Configuration-Adaptive Visual Relative Localization for Spherical Modular Self-Reconfigurable Robots

Yuming Liu, Qiu Zheng, Yuxiao Tu, Yuan Gao, Guanqi Liang, and Tin Lun Lam

***Abstract*— Spherical Modular Self-reconfigurable Robots (SMSRs) have been popular in recent years. Their Self-reconfigurable nature allows them to adapt to different environments and tasks, and achieve what a single module could not achieve. To collaborate with each other, relative localization between each module and assembly is crucial. Existing relative localization methods either have low accuracy, which is unsuitable for short-distance collaborations, or are designed for fixed-shape robots, whose visual features remain static over time. This paper proposes the first visual relative localization method for SMSRs. We first detect and identify individual modules of SMSRs, and adopt visual tracking to improve the detection and identification robustness. Using an optimization-based method, tracking result is then fused with odometry to estimate the relative pose between assemblies. To deal with the non-convexity of the optimization problem, we adopt semi-definite relaxation to transform it into a convex form. The proposed method is validated and analysed in real-world experiments. The overall localization performance and the performance under time-varying configuration are evaluated. The result shows that the relative position estimation accuracy reaches 2%, and the orientation estimation accuracy reaches 6.64°, and that our method surpasses the state-of-the-art methods.**

I. INTRODUCTION

SMSRs [1]–[4] have been popular in the field of self-reconfigurable robots. By leveraging their modular nature, SMSRs can combine and reconfigure themselves into various forms, allowing them to adapt to different environments and tasks. This collaborative behaviour enhances their overall capabilities beyond what a single module could achieve independently. To collaborate with each other, the relative pose between each module and assembly needs to be determined, thus putting forward the relative localization problem for SMSRs.

Some recent literature has proposed relative localization for modular robots. [5] proposed a graph convolutional network-based configuration detection method for FreeSN [1], which was later improved by fusing more sensor measurements in [6]. [7] proposed a decentralized localization method based on magnetic measurement for UBot. [8] introduced a 3D coordinate system to handle relative localization problems when orientation between each module is not accessible. However, these methods mainly focus on relative localization within a connected assembly, leaving relative localization between separate assemblies an open problem.

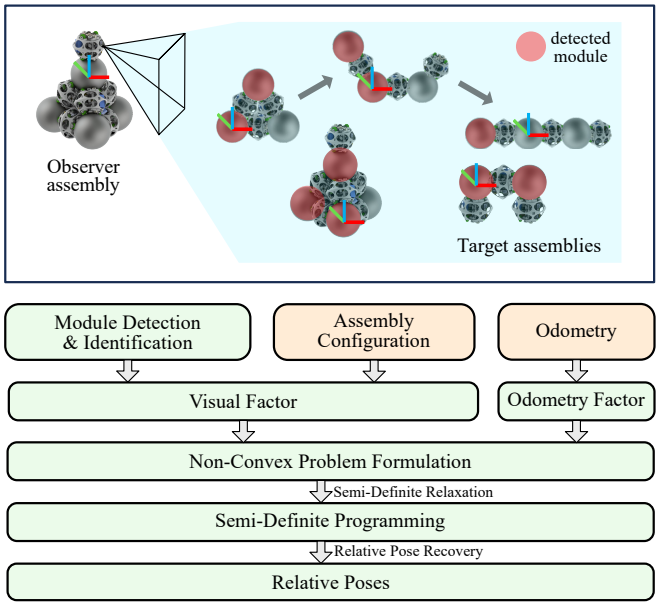

Fig. 1. The overview of our work. The observer detects and identifies the target assemblies' modules, whose positions are estimated and fused with assemblies' configuration and odometry to formulate the non-convex optimization problem for relative pose estimation. To deal with the non-convexity of the original problem, semi-definite relaxation is adopted to relax the problem into a semi-definite programming (SDP). The solution of the SDP is then used to recover the relative poses between the observer and target assemblies.

Many literatures have studied relative localization problem between separate robots, or robot swarms [9]–[26]. Most existing relative localization methods can be categorized as UWB-odometry-based methods and visual-object-detection methods [27]. For UWB-odometry-based methods [10]–[16], the distance between two robots measured by UWB is usually fused with odometry to estimate the relative pose between two robots. In SMSR applications, a typical scenario is that a group of SMSRs gradually come together and connect with each other to form an assembly. In this application, the localization accuracy requirement increases as the modules approach each other. However, UWB-odometry-based methods only provide decimeter-level accuracy and are not accurate enough for short-distance collaboration tasks.

Vision-based methods have also been proposed to solve robot swarm relative localization problems [17]–[26]. A widely used solution for visual localization was proposed

This work was supported in part by the National Natural Science Foundation of China under Grant 62073274; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023B1515020089; and in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society under Grant AC01202101103. *(Corresponding author: Tin Lun Lam.)*

Yuming Liu, Qiu Zheng, Yuxiao Tu, Yuan Gao, Guanqi Liang, and Tin Lun Lam are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172; Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, Guangdong, 518172, P.R. China (e-mail: yumingliu1@link.cuhk.edu.cn; 223015044@link.cuhk.edu.cn; yuxiaotu@link.cuhk.edu.cn; gaoyuan@cuhk.edu.cn; guanqiliang@link.cuhk.edu.cn; tllam@cuhk.edu.cn)

by Olson et al. [22]–[24]. They designed a square passive marker called AprilTag that not only encodes identity (ID) information but also provides pose information. To extract pose information from AprilTag, morphology features are used in the algorithm. However, the spherical surface of SMSRs brings distortion to those features, leading to estimation failure. Some other vision-based methods directly estimated the relative pose of the target robots, instead of markers on them. [19] detects keypoints on a drone and estimate its 6-degrees-of-freedom (DoF) pose. [20] applies yolov5 to detect drones at relatively low frequency and utilizes MOSSE [21] to generalize the low-frequency detection result to subsequent frames for higher detection frequency. These methods train neural networks with pictures of fixed-shape drones. However, the shape of a reconfigurable assembly may vary over time, which brings challenges to existing practices. Although the pose of a single module can be estimated using existing methods, the modules are generally small targets, so only estimating the pose of individual modules may produce inaccurate estimation. As a result, multiple modules' information should be fused to produce a more accurate pose estimation for the whole assembly.

