POLITECNICO DI TORINO
Repository ISTITUZIONALE

Simultaneous Robot and Feature Localization Via Angle-Based Sensing With Barycentric Coordinate Representation

Original
Simultaneous Robot and Feature Localization Via Angle-Based Sensing With Barycentric Coordinate Representation / Wu, J., Zino, L., Lin, Z., Rizzo, A.. - In: IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. - ISSN 2373-776X. - ELETTRONICO. - 12:(2026), pp. 1184-1199. [10.1109/tsipn.2026.3739592]

Availability:
This version is available at: 11583/3016236 since: 2026-10-18T07:18:50Z

Publisher:
IEEE

Published
DOI:10.1109/tsipn.2026.3739592

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

16 November 2026

Simultaneous Robot and Feature Localization via Angle-Based Sensing with Barycentric Coordinate Representation

Jinze Wu, *Student Member, IEEE*, Lorenzo Zino, *Senior Member, IEEE*, Zhiyun Lin, *Fellow, IEEE*, and Alessandro Rizzo, *Senior Member, IEEE*

***Abstract*—This paper studies the Simultaneous Robot and Feature Localization (SRAFL) problem leveraging angle-based sensing from a monocular camera, and addresses it from a networked signal processing perspective. By modeling the robot-feature interaction as a sensing graph, in which the robot and environmental features function as nodes connected through bearing angle measurements, we reformulate the inherently nonlinear SRAFL problem into a dynamic network localization task. Based on this formulation, we propose a Barycentric Linear method, termed BL-SRAFL, which exploits barycentric coordinate representations to obtain a linear localization framework for jointly estimating the robot trajectory and the Euclidean positions of environmental features. The proposed method admits direct and efficient iterative solutions with linear computational and memory complexity, making it particularly suitable for resource-constrained applications. Rigorous theoretical analysis is provided to characterize the solvability and establish the convergence of the proposed method. For scenarios requiring enhanced accuracy, we further introduce the BL-SRAFL+BA pipeline, which integrates an optional Bundle Adjustment (BA) refinement step. Numerical simulations and real-world experiments demonstrate the robustness and effectiveness of the proposed methods.**

***Index Terms*—Localization, sensor network, angle measurement, barycentric coordinate**

I. Introduction

MOBILE robots have attracted significant attention for their diverse applications, spanning from industrial manufacturing, social sciences, medical services, to space exploration [1]–[3]. At the core of mobile robotics lies the task of localization: determining the robot's position within its environment using sensors [4], [5]. In particular, cameras are frequently adopted for their affordability, lightweight nature, and low power consumption. This has spurred considerable interest in vision-based robot localization [6]–[8], where vision information is the sole external sensory input.

J. Wu, L. Zino, and A. Rizzo are with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy ({jinze.wu, lorenzo.zino, alessandro.rizzo}@polito.it). J. Wu and Z. Lin are with the Guangdong Provincial Key Laboratory of Fully Actuated System Control Theory and Technology, School of Automation and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen, China (linzy@sustech.edu.cn). Z. Lin is also with the Peng Cheng Laboratory, Shenzhen, China. J. Wu is currently with Guangzhou Automobile Group Co., Ltd., Guangzhou, China. This work was supported in part by the Joint Funds of the National Natural Science Foundation of China under Grant U23A20325; the Guangdong Science and Technology Program under Grant 2024B1212010002; and the Shenzhen Science and Technology Program under Grant KQTD20221101093557010. *(Corresponding authors: Zhiyun Lin and Alessandro Rizzo.)*

In this work, we consider robot localization using limited visual information captured by a monocular camera. Existing vision-based localization methods can be broadly categorized into two classes. Map-based visual localization estimates the robot pose relative to a prebuilt map or a set of landmarks with known positions [6], [7], and therefore relies on an externally provided spatial reference without reconstructing previously unknown environmental features. By contrast, visual Simultaneous Localization and Mapping (SLAM) and Structure-from-Motion (SfM) jointly estimate the camera motion and scene structure when no prior map is available [8]–[12]. Representative modern approaches include the feature-based ORB-SLAM3 system [12], the direct sparse method DSO [13], and incremental or global SfM pipelines such as COLMAP [14] and GLOMAP [15]. These methods operate directly on image data and typically recover camera trajectories and scene structure through nonlinear reprojection, photometric, or graph optimization.

The problem considered in this paper belongs to the latter category, but focuses on localization from sparse bearing-only measurements. Specifically, we assume that repeatably identifiable sparse features and their normalized bearing measurements are available from the visual front-end. Starting from a metric initialization that fixes the global coordinate frame and metric scale, the proposed method jointly estimates the robot trajectory and the positions of previously unknown environmental features. Unlike map-based visual localization, it does not require a pre-surveyed feature map; unlike complete visual SLAM or SfM pipelines, our contribution does not lie in feature extraction or data association, but in the subsequent geometric estimation problem. In particular, we reformulate the robot-feature geometry from a network localization perspective, organize the sparse bearing observations into directed sensing graphs, and exploit barycentric coordinate representations to derive sparse linear constraints for joint robot and feature localization without nonlinear joint optimization in the front-end. This formulation is particularly relevant when a limited number of persistent and reliably identifiable features are available and computational or memory resources are constrained, as it relies on sparse monocular bearing measurements and admits linearly scalable online computation and memory. We refer to this task as the Simultaneous Robot and Feature Localization (SRAFL) problem, emphasizing the dual objective of estimating the position of both the robot and environmental features.

To obtain a tractable formulation, we reinterpret the SRAFL problem from a network localization perspective and model the robot-feature interactions through an angle-based sensing graph. Specifically, the robot positions along its trajectory and the environmental features are represented as graph nodes, while the visual observations are expressed as local bearing angle measurements acquired by the robot at each position. As the robot moves and its field of view changes, the resulting directed sensing graph evolves over time, thereby transforming the original SRAFL problem into a dynamic network-localization problem. This reformulation enables the use of graph-theoretic tools and network localization principles to analyze and solve the SRAFL problem, while naturally establishing its connection to the related network localization literature.

Network localization has been extensively studied through both nonlinear and linear formulations. Early approaches typically cast localization as a nonlinear estimation problem and employed distributed optimization, maximum-likelihood estimation, Bayesian inference, or related techniques [16]–[21]. Although these approaches can accommodate realistic measurement models, they generally require iterative nonlinear optimization and may be sensitive to initialization. To alleviate these limitations, linear localization methods were subsequently developed by replacing nonlinear geometric constraints with affine or barycentric coordinate relations. Representative barycentric methods enable distributed localization with a small set of anchor nodes [22]–[24], and have been further extended to three-dimensional localization, node localizability, reduced communication, and finite-time cooperative localization [25]–[32].

Despite these advances, most existing network localization methods assume that unknown nodes can actively sense, communicate with, or perform computation using information from neighboring nodes. Moreover, existing studies on dynamic topologies either consider a mobile anchor relying on range measurements or are restricted to two-dimensional undirected networks [25], [26]. In contrast, the SRAFL problem involves passive environmental features that neither sense, communicate, nor perform computation; all measurements are acquired by the moving robot, resulting in a directed and dynamic topology. The problem further combines three-dimensional bearing angle measurements with the simultaneous estimation of the robot trajectory and passive feature positions. These fundamental differences prevent the direct application of existing network localization methods.

Motivated by these differences, we develop a barycentric linear formulation tailored to the SRAFL problem. By exploiting barycentric coordinate representations [33], the bearing angle measurements induced by the sensing graph are transformed into linear algebraic constraints. The resulting sparse linear system jointly recovers the robot trajectory and environmental feature positions, providing an efficient and scalable solution suitable for resource-constrained sensing platforms.

Specifically, we consider a mobile robot equipped with a monocular camera in the region where the sparse features with unknown positions are situated. As the robot moves around, it captures bearing angle measurements of features with respect to its own position through the images from the monocular camera. Then, the robot is tasked with iteratively updating its own trajectory and estimating the positions of these features based on these step-by-step measurements. The specific process can be described as follows. As the robot moves, certain features become observable by the robot at two consecutive time steps, comprising both those with known positions and those with unknown positions. Utilizing local bearing angle measurements and the known positions of features, the robot can estimate its own trajectory and the positions of these previously unknown features based on the barycentric coordinate representation. Once a feature's position is determined, it is then used to locate further features in an iterative fashion. This process is continuously iterated as the robot observes novel features, until all features' positions are determined, and the robot ceases its movement.

The main contributions of this paper are the following:

- We reformulate the SRAFL problem as a series of dynamic network localization tasks by utilizing sequential angle-based sensing. This innovative perspective, based on barycentric coordinate representation, provides fresh insights into vision-based localization problems.
- We develop a novel Barycentric Linear SRAFL (BL-SRAFL) method. By establishing and efficiently solving linear equations derived from local bearing angle measurements, BL-SRAFL not only guarantees global convergence of localization results but also significantly reduces computational complexity, making it particularly advantageous for resource-constrained applications.
- Recognizing that certain high-precision applications necessitate enhanced accuracy, we introduce the BL-SRAFL+BA pipeline. In this strategy, BL-SRAFL serves as an efficient linear front-end that provides globally convergent solutions for the individual localization subprocesses, which is complemented by an optional Bundle Adjustment (BA) back-end for refinement. This integrated pipeline substantially improves localization accuracy in noisy scenarios.
- We conduct numerical simulations and real-world experiments to demonstrate the effectiveness and robustness of the proposed approach. Experimental results confirm the advantages of BL-SRAFL and the enhanced accuracy achievable through BL-SRAFL+BA, highlighting their suitability for realistic robotic applications.

The rest of this paper is organized as follows. In Section II, the notations and preliminaries are provided. In Section III, we present the problem formulation. Section IV details the proposed method along with its theoretical analysis. Numerical simulations and real-world experimental validation are provided in Section V. Finally, Section VI concludes the paper.

II. Notations and Preliminaries

In this section, after introducing some notational conventions, we review important facts about graph theory and barycentric coordinate representation.

A. Notations

We denote by $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{R}^n$ the sets of real and natural numbers and the n-dimensional Euclidean space, respectively. The cardinality of a set is denoted by $|\cdot|$, and $\|\cdot\|$ is the 2-norm. For two sets $\mathcal{A}$ and $\mathcal{B}$, we denote the set subtraction operation by $\mathcal{A} \setminus \mathcal{B}$, which contains all elements belonging to $\mathcal{A}$ but not to $\mathcal{B}$. Symbols $\mathbf{1}_n$ and $\mathbf{0}_n$ are n-dimensional vectors of ones and zeros, respectively, and I_n is the $n \times n$ identity matrix. The rank of a matrix is denoted by $\text{rank}(\cdot)$, and $\otimes$ indicates the Kronecker product. For a matrix A, we denote by $[A]_{ij}$ its element in the i-th row and j-th column. Given a vector $\mathbf{v}$ indexed by labels and a set $\mathcal{V}$, $[\mathbf{v}]_{\mathcal{V}}$ denotes the subvector containing the entries whose indices belong to $\mathcal{V}$, preserving the original ordering.

B. Graph Theory

A *directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set $\mathcal{V}$ of *nodes* and a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of *edges*. A *configuration* in $\mathbb{R}^3$ of n nodes is defined by their coordinates in a Euclidean space, captured by the vector $\mathbf{p} = [\mathbf{p}_1^\top, \ldots, \mathbf{p}_n^\top]^\top \in \mathbb{R}^{3n}$, where $\mathbf{p}_i \in \mathbb{R}^3$ is the position of node i in the three-dimensional Euclidean space, for $i \in \{1, \ldots, n\}$. A configuration $\mathbf{p}$ is considered *generic* if its coordinates do not satisfy any algebraic equation with rational coefficients [34]. A *framework* in $\mathbb{R}^3$ consists of a graph $\mathcal{G}$ combined with a configuration $\mathbf{p}$, represented by the pair $(\mathcal{G}, \mathbf{p})$. Two frameworks sharing the same graph $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ are considered *similar*, represented as $(\mathcal{G}, \mathbf{p}) \sim (\mathcal{G}, \mathbf{q})$, if for all $i, j \in \mathcal{V} (i \neq j)$, it holds $\mathbf{p}_i - \mathbf{p}_j = sR(\mathbf{q}_i - \mathbf{q}_j)$, where $s > 0$ is a scaling factor and R is a rotation matrix, common for all $i, j \in \mathcal{V}$.

