Iterative Localization of Wireless Sensor Networks: An Accurate and Robust Approach

Qingjun Xiao, *Member, IEEE*, Bin Xiao, *Senior Member, IEEE, Member, ACM*, Kai Bu, and Jiannong Cao, *Senior Member, IEEE, Member, ACM*

***Abstract*—In wireless sensor networks, an important research problem is to use a few anchor nodes with known locations to derive the locations of other nodes deployed in the sensor field. A category of solutions for this problem is the *iterative localization*, which sequentially merges the elements in a network to finally locate them. Here, a network element is different from its definition in iterative trilateration. It can be either an individual node or a group of nodes. For this approach, we identify a new problem called *inflexible body merging*, whose objective is to align two small network elements and generate a larger element. It is more generalized than the traditional tools of trilateration and patch stitching and can replace them as a new merging primitive. We solve this problem and make the following contributions. 1) Our primitive can tolerate ranging noise when merging two network elements. It adopts an optimization algorithm based on rigid body dynamics and relaxing springs. 2) Our primitive improves the robustness against flip ambiguities. It uses orthogonal regression to detect the rough collinearity of nodes in the presence of ranging noise, and then enumerate flip ambiguities accordingly. 3) We present a condition to indicate when we can apply this primitive to align two network elements. This condition can unify previous work and thus achieve a higher percentage of localizable nodes. All the declared contributions have been validated by both theoretical analysis and simulation results.**

***Index Terms*—Ambiguity enumeration, noise toleration, fine-grained localization, wireless sensor networks.**

I. Introduction

FOR WIRELESS sensor networks, an important research problem is to use a few anchor nodes with known locations to derive the locations of other nodes in the sensor fields. Besides anchor locations, the additional information that are assumed to be known are the distance measurements between neighboring sensor nodes. These measurements can be taken by a so-called *ranging technique*. We divide the available ranging techniques into two categories according to their accuracy.

- Coarse-grained ranging techniques have low accuracy only at meter level because they leverage the anisotropic attenuation of radio signals for distance measurement. This category covers both the RSSI-based methods [1], [2] and the hopcount-based methods [3]–[7].
- Fine-grained ranging techniques can achieve much higher accuracy at (sub)centimeter level since they utilize the stable time of arrival (TOA) of certain signals, e.g., ultra-wideband radio signals, or ultrasound signals [8].

In this paper, we adopt the fine-grained ranging techniques to obtain the accurate measurements of internode distances.

For fine-grained localization, the existing solutions are divided into two categories: *whole-topology approach* and iterative approach. The former approach analyzes the whole network topology directly, using some numerical optimization algorithms. For example, anchor-free localization (AFL) method models the network topology as a group of nodes interconnected by springs [9]. The spring forces acting on the nodes are used as heuristics that guide the nodes' movement toward their lowest-energy positions. Another example is MDS-Map, which analyzes the network topology directly using a technique called multidimensional scaling [10]. These whole-topology methods, however, share a drawback. The network topology they manipulate contains too many free variables. Just imagine a topology with 100 nodes. Since each node has two variables x, y in 2-D space, this system would contain 200 free variables in total. Any methods that analyze it directly will get stuck easily in local minima. This problem is exaggerated particularly in concave-shaped topologies where the good initial guesses of node locations are difficult to obtain.

In contrast, the *iterative approach* can reduce the number of system variables under consideration by splitting a large network topology into many small network elements [11]–[16]. An element can be either an individual node or a group of nodes. Each element has its own coordinate frame. To localize these elements, we pick up two following certain rules and merge them to share a coordinate frame and generate a larger element. This merging operation is a primitive that can be applied recursively. All the network elements can be localized when they share one unified coordinate frame. The advantage of iterative approach is that it can help avoid local minima by a *dimension reduction* technique. Just imagine when merging two network elements, we will tune their relative position and orientation to fit their constraints. This means we only need to manipulate three free variables in 2-D space, rather than handling the whole topology.

For the iterative approach, the previous works have investigated different necessary conditions for merging two network elements. Iterative multilateration can merge an individual node with a patch when the node is connected to the patch by at least three distance measurements [11]. Patch stitching can align two patches when they share at least three nodes [12]. CALL proposed more generalized conditions than patch stitching, i.e., two patches share two nodes and are connected by a link, or they share one node and are connected by two links, or they are connected by four links [13]. SWEEPS can merge a sensor with

Manuscript received November 19, 2010; revised December 22, 2012; accepted March 19, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Kodialam. Date of publication April 25, 2013; date of current version April 14, 2014.

Q. Xiao was with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China. He is now with the University of Florida, Gainesville, FL 32611 USA (e-mail: csqjxiao@comp.polyu.edu.hk).

B. Xiao, K. Bu, and J. Cao are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: csbxiao@comp.polyu.edu.hk; cskbu@comp.polyu.edu.hk; csjcao@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2013.2257839

a patch ambiguously when the sensor connects to the patch by only two links. The location ambiguities can be eliminated later when sufficient constraints are available [14].