In this work, we propose the first visual relative localization method for SMSRs.

Our approach leverages a monocular camera to detect each module's ID, position, and radius in the image. To handle cases where certain modules (nodes) fail to be identified, we employ a tracking algorithm to continuously identify those nodes. However, the small size of the modules presents challenges, such as inaccurate pose estimation or localization failure. To mitigate this, we integrate an optimization-based method that fuses tracking data from multiple modules with configuration and odometry information to achieve a more accurate pose estimation for the entire assembly. To address the non-convexity of the optimization problem, we adopt a semi-definite relaxation, transforming the problem into a convex form for more efficient and precise solving, and the relaxed solution is then recovered to provide relative pose estimation. Fig. 1 provides an overview of our proposed method. The key contributions of this paper are:

1) We develop a detection and identification method to detect and identify individual modules for SMSRs, so that our method can adapt to different configurations without the need of retraining the network.

2) Based on detection, odometry and configuration [6], we propose an optimization-based relative localization method, which is validated through real-world experiments.

II. PROBELM STATEMENT

In this work, we use FreeSN (Fig. 2) [6] as a representative for SMSRs. We define some useful frames and notations for FreeSN. We define a group of nodes and struts as an assembly if they are connected by magnetic force. An assembly i is denoted as A_i. For a node j in assembly A_i, denoted as N_{ij}, we define a horizontal frame H_{ij}. The origin of H_{ij} is located at the center of N_{ij}. Its z-axis is always parallel to gravity, and its rotation motion follows the

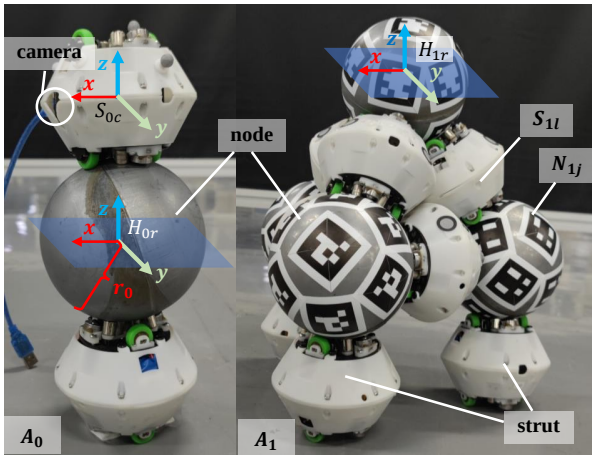

Fig. 2. Notation and frame definition of FreeSN. The observer assembly is on the left, and the target assembly is on the right.

node module around z-axis of H_{ij} ideally. Each assembly has a root node, whose horizontal frame is denoted as H_{ir}. Similarly, a strut l of A_i is denoted as S_{il}, and S_{il} is also used to denote its inertial frame. The node's radius is r_0.

In this paper, we study the relative localization problem between assemblies. Without loss of generality, we take two assemblies as an example, namely, an observer A_0 and a target A_1. One of the struts S_{0c} in the observer is equipped with a camera.

We use $\boldsymbol{p}_{F_2}^{F_1}$ and $\boldsymbol{R}_{F_2}^{F_1}$ to represent the position and orientation of frame F_2 w.r.t. frame F_1 respectively. We assume the extrinsic parameter of the camera $\boldsymbol{R}_c^{S_{0c}}$ and $\boldsymbol{p}_c^{S_{0c}}$, representing rotation and translation from frame S_{0c} to the camera frame c, is known. Moreover, applying our previous method [6], $\boldsymbol{R}_{S_{0c}}^{H_{0r}}$, $\boldsymbol{p}_{S_{0c}}^{H_{0r}}$, and $\boldsymbol{p}_{H_{1j}}^{H_{1r}}$ for all nodes j in A_1 can be calculated, i.e., the configuration of each assembly is available. As a result, if the relative position and orientation between H_{0r} and H_{1r} are determined, relative poses between all nodes and struts can be determined.

Moreover, since each strut is equipped with wheels and encoders, we also assume the odometry of the root node is available at all time k, denoted as $\hat{\boldsymbol{p}}_{H_{ir,k}}^{H_{ir,k-1}}$ and $\hat{\boldsymbol{R}}_{H_{ir,k}}^{H_{ir,k-1}}$ for $i \in \{0, 1\}$. The specific method by which the odometry is computed is beyond the scope of this paper.

Our goal is to estimate the pose of the target assembly's root horizontal frame H_{1r} w.r.t. the observer assembly's root horizontal frame H_{0r}, given image, configurations, and odometry measurement. Note that the pose to be estimated is 4-DoFs since the orientation between two horizontal frames only allows a single DoF. The relative position at time k is denoted as $\boldsymbol{p}_{H_{1r,k}}^{H_{0r,k}}$, and the relative angle at time k is denoted as $\theta_{H_{1r,k}}^{H_{0r,k}}$.

III. MODULE DETECTION AND IDENTIFICATION

To estimate the relative pose between two assemblies, we first detect modules. Although there are two types of modules in FreeSN, we only detect nodes, since their shape and other main features are rotation invariant, which makes it easier for detection and position estimation. In this section, we describe the detection and identification methods for nodes. We first

modify yolo series to detect nodes and use AprilTag to identify the detected nodes. However, AprilTag detection fails occasionally, leaving detected nodes unidentified. To solve this problem, we adopt a tracking algorithm to associate node detections across time steps, and to generalize ID information to frames where AprilTag detection fails.

A. Detection

We adopt yolo series [28], one of the state-of-the-art object detection methods. Different from general object detection tasks, we only need to detect nodes, which is a sphere. The traditional rectangular bounding box has a redundant DoF, when only 3 parameters are required to specify the location and size of a circle. So instead of using bounding box, we adopt bounding circle as the output of the network [29], [30]. Instead of predicting the center, width and height of a rectangle, we predict the center and radius of a circle. The detected bounding circle is denoted as $\boldsymbol{y} = [u, v, r]^T$, where u, v and r, represents the $x-$, $y-$coordinate and radius in the image plane.