C. Barycentric Coordinate Representation

The *barycentric coordinate* system is a geometric concept that represents the position of a node relative to others [33]. Given node i and four other nodes j, k, l, and h, the three-dimensional coordinates $\mathbf{p}_i$ of node i can be written as a unit-sum linear combination of the coordinates of the other nodes:

$$\begin{cases} \mathbf{p}_i = a_{ij}\mathbf{p}_j + a_{ik}\mathbf{p}_k + a_{il}\mathbf{p}_l + a_{ih}\mathbf{p}_h, \\ a_{ij} + a_{ik} + a_{il} + a_{ih} = 1. \end{cases} \tag{1}$$

We refer to the four weights of such a linear combination $\{a_{ij}, a_{ik}, a_{il}, a_{ih}\}$ as the barycentric coordinates of node i relative to nodes j, k, l, and h. Consequently, these barycentric coordinates can be determined by solving the linear system described in (1). If the configuration $\mathbf{p}$ is generic, the linear system (1) admits a unique solution [33], which is given by

$$\begin{bmatrix} a_{ij} & a_{ik} & a_{il} & a_{ih} \end{bmatrix}^\top = \begin{bmatrix} \mathbf{p}_j & \mathbf{p}_k & \mathbf{p}_l & \mathbf{p}_h \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix}. \tag{2}$$

III. Problem Formulation

This section first clarifies the system setup of the SRAFL problem studied in this paper, followed by the description of the sensing model, the construction of the sensing graph, and the formal statement of the SRAFL objective.

A. Problem Setup

We consider a mobile robot equipped with a rigidly mounted monocular camera navigating a static three-dimensional environment containing sparse features with unknown positions. The proposed formulation applies to both indoor and outdoor settings, including warehouses, factories, laboratories, tunnels, underground facilities, and structured inspection sites, provided that the scene contains stationary features that can be reliably reidentified across consecutive views. Representative features include fiducial markers, reflective targets, and wall or ceiling corners in indoor environments, as well as traffic signs, poles, and building corners in outdoor environments. Each feature is modeled as a fixed point in three-dimensional space. As the robot moves, the camera acquires bearing angle measurements of the visible features from successive viewpoints. Our objective is to simultaneously estimate the robot trajectory and the positions of the observed features in a common metric coordinate frame using only these vision-based bearing measurements.

The robot acquires measurements over $n_t + 1 \in \mathbb{N}$ discrete time frames, indexed by the set $\mathcal{T} = \{0, 1, \ldots, n_t\}$. For each time frame $t \in \mathcal{T}$, we associate a unique robot position label r_t, collectively forming the set $\mathcal{R} = \{r_0, r_1, \ldots, r_{n_t}\}$. Without loss of generality, we align the global coordinate system Σ^g with the initial robot local coordinate system Σ^{r_0}, and assume that the robot and camera coordinate systems coincide, denoted uniformly as Σ^{r_t}. This simplification is valid when the camera is rigidly mounted on the robot without relative motion (e.g., a front-facing camera on a mobile robot). For any two coordinate systems Σ^a and Σ^b, we use $R_{ab} \in \text{SO}(3)$ to denote the rotation that maps vector coordinates from Σ^a to Σ^b. Consequently, at each time frame $t \in \mathcal{T}$, the robot occupies a distinct position r_t with coordinates $\mathbf{p}_{r_t} \in \mathbb{R}^3$ in Σ^g. At each position $r_t \in \mathcal{R}$, the robot observes a subset of static environmental features, represented by the instantaneous feature set $\mathcal{F}_{r_t}$. Each observed feature $f \in \mathcal{F}_{r_t}$ possesses global coordinates $\mathbf{p}_f \in \mathbb{R}^3$. Furthermore, the local coordinates of each feature $f \in \mathcal{F}_{r_t}$, expressed in the robot's local coordinate system Σ^{r_t}, are denoted by $\mathbf{p}_f^{r_t} \in \mathbb{R}^3$. These local coordinates form the basis of the angle-based sensing model introduced in the following subsection.

B. Sensing Model

Next, we formalize the corresponding angle-based sensing model, which defines the bearing angle measurements acquired by the robot for observed features at each time frame $t \in \mathcal{T}$.

Consider a feature $f \in \mathcal{F}_{r_t}$ observed at robot position $r_t \in \mathcal{R}$. Since the robot is located at the origin of its local frame, i.e., $\mathbf{p}_{r_t}^{r_t} = \mathbf{0}_3$, the bearing angle measurement $\mathbf{b}_{r_t f}^{r_t}$ directed from the robot toward the feature f is defined as:

$$\mathbf{b}_{r_t f}^{r_t} = \frac{\mathbf{p}_f^{r_t} - \mathbf{p}_{r_t}^{r_t}}{\|\mathbf{p}_f^{r_t} - \mathbf{p}_{r_t}^{r_t}\|} = \frac{\mathbf{p}_f^{r_t}}{\|\mathbf{p}_f^{r_t}\|}. \tag{3}$$

Here, the superscript r_t explicitly indicates that the vector is expressed in the robot's local coordinate system Σ^{r_t}, and the subscript $r_t f$ specifies the measurement direction from the robot at position r_t toward feature f (see Fig. 1). Throughout

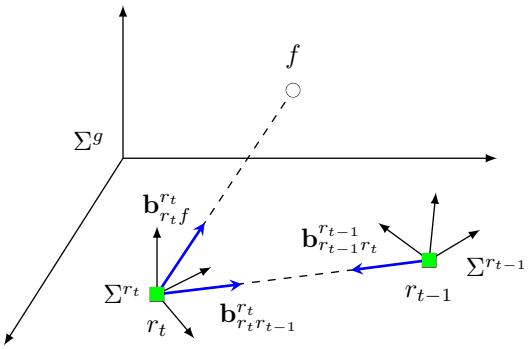


Fig. 1. An illustration of angle-based sensing. The green square represents the monocular camera, while the circle indicates features. Blue lines with arrows depict bearing angle measurements, black lines with arrow denote coordinate axes, and the dashed line serves as an auxiliary reference.

this paper, the term bearing angle measurement refers to this normalized direction vector. This angle-based sensing model directly establishes the geometric relationship between robot positions and observed features, providing critical constraints for solving the SRAFL problem.

C. Sensing Graph

To explicitly characterize the sensing interactions between the robot and the environmental features observed simultaneously at consecutive time frames $t-1$ and t, we now introduce a dynamic directed *sensing graph* $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$.

The node set $\mathcal{V}(t)$ of the sensing graph $\mathcal{G}(t)$ comprises robot positions and commonly observed features at consecutive time frames, formally defined as $\mathcal{V}(t) = \mathcal{V}_r(t) \cup \mathcal{V}_f(t)$. Specifically, the set $\mathcal{V}_r(t) = \{r_{t-1}, r_t\}$ denotes the consecutive robot positions, with combined coordinates $\mathbf{p}_r(t) = [\mathbf{p}_{r_{t-1}}^\top, \mathbf{p}_{r_t}^\top]^\top \in \mathbb{R}^6$. The corresponding feature set $\mathcal{V}_f(t)$ consists of features simultaneously observed from both robot positions and is explicitly defined as the intersection $\mathcal{V}_f(t) = \mathcal{F}_{r_{t-1}} \cap \mathcal{F}_{r_t}$. It is further partitioned into localized and unlocalized subsets, denoted by $\mathcal{V}_l(t)$ and $\mathcal{V}_u(t)$. Specifically, the localized subset $\mathcal{V}_l(t)$ includes features whose global positions are already known at both consecutive robot positions, while the unlocalized subset $\mathcal{V}_u(t)$ consists of features whose global coordinates remain unknown and must be estimated. The global coordinates corresponding to these two subsets are respectively stacked into $\mathbf{p}_l(t) \in \mathbb{R}^{3n_l(t)}$ and $\mathbf{p}_u(t) \in \mathbb{R}^{3n_u(t)}$, with the numbers of localized and unlocalized features denoted by $n_l(t) = |\mathcal{V}_l(t)|$ and $n_u(t) = |\mathcal{V}_u(t)|$. To facilitate subsequent analysis, we further define the combined set of unlocalized nodes as $\mathcal{V}_z(t) = \mathcal{V}_r(t) \cup \mathcal{V}_u(t)$, where $n_z(t) = |\mathcal{V}_z(t)| = 2 + |\mathcal{V}_u(t)|$ represents the total number of unlocalized features and two robot positions in $\mathcal{G}(t)$. Accordingly, their stacked global coordinates are denoted by $\mathbf{p}_z(t) = [\mathbf{p}_u(t)^\top, \mathbf{p}_r(t)^\top]^\top \in \mathbb{R}^{3n_z(t)}$. Notably, the global coordinates of all nodes within the set $\mathcal{V}_z(t)$ require estimation.

An edge $(f, r_t) \in \mathcal{E}(t)$ (and analogously (f, r_{t-1})) is established in $\mathcal{G}(t)$ if and only if (iff) the feature f is observed from both the robot position r_{t-1} and r_t, acquiring the corresponding bearing angle measurements $\mathbf{b}_{r_{t-1} f}^{r_{t-1}}$ and $\mathbf{b}_{r_t f}^{r_t}$ defined in (3). Additionally, the relative movement between robot positions r_{t-1} and r_t yields the bearing angle measurements $\mathbf{b}_{r_t r_{t-1}}^{r_t}$ and $\mathbf{b}_{r_{t-1} r_t}^{r_{t-1}}$, justifying the inclusion of directed edges $(r_t, r_{t-1}), (r_{t-1}, r_t) \in \mathcal{E}(t)$. It is important to note that the sensing graph $\mathcal{G}(t)$ dynamically evolves as the robot moves over time. Consequently, the sets $\mathcal{V}_r(t)$ and $\mathcal{V}_f(t)$, their localized and unlocalized subsets, associated coordinates, and the edge set $\mathcal{E}(t)$ are continually updated as new measurements become available.

We define the cumulative sensing graph collection up to the time frame t as $\mathcal{G}^{\text{cum}}(t) = \{\mathcal{G}(\tau)\}_{\tau \le t}$. We introduce the cumulative node sets as follows: the cumulative robot position set $\mathcal{V}_r^{\text{cum}}(t) = \{r_0, \ldots, r_t\}$, the cumulative feature set $\mathcal{V}_f^{\text{cum}}(t) = \bigcup_{\tau \le t} \mathcal{V}_f(\tau)$, the cumulative localized feature set $\mathcal{V}_l^{\text{cum}}(t) = \bigcup_{\tau \le t} \mathcal{V}_l(\tau)$, and the cumulative unlocalized feature set $\mathcal{V}_u^{\text{cum}}(t) = \bigcup_{\tau \le t} \mathcal{V}_u(\tau)$. Their stacked coordinates $\mathbf{p}_r^{\text{cum}}(t)$, $\mathbf{p}_f^{\text{cum}}(t)$, $\mathbf{p}_l^{\text{cum}}(t)$, and $\mathbf{p}_u^{\text{cum}}(t)$ are constructed by concatenating the coordinates of unique elements from these cumulative sets in ascending label order. By definition, these cumulative sets contain distinct labels without repetition, ensuring consistent indexing and fixed dimensions.

D. Problem Statement

Estimating the positions of the robot and unlocalized features using relative bearing angle measurements fundamentally relies on the observability provided by previously localized features. Specifically, to ensure adequate geometric constraints for localization, maintaining visibility of a minimal subset of these previously localized features across sensing graphs is essential. Thus, we introduce the following assumption.

Assumption 1: At each time frame $t \ge 1$, the sensing graph $\mathcal{G}(t)$ contains at least four features previously observed in the cumulative sensing graph collection $\mathcal{G}^{\text{cum}}(t-1)$. That is, the feature set $\mathcal{V}_f(t)$ must include no fewer than four features that were observed in some earlier time frame $\tau < t$.

Remark 1: Assumption 1 ensures that, at every time frame $t \ge 1$, the intersection between currently observed features and the cumulative set of previously localized features satisfies the minimal cardinality condition $|\mathcal{V}_f(t) \cap \mathcal{V}_l^{\text{cum}}(t-1)| \ge 4$, preserving the essential localizability constraints established in [35]. Specifically, these constraints provide the minimum number of spatial references required to uniquely determine the robot pose and the positions of unlocalized features in a generic three-dimensional configuration. Thus, Assumption 1 is a structural identifiability condition for the proposed method rather than a guarantee of localization accuracy under measurement noise. In practice, this requirement can be facilitated by appropriate measurement intervals and, when available, robot motion planning or control. As localization proceeds, the cumulative set of localized features generally expands, increasing the availability of spatial references. However, limited fields of view, rapid rotations, or feature occlusions may temporarily violate this condition. The current update can then be postponed until sufficient valid correspondences are recovered, while external relocalization or auxiliary motion estimation may be used if continuous localization is required.

With the above assumption in place, the SRAFL problem can now be formally stated as follows.

Problem 1: Given each sensing graph $\mathcal{G}(t)$ within the cumulative collection $\mathcal{G}^{\mathrm{cum}}(n_t)$ and the associated local bearing angle measurements $\mathbf{b}_{r_t f}^{r_t}$, simultaneously estimate the robot trajectory and the positions of unlocalized features. Specifically, gradually estimate the cumulative robot position coordinates $\mathbf{p}_r^{\mathrm{cum}}(n_t)$ and cumulative unlocalized feature coordinates $\mathbf{p}_u^{\mathrm{cum}}(n_t)$ in the global coordinate system Σ^g.

