Iterative Shaping of Multi-Particle Aggregates based on Action Trees and VLM

Hoi-Yin Lee, *Member, IEEE*, Peng Zhou, *Member, IEEE*, Anqing Duan, *Member, IEEE*, Chenguang Yang, *Fellow, IEEE*, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—In this paper, we address the problem of manipulating multi-particle aggregates using a bimanual robotic system. Our approach enables the autonomous transport of dispersed particles through a series of shaping and pushing actions using robotically controlled tools. Achieving this advanced manipulation capability presents two key challenges: high-level task planning and trajectory execution. For task planning, we leverage Vision Language Models (VLMs) to enable primitive actions such as tool affordance grasping and non-prehensile particle pushing. For trajectory execution, we represent the evolving particle aggregate's contour using truncated Fourier series, providing efficient parametrization of its closed shape. We adaptively compute trajectory waypoints based on group cohesion and the geometric centroid of the aggregate, accounting for its spatial distribution and collective motion. Through real-world experiments, we demonstrate the effectiveness of our methodology in actively shaping and manipulating multi-particle aggregates while maintaining high system cohesion.**

***Index Terms*—Robot manipulation, shape control, action trees, multi-particle manipulation, VLM.**

I. Introduction

THROUGHOUT history, the task of guiding large sheep flocks to designated pastures has relied primarily on herding dogs such as Border Collie. Rather than directly pushing individual sheep, which risks scattering the flock, these dogs skillfully maneuver around the group, guiding the entire flock as a *cohesive* unit towards the desired location [1]. This herding technique is not limited to the animal world, it is also commonly observed in our human daily lives as we often adopt similar strategies when dealing with dispersed elements. Taking floor sweeping as an example. Instead of painstakingly collecting each speck of dust individually, we gather them into a cohesive pile before sweeping it away with a dustpan. This intuitive approach mirrors the principles of herding, where the focus is on managing the group as a whole, rather than individually controlling each element.

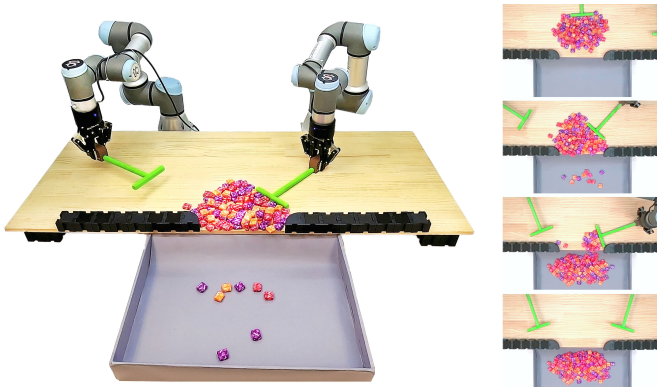

Fig. 1. Setup of the addressed multi-particle shaping task.

This manipulation method is well-suited for multi-particle aggregation tasks, which involve consolidating dispersed elements and guiding them towards a target location (e.g., a storage area) while maintaining the group's cohesion. Taking inspiration from these examples, we can develop bio-inspired herding-like approaches for robots to automatically gather, shape, and transport multi-particle aggregates.

During the initial consolidation stage, this type of robotic herder can skillfully maneuver around the dispersed elements, gradually changing the overall shape and guiding it towards a target point. This strategy ensures that individual objects are not simply pushed in isolation, but rather gathered into a cohesive unit. Throughout this process, the robot can compute and adjust the shape and cohesion of the ensemble, thus, maintaining its compactness and preventing fragmentation.

With the elements gathered into a cohesive pile, the robotic herder can then focus on shaping and guiding group towards the designated location (or even along a path). By treating the ensemble as a unit, the robot can actively change the group's morphology to effectively guide it through obstacles and tight spaces. This process can also reduce the risk of losing individual components along the way, as it ensures that the particle ensemble remains cohesive throughout the task.

To mimic this herding-like guiding strategy, two main approaches have been explored in the literature: multi-robot systems [2]–[8] and robotic manipulators [9]–[18]. For the former, researchers have developed different active methods, for example, [2] presented a distributed strategy to herd groups of active (robotic) evaders to a predefined area, akin to a

Manuscript received: January, 4, 2025; Revised February, 24, 2025; Accepted April, 30, 2025. This paper was recommended for publication by Editor Júlia Borràs Sol upon evaluation of the Associate Editor and Reviewers' comments. This work is supported by the Research Grants Council of Hong Kong under grant number 15201824 and in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62403211. *Corresponding authors: Peng Zhou, David Navarro-Alarcon.*

H.-Y. Lee and D. Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong. hyinlee@polyu.edu.hk, dnavar@polyu.edu.hk

P. Zhou is with the School of Advanced Engineering, The Great Bay University, Dongguan, China. pzhou@gbu.edu.cn

A. Duan is with the Department of Robotics, Mohamed Bin Zayed University of Artificial Intelligence, UAE. anqing.duan@mbzuai.ac.ae

C. Yang is with the Department of Computer Science, University of Liverpool, Liverpool, UK. cyang@ieee.org

Digital Object Identifier (DOI): see top of this page.