However, the previous studies in iterative localization presented only the necessary conditions for merging network elements. They fail to provide the sufficient conditions that are needed in practice. Thus, we unify the previously proposed conditions and enhance them to consider also the collinear geometry of sensor nodes. Besides the condition specification, there exist two other fundamental issues that are far from being fully addressed. 1) How to tolerate the ranging noise when we try to align two network elements? The traditional tool of trilateration can tolerate noise only when aligning a sensor to a patch [17]. The patch stitching can merge two patches only when they share three nodes [12]. 2) How to enumerate the flip ambiguities when merging network elements to guarantee the robustness of localization? These flip ambiguities can be caused by the nodes that are just roughly collinear due to the interference of ranging noise. The failure to enumerate them may cause the localization error to be abnormally large.

In this paper, we solve the problem of merging two network elements to form a larger one, which is called *body merging*. Here, "body" is a term that refers to either a single node or a patch. Note that a body may have a single or multiple ambiguities. Thus, we describe them by the following three terms, which are used throughout this paper:

- Rigid body: can have only one ambiguity;
- Inflexible body: has finite ambiguities;
- Flexible body: has a flexible part that can move continuously and produces infinite ambiguities.

We focus on the problem of *merging two inflexible bodies* to form a larger one, which has the following two subproblems:

- Accuracy: how to align the two bodies accurately by finding their optimal relative position and orientation that can minimize the mean squared error of their constraints;
- Robustness: The above optimal alignment can find only one of the possibilities of aligning two bodies. We should enumerate all the merging possibilities to ensure the robustness. Thus, we need to detect the collinear geometry of nodes and enumerate flip ambiguities accordingly.

By solving the problem, we make the following contributions.

- Condition: We present a precondition for merging inflexible bodies: Degree of Constraint $\geq$ Degree of Freedom. Its intuition is that two bodies should have enough constraints (sharing nodes or connected by links) to confine their relative motions. This condition can unify the previous works and achieve higher localization percentage than the state-of-the-art SWEEPS and CALL [14], [13].
- Accuracy: We propose an optimization algorithm to tolerate the noise in the constraints that connect two inflexible bodies. Our basic idea is to model the two bodies as connected by springs. Due to the spring forces, the bodies can move and rotate to their minimum energy states that minimize the mean squared error of constraints. The body motions are modeled by the canonical physical model of rigid body dynamics. This noise toleration algorithm is more generalized than the traditional multilateration [17] and patch stitching [12] since our algorithm can align two bodies, no matter if they are nodes or patches. Our algorithm is also fundamentally different from AFL [9], which also adopts the heuristic of spring relaxation. Our method is resilient against local minima, but AFL can easily trap in local minima when dealing with concave networks. This is because our method is essentially a dimension reduction technique, which divides the network into numerous small bodies and merges them recursively.
- Robustness: We present an algorithm to enumerate flip ambiguities when merging two bodies. Such enumeration is achieved by flipping one of the ambiguities across the node set that is detected to be collinear. Our contribution is that the previous work only considers the collinearity of anchor nodes [14]. However, for body merging, the collinearity of normal nodes can also produce flip ambiguities. Moreover, a set of nodes can be just roughly collinear as interfered by ranging noise. We can detect the rough collinearity using the tool of orthogonal regression.

The declared contributions have been validated by high-fidelity simulations configured with practical system parameters [15].

The rest of this paper is organized as follows. Section II reviews network localization problem and the iterative localization. Section III presents an optimization algorithm to merge two bodies accurately. Section IV presents the condition for merging two bodies uniquely. This condition considers both the constraints between two bodies and their collinear geometry. Section V proposes an algorithm to enumerate flip ambiguities and guarantee merging robustness. Section VI presents the experimental results. Section VII concludes this paper.

II. Background of Network Localization Problem and Iterative Localization

This section introduces the network localization problem and describes the iterative localization approach that divides the whole topology into bodies and merges them recursively.

Definition 1: The network localization problem is, given the following inputs, to generate the outputs described as follows.

Inputs are twofold: 1) distance measurements between neighboring nodes, and 2) a small proportion of special nodes called GCF anchors whose locations are known in Global Coordinate Frame (GCF). The anchor locations are assumed to contain noise that is negligibly small, as compared to ranging noise. The purpose of deploying anchors is to ensure the location estimates of other nodes are defined in GCF.

Outputs are the sensor nodes, each of which has a unique location estimate. Note that these location estimates should be defined in the GCF, which is meaningful to end-users. The typical GCF is the globally accepted GPS coordinate frame.

Example: An example of network localization is given in Fig. 1. Fig. 1(a) illustrates the inputs. The pairwise distance measurements are drawn as the edges between different nodes, and the special GCF anchors are depicted as triangles.Fig. 1(b) illustrates the outputs. The generated location estimates are drawn as black dots. Some nodes are not covered by the black dots because they have ambiguous location assignments. For example, node 20 does not have a black dot since it can be flipped across the line through nodes 15 and 21.

Iterative Localization: The above research problem can be solved by the iterative localization approach, which is composed of two phases: 1) divide the network topology into a set of small bodies, each of which has its own coordinate frame; 2) merge these bodies recursively to minimize the number of local coordinate frames. We detail these two phases as follows.