The joint estimation objective can be expressed in terms of bearing residuals. For a candidate robot position $\mathbf{p}_{r_t}$, robot orientation $R_{r_0 r_t}$, and feature position $\mathbf{p}_f$, the predicted bearing in Σ^{r_t} is

$$\hat{\mathbf{b}}_{r_t f}^{r_t} = \frac{R_{r_0 r_t}\left(\mathbf{p}_f - \mathbf{p}_{r_t}\right)}{\left\|\mathbf{p}_f - \mathbf{p}_{r_t}\right\|}. \tag{4}$$

Accordingly, the SRAFL objective is formulated as

$$\min_{\substack{\{\mathbf{p}_{r_t}, R_{r_0 r_t}\}_{t\in\mathcal{T}},\\ \{\mathbf{p}_f\}_{f\in\mathcal{V}_u^{\mathrm{cum}}(n_t)}}} \sum_{t\in\mathcal{T}} \sum_{f\in\mathcal{F}_{r_t}\cap\mathcal{V}_f^{\mathrm{cum}}(n_t)} \left\| \mathbf{b}_{r_t f}^{r_t} - \hat{\mathbf{b}}_{r_t f}^{r_t} \right\|^2, \tag{5}$$

where the positions of previously localized features are treated as fixed spatial references.

IV. Barycentric Linear SRAFL Methodology

To address the SRAFL problem, we propose an incremental localization strategy that estimates robot positions and previously unlocalized feature coordinates through discrete, computationally lightweight subprocesses. Each subprocess employs a linear formulation based on barycentric coordinate representation, enabling efficient computation of robot and feature positions. Integrating these subprocesses across the entire trajectory yields a globally consistent localization solution.

First, we showcase the proposed SRAFL approach. Second, we detail the barycentric linear formulation and the efficient conjugate gradient (CG) solver utilized within each subprocess. Third, we present the comprehensive BL-SRAFL+BA implementation pipeline, integrating these subprocesses into a cohesive localization framework, with an optional BA refinement step to further enhance localization accuracy. Fourth, we analyze the computational complexity and memory footprint of our approach, establishing its theoretical scaling for online estimation on resource-constrained platforms.

A. Method Overview

This subsection introduces the general structure of our proposed SRAFL approach and briefly outlines its key components. The localization process begins with an initialization procedure to determine four localized features, whose global coordinates serve as initial spatial references. Using these localized features, the robot iteratively localizes itself and other unknown features as it traverses the environment.

During navigation, the robot acquires bearing angle measurements of environmental features from successive positions, identifying overlapping observed features between viewpoints. At each step, these bearing angle measurements are transformed into a linear system using barycentric coordinate representation. This linear system is efficiently solved via a CG method, rapidly yielding accurate estimates for the previously unknown features as well as updating the robot's own position. Newly estimated features are incrementally added to the localized feature set, thereby improving the localized feature distribution for subsequent localization steps. This iterative process continues until the robot completes its movement.

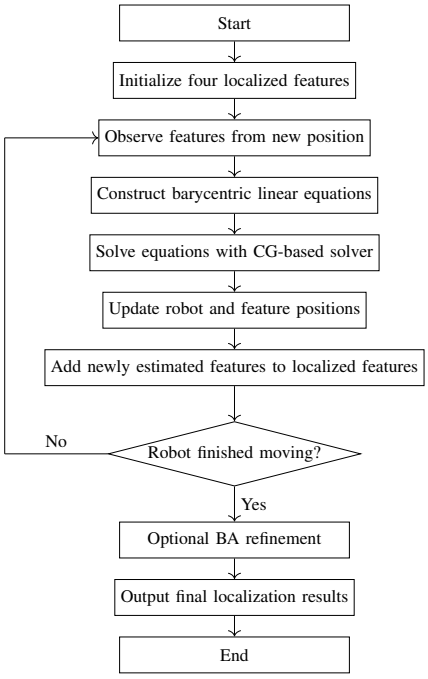


Fig. 2. Workflow of the proposed SRAFL approach. The lightweight BL-SRAFL provides online estimates, while an optional BA back-end refines the solution offline when higher accuracy is required.

Optionally, after completing the linear localization procedure, a nonlinear BA refinement step may be performed to minimize residual reprojection errors, further improving the localization precision, especially in the presence of noise. Integrating the BA step yields BL-SRAFL+BA, in which BL-SRAFL provides efficient and globally consistent initial estimates, while BA serves as a back-end refinement module.

Fig. 2 summarizes our approach, illustrating the method's progression from linear localization to the optional BA refinement step. Our proposed approach contributes to addressing the SRAFL problem in three significant ways: firstly, it introduces a linear barycentric formulation for computational efficiency; secondly, it employs an efficient CG-based solver guaranteeing rapid global convergence; third, an optional BA refinement step integrated into a cohesive BL-SRAFL+BA pipeline, providing enhanced accuracy in practical scenarios. Each component will be discussed in the following.

B. Barycentric Linear Formulation

Here, we detail the conversion of bearing angle measurements into a linear equation system using barycentric coordinates, laying the foundation for efficient solution algorithms.

1) Bearing Angle Measurements Acquisition: Within the sensing graph $\mathcal{G}(t)$, the bearing angle measurements $\mathbf{b}_{r_t f}^{r_t}$ defined in (3), are obtained directly from pixel observations captured by the monocular camera. Let $\mathbf{x}_f^{r_t} = [u_f^{r_t}, v_f^{r_t}, 1]^\top$

represent the homogeneous pixel coordinates of feature f observed in the image acquired at r_t. Given a calibrated intrinsic camera matrix $K \in \mathbb{R}^{3\times3}$, each bearing angle measurement is computed as

$$\mathbf{b}_{r_t f}^{r_t} = \frac{K^{-1}\mathbf{x}_f^{r_t}}{\|K^{-1}\mathbf{x}_f^{r_t}\|}. \tag{6}$$

Similarly, bearing angle measurements at r_{t-1} are obtained using the corresponding pixel coordinates of its image.

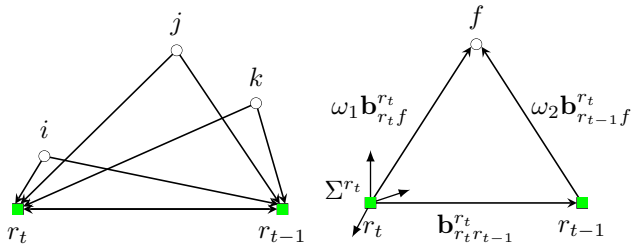
To determine the bearing angle measurement $\mathbf{b}_{r_t r_{t-1}}^{r_t}$ and the relative rotation matrix $R_{r_{t-1}r_t}$, which maps vectors from $\Sigma^{r_{t-1}}$ to Σ^{r_t}, we utilize a set of previously localized features $\mathcal{V}_l(t) \subset \mathcal{V}_l^{\text{cum}}(t-1)$. This rotation is necessary because bearing vectors observed at different times are expressed in different camera frames; they must be represented in a common frame before the barycentric relations can be constructed. Specifically, we first compute the local coordinates $\mathbf{p}_l^{r_{t-1}}$ of each localized feature $l \in \mathcal{V}_l(t)$, expressed in the robot's local coordinate system $\Sigma^{r_{t-1}}$. Given the global coordinates $\mathbf{p}_l$ and the robot's global position $\mathbf{p}_{r_{t-1}}$, both previously estimated by the BL-SRAFL at time frame $t-1$, the local coordinates are computed through $\mathbf{p}_l^{r_{t-1}} = R_{r_0 r_{t-1}}(\mathbf{p}_l - \mathbf{p}_{r_{t-1}})$. Here, the global-to-local rotation matrix $R_{r_0 r_{t-1}}$ of the robot at time frame $t-1$ is recursively updated from previously estimated relative rotation matrices as follows: $R_{r_0 r_{t-1}} = R_{r_{t-2}r_{t-1}} R_{r_0 r_{t-2}}$, with $R_{r_{t-2}r_{t-1}}$ representing the relative rotation from $\Sigma^{r_{t-2}}$ to $\Sigma^{r_{t-1}}$. Utilizing these local coordinates $\mathbf{p}_l^{r_{t-1}}$ together with the corresponding homogeneous pixel coordinates $\mathbf{x}_l^{r_t}$ directly observed from Σ^{r_t}, the estimation of the relative rotation matrix $R_{r_{t-1}r_t}$ and translation matrix $T_{r_t r_{t-1}}^{r_t}$ can thus be formulated as a Perspective-n-Point (PnP) optimization problem:

$$\min_{R_{r_{t-1}r_t}, T_{r_t r_{t-1}}^{r_t}} \sum_{l\in\mathcal{V}_l(t)} \left\| \mathbf{x}_l^{r_t} - \pi\Big(K[R_{r_{t-1}r_t} | T_{r_t r_{t-1}}^{r_t}] \begin{bmatrix} \mathbf{p}_l^{r_{t-1}} \\ 1 \end{bmatrix}\Big) \right\|^2, \tag{7}$$

where $\pi(\cdot)$ denotes the perspective projection operation. Under the adopted coordinate convention, the rigid transformation satisfies $\mathbf{p}_l^{r_t} = R_{r_{t-1}r_t}\mathbf{p}_l^{r_{t-1}} + T_{r_t r_{t-1}}^{r_t}$. Hence, the 3×1 translation matrix $T_{r_t r_{t-1}}^{r_t}$ represents the displacement from the current camera center r_t to the previous camera center r_{t-1}, expressed in Σ^{r_t}. To efficiently solve (7), we employ the Efficient Perspective-n-Point (EPnP) algorithm [36], which parametrizes each three-dimensional point with four virtual control points. This reformulation transforms the original nonlinear optimization into a linear system solvable by Singular Value Decomposition (SVD), followed by a Gauss-Newton iteration. Notably, the EPnP algorithm inherently requires a minimum of four localized feature correspondences ($|\mathcal{V}_l(t)| \geq 4$) to ensure a unique and robust solution, a condition explicitly guaranteed by Assumption 1. Therefore, normalizing the resultant displacement removes its magnitude and yields exactly the inter-frame bearing direction illustrated in Fig. 1:

$$\mathbf{b}_{r_t r_{t-1}}^{r_t} = \frac{T_{r_t r_{t-1}}^{r_t}}{\|T_{r_t r_{t-1}}^{r_t}\|}. \tag{8}$$

These rigorously derived normalized vectors explicitly relate the raw camera measurements to geometric constraints, constituting essential input data for subsequent computational steps within the BL-SRAFL.



(a) Sensing subgraph $\mathcal{G}_5^{(c)}$ (b) Vector relationships

Fig. 3. Illustrations of the barycentric linear formulation.

2) Barycentric Coordinates Computation: With the bearing angle measurements obtained, we proceed to compute the barycentric coordinates within the sensing graph $\mathcal{G}(t)$, which enables the formulation of linear constraints among the robot and feature positions.

As described in Subsection III-C, the sensing graph $\mathcal{G}(t)$ comprises two consecutive robot positions r_t and r_{t-1}, together with a set of simultaneously observed features $\mathcal{V}_f(t)$. To facilitate the calculation of barycentric coordinates for all unlocalized nodes in $\mathcal{V}_z(t)$, we partition the entire sensing graph $\mathcal{G}(t)$ into smaller subsets termed *sensing subgraphs*. Given that edges within $\mathcal{G}(t)$ exist exclusively between robot positions and features (without any direct edges among features themselves), each sensing subgraph must contain at least five nodes to ensure valid barycentric coordinate computations. Hence, we define each sensing subgraph $\mathcal{G}_5^{(c)} = (\mathcal{V}_5^{(c)}, \mathcal{E}_5^{(c)}) \in \mathcal{G}(t)$ with exactly five nodes: the two consecutive robot positions r_t and r_{t-1} and any common features $\{i,j,k\} \in \mathcal{V}_f(t)$ (see Fig. 3a). The superscript (c) denotes the index of each specific sensing subgraph. Hence, we first compute barycentric coordinates individually within each sensing subgraph and then merge these to form a comprehensive representation.