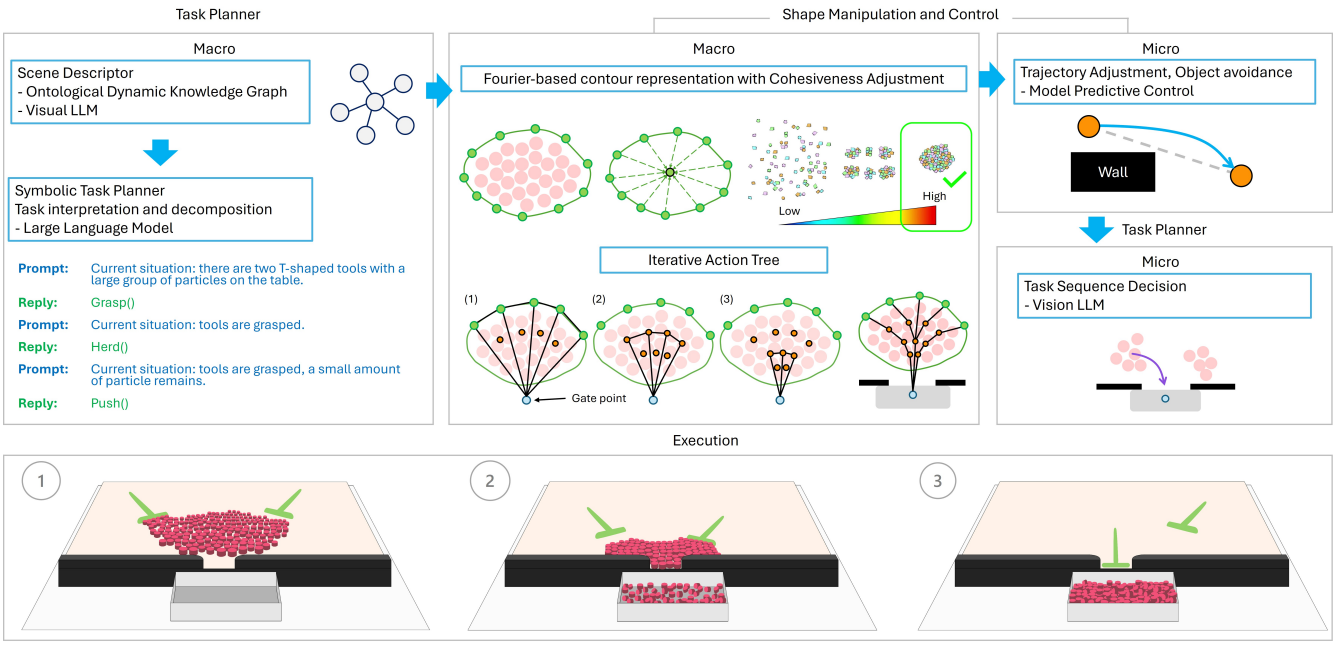

Fig. 2. System Structure: The system consists of a Task Planner with a Shape Manipulation and Control module. An ontological knowledge graph and VLM describe the scene, interpreted by an LLM for task decomposition. Shape is represented by Fourier Series to generate an action tree based on particle cohesiveness. The proposed iterative action tree is conceptually represented. (1)–(3): The orange point represents a waypoint on a trajectory, which is positioned at the centroid of a triangle. (4): The overall trajectory can be obtained by applying graph theory principles to connect these strategically placed waypoints. An MPC guides manipulation while avoiding obstacles. The VLM confirms particle status when detection becomes difficult.

sheepdog pushing the outmost evader, which inspired us to explore similar approaches. Similarly, [3] developed an adaptive density-based interaction controller for trapping heterogeneous targets with swarm robots, allowing them to self-organize and adjust the encirclement based on target strength. Dynamic path planning is also crucial for herding groups to a destination while avoiding obstacles, as demonstrated in [5] and [6]. Additional multi-agent studies, such as [7] and [8], have also contributed to shaping ensembles, emphasizing the versatility of multi-robot approaches.

Robotic manipulators have also been used to shape and manipulate multi-particle aggregates. For example, [10]–[13] have proposed learning-based methods to compute the dynamics of granular pieces and thus perform the task. Among these, [14] presents the most comparable and closest approach to our current work, wherein a group of object piles is manipulated into the desired location using dynamic-resolution model learning. Yet, they emphasize the final shape of the group rather than the shape cohesiveness throughout the manipulation. To enhance scene understanding and optimize action computation, researchers have developed efficient strategies to learn models from visual data and text-based scene interpretation [19]–[32]. These works collectively inform our approach by providing insights into both robotic interactions and the integration of perception in task execution.

While there has been substantial work in multi-object/particle manipulation, the preservation of group cohesion during the manipulation process remains an under-explored area in the manipulator field. In contrast, multi-robot systems often consider cohesion. By integrating these insights, our research adopts shepherding strategies based on the iterative shaping of the aggregate, offering a promising direction that warrants further investigation.

The original contributions of this work are as follows:

- A novel contour-based strategy to shape and transport multi-particle systems with non-prehensile actions.
- An iterative action tree for path planning that preserves the cohesion and holistic nature of the particle group.
- A cohesiveness metric to quantify the compactness of the particle ensemble.
- A LLM-based planner that leverages the feedback shape to guide the herding of particles.

The rest of this manuscript is organized as follows: Sec. II presents the detailed methodology of our approach, Sec. III reports the empirical results and evaluations, and Sec. IV provides the final conclusions drawn from this work.

II. Methodology

A. Problem Formulation

Our objective in this work is to develop a tool-based method to "guide" a group of particles that are initially located outside a designated gate area. These particles must be iteratively pushed through a gate to reach a containment box. This can be viewed as a herding problem, where the particles are akin to a flock of sheep (although passive) and the robotic tool serves as a "sheepdog" to guide them towards the desired goal location. A key requirement to preserve cohesion among the group of particles throughout the manipulation task. In simple words, we want the aggregate to **Stay Together and Stay Connected**.

In our proposed method, the guiding and incremental pushing of the ensemble is referred to as "herding". The slot or space through which the particles must pass to enter the container is called the "gate". The term "cohesion" is used to describe the degree to which the group of particles is united, compact, and stuck together. It refers to how tightly packed or cohesive the particle group is as a whole.

To manipulate a group of particles, a systemic methodology has been developed, as depicted in Fig. 2. The entire process is based on the image input and is divided into two aspects: (1) shape manipulation and control, and (2) task planning. Each aspect has a macro and a micro section, where the macro provides an approximate solution, which is then refined by the micro part.

