

# Mean-shift exploration in shape assembly of robot swarms

Received: 20 October 2022

Accepted: 6 June 2023

Published online: 13 June 2023

 Check for updates

Guibin Sun <sup>1,2</sup>, Rui Zhou<sup>1</sup>, Zhao Ma<sup>2</sup>, Yongqi Li<sup>2</sup>, Roderich Groß <sup>3</sup>, Zhang Chen <sup>4</sup> & Shiyu Zhao <sup>2,5,6,7</sup> 

The fascinating collective behaviors of biological systems have inspired extensive studies on shape assembly of robot swarms. Here, we propose a strategy for shape assembly of robot swarms based on the idea of mean-shift exploration: when a robot is surrounded by neighboring robots and unoccupied locations, it would actively give up its current location by exploring the highest density of nearby unoccupied locations in the desired shape. This idea is realized by adapting the mean-shift algorithm, which is an optimization technique widely used in machine learning for locating the maxima of a density function. The proposed strategy empowers robot swarms to assemble highly complex shapes with strong adaptability, as verified by experiments with swarms of 50 ground robots. The comparison between the proposed strategy and the state-of-the-art demonstrates its high efficiency especially for large-scale swarms. The proposed strategy can also be adapted to generate interesting behaviors including shape regeneration, cooperative cargo transportation, and complex environment exploration.

In nature, groups of insects and animals can self-assemble various spatial shapes that are functional for the groups adapting to the environments<sup>1–3</sup>. As a remarkable example, army ants can assemble shapes to transport food cooperatively or construct bridges using their bodies to overcome spatial gaps<sup>4,5</sup>. These shape assembly behaviors are generated spontaneously by local interactions among the individuals. They exhibit strong adaptability to individual faults and can be easily scaled up to groups of thousands or millions of individuals<sup>1,3</sup>.

The fascinating collective behaviors of biological systems have inspired extensive studies on shape assembly of robot swarms<sup>6–9</sup>. One class of strategies widely studied in the literature are based on goal assignment in either centralized or distributed ways<sup>10–12</sup>. Once a swarm of robots are assigned unique goal locations in a desired shape, the consequent task is simply to plan collision-free trajectories for the robots to reach their goal locations<sup>10</sup> or conduct distributed formation control based on locally sensed information<sup>6,13,14</sup>. It is notable that

centralized goal assignment is inefficient to support large-scale swarms since the computational complexity increases rapidly as the number of robots increases<sup>15,16</sup>. Moreover, when robots fail to function normally, additional algorithms for fault-tolerant detection and goal re-assignment are required to handle such situations<sup>17</sup>. As a comparison, distributed goal assignment can support large-scale swarms by decomposing the centralized assignment into multiple local ones<sup>11,12</sup>. It also exhibits better robustness to robot faults. However, since distributed goal assignments are based on locally sensed information, conflicts among local assignments are inevitable and must be resolved by sophisticated algorithms such as local task swapping<sup>11,12</sup>.

Another class of strategies for shape assembly that have also attracted extensive research attention are free of goal assignment<sup>18–21</sup>. For instance, the method proposed in ref. 18 can assemble complex shapes using thousands of homogeneous robots. An interesting feature of this method is that it does not rely on external global positioning systems. Instead, it establishes a local positioning system based

<sup>1</sup>School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. <sup>2</sup>School of Engineering, Westlake University, Hangzhou, China. <sup>3</sup>Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK. <sup>4</sup>Department of Automation, Tsinghua University, Beijing, China. <sup>5</sup>Research Center for Industries of the Future, Westlake University, Hangzhou, China. <sup>6</sup>Key Laboratory of Coastal Environment and Resources of Zhejiang Province, Westlake University, Hangzhou, China. <sup>7</sup>Westlake Institute for Advanced Study, Hangzhou, China. ✉ e-mail: [zhaoshiyu@westlake.edu.cn](mailto:zhaoshiyu@westlake.edu.cn)

on a small number of pre-localized seed robots. As a consequence of the local positioning system, the proposed edge-following control method requires that only the robots on the edge of a swarm can move while those inside must stay stationary. The method in ref. 19 can generate swarm shapes spontaneously from a reaction-diffusion network similar to embryogenesis in nature. However, this method is not able to generate user-specified shapes precisely. The method in ref. 21 can aggregate robots on the frontier of shapes based on saliency detection. The user-defined shape is specified by a digital light projector. An interesting feature of this method is that it does not require centralized edge detectors. Instead, edge detection is realized in a distributed manner by fusing the beliefs of a robot with its neighbors. However, since the robots cannot self-localize themselves relative to the desired shape, they make use of random walks to search for the edges, which would lead to random trajectories. Another class of methods that do not require goal assignment is based on artificial potential fields<sup>22–25</sup>. One limitation of this class of methods is that robots may easily get trapped in local minima, making it difficult to assemble nonconvex complex shapes.

Here, we propose a strategy for shape assembly of robot swarms based on the idea of mean-shift exploration: when a robot is surrounded by neighboring robots and unoccupied locations, it would actively give up its current location by exploring the highest density of nearby unoccupied locations in the desired shape. This idea does not rely on goal assignment. It is realized by adapting the mean-shift algorithm<sup>26–28</sup>, which is an optimization technique widely used in machine learning for locating the maxima of a density function. Moreover, a distributed negotiation mechanism is designed to allow robots to negotiate the final desired shape with their neighbors in a distributed manner. This negotiation mechanism enables the swarm to

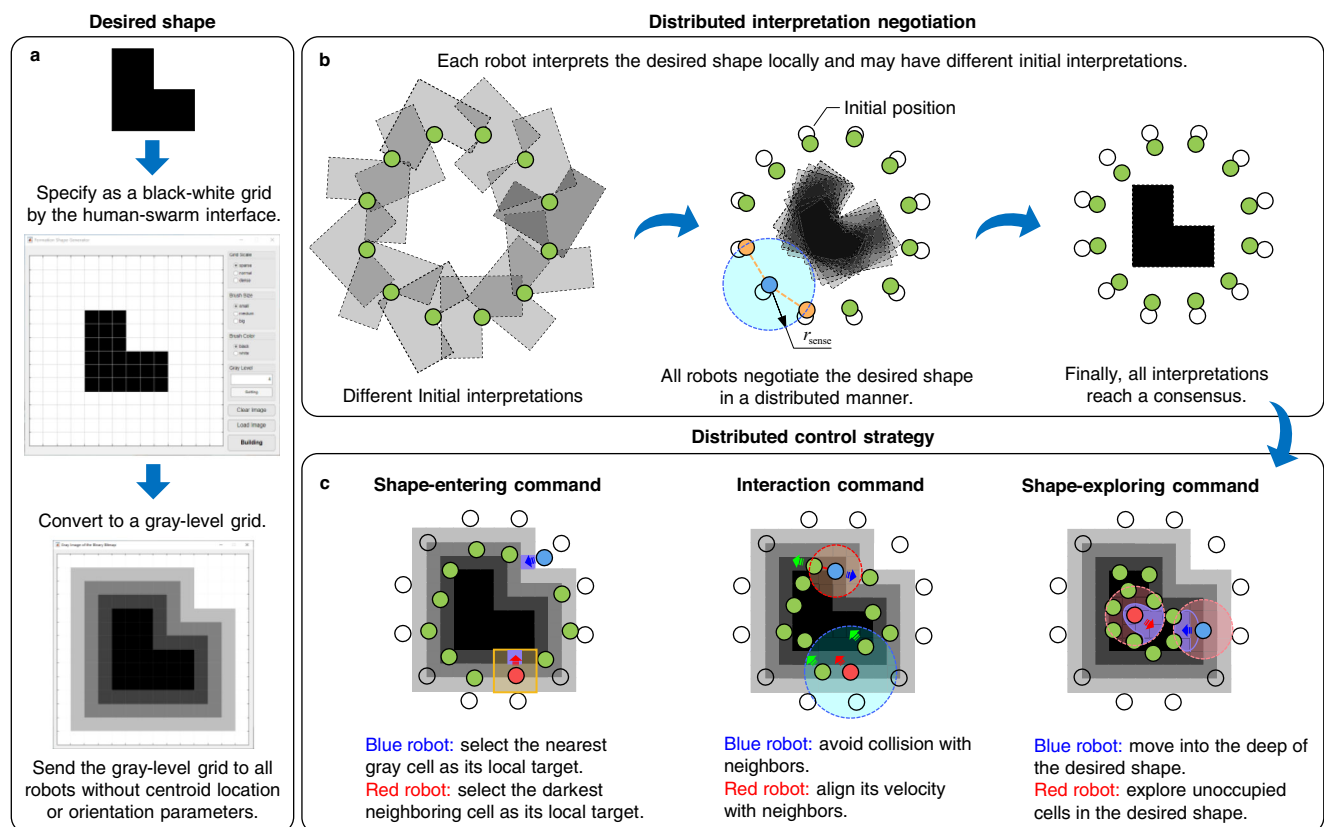
maneuver while maintaining a desired shape based on a small number of informed robots. The proposed strategy empowers robot swarms to assemble nonconvex complex shapes with strong adaptability and high efficiency, as verified by numerical simulation results and real-world experiments with swarms of 50 ground robots. The strategy can be adapted to generate interesting behaviors including shape regeneration, cooperative cargo transportation, and complex environment exploration.

## Results

The proposed shape assembly strategy consists of three components. The first is a human-swarm graphical interface that can specify user-defined shapes. The second is a distributed negotiation process that can autonomously reach an agreement among the robots on the final location and orientation of the swarm shape. The third, which is the core of the proposed strategy, is a distributed control algorithm. The three components are detailed as follows.