To determine barycentric coordinates for each sensing subgraph $\mathcal{G}_5^{(c)}$, we construct an auxiliary framework $(\mathcal{G}_5^{(c)}, \mathbf{q}_5)$, which is similar to the actual spatial framework $(\mathcal{G}_5^{(c)}, \mathbf{p}_5)$. Due to this similarity, the barycentric coordinates of both frameworks coincide. Thus, computing the barycentric coordinates of the auxiliary coordinates $\mathbf{q}_5$ directly yields those of the actual coordinates $\mathbf{p}_5$. Specifically, $\mathbf{p}_5 = [\mathbf{p}_{r_t}^\top, \mathbf{p}_{r_{t-1}}^\top, \mathbf{p}_i^\top, \mathbf{p}_j^\top, \mathbf{p}_k^\top]^\top$ and $\mathbf{q}_5$ is structured as $\mathbf{q}_5 = [\mathbf{q}_{r_t}^\top, \mathbf{q}_{r_{t-1}}^\top, \mathbf{q}_i^\top, \mathbf{q}_j^\top, \mathbf{q}_k^\top]^\top$. We begin by setting $\mathbf{q}_{r_t} = \mathbf{0}_3$ and $\mathbf{q}_{r_{t-1}} = \mathbf{b}_{r_t r_{t-1}}^{r_t}$. For each feature $f \in \{i,j,k\}$, we then transform the bearing angle measurement observed at r_{t-1} from its original coordinate system $\Sigma^{r_{t-1}}$ into Σ^{r_t} via the rotation matrix: $\mathbf{b}_{r_{t-1}f}^{r_t} = R_{r_{t-1}r_t}\mathbf{b}_{r_{t-1}f}^{r_{t-1}}$. Due to the coplanarity of vectors $\mathbf{b}_{r_t r_{t-1}}^{r_t}, \mathbf{b}_{r_{t-1}f}^{r_t}$, and $\mathbf{b}_{r_t f}^{r_t}$ (see Fig. 3b), there exist positive scalar parameters ω_1, ω_2 satisfying $\omega_1\mathbf{b}_{r_t f}^{r_t} = \mathbf{b}_{r_t r_{t-1}}^{r_t} + \omega_2\mathbf{b}_{r_{t-1}f}^{r_t}$, which presents an overconstrained equation; hence the least squares formula $\min_{\omega_1,\omega_2>0} \|\omega_1\mathbf{b}_{r_t f}^{r_t} - (\mathbf{b}_{r_t r_{t-1}}^{r_t} + \omega_2\mathbf{b}_{r_{t-1}f}^{r_t})\|^2$ is used to compute ω_1 and ω_2. The auxiliary coordinates $\mathbf{q}_f$ are subsequently calculated as either $\mathbf{q}_f = \omega_1\mathbf{b}_{r_t f}^{r_t}$ or $\mathbf{q}_f = \mathbf{b}_{r_t r_{t-1}}^{r_t} + \omega_2\mathbf{b}_{r_{t-1}f}^{r_t}$.

Repeating this procedure for all features $f \in \{i, j, k\}$, we finally determine $\mathbf{q}_5$. Consequently, with $\mathbf{q}_5$, the barycentric coordinates $a_{xy}^{(c)}$ for each unlocalized node $x \in \mathcal{V}_5^{(c)}$ are specifically determined using (2), where each y ranges over all nodes in $\mathcal{V}_5^{(c)} \setminus \{x\}$. Since $(\mathcal{G}_5^{(c)}, \mathbf{p}_5) \sim (\mathcal{G}_5^{(c)}, \mathbf{q}_5)$, these barycentric coordinates directly apply to the actual positions, resulting in the linear constraint $\mathbf{p}_x = \sum_{y \in \mathcal{V}_5^{(c)} \setminus \{x\}} a_{xy}^{(c)} \mathbf{p}_y$. In practice, nearly degenerate sensing subgraphs are excluded during implementation to improve numerical robustness.

In general, each unlocalized node $x \in \mathcal{V}_z(t)$ may be involved in multiple sensing subgraphs. Let c_x denote the total number of sensing subgraphs containing node x. The corresponding barycentric coordinates are aggregated using nonnegative weights $\{\gamma_c\}$ satisfying $\sum_{c=1}^{c_x} \gamma_c = 1$. Unless otherwise specified, uniform weights, i.e., $\gamma_c = 1/c_x$, are adopted throughout this paper. Accordingly, the weighted barycentric coordinates a_{xy} are obtained by aggregating the barycentric coordinates from all involved sensing subgraphs, i.e., $a_{xy} = \sum_{c=1}^{c_x} \gamma_c a_{xy}^{(c)}$. Consequently, the final weighted barycentric coordinate representation for each unlocalized node $x \in \mathcal{V}_z(t)$ is thus expressed as:

$$\mathbf{p}_x = \sum\nolimits_{y \in \bigcup_c \mathcal{V}_5^{(c)}} a_{xy} \mathbf{p}_y, \tag{9}$$

where the summation includes all nodes $y \in \mathcal{V}(t)$ sharing at least one sensing subgraph with unlocalized node x.

3) Linear System Assembly: After obtaining the weighted barycentric coordinate representation for r_t and r_{t-1}, and all unlocalized features $u \in \mathcal{V}_u(t)$, we write the linear equation constraints (9) into a matrix representation, yielding the following set of equations:

$$\begin{bmatrix} \mathbf{p}_l(t) \\ \mathbf{p}_u(t) \\ \mathbf{p}_r(t) \end{bmatrix} = (B_0(t) \otimes I_3) \begin{bmatrix} \mathbf{p}_l(t) \\ \mathbf{p}_u(t) \\ \mathbf{p}_r(t) \end{bmatrix}, \tag{10}$$

which are reformulated as

$$\begin{bmatrix} \mathbf{p}_l(t) \\ \mathbf{p}_z(t) \end{bmatrix} = (B_0(t) \otimes I_3) \begin{bmatrix} \mathbf{p}_l(t) \\ \mathbf{p}_z(t) \end{bmatrix}, \tag{11}$$

where the matrix $B_0(t)$ is defined as

$$[B_0(t)]_{ij} = \begin{cases} 1, & \text{if } i \in \mathcal{V}_l(t) \text{ and } i = j, \\ 0, & \text{if } i \in \mathcal{V}_l(t) \text{ and } i \neq j, \\ a_{ij}, & \text{if } i \in \mathcal{V}_z(t). \end{cases} \tag{12}$$

To further simplify the linear system, we define $B(t) = I_{n(t)} - B_0(t)$, where $n(t) = n_l(t) + n_z(t)$, and partition it based on $\mathcal{V}_l(t)$ and $\mathcal{V}_z(t)$, it takes the following form:

$$B(t) = \begin{bmatrix} \mathbf{0}_{n_l(t) \times n_l(t)} & \mathbf{0}_{n_l(t)} \times n_z(t) \\ B_l(t) & B_z(t) \end{bmatrix}, \tag{13}$$

Combining (11) and (13), we get

$$(B_z(t) \otimes I_3)\, \mathbf{p}_z(t) = -\,(B_l(t) \otimes I_3)\, \mathbf{p}_l(t), \tag{14}$$

where we refer to $B_z(t)$ and $B_l(t)$ as the BL-SRAFL matrices, which serves as the essential input for the CG-based solver.

C. Efficient Conjugate Gradient Solver

Building upon the linear formulation derived above, we introduce an efficient CG-based solver tailored for sparse linear systems arising from barycentric coordinate representation, ensuring global convergence. We first discuss the solvability conditions, then detail the CG-based BL-SRAFL solver, and finally analyze its robustness to measurement noise.

1) Solvability Guarantees: We begin by establishing theoretical conditions for solvability based on the linear system in (14). Two lemmas provide the necessary framework.

Lemma 1: The positions of the robot $r_t \in \mathcal{V}_r(t)$ and unlocalized features $u \in \mathcal{V}_u(t)$ are uniquely solvable iff matrix $B_z(t)$ is nonsingular, i.e., $\text{rank}(B_z(t)) = n_z(t)$.

Proof: The solvability of (14) for $\mathbf{p}_z(t)$ directly hinges on the invertibility of the matrix $B_z(t)$. Therefore, achieving unique positions necessitates that $B_z(t)$ be of full rank, as established by standard linear algebra principles [37]. ■

Lemma 2: Under Assumption 1, the BL-SRAFL matrix $B_z(t)$ is always nonsingular, i.e., $\text{rank}(B_z(t)) = n_z(t)$.

Proof: Based on the barycentric coordinate representation detailed in Subsection IV-B2, we define the barycentric coordinate induced graph $\mathcal{G}(t)'$ associated with the matrix $B_0(t)$, such that an edge (i, j) exists within $\mathcal{G}(t)'$ if and only if the corresponding entry $[B_0(t)]_{ij}$ is nonzero. According to (12), each nonzero entry of $B_0(t)$ for $i \in \mathcal{V}_z(t)$ is equal to the weighted barycentric coordinates a_{ij}, obtained by aggregating the barycentric coordinates $a_{ij}^{(c)}$ derived from multiple sensing subgraphs. Hence, to illustrate the construction of $\mathcal{G}(t)'$, we consider an arbitrary sensing subgraph $\mathcal{G}_5^{(c)} \in \mathcal{G}(t)$ consisting of two robot positions r_t and r_{t-1}, and an arbitrary triple of observed features $\{i, j, k\}$. Within such a subgraph, the barycentric coordinates of an unlocalized feature (e.g., feature i) are given by $\mathbf{p}_i = a_{ir_t}^{(c)} \mathbf{p}_{r_t} + a_{ir_{t-1}}^{(c)} \mathbf{p}_{r_{t-1}} + a_{ij} \mathbf{p}_j + a_{ik}^{(c)} \mathbf{p}_k$, indicating the presence of edges connecting i with r_t, r_{t-1}, j, and k. These edges form a barycentric coordinate induced subgraph, denoted $\mathcal{G}_5'$. The complete barycentric coordinate induced graph $\mathcal{G}(t)'$ is then formed as the union of all such induced subgraphs across every possible combination of features (over all c). Given Assumption 1, at least four previously localized features are guaranteed to exist within $\mathcal{V}_f(t)$. Consequently, every unlocalized node in $\mathcal{V}_z(t)$ (including robot positions and unlocalized features) is connected to these previously localized features within $\mathcal{G}(t)'$. Such connectivity ensures the existence of at least four vertex-disjoint paths linking the known localized feature set $\mathcal{V}_l(t)$ to each unlocalized node in $\mathcal{V}_z(t)$. This condition, in line with the Lemma 4.1 from [38], assures that $\text{rank}(B_z(t)) = n_z(t)$, hence is nonsingular. ■

Lemmas 1 and 2 immediately give the following result.

Theorem 1: The robot trajectory and the unlocalized feature positions at time frame t are uniquely determinable under Assumption 1.

Proof: We rephrase the claim as follows: The positions of robot $r_t \in \mathcal{V}_r(t)$ and unlocalized features $u \in \mathcal{V}_u(t)$ are achievable iff there are at least four localized features in $\mathcal{G}(t)$. The proof is a straightforward consequence of Lemmas 1 and 2. ■

2) CG-based Solver: Based on Theorem 1, we present an efficient solver based on the CG method. First, define $\bar{B}_l(t) = B_l(t) \otimes I_3$ and $\bar{B}_z(t) = B_z(t) \otimes I_3$. By multiplying the transpose matrix of $\bar{B}_z(t)$ in both sides of (14), we obtain

$$\bar{B}_z(t)^\top \bar{B}_z(t)\mathbf{p}_z(t) = -\bar{B}_z(t)^\top \bar{B}_l(t)\mathbf{p}_l(t). \tag{15}$$

Since $B_z(t)$ is nonsingular (Lemma 2), $\bar{B}_z(t)^\top \bar{B}_z(t)$ is symmetric positive definite (SPD), enabling CG-based solutions.

The detailed implementation of the CG-based BL-SRAFL solver is summarized in Algorithm 1. Given an arbitrary initial estimate of $\hat{\mathbf{p}}_z(t)[0]$ and a predefined small tolerance threshold $\varepsilon > 0$, this algorithm converges to the global solution, avoiding local minima issues typically encountered in nonlinear approaches. Specifically, during the iteration phase, the scalar step size $\alpha[k]$ is determined to ensure orthogonality of residuals $\mathbf{r}[k+1]$, satisfying $\mathbf{r}[k]^\top \mathbf{r}[k+1] = 0$. In addition, step size $\beta[k]$ ensures conjugacy of direction vectors $\mathbf{d}[k+1]$, fulfilling $\mathbf{d}[k]^\top \bar{B}_z(t)^\top \bar{B}_z(t)\mathbf{d}[k+1] = 0$. These crucial properties guarantee fast and stable convergence to the exact solution within a finite number of iterations, typically significantly fewer than $n_z(t)$ steps [39]. We formally summarize these guarantees in the following theorem.