B. Macro-scale Shape Manipulation

1) Shape Representation: Representing and controlling the shape of a particle group poses unique challenges compared to manipulating a deformable soft object. Unlike a soft object, whose shape is bounded by its volume and mass distribution, the configuration of a particle group can be highly unconstrained. The particles may be distributed in an arbitrary spatial arrangement, with no inherent limits on their relative positions or the overall shape of the ensemble. A small movement of a single particle may not contribute to the majority of shape-changing. For example, the particles can squeeze into the spaces among the ensemble with no significant total shape deformation. Thus, instead of an accurate model to represent the details ensemble's shape and the dynamic motion of particles, we propose an arbitrary shape representation approach based on Fourier descriptors, see 2. By representing the boundary of the particle group using a compact set of Fourier coefficients, we can track and mould the macro-scale shape without requiring a detailed dynamical model of the individual particle motions.

Since no prior information is available about the shape of the particle group, we extract the contour with computer vision. Subsequently, we integrate all particle contours to construct a comprehensive contour that represents the overall outline of the particle group. This contour is then represented using a Fourier series. The complex-valued function $f(\tau)$ is used to express the contour as:

$$f(\tau) = \sum_{n=-N}^{N} c_n e^{in\tau} \tag{1}$$

where c_n are the coefficients of the Fourier series and the scalar N represents the number of *finite* Fourier coefficients used. The coefficients c_n can be computed with the integral:

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(\tau) e^{-in\tau} d\tau \tag{2}$$

The contour is then reconstructed by summing the contributions of the Fourier coefficients at each time point τ with the range set to $[0, 2\pi]$ in a centered coordinate system.

The choice of the number of Fourier harmonics used to represent the particle group's shape is a key parameter in this approach. A larger number of harmonics allows for a more detailed and accurate shape representation, capturing finer contour features of the particle distribution. However, this increased complexity also comes with more redundant contour points and a rougher, less streamlined overall shape. Conversely, using a smaller number of harmonics results in a more simplified, smoother contour representation, but at the cost of losing some of the detailed shape information. For the purposes of this particle herding task, we are primarily interested in controlling the macro-scale shape of the particle ensemble, rather than tracking its micro-scale features. Therefore, a lower number of harmonics is preferred, as it provides a sufficiently accurate yet compact shape representation that can be effectively servoed by the robot.

2) Cohesiveness Metric: Maintaining the cohesiveness of a particle group and keeping the particles compact are crucial objectives. To quantify this cohesiveness, we introduce a cohesiveness metric that measures how tightly the particles are encircled within the group's contour. Based on the geometric properties of the particle group, we observe that the distance between any point on a circle and its centroid is always constant. In contrast, the distance between a corner point of a rectangle/square and its centroid is not identical to any other non-corner point, as it is much farther away. Therefore, a regular circle represents the optimal enclosure for a group of objects given the same density, and the size of the group is irrelevant in this context. The same principle can be applied to other shapes, where a square is better than a wide rectangle in terms of cohesiveness.

We refer to this concept as the shape regularity, which is calculated based on the average distance between a set of n points on the contour of the particle group and the group's centroid, compared to the optimal, minimalistic area. The shape regularity metric ranges from 0 to 1, where higher values indicate a more regular, compact shape.

Density is another important factor in determining the level of compactness within a region. It can be calculated by finding the ratio between the area occupied by the particles and the total area of the particle group. We can mathematically express the cohesiveness metric as follows:

$$\zeta = \frac{\sqrt{\frac{\alpha}{\pi}}}{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p}_i - \text{mean}(\mathbf{p})||} \times \frac{\alpha}{\beta} \times 100\% \tag{3}$$

where α is the area occupied by the particles, which can be obtained through contour area computation using computer vision. It is the sum of the individual contour areas prior to forming the comprehensive contour: $\alpha = \sum \alpha^i$ where α^i is the contour area of the i-th particle (computed using vision). β is the total area of the particle group calculated by Fourier-series analysis. $\mathbf{p}_i$ represents the i-th point on the Fourier-based contour of the particle group. The function mean$(\mathbf{p})$ is the centroid of the particle group. The first part of the equation measures the shape regularity of the particle group by comparing it with the optimal encirclement (i.e. a circle), where the radius r is solved from the area equation $\alpha = \pi r^2$ and is divided by the average Euclidean distance of the contour points from the centroid. The second part calculates the density of the particle group by dividing the occupied area of the particles α by the total area of the whole group β.

Algorithm 1: Iterative Action Tree for Path Planning

Input: $\mathbf{P}$, gate ▷ Set of farthest points, gate location
Output: $\mathbf{\Pi}$ ▷ Set of trajectories
$\mathbf{C}_0 \leftarrow \text{ComputeInitialCentroids}(\mathbf{P}, \text{gate})$;
$\mathbf{C} \leftarrow \mathbf{C}_0$;
$i \leftarrow 1$;
while $|\mathbf{C}_i| > 2$ **do**
  for $j \leftarrow 1$ *to* $(|\mathbf{C}_i| - 1)$ **do**
    $\mathbf{c} \leftarrow \text{FindCentroid}(\mathbf{C}_{i-1,j}, \mathbf{C}_{i-1,j+1}, \text{gate})$;
    $\mathbf{C}_i \leftarrow \mathbf{C}_i \cup \mathbf{c}$;
  $i \leftarrow i + 1$;
  $\mathbf{C} \leftarrow \mathbf{C} \cup \mathbf{C}_i$;
$\mathbf{G} \leftarrow \text{Graph}(\text{gate}, \mathbf{C}, \mathbf{P})$;
$\mathbf{\Pi} \leftarrow \text{ApplyDijkstra}(\mathbf{G})$;
return $\mathbf{\Pi}$

By combining these two factors, the cohesiveness metric ζ is obtained, which ranges from 0 to 100%, with higher values indicating a more cohesive particle group. This metric provides a quantitative measure of the particle group's cohesion, which can be used to optimize the particle arrangement and overall system performance.

