IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Mean-Shift Shape Formation of Multi-Robot Systems without Target Assignment

Yunjie Zhang, Rui Zhou, Xing Li, and Guibin Sun

***Abstract*— The methods of shape formation in robot swarms are usually classified into two categories by whether assignment is used or not. The first is to use target assignment to assemble precise formation. However, the additional algorithm for re-assignment is required to handle unreasonable situations, which results in lower efficiency. The second, also called assignment-free method, is to use local behaviors to assemble formation, however, existing methods can rarely achieve the precise formation. In this paper, we present a distributed assignment-free algorithm to achieve the precise shape formation based on the mean-shift algorithm. Specifically, each target location in robot's perception range is equally regarded as a point of the mean-shift vector. Then, the weight value of each point is computed according to the density of the target location. Here, each robot obtains the density of the target location according to the distribution of its neighbors. Moreover, this density calculation also considers the states of non-neighboring robots via the hop-count algorithm, thus avoiding conflicts among robots. Subsequently, each robot can regard the calculated mean-shift vector as its control command. Finally, simulation results show that our algorithm can form precise shapes at least 8 times more efficient than the assignment-based approach and physical experiment results confirm that the proposed algorithm exhibits promising potential for practical applications.**

I. INTRODUCTION

Shape formation is an intriguing and enigmatic phenomenon that finds widespread occurrence across many natural processes. The well-known examples encompass the development of multi-organism biological systems [1], [2], as well as the shape assembly of army ants for cooperatively transporting food [3], [4]. Moreover, the shape formation phenomenon has extensive applications in many fields, such as smart warehouses [5], modular robots [6] and entertainment applications [7]. Due to the attractive and potential application value of shape formation, many researchers are actively pursuing its integration into multi-robot systems [8], [9]. Specifically, their primary objective is to assemble a swarm of robots into arbitrary target shapes.

A common strategy for shape formation is to decompose this problem into two subproblems: target assignment and collision-free movement [10]. In particular, this decomposition is often based on the condition that the target shape is represented by a set of target locations. Depending on whether the target assignment is tackled concurrently with the collision-free movement, this strategy can be further divided into the prior-assignment and dynamic-assignment approaches [11]. The prior-assignment approaches mean that each robot is initially assigned a unique target through a centralized or distributed manner [12]–[14]. For instance, the authors in [13] employ the Hungarian algorithm to minimize the traveled path by all robots at the initial time. However, the fixed results of assignment can restrict the maneuverability of the robots, thus leading to a significant challenge from the perspective of collision avoidance among robots [15].

Fig. 1. The physical experiment with 10 robots forming "R", "A", and "L".

Alternatively, the dynamic-assignment approaches try to achieve both target assignment and the collision-free movement towards the target locations concurrently. In [16], a global coordinator is required to compute the result of target assignment in each control cycle. Obviously, this centralized computation suffers from high computation cost when handling large-scale swarms. As a comparison, many researchers focus on the distributed dynamic-assignment approaches [10], [11], [17]–[19]. This approach decomposes the centralized assignment into multiple local-scale assignments. However, since local-scale assignments are based on locally sensed information, they may still generate conflict with each other. An obvious strategy to deal with the conflict is achieved via the local coordination according to complex rules [10], [18]. In addition, some researchers introduce the idea of exploration to avoid the conflict by allowing robots to actively explore unoccupied target locations [11], [19]. Although distributed target assignment has better computational efficiency for large-scale swarms, the complex procedure is also required to cope with potential conflicts, such as the local task swapping [10].

Another strategy focuses on removing the process of target assignment, that is, assignment-free approaches [20]–[26]. Many researchers employ the behavior-based strategy for each robot in the swarm [20]–[23]. For instance, the authors in [20] propose an edge-following-based strategy

[1]This work was supported in part by the STI 2030-Major Projects (Grant Nos.2022ZD0208804), China Postdoctoral Science Foundation (Grant Nos. 2023M740185), and the Postdoctoral Fellows of Beihang "Zhuoyue" Program, China. *(Corresponding author: Xing Li; Guibin Sun.)*

[2]Y. Zhang, R. Zhou, X. Li and G. Sun are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, BJ 100083 China (e-mail: zhangyunjie@buaa.edu.cn; zhr@buaa.edu.cn; lixingx@buaa.edu.cn; sunguibinx@buaa.edu.cn.).

for achieving a large set of connected shapes. However, this approach has a low efficiency since only the robots at the edge are allowed to move, that is, the low parallelism among robots. A more typical behavior-based approaches are artificial-potential-field-based approaches [24], [25]. Among these approaches, the target shape is transformed into the attraction region such that robots can be guided towards the shape. However, these approaches make a poor showing in shape completion rate because of the existence of local minima. In particular, the local minima will be more likely to appear as the number of robots increases. Recently, the authors in [26] have adopted the concept of selfless exploration to achieve efficient large-scale shape formation with less local minima and the results are compelling. One feature of these assignment-free approaches is that resulting shapes are not precise, limiting their applications in certain scenarios, such as the deployment of multi-sensor network [27], express sorting in logistics industry [28].