Algorithm 1 CG-based BL-SRAFL Solver

Input: BL-SRAFL matrices $B_z(t)$ and $B_l(t)$, localized feature positions $\mathbf{p}_l(t)$, and small tolerance threshold $\varepsilon > 0$
Output: Estimated robot positions $\mathbf{p}_r(t)$ and feature positions $\mathbf{p}_u(t)$

Set matrix: $\bar{B}_z(t) \leftarrow B_z(t) \otimes I_3$.
Set matrix: $\bar{B}_l(t) \leftarrow B_l(t) \otimes I_3$.
Compute initial residual: $\mathbf{r}[0] \leftarrow -\bar{B}_z(t)^\top \bar{B}_l(t)\mathbf{p}_l(t) - \bar{B}_z(t)^\top \bar{B}_z(t)\hat{\mathbf{p}}_z(t)[0]$.
Set initial direction: $\mathbf{d}[0] \leftarrow \mathbf{r}[0]$.
Initialize iteration count: $k \leftarrow 0$.
repeat
  Compute step size: $\alpha[k] \leftarrow \frac{\mathbf{r}[k]^\top \mathbf{r}[k]}{\mathbf{d}[k]^\top \bar{B}_z(t)^\top \bar{B}_z(t)\mathbf{d}[k]}$.
  Update estimate: $\hat{\mathbf{p}}_z(t)[k+1] \leftarrow \hat{\mathbf{p}}_z(t)[k] + \alpha[k]\mathbf{d}[k]$.
  Update residual: $\mathbf{r}[k+1] \leftarrow \mathbf{r}[k] - \alpha[k]\bar{B}_z(t)^\top \bar{B}_z(t)\mathbf{d}[k]$.
  Compute step size: $\beta[k] \leftarrow \frac{\mathbf{r}[k+1]^\top \mathbf{r}[k+1]}{\mathbf{r}^\top[k]\mathbf{r}[k]}$.
  Update direction: $\mathbf{d}[k+1] \leftarrow \mathbf{r}[k+1] + \beta[k]\mathbf{d}[k]$.
  Increase iteration count: $k \leftarrow k+1$.
until $|\mathbf{r}[k+1]| < \varepsilon$
Extract solution: $[\mathbf{p}_u(t)^\top, \mathbf{p}_r(t)^\top\ ]^\top \leftarrow \hat{\mathbf{p}}_z(t)[k+1]$.
return $\mathbf{p}_r(t)$ and $\mathbf{p}_u(t)$

Theorem 2: Under a noise-free scenario, the robot and feature positions $\hat{\mathbf{p}}_r(t)$ and $\hat{\mathbf{p}}_u(t)$ estimated with Algorithm 1 converge to their true positions $\mathbf{p}_r(t)$ and $\mathbf{p}_u(t)$, uniquely determining the robot trajectory and feature positions in $\mathcal{G}(t)$.

Proof: Since residuals $\mathbf{r}$ are mutually orthogonal and direction vectors $\mathbf{d}$ remain mutually conjugate, the iterates $\hat{\mathbf{p}}_z(t)[k]$ effectively represent the projection of the true solution $\mathbf{p}_z(t)$ onto the Krylov subspace. Thus, under noise-free conditions, convergence to the exact solution occurs within a finite number of iterations, as demonstrated in [39]. ∎

3) Robustness Analysis: We investigate the impact of noise measurement perturbations on the solver's performance and establish sufficient conditions ensuring global convergence while providing an approximate optimal solution.

In the presence of noise, the BL-SRAFL matrices $B_z(t)$ and $B_l(t)$ are perturbed to $\hat{B}_z(t)$ and $\hat{B}_l(t)$, respectively. Accordingly, $\bar{B}_z(t)$ and $\bar{B}_l(t)$ are perturbed to $\hat{\bar{B}}_z(t) = \hat{B}_z(t) \otimes I_3$ and $\hat{\bar{B}}_l(t) = \hat{B}_l(t) \otimes I_3$, transforming (15) into:

$$\hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_z(t)\mathbf{p}_z(t) = -\hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_l(t)\mathbf{p}_l(t). \tag{16}$$

Provided that the perturbed matrix $\hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_z(t)$ remains full rank, Algorithm 1 solves the least squares problem

$$\min_{\mathbf{p}_z(t)} \|\hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_z(t)\mathbf{p}_z(t) + \hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_l(t)\mathbf{p}_l(t)\|^2, \tag{17}$$

whose solution is explicitly given by:

$$\mathbf{p}_z(t)^* = -\big(\hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_z(t)\big)^{-1} \hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_l(t)\mathbf{p}_l(t). \tag{18}$$

This solution is the unique least-squares solution of the perturbed linear system.

Next, we define the perturbation matrix as $\Delta B_z(t) = \hat{B}_z(t) - B_z(t)$, and derive a spectral norm bound to guarantee algorithm convergence.

Theorem 3: Algorithm 1 converges to the least squares solution (18), if the perturbation $\|\Delta B_z(t)\| < \sigma_{\min}(B_z(t))$, where $\sigma_{\min}(B_z(t))$ is the smallest singular value of $B_z(t)$.

Proof: According to Mirsky's theorem [40], we have that $|\sigma_{\min}(\hat{B}_z(t)) - \sigma_{\min}(B_z(t))| \le \|\Delta B_z(t)\|$. Under the stated bound, it immediately follows that $\sigma_{\min}(\hat{B}_z(t)) \ge \sigma_{\min}(B_z(t)) - \|\Delta B_z(t)\| > 0$, which indicates that $\hat{B}_z(t)$ retains full column rank. Consequently, $\hat{B}_z(t)^\top \hat{B}_z(t)$ is SPD, and thus, the matrix $\hat{\bar{B}}_z(t) = \hat{B}_z(t) \otimes I_3$ is likewise SPD. Therefore, Algorithm 1 reliably converges to (18). ∎

The condition in Theorem 3 directly relates to the geometric arrangement of robot and feature positions. Specifically, the smallest singular value $\sigma_{\min}(B_z(t))$ quantifies geometric degeneracy. When this singular value approaches zero, the geometric configuration becomes nearly degenerate (e.g., features are collinear or coplanar), significantly amplifying the effect of measurement noise. Therefore, the eigenvalue criterion serves as a geometric identifiability constraint, ensuring the localization problem remains well-conditioned.

Remark 2: The conditions specified in Theorem 3 provide sufficient criteria to ensure bounded geometric error. However, even if the noise magnitude exceeds these bounds, as long as the matrix $\hat{\bar{B}}_z(t)^\top \hat{\bar{B}}_z(t)$ remains SPD, Algorithm 1 will still converge monotonically to the unique least squares solution (18) of the corresponding linear system (16). Practically, in scenarios of higher noise levels, geometric errors tend to increase proportionally with the system's condition number, resulting in slower numerical convergence.

In summary, the CG-based BL-SRAFL solver maintains global convergence and converges to the unique least-squares solution under realistic measurement noise, provided the geometric configuration is sufficiently non-degenerate and measurement perturbations remain within the established spectral bound.

D. BL-SRAFL+BA Implementation Pipeline

Building on the workflow outlined in Fig. 2 and the contents of above subsections, this subsection presents the detailed implementation of the BL-SRAFL+BA pipeline.

1) Initialization: This subsection describes a systematic initialization procedure which aims to establish at least four localized features in a three-dimensional space. Given the intrinsic calibration matrix K of the monocular camera, a known metric baseline length b between the first two time frames $t = 0$ and $t = 1$ (e.g., obtained from odometry or a calibrated motion), and four reliable feature correspondences that are simultaneously observed at $t = 0, 1, 2$, we adopt the minimal 4-point-in-3-view (4p3v) solver [41] to jointly estimate the relative rotation matrices $R_{r_0r_1}$ and $R_{r_0r_2}$ along with the associated unscaled translation matrices $\bar{T}^{r_0}_{r_0r_1}$ and $\bar{T}^{r_0}_{r_0r_2}$. The inherent scale ambiguity of monocular vision is resolved by a single global scale factor $s = b/\|\bar{T}^{r_0}_{r_0r_1}\|$, yielding the scaled translation matrices $T^{r_0}_{r_0r_1} = s\bar{T}^{r_0}_{r_0r_1}$ and $T^{r_0}_{r_0r_2} = s\bar{T}^{r_0}_{r_0r_2}$. The camera projection matrices for the three viewpoints are therefore defined as $P_{r_0} = [K \mid \mathbf{0}], P_{r_1} = [KR_{r_0r_1} \mid -KR_{r_0r_1}T^{r_0}_{r_0r_1}], P_{r_2} = [KR_{r_0r_2} \mid -KR_{r_0r_2}T^{r_0}_{r_0r_2}]$. Let $\mathbf{x}^{r_t}_f = [u^{r_t}_f, v^{r_t}_f, 1]^\top$ represent the homogeneous pixel coordinates of feature f in Σ^{r_t}, and $\mathbf{X}_f = [X, Y, Z, 1]^\top$ the corresponding homogeneous coordinates in three-dimensional space. For each feature f, three-view geometry provides a linear homogeneous system:

$$A_f\mathbf{X}_f = 0, \quad A_f = \begin{bmatrix} [\mathbf{x}^{r_t}_f]_\times P_{r_0} \\ [\mathbf{x}^{r_t}_f]_\times P_{r_1} \\ [\mathbf{x}^{r_t}_f]_\times P_{r_2} \end{bmatrix}, \tag{19}$$

solved by SVD. De-homogenizing the resulting $\mathbf{X}_f$ gives the global coordinates $\mathbf{p}_l$, for $l \in \{1, \ldots, 4\}$, serving as the initial localized feature set. Consequently, the scaled translation matrices obtained from the 4p3v initialization directly yield the first two robot positions in Σ^g, $\mathbf{p}_{r_1} = T^{r_0}_{r_0r_1}$ and $\mathbf{p}_{r_2} = T^{r_0}_{r_0r_2}$. Together with $R_{r_0r_2}$, these initial estimates provide all necessary inputs for the BL-SRAFL front-end.

2) BL-SRAFL Front-End: With the initialization procedure completed, we iteratively execute the BL-SRAFL front-end, performing incremental localization at each subsequent time frame. Specifically, at every time frame t, the sensing graph $\mathcal{G}(t)$ is updated by incorporating currently localized features and newly observed, unlocalized features. Although the global coordinates of the robot position r_{t-1} have been previously estimated at time frame $t-1$, we still include it in the unknown node set $\mathcal{V}_z(t)$ at the current iteration. This repeated estimation allows the solver to integrate additional available measurements from the current time frame, effectively reducing cumulative errors and mitigating drift that would otherwise accumulate if r_{t-1} were treated as a fixed localized node. After solving the localization problem at time t, the newly updated estimate of r_{t-1} replaces the previous value. Subsequently, the essential bearing angle measurements are acquired, and the associated weighted barycentric coordinates are computed according to the methods outlined in Subsection IV-B. These weighted barycentric coordinates are assembled systematically into a sparse linear system, as previously detailed. Next, the CG-based BL-SRAFL solver, presented in Subsection IV-C, efficiently solves this linear system, estimating the positions of the unlocalized features and updating the robot's positions. Newly estimated features are promptly incorporated into the localized feature set, progressively refining the spatial references. This iterative localization process continues as the robot navigates the environment and terminates upon the robot's cessation of movement or when all features have been localized.

3) Optional BA Back-End: Although BL-SRAFL provides globally consistent estimates under ideal conditions, real-world measurement noise and disturbances may degrade estimation accuracy. Moreover, because BL-SRAFL operates sequentially, estimation errors may propagate across successive steps and accumulate along the trajectory. The finite-step convergence of the CG solver applies to each individual linear system and therefore does not guarantee drift-free estimation over the entire trajectory. To further improve precision, we optionally apply a single offline BA refinement, utilizing the BL-SRAFL results as initial estimates. This BA step employs a sparse Levenberg-Marquardt solver [42] to minimize residual reprojection errors, improving localization performance at the cost of additional offline computation.

Remark 3: In this work, we adopt a global BA refinement executed once after completing the entire trajectory estimation. Although incremental or sliding-window BA approaches could further reduce drift accumulation, they significantly increase computational overhead per step. Consequently, the online latency is determined by the BL-SRAFL front-end, whereas the global BA is enabled only when the available offline computational budget and accuracy requirements justify its cost. Nevertheless, BL-SRAFL can be straightforwardly integrated with a sliding-window BA scheme if further drift mitigation is required and sufficient computational resources are available.

4) End-to-End Algorithm: Algorithm 2 provides detailed pseudocode illustrating the full BL-SRAFL+BA pipeline. It clarifies the seamless integration between the computationally efficient linear localization front-end and the precision-enhancing nonlinear BA back-end. The pseudocode defines computational procedures, parameter definitions, and algorithmic flow, facilitating straightforward and efficient implementation of the pipeline in practical tasks.

In summary, BL-SRAFL+BA combines the computational efficiency and rapid global convergence of the BL-SRAFL front-end with the enhanced accuracy and robustness of the BA back-end, ensuring robust and precise performance even in the presence of noise.

E. Computational Complexity and Memory

Building upon the complete BL-SRAFL+BA pipeline developed in the previous subsection, this subsection analyzes its computational complexity and memory requirements.