### A human-swarm graphical interface

The first step of shape assembly is to specify a desired geometric shape. We designed a human-swarm interface that allows a human operator to specify the desired shape by drawing or loading a binary graphical image (Fig. 1a). The graphical image is converted to a black-white binary grid, where the black cells in the grid correspond to the locations that should be occupied by the robots. The binary grid is then converted to a grayscale grid by the distance transformation algorithm<sup>29</sup> so that the influence of the desired shape is gradually expanded and the robots can move into the shape more smoothly (Fig. 1a). With this human-swarm interface, the operator can specify the desired shape in a graphical way without specifying the physical



**Fig. 1 | An illustration of the proposed shape assembly strategy. a** The human-swarm interface. The desired shape is specified via the interface and sent to all the robots as a nonparametric grid. **b** The robots have different initial interpretations of the translation and orientation of the desired shape. They gradually reach a

consensus via the proposed distributed negotiation algorithm. **c** Examples to illustrate the three control commands in the proposed control strategy. Details of the algorithms can be found in Methods and Sections 1 to 3 in the Supplementary Information.

parameters of the grid. The parameters such as the physical size of each cell can be autonomously generated. Moreover, the number of cells in the grid does not have to be the same as the number of robots because the proposed method can handle mismatches between the grid and robot numbers (Fig. 4 as shown later). Although the desired shape is represented as a discrete grid, the robots can move freely across the boundaries of the cells in the grid. More information can be found in Methods and Section 1 in the Supplementary Information.

### Distributed negotiation on the desired shape

Once the grayscale grid has been generated by the human-swarm interface, it is sent to all the robots and stored in each robot's memory. The memory sizes of the grids used in our work are around tens of kilobytes (details are given in Section 1 in the Supplementary Information), which is affordable for mainstream embedded computers nowadays. The Euclidean parameters of the desired shape including the position or orientation are not assigned manually. Instead, the robots negotiate with each other to autonomously determine them in a distributed manner. In particular, each robot first interprets the desired shape based on its local interest. For example, every robot may initially interpret itself as the center of the desired shape (Fig. 1b), which is locally optimal for each robot in the sense that they do not need to move because they are already inside the desired shape. While the initial interpretations of different robots may conflict, each robot negotiates its interpretation with its neighbors through local wireless communication by a distributed consensus algorithm. With the negotiation algorithm, all the robots would eventually reach a consensus on the translation and orientation of the final desired shape (Fig. 1b). By introducing a small number of informed robots, the swarm can reach a consensus on a user-specified trajectory of the desired shape. More information about shape negotiation is given in Methods and Section 2 and Section 6.1 in the Supplementary Information.

### Distributed control of shape assembly

The aim of each robot is to move into the desired shape. This process is highly dynamic because its goal location is time-varying due to the dynamical convergence process of its interpretation and, more importantly, the avoidance of inter-robot collision or goal location competition. In our method, each robot executes the same control strategy based on its locally sensed information so as to enter the desired shape while avoiding collision. In particular, the velocity command for the  $i$ th robot is

$$\mathbf{v}_i = \mathbf{v}_i^{\text{ent}} + \mathbf{v}_i^{\text{exp}} + \mathbf{v}_i^{\text{int}},$$

where  $\mathbf{v}_i^{\text{ent}}$ ,  $\mathbf{v}_i^{\text{exp}}$ , and  $\mathbf{v}_i^{\text{int}}$  represent the shape-entering, shape-exploring, and interaction velocity commands, respectively. The roles of the three velocity commands are explained as follows, whereas their mathematical expressions are given in Methods.

The shape-entering command  $\mathbf{v}_i^{\text{ent}}$  aims to steer robot  $i$  to the desired shape by seeking the darkest grid cells around them (Fig. 1c and Methods). More specifically, when robot  $i$  is too far away from the desired shape,  $\mathbf{v}_i^{\text{ent}}$  would drive the robot toward the closest gray cell. When robot  $i$  has entered the area of the gray cells,  $\mathbf{v}_i^{\text{ent}}$  would push the robot toward one of the neighboring gray cells that has the lowest gray level. In this way, the robot can gradually reach a black cell in the desired shape. It is notable that  $\mathbf{v}_i^{\text{ent}}$  would vanish once a robot steps into the desired shape area. In this case, the robot would stop moving and hence block those robots behind. To resolve this problem, the following velocity command  $\mathbf{v}_i^{\text{exp}}$  is designed so that the robots can continue moving into the deep of the desired shape.

The shape-exploring command  $\mathbf{v}_i^{\text{exp}}$ , which is realized by an adapted version of the mean-shift algorithm<sup>27</sup>, aims to push each robot into the desired shape and then explore unoccupied cells inside the desired shape (Fig. 1c and Methods). More specifically, there are two

cases. In the first case where robot  $i$  is near the edge of the desired shape so that the cells within its sensing radius are either black or gray,  $\mathbf{v}_i^{\text{exp}}$  would drive the robot toward the highest density of black cells. In this way, the robot is attracted into the shape. In the second case where robot  $i$  is already inside the desired shape so that the cells within its sensing radius are all black,  $\mathbf{v}_i^{\text{exp}}$  would drive the robot toward the direction with the highest density of the unoccupied black cells. A cell is defined as unoccupied if the distance between its center and any robot is greater than half of the robot's collision avoidance radius (F 3B). The shape-exploring command is the key to resolving inter-robot competition. When a robot is surrounded by neighboring robots and unoccupied locations, it would actively give up its current location by exploring the highest density of nearby unoccupied locations.

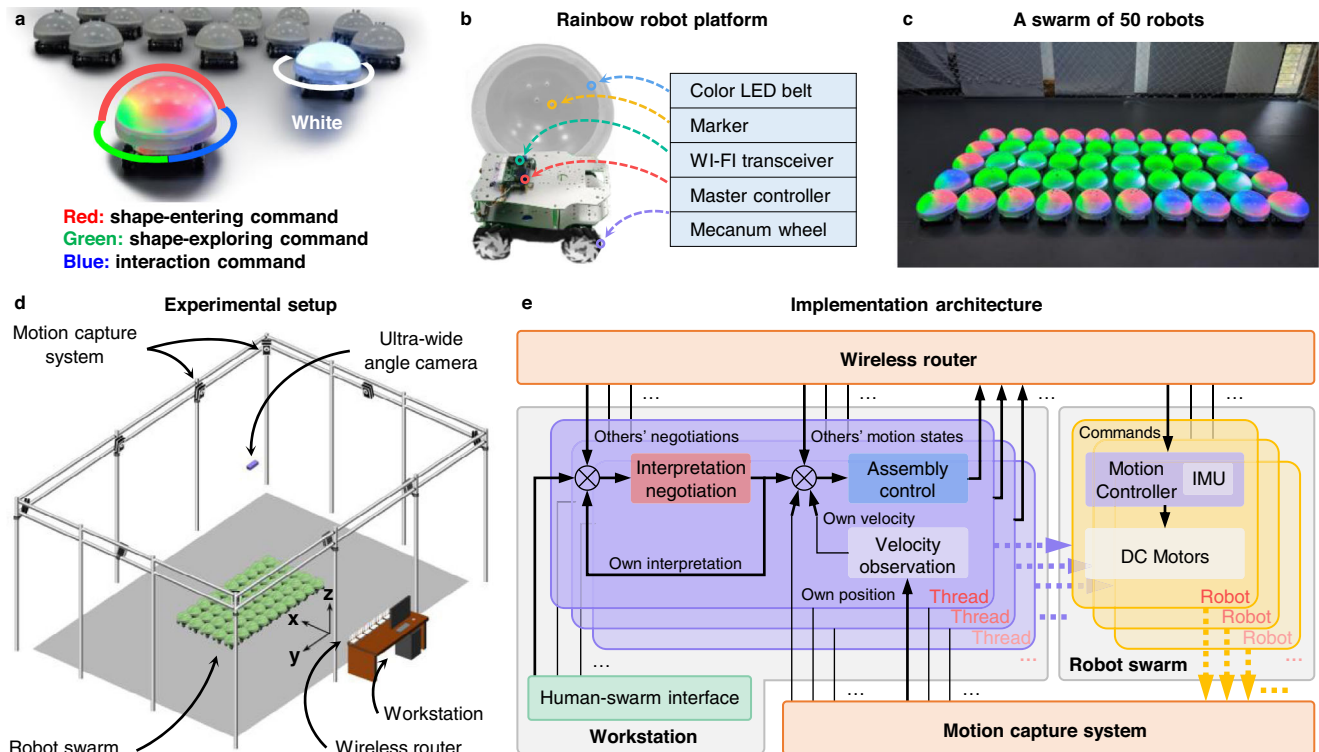
The interaction command  $\mathbf{v}_i^{\text{int}}$  consists of two sub-terms (Fig. 1c and Methods). The first is a collision-avoidance term that generates repulsive velocity commands when robot  $i$  is too close to its surroundings. The second is a velocity-alignment term that aligns the velocity of robot  $i$  with its neighbors. The velocity-alignment term plays two roles. First, it can reduce velocity mismatches among the robots to reduce the chances of inter-robot collision. Second, since the desired shape may maneuver across in the motion space, the velocity-alignment term is necessary for robot  $i$  to track the velocity of the moving desired shape.