Motivated by above observations, we focus on the precise shape formation without target assignment for a large-scale multi-robot system. As mentioned before, the precise formation control generally requires to assign target location to each robot. However, the target assignment in a distributed manner can result in conflicts among robots which requires complex algorithms to deal with [19]. As a result, it is necessary to remove the procedure of target assignment for shape formation. Our previous work employs the mean-shift theory to achieve the shape formation without target assignment [26]. However, similar to the other existing assignment-free approaches, it cannot guarantee a precise shape formation. Therefore, the goal of this paper is to achieve the precise shape formation without target assignment. The contribution and novelty of this paper are summarized as follows:

1) We adopt the mean-shift theory to achieve the precise and assignment-free shape formation. In each control cycle, the robot simply calculates the mean-shift vector as its control command according to the local information, thus removing the target assignment. Different from our previous work in [26], the target density calculated in a distributed manner is employed to design weight values of mean-shift vector instead of the distance. The main advantage is that the density design takes into account both local distribution of neighboring robots and remote distribution of non-neighboring ones via hop propagation. In this way, conflicts among robots can be avoided.
2) We conduct both simulations and experiments to verify the effectiveness and performance of our proposed algorithm. We also present a simulation comparison of our method with two state-of-the-art algorithms [10], [26]. Simulation results show that our proposed algorithm can achieve precise formation control for large-scale robot swarms, and have at least 8 times improvement in efficiency compared with the assignment-based algorithm [10]. Furthermore, the physical experiment results (shown in Fig .1) confirm that the proposed algorithm exhibits promising potential for practical applications.

The remainder of this paper is organized as follows. In Section II, we show the problem statement. Then, methodology to cope with this problem is detailed in Section III. The simulation and experimental results are presented in Section IV. Finally, we conclude this paper in Section V.

II. Problem Statement

Consider n mobile robots with the same communication range r_{s} and maximum velocity $v_{\max}$. Each robot is considered as a circle with its radius as r_{body} and the dynamics of robot i can be described by

$$\dot{p_i} = v_i, \;\; i = 1, 2, \ldots, n \tag{1}$$

where $p_i,\ v_i \in \mathbb{R}^2$ represent the position and the velocity control input, respectively. For two robots i and j, they can establish a connection if and only if $\|p_i - p_j\| \leq r_{\mathrm{s}}$. Here, we utilize an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to describe the interaction topology among robots, where the vertex set $\mathcal{V} = \{1, 2, ..., n\}$ denotes n robots, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ includes all connections among robots. Let $\mathcal{N}_i = \{j \in \mathcal{V} : \|p_i - p_j\| \leq r_{\mathrm{s}}\}$ denote the neighbor set of robot i, where the operator $\|\cdot\|$ represents the Euclidean norm on $\mathbb{R}^2$.

In this paper, the control objective is to guide a swarm of robots from an arbitrary initial configuration to a desired precise shape while ensuring collision avoidance. In particular, the desired shape is represented by a set of target locations $\mathcal{Q} = \{q_1, q_2, ..., q_m\}$ and the number of target locations is equal to the number of robots (i.e., $m = n$) for simplifying the issue. Then, the objective of this paper can be expressed mathematically as:

- $\exists t_{\max} > 0$ such that at any time $t > t_{\max}$, it holds that: $\forall i \in \mathcal{V}$, $\exists q_k \in \mathcal{Q}$ s.t. $p_i(t) = q_k$,
- $\forall t \geq 0$, for any two robots $i \neq j$, $\|p_i(t) - p_j(t)\| > 2r_{\mathrm{body}}$.

In the previous work [26], the mean-shift theory is employed to implement shape formation control without target assignment. However, it can only drive a swarm of robots evenly distributed in a desired shape, which is denoted by a binary image. In other words, it cannot achieve a precise shape formation denoted by a series of target locations. In this paper, we focus on the precise shape formation. Due to the high efficiency of mean-shift theory, we continue to employ it to deal with the precise shape formation control. Meanwhile, this deployment removes the procedure of target assignment. In general, the mean-shift theory is an optimization technology widely used in machine learning for locating the maxima of a density function [29]. The mean-shift vector

$$m(x) = \frac{\sum\limits_{x_i \in D} \psi(\|x - x_i\|) x_i}{\sum\limits_{x_i \in D} \psi(\|x - x_i\|)} - x \tag{2}$$

always points towards the local location with the maximum density, where $D \subset \mathbb{R}^n$ is the vector space, and $\psi(\cdot)$ is a kernel function that denotes the weight. Fig. 2 illustrates the meaning of the mean-shift theory. In this paper, each target

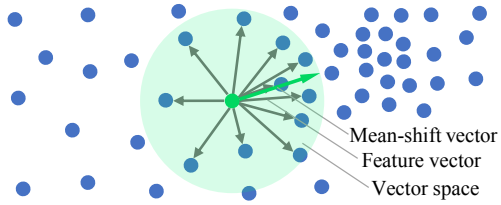

Fig. 2. Illustration of the mean-shift theory.

location in robot i's sensing range is regarded as a point of mean-shift vector, and the density of target location is employed to design the weight value, that is,

$$m(p_i) = \frac{\sum\limits_{q_k \in \mathcal{Q}_i} \psi(\rho_{i,k}) q_k}{\sum\limits_{q_k \in \mathcal{Q}_i} \psi(\rho_{i,k})} - p_i \tag{3}$$

where $\mathcal{Q}_i = \{q_k \in \mathcal{Q} : \|q_k - p_i\| \leq r_s\}$ represents the target locations within the sensing range of robot i, and $\rho_{i,k}$ is the density of location q_k calculated by robot i via the distributed manner. It can be seen from (3) that with the adaptation of the mean-shift method to shape formation, the robot would actively give up its current location with higher density (i.e., location with more robots nearby) by exploring the location with the lowest density. As a result, the mean-shift method has the advantage of avoiding conflicts among robots.

III. Approach

In this section, we first provide the calculation of target density for each robot. Here, the target density is utilized to design the weight values in mean-shift theory. Furthermore, to deal with the local minima, we introduce the hop-count gradient direction to improve the design of target density. Finally, the mean-shift control command with the target density is presented to realize our control objective in Section II. The detailed description is shown in Algorithm 1.