The initialization stage employs the minimal 4p3v solver exactly once, which involves assembling a constant-size matrix and solving a fixed-degree polynomial equation. Subsequently, it performs linear triangulation for four initial features, each requiring a SVD of constant dimension. Consequently, the entire initialization procedure has constant computational complexity, i.e., $\mathcal{O}(1)$. This initial overhead remains negligible compared to the computational demands of subsequent iterative BL-SRAFL operations. Additionally, the optional BA refinement employs a sparse Levenberg–Marquardt solver with

Algorithm 2 BL-SRAFL+BA Implementation Pipeline

Input: Image sequence, intrinsic matrix K, baseline length b
Output: Cumulative robot trajectory $\mathbf{p}_r^{\mathrm{cum}}(n_t)$ and cumulative unlocalized feature positions $\mathbf{p}_u^{\mathrm{cum}}(n_t)$

Solve initial global rotation and unscaled translation matrices $R_{r_0r_1}, R_{r_0r_2}, \bar{T}^{r_0}_{r_0r_1}$ and $\bar{T}^{r_0}_{r_0r_2}$ at $t = 0, 1, 2$ using 4p3v solver.
Initialize cumulative robot position set and compute corresponding coordinates with baseline length b: $\mathcal{V}_r^{\mathrm{cum}}(2) \leftarrow \{r_0, r_1, r_2\}$, $\mathbf{p}_r^{\mathrm{cum}}(2) \leftarrow [\mathbf{p}_{r_0}^\top, \mathbf{p}_{r_1}^\top, \mathbf{p}_{r_2}^\top]^\top$.
Initialize cumulative localized feature set and compute corresponding coordinates via (19): $\mathcal{V}_l^{\mathrm{cum}}(2) \leftarrow \{1, 2, 3, 4\}$, $\mathbf{p}_l^{\mathrm{cum}}(2) \leftarrow [\mathbf{p}_1^\top, \mathbf{p}_2^\top, \mathbf{p}_3^\top, \mathbf{p}_4^\top]^\top$.
Set initial time frame: $t = 3$.
repeat
  Extract matched feature sets $\mathcal{F}_{r_{t-1}}$ and $\mathcal{F}_{r_t}$ from consecutive images captured at time frames $t-1$ and t.
  Update sensing graph $\mathcal{G}(t)$ with the robot position set and feature set: $\mathcal{V}_r(t) \leftarrow \{r_{t-1}, r_t\}$, $\mathcal{V}_f(t) \leftarrow \mathcal{F}_{r_{t-1}} \cap \mathcal{F}_{r_t}$.
  Identify the localized feature set and extract their coordinates: $\mathcal{V}_l(t) \leftarrow \mathcal{V}_f(t) \cap \mathcal{V}_l^{\mathrm{cum}}(t-1)$, $\mathbf{p}_l(t) = [\mathbf{p}_l^{\mathrm{cum}}(t-1)]_{\mathcal{V}_l(t)}$.
  Identify the unlocalized feature set and update the cumulative unlocalized set: $\mathcal{V}_u(t) \leftarrow \mathcal{V}_f(t) \setminus \mathcal{V}_l(t)$, $\mathcal{V}_u^{\mathrm{cum}}(t) \leftarrow \mathcal{V}_u^{\mathrm{cum}}(t-1) \cup \mathcal{V}_u(t)$.
  Compute relative rotation and translation matrices $R_{r_{t-1}r_t}$ and $T^{r_t}_{r_tr_{t-1}}$ using the EPnP algorithm in Subsection IV-B1.
  Update global-to-local rotation matrix: $R_{r_0r_t} \leftarrow R_{r_{t-1}r_t} R_{r_0r_{t-1}}$.
  Acquire bearing angle measurements $\mathbf{b}^{r_t}_{r_tf}$ and $\mathbf{b}^{r_{t-1}}_{r_{t-1}f}$ via (6), and compute $\mathbf{b}^{r_t}_{r_tr_{t-1}}$ through (8).
  Compute weighted barycentric coordinates a_{xy} for each unlocalized node $x \in \mathcal{V}_z(t)$ following Subsection IV-B2.
  Construct BL-SRAFL matrices $B_z(t)$ and $B_l(t)$ according to (12) and (13).
  Solve (15) for robot positions $\mathbf{p}_r(t)$ and unlocalized feature positions $\mathbf{p}_u(t)$ using Algorithm 1.
  Update cumulative robot position set and corresponding coordinates: $\mathcal{V}_r^{\mathrm{cum}}(t) \leftarrow \mathcal{V}_r^{\mathrm{cum}}(t-1) \cup \mathcal{V}_r(t)$, $\mathbf{p}_r^{\mathrm{cum}}(t) \leftarrow [[\mathbf{p}_r^{\mathrm{cum}}(t-1)]_{\mathcal{V}_r^{\mathrm{cum}}(t-1)\setminus\{r_{t-1}\}}, \mathbf{p}_r(t)^\top]^\top$.
  Update cumulative localized feature set and corresponding coordinates: $\mathcal{V}_l^{\mathrm{cum}}(t) \leftarrow \mathcal{V}_l^{\mathrm{cum}}(t-1) \cup \mathcal{V}_u(t)$, $\mathbf{p}_l^{\mathrm{cum}}(t) \leftarrow [\mathbf{p}_l^{\mathrm{cum}}(t-1)^\top, \mathbf{p}_u(t)^\top]^\top$.
  Increment time frame: $t \leftarrow t+1$.
until Robot stops moving ($t = n_t + 1$)
Update cumulative unlocalized feature coordinates: $\mathbf{p}_u^{\mathrm{cum}}(n_t) \leftarrow [\mathbf{p}_l^{\mathrm{cum}}(n_t)]_{\mathcal{V}_u^{\mathrm{cum}}(n_t)}$.
if Optional BA refinement is enabled **then**
  Initialize BA solver with cumulative robot position coordinates $\mathbf{p}_r^{\mathrm{cum}}(n_t)$ and unlocalized feature coordinates $\mathbf{p}_u^{\mathrm{cum}}(n_t)$.
  Refine positions by minimizing residual reprojection errors using sparse Levenberg–Marquardt optimization.
  Update cumulative robot trajectory $\mathbf{p}_r^{\mathrm{cum}}(n_t)$ and cumulative unlocalized feature positions $\mathbf{p}_u^{\mathrm{cum}}(n_t)$.
end if
return $\mathbf{p}_r^{\mathrm{cum}}(n_t)$ and $\mathbf{p}_u^{\mathrm{cum}}(n_t)$

complexity typically between $\mathcal{O}((3n)^{1.5})$ and $\mathcal{O}((3n)^2)$ per iteration, where $n = n_f^{\mathrm{cum}}(n_t) + n_t$ is the number of unknowns. Given that BA is performed offline and converges in a limited number of iterations, the complexity analysis presented following focuses on the online BL-SRAFL front-end.

Theorem 4: Given a fixed maximum iteration cap $k_{\max}$ for Algorithm 1, the cumulative computational complexity over the entire trajectory scales linearly as $T_{\text{total}} = \mathcal{O}(k_{\max} \sum_{t=3}^{n_t} n_z(t))$.

Proof: Consider first the bearing angle measurement acquisition. Each consecutive pair of time frames invokes the EPnP algorithm once to compute the relative rotation and translation matrices, incurring a complexity of $\mathcal{O}(n_l(t))$. Constructing the corresponding bearing angle measurements involves traversing all matched features with $\mathcal{O}(n_f(t))$. Next, computing barycentric coordinates for unlocalized nodes at each time step involves solving a fixed-size least squares problem, resulting in $\mathcal{O}(n_z(t))$ scalar operations. Under typical visual geometric constraints, $n_l(t), n_f(t)$, and $n_z(t)$ remain asymptotically of the same order, thus these linear-time complexities are naturally subsumed within the overall complexity bound. The sparse linear system formulated in (15) has a dimension of $3n_z(t)$, whose sparsity is determined by the underlying barycentric coordinate induced graph structure. Specifically, each row of $B_z(t)$ corresponds to an unlocalized node, with nonzero entries located exclusively at columns indexed by neighboring nodes in the barycentric coordinate induced graph and at the diagonal. Hence, the number of nonzero elements in $B_z(t)$ is given by $\operatorname{nnz}(B_z(t)) = \sum_{i=1}^{n_z(t)} (d_i(t)+1) = 2m_z(t) + n_z(t)$, where $d_i(t)$ is the node degree and $m_z(t)$ the edge count of the induced graph. Typical geometric constraints, such as limited fields of view and occlusions, naturally keep the average node degree bounded, implying $\operatorname{nnz}(B_z(t)) = \mathcal{O}(n_z(t))$ and $\operatorname{nnz}\big(B_z(t)^\top B_z(t)\big) = \mathcal{O}(n_z(t))$. Constructing matrices $B_z(t)$ and their Kronecker expansion $\bar{B}_z(t) = B_z(t) \otimes I_3$ requires traversing every nonzero entry exactly once, contributing another $\mathcal{O}(n_z(t))$ complexity per step. But it is of the same order as other linear terms and also absorbed within the total bound. Algorithm 1 involves two sparse matrix-vector multiplications per iteration, resulting in complexity of $\Theta(\operatorname{nnz}(B_z(t))) = \Theta(n_z(t))$. Hence, with the fixed iteration limit $k_{\max}$, the complexity per step is $\mathcal{O}(k_{\max} n_z(t))$, and summing across all steps yields the total complexity T_{total} as claimed. ∎

Theorem 5: The peak memory usage per localization step scales as $M_{\text{peak}}(t) = \mathcal{O}(n_z(t) + n_l^{\mathrm{cum}}(t-1) + n_r^{\mathrm{cum}}(t-1))$.

Proof: Each localization step involves storing three primary data blocks. Firstly, the sparse matrix $\bar{B}_z(t)$ requires storage for $3\operatorname{nnz}(B_z(t))$ floats. Secondly, Algorithm 1 maintains five working vectors (residual, search direction, intermediate product, current solution estimate, and constant right-hand side), each of dimension $3n_z(t)$, totaling approximately $15n_z(t)$ floats. Lastly, coordinates of already localized features and robot positions are persistently stored, requiring $3(n_l^{\mathrm{cum}}(t-1) + n_r^{\mathrm{cum}}(t-1))$ floats. The computation of barycentric coordinates introduces only a constant-size temporary storage per sensing subgraph, and thus contributes negligibly to the overall memory footprint. Therefore, combining these factors and ignoring constant terms, the peak memory requirement scales as described in Theorem 5. ∎

In summary, the computational complexity and memory analysis demonstrate that the proposed BL-SRAFL method achieves linear scalability in terms of both computational effort and memory usage. These characteristics support its theoretical scalability for online localization on resource-constrained embedded platforms.

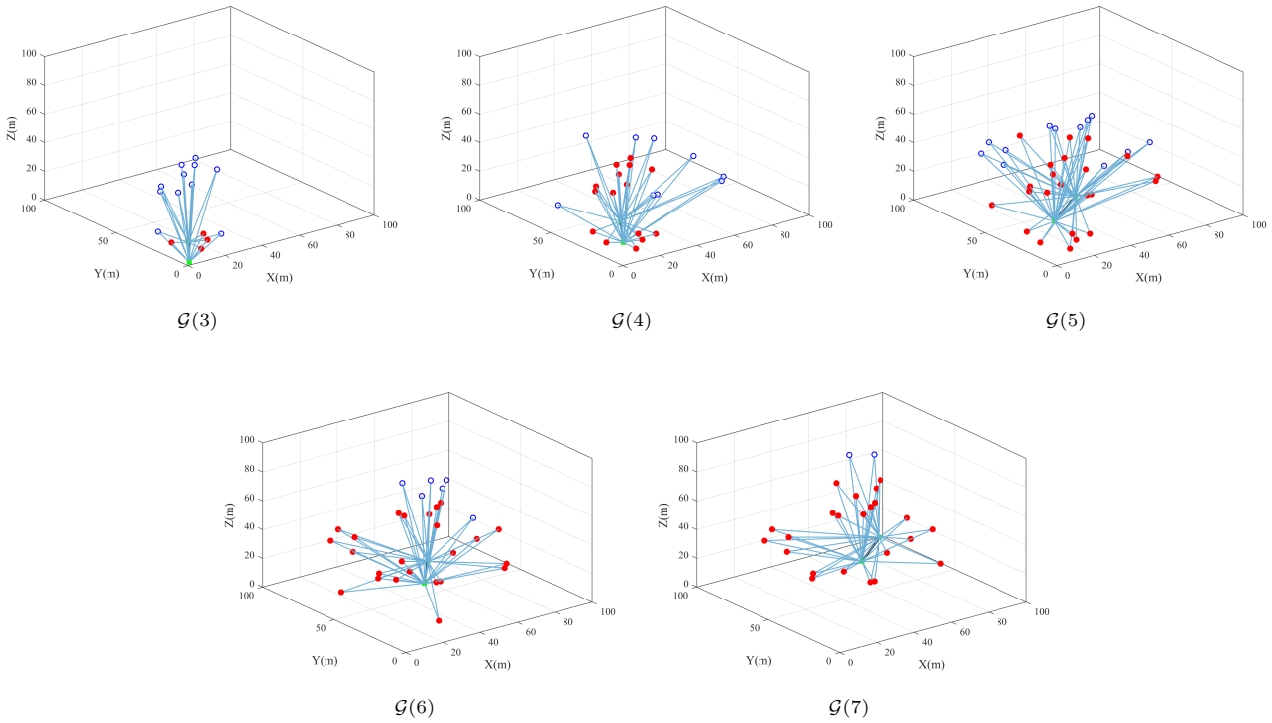


Fig. 4. Plots illustrating the evolution of the dynamic sensing graph $\mathcal{G}(t)$ for the five moves of the robot. Localized features are represented in red, unlocalized features in blue, and the robot in green.

V. Validation

This section validates the proposed approach through numerical simulations and real-world experiments, complemented by a comparison of iterative linear solvers.