### Platforms and experimental setup

We implemented the proposed strategy on a swarm of 50 holonomic wheeled robots (Fig. 2a, b). Details of the robotic platforms are given in Section 5.1 in the Supplementary Information. Each robot can exhibit its velocity command by using a LED belt inside the robot's dome. In particular, each LED can show four colors: red, green, blue, and white (Fig. 2a). The numbers of LEDs showing the four colors correspond to the magnitudes of  $\mathbf{v}_i^{\text{ent}}$ ,  $\mathbf{v}_i^{\text{exp}}$ ,  $\mathbf{v}_i^{\text{int}}$ , and  $\mathbf{v}_i$ , respectively (Section 5.1 in the Supplementary Information). The experiments were conducted in an indoor environment with the support of a motion capture system (Fig. 2c, d). To implement the proposed strategy, each robot needs to acquire certain information about its neighbors and sense its surroundings to identify unoccupied cells and obstacles. These functions are realized through a parallel multi-thread system in a distributed manner (Fig. 2e). More specifically, each robot is assigned an independent control thread running on the workstation and communicates with its thread using wireless routers. Although all the information is available in the workstation, the control command for each robot merely uses the local information that each robot is supposed to have. Details of the experimental setup are given in Section 5.2 in the Supplementary Information.

### Complex shape assembly

The proposed strategy is able to assemble nonconvex complex shapes. One representative complex shape is the snowflake as shown in Fig. 3a. The snowflake shape has 6 major branches and 18 minor branches, making the shape highly nonconvex and challenging to assemble. The proposed strategy can assemble the snowflake shape, which is quantitatively verified by the fact that the coverage rate converges to 100% (Fig. 3a). The coverage rate and other metrics to evaluate the assembly performance are defined in Methods. The shape-exploring control term of the strategy plays a key role in this process. In particular, when this term is disabled, the swarm fails to assemble the snowflake shape since many robots get trapped in local minima and the coverage rate drops sharply to 75% (Fig. 3a). The reason for the failure is that a robot would stop moving and hover near the boundary once it has entered the desired shape in the absence of this term. As a result, it may block those robots behind from entering the shape (Fig. 3b). By contrast, with this term, a robot would continue moving toward the deep of the desired shape after passing the boundary so that those robots behind it



**Fig. 2 | Platforms and implementation setup.** **a** The specifically designed Rainbow robots. Each robot can show different colors to reflect the ingredients of its control commands. **b** Hardware components of robot platform. See Section 5.1 in the Supplementary Information for more details. **c** A swarm of 50 Rainbow robots. **d** An

illustration of the experimental setup. The detailed functions of each experimental element are given in Section 5.2 in the Supplementary Information. **e** The multi-thread implementation architecture. Each robot corresponds to a unique control thread in the workstation, and the threads are executed in a parallel way.

can enter the shape smoothly (Fig. 3b). In addition to the shape-exploring term, the shape-entering term in the proposed strategy also plays a necessary role to steer all the robots into the desired shape. As shown in Fig. 3b, not all the robots can enter the shape in the absence of this shape-entering term even though the shape can still be fulfilled by a subset of the swarm. The proposed strategy exhibits smooth swarming motion in shape forming and switching tasks. As shown in the experimental results in Fig. 3c, the swarm can assemble different nonconvex shapes and switch from one to another smoothly.

In the following, the efficiency of the proposed method is compared with that of two state-of-the-art methods. The first is the assignment-based method in ref. 12 and the second is the assignment-free method in ref. 21. Both of the methods are decentralized. The method in ref. 12 and our proposed method require external global positioning, whereas the one in ref. 21 does not. The comparison results are presented in Fig. 4 and Supplementary Fig. 8. It is shown that, given the same initial configurations, the three methods have similar convergence times when the number of robots is as small as 20. However, as the number of robots increases, the convergence times of the state-of-the-art methods increase rapidly while that of the proposed one increases slightly. Specifically, when the number of robots is as large as 300, the convergence time of the proposed method is at least 20 times shorter than the others. The reason why the method in ref. 12 requires longer convergence time is that it executes local goal swaps constantly and the robots must move along the grid lines. By contrast, the proposed method does not rely on goal assignment due to the mean-shift exploration control. It also allows the robots to move continuously in the working space. The merit of the method in ref. 21 is that it does not rely on any global or local positioning system. However, since the robots do not know their positions relative to the desired shape, they make use of random walks to search for the edges

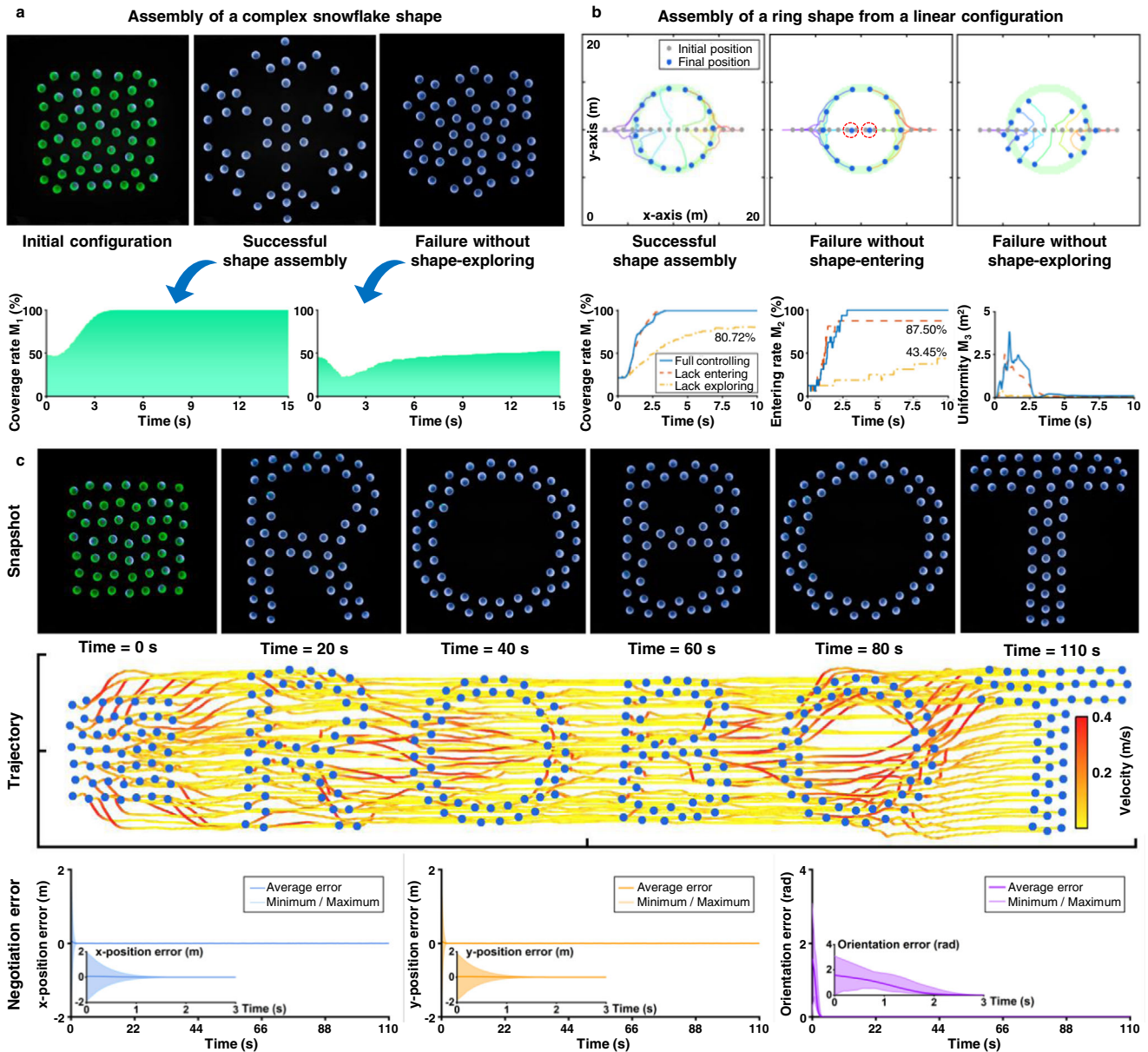
of the desired shape, which would lead to random and long trajectories.

### Adaptability to swarm scale variants

The proposed strategy is adaptive to the variant of robot number in a swarm. When some robots leave the swarm, the remaining robots can autonomously assemble the desired shape without any goal re-assignment or fault-tolerant control. A representative example is the shape regeneration behavior as demonstrated in Fig. 5, where the desired geometric shape is a starfish. After one arm of the starfish shape has been removed, the rest of the swarm spontaneously grows a new arm and assembles the starfish shape again. The adaptability is due to the shape-exploring term, by which the robots can actively search unoccupied cells and hence replace the roles of the removed robots.

Notably, the newly generated starfish has less number of robots than the original one, verifying that the proposed strategy can assemble the same shape with different numbers of robots. In fact, the strategy exhibits stable performance in the presence of mismatches between the number of robots, denoted as  $n_{\text{robot}}$ , and the number of black cells in the desired shape, denoted as  $n_{\text{cell}}$  (Fig. 5d). For the starfish example, it is shown by the statistical results in Fig. 5e that the proposed strategy can successfully assemble the desired shape with a wide range of  $n_{\text{cell}}/n_{\text{robot}}$ . Specifically, the coverage rate of robots for the desired shape is greater than 93% and the entering rate remains 100% when the ratio  $n_{\text{cell}}/n_{\text{robot}}$  varies from 0.45 to 128.06. As a consequence, the strategy avoids the requirement of the condition  $n_{\text{robot}} = n_{\text{cell}}$ , a condition widely adopted in shape assembly methods<sup>11</sup>, and hence makes the strategy more adaptive. In addition, when the swarm scale increases from  $n_{\text{robot}} = 16$  to 1024, the convergence time of the entire swarm increases mildly as shown in the rightmost sub-figure of Fig. 4E, verifying the high motion efficiency of the strategy.