A. Local-State Density Design

Each robot i can calculate the local density of target location $q_k \in \mathcal{Q}_i$ based on the positions of all neighbors in $\mathcal{N}_i$. Specifically, robot i considers that neighbors closer to target have a greater influence on its density, and this influence diminishes as the distance increases. Hence, the local density $\rho_{i,k}$ of target location q_k for robot i can be computed as the sum of influence generated by all neighbors:

$$\rho_{i,k} = \begin{cases} \sum\limits_{j \in \mathcal{N}_i} \kappa_1 g(\|p_j - q_k\|), & |\mathcal{N}_i| \neq 0 \\ 0, & |\mathcal{N}_i| = 0 \end{cases} \tag{4}$$

where $g(\cdot)$ is a function that can calculate the influence of neighbors on the density of the target location, and $\kappa_1 > 0$ is a scaling factor, and $|\cdot|$ represents the number of elements in a countable set. This paper sets the function $g(\cdot)$ as follows:

$$g(d) = \begin{cases} \exp\left(-\frac{d^2}{2\sigma^2}\right), & d \leq r_s \\ 0, & d > r_s \end{cases} \tag{5}$$

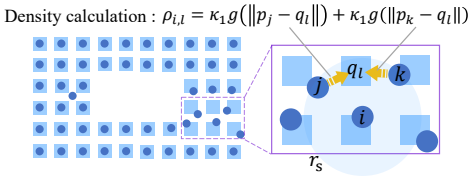

Fig. 3. Illustration of robot i calculating the local density $\rho_{i,l}$ of target q_l. Each robot is shown as a blue circle and each target location is shown as a blue square.

Algorithm 1: General Behavior of Robot i.

Input: $\mathcal{Q}$: The set of target locations;
while *true* **do**
  if $|\mathcal{Q}_i| \neq 0$ **then**
    Calculate the target location density $\rho_{i,k}$ for all q_k in $\mathcal{Q}_i$ by (4);
    if i *is a blocked robot* **then**
      Set hop count $h_i = 0$ and broadcast it;
    end
    else
      Update and broadcast the hop count h_i by (7);
      Compute the hop-count gradient ∇h_i by (8);
      Compute the improved target location density $\rho^*_{i,k}$ for all q_k in $\mathcal{Q}_i$ by (9);
    end
  end
  Calculate the control command v_i by (11);
end

where $\sigma > 0$ represents the bandwidth of the Gaussian kernel, determining the sensitivity of target density to changes in distances between neighbors and targets. As can be seen, the formula (5) essentially utilizes the Gaussian kernel function to transform the distance $d_{jk} = \|p_j - q_k\|$ into the contribution to local target density. Fig. 3 depicts the local density calculation of robot i for the target location q_l.

According to the computation of target density, the fewer robots are near the target locations, the lower its density is. If the kernel function in (3) is monotonically decreasing with respect to the density, then the target location with smaller density contributes greater in the mean-shift vector. As a result, the robot will actively explore the unoccupied target locations, which is critical for large-scale shape formation [19]. However, as the majority of target locations become occupied by the robots, the effect of this exploration gradually decreases, resulting in longer convergence time or even failure to converge. As depicted in Fig. 4 (left), when the shape is about to complete, the robot l is in a state of equilibrium, thus resulting in the robot j being blocked. To cope with this issue, a hop-count algorithm for transmitting the global information is introduced to accelerate and ensure system convergence.

B. Hop-Count Density Design

We here detail the application of the hop-count algorithm in improving the density design. The hop-count algorithm, also known as the gradient algorithm, is often employed to improve the goal selection by introducing global information

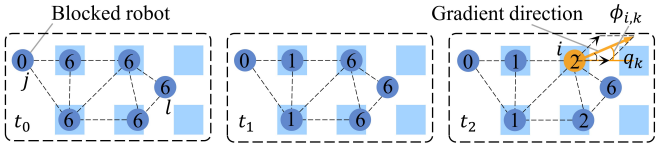

Fig. 4. Illustration of the process of our improved density design. At time t_0, the blocked robot sets its hop-count value to 0, and then each robot updates the hop-count value according to the neighboring hop-count information. At time t_2, robot i calculates the hop-count gradient according to (8) and calculates $\phi_{i,k}$ according to (10).

[20]. In general, two robot roles are involved in the hop-count algorithm, the common robot and the anchoring robot. Both roles transmit a single message containing a position and a hop-count value h. This common hop-count algorithm can be modified in the following way to improve the density design.

First, robot i will take on the anchoring role when all targets in its sensing range are approximately occupied, i.e.,

$$\forall q_k \in \mathcal{Q}_i, \exists j \in \mathcal{N}_i \text{ s.t. } \|p_j - q_k\| \leq \delta_1 \tag{6}$$

where $\delta_1 \geq 0$ is a given constant. In this paper, we also call the anchoring robot as the blocked robot. The robot i is a common robot if the condition (6) is not satisfied. The blocked robot i sets its own hop-count value to 0 and broadcasts it. The common robot i updates its own hop-count value with neighboring hop-count values as follows:

$$h_i = \min_{j \in \mathcal{N}_i} h_j + 1 \tag{7}$$

and then broadcasts this message. Fig. 4 illustrates the updating process of hop-count values. In particular, we set $h_i(t_0) = n$ for each robot at the initial time. Moreover, the hop-count value of a robot is set as n when it exceeds n.