B. Identification

AprilTag [22], [23] is used for identification in our method. Each node is allocated a unique ID and this ID is represented by the AprilTag attached to the node's surface. When a bounding circle is given, we run AprilTag detection algorithm in the bounding circle. The AprilTag detection result is filtered by the decision margin and hamming distance.

In [22], algorithms are developed to enable pose estimation using AprilTag itself. However, when an AprilTag is attached to a non-flat surface (e.g. sphere for node), its geometry is distorted, which leads to inaccurate pose estimation. Moreover, since the tag attached to a node is much smaller than the node itself, the relative estimation error using the algorithm in [22] is also larger as shown in section V-D. As a result, although AprilTag can also be used to estimate pose, we use it only for identification.

C. Tracking

Although AprilTag detection algorithm is designed to be robust, in our experiment, there is occasional detection failure due to low image quality, motion blur or other reasons. When AprilTag detection fails, the bounding circle in the image is useless unless it can be associated with a physical node. We tackle this vulnerability by adopting BoT-SORT [31], one of the state-of-the-art visual tracking methods, so that IDs can be assigned to node detections even facing occasional AprilTag detection failure. We denote the tracking result at time step k as $\mathcal{T}_k$.

IV. OPTIMIZATION-BASED RELATIVE LOCALIZATION

In this section, we propose an optimization-based method to fuse the tracking results with assembly configuration and odometry to estimate relative poses between the two assemblies. We first derive the residuals of vision and odometry terms to formulate the optimization problem. Then we adopt semi-definite relaxation to transform the optimization problem into a convex form for accurate and accelerated optimization. Finally, we recover the relative poses from the optimization result.

A. Visual Factor

According to the previous section, position and ID of each node in the image plane are obtained. We denote the position of the detected node with ID i in the image plane as (u_i, v_i) and its radius as r_i. Since we assume the camera is perfectly calibrated, the intrinsic parameter $\boldsymbol{K}$, and the focal length f are known. Recall that the real radius of a node is denoted as r_0. By the pinhole model of camera, the position measurement of the node i in c is given by

$$\hat{\boldsymbol{p}}^c_{N_i} = \boldsymbol{K}^{-1} \frac{f r_0}{r_i} [u_i, v_i, 1]^T \ , \tag{1}$$

We further represent the position of node i in c with assembly configurations and poses to be estimated as

$$\begin{aligned} \boldsymbol{p}^c_{N_i} =& \boldsymbol{R}_c^{S_{0c}{}^T} \Big(\boldsymbol{R}_{S_{0c}}^{H_{0r}{}^T} \Big(\boldsymbol{R}_{H_{0r}}{}^T \boldsymbol{R}_{H_{1r}} \big(\boldsymbol{p}_{N_i}^{H1r} \\ & - \boldsymbol{R}_{H_{1r}}{}^T (\boldsymbol{p}_{H_{0r}} - \boldsymbol{p}_{H_{1r}}) \big) - \boldsymbol{p}_{S_{0c}}^{H_{0r}} \Big) - \boldsymbol{p}_c^{S_{0c}} \Big) \ , \end{aligned} \tag{2}$$

where the omitted superscripts represent some arbitrary but unified frame and we drop the time index here for simplicity. The resulting residual is

$$\boldsymbol{e}_i^v = \boldsymbol{p}^c_{N_i} - \hat{\boldsymbol{p}}^c_{N_i}. \tag{3}$$

Observe from (2) that the rotation and position to be estimated are coupled. To decouple the rotation from position, we multiply both sides with $\boldsymbol{R}_{H_{0r}} \boldsymbol{R}_{S_{0c}}^{H_{0r}} \boldsymbol{R}_c^{S_{0c}}$, which gives

$$\begin{aligned} & \boldsymbol{R}_{H_{0r}} \boldsymbol{R}_{S_{0c}}^{H_{0r}} \boldsymbol{R}_c^{S_{0c}} \boldsymbol{e}_i^v \\ =& \boldsymbol{R}_{H_{1r}} \boldsymbol{p}_{N_i}^{H_{1r}} - (\boldsymbol{p}_{H_{0r}} - \boldsymbol{p}_{H_{1r}}) - \boldsymbol{R}_{H_{0r}} \left(\boldsymbol{p}_c^{H_{0r}} + \boldsymbol{R}_c^{H_{0r}} \hat{\boldsymbol{p}}^c_{N_i} \right) \\ \coloneqq& \boldsymbol{R}_{H_{1r}} \boldsymbol{p}_{N_i}^{H_{1r}} - (\boldsymbol{p}_{H_{0r}} - \boldsymbol{p}_{H_{1r}}) - \boldsymbol{R}_{H_{0r}} \boldsymbol{t}_i \end{aligned} \tag{4}$$

Using the vectorization trick, at timestep k, define $\boldsymbol{x}_k = \left[\boldsymbol{p}_{H_{0r,k}}{}^T, \boldsymbol{p}_{H_{1r,k}}{}^T, \mathrm{vec}\left(\boldsymbol{R}_{H_{0r,k}}\right)^T, \mathrm{vec}\left(\boldsymbol{R}_{H_{1r,k}}\right)^T \right]$. Applying (4), when $|\mathcal{T}_k| > 0$, the visual cost is defined as the visual residual norm as

$$c_k^v = \sum_{i \in \mathcal{T}_k} \boldsymbol{e}_{i,k}^v{}^T \boldsymbol{e}_{i,k}^v \coloneqq \boldsymbol{x}_k{}^T \left(\boldsymbol{V}_k \otimes \boldsymbol{I}_3 \right) \boldsymbol{x}_k \ , \tag{5}$$

where all the measurement-related terms are encoded in matrix $\boldsymbol{V}_k$. Although the configurations of the observer and target assembly may vary with time, (5) encodes the configurations along with all the visual measurements in the matrix $\boldsymbol{V}_k$ and gives a unified representation of the visual cost, which enables configuration-adaptive relative localization.