**Fig. 3 | Complex shape assembly by the proposed strategy.** **a** Experimental results of 50 real Rainbow robots assembling a complex snowflake shape. The shape-exploring control term is necessary since the shape assembly task would fail in the absence of the term as shown in the rightmost subfigure. **b** Numerical simulation of 16 robots assembling a ring shape starting from an initial linear configuration. Both the shape-exploring and shape-entering control terms are

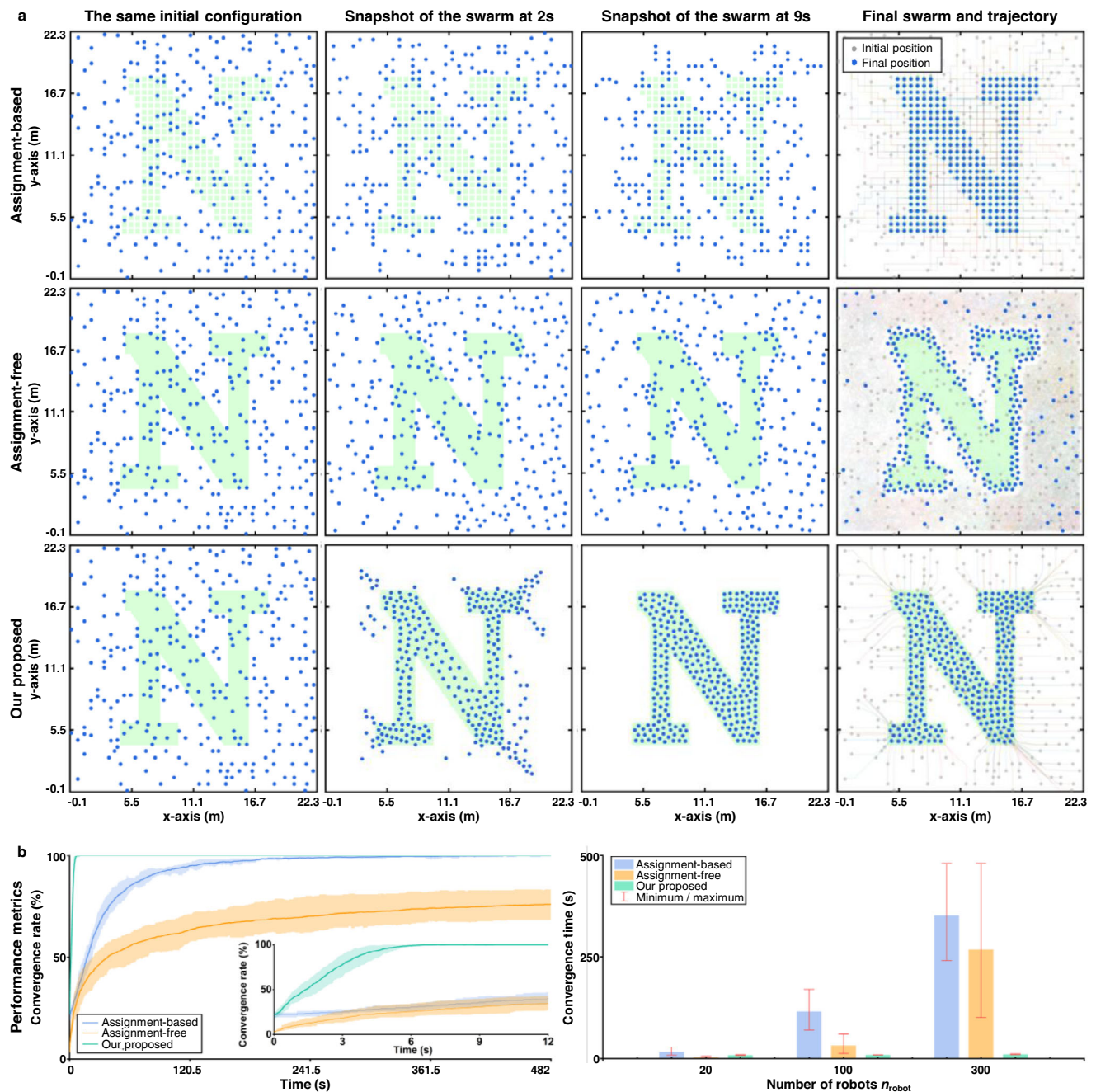
necessary since shape assembly would fail in the absence of either of them. **c** Experimental results of 50 real Rainbow robots assembling different shapes in a sequence. The distributed negotiation process among the robots converges fast. The shape switches smoothly due to the ability of active exploration and independence of goal assignment of the proposed strategy. See Supplementary Movies 1 and 2.

### Cooperative transportation and shape maneuvering

In nature, ants can surround and transport a piece of food that is much larger than their individual body size in a cooperative manner<sup>4</sup>. The proposed strategy can be applied to cooperative cargo transportation. To do that, we can specify a hollow shape centered at a cargo (Fig. 6a). When the robots assemble the hollow shape, they would encircle the cargo tightly. The consequent task is to steer the swarm to move collectively while maintaining the desired shape. In particular, we introduce a small number of informed robots that know the desired translational and rotational trajectory of the shape. The informed robots have counterparts in ant swarms<sup>4</sup>, where some leader ants can guide the others to transport food to a goal location. During the inter-robot negotiation process, the informed robots play a stubborn role by insisting on their knowledge of the desired

trajectory. The interpretations of the rest uninformed robots would gradually converge to the informed ones so that all the robots reach a consensus on the desired trajectory of the shape (see the details of the algorithm in Section 2 in the Supplementary Information). While the robots attempt to assemble the desired moving shape, they automatically transport the cargo encircled in the center (Fig. 6b). The velocity alignment component in the interaction control term of the proposed strategy is essential for all the robots tracking the maneuvering swarm shape.

The number of informed robots is required to be at least one. The more informed robots there are, the faster the negotiation process can converge. As shown in Supplementary Figure 6, seven informed robots are randomly selected out of 128 ones, which can reach a consensus on the time-varying translation and orientation of



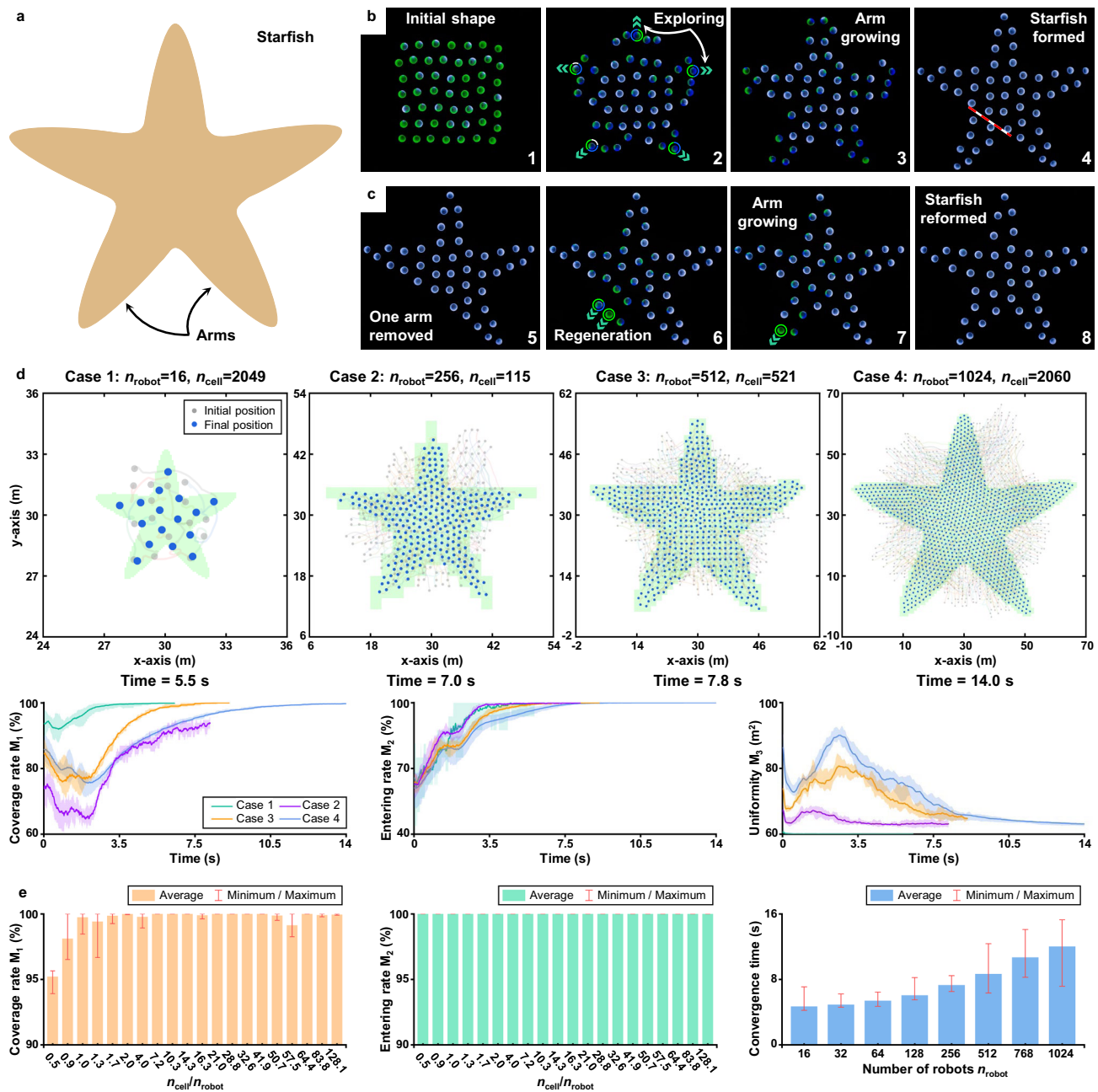
**Fig. 4 | Comparison between the proposed method, the decentralized assignment-based method in ref. 12 and the decentralized assignment-free method in ref. 21. a** Snapshots and trajectories of the shape assembly processes by the three methods. There are 300 robots assembling the shape of “N”. **b** Statistics of convergence rate and convergence time. Each average value is calculated based on 10 trials. For the proposed method and the assignment-based one in ref. 12, the convergence rate is defined as the ratio between the number of robots that are inside the shape and the total number of robots. The convergence time is defined as

the time when the convergence rate is equal to 100%. For the assignment-free method in ref. 21, the convergence rate is defined as the ratio between the number of robots that have reached the edge and the total number of robots. The width of the border, a parameter in the method in ref. 21, is set to zero so that the robots can aggregate evenly around the shape as much as possible. Since it is difficult for all the robots evenly distribute along the edge, the convergence time is defined as the time when the convergence rate reaches 70%. The parameters of the three algorithms are given in Supplementary Table 2.