Then, based on the introduced hop-count values, we can define the hop-count gradient for improving the density design, which can be computed as follows:

$$\nabla h_i = (h_i - h_i^{\min}) \sum_{j \in \mathcal{N}_i^h} \frac{p_i - p_j}{\|p_i - p_j\|} \tag{8}$$

where the set $\mathcal{N}_i^h = \{j \in \mathcal{N}_i : h_j = h_i^{\min}\}$ denotes the neighbors with the minimum hop-count values $h_i^{\min}$. As can be seen from (8), the gradient direction essentially characterizes the relative orientation of the neighbors with minimum hop-count values to the robot i, implying the direction of blocked robots, as depicted in Fig. 4. Subsequently, the improved density design for i can be derived by

$$\rho_{i,k}^* = \omega_1 \rho_{i,k} + \omega_2 \phi_{i,k} \tag{9}$$

where $\omega_1, \omega_2 > 0$ are weight coefficients. The term $\phi_{i,k}$ in (9) is the improved density component, which is defined by

$$\phi_{i,k} = \begin{cases} \arccos\left(\dfrac{\langle \nabla h_i, q_k - p_i \rangle}{\|\nabla h_i\| \cdot \|q_k - p_i\|}\right)/\pi, & \|q_k - p_i\| \neq 0 \\ \dfrac{1}{2}, & \|q_k - p_i\| = 0 \end{cases} \tag{10}$$

where $\langle a, b \rangle$ represents the inner product of the vector a and b. The formula (10) implies that the magnitude of $\phi_{i,k}$ mainly depends on the angle between the hop-count gradient and the vector $q_k - p_i$, as shown in Fig. 4 (right). It should be noted that $\phi_{i,k} = 0$ when the hop-count gradient $\nabla h_i = 0$.

In terms of the above procedure, our density improvement essentially allows the blocked robot to exert influence on the target location. To be specific, our density design combines the local and global information compared to the algorithm in [26] that designs the mean-shift weights based solely on distance, thus enhancing the exploration intensity and avoiding conflicts among robots. In addition, the computation of target density is performed in a fully distributed manner, which is more suitable for a large-scale multi-robot system.

C. Distributed Shape Formation Controller

The complete control command of robot i is designed by

$$v_i = v_i^a + v_i^\rho + v_i^c. \tag{11}$$

When the robot is outside the shape, the component v_i^a can guide the robot into the shape. When the robot is in the shape, the mean-shift component v_i^ρ drives the robot to occupy the target location without target assignment. The component v_i^c is employed to avoid collisions among robots in the whole process.

1) Guidance Velocity: This component v_i^a can be derived as follows:

$$v_i^a = \begin{cases} \kappa_a \dfrac{q_i^{\mathrm{m}} - p_i}{\|q_i^{\mathrm{m}} - p_i\|}, & |\mathcal{Q}_i| = 0 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where κ_a is a positive constant and the location q_i^{m} is the closest target location to robot i, that is,

$$q_i^{\mathrm{m}} = \arg\min_{q_k \in \mathcal{Q}} \|p_i - q_k\|. \tag{13}$$

As can be seen from (12) and (13), this component will guide the robot towards the nearest target location at a fixed constant speed κ_a.

2) Mean-Shift Velocity: With the computed density, this component based on the mean-shift vector is designed as follows:

$$v_i^\rho = \kappa_\rho \frac{\sum_{q_k \in \mathcal{Q}_i} \psi(\rho_{i,k}^*/\rho_0)(q_k - p_i)}{\sum_{q_k \in \mathcal{Q}_i} \psi(\rho_{i,k}^*/\rho_0)} \tag{14}$$

where $\kappa_\rho > 0$ is a positive constant, and $\rho_0 > 0$ is the density limit to ensure that the weight of the fully occupied target location is 0. The weight $\psi(\cdot)$ defined by

$$\psi(z) = \begin{cases} 1, & z \leq 0 \\ \dfrac{1}{2}(1 + \cos \pi z), & 0 < z < 1 \\ 0, & z \geq 1 \end{cases} \tag{15}$$

is used to measure the contribution of the density of target q_k to v_i^ρ. As can be seen from (15), the smaller the density is, the greater the weight value is. In other words, this mean-shift velocity can drive the robot to actively explore the target location with lower robots. In particular, this process effectively removes the target assignment. Instead, each robot

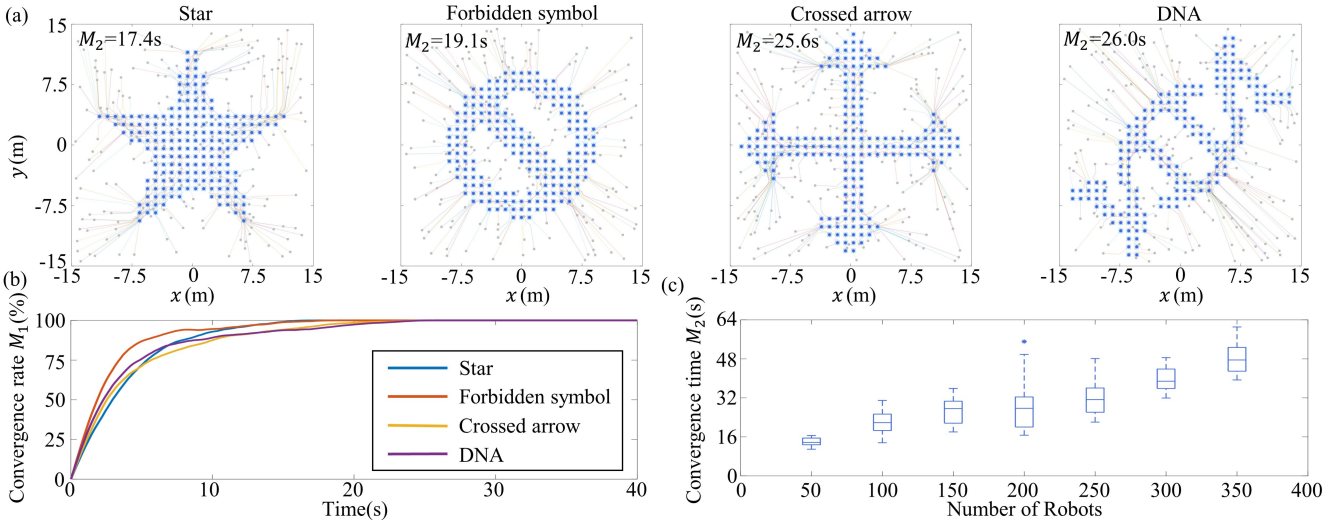

Fig. 5. Simulation results for the effectiveness of our algorithm. (a) The trajectories and convergence time of forming the shapes Star, Forbidden symbol, Crossed arrow, and DNA with $n = 200$. (b) The convergence rate over time of forming these four shapes. (c) The statistical convergence time for forming DNA with the number ranging from 50 to 350.

just computes the target density and then obtains the mean-shift velocity to occupy the target location.