B. Odometry Factor

It is assumed that the odometry measurements of both assemblies are available, i.e., at time k, for $j \in \{0, 1\}$ odometry measurements $\hat{\boldsymbol{p}}_{H_{jr,k}}^{H_{jr,k-1}}$ and $\hat{\boldsymbol{R}}_{H_{jr,k}}^{H_{jr,k-1}}$ are available. We can derive the corresponding $\boldsymbol{p}_{H_{jr,k}}^{H_{jr,k-1}}$ and $\boldsymbol{R}_{H_{jr,k}}^{H_{jr,k-1}}$ from poses to be estimated as

$$\begin{aligned} \boldsymbol{p}_{H_{jr,k}}^{H_{jr,k-1}} &= \boldsymbol{R}_{H_{jr,k-1}}{}^T \left(\boldsymbol{p}_{H_{jr,k}} - \boldsymbol{p}_{H_{jr,k-1}} \right) \\ \boldsymbol{R}_{H_{jr,k}}^{H_{jr,k-1}} &= \boldsymbol{R}_{H_{jr,k-1}}{}^T \boldsymbol{R}_{H_{jr,k}} \end{aligned} \tag{6}$$

Similar to (4), we decouple the poses, and the residuals of position and orientation are given by

$$\begin{aligned}\boldsymbol{R}_{H_{jr,k-1}}\boldsymbol{e}_{H_{jr,k}}^{H_{jr,k-1}} &\coloneqq \boldsymbol{p}_{H_{jr,k}} - \boldsymbol{p}_{H_{jr,k-1}} - \boldsymbol{R}_{H_{jr,k-1}}\hat{\boldsymbol{p}}_{H_{jr,k}}^{H_{jr,k-1}} \\ \boldsymbol{R}_{H_{jr,k-1}}\boldsymbol{\epsilon}_{H_{jr,k}}^{H_{jr,k-1}} &\coloneqq \boldsymbol{R}_{H_{jr,k}} - \boldsymbol{R}_{H_{jr,k-1}}\hat{\boldsymbol{R}}_{H_{jr,k}}^{H_{jr,k-1}}\end{aligned} \tag{7}$$

As a result, the odometry cost is defined by the residual norms as

$$\begin{aligned} c_k^o &= \sum_{j\in\{0,1\}} {\boldsymbol{e}_{H_{jr,k}}^{H_{jr,k-1}}}^T \boldsymbol{e}_{H_{jr,k}}^{H_{jr,k-1}} + \left\|\boldsymbol{\epsilon}_{H_{jr,k}}^{H_{jr,k-1}}\right\|_F^2 \\ &= \left[{\boldsymbol{x}_{k-1}}^T, {\boldsymbol{x}_k}^T\right](\boldsymbol{O}_k \otimes \boldsymbol{I}_3)\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{x}_k\end{bmatrix}\end{aligned} \tag{8}$$

where $\|\cdot\|_F$ represents the Frobenius norm, and all the measurement-related terms are encoded in matrix $\boldsymbol{O}_k$.

C. Problem Formulation and Semi-Definite Relaxation

Based on the visual and odometry cost derived in (5) and (8), we can formulate the optimization problem for relative localization. Denote the relative pose of $(K+1)$ frames as a vector

$$\boldsymbol{x}_{k:k+K} = \left[{\boldsymbol{x}_k}^T, {\boldsymbol{x}_{k+1}}^T, \cdots, {\boldsymbol{x}_{k+K}}^T\right]^T \tag{9}$$

Utilizing (5) and (8), the sliding-window optimization problem is defined as follows

$$\begin{aligned}\mathcal{P}_1: \quad \min_{\boldsymbol{x}_{k:k+K}} \quad & \sum_{t=k}^{k+K} c_t^v + \sum_{t=k+1}^{k+K} c_t^o\,, \\ \text{s.t.} \quad & \boldsymbol{R}_{H_{jr,t}} \in \mathrm{SO}(3)\,, \quad \boldsymbol{R}_{H_{jr,t}}\boldsymbol{e}_z = \boldsymbol{e}_z \\ & \forall j \in \{0,1\}\,, \quad t = k, k+1, \ldots, k+K,\end{aligned} \tag{10}$$

where $\mathrm{SO}(n)$ represents the n-dimensional special orthogonal group, and $\boldsymbol{e}_z$ is the natural basis of z-axis. By utilizing the measurement matrices $\boldsymbol{V}_k$ and $\boldsymbol{O}_k$, the objective of (10) can be written in quadratic form as

$$\sum_{t=k}^{k+K} c_t^v + \sum_{t=k+1}^{k+K} c_t^o = \boldsymbol{x}_{k:k+K}^T(\boldsymbol{M}_{k:k+K} \otimes \boldsymbol{I}_3)\boldsymbol{x}_{k:k+K}\,, \tag{11}$$

where the $\boldsymbol{M}_{k:k+K-1}$ is defined block-wise by $\boldsymbol{O}_k$ and $\boldsymbol{V}_k$. We further define the positions and rotations as separate variables as

$$\begin{aligned}\mathfrak{p} &\coloneqq \left[\boldsymbol{p}_{H_{0r,k}}^T, \boldsymbol{p}_{H_{1r,k}}^T, \boldsymbol{p}_{H_{0r,k+1}}^T, \cdots, \boldsymbol{p}_{H_{1r,k+K}}^T\right]^T \in \mathbb{R}^{6K} \\ \mathfrak{R} &\coloneqq \left[\boldsymbol{R}_{H_{0r,k}}, \boldsymbol{R}_{H_{1r,k}}, \boldsymbol{R}_{H_{0r,k+1}}, \cdots, \boldsymbol{R}_{H_{1r,k+K}}\right] \in \mathbb{R}^{3\times 6K}\end{aligned} \tag{12}$$

