 1

[29] R. Narain, A. Golas, S. Curtis, and M. C. Lin. Aggregate dynamics for dense crowd simulation. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–8. 2009. 2

[30] S. Paris, J. Pettré, and S. Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Computer Graphics Forum*, volume 26, pages 665–674. Wiley Online Library, 2007. 2

[31] 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*, 17(2):244–254, 2010. 2

[32] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 189–198, 2009. 2

[33] G. Peyré and M. Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11:355–607, 2019. 3

[34] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira. Sensing and coverage for a network of heterogeneous robots. In *2008 47th IEEE conference on decision and control*, pages 3947–3952. IEEE, 2008. 1

[35] G. Qiao, S. Yoon, M. Kapadia, and V. Pavlovic. The role of data-driven priors in multi-agent crowd trajectory estimation. *arXiv preprint arXiv:1710.03354*, 2017. 2

[36] C. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM Siggraph Computer Graphics*, 21(4):25–34, 1987. 2

[37] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987. 1, 2

[38] J. Sewall, D. Wilkie, P. Merrell, and M. C. Lin. Continuum traffic simulation. In *Computer Graphics Forum*, volume 29, pages 439–448. Wiley Online Library, 2010. 1

[39] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha. The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, 27(4):696–706, 2011. 2

[40] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. 2, 3, 9

[41] S. Takahashi, K. Yoshida, T. Kwon, K. H. Lee, J. Lee, and S. Y. Shin. Spectral-based group formation control. In *Computer Graphics Forum*, volume 28, pages 639–648. Wiley Online Library, 2009. 2

[42] D. Thalmann. Crowd simulation. *Wiley Encyclopedia of Computer Science and Engineering*, 2007. 1, 2

[43] P. C. Tissera, A. M. Printista, and E. Luque. A hybrid simulation model to test behaviour designs in an emergency evacuation. *Procedia Computer Science*, 9:266–275, 2012. 2

[44] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Transactions on Graphics (TOG)*, 25(3):1160–1168, 2006. 2

[45] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer, 2011. 7, 8

[46] J. Van den Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008. 1

[47] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. 3

[48] I. von Sivers, A. Templeton, G. Köster, J. Drury, and A. Philippides. Humans do not always act selfishly: Social identity and helping in emergency evacuation simulation. *Transportation Research Procedia*, 2:585–593, 2014. 1

[49] X. Wang, X. Ying, Y. Liu, S.-Q. Xin, W. Wang, X. Gu, W. Müller-Wittig, and Y. He. Intrinsic computation of centroidal Voronoi tessellation (cvt) on meshes. *Comput. Aided Des.*, 58:51–61, 2015. 5

[50] T. Weiss, A. Litteneker, C. Jiang, and D. Terzopoulos. Position-based real-time simulation of large crowds. *Computers & Graphics*, 78:12–22, 2019. 2

[51] M. Xiong, M. Lees, W. Cai, S. Zhou, and M. Y. H. Low. Hybrid modelling of crowd simulation. *Procedia Computer Science*, 1(1):57–65, 2010. 2

[52] M. Xiong, S. Tang, and D. Zhao. A hybrid model for simulating crowd evacuation. *New Generation Computing*, 31(3):211–235, 2013. 2

[53] J. Xu, X. Jin, Y. Yu, T. Shen, and M. Zhou. Shape-constrained flock animation. *Computer Animation and Virtual Worlds*, 19(3-4):319–330, 2008. 1, 2

[54] M. Xu, Y. Wu, Y. Ye, I. Farkas, H. Jiang, and Z. Deng. Collective crowd formation transform with mutual information–based runtime feedback. In *Computer Graphics Forum*, volume 34, pages 60–73. Wiley Online Library, 2015. 3, 8

[55] L. Zheng, J. Zhao, Y. Cheng, H. Chen, X. Liu, and W. Wang. Geometry-constrained crowd formation animation. *Computers & Graphics*, 38:268–276, 2014. 3, 8