3) Collision-Avoidance Velocity: This paper assumes an obstacle-free environment, that is, we only consider the collision avoidance among robots. If $d_{ij} > r_{\text{avoid}}$ is satisfied, the collision-avoidance velocity v_i^c can be set to 0. Otherwise, it can be designed as follows:

$$v_i^c = \kappa_c \sum_{j \in \mathcal{N}_i} \frac{r_{\text{avoid}} - d_{ij}}{d_{ij}} (p_i - p_j) \tag{16}$$

where κ_c is a positive constant, and $d_{ij} = \|p_i - p_j\|$. Here, r_{avoid} represents the collision-avoidance range.

It should be noted that the control command (11) cannot completely achieve no collision in some specific cases. Here, to rigorously guarantee collision avoidance, we impose the upper bounded constraint [30] on the control command (11). Specifically, the mobility constraint of robot i for collision avoidance from time t to $t + \Delta t$ can be derived by

$$\|v_i\| < v_i^{\text{col}} := \min_{j \in \mathcal{N}_i} \frac{1}{\Delta t} (\|p_j(t) - p_i(t)\| - 2r_{\text{body}}). \tag{17}$$

Then, each robot can compute the final mobility constraint $v_i^{\text{in}} = \min\{v_{\max}, v_i^{\text{col}}\}$. As a result, the control command can be updated as follows:

$$v_i = \begin{cases} v_i, & \|v_i\| \le v_i^{\text{in}} \\ \dfrac{v_i}{\|v_i\|} v_i^{\text{in}}, & \|v_i\| > v_i^{\text{in}} \end{cases}. \tag{18}$$

IV. Simulation and Experiment Results

To verify the performance of the proposed shape formation control algorithm in this paper, we test it in both simulations and physical experiments. Meanwhile, we employ the following metrics to evaluate the performance.

1) Convergence rate M_1: This metric is used to evaluate the completion rate of the shape formation, defined by

$$M_1 = \left(1 - \frac{\text{dist}(\mathcal{P}_t, \mathcal{Q})}{\text{dist}(\mathcal{P}_0, \mathcal{Q})}\right) \times 100\%$$

where $\mathcal{P}_t = \{p_1(t), p_2(t), \ldots, p_n(t)\}$ represents the set of the positions of the robots at time t and $\text{dist}(\mathcal{P}_t, \mathcal{Q})$ represents the total distance cost required for achieving the optimal assignment between $\mathcal{P}_t$ and $\mathcal{Q}$ based on distance.

2) Convergence time M_2: This metric represents the time required for the desired formation to be achieved. In general, the completion of the shape formation means that each target location is occupied by only one robot, i.e., the convergence rate $M_1 = 100\%$.

A. Simulation Results

In simulation, we set the parameters of the robot system as $r_{\text{s}} = 1.7\,\text{m}$, $r_{\text{body}} = 0.15\,\text{m}$, $r_{\text{avoid}} = 0.5\,\text{m}$, $v_{\max} = 2\,\text{m/s}$ and the minimum distance between target locations as $l = 1\,\text{m}$. The parameters in the approach are set as $\kappa_1 = 1.25$, $\sigma = 0.32$, $\delta_1 = 0.3$, $\omega_1 = 1$, $\omega_2 = 0.4$, $\kappa_a = 2$, $\kappa_\rho = 2$, $\kappa_c = 5$. Then, we present two examples to test and verify the effectiveness and efficiency of our proposed algorithm.

The first example demonstrates the effectiveness of our algorithm under different complicated shapes and swarm sizes. We first utilize our proposed algorithm to form four complicated shapes with $n = 200$. As shown in Fig. 5(a), a swarm of robots can successfully complete the desired shapes in these four trials, even if the sophisticated shape DNA with double helix structure. Furthermore, Fig. 5(b) illustrates convergence rate M_1 over time of the four trials. It can be observed that our algorithm can drive 200 robots to form the shapes within less than 30 seconds, demonstrating the effectiveness of our proposed algorithm in complicated shapes.

To ensure the reliability of the proposed algorithm, we conduct statistical simulations using a representative DNA as the desired shape. Specifically, we conduct 20 trials for the robots with robot number ranging from 50 to 350 and record convergence time for all trials. In each trial, the robots are initialized with a random distribution and eventually converge to the desired shape. The statistical results are