Then with a proper permutation matrix $\boldsymbol{T}$, $\mathcal{P}_1$ can be re-written as

$$\begin{aligned}\mathcal{P}_2: \min_{\mathfrak{p},\mathfrak{R}} \quad & \begin{bmatrix}\mathfrak{p}\\ \mathrm{vec}(\mathfrak{R})\end{bmatrix}^T (\boldsymbol{T}\boldsymbol{M}\boldsymbol{T}^T \otimes \boldsymbol{I}_3)\begin{bmatrix}\mathfrak{p}\\ \mathrm{vec}(\mathfrak{R})\end{bmatrix} \\ \text{s.t.} \quad & \boldsymbol{R}_{H_{jr,t}} \in \mathrm{SO}(3)\,, \quad \boldsymbol{R}_{H_{jr,t}}\boldsymbol{e}_z = \boldsymbol{e}_z \\ & \forall j \in \{0,1\}\,, \quad t = k, k+1, \ldots, k+K,\end{aligned} \tag{13}$$

We segment the cost matrix above as

$$\boldsymbol{T}\boldsymbol{M}\boldsymbol{T}^T = \left[\begin{array}{c|c}\boldsymbol{M}_{pp} & \boldsymbol{M}_{pR}\\ \hline \boldsymbol{M}_{pR}^T & \boldsymbol{M}_{RR}\end{array}\right] \tag{14}$$

And define $\boldsymbol{Q} = \boldsymbol{M}_{RR} - \boldsymbol{M}_{pR}^T\boldsymbol{M}_{pp}^\dagger\boldsymbol{M}_{pR}$, where $*^\dagger$ is the Moore–Penrose inverse. By [32], the position terms and rotation terms in $\mathcal{P}_2$ can be decoupled, and the position can be estimated in closed form if the optimal rotation is given. The optimal rotation can be obtained by solving the following relaxed problem.

$$\mathcal{P}_3: \min_{\boldsymbol{Z}\in\mathbb{S}_+^{6K}} \quad \mathrm{tr}(\boldsymbol{Q}\boldsymbol{Z}) \tag{15a}$$

$$\text{s.t.} \quad \boldsymbol{Z}[3l:3l+3, 3l:3l+3] = \boldsymbol{I}_3 \quad l = 1,2,\ldots,2K \tag{15b}$$

$$\boldsymbol{Z}[3i:3i+3, 3j:3j+3] = \begin{bmatrix}* & * & 0\\ * & * & 0\\ 0 & 0 & 1\end{bmatrix} \quad i,j = 1,2,\ldots,2K \tag{15c}$$

where $\boldsymbol{Z} = \mathfrak{R}^T\mathfrak{R}$; $\mathbb{S}_+^n$ represents set of n-dimensional positive semi-definite matrices; $\mathrm{tr}(\cdot)$ is the trace of a matrix. Different from the original relaxation in [32], there is an additional convex constraint (15c) in $\mathcal{P}_3$. As mentioned in section II, we only consider single rotation DoF along z-axis, and constraint (15c) ensures the rotation DoF. Although additional constraints are enforced, $\mathcal{P}_3$ is still a SDP problem, and can be solved accurately and efficiently with numerical solvers like MOSEK.

D. Relative Pose Recovery

Since we only consider 1-DoF rotation along the z-axis, any estimated rotation matrix $\boldsymbol{R} \in \mathrm{SO}(3)$ can be written as

$$\boldsymbol{R} = \left[\begin{array}{c|c}\underline{\boldsymbol{R}} & \boldsymbol{0}\\ \hline \boldsymbol{0}^T & 1\end{array}\right]\,, \tag{16}$$

where $\underline{\boldsymbol{R}} \in \mathrm{SO}(2)$ is the free parts of $\boldsymbol{R}$, and other parts are fixed as constraints as in (15c). After solving $\mathcal{P}_3$, we extract the free parts of $\boldsymbol{Z}^*$ as $\underline{\boldsymbol{Z}}^*$. Since $\boldsymbol{Z} = \mathfrak{R}^T\mathfrak{R}$, we have $\underline{\boldsymbol{Z}}^* = \underline{\mathfrak{R}}^{*T}\underline{\mathfrak{R}}^*$, where $\underline{\mathfrak{R}}^* \in \mathbb{R}^{2\times 4K}$. As a result, rank-2 decomposition is performed for $\underline{\boldsymbol{Z}}^*$ to obtain $\underline{\boldsymbol{Z}}^* = \boldsymbol{Y}^{*T}\boldsymbol{Y}^*$, where $\boldsymbol{Y}^* \in \mathbb{R}^{2\times 4K}$. And each 2-D rotation matrix $\underline{\boldsymbol{R}}_{H_{jr,k}}^*$ in $\underline{\mathfrak{R}}^*$ is recovered from the corresponding component $\boldsymbol{Y}_{H_{jr,k}}^*$ in $\boldsymbol{Y}^*$ as

$$\underline{\boldsymbol{R}}_{H_{jr,k}}^* = \boldsymbol{U}\begin{bmatrix}1 & 0\\ 0 & \det(\boldsymbol{U}\boldsymbol{V}^T)\end{bmatrix}\boldsymbol{V}^T\,, \tag{17}$$

where $\boldsymbol{Y}_{H_{jr,k}}^* = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the singular value decomposition of $\boldsymbol{Y}_{H_{jr,k}}^*$. And the original rotation matrices $\mathfrak{R}^*$ can be constructed by (16). By [32], the optimal position estimation is given by the closed-form solution as

$$\mathfrak{p}^* = -\mathrm{vec}\left(\mathfrak{R}^*\boldsymbol{M}_{pR}^T\boldsymbol{M}_{pp}^\dagger\right)\,. \tag{18}$$

Finally, the relative pose of the two assemblies' root horizontal frame $H_{0r,K}$ and $H_{1r,K}$ is given by

$$\begin{aligned}{\boldsymbol{R}_{H_{1r,K}}^{H_{0r,K}}}^* &= {\boldsymbol{R}_{H_{0r,K}}^*}^T\boldsymbol{R}_{H_{1r,K}}^* \\ {\boldsymbol{p}_{H_{1r,K}}^{H_{0r,K}}}^* &= {\boldsymbol{R}_{H_{0r,K}}^*}^T\left(\boldsymbol{p}_{H_{1r,K}}^* - \boldsymbol{p}_{H_{0r,K}}^*\right)\end{aligned} \tag{19}$$

The relative angle ${\theta_{H_{1r,k}}^{H_{0r,k}}}^*$ can be calculated by converting ${\boldsymbol{R}_{H_{1r,K}}^{H_{0r,K}}}^*$ to Euler angle and taking the yaw component.