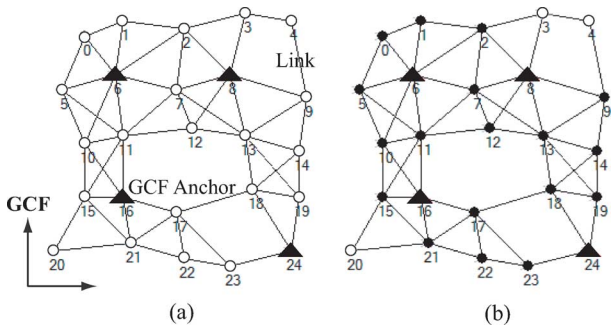
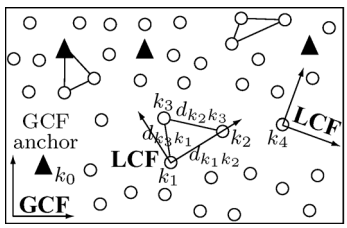


Fig. 1. Input and output of network localization problem. (a) Input: links and GCF anchors. (b) Output: location estimates.



Global Body with the GCF contains all the GCF anchors

Local Bodies with LCFs :

a triangle with k_1 at [0, 0], k_2 on x-axis at $[d_{k_1k_2}, 0]$, and k_3 above x-axis

a node with k_4 at [0, 0]

Fig. 2. Divide the network topology into elementary bodies.

Phase 1: Split the network topology into a list of bodies. These bodies fall into two types: a *global body* whose coordinate frame is the GCF, and numerous *local bodies*, whose coordinate frames are their local coordinate frames (LCFs). The global body contains all the anchor nodes initially. A local body can be either a triangle or an individual node. For example, in Fig. 2, the three nodes k_1, k_2, k_3 form a triangle with edges $d_{k_1k_2}, d_{k_2k_3}, d_{k_3k_1}$. For this triangle, its LCF can be constructed by placing the original point at node k_1, letting the x-axis go through node k_2, and placing node k_3 above the x-axis. Another example is the node k_4, which constitutes a local body with only one node. Its LCF can be built by placing the original point at k_4.

Phase 2: Merge the bodies iteratively to minimize the number of bodies. After phase 1, we obtain a set of bodies, each of which has its own coordinate frame. We can organize them as a list: $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_i, \ldots$, where $\mathcal{B}_0$ denotes the global body and $\mathcal{B}_i (i \geq 1)$ denotes a local body. From this body list, we select two bodies (e.g., bodies $\mathcal{B}_i$ and $\mathcal{B}_j$) and merge them to share a coordinate frame. This operation of selection and merging will be performed recursively until we cannot find a body pair that can be merged. This procedure finally outputs the global body $\mathcal{B}_0$ since all the nodes in this body have been localized in the GCF. As a summary, this iterative localization framework has three key research subproblems to address.

- How to choose two bodies that can be merged? This problem of specifying "precondition" is to be addressed in Section IV for unique body merging, and revisited in Section V-A for body merging with finite ambiguities.
- We need to figure out how to merge two bodies. This problem is twofold: 1) how to merge two bodies accurately by tolerating the ranging noise, and 2) how to merge two bodies robustly by enumerating flip ambiguities. The accuracy issue will be handled in Section III, and the robustness issue will be addressed in Section V-B.

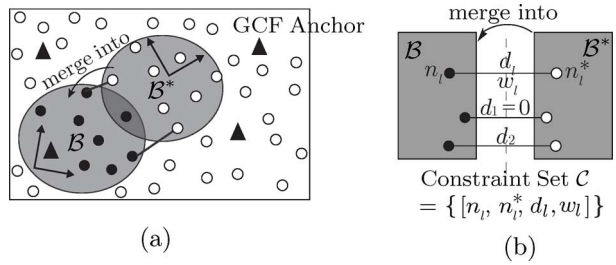


Fig. 3. Merging of two bodies $\mathcal{B}, \mathcal{B}^*$ with constraint set $\mathcal{C}$. (a) Merge body $\mathcal{B}^*$ into body $\mathcal{B}$. (b) Constraint set $\mathcal{C}$.

III. Accurate Body Merging Against Noise

When aligning two bodies, a challenge is the inevitable presence of ranging noise, which degrades the accuracy of body merging. To tolerate the ranging noise, we identify the problem of minimizing the mean-squared error of the constraints that connect two bodies. The solution we propose is more generalized than multilateration [17] and patch stitching [12].

A. Body Merging Optimization Problem

Merging two bodies is essentially to find a transformation function between their two coordinate frames. This transformation contains two subcomponents: translation and rotation. The transformation can be found due to the presence of constraints that confine the two bodies' relative moving and rotating. The constraints can be either shared nodes or links with known lengths to connect them. For example, in Fig. 3(a), the two bodies share one node and are connected by two links. However, the challenge is, when given a set of constraints, how to calculate the optimal transformation that can tolerate the ranging noises within the constraints.

This paper probably is the first to identify the generalized optimization problem of accurately merging body $\mathcal{B}^*$ into body $\mathcal{B}$ as shown in Fig. 3(a). Trilateration can be regarded a special case of our problem, in which body $\mathcal{B}^*$ is a single node and body $\mathcal{B}$ is a patch. Patch stitching is also a special case that aligns two patches when they share at least three nodes [12].

We introduce the notations used to model *constraints*, the inputs of our problem. As shown in Fig. 3(b), a constraint is depicted as an edge, whether it is a sharing node or a connecting link. One end of the edge is incident to body $\mathcal{B}$ and is drawn as a black dot. The other end is incident to body $\mathcal{B}^*$ and is drawn as a white node. When the length of the edge is zero, the corresponding constraint is a shared node. Otherwise, it is a real distance measurement. The two bodies $\mathcal{B}$ and $\mathcal{B}^*$ can have multiple constraints to confine their relative motions.