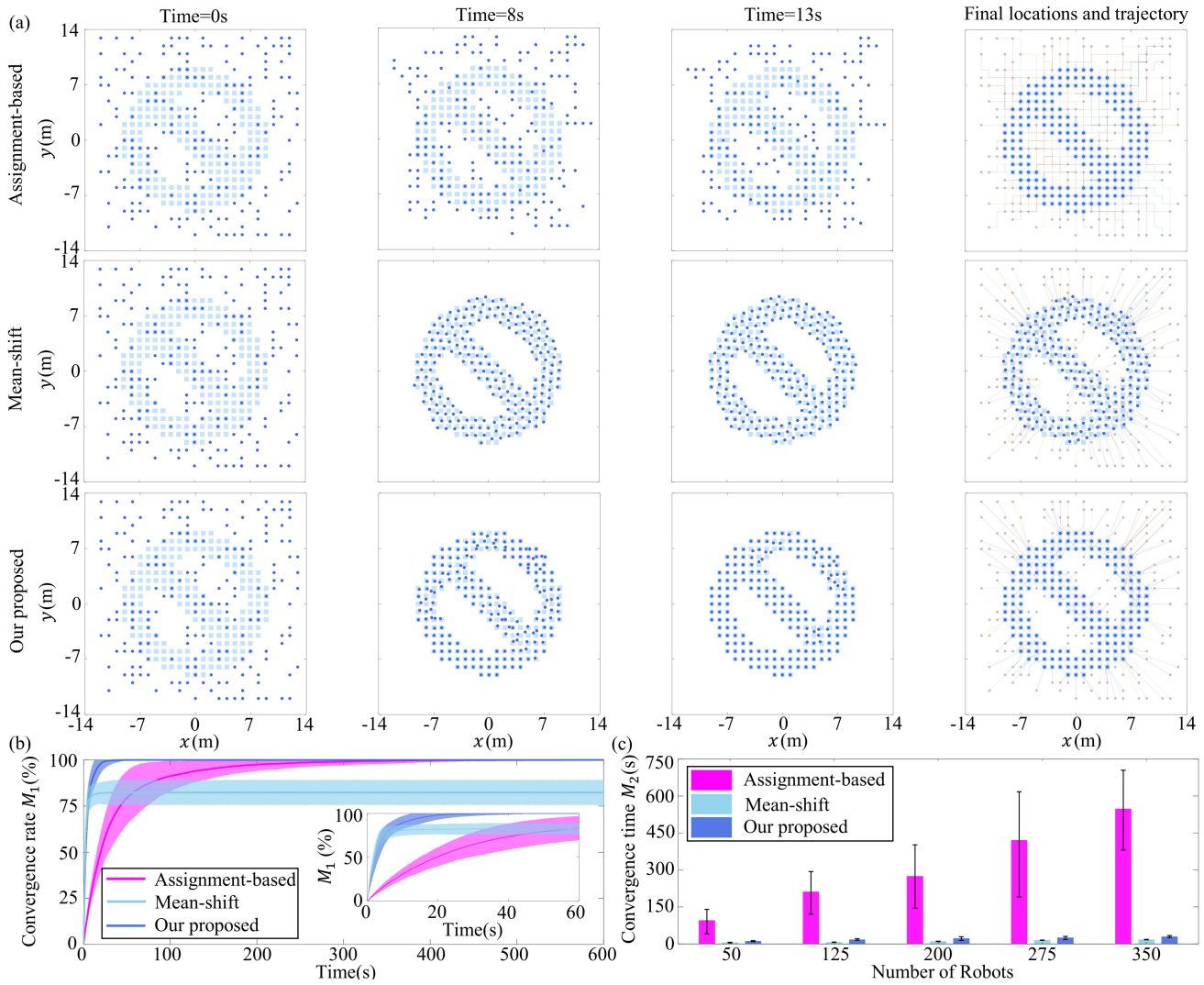

Fig. 6. Comparison results for the efficiency between the proposed algorithm, the assignment-based algorithm, and the mean-shift algorithm. (a) Snapshots and trajectories of forming Forbidden symbol with $n = 200$ by three algorithms. (b) Comparison results of statistical convergence rate over time with $n = 200$. (c) Comparison results of statistical convergence time for forming Forbidden symbol with n = 50, 125, 200, 275, and 350.

shown in Fig.5(c). As can be seen, with the increase in swarm size from 50 to 350, the convergence time M_2 increases from 10s to 65s. It should be noted that the increase in swarm size does not lead to a significant increase in the convergence time, thus demonstrating the performance of the proposed algorithm.

The second example is designed to test the performance of our proposed algorithm by comparing with two state-of-the-art algorithms, the assignment-based algorithm in [10] and the mean-shift algorithm in [26]. In the comparative simulations, we form 10 types of shapes with $n = 200$ and each shape is subjected to 20 repeated trials. In each comparative trial, our proposed algorithm and the assignment-based algorithm can drive a swarm of robots from same initial positions to the given target locations while the mean-shift algorithm [26] cannot achieve such precise shape formation. Fig. 6(a) illustrates snapshots of the Forbidden symbol shape formed by these three algorithms. Furthermore, we compare the average convergence rate and the confidence interval of the convergence rate at a 2σ confidence level for the three algorithms, as shown in Fig. 6(b). It can be observed that the assignment-free algorithms (that is, our proposed algorithm and mean-shift algorithm) have higher efficiency compared to the assignment-based algorithm. However, although the mean-shift algorithm possesses better efficiency, it cannot form precise shapes in these trials. In contrast, our proposed algorithm and the assignment-based algorithm can successfully achieve precise shapes, with our algorithm exhibiting better performance. The main reason is that the target assignment is removed in our algorithm while the task swapping in the assignment-based algorithm is time-consuming.

Then, we conduct statistical simulations for different swarm sizes with n = 50, 125, 200, 275, and 350. The statistical results of convergence time M_2 for different swarm sizes are shown in Fig. 6(c). It can be seen that as the number of robots increases, the convergence time of the assignment-based algorithm increases rapidly compared to