V. EXPERIMENT

A. Data Collection and Network Training

We assume that nodes appear uniformly and randomly within the camera's field of view (FoV). Therefore, a total of 1568 images are collected according to this distribution, 80% among which are used as training set, and the rest as test set. As mentioned before, we adopt yolo series as the detector with our customized bounding circle head. We train detection network using mmyolo [33] on the dataset, and use mmdeploy [34] to deploy the model after training for accelerated inference.

B. Localization Experiment Setup

We use ESP32-CAM as the camera, and it's also equipped with a Wi-Fi module. The raw images (1024 pixels $\times$ 768 pixels) are sent to a computer at about 5Hz through 2.4G Wi-Fi. The camera is mounted on one of the struts of the observer as shown in Fig. 2. The odometry data is obtained by integrating the velocity measured by encoders of the differential wheels of struts, and is sent to the computer at about 10Hz through 2.4G Wi-Fi. All detection, identification, tracking, and optimization algorithms run on the computer, whose central processing unit (CPU) is Intel i5-13600KF and graphics processing unit (GPU) is NVIDIA RTX4070Ti. We set the optimization window $K = 5$, and the algorithm runs at about 27Hz with offline data on the computer. Since the image transmission frequency is about 5Hz, the algorithm runs at about 5Hz with online data. A motion capture (OptiTrack) serves as the experiment's ground truth system, and the ground truth's localization accuracy is about 0.8mm.

C. Baselines

1) **AprilTag** [22], [23]: AprilTag is a widely used visual marker that provides both identity and pose information, and we also use it for identification in our method. In our experiment, it serves as one of the baselines. We detect the tag on the target's root node, and then compute the root's center position based on the pose information the tag provides.

2) **Bearing** [35]: Bearing-based method is another popular method that fuses visual detections with odometry. We use one state-of-the-art method [35] as another baseline. We use the detection of the target's root node to compute the bearing information and then fuse it with odometry to produce the estimation of the root's pose.

D. Localization Experiment Result

In this experiment, we evaluate the overall algorithm performance. As shown in Fig. 3 (a), the target assemblies have different configurations (robot 1 and robot 2), and are placed on different terrains (robot 2 and robot 3). And during the experiment, the observer is placed on a table about 0.75m high and moves horizontally. We evaluate our method and the baselines in the described settings. The qualitative result for position trajectories is shown in Fig. 3 (b), and the orientation trajectories are shown in Fig. 3 (c) - (e). From the qualitative result, compared with **Bearing**, the proposed method demonstrates significant accuracy improvement, since unlike bearing-based methods, we are able to utilize the bounding circle radius as depth measurement, which provides more information about the target's pose. Compared with **AprilTag**, our method is both more accurate and provides smoother estimation results, due to the fusion with odometry and the use of larger visual features.

TABLE I
COMPARISON BETWEEN OUR METHOD AND BASELINES

Method	Positional Error [m] ↓	Relative Positional Error [%] ↓	Angular Error [°] ↓	FPS [Hz] ↑
AprilTag [22]	0.112	5.384	/	13.92
Bearing [35]	0.304	16.942	22.035	25.53
Ours	**0.039**	**2.092**	**6.642**	**27.24**

The quantitative result is shown in Table I. We compare the **mean positional error**, **mean relative positional error**, **mean absolute angular error**, and the **frame per second (FPS)**. The relative positional error is computed by dividing the positional error by the current observer-target distance. From table I, our method demonstrates significant accuracy improvements compared to baselines, and reaches 2% in terms of relative positional error. Apart from accuracy, our method also demonstrates higher or comparable FPS, which is more than sufficient in our settings.

For FreeSN, which we use in our experiments, its node's radius $r_0 = 6\text{cm}$. The experiments show that our visual localization algorithm reaches a comparable accuracy level as the module size of our SMSR overall and outperforms some other vision-based methods. Consequently, our visual localization method is better suited for SMSRs.

E. Localization for Real-Time Reconfiguration

Our method can adapt to different, and possibly time-varying, configurations of the target assembly. We use the same setting as Fig. 3 (a), but keep the observer still and replace the targets with an assembly shown in Fig. 4 (a). During the experiment, the target assembly automatically changes its configuration as shown in Fig. 4 (a)-(f). The red circles in Fig. 4 represent that the corresponding node is detected and identified. The localization accuracy is also shown in Fig. 4. From Fig. 4, we can observe two prime accuracy improvements, roughly at (b) and (e). Before (b), there are two nodes used for localization, however, due to occlusion, the detection accuracy of one of the two nodes is relatively low. But since (b), another non-occluded node is detected, leading to localization accuracy improvement. Similarly, at (e), three non-occluded nodes are detected, leading to another accuracy improvement. From the experiment, we observe that the accuracy of our method is closely related to the number and quality of node detection of the target assembly. And most importantly, by fusing the configuration and visual detections, our method enables localization for real-time reconfiguration of SMSRs.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed a visual relative localization method that can adapt to different configurations of SMSRs. De-