3) Path Planning: Iterative Action Tree: To guide the particle group towards the designated gate location, we propose a path-planning approach based on an iterative action tree. This method leverages the spatial distribution of the particles to define a sequence of waypoints that the robotic tool can follow to effectively herd the ensemble.

In the first step of our approach, we identify a subset of particles that are farther away. These distant particles represent the most outlying members of the aggregated group that require guidance to be brought back towards the target. We utilize the cohesion metric to analyze which specific particles are contributing to a lower overall cohesion of the group. If the cohesion is observed to be low, we select the particles that are farther away from the group's centroid for targeted manipulation. Conversely, if the cohesion is relatively high, we focus our attention on the particles that are farthest from the goal location. This strategic selection of particles allows us to efficiently improve the compactness of the aggregated group while also drawing it back towards the desired target. The distance between the points taken is less than the length of the tool segment to avoid losing 'sheep' during the manipulation stage. In our case, we set the number of points to be 5 based on our tool's dimension and we refer to these points as $\mathbf{P}$.

From this set of 5 farthest particles, we compute the initial centroids by finding the centroids of the triangles formed by connecting each pair of points with the gate location, see Fig. 2 and Alg. 1. Starting from these initial centroids, we iteratively refine the trajectory by computing new centroids. This is done by finding the centroid of the triangle formed by the two consecutive centroids from the previous iteration and the gate location. The new centroid is calculated as the average of the two previous centroids and the gate point (see Fig. 2). This process effectively shrinks the size of the particle group with each iteration, as the centroids move closer to the gate. The algorithm continues until only two centroids remain, resulting in a sequence of centroid sets that progressively reduce the spatial extent of the particle distribution.

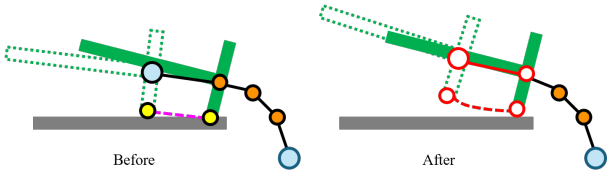

Fig. 3. Before the micro-scale shape refinement, the tooltip (the purple dashed trajectory) will hit the grey wall. After the refinement, the tooltip (the red dashed trajectory) will be smoothly avoiding the grey wall.

Next, we apply graph theory to path-finding. We treat the gate location as the root of a graph, and the computed centroids with the farthest points as the graph nodes. To identify the optimal trajectories from the farthest particles to the gate, we apply a path-finding algorithm: Dijkstra's algorithm. This allows us to construct a set of candidate trajectories that the robotic tool can follow to herd the particles back towards the desired goal location.

C. Micro-scale Shape Manipulation

In each iteration, the tool approaches and moves along the waypoints of a candidate trajectory. To mimic the herding behavior where the group is gradually moved closer to the gate, the tool only travels a short distance in each herding task. Specifically, the tool moves between two waypoints at a time, unless the final waypoint is not within the ensemble or the remaining group size is too small and near the gate.

After generating the candidate trajectories using the centroid-based approach, we apply an off the shelf model predictive control (MPC) framework to refine the trajectories and ensure collision-free motion of the robotic tool. The goal of this refinement is to optimize the tooltip trajectory to avoid obstacles, such as the grey wall shown in Fig. 3.

We first obtain the dimensions of the tool from the computer vision system, which allows us to determine the originally planned start and end points for the tooltip (shown as yellow points in Fig. 3). We then apply a refinement controller using these initial tooltip waypoints as the optimization objective. The controller computes the refined, obstacle-avoiding trajectory (shown as the red dashed line in Fig. 3) by minimizing a cost function that penalizes deviations from the reference trajectory and control effort while satisfying the dynamic constraints and obstacle avoidance requirements.

The refinement controller uses a dynamic model of the tool and the environment to predict the future states of the system. This model includes the differential kinematic equations of the tool, as well as the positions and dimensions of the obstacle in the workspace, such as the gate walls.

The optimization problem is formulated to minimize a cost function that penalizes deviations from the desired trajectory, as well as control inputs that exceed the robot's actuation limits (e.g. having a sudden quick move). The optimization is subject to constraints that ensure the robot's predicted states do not

collide with any obstacles over the prediction horizon. The controller optimization problem is formulated as:

$$J = \arg\min_{\mathbf{u}} \sum_{i=0}^{H-1} ||\boldsymbol{\chi}_i - \boldsymbol{\chi}_{\text{ref}}||_Q^2 + ||\mathbf{u}_i||_R^2 \tag{4}$$

where it is subject to $\dot{\boldsymbol{\chi}} = f(\boldsymbol{\chi}, \mathbf{u})$ and $g(\boldsymbol{\chi}_i) \geq 0$. $\boldsymbol{\chi} = [x, y, \theta]^\top$ is the state vector of the x-y coordinates and orientation of the tooltip, $\mathbf{u} = [v, \omega]^\top$ is the control input vector represents the linear velocity v and angular velocity ω of the tooltip, H is the prediction horizon, $\boldsymbol{\chi}_{\text{ref}}$ is the constant reference state, and Q and R are positive definite weight matrices. The function g ensures that the tooltip maintains a minimum distance from the obstacles (i.e. the gate wall).

By solving the optimization problem, we can refine the trajectory of the tool and determine the optimized boundary for the tooltip's motion.