Definition 2 (Constraint Set During Body Merging): During the merging of body $\mathcal{B}^*$ into body $\mathcal{B}$, the two bodies have a set of constraints $\mathcal{C} = \{\ell_l\}$, where ℓ_l is the lth $(0 \leq l < |\mathcal{C}|)$ constraint contained in $\mathcal{C}$. Here, $\ell_l = [n_l, n_l^*, d_l, w_l]$, where

- d_l measured length of the constraint ℓ_l.
- w_l weight of the constraint ℓ_l.
- n_l node index of ℓ_l that is contained in body $\mathcal{B}$.
- n_l^* node index of ℓ_l that is contained in body $\mathcal{B}^*$.

- (Note: In all figures throughout the paper, we draw node n_l as a black node and draw node n_l^* as a white node.)

When a constraint ℓ_l corresponds to a node shared by body $\mathcal{B}$ and body $\mathcal{B}^*$, its length d_l is zero and its two node indices n_l,

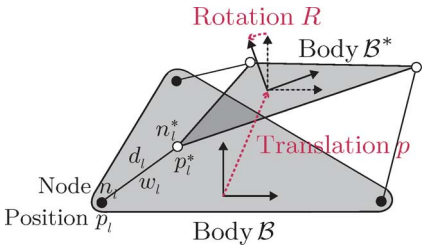


Fig. 4. Body merging optimization problem.

n_l^* are equal. Otherwise, a constraint ℓ_l is a link connecting the two bodies. Its two node indices n_l, n_l^* are different, and note that n_l, n_l^* should not be the indices of shared nodes.

For the constraints $\mathcal{C}$, it is inevitable for their length measurements to contain noises, which can degrade the accuracy of body merging. We thus formalize a problem to tolerate ranging noise by minimizing the mean squared error in the constraints. This problem is called *body merging optimization* problem.

Definition 3 (Body Merging Optimization Problem): Inputs are the set of constraints $\mathcal{C}$ that confines the relative motion between body $\mathcal{B}$ and body $\mathcal{B}^*$, and the node locations in the two bodies' local coordinate frames. We assume the following:

- p_l is the location of node n_l contained in body $\mathcal{B}$.
- p_l^* is the location of node n_l^* contained in body $\mathcal{B}^*$.

Note that p_l is defined in body $\mathcal{B}$'s coordinate frame, and p_l^* is defined in body $\mathcal{B}^*$'s coordinate frame. Thus, we can define the constraint set $\mathcal{C}'$ with each of its constraints ℓ_l' to have four fields $[p_l, p_l^*, d_l, w_l]$, where d_l is the measured length of constraint ℓ_l', and w_l is the weight to reflect d_l's measurement accuracy. We have illustrated all these notations in Fig. 4.

Variable to optimize is $T_{p,R}$, which is a transformation function from body $\mathcal{B}^*$'s coordinate frame to $\mathcal{B}$'s coordinate frame. The function $T_{p,R}$ can be defined as

$$T_{p,R}(p^*) = p + Rp^*, \text{ where} \tag{1}$$

where:

- p is the translation component, which, as illustrated in Fig. 4, is the position of axis origin of body $\mathcal{B}^*$ in body $\mathcal{B}$'s coordinate frame;
- R is the rotation of body $\mathcal{B}^*$ about body $\mathcal{B}^*$'s axis origin.

With this transformation function, a position p^* in body $\mathcal{B}^*$'s coordinate frame can be transformed to body $\mathcal{B}$'s coordinate frame as $T_{p,R}(p^*)$. Then, we can calculate the weighted error of constraint ℓ_l' as

$$\text{err}_l(T_{p,R}) = w_l(d_l - \|p_l - T_{p,R}(p_l^*)\|) \tag{2}$$

where $\|p_l - T_{p,R}(p_l^*)\|$ is the length of constraint ℓ_l' estimated from the two positions p_l and p_l^*. Thus, $\text{err}_l(T_{p,R})$ has reflected the difference between the measured error d_l and the estimated error when given the function $T_{p,R}$.

Objective function is the mean-squared error for all the constraints in set $\mathcal{C}'$, which is noted as $f_{\mathcal{C}'}(T_{p,R})$

$$f_{\mathcal{C}'}(T_{p,R}) = \sqrt{\frac{1}{|\mathcal{C}'|}\sum_{\ell_l' \in \mathcal{C}'} \text{err}_l(T_{p,R})^2} \tag{3}$$

where $|\mathcal{C}'|$ is the number of constraints in constraint set $\mathcal{C}'$.

Output is the optimal transformation that can minimize the objective function $f_{\mathcal{C}'}(T_{p,R})$, which is noted as $T_{p,R}^{\min}$

$$T_{p,R}^{\min} = \arg\min f_{\mathcal{C}'}(T_{p,R}).$$

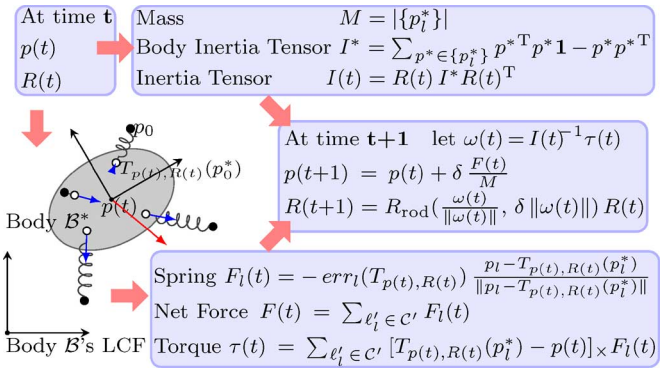


Fig. 5. Iterative optimization based on rigid body dynamics.