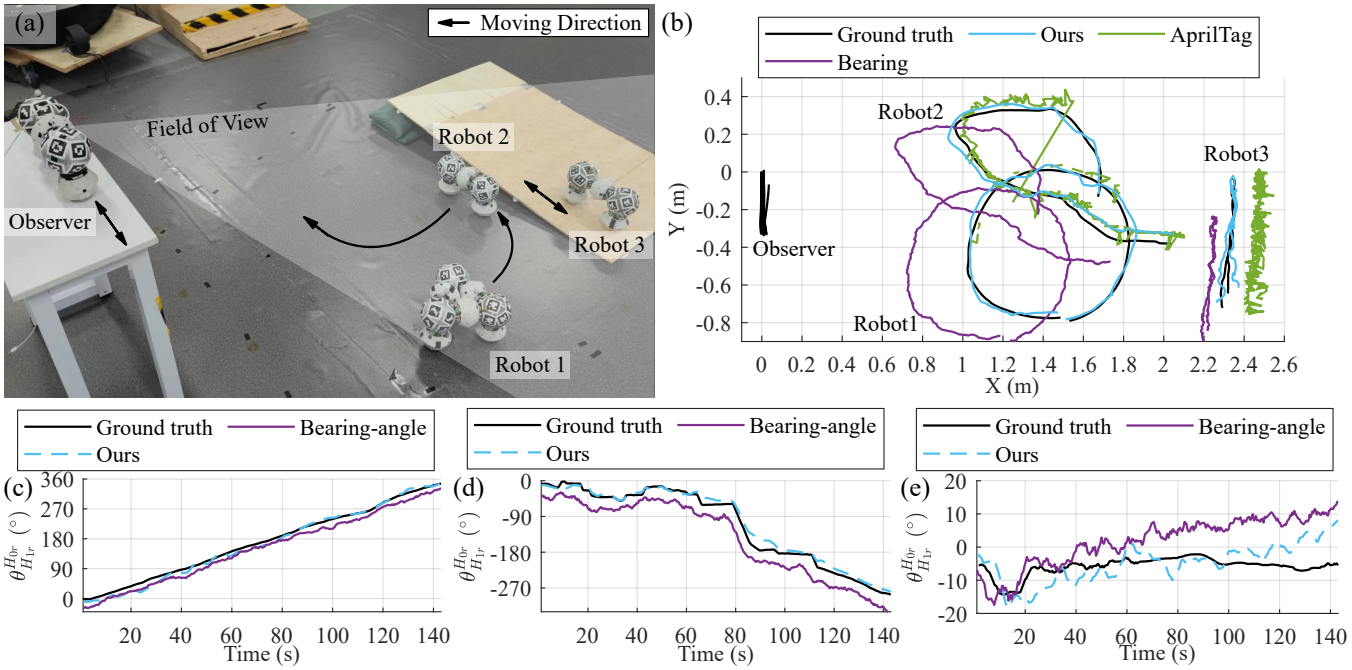

Fig. 3. Settings and results localization experiments. (a) Experiment setting of the localization experiment, where there is 1 observer and 3 target assemblies of different configurations. All of the assemblies are moving in different trajectories and on different terrain as shown in the figure. (b) Ground truth and estimated trajectories of the robots, where our method is compared against **AprilTag** [22], [23] and **Bearing** [35]. (c) - (e) Ground truth and estimated $\theta_{H_{1r}}^{H_{0r}}$ of target assembly 1 - 3.

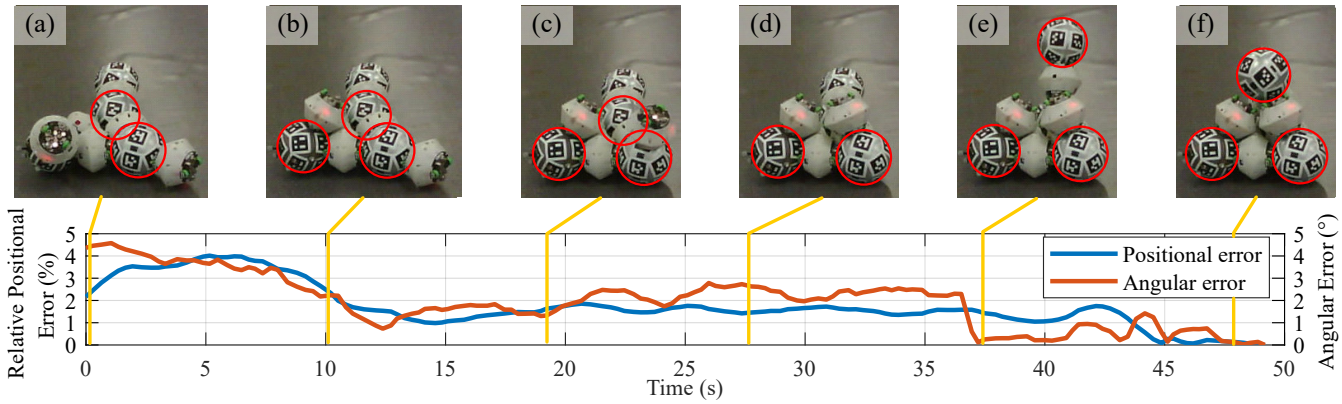

Fig. 4. Experiment result of the reconfiguration experiment. Down: The relative positional error and angular error in the experiment. (a)-(f): Configuration of the target assembly at the corresponding time steps, where the red circle represents the node is both detected and identified.

tection, identification, and tracking methods were designed to acquire module position and ID, which were fused with odometry and assemblies' configuration to formulate an optimization-based localization problem. Semi-definite relaxation was adopted to transform the non-convex formulation into an SDP problem. Finally, the relative poses were recovered from the solution of the SDP problem. To analyze the performance of the proposed algorithm, we conducted comprehensive experiments and the overall relative position estimation accuracy reaches 2% and the orientation estimation accuracy reaches 6.642°, which is superior to baselines. We also conducted localization experiments for real-time SMSR reconfiguration, and demonstrated that our method can adapt to different and time-varying SMSR configurations. Moreover, our method can also be applied to other SMSRs like Freebot [2] and Snailbot [4] just by slight modification.

There are also some limitations in our method that can be improved in the future, including: 1) The current algorithm only considers a single observer. When multiple assemblies exist, the visual measurements between each assembly are not fully exploited. 2) The current algorithm is centralized, which limits its robustness and scalability. In the future, we will decentralize our algorithm and consider more observers to fully utilize all visual measurements for improved robustness and scalability.

REFERENCES

[1] Yuxiao Tu, Guanqi Liang, and Tin Lun Lam. Freesn: A freeform strut-node structured modular self-reconfigurable robot-design and implementation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4239–4245. IEEE, 2022.