the desired shape efficiently. As a consequence, the swarm can maneuver while the overall shape varies. Being able to track moving shapes is an important feature of the proposed strategy. By contrast, the state-of-the-art methods for homogeneous swarms are only applicable to static shapes<sup>10,12,18,19,30</sup>. Although there is a rich body of control-theoretic methods that can achieve maneuvering formations<sup>14,31,32</sup>, these methods require goal assignments or unique robot identities.

### Complex environment exploration

The proposed shape assembly strategy can be applied to environment exploration tasks, in which a swarm must evenly fulfill an environment while avoiding obstacles. As a representative example, a swarm can flood into a room through a narrow passage without getting trapped at the entrance (Fig. 7a). This example mimics the process of pedestrians entering a passenger elevator, which well demonstrates the idea of mean-shift exploration of the proposed



**Fig. 5 | Adaptability of the proposed strategy.** **a** The desired shape is a starfish<sup>33</sup>. **b**, **c** Experimental results of 50 real Rainbow robots assembling a starfish shape. If one arm of the starfish shape is removed, the swarm spontaneously grows a new arm without any goal re-assignment or fault-tolerant control thanks to the shape-exploring control term. See Supplementary Movie 3. **d** It is shown by numerical simulation that the strategy shows stable performance in the presence of the mismatch between  $n_{robot}$  and  $n_{cell}$ , quantitatively verified by the three metrics of coverage rate, entering rate, and uniformity. **e** Statistical results of 512 simulations

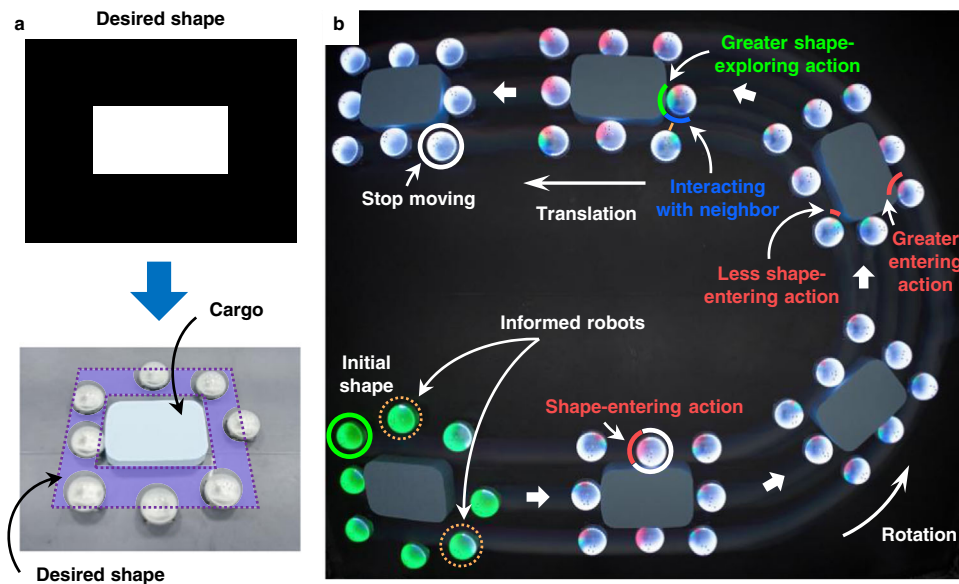
of the starfish assembly task given mismatched values of  $n_{robot}$  and  $n_{cell}$ . In each simulation,  $n_{robot}$  and  $n_{cell}$  take values respectively from {16, 32, 64, 128, 256, 512, 768, 1024} and {115, 521, 920, 1321, 1623, 2049, 2060, 2682}. Each pair of  $n_{robot}$  and  $n_{cell}$  further corresponds to eight trials with random initializations (including robots' locations and their interpretations of the desired shape). The  $8 \times 8 \times 8 = 512$  statistical results show that the proposed strategy exhibits stable performance in terms of the evaluation metrics in the presence of strong mismatches between  $n_{robot}$  and  $n_{cell}$ .

strategy: The pedestrians that enter the elevator first should move into the deep rather than staying near the entrance to block the people behind.

The proposed strategy can also be used to explore more complex environments such as a maze as shown in Fig. 7b. Although there are many nonconvex corners in the maze, the swarm can successfully fulfill the maze without getting deadlocked at any corner, verifying the strong exploration ability of the proposed strategy. The strategy can well balance the exploration speed and inter-robot connectivity.

That is, a robot inside the desired shape would stop exploring when it has no neighbors inside its sensing radius. As a result, the robots would not explore too fast to depart from the group. This feature is useful in practice because robots may have to explore an environment while keeping sufficiently close to others to maintain wireless connections. In addition, the system is resilient to robot additions or removals. For example, if the connectivity of the swarm is broken due to the removal of some robots, the isolated ones that are already in the maze would stop moving. As more and more robots





**Fig. 6 | Cooperative cargo transportation by a swarm of eight real Rainbow robots.** **a** The desired shape is specified as a hollow rectangle centered at the cargo. **b** The process of cooperative transportation by the swarm. When the robots assemble the desired shape, they would encircle the cargo tightly. Since the desired

shape is moving, the robots would transport the cargo while tracking the desired moving shape. The entire process is led by two informed robots. The color of each robot indicates its velocity command. See Supplementary Movies 4.

move into the maze, the isolated ones would get connected to the others again.

## Discussion

The distributed negotiation and control strategy proposed in this work is free of both centralized and decentralized goal assignment procedures. This strategy introduces a mean-shift exploration mechanism so that each robot can actively explore unoccupied locations around them to avoid inter-robot competition for the same locations. The mean-shift exploration is achieved through an adapted version of the mean-shift algorithm so that each robot keeps seeking the highest density of unoccupied cells. It is noticed that the mean-shift algorithm is also adopted in ref. 25 to achieve shape assembly. However, the mean-shift algorithm therein is used as an alternative to generating attractive and repulsive forces and hence is not able to resolve the inter-robot competition problem.

Experimental results verified that the proposed strategy can assemble highly nonconvex shapes and exhibits strong adaptability against mismatches between the number of robots and the number of cells in the desired shape. The strategy can also be applied to solve some challenging tasks such as cooperative cargo transportation and environment exploration with minimal modifications of the algorithms. It, therefore, provides a promising method for efficient and adaptive shape assembly tasks of robot swarms.

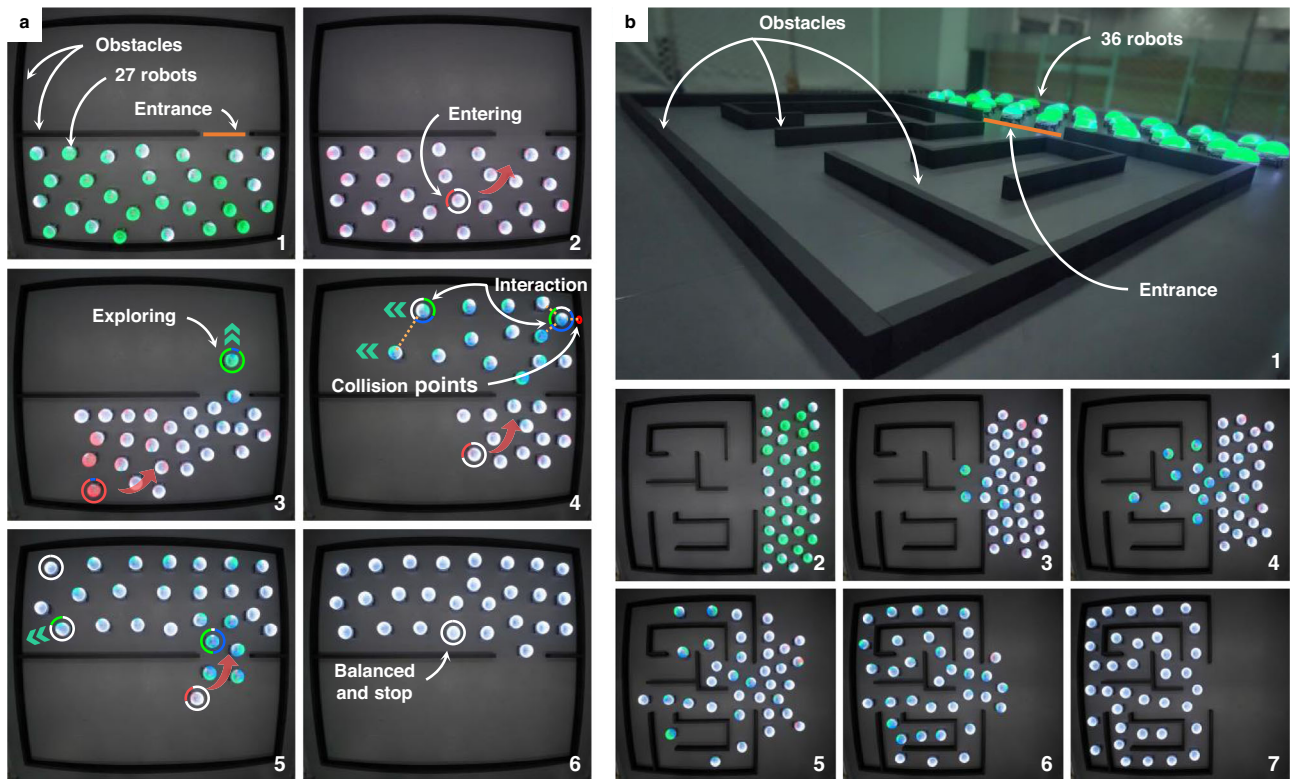
The desired shape in a shape assembly task is a global constraint for all the robots. The global constraint sets a fundamental requirement that certain global information must be used during the process of shape assembly. However, the required global information may exhibit in different forms in different methods. For example, in the proposed strategy and the state-of-the-art ones<sup>10,12,30</sup>, the required global information is that every robot has a common sense on a global reference frame. With this global information, every robot can evaluate its location relative to the desired shape. In practice, the global reference frame can be obtained by GPS in outdoor open spaces. If the global reference frame is unavailable, there must be alternatives to acquire global information. For example, the method in ref. 18 replaces the global reference frame by using a relative positioning system based on four seed robots.