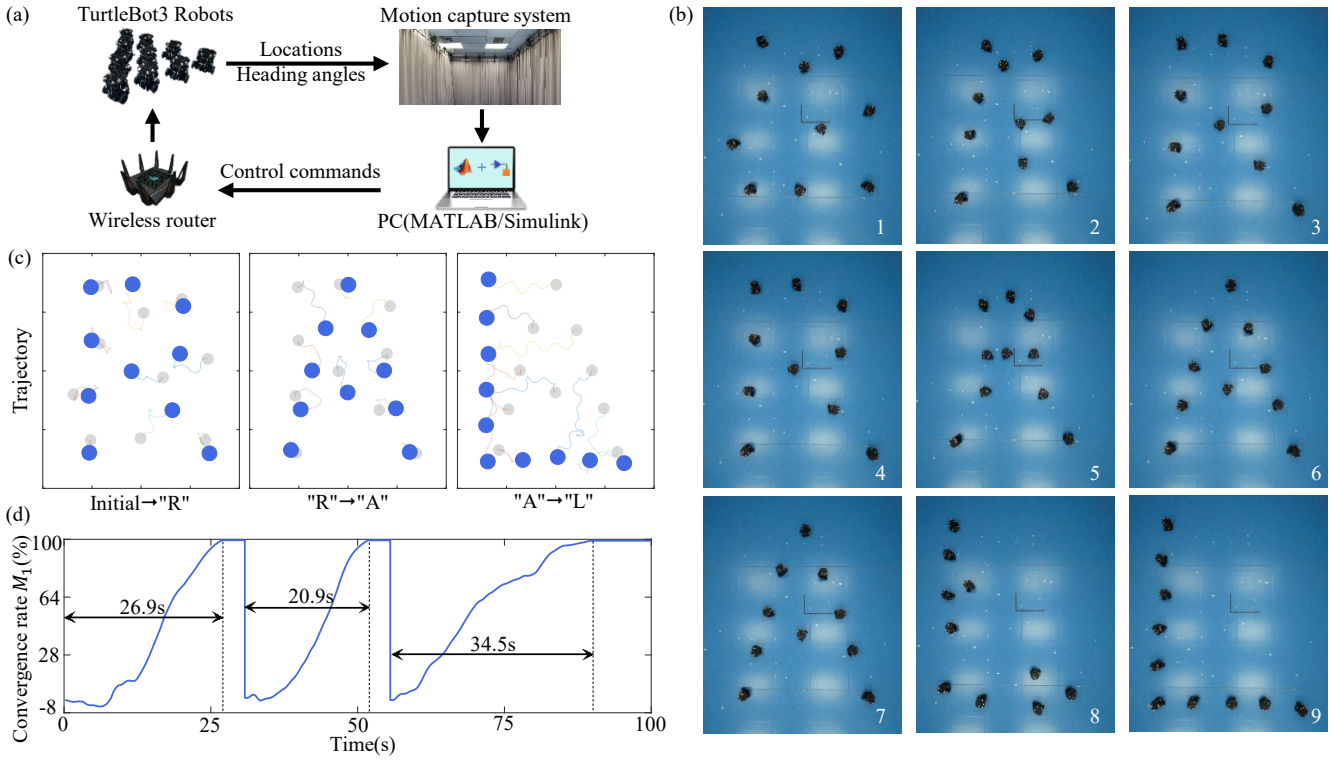

Fig. 7. System configuration and experiment results. (a) Experiment system configuration. (b) Snapshots of the experiment in forming "R", "A", and "L". The snapshots 1–3, 4–6, and 7–9 show the robots from initial state to the letter "R", the letter "R" to "A", and the letter "A" to "L", respectively. (c) Trajectories of the experiment process. (d) The convergence rate of the experiment over time.

our algorithm. Specifically, when the number is 50, the convergence time of our proposed algorithm is 8 times shorter than the assignment-based algorithm, and when the number increases to 350, it takes 18 times shorter than the assignment-based algorithm. Obviously, the assignment-free algorithms possess better performance compared to the assignment-based algorithm, but the mean-shift algorithm fails to achieve precise shape in all trials despite higher efficiency as mentioned before.

B. Experiment Results

In the physical experiment, we form the letter shape "R", "A", and "L" sequentially from the initial configuration with 10 mobile robots (TurtleBot3 Burgers), as shown in Fig. 1. The experiment video can be found in the supplementary materials.

Our experiment system configuration is shown in Fig. 7(a), including 10 robots, a PC, a wireless router, and a Nokov Mocap system (motion capture system). The experimental platform spans an area of $4\mathrm{m} \times 5\mathrm{m}$. Here, the Nokov Mocap system is used to obtain real-time positions of robots. Since the motion capture system requires a PC for receiving robots' state data, we have to increase a PC in our experimental system to derive the real-time states of robots. Then, considering the instability of peer-to-peer communications among robots, we perform the proposed algorithm by a distributed parallel mechanism on the PC (MATLAB/Simulink) to calculate the control commands. This kind of architecture is widely adopted for the verification of distributed algorithms in robot systems [26], [31]. Finally, these control commands are transmitted to robots via the wireless router at a control frequency of 10Hz. It is worth noting that the control command (11) is transformed into the linear velocity and angular velocity based on the results in [32], as the dynamics of the TurtleBot3 Robot is nonholonomic. In addition, the changed parameters compared to simulations are $r_\mathrm{s} = 1.2\,\mathrm{m}$, $v_\mathrm{max} = 0.2\,\mathrm{m/s}$, $l = 0.6\,\mathrm{m}$, $\omega_1 = 0.75$.

The snapshots and trajectories of experimental results are shown in Fig. 7(b) and Fig. 7(c), demonstrating that the robots successfully formed the letter shapes in a sequence. Furthermore, it should be noted that the localization errors in the motion capture system can cause robots to oscillate back and forth in their final positions. To handle this issue, when all robots enter within 5 cm of the neighboring target locations, the PC will transmit the stop control commands to all robots, as shown in Fig. 7(c). Meanwhile, we set the real final positions of robots as the target locations to calculate the convergence rate over time, as depicted in Fig. 7(d). It can be seen that the convergence time M_2 for each shape does not exceed 40 seconds, verifying the practicality of our algorithm in real-world scenarios. In particular, the difference in convergence time between experiments and simulations results from the differences in the maximum velocity setting of robots as well as the dynamics. In detail, the maximum velocity of robots is set to 2 m/s in simulations, while 0.2 m/s in experiments due to robots' hardware limitations.