A. Simulation Studies

We start by conducting numerical simulations to validate the BL-SRAFL in an idealized, noise-free scenario. Specifically, we consider an environment consisting of 44 features randomly distributed in a three-dimensional 100m$\times$100m$\times$100m domain. Among these features, 4 of them are initially localized, while the remaining 40 are unlocalized and their positions are unknown. A mobile robot performs 5 sequential moves after initialization, and we execute Algorithm 1 to iteratively estimate the positions of both the 40 unlocalized features and the robot step by step.

Fig. 4 illustrates the plots of the dynamic sensing graph $\mathcal{G}(t)$ as the robot moves. These plots visually represent the changes in the graph topology. It can be seen that certain edges vanish, while new ones are formed as the robot moves, based on the visibility determined by the robot's field of view. Additionally, we observe how previously unlocalized features become localized through earlier estimation steps, subsequently acting as localized features in subsequent steps. This iterative process gradually increases the localized feature set.

All subprocess convergence procedures are shown in Fig. 5, where each plot illustrates the estimated positions at the end of the iterations using Algorithm 1 compared to the exact positions. It should be emphasized that each panel in Fig. 5 shows the convergence result for the nodes contained in the corresponding sensing graph $\mathcal{G}(t)$. At each robot move, Algorithm 1 estimates the robot position and the unlocalized feature positions associated with the current sensing graph, after which the newly estimated features are added to the cumulative localized feature set. Consequently, the number of localized features increases progressively from $\mathcal{G}(3)$ to $\mathcal{G}(7)$. After the estimation associated with $\mathcal{G}(7)$ is completed, all 40 initially unlocalized features have been incorporated into the cumulative localized feature set, and the robot motion terminates. The comprehensive results of the SRAFL process are depicted in Fig. 6, where successive estimated robot positions are connected by line segments to illustrate the estimated trajectory. As shown, BL-SRAFL jointly estimates the robot trajectory and the positions of the environmental features.

To evaluate the numerical convergence of Algorithm 1, Fig. 7 plots the normalized iterative error $\|\hat{\mathbf{p}}_z(t)[k] - \mathbf{p}_z(t)\|/\|\hat{\mathbf{p}}_z(t)[0] - \mathbf{p}_z(t)\|$ over the iterations for each sensing graph $\mathcal{G}(t)$. These plots confirm the fast global convergence of the proposed method, validating our earlier claims regarding the computational complexity. Note that the convergence speed varies across different sensing graphs $\mathcal{G}(t)$ because each graph yields a different linear system as the observed feature set and sensing geometry evolve. Therefore, the faster convergence observed in some later steps is specific to this simulation and is not expected to occur monotonically for every trajectory.

Overall, our simulation results show that BL-SRAFL achieves accurate simultaneous localization of the robot and

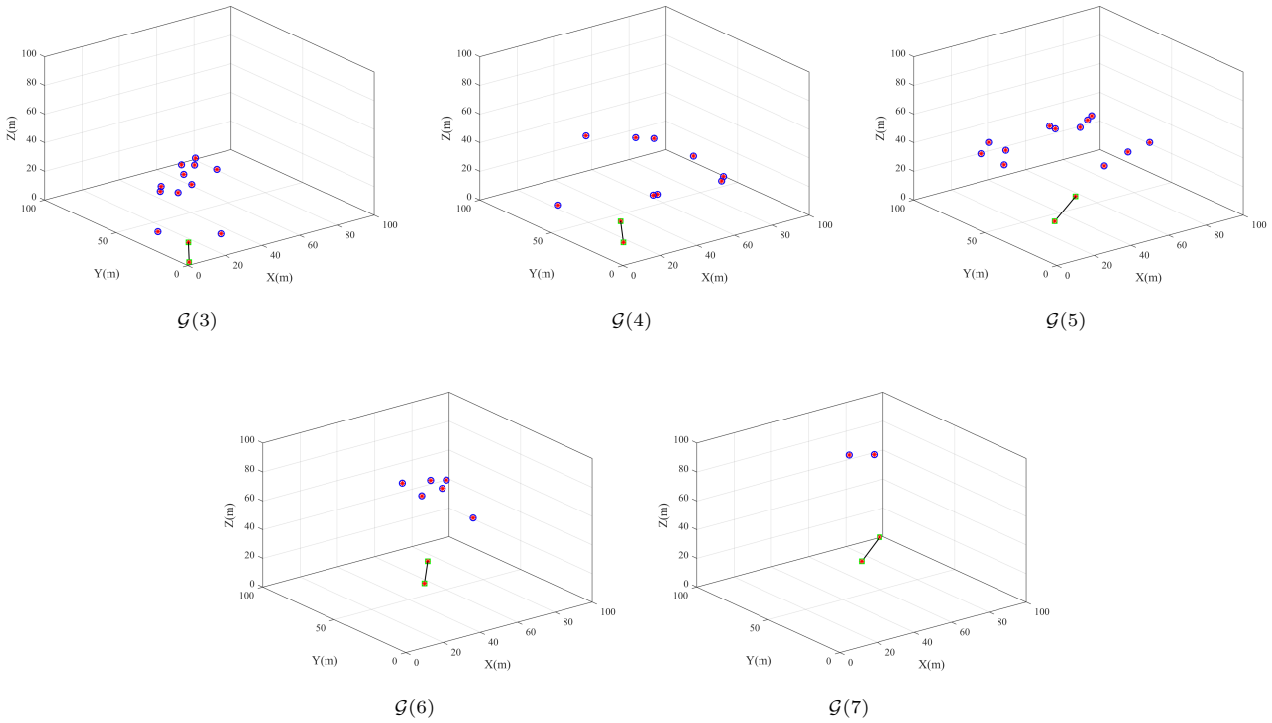


Fig. 5. Plots illustrating the convergence results of Algorithm 1 for the five moves of the robot. Blue circles and green squares are positions of features and robot, respectively, red asterisks are the final estimated positions of the robot and features, and black lines represent the estimated robot trajectory.

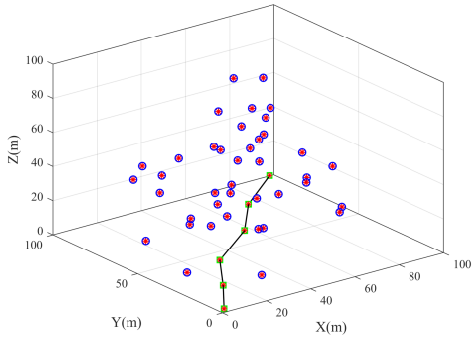


Fig. 6. Illustration of the final three-dimensional robot trajectory and feature positions generated by BL-SRAFL for the case study in Subsection V-A.

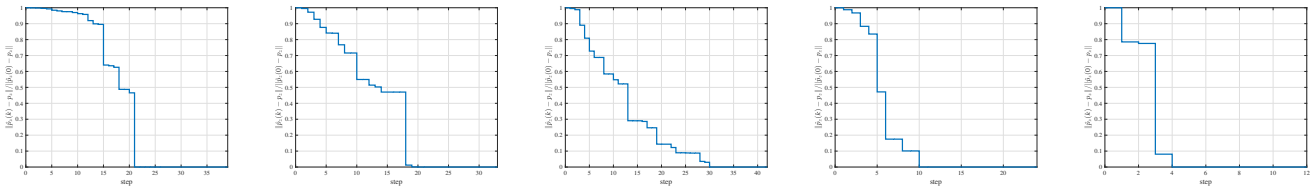
environmental features. The method effectively manages dynamic topology changes and incrementally incorporates newly localized features into the existing localized set, ensuring estimation performance throughout the robot's trajectory.

B. Real-world Experiments

We evaluate the robustness of the proposed approach by conducting a real-world experiment using a vertically-oriented handheld monocular camera, emulating the motion of a mobile robot traveling at a constant height. Ten reflective spheres are mounted at varying heights on brackets in a 1.5m$\times$1.5m$\times$1.5m domain, four of which serve as the initial localized features, while the remaining six require localization. In the visual front-end, the reflective spheres are detected from their appearance and geometric properties, and correspondences between consecutive frames are established by matching the detected sphere centers in the image plane. Ground-truth positions are captured by an OptiTrack Motion Capture system. Localization accuracy is evaluated by the Euclidean position errors between the estimated and ground-truth positions of the robot and unlocalized features.

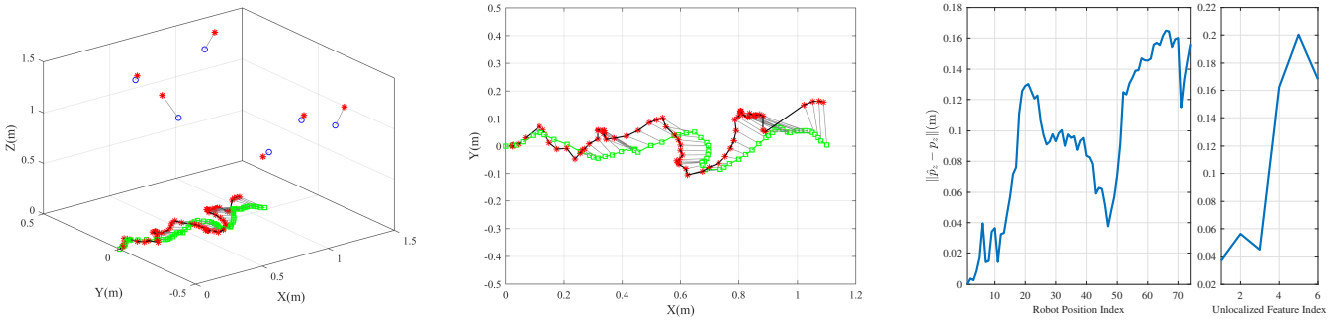
Experimental results through BL-SRAFL are presented in Fig. 8. Specifically, Fig. 8(a) illustrates the three-dimensional visualization of the estimated robot trajectory and feature positions, while Fig. 8(b) offers a complementary top-down XY-plane projection of the estimated robot trajectory. These results clearly indicate that our BL-SRAFL effectively captures both the robot's trajectory and the feature positions even in the presence of noise and limited visual cues. However, due to these practical constraints, the localization estimates naturally exhibit certain discrepancies. Quantitatively, as depicted in Fig. 8(c), the Euclidean position errors along the robot trajectory generally range from approximately 0.05m to 0.17m, with some unlocalized features exhibiting slightly larger errors up to around 0.2m.

Fig. 9 illustrates the experimental localization results of BL-SRAFL+BA. The effectiveness of BA is evident from Fig. 9(a) and Fig. 9(b), showing substantially improved accuracy and smoothness in both three-dimensional and XY-plane representations of the robot's trajectory and feature positions. From a quantitative perspective, Fig. 9(c) clearly illustrates a significant reduction in localization errors after BA optimization: the robot trajectory's maximum Euclidean position error decreases substantially from about 0.17m to less than 0.09m, while



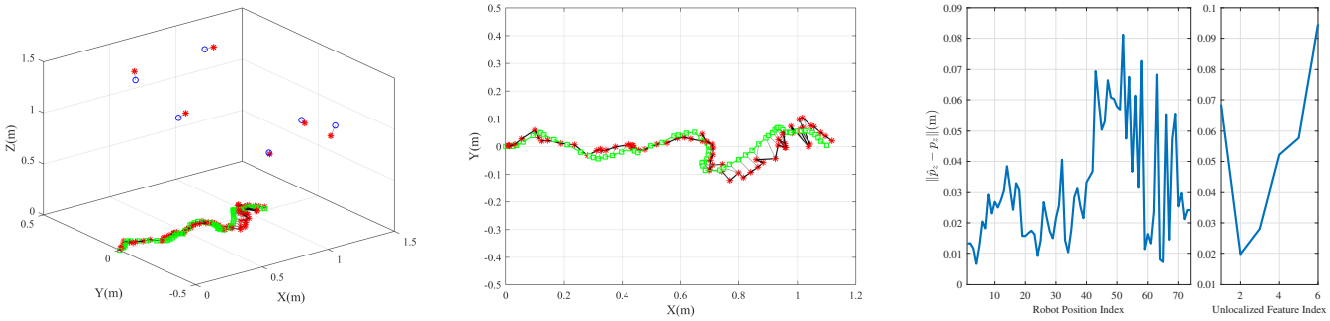
$\mathcal{G}(3)$ $\mathcal{G}(4)$ $\mathcal{G}(5)$ $\mathcal{G}(6)$ $\mathcal{G}(7)$

Fig. 7. Plots illustrating the temporal evolution of the normalized error ratio for the five moves of the robot.



(a) Three-dimensional view (b) XY-plane projection (c) Localization error

Fig. 8. Illustration of the experimental localization results of BL-SRAFL for the case study in Subsection V-B.



(a) Three-dimensional view (b) XY-plane projection (c) Localization error

Fig. 9. Illustration of the experimental localization results of BL-SRAFL+BA for the case study in Subsection V-B.

the Euclidean position errors of the unlocalized features also decrease from an initial maximum of around 0.2m to under 0.1m. This demonstrates the effectiveness of BL-SRAFL+BA, where the lightweight linear front-end estimator combined with the classical nonlinear BA back-end yields enhanced accuracy and robustness for practical localization tasks.