The proposed negotiation and control algorithms are applied to assemble 2D shapes in this work. They can be extended to 3D scenarios since these algorithms are based on vector calculations where the position and velocity vectors are not restricted to 2D. However, a new human-swarm interface that can specify 3D shapes is required. In addition to user-specified geometric shapes, the proposed algorithms may be extended to assemble shapes specified in a natural way. For example, a swarm of robots may be used to clean an area of chemical pollutants in a water body. Here, the area of the densest pollutants can be treated as the desired shape and the density change in the nearby water can be regarded as the gradient. In this case, the shape is “specified” by the environment naturally and the proposed algorithms can be potentially applied.

One limitation of the proposed strategy is that it can only assemble shapes with single connected components. That is because the mean-shift exploration of each robot is based on local sensing of unoccupied locations. Since the sensing range of each robot is limited, if two disconnected shape components are far away from each other, it is difficult for a robot to explore unoccupied locations across different shape components. This problem may be potentially solved in several ways in practice. The first is to globally assign the robots to different components in advance. This way requires a centralized assignment process. A simple distributed solution is to add a line segment to bridge two disconnected components so that the entire shape becomes connected. In this way, each robot can explore the bridge to move from one shape component to another.

Moreover, we achieved the environment exploration tasks by treating them as shape assembly tasks, where the map of the environment is known and specified as the desired shape. While the map of the environment to be explored may not be available in practical search and rescue tasks, the proposed strategy can be applied if a minimal modification is made so that any regions unoccupied by obstacles are treated as desired shape regions to fulfill. Finally, although the algorithms proposed in this work are fully distributed, the physical robots in our experiments were provided data from a centralized motion capture system. It will be a promising research direction to develop robotic hardware systems with onboard sensing units to realize fully distributed shape assembly tasks of large-scale robot swarms.





**Fig. 7 | Environment exploration by the proposed strategy. a** Experiments of 27 real Rainbow robots flooding into a room through a narrow passage. This scenario mimics the process of pedestrians entering a passenger elevator: The pedestrians that enter the elevator first should move into the deep rather than staying near the

entrance to block the people behind. **b** Experiments of 36 real Rainbow robots exploring a complex maze without getting deadlocked at any nonconvex corners. See Supplementary Movies 5 and 6.

## Methods

### Specify a desired shape as a black-white grid

The proposed human-swarm interface allows the user to either load a pre-designed image or manually draw one in the workspace (Supplementary Figure 1). Either way, we can obtain a binary grid with black and white cells. Each cell of the grid is described by two basic parameters  $\rho$  and  $\xi_\rho$ . Here,  $\rho = (\rho_x, \rho_y)$  is the column and row indexes of the cell and  $\xi_\rho \in \{0, 1\}$  represents the color of the cell:  $\xi_\rho = 0$  if the cell is black and  $\xi_\rho = 1$  if the cell is white. The desired shape corresponds to the set of black cells. More information is given in Section 1 in the Supplementary Information.

### Convert the desired shape to a grayscale grid

The purpose to convert the black-white grid to a grayscale grid is to expand the influence scope of the desired shape so that the robots can move into the desired shape more smoothly. The gray conversion is based on the distance transformation algorithm<sup>29</sup>. Specifically, we expand the set of black cells out by  $h$  cells to generate an  $h$ -level grayscale grid. The gray level of each cell denoted as  $\xi_\rho$  is calculated based on a local parallel method described in ref. 29. In particular, for any cell  $\rho$  in the grid, its gray value is calculated iteratively by

$$\xi_\rho^k = \min_{\rho' \in \mathcal{M}_\rho} \left( \xi_{\rho'}^{k-1} + \frac{1}{h} \right), \quad k=1, 2, \dots, h-1. \quad (1)$$

Here, the superscript  $k$  denotes the  $k$ -th iteration. In every iteration, we conduct (1) for each cell. We need at most  $h-1$  iterations to obtain the final grayscale grid<sup>29</sup>. Here,  $\mathcal{M}_\rho$  is a  $3 \times 3$  mask centered at the cell  $\rho$ . Illustrative examples and more information can be found in Section 1 in the Supplementary Information.

### Parameterization of the size of each cell in the grid

The desired shape represented by a grid obtained from the human-swarm interface is merely graphical. Some important parameters of the desired shape are to be determined automatically. The first parameter is the size length of each cell denoted as  $\ell_{\text{cell}}$ . On the one hand, the total area of all the black cells is  $\ell_{\text{cell}}^2 n_{\text{cell}}$ . On the other hand, let  $r_{\text{avoid}}$  be the collision avoidance distance between the centers of any two robots. It is expected that the distance among each pair of robots in the desired shape is equal to  $r_{\text{avoid}}$  (Supplementary Figure 2c). As a result, the space occupied by each robot could be approximated by a circle with the center at the robot and the radius as  $r_{\text{avoid}}/2$ . Thus, the area occupied by all the  $n_{\text{robot}}$  robots is  $\pi(r_{\text{avoid}}/2)^2 n_{\text{robot}}$ . The above two areas are expected to be equal so that the robots could cover the desired shape (Supplementary Figure 2c):

$$\frac{\pi}{4} r_{\text{avoid}}^2 n_{\text{robot}} \approx \ell_{\text{cell}}^2 n_{\text{cell}}$$

from which  $\ell_{\text{cell}}$  can be solved as

$$\ell_{\text{cell}} = \sqrt{\frac{\pi n_{\text{robot}}}{4 n_{\text{cell}}}} r_{\text{avoid}}. \quad (2)$$

Equation (2) indicates that, when  $n_{\text{robot}}$  and  $r_{\text{avoid}}$  are given,  $\ell_{\text{cell}}$  is inversely proportional to  $\sqrt{n_{\text{cell}}}$ .

It is notable that  $n_{\text{cell}}$  is specified by the user when drawing a shape in the human-swarm interface. If  $n_{\text{cell}} = n_{\text{robot}}$ , equation (2) becomes  $\ell_{\text{cell}} = \sqrt{\frac{\pi}{4}} r_{\text{avoid}} \approx r_{\text{avoid}}$ , which means the side length of each cell is equal to the avoidance distance. In this case, each robot can approximately occupy one cell. However, our method does not necessarily require  $n_{\text{cell}} = n_{\text{robot}}$ , which is a strict condition. The proposed method allows

more general cases where  $n_{\text{cell}}$  and  $n_{\text{robot}}$  may mismatch (see examples in Fig. 5 and Supplementary Figure 5).

### Parameterization of the position and orientation of the desired shape

Another two parameters of the desired shape are the center position and orientation. There is no centralized assignment of the position or orientation. The robots can negotiate the two parameters in a distributed manner.

Before proceeding further, we need to introduce some necessary notations that will be used frequently later. Suppose there are  $n_{\text{robot}}$  mobile robots in  $\mathbb{R}^2$  and  $n_{\text{robot}} \geq 2$ . Each robot is regarded as a circle with the radius as  $r_{\text{body}}$ . Let  $\mathbf{p}_i \in \mathbb{R}^2$  be the position of the center point of robot  $i$  in a global coordinate frame. The dynamic model of each robot is assumed to be  $\dot{\mathbf{p}}_i = \mathbf{v}_i$  where  $i = 1, \dots, n_{\text{robot}}$ . Here,  $\mathbf{v}_i$  is the velocity command to be designed. When the distance between two robots is less than a threshold  $r_{\text{sense}}$ , the two robots could share information with each other. The information network defines an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , which consists of a vertex set  $\mathcal{V} = \{1, \dots, n_{\text{robot}}\}$  and an edge set  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  such that  $\mathcal{E} = \{(i, j) : \|\mathbf{p}_i - \mathbf{p}_j\| < r_{\text{sense}}, j \neq i\}$ . Here,  $\|\cdot\|$  is the Euclidean norm. The set of neighbors of robot  $i$  is  $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ .

Suppose  $\rho_o$  is the cell located closest to the center of the desired shape. Let  $\mathbf{p}_{\rho_o}$  and  $\mathbf{v}_{\rho_o}$  be the position and velocity of the center point of cell  $\rho_o$  in a global reference frame. Let  $\phi$  denote the orientation angle of the desired shape. Then, the position and orientation of the desired shape could be represented by  $\mathbf{p}_{\rho_o}$  and  $\phi$ , respectively.

Every robot has its own interpretation of  $\mathbf{p}_{\rho_o}$  and  $\mathbf{v}_{\rho_o}$ , denoted as  $\mathbf{p}_{\rho_o,i}$  and  $\mathbf{v}_{\rho_o,i}$ . Initially,  $\mathbf{p}_{\rho_o,i}(t_0) = \mathbf{p}_i(t_0)$  and  $\mathbf{v}_{\rho_o,i}(t_0) = \mathbf{v}_i(t_0)$ , which means that each robot initially treats itself as the center of the desired group shape. The interpretations of different robots can gradually reach a consensus by the following distributed negotiation algorithm:

$$\mathbf{v}_{\rho_o,i} = -\frac{c_1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \text{sign}(\mathbf{p}_{\rho_o,i} - \mathbf{p}_{\rho_o,j}) \|\mathbf{p}_{\rho_o,i} - \mathbf{p}_{\rho_o,j}\|^\alpha + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{v}_{\rho_o,j}. \quad (3)$$

There are two terms in (3). The first term in (3) is the average of the deficiency of the position interpretations between robot  $i$  and its neighbors. Here,  $\text{sign}(\cdot)$  and  $|\cdot|$  denote the sign and the absolute value of a real number. In addition,  $c_1 > 0$  and  $0 < \alpha < 1$  are two constant gains. The role of the first term is to drive  $\mathbf{p}_{\rho_o,i} \rightarrow \mathbf{p}_{\rho_o,j}$  where  $j \in \mathcal{N}_i$ . The second term is the average of the velocities of the neighboring robots. Its role is to drive  $\mathbf{v}_{\rho_o,i} \rightarrow \mathbf{v}_{\rho_o,j}$  where  $j \in \mathcal{N}_i$ . Since  $\alpha < 1$ , consensus can be achieved in a finite time. This is important for speeding up the negotiation process. The convergence analysis is given in Theorem 1 in the Supplementary Information.

Regarding the orientation negotiation, let  $\phi_i$  be the interpretation of  $\phi$  by robot  $i$ . The initial values of  $\phi_i$  could be randomly selected or based on task-oriented requirements. The distributed orientation negotiation algorithm is

$$\dot{\phi}_i = -\frac{c_2}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \text{sign}(\phi_i - \phi_j) |\phi_i - \phi_j|^\alpha + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \dot{\phi}_j \quad (4)$$

where  $c_2$  is a positive constant coefficient. This algorithm has the same structure as (3) and can be analyzed analogously.

If the desired shape is required to track a specified trajectory, we can introduce informed robots to achieve that. Details are given in Section 2 in Supplementary Information.

### Shape-entering velocity command

The shape-entering velocity command  $\mathbf{v}_i^{\text{ent}}$  in (5) aims to drive robot  $i$  into its interpretation of the desired shape, as depicted in

Supplementary Figure 3a. In particular, it is designed as

$$\mathbf{v}_i^{\text{ent}} = \kappa_1 \xi_{\rho_i} \frac{\mathbf{p}_{T,i} - \mathbf{p}_i}{\|\mathbf{p}_{T,i} - \mathbf{p}_i\|} + \mathbf{v}_{\rho_o,i} \quad (5)$$

where  $\mathbf{p}_i$  is the position of robot  $i$  and  $\mathbf{p}_{T,i}$  is a target location for robot  $i$  to move toward. Hence,  $\frac{\mathbf{p}_{T,i} - \mathbf{p}_i}{\|\mathbf{p}_{T,i} - \mathbf{p}_i\|}$  is a unit vector pointing from  $\mathbf{p}_i$  to  $\mathbf{p}_{T,i}$ . Therefore, the first term in (5) drives the robot toward its target location. Here,  $\kappa_1 \xi_{\rho_i}$  is a control gain where  $\rho_i$  is the index of the cell that is closest to  $\mathbf{p}_i$ ,  $\xi_{\rho_i}$  is the gray level of that cell, and  $\kappa_1 > 0$  is a constant gain. The second term  $\mathbf{v}_{\rho_o,i}$  in (5) is the local interpretation of robot  $i$  on the moving velocity of the entire shape. This term is necessary when the desired shape is moving. The calculation of the cell index  $\rho_i$  and the target location  $\mathbf{p}_{T,i}$  is given in Section 3 in the Supplementary Information.

### Shape-exploration velocity command

The shape-exploration velocity command  $\mathbf{v}_i^{\text{exp}}$  in (6) aims to push robot  $i$  into the desired shape and explore unoccupied black cells. Here, a cell is defined as occupied if the distance between its center and any robot is less than  $r_{\text{avoid}}/2$ ; and unoccupied otherwise.

In particular, we employ the mean-shift concept<sup>26</sup> to design  $\mathbf{v}_i^{\text{exp}}$  as

$$\mathbf{v}_i^{\text{exp}} = \frac{\sum_{\rho \in \mathcal{M}_i^{\text{neigh}}} \kappa_2 \psi(\|\mathbf{p}_{\rho} - \mathbf{p}_i\| / r_{\text{sense}}) (\mathbf{p}_{\rho} - \mathbf{p}_i)}{\sum_{\rho \in \mathcal{M}_i^{\text{neigh}}} \psi(\|\mathbf{p}_{\rho} - \mathbf{p}_i\| / r_{\text{sense}})} \quad (6)$$

which is a normalized weighted average of  $\mathbf{p}_{\rho} - \mathbf{p}_i$ , where  $\mathbf{p}_i$  is robot  $i$ 's position and  $\mathbf{p}_{\rho}$  is a position of a valid cell whose index is  $\rho$ . Here,  $\rho$  belongs to the set  $\mathcal{M}_i^{\text{neigh}}$ . The factor  $\kappa_2 \in \{\sigma_1, \sigma_2\}$  with  $\sigma_1, \sigma_2 > 0$  is a positive constant. The weight for  $\mathbf{p}_{\rho} - \mathbf{p}_i$  is a function  $\psi$  defined as

$$\psi(z) = \begin{cases} 1, & z \leq 0 \\ \frac{1}{2}(1 + \cos \pi z), & 0 < z < 1 \\ 0, & z \geq 1 \end{cases}$$

This function is monotonically decreasing from 1 to 0 as  $z$  increases. As a result, the weight  $\psi(\|\mathbf{p}_{\rho} - \mathbf{p}_i\| / r_{\text{sense}})$  is large when the distance between  $\mathbf{p}_{\rho}$  and  $\mathbf{p}_i$  is small. Hence, more weights are given to the cells that are closer to robot  $i$ .

The velocity command in (6) encourages robot  $i$  to explore unoccupied black cells. Here,  $\mathcal{M}_i^{\text{neigh}}$  is defined as the set of all the unoccupied black cells that are within the sensing radius  $r_{\text{sense}}$  of the robot (see Supplementary Figure 3b) and  $\kappa_2 = \sigma_2$ . In addition, when robot  $i$  is close to the boundary of the shape so that there are non-black cells within the sensing radius, on top of exploring unoccupied black cells, the velocity command also aims to push robot  $i$  into the desired shape. In this case,  $\mathcal{M}_i^{\text{neigh}}$  includes all the black cells no matter whether they are occupied or not and  $\kappa_2 = \sigma_1$ .

### Interaction velocity command

The aim of the interaction velocity command  $\mathbf{v}_i^{\text{int}}$  in (7) is to achieve collision avoidance and velocity alignment. To that end, it is designed as

$$\mathbf{v}_i^{\text{int}} = \underbrace{\kappa_3 \sum_{j \in \mathcal{N}_i \cup \mathcal{O}_i} \mu(\|\mathbf{p}_i - \mathbf{p}_j\|) (\mathbf{p}_i - \mathbf{p}_j)}_{\text{first term}} - \underbrace{\sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} (\mathbf{v}_i - \mathbf{v}_j)}_{\text{second term}} \quad (7)$$

where  $\kappa_3$  is a positive control gain.

The first term in (7) is a weighted sum of  $\mathbf{p}_i - \mathbf{p}_j$ , where  $\mathbf{p}_i$  is the position of robot  $i$  and  $\mathbf{p}_j$  is the position of the neighboring robot  $j$  or a collision point  $j$ . Here,  $j \in \mathcal{N}_i \cup \mathcal{O}_i$  where  $\mathcal{N}_i$  is the set of neighboring robots and  $\mathcal{O}_i$  is the set of collision points (see Supplementary

Figure 3c). The weight  $\mu(\|\mathbf{p}_i - \mathbf{p}_j\|)$  is defined as

$$\mu(\|\mathbf{p}_i - \mathbf{p}_j\|) = \begin{cases} \frac{r_{\text{avoid}}}{\|\mathbf{p}_i - \mathbf{p}_j\|} - 1, & \|\mathbf{p}_i - \mathbf{p}_j\| \leq r_{\text{avoid}} \\ 0, & \|\mathbf{p}_i - \mathbf{p}_j\| > r_{\text{avoid}} \end{cases}$$

By definition, when  $\|\mathbf{p}_i - \mathbf{p}_j\|$  is close to zero, the weight  $\mu(\|\mathbf{p}_i - \mathbf{p}_j\|)$  methods to infinity. When  $\|\mathbf{p}_i - \mathbf{p}_j\|$  is close to  $r_{\text{avoid}}$ , the weight  $\mu(\|\mathbf{p}_i - \mathbf{p}_j\|)$  monotonically decreases to zero. Since  $\mathbf{p}_i - \mathbf{p}_j$  is a vector pointing from  $\mathbf{p}_j$  to  $\mathbf{p}_i$ , the first term is a repulsive velocity that pushes robot  $i$  away from its surroundings to avoid a collision. The second term in (7) aims to align robot  $i$ 's velocity with its neighbors. Velocity alignment can help reduce the chance of inter-robot collision. In the meantime, it is necessary when the entire shape needs to track a moving trajectory.