D. High-Level Symbolic Task Planner

To conduct a long-horizon task that involves logical action planning, we implemented an LLM-based planner for high-level symbolic task decomposition and decision-making. The motion for aggregating particles can be spit into macro and micro planning, see Fig. 2. In terms of the macro aspect, it involves task interpretation and action control and we use a description-based task planner for this. In terms of the micro aspect, it involves the detail decision-making for small adjustments to improve the performance. For this, we implement a vision-based task planner. Although the task of pushing a particle group through a gate may seem straightforward, the LLM framework provides essential flexibility, enabling adaptation to diverse scenarios and enhancing human-robot interaction. This approach supports not only the current task but also future generalization.

1) Description-based Task Planner: To interpret the scene information and the task requirement to generate the next action with predefined action functions, we implement a high-level symbolic planner similar to the recent studies [28], [33]. Instead of fine-tuning an LLM, we use the prompt approach for this herding task.

We leverage the ontological dynamic knowledge graph (ODKG) presented in [33] to store and provide the existing physical and virtual interaction information about the scene. We then apply the VLM to update the ODKG with the latest visual observations, generating textual grounding information that represents the current state of the environment. Then, an LLM takes the natural language input along with the grounding information as input and outputs the appropriate next action function. This iterative process continues, with the system executing the action, observing the updated scene, and generating the next action based on the evolving grounding information. Notably, the selection of the manipulator side is determined by the coordinates computed during the shape manipulation process, rather than through the task planner.

The available action functions are adjusted to fit our current particle aggregation scenarios, including "`grasp`, `herd`, `push`, `release`". The detailed description of the functions is stated in Table I.

TABLE I
ACTION FUNCTIONS AND DESCRIPTIONS

Action Functions	Descriptions
`Grasp()`	To move and grasp the tool.
`Herd()`	To gently push the particle toward the gate while considering particle cohesiveness.
`Push()`	To push the particle directly to the gate.
`Release()`	To release the tool back to its original location and the robot to its home position.

To increase robustness, a few-shot learning method is adopted, and several concrete examples are given in the prompt to illustrate the details of the predefined actions. Consider a simple particle aggregation task, the possible action sequence could be "`grasp()`, `herd()`, `push()`, `release()`", where the dual-arm robot grasps tools, then applied the shape manipulation module to compute the trajectory, and the dual-arm robot herds the 'sheep' back to the desired point. The system continuously checks the status of the particles to confirm that all the 'sheep' are inside the gate. The tools are returned to their original places when the task is considered completed by the micro task planner (more details are mentioned in the next section).

2) Vision-based Task Planner: While the text description-based task planner is sufficient for simple, fast action control, it may miss some small details that require further adjustments to enhance performance. For instance, when only a small amount of particles remain, basic image processing could fail to detect them due to factors like varying lighting conditions affecting color detection accuracy.

To ensure no particles are left behind and the task is truly completed, we complement the image processing with a vision language model (VLM). When the image processing suggests the particle count is low, the system captures the scene and passes it to the VLM for verification. If there are no remains, the task is considered complete, and the "release()" command is sent to the robot, instructing it to return the tools and move back to the home position. If the VLM detects any remaining particles, the system continues the task.

As the particle ensemble becomes smaller (i.e., with only a few particles left), some of the candidate trajectories generated in Sec. II-B3 may become redundant and unnecessary to follow. To address this, the VLM takes the candidate trajectories along with the current image as input, and outputs the most suitable starting point for the pushing action. The system then matches this starting point to the available candidate trajectories and selects the corresponding path to follow.

III. RESULTS

The performance of the proposed method is evaluated through a series of experiments using a pair of UR-3 robots and two T-shaped tools. The robot observes the scene from the top through a RealSense camera D415. Data is passed to a Linux-based computer (i.e. Ubuntu 20) with the Robot Operating System (ROS) for image processing, decision-making, and robot control. We use GPT-4o as the task planner in the experiment. We evaluate the performance of our method across