[2] Guanqi Liang, Haobo Luo, Ming Li, Huihuan Qian, and Tin Lun Lam. Freebot: A freeform modular self-reconfigurable robot with arbitrary connection point-design and implementation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6506–6513. IEEE, 2020.

[3] Guanqi Liang, Lijun Zong, and Tin Lun Lam. Disg: Driving-integrated spherical gear enables singularity-free full-range joint motion. *IEEE Transactions on Robotics*, 2023.

[4] Da Zhao and Tin Lun Lam. Snailbot: A continuously dockable modular self-reconfigurable robot using rocker-bogie suspension. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4261–4267. IEEE, 2022.

[5] Yuxiao Tu, Guanqi Liang, and Tin Lun Lam. Graph convolutional network based configuration detection for freeform modular robot using magnetic sensor array. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4252–4258. IEEE, 2021.

[6] Yuxiao Tu and Tin Lun Lam. Configuration identification for a freeform modular self-reconfigurable robot-freesn. *IEEE Transactions on Robotics*, 2023.

[7] Dongyang Bie, Iqbal Sajid, Jianda Han, Jie Zhao, and Yanhe Zhu. Natural growth-inspired distributed self-reconfiguration of ubot robots. *Complexity*, 2019(1):2712015, 2019.

[8] Benoît Piranda, Frédéric Lassabe, and Julien Bourgeois. Disco: A multiagent 3d coordinate system for lattice based modular self-reconfigurable robots. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11915–11921. IEEE, 2023.

[9] Xiyue Guo, Junjie Hu, Junfeng Chen, Fuqin Deng, and Tin Lun Lam. Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment. *IEEE Robotics and Automation Letters*, 6(4):8349–8356, 2021.

[10] Yue Wang, Muhan Lin, Xinyi Xie, Yuan Gao, Fuqin Deng, and Tin Lun Lam. Asymptotically efficient estimator for range-based robot relative localization. *IEEE/ASME Transactions on Mechatronics*, 2023.

[11] Thien Hoang Nguyen and Lihua Xie. Relative transformation estimation based on fusion of odometry and uwb ranging data. *IEEE Transactions on Robotics*, 2023.

[12] Thien Hoang Nguyen and Lihua Xie. Estimating odometry scale and uwb anchor location based on semidefinite programming optimization. *IEEE Robotics and Automation Letters*, 7(3):7359–7366, 2022.

[13] Luca Barbieri, Mattia Brambilla, Andrea Trabattoni, Stefano Mervic, and Monica Nicoli. Uwb localization in a smart factory: Augmentation methods and experimental assessment. *IEEE Transactions on Instrumentation and Measurement*, 70:1–18, 2021.

[14] Valerio Brunacci, Alessio De Angelis, Gabriele Costante, and Paolo Carbone. Development and analysis of a uwb relative localization system. *IEEE Transactions on Instrumentation and Measurement*, 72:1–13, 2023.

[15] Ming Li, Guanqi Liang, Haobo Luo, Huihuan Qian, and Tin Lun Lam. Robot-to-robot relative pose estimation based on semidefinite relaxation optimization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4491–4498, 2020.

[16] Ming Li, Tin Lun Lam, and Zhiyong Sun. 3-d inter-robot relative localization via semidefinite optimization. *IEEE Robotics and Automation Letters*, 7(4):10081–10088, 2022.

[17] Viktor Walter, Nicolas Staub, Antonio Franchi, and Martin Saska. Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs. *IEEE Robotics and Automation Letters*, 4(3):2637–2644, 2019.

[18] Jiri Horyna, Viktor Walter, and Martin Saska. Uvdar-com: Uv-based relative localization of uavs with integrated optical communication. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1302–1308. IEEE, 2022.

[19] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2011–2018. IEEE, 2017.

[20] Hao Xu, Yichen Zhang, Boyu Zhou, Luqi Wang, Xinjie Yao, Guotao Meng, and Shaojie Shen. Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms. *IEEE Transactions on Robotics*, 38(6):3374–3394, 2022.

[21] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010.

[22] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.

[23] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pages 3400–3407. IEEE, 2011.

[24] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. Flexible layouts for fiducial tags. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2019.

[25] Maxim Pavliv, Fabrizio Schiano, Christopher Reardon, Dario Floreano, and Giuseppe Loianno. Tracking and relative localization of drone swarms with a vision-based headset. *IEEE Robotics and Automation Letters*, 6(2):1455–1462, 2021.

[26] Rong Ou, Guanqi Liang, and Tin Lun Lam. Fpecmv: Learning-based fault-tolerant collaborative localization under limited connectivity. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11095–11102, 2023.

[27] Siyuan Chen, Dong Yin, and Yifeng Niu. A survey of robot swarms' relative localization method. *Sensors*, 22(12):4424, 2022.

[28] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO. `https://github.com/ultralytics/ultralytics`, January 2023.

[29] Ethan H Nguyen, Haichun Yang, Ruining Deng, Yuzhe Lu, Zheyu Zhu, Joseph T Roland, Le Lu, Bennett A Landman, Agnes B Fogo, and Yuankai Huo. Circle representation for medical object detection. *IEEE transactions on medical imaging*, 41(3):746–754, 2021.

[30] Haichun Yang, Ruining Deng, Yuzhe Lu, Zheyu Zhu, Ye Chen, Joseph T Roland, Le Lu, Bennett A Landman, Agnes B Fogo, and Yuankai Huo. Circlenet: Anchor-free glomerulus detection with circle representation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, pages 35–44. Springer, 2020.

[31] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

[32] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.

[33] MMYOLO Contributors. MMYOLO: OpenMMLab YOLO series toolbox and benchmark. `https://github.com/open-mmlab/mmyolo`, 2022.

[34] MMDeploy Contributors. Openmmlab's model deployment toolbox. `https://github.com/open-mmlab/mmdeploy`, 2021.

[35] Yingjian Wang, Xiangyong Wen, and Fei Gao. Certifiable mutual localization and trajectory planning for bearing-based robot swarm. *arXiv preprint arXiv:2401.07784*, 2024.