**Performance metrics**

To evaluate the performance of the proposed strategy, we consider the following four metrics.

The first metric, coverage rate, is to evaluate the proportion of the black cells in the desired shape that are occupied by robots. Here, a cell is defined as occupied if the distance between its center and any robot is less than  $r_{\text{avoid}}/2$ . The metric is defined as

$$M_1 = \frac{n_{\text{occ}}}{n_{\text{cell}}} \times 100\%$$

where  $n_{\text{occ}}$  is the number of black cells occupied by robots. If all black cells are occupied, that is  $n_{\text{occ}} = n_{\text{cell}}$ , then  $M_1 = 100\%$ . If no black cells are occupied, then  $M_1 = 0$ .

The second metric, entering rate, is the proportion of the robots that have entered the desired shape. Here, a robot is said to be inside the shape if it occupies a black cell. The metric is defined as

$$M_2 = \frac{n_{\text{in}}}{n_{\text{robot}}} \times 100\%$$

where  $n_{\text{in}}$  is the number of robots inside the desired shape. If all robots are inside the desired shape, then  $M_2 = 100\%$ , and otherwise  $M_2 < 100\%$ .

The third metric, distribution uniformity, measures the distribution uniformity of the robots in the desired shape. Denote  $r_{\text{min},i} = \min_{j \in \mathcal{N}_i} \|\mathbf{p}_i - \mathbf{p}_j\|$  as the minimum distance from robot  $i$  to its neighbors. Then, the metric is defined as

$$M_3 = \sum_{i=1}^n (r_{\text{min},i} - \bar{r}_{\text{min},i})^2$$

where  $\bar{r}_{\text{min},i} = \frac{1}{n} \sum_{i=1}^n r_{\text{min},i}$ . If the distance between every pair of robots is the same, then  $M_3 = 0$ ; otherwise,  $M_3 > 0$ .

The fourth metric, velocity polarization, is to measure the polarization of the velocities of the robots. The metric is defined as:

$$M_4 = \frac{|\sum_{i=1}^n \mathbf{v}_i|}{\sum_{i=1}^n |\mathbf{v}_i|}$$

If all robots move with the same velocity, then  $M_4 = 1$ . If the robots' velocities are random, then  $M_4$  is close to zero.

**Statistics and reproducibility**

Data analysis was done by using the native functions of Originlab. All the statistical results are displayed in the format of "average  $\pm$  minimum/maximum". During the statistical analysis, no statistical method was used to predetermine sample size, and no data were excluded. All statistical trials used random initialization of robots' locations and their interpretations of the desired shape. In all bar

charts, the lower and upper edges of the bar represent the minimum and maximum, and the shaded region represents the average.

We implemented the simulation in Matlab. Details of the simulation setup and parameters used are given in Section 4 in the Supplementary Information. In our experiments, we use 50 holonomic wheeled robots. We implemented the experiments in Visual Studio using C++. Details of the experimental setup, robotic platform, and parameter settings are given in Section 5 in the Supplementary Information.

**Data availability**

The data used in this study are available from the corresponding author upon request.

**Code availability**

The code of the human-swarm interface used in this study is publicly available at <https://github.com/WestlakeAerialRobotics/Human-swarm-interface>(DOI: 10.5281/zenodo.7960508).

**References**

- Anderson, C., Theraulaz, G. & Deneubourg, J. L. Self-assemblages in insect societies. *Insectes Soc.* **49**, 99–110 (2002).
- Chen, Y. & Kolokolnikov, T. A minimal model of predator-swarm interactions. *J. R. Soc. Interface* **11**, 20131208 (2014).
- Vicsek, T. & Zafeiris, A. Collective motion. *Phys. Rep.* **517**, 71–140 (2012).
- Gelblum, A. et al. Ant groups optimally amplify the effect of transiently informed individuals. *Nat. Commun.* **6**, 7729 (2015).
- Berman, S., Lindsey, Q., Sakar, M. S., Kumar, V. & Pratt, S. C. Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems. *Proc. IEEE* **99**, 1470–1481 (2011).
- Ren, W., Beard, R. W. & Atkins, E. M. Information consensus in multi-vehicle cooperative control. *IEEE Control Syst. Mag.* **27**, 71–82 (2007).
- Brambilla, M., Ferrante, E., Birattari, M. & Dorigo, M. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* **7**, 1–41 (2013).
- Oh, K. K., Park, M. C. & Ahn, H. S. A survey of multi-agent formation control. *Automatica* **53**, 424–440 (2015).
- Oh, H., Shirazi, A. R., Sun, C. & Jin, Y. Bio-inspired self-organising multi-robot pattern formation: a review. *Rob. Autom. Syst.* **91**, 83–100 (2017).
- Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R. & Beardsley, P. Image and animation display with multiple mobile robots. *Int. J. Robot. Res.* **31**, 753–773 (2012).
- Sakurama, K. & Ahn, H. S. Multi-agent coordination over local indexes via clique-based distributed assignment. *Automatica* **112**, 108670 (2020).
- Wang, H. & Rubenstein, M. Shape formation in homogeneous swarms using local task swapping. *IEEE Trans. Robot.* **36**, 597–612 (2020).
- Anderson, B. D., Yu, C., Fidan, B. & Hendrickx, J. M. Rigid graph control architectures for autonomous formations. *IEEE Control Syst. Mag.* **28**, 48–63 (2008).
- Zhao, S. & Zelazo, D. Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity. *IEEE Control Syst. Mag.* **39**, 66–83 (2019).
- Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Res. Logistics* **2**, 83–97 (1955).
- Bertsekas, D. P. The auction algorithm: a distributed relaxation method for the assignment problem. *Ann. Oper. Res.* **14**, 105–123 (1988).
- Kamel, M. A., Yu, X. & Zhang, Y. Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: a review. *Annu. Rev. Control* **49**, 128–144 (2020).



18. Rubenstein, M., Cornejo, A. & Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **345**, 795–799 (2014).
  19. Slavkov, I. et al. Morphogenesis in robot swarms. *Sci. Robot.* **3**, eaau9178 (2018).
  20. Gauci, M., Nagpal, R. & Rubenstein, M. Programmable self-disassembly for shape formation in large-scale robot collectives, pp. 573–586 (Springer, Cambridge, 2018).
  21. Alhafnawi, M., Hauert, S. & O’Dowd, P. Self-organised saliency detection and representation in robot swarms. *IEEE Robot. Autom. Lett.* **6**, 1487–1494 (2021).
  22. Hsieh, M. A., Kumar, V. & Chaimowicz, L. Decentralized controllers for shape generation with robotic swarms. *Robotica* **26**, 691–701 (2008).
  23. Sabattini, L., Secchi, C. & Fantuzzi, C. Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation. *Auto. Robots* **30**, 385 (2011).
  24. Hou, S. P. & Cheah, C. C. Dynamic compound shape control of robot swarm. *IET Control Theory Appl.* **6**, 454–460 (2012).
  25. Vickery, C. & Salehi, S.A. A mean shift-based pattern formation algorithm for robot swarms. in *7th Int. Conf. Autom., Robot. Appl.*, pp. 16–20 (Prague, Czech Republic, IEEE, 2021).
  26. Fukunaga, K. & Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **21**, 32–40 (1975).
  27. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 790–799 (1995).
  28. Comaniciu, D. & Meer, P. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 603–619 (2002).
  29. Borgefors, G. Distance transformations in digital images. *Comput. Vis. Graph. Image Process.* **34**, 344–371 (1986).
  30. Morgan, D., Subramanian, G. P., Chung, S. J. & Hadaegh, F. Y. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. *Int. J. Robot. Res.* **35**, 1261–1285 (2016).
  31. Zhao, S. Affine formation maneuver control of multiagent systems. *IEEE Trans. Autom. Control* **63**, 4140–4155 (2018).
  32. Dong, X., Yu, B., Shi, Z. & Zhong, Y. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Trans. Control Syst. Technol.* **23**, 340–348 (2014).
  33. Yang, T. et al. A damage-tolerant, dual-scale, single-crystalline microlattice in the knobby starfish, protoreaster nodosus. *Science* **375**, 647–652 (2022).
- discussions. This work was supported by the STI 2030-Major Projects (Grant No. 2022ZD0208804), the Beihang Zhuoyue Postdoctoral Fellow Program (G.S.), the Hangzhou Key Technology Research and Development Program (Grant No. 20200416A16), and Research Center for Industries of the Future at Westlake University (Grant No. WU2022C027).

### Author contributions

S.Z., G.S., and R.Z. designed research; G.S., Z.M., Y.L., and S.Z. performed research; G.S., S.Z., R.G., and Z.C. analyzed data; S.Z., G.S., R.G., and Z.C. wrote the paper.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41467-023-39251-5>.

**Correspondence** and requests for materials should be addressed to Shiyu Zhao.

**Peer review information** *Nature Communications* thanks Nicola Bastianello, Sabine Hauert, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023

### Acknowledgements

We thank Fei Chen, Canlun Zheng, Weicong Zhang, Zian Ning, Ye Zheng, and Yin Zhang at Westlake University for their help in the experiments, and Chen Wei at Beihang University for helpful