Restriction: For the optimized transformation $T_{p,R}^{\min}$, the error of each constraint ℓ_l' has its magnitude below threshold $2c\sigma$

$$\forall \ell_l' \in \mathcal{C}' \quad \text{err}_l(T_{p,R}^{\min}) < 2c\sigma \tag{4}$$

where:

- σ is the expected magnitude of ranging noise;
- c is a constant that can be 3 for Gaussian noise.

B. Body Merging Optimization Algorithm

For the problem in Definition 3, we propose a solution called *body merging optimization* algorithm. This algorithm revises the transformation function $T_{p,R}$ recursively and finally outputs the optimal transformation $T_{p,R}^{\min}$. Its basic idea is to model each constraint as an elastic spring (see Fig. 5). Each spring casts its force on body $\mathcal{B}^*$ due to spring deformation. As directed by the multiple spring forces, body $\mathcal{B}^*$ moves and rotates, which can be simulated by the physical model of *rigid body dynamics* [18]. In this process, body $\mathcal{B}^*$ moves step by step toward the equilibrium where spring deformations are minimized. This is exactly the objective function of the problem in Definition 3 that minimizes the mean-squared error.

Although the heuristic of spring relaxation is also adopted by AFL [9], our solution contributes the following novelties. 1) From theoretical perspective, AFL is susceptible to trap in local minima, while our solution can alleviate such suboptimality problem, since our method is applied only to a small-scale problem with three unknowns. 2) From technical perspective, in AFL's model, the entities that are connected by springs are just particles without volumes. These particles can move, but they cannot rotate as our rigid bodies.

Pseudocode: We present in Algorithm 1 the pseudocode of our body merging optimization algorithm. Its output is the optimal transformation function $T_{p,R}^{\min}$. With $T_{p,R}^{\min}$, the node positions in body $\mathcal{B}^*$ can be converted to the coordinate frame of body $\mathcal{B}$, and body $\mathcal{B}^*$ can thus be merged into body $\mathcal{B}$. The inputs of Algorithm 1 are the initial guesses of translation p and rotation R and the constraint set $\mathcal{C}' = \{[p_l, p_l^*, d_l, w_l]\}$. Please check Fig. 4 for the illustration of notations.

First, line 1 assumes that the axis origin of body $\mathcal{B}^*$'s coordinate frame has been moved to the centroid of all the points $\{p_l^*\}$ that are contained by body $\mathcal{B}^*$ (i.e., white points in Fig. 4). The purpose of such moving of body $\mathcal{B}^*$ is to make translation p of body $\mathcal{B}^*$ independent from its rotation R. According to Euler's laws of motion, *for a rigid body, the translation of its centroid is independent from the rotation of this body about its centroid* [18]. This axis moving operation can be reversed after the termination of Algorithm 1.

Algorithm 1: GetOptimizedTransformation (3D compatible)

Input: $\mathcal{C}' = \{\ell'_l\} = \{\,[p_l,\, p^*_l,\, d_l,\, w_l]\,\}$; initial guess of p, R
Output: transformation $T^{\min}_{p,\,R}$ with mean squared-error e
1 **assert** centroid of $\mathcal{C}'$'s position set $\{p^*_l\}$ equals [0, 0, 0]
2 **repeat** // start from guess p, R at time $2t$
3 optimize guess p, R to guess $\_p$, $\_R$ at time $2t{+}1$
4 optimize guess $\_p$, $\_R$ to guess p, R at time $2t{+}2$
5 **until** *both* $\|p-\_p\|$ *and Frobenius norm* $\|R-\_R\|_F$ *are small*
6 **return** $T^{\min}_{p,\,R}$ with mean squared-error $e = f_{\mathcal{C}'}\,(T^{\min}_{p,\,R})$

Second, at lines 2–5, we use a loop to optimize the translation and rotation of body $\mathcal{B}^*$ iteratively. At the beginning of an iteration, we assume the current time is $2t$. At line 3, we optimize the current translation p and rotation R at time $2t$ to $\_p$, $\_R$ at time $2t{+}1$, and then at line 4, optimize $\_p$, $\_R$ to p, R at time $2t{+}2$. The loop terminates at line 5 if the changes from time $2t{+}1$ to time $2t{+}2$ are negligibly small. Note that both lines 3 and 4 need to optimize translation $p(t)$ and rotation $R(t)$ at time t to the next time-slot, which is described below. Note that we represent translation $p(t)$ by $3{\times}1$ vectors and represent rotation $R(t)$ by a $3{\times}3$ orthogonal matrix, for the ease of extension to 3-D spaces.