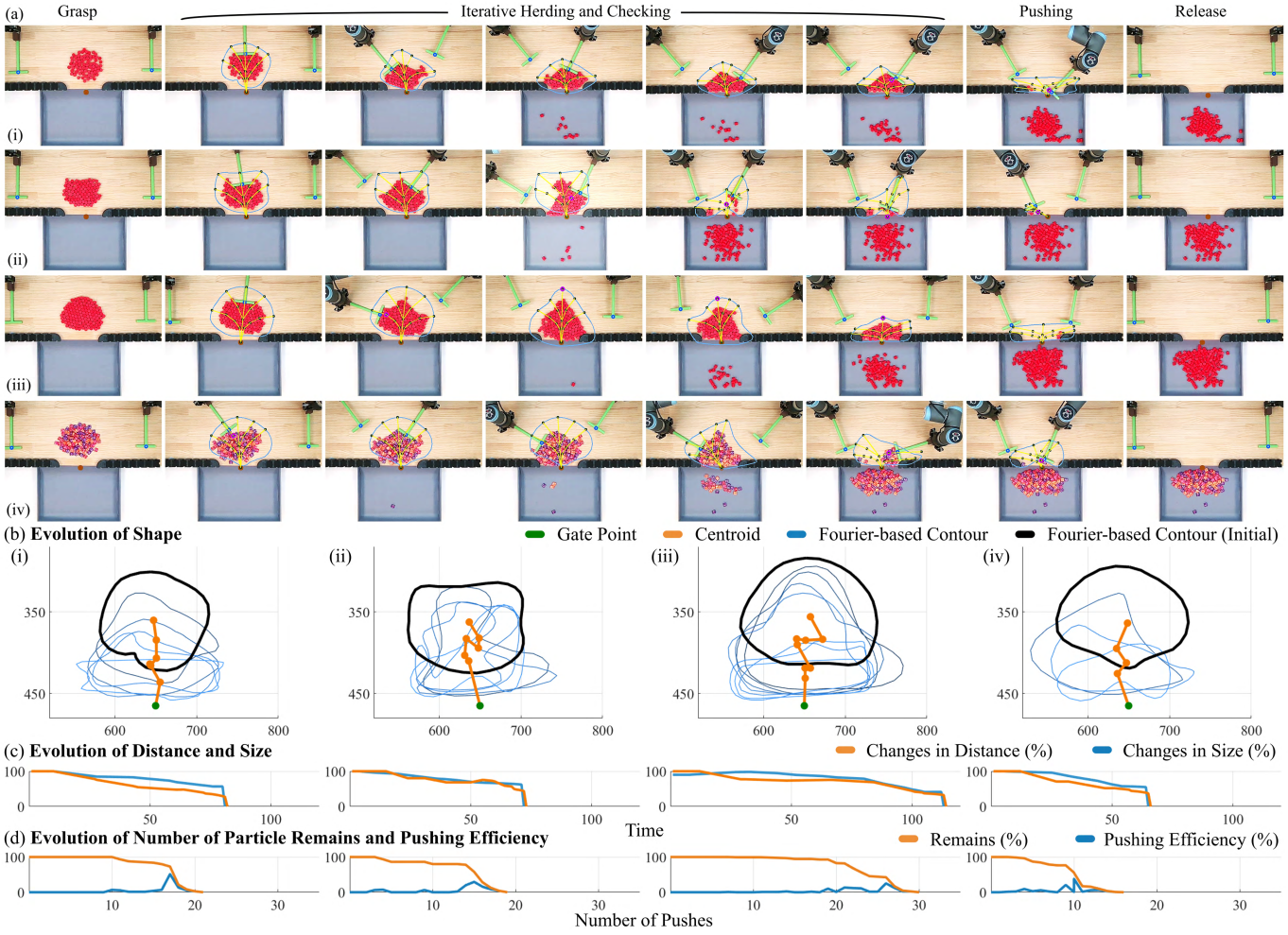

Fig. 4. Experiments: (a) Different shapes and amounts of particles are used to evaluate the performance of the proposed shape representation method. The initial setting of the group is shown on the left. The contour of the particle is expanded and shown in blue lines. The black points are the waypoints of the yellow trajectories. The brown circle represents the gate point which is towards the grey box; (b) The evolution of the shape described by Fourier-based Representation; (c) The evolutionary changes in distance between the centroid of the particle group with the gate and the changes of the group size; (d) The evolution of the number of particles remains on the table and the pushing efficiency of a push.

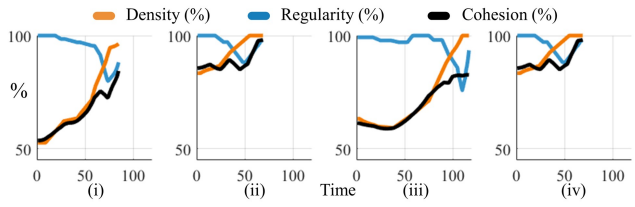

Fig. 5. Evolution of the cohesion: (i)–(iv) reflect the changes in density, regularity, and cohesiveness among the particle groups in Fig. 4(i)–(iv) cases.

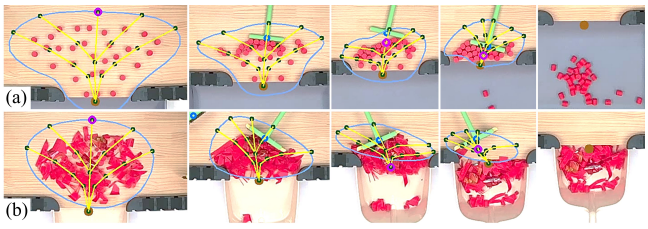

Fig. 6. (a) Dispersion scenario in which particles are separated; (b) Debris sweeping scenario where brushes are used to collect debris into the dustpan.

several aspects: (1) shape representation, (2) path planning, and (3) cohesiveness of the particle group shape.

A. Evaluation and Analysis

To validate the robustness of our proposed model-free shape representation, we tested various particle group sizes and shapes, ranging from small (74 particles) to large (140 particles), as shown in Fig. 4. We conducted at least four trials for each object type.

The robot first observes the scene, and the VLM provides a textual description with the information in the knowledge graph for the LLM to generate the next action. In the experiment, the robot first grasps the tool, then iteratively calculates the pushing direction. Through a series of guiding and pushing steps, the entire particle group is manipulated and guided into the designated grey container.

We begin by expanding the outline of the particle group

TABLE II
COHESION COMPARISON AND ANALYSIS

Cases/Methods	Regular Shapes						Experiments			
	Circle			Square		Rectangle	Ours	MPC [34]	Landmark [7]	Manual
	$r = 2$cm	$r = 4$cm	$r = 8$cm	4cm×4cm	8cm×8cm	8cm×2cm				
Density Ratio	50.0%	50.0.%	50.0%	50.0%	50.0%	50.0%	84.6%	**91.7%**	69.8%	86.6%
Regularity	100%	100%	100%	93.4%	93.4%	68.2%	81.1%	68.5%	**82.8%**	80.9%
Cohesiveness	50.0%	50.0%	50.0%	46.7%	46.7%	34.1%	68.5% ± 5%	62.4% ± 6 %	57.8% ± 7%	**70.1%** ± 5%

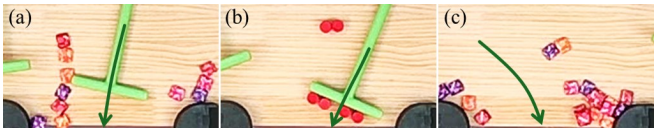

Fig. 7. Failure cases: (a) Redundant pushing path selected by the task planner; (b)-(c) Premature pushing resulting in outliers.

(represented by the blue contour in Fig.4(a)). We then apply the Fourier series method described in Sec. II-B1 to this contour to represent the shape of the particle group. To strike a balance between shape representation fidelity and computational complexity, we set the number of harmonics N to 5 and the number of farthest points concerned to 5.