Overall, these experimental results underscore the practical robustness and utility of the proposed BL-SRAFL and BL-SRAFL+BA, particularly highlighting the significant accuracy gains achieved through the integrated BL-SRAFL+BA strategy under realistic measurement conditions.

C. Comparison of Linear Solvers

To further evaluate the CG-based BL-SRAFL solver, we compare it with several representative iterative linear solvers, including Jacobi Under-Relaxation (JURI) [28], classical Jacobi (JACOBI), Successive Over-Relaxation (SOR) [43], and LSQR [44]. All methods are applied to the same BL-SRAFL linear systems generated from the local five-node construction described in Subsection IV-B2, with four localized features and $n_z(t) = 20$ or 50 unknown nodes. This setting allows the comparison to focus specifically on the numerical solution of the resulting linear systems. Consistent with the solvability analysis in Subsection IV-C1, the BL-SRAFL systems considered here have full-rank coefficient matrices and therefore admit unique solutions. For a consistent comparison, all solvers are evaluated using the relative residual of the original BL-SRAFL linear system in (14), defined as $\rho = \frac{\left\|\bar{B}_z(t)\hat{\mathbf{p}}_z(t)+\bar{B}_l(t)\mathbf{p}_l(t)\right\|}{\left\|\bar{B}_l(t)\mathbf{p}_l(t)\right\|}$. The same convergence criterion, $\rho \leq 10^{-8}$, and the same maximum iteration limit, $\max(100, 3n_z(t))$, are used for all solvers. A solver is regarded as converged if the prescribed relative residual tolerance is reached within this common iteration budget.

TABLE I: Comparison of linear solvers on the same full-rank BL-SRAFL systems. Each run terminates when $\rho \leq 10^{-8}$ or the common iteration limit $\max(100, 3n_z(t))$ is reached. Runtime is reported as the median over five repetitions.

$n_z(t)$	Solver	Iterations	Time (ms)	Relative residual	Converged
20	CG	29	0.68	1.8×10^{-9}	Yes
20	JURI	100	24.19	3.2×10^{-3}	No
20	JACOBI	100	24.11	3.7×10^{13}	No
20	SOR	100	551.94	6.4×10^{-4}	No
20	LSQR	50	2.31	1.2×10^{-8}	No
50	CG	111	1.97	7.0×10^{-9}	Yes
50	JURI	150	29.98	2.9×10^{14}	No
50	JACOBI	150	29.02	3.6×10^{38}	No
50	SOR	150	1999.55	4.7×10^{-4}	No
50	LSQR	150	6.55	1.5×10^{-4}	No

As shown in Table I, CG is the only solver that reaches the prescribed relative-residual tolerance for both system sizes, requiring 29 iterations for $n_z(t) = 20$ and 111 iterations for $n_z(t) = 50$. For the smaller system, LSQR approaches the prescribed tolerance with a relative residual of 1.2×10^{-8}, but does not satisfy the convergence criterion within the iteration limit. For the larger system, its residual remains at 1.5×10^{-4} after 150 iterations. SOR also reduces the residual for both system sizes but does not reach the prescribed tolerance and incurs substantially higher computational time. In contrast, JURI and JACOBI exhibit unfavorable convergence behavior on the tested systems, with their residuals increasing markedly for $n_z(t) = 50$. Overall, CG achieves the prescribed accuracy for both system sizes while requiring substantially less runtime than the stationary iterative methods considered here. Together with the SPD structure established in Subsection IV-C2, these results support the use of CG as the linear solver for BL-SRAFL.

VI. Conclusion and Future Work

We proposed a barycentric linear method for the SRAFL problem in a three-dimensional space, relying solely on angle-based sensing. By reformulating the nonlinear SRAFL problem into a linear framework, the proposed BL-SRAFL achieves linear scalability in computational and memory complexity with respect to the number of unknown nodes. This innovation enables efficient localization suitable for resource-constrained embedded platforms. Furthermore, we have developed a CG-based solver designed for the sparse linear system, which guarantees global convergence under realistic observability conditions. To further enhance the localization accuracy, especially in noisy real-world conditions, an optional single BA refinement step complements the linear front-end offline at additional offline computational cost. Theoretical analyses supported by simulations and experiments validate the proposed approach. Although the proposed framework is developed using a monocular camera, the underlying barycentric formulation is not restricted to this sensing modality. It only relies on local bearing or relative-direction constraints together with metric scale information, and is therefore applicable to other sensing modalities capable of providing equivalent geometric measurements.

Future work will extend this approach to address more complex and challenging environments, including dynamic scenarios with moving features and more detailed modeling and propagation of measurement noise. Furthermore, real-time online implementation on embedded hardware platforms will be explored to facilitate practical deployment, alongside integration into broader robotic perception and navigation frameworks.

References

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. MIT Press, 2011.

[2] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Elsevier, 2013.

[3] Z. Li, Y. Yang, and H. Cheng, "Efficient multi-agent cooperation: Scalable reinforcement learning with heterogeneous graph networks and limited communication," *Knowl.-Based Syst.*, vol. 300, p. 112124, 2024.

[4] M. R. Jabbari, M. R. Taban, S. Gazor, and M. Kaimasi, "Asymptotically efficient moving target localization in distributed radar networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 9, pp. 569–580, 2023.

[5] I. Ullah, D. Adhikari, H. Khan, M. S. Anwar, S. Ahmad, and X. Bai, "Mobile robot localization: Current challenges and future prospective," *Comput. Sci. Rev.*, vol. 53, p. 100651, 2024.

[6] S. Atiya and G. D. Hager, "Real-time vision-based robot localization," *IEEE Trans. Robot. Autom.*, vol. 9, no. 6, pp. 785–800, 1993.

[7] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 1999, pp. 588–594.

[8] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artif. Intell. Rev.*, vol. 43, pp. 55–81, 2015.

[9] Davison, "Real-time simultaneous localisation and mapping with a single camera," in *IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1403–1410.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[12] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[14] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[15] L. Pan, D. Barath, M. Pollefeys, and J. L. Schönberger, "Global structure-from-motion revisited," in *European Conference on Computer Vision*, 2024.

[16] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Int. Symp. Info. Process. Sens. Netw.*, 2004, pp. 20–27.

[17] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Trans. Mob. Comput.*, vol. 5, no. 12, pp. 1663–1678, 2006.

[18] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007.

[19] T. Erseghe, "A distributed and maximum-likelihood sensor network localization algorithm based upon a nonconvex problem formulation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 247–258, 2015.

[20] N. Piovesan and T. Erseghe, "Cooperative localization in wsns: A hybrid convex/nonconvex solution," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 162–172, 2018.

[21] H. Naseri and V. Koivunen, "A bayesian algorithm for distributed network localization using distance and direction data," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 290–304, 2019.

[22] U. A. Khan, S. Kar, and J. M. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, 2009.

[23] Y. Diao, Z. Lin, and M. Fu, "A barycentric coordinate based distributed localization algorithm for sensor networks," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4760–4771, 2014.
[24] Z. Lin, M. Fu, and Y. Diao, "Distributed self localization for relative position sensing networks in 2-D space," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3751–3761, 2015.
[25] P. Cheng, T. Han, X. Zhang, R. Zheng, and Z. Lin, "A single-mobile-anchor based distributed localization scheme for sensor networks," in *Chin. Control Conf.*, Chengdu, China, 2016, pp. 8026–8031.
[26] Z. Lin, T. Han, R. Zheng, and M. Fu, "Distributed localization for 2-D sensor networks with bearing-only measurements under switching topologies," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6345–6359, 2016.
[27] L. Chen, "Triangular angle rigidity for distributed localization in 2D," *Automatica*, vol. 143, p. 110414, 2022.
[28] Y. Xia, C. Yu, and C. He, "An exploratory distributed localization algorithm based on 3D barycentric coordinates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 702–712, 2022.
[29] L. Chen, K. Cao, L. Xie, X. Li, and M. Feroskhan, "3-D network localization using angle measurements and reduced communication," *IEEE Trans. Signal Process.*, vol. 70, pp. 2402–2415, 2022.
[30] L. Chen, Z. Lin, and L. Xie, "Angle-based distributed node localizability and localization," *IEEE Trans. Autom. Control*, vol. 69, no. 3, pp. 1890–1897, 2023.
[31] L. Shi, S. Yao, N. Ning, and Y. Zhou, "Barycentric coordinate-based distributed localization over 3-D underwater wireless sensor networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 10, pp. 640–649, 2024.
[32] J. Wu, L. Zino, Z. Lin, and A. Rizzo, "Distributed finite-time cooperative localization for three-dimensional sensor networks," *IEEE Trans. Netw.*, vol. 33, no. 5, pp. 2764–2778, 2025.
[33] A. F. Möbius, *Der Barycentrische Calcul*. Verlag von Johann Ambrosius Barth, 1827.
[34] S. J. Gortler and D. P. Thurston, "Characterizing the universal rigidity of generic frameworks," *Discrete Comput. Geom.*, vol. 51, no. 4, pp. 1017–1036, 2014.
[35] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *IEEE Int. Conf. Comput. Commun.*, vol. 4, HongKong, China, 2004, pp. 2673–2684.
[36] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, 2009.
[37] G. Strang, *Introduction to Linear Algebra*. SIAM, 2022.
[38] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han, "Necessary and sufficient graphical conditions for affine formation control," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2877–2891, 2015.
[39] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.
[40] L. Mirsky, "Symmetric gauge functions and unitarily invariant norms," *Quart. J. Math.*, vol. 11, no. 1, pp. 50–59, 1960.
[41] D. Nistér and F. Schaffalitzky, "Four points in two or three calibrated views: Theory and practice," *Int. J. Comput. Vis.*, vol. 67, pp. 211–231, 2006.
[42] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
[43] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.
[44] C. C. Paige and M. A. Saunders, "Lsqr: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, 1982.

Jinze Wu received the B.Eng. degree in vehicle engineering from Dalian University of Technology, Dalian, China, in 2018, the M.Eng. degree in aeronautical and astronautical science and technology from Sun Yat-sen University, Shenzhen, China, in 2022, and the Ph.D. degree in electrical, electronics and communications engineering from Politecnico di Torino, Turin, Italy, in 2026. During her doctoral studies, she participated in a joint doctoral programme between Politecnico di Torino and Southern University of Science and Technology. She is currently a postdoctoral researcher with Guangzhou Automobile Group Co., Ltd., Guangzhou, China. Her research interests include multiagent systems and distributed control and algorithms.

Lorenzo Zino is an Associate Professor with the Department of Electronics and Telecommunications, Politecnico di Torino, Italy, since 2026. He received BS (2012), MS (summa cum laude, 2014), and PhD (with honors, 2018) in Applied Mathematics from Politecnico di Torino. He held Research Fellowships at Politecnico di Torino, University of Groningen, and New York University. He has co-authored more than 100 international scientific publications, including more than 60 journal papers. In 2024, he received the Best Young Author Journal Paper Award from the IEEE CSS Italy Chapter. He is Member of the Editorial Board of Automatica, IEEE Control Systems Letters, International Journal of Control, and Scientific Reports, and of the EUCA Conference Editorial Board.

Zhiyun Lin received the PhD in electrical and computer engineering from University of Toronto, Canada, in 2005. He is currently a Full Professor with the Southern University of Science and Technology, China. Preceding to this position, he was a Postdoctoral Research Associate (2005–07) with the University of Toronto, then worked as a Research Professor (2007–11) and a Full Professor (2012–17) at Zhejiang University, China, and served as the Director of Artificial Intelligence Institute (2017–21), Hangzhou Dianzi University, China. He has held many visiting Professor appointments, including Yale University, the Australian National University, University of Groningen, University of Cagliari, University of Newcastle, University of Technology Sydney. His research interests focus on multiagent systems, distributed artificial intelligence, autonomous systems and swarm robots, and cyber-physical systems. He is an Associate Editor for IEEE Systems Journal. He is a fellow of IEEE and IET.

Alessandro Rizzo is a Professor in the Department of Electronics and Telecommunications at Politecnico di Torino, where he directs the Complex Systems Laboratory. He received his Laurea degree in Computer Engineering, summa cum laude, and his PhD in Automation and Electronics Engineering from the University of Catania in 1996 and 2000, respectively. He joined Politecnico di Torino as a tenured Associate Professor in 2015 and was promoted to Full Professor in 2026. His previous affiliations include the JET Joint Undertaking, STMicroelectronics, the University of Messina, Politecnico di Bari, and New York University. His research interests span complex networks and systems, robotics, and the modeling and control of nonlinear systems. He has authored two books, holds two international patents, and has published more than 200 papers in peer-reviewed international journals and conference proceedings. An IEEE Nuclear and Plasma Sciences Society Distinguished Lecturer, he received the Best Application Paper Award at the 2002 IFAC World Congress and two Amazon Research Awards in Robotics, in 2019 and 2021.