1) Translation Optimization: Our equations in Fig. 5 can optimize the translation $p(t)$ at time t to translation $p(t{+}1)$ at the next time. This optimization is directed by linear acceleration of body $\mathcal{B}^*$, which equals $F(t)/M$, where M is mass,k which is the number of nodes in body $\mathcal{B}^*$, and $F(t)$ is net force combining all forces acting on body $\mathcal{B}^*$. The net force requires the calculation of each individual force, e.g., the lth spring force $F_l(t)$. The direction of this force is a unit vector pointing from p_l to $T_{p(t),R(t)}(p^*_l)$. The magnitude of this force is the weighted error of this constraint, which is given in (2). Here, weight w_l within $\mathrm{err}_l(T_{p(t),R(t)})$ can be regarded as the spring constant to capture the different error characteristics in different constraints. The parameter δ when calculating $p(t{+}1)$ is the optimization step size. A rule of thumb is to configure it as $\delta{\in}[0.5a,\, a](a{=}M/\sum_{\ell'_l \in \mathcal{C}'} w_l)$ to balance between convergence speed and potential oscillations.

2) Rotation Optimization: In Fig. 5, rotation $R(t)$ at time t is optimized to $R(t{+}1)$ at the next time. This optimization is directed by angular acceleration of body $\mathcal{B}^*$, which equals $\omega(t) = I(t)^{-1}\tau(t)$, where $I(t)$ is inertial tensor of body $\mathcal{B}^*$ and $\tau(t)$ is torque acting on it. With this angular acceleration $\omega(t)$, the next orientation $R(t{+}1)$ can be calculated by multiplying Rodrigues' rotation formula $R_{\mathrm{rod}}(\phi,\theta)$ to $R(t)$

$$R_{\mathrm{rod}}(\phi,\theta){=}I{+}\sin\theta[\phi]_\times{+}(1{-}\cos\theta)[\phi]^2_\times$$

where we can set $\phi{=}\omega(t)/\|\omega(t)\|$ and $\theta{=}\delta\|\omega(t)\|$. The meaning of left multiplying $R(t)$ with $R_{\mathrm{rod}}(\phi,\theta)$ is to rotate $R(t)$ around 3-D unit vector ϕ by angle θ. Here, the operator $[\cdot]_\times$ is to convert vector a to its skew-symmetric matrix $[a]_\times$. This operator can be used to transform vector cross product to matrix multiplication, i.e., $a{\times}b{=}[a]_\times b$

$$\text{Given vector } a{=}\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix},\quad [a]_\times \overset{\mathrm{def}}{=} \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Although these equations are defined in 3-D, they can easily handle the cases in 2-D by setting z value of each position to zero and by setting each orientation vertical to z-axis.

When implementing the equations in Fig. 5, an issue that one needs to be cautious about is that our rotation representation by 3×3 rotation matrices has the well-known problem of numerical drift [18]. That is, after calling equation $R(t{+}1) = R_{\mathrm{rod}}(\phi,\theta)R(t)$ for tens of iterations, $R(t{+}1)$ may differ from an orthogonal matrix. A mitigation is to replace $R(t{+}1)$ by its nearest orthogonal matrix, e.g., for every 20 iterations. Another point that needs to take care of is that the denominators in Fig. 5 can equal to zero occasionally. Thus, the protection code for such boundary conditions needs to be added.

C. Two Technical Issues Toward Robust Implementation

Conquer Local Minima Problem: Our body merging optimization algorithm is essentially a greedy optimization algorithm using the heuristic of linear acceleration and angular acceleration of body $\mathcal{B}^*$. However, any heuristic algorithm that is applied to nonlinear optimization would have the problem of trapping in local minima with a nonnegligible possibility. (Note: Trilateration and patch stitching also have the problem.)

We therefore adopt the following strategy to address the problem of local minima. We first generate multiple initial guesses p, R for the optimal transformation function $T^{\min}_{p,R}$. We then feed these guesses separately into Algorithm 1. When all the optimization terminates, we select the best solution with the minimum mean-squared error as defined in (3). The key issue that remains is how to generate a set of good-quality initial guesses p, R, which is briefly described as follows.

1) Guesses of Rotation R in $T^{\min}_{p,R}$: We generate rotation R as $R{=}\begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$, where γ is Euler angle that falls randomly into $(-\pi, \pi]$. Half of the generated guesses of rotation matrix R will be multiplied with a reflection matrix.

2) Guesses of Translation p in $T^{\min}_{p,R}$: We generate translation p by the following bounding box algorithm. First, from Fig. 4, we know that the physical meaning of translation p is the position of body $\mathcal{B}^*$'s centroid in body $\mathcal{B}$'s coordinate frame. Thus, if we can find a bounding box that contains all the possible positions of body $\mathcal{B}^*$'s centroid, then we can generate the initial guesses of translation p randomly inside the box. We can find such a bounding box, for each constraint that connects the two bodies, as shown in Fig. 6. The overlapped box of all the bounding boxes can be used to generate the guesses of p.

The bounding box in Fig. 6 corresponds to the constraint whose two vertices are p_l, p^*_l and whose length is d_l. We denote the centroid of body $\mathcal{B}^*$ as $\overline{p^*}$. Note that position p_l is defined in body $\mathcal{B}$'s coordinate frame, but the positions p^*_l and centroid $\overline{p^*}$ are defined in body $\mathcal{B}^*$'s coordinate frame. Then, we know that this centroid $\overline{p^*}$ must be included in the bounding box centered at position p_l with its radius to be $d_l{+}\|p^*_l{-}\overline{p^*}\|{+}2c\sigma/w_l$, where $\|p^*_l{-}\overline{p^*}\|$ is distance between position p^*_l and the centroid $\overline{p^*}$ and $2c\sigma/w_l$ is the upper bound of measurement error of constraint length d_l.