Our experiments demonstrate that our system can successfully generate a Fourier-based shape representation to capture the arbitrary shape of the particle group in various circumstances. Fig. 4 illustrates the evolution of the particle group throughout the experiment, showing how the initially large, arbitrary shape is gradually moulded and compacted into a smaller form.

During the manipulation process, the particles always remain cohesive, staying together and connected with one another. To quantify the cohesion of the particle group, we compare the cohesiveness metrics calculated using Eq. (3). As shown in Fig. 4, the cohesiveness of the particle shape is preserved as the robot guides it to the desired location using the herding manipulation approach. The evolution of the cohesiveness, from an initially low rate to the final high rate, is reflected in the changes in particle density and regularity, as plotted in Fig. 5. The experiments also demonstrate that when the particle density is high, the regularity tends to be at a lower value. This is primarily due to the iterative shaping of the particle group, which aims to keep it tightly packed while moving towards the gate. Importantly, the particle group is maintained as a single, cohesive entity, rather than dividing into multiple clusters or segmenting with large spaces. This proves the effectiveness of our proposed system in molding and herding the particle group as a whole.

We also demonstrated the robustness of the system in two additional scenarios: one with particles initially spread apart and another involving debris sweeping. As shown in Fig. 6, our robot successfully aggregates and herds all particles and debris into the container or dustpan. This demonstrates the system's potential for various applications.

With the Fourier-based shape descriptor, the system computes a tree-like trajectory, as shown by the yellow path with black waypoints in Fig. 4 and 6. Given the arbitrary initial shape of the particle group, this demonstrates the system's capability to generate a reasonable trajectory and adapt to collisions for shape manipulation.

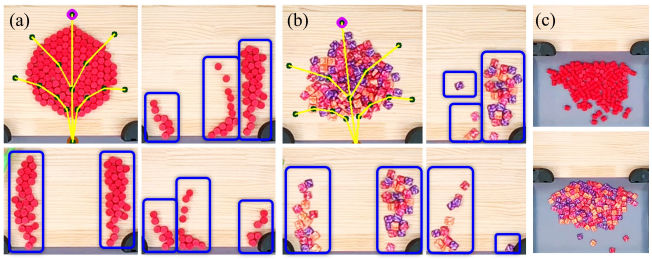

Fig. 8. Performance of adopting direct pushing method: (a) 134 particles; (b) 140 candies; (c) All particles are successfully pushed into the container.

Overall, we achieved about 97% success, with the remaining 3% due to premature pushing and misidentifying the particle status from the image as shown in Fig.7.

B. Comparison

To demonstrate the advantage of the herding method over the direct brute force pushing approach, we conducted a comparative evaluation of the particle manipulation performance in terms of cohesiveness preservation.

We applied the same Fourier-based action tree strategy to derive the trajectory, executing it through the direct pushing technique, which involves moving the tool along the full trajectory rather than just one period. As shown in Fig. 8, all particles were successfully pushed into the container, highlighting the robustness of our strategy in computing the trajectory even when the particles are completely split.

The direct pushing approach failed to preserve the cohesion of the particle group, resulting in the division of the aggregate into multiple smaller subgroups and low cohesiveness. In contrast, our proposed shape manipulation action tree has proven effective in handling scenarios with multiple subgroups, successfully guiding all particles into the target box, as illustrated in the same figure.

The key distinction between the two methods lies in their underlying strategies. Our herding-based approach focuses on guiding the particles as a cohesive group, whereas the direct pushing technique tends to scatter the particles, leading to a loss of group cohesiveness.

To further quantify the cohesion properties, we present the cohesiveness metric results in Table II. We first establish a baseline by comparing the cohesiveness of regular geometric shapes, including circles of varying radii, squares, and rectangles, all with a density ratio of 0.5. The results demonstrate that even when the density is the same, the cohesion can

differ based on the shape. Specifically, circles exhibit the highest cohesion, followed by squares, while rectangles show the lowest cohesion due to the increased distance between the particles and the centroid.

We then apply our proposed cohesiveness metric to evaluate the performance of our system against state-of-the-art manipulation methods, including the MPC approach [34] and a landmark-inspired technique [7], using the same type of particle object. The results indicate that while the traditional MPC method achieves the highest density ratio, it exhibits the lowest cohesion among the evaluated approaches. This is primarily due to its focus on efficiently aggregating particles rather than maintaining an optimal shape.

In contrast, the landmark-inspired method achieves the best regularity score, producing shapes close to the optimal configuration. This can be attributed to its shape-molding capabilities. Our proposed approach, on the other hand, generally demonstrates better performance in preserving group cohesion, with high average density and regularity. However, compared to manual human aggregation (with 3 trials for each of the 5 participants), our robotic system is still outperformed by approximately 1.5% in cohesiveness. We illustrate the performance of the proposed methods in the accompanying video https://vimeo.com/1043879712.

IV. Conclusion

In this work, we have introduced a new Fourier-based shape control method and an iterative action tree for guiding and manipulating multiple particles in an aggregation task. To quantify the effectiveness of the proposed methodology, we have developed a cohesiveness metric to measure the compactness of a particle group. We have implemented the system with the ODKG and VLM for task planning and validated it using a dual-arm robotic platform. The experimental results show that our methodology achieves a cohesiveness of 68%, while the human performance is slightly higher at 70%. Although our system did not outperform humans, these results are promising and indicate the potential of our approach. Moving forward, we plan to continue improving the cohesion preservation capabilities of our system. Future research directions may include enhancing the cohesiveness metric, exploring advanced control strategies and optimization techniques, investigating scalability to handle larger systems, and extending the system to dynamic environments. By addressing these aspects, we aim to further advance the state-of-the-art in multi-object aggregation and manipulation, with the goal of developing intelligent robotic systems capable of efficient and cohesive group management in a wide range of applications.