V. Conclusion

This paper has presented an assignment-free algorithm to achieve distributed shape formation in a multi-robot system. To remove the process of target assignment, we treat all the target locations in robot's perception as the points of mean-shift vector. Then, each robot can compute the density of each target location according to the neighbors' states for computing the weight values of mean-shift vector. To ensure and speed up the shape formation task, the hop count propagated via the multi-robot network is introduced to improve the density design. Subsequently, the calculated mean-shift vector is regarded as the control command of the robot. Finally, the comparative simulations with the-state-of-art approaches highlight the performance of our algorithm, and the physical experiment with 10 TurtleBot3 robots further demonstrates the effectiveness of our proposed algorithm. Although the simulation and experiment results show the advantage of our proposed algorithm in avoiding the local optima, a rigorous theoretical proof is still required. Specifically, this proof will be the focus of our future work. In addition, we plan to achieve a fully distributed approach with peer-to-peer communications by deploying our algorithm on each robot in the future.

References

[1] N. J. Mlot, C. A. Tovey, and D. L. Hu, "Fire ants self-assemble into waterproof rafts to survive floods," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 19, pp. 7669–7673, 2011.

[2] J. Sneyd, G. Theraula, E. Bonabeau, J.-L. Deneubourg, and N. R. Franks, *Self-organization in biological systems*. Princeton university press, 2001.

[3] A. Gelblum, I. Pinkoviezky, E. Fonio, A. Ghosh, N. Gov, and O. Feinerman, "Ant groups optimally amplify the effect of transiently informed individuals," *Nature Communications*, vol. 6, no. 1, p. 7729, 2015.

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

[5] K. F. E. Tsang, Y. Ni, C. F. R. Wong, and L. Shi, "A novel warehouse multi-robot automation system with semi-complete and computationally efficient path planning and adaptive genetic task allocation algorithms," in *Proceedings of International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, Nov. 2018, pp. 1671–1676.

[6] C.-H. Yu and R. Nagpal, "Sensing-based shape formation on modular multi-robot systems: A theoretical study," in *Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems*. Citeseer, 2008, pp. 71–78.

[7] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Image and animation display with multiple mobile robots," *International Journal of Robotics Research*, vol. 31, no. 6, pp. 753–773, May. 2012.

[8] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar. 2013.

[9] H. Oh, A. R. Shirazi, C. Sun, and Y. Jin, "Bio-inspired self-organising multi-robot pattern formation: A review," *Robotics and Autonomous Systems*, vol. 91, pp. 83–100, May. 2017.

[10] H. Wang and M. Rubenstein, "Shape formation in homogeneous swarms using local task swapping," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 597–612, 2020.

[11] M. M. Zavlanos and G. J. Pappas, "Dynamic assignment in distributed motion planning with local coordination," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 232–242, 2008.

[12] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *Proceedings of IEEE Conference on Decision and Control*, 2008, pp. 1212–1217.

[13] J. Yu and S. M. LaValle, "Shortest path set induced vertex ordering and its application to distributed distance optimal formation path planning and control on graphs," in *Proceedings of IEEE Conference on Decision and Control*, 2013, pp. 2775–2780.

[14] P. MacAlpine, E. Price, and P. Stone, "Scram: Scalable collision-avoiding role assignment with minimal-makespan for formational positioning," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 29, no. 1, 2015.

[15] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.

[16] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Multi-robot system for artistic pattern formation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011, pp. 4512–4517.

[17] K. Sakurama and H.-S. Ahn, "Multi-agent coordination over local indexes via clique-based distributed assignment," *Automatica*, vol. 112, p. 108670, 2020.

[18] J. Hu, H. Zhang, L. Liu, X. Zhu, C. Zhao, and Q. Pan, "Convergent multiagent formation control with collision avoidance," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1805–1818, 2020.

[19] W. Chu, W. Zhang, H. Zhao, Z. Jin, and H. Mei, "Massive shape formation in grid environments," *IEEE Transactions on Automation Science and Engineering*, 2022.

[20] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[21] M. Alhafnawi, S. Hauert, and P. O'Dowd, "Self-organised saliency detection and representation in robot swarms," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1487–1494, 2021.

[22] X. Xu, R. Li, Z. Zhao, and H. Zhang, "Stigmergic independent reinforcement learning for multiagent collaboration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4285–4299, 2021.

[23] P. Rezeck and L. Chaimowicz, "Chemistry-inspired pattern formation with robotic swarms," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9137–9144, 2022.

[24] L. Sabattini, C. Secchi, and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation," *Autonomous Robots*, vol. 30, pp. 385–397, 2011.

[25] Q. Bi and Y. Huang, "A self-organized shape formation method for swarm controlling," in *Proceedings of Chinese Control Conference (CCC)*, jul. 2018, pp. 7205–7209.

[26] G. Sun, R. Zhou, Z. Ma, Y. Li, R. Groß, Z. Chen, and S. Zhao, "Mean-shift exploration in shape assembly of robot swarms," *Nature Communications*, vol. 14, no. 1, p. 3476, 2023.

[27] H. Zhao, J. Wei, S. Huang, L. Zhou, and Q. Tang, "Regular topology formation based on artificial forces for distributed mobile robotic networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 10, pp. 2415–2429, Oct. 2019.

[28] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Magazine*, vol. 29, no. 1, pp. 9–9, 2008.

[29] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[30] X. Li, R. Zhou, G. Sun, and J. Zhang, "Connectivity-preserving flocking of multiagent systems via selecting critical neighbors," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3779–3792, 2023.

[31] D. Mulroy, E. Lopez, M. Spenko, and A. Srivastava, "Using r-functions to control the shape of soft robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8598–8603, 2022.

[32] S. Zhao, D. V. Dimarogonas, Z. Sun, and D. Bauso, "A general approach to coordination control of mobile agents with motion constraints," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1509–1516, 2018.