Mitigate Error Accumulation by Refinement Step: A common problem exists for various iterative localization methods, called *error accumulation*. It means that for each time we merge two bodies, we introduce a small amount of error into the merged body. Such error due to body merging can accumulate, as the iterative merging process continues. This accumulation will cause the increase of location error as the size of bodies grows. However, our body merging optimization problem in

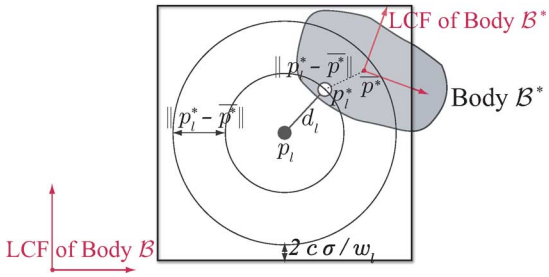


Fig. 6. Bounding box for centroid $\overline{p^*}$ of body $\mathcal{B}^*$.

Definition 3 neglects such location noise, i.e., the inaccuracy in position estimates of p_l and p_l^* within the two bodies $\mathcal{B}$ and $\mathcal{B}^*$. It focuses only on tolerating the ranging noise in d_l.

To address this error accumulation problem, we propose an additional refinement step that executes each time after merging two bodies by Algorithm 1. This refinement step is similar to the AFL method [9]. It treats the resultant body $\mathcal{B}^+$ merged from $\mathcal{B}$ and $\mathcal{B}^*$ as a network of nodes connected by springs and relaxes these springs to fine-tune the node positions (see line 2 and 3 of Algorithm 2). By relaxing the springs, the location errors of nodes within a body can be redistributed evenly.

Algorithm 2: MergeTwoBodiesOptimally

Input: Bodies $\mathcal{B}$,$\mathcal{B}^*$; Constraint Set $\mathcal{C}=\{\ell_l\}=\{[n_l, n_l^*, d_l, w_l]\}$
Output: Body $\mathcal{B}^+$ by merging body $\mathcal{B}^*$ into body $\mathcal{B}$
1 Use Algorithm 1 to get the optimal transformation function $T_{p,R}^{\min}$ from $\mathcal{B}^*$'s coordinate frame to $\mathcal{B}$'s coordinate frame
2 Get the resultant body $\mathcal{B}^+$ by merging the nodes in $\mathcal{B}$ with the nodes in $\mathcal{B}^*$ whose coordinates are transformed by $T_{p,R}^{\min}$
3 Refine this body $\mathcal{B}^+$ by AFL spring relaxation [9]
4 Verify $\mathcal{B}^+$ by testing whether the error of each edge is below a threshold (see (4)), and if it fails, return $\emptyset$

However, the refinement step alone cannot solve the error accumulation problem. The ultimate solution is the dense deployment of anchor nodes, whose location knowledge contains negligible noise as compared to the ranging noise. We have modified the refinement step to utilize such valuable anchor information. Initially, all the anchors are contained in the global body. Then, in the merging phase, when we merge global body $\mathcal{B}$ with a local body $\mathcal{B}^*$ to get the resultant body $\mathcal{B}^+$, body $\mathcal{B}^+$ will become the new global body that contains the anchors. The refinement step, when applied to such $\mathcal{B}^+$, will not modify the anchor positions. Thus, other nodes, by relaxing their springs to these anchors, can adjust their location estimates to reduce noise and tame error accumulation.

To better utilize the deployed anchors, we make an adjustment to the merging order of iterative merging process.

- When multiple pairs of bodies are available that can satisfy body merging condition DOC $\geq$ DOF, we prefer the body pairs that merge the global body with a local body.

There are two reasons for using this rule. One is that the anchor nodes within global body can help tame error accumulation. The other reason is that the anchor nodes can help reject incorrect ambiguities during body merging. Note that line 4 of Algorithm 2 checks whether the body $\mathcal{B}^+$ after refinement can satisfy the restriction that the error of each edge is below a threshold. If this verification fails, Algorithm 2 will indicate the failure to merge the two bodies $\mathcal{B}$ and $\mathcal{B}^*$ by returning an empty node set. The topic of ambiguities enumeration and rejection will be discussed in detail in Section V.

IV. Unique Body Merging Conditions

In this section, we describe when we can apply the merging primitive to uniquely merge two bodies. There are two necessary conditions. One is the redundancy in constraints between two bodies. The other is the noncollinear geometry of these constraints. Our conditions can unify the previous works based on global rigidity [11]–[13], and meanwhile consider the noncollinearity that is not fully explored before [1], i.e., either vertices of constraints can be collinear.

A. Unique Body Merging Condition: Redundant Constraints

For the unique merging of two bodies, a necessary condition is that, in body $\mathcal{B}$'s coordinate frame ($\mathcal{B}$ contains at least three nodes), the free variables of body $\mathcal{B}^*$ are redundantly constrained. We formulate this condition as follows:

$$\text{Degree-of-Constraint (DOC)} > \text{Degree-of-Freedom (DOF)}.$$

- DOF Calculation: If body $\mathcal{B}^*$ contains only one node, DOF $= 2$; otherwise, DOF $= 3$, due to the rotation of $\mathcal{B}^*$.
- DOC Calculation: In constraint set $\mathcal{C}$, a shared node contributes two DOC, and a connecting link contributes one DOC. Sum all contributions to get the overall DOC.