References

[1] J.-M. Lien, *et al.*, "Shepherding behaviors," in *IEEE Int. Conf. on Robot. and Autom., Proceedings.*, vol. 4, 2004, pp. 4159–4164.

[2] S. Zhang, *et al.*, "A distributed outmost push approach for multi-robot herding," *IEEE Trans. Robot.*, 2024.

[3] ——, "Heterogeneous targets trapping with swarm robots by using adaptive density-based interaction," *IEEE Trans. Robot.*, 2024.

[4] S. Zhang *et al.*, "Collecting a flock with multiple sub-groups by using multi-robot system," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6974–6981, 2022.

[5] S. Van Havermaet, *et al.*, "Reactive shepherding along a dynamic path," *Scientific Reports*, vol. 14, no. 1, p. 14915, 2024.

[6] M. Hamandi, *et al.*, "Robotic shepherding in cluttered and unknown environments using control barrier functions," *arXiv preprint arXiv:2407.15701*, 2024.

[7] A. Vardy, "Landmark-guided shape formation by a swarm of robots," in *Distributed Autonomous Robotic Systems: The 14th International Symposium*. Springer, 2019, pp. 371–383.

[8] D. Strömbom *et al.*, "Robot collection and transport of objects: A biomimetic process," *Frontiers in Robotics and AI*, vol. 5, p. 48, 2018.

[9] S. Zimmermann, *et al.*, "Automated robotic manipulation of individual colloidal particles using vision-based control," *IEEE/ASME Trans. on Mech.*, vol. 20, no. 5, pp. 2031–2038, 2014.

[10] H. T. Suh *et al.*, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," in *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*. Springer, 2021, pp. 347–363.

[11] C. Schenck, *et al.*, "Learning robotic manipulation of granular media," in *Conf. on Robot Learning*. PMLR, 2017, pp. 239–248.

[12] N. Tuomainen, *et al.*, "Manipulation of granular materials by learning particle interactions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5663–5670, 2022.

[13] Q. Lu *et al.*, "Excavation learning for rigid objects in clutter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7373–7380, 2021.

[14] Y. Wang, *et al.*, "Dynamic-resolution model learning for object pile manipulation," *arXiv preprint arXiv:2306.16700*, 2023.

[15] K. Takahashi, *et al.*, "Uncertainty-aware self-supervised target-mass grasping of granular foods," in *IEEE Int. Conf. on Robot. and Autom.*, 2021, pp. 2620–2626.

[16] A. Cherubini, *et al.*, "Model-free vision-based shaping of deformable plastic materials," *The Int. J. of Robot. Research*, vol. 39, no. 14, pp. 1739–1759, 2020.

[17] Y. Zhu, *et al.*, "A data-driven approach for fast simulation of robot locomotion on granular media," in *IEEE Int. Conf. on Robot. and Autom.*, 2019, pp. 7653–7659.

[18] C. Matl, *et al.*, "Inferring the material properties of granular media for robotic tasks," in *IEEE Int. Conf. on Robot. and Autom.*, 2020, pp. 2770–2777.

[19] X. Lin, *et al.*, "Learning visible connectivity dynamics for cloth smoothing," in *Conf. on Robot Learning*. PMLR, 2022, pp. 256–266.

[20] P. Wu, *et al.*, "Daydreamer: World models for physical robot learning," in *Conf. on Robot Learning*. PMLR, 2023, pp. 2226–2240.

[21] C. Finn *et al.*, "Deep visual foresight for planning robot motion," in *IEEE Int. Conf. on Robot. and Autom.*, 2017, pp. 2786–2793.

[22] H. Shi, *et al.*, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks," *The Int. J. of Robot. Research*, vol. 43, no. 4, pp. 533–549, 2024.

[23] F. Liu, *et al.*, "Robouniview: Visual-language model with unified view representation for robotic manipulation," *arXiv preprint arXiv:2406.18977*, 2024.

[24] K. Kawaharazuka, *et al.*, "Continuous object state recognition for cooking robots using pre-trained vision-language models and black-box optimization," *IEEE Robot. Autom. Lett.*, 2024.

[25] N. Wake, *et al.*, "Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration," *IEEE Robot. Autom. Lett.*, 2024.

[26] C. Huang, *et al.*, "Visual language maps for robot navigation," in *IEEE Int. Conf. on Robot. and Autom.*, 2023, pp. 10 608–10 615.

[27] A. Mei, *et al.*, "Replanvlm: Replanning robotic tasks with visual language models," *IEEE Robot. Autom. Lett.*, 2024.

[28] H.-Y. Lee, *et al.*, "Non-prehensile tool-object manipulation by integrating llm-based planning and manoeuvrability-driven controls," *arXiv preprint arXiv:2412.06931*, 2024.

[29] H. Jeong, *et al.*, "A survey of robot intelligence with large language models," *Applied Sciences*, vol. 14, no. 19, p. 8868, 2024.

[30] J. Wang, *et al.*, "Large language models for robotics: Opportunities, challenges, and perspectives," *arXiv preprint arXiv:2401.04334*, 2024.

[31] J. Gao, *et al.*, "Physically grounded vision-language models for robotic manipulation," in *IEEE Int. Conf. on Robot. and Autom.*, 2024, pp. 12 462–12 469.

[32] F. Zeng, *et al.*, "Large language models for robotics: A survey," *arXiv preprint arXiv:2311.07226*, 2023.

[33] H.-Y. Lee, *et al.*, "A distributed dynamic framework to allocate collaborative tasks based on capability matching in heterogeneous multirobot systems," *IEEE Trans. on Cognitive and Developmental Syst.*, vol. 16, no. 1, pp. 251–265, 2023.

[34] T. Gold, *et al.*, "Model predictive interaction control for robotic manipulation tasks," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 76–89, 2022.