This condition can unify the previous work that localizes individual nodes and patches, including trilateration, collaborative multilateration [11], patch stitching [12], and CALL [13]. We explain this condition case by case as follows.

Case (a): Trilateration can be considered as an operation that merges body $\mathcal{B}^*$ containing only one node into body $\mathcal{B}$. As shown in Fig. 7(a), body $\mathcal{B}$ contains three nodes 1, 2, 3. Body $\mathcal{B}^*$ contains only node 0. Body $\mathcal{B}^*$ has two DOF in the coordinate frame of body $\mathcal{B}$ because node 0 can move in two directions, i.e., x- and y-directions. Trilateration requires at least three links connecting body $\mathcal{B}^*$ to body $\mathcal{B}$ because the three links can provide three DOC to redundantly constrain the two DOF of body $\mathcal{B}^*$. Otherwise, as shown in Fig. 8(a), node 0 can have an ambiguous position assignment $0'$, if there are just two links $[0, 1]$, $[0, 2]$ providing two DOC.

Case (b): Collaborative multilateration [11] can be considered as an operation that merges body $\mathcal{B}^*$ containing two nodes into body $\mathcal{B}$. As shown in Fig. 7(b), body $\mathcal{B}$ contains three nodes 1, 2, 4, and body $\mathcal{B}^*$ contains two nodes 0, 3. Because body $\mathcal{B}^*$ contains more than one node, it has three DOF with two translations and one rotation. Collaborative multilateration requires four links connecting body $\mathcal{B}^*$ to body $\mathcal{B}$, which can provide four DOC and redundantly constrain the three DOF. Otherwise, as shown in Fig. 8(b), the body $\mathcal{B}^*$ can have four spatial poses, i.e., $[0, 3]$, $[0, 3']$, $[0', 3'']$, and $[0', 3''']$, if there are just three links providing three DOC.

Case (c): Patch stitching [12] is an operation that merges body $\mathcal{B}^*$ containing at least three nodes into body $\mathcal{B}$. As shown in Fig. 7(c), body $\mathcal{B}$ contains three nodes 1, 2, 3, and body $\mathcal{B}^*$ is drawn as a gray block. Patch stitching requires body $\mathcal{B}$ and

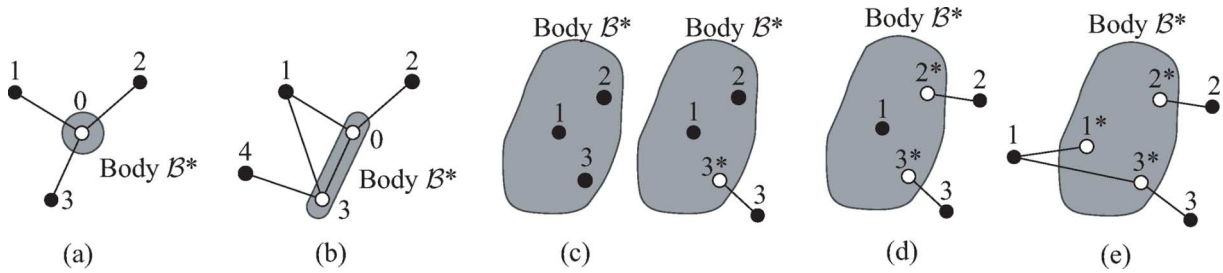


Fig. 7. Unique merging of two bodies whose constraints are redundant by satisfying the condition DOC $>$ DOF. (a) Node merging: DOC $= 3$, DOF $= 2$. (b) Link merging: DOC $= 4$, DOF $= 3$. (c) Patch merging I: DOC $= 6$ or 5, DOF $= 3$. (d) Patch merging II: DOC $= 4$, DOF $= 3$. (e) Patch merging III: DOC $= 4$, DOF $= 3$.

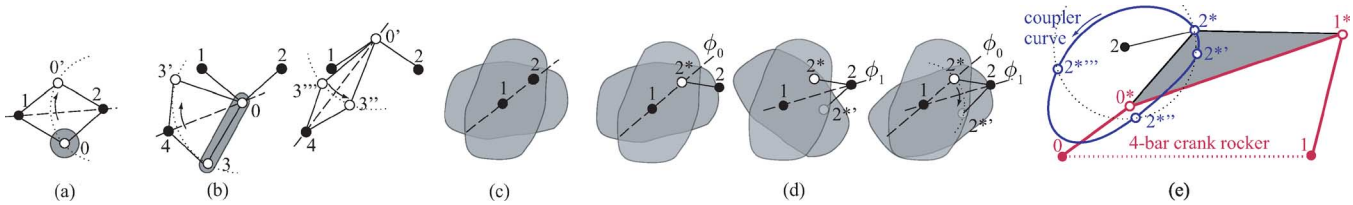


Fig. 8. Ambiguous body merging with finite ambiguities due to nonredundant constraints that only satisfy DOC$\geq$DOF. (a) Node merging: DOC $= 2$, DOF $= 2$. (b) Link merging: DOC $= 3$, DOF $= 3$. (c) Patch merging I: DOC $= 4$, DOF $= 3$. (d) Patch merging II: DOC $= 3$, DOF $= 3$. (e) Patch merging III: DOC $= 3$, DOF $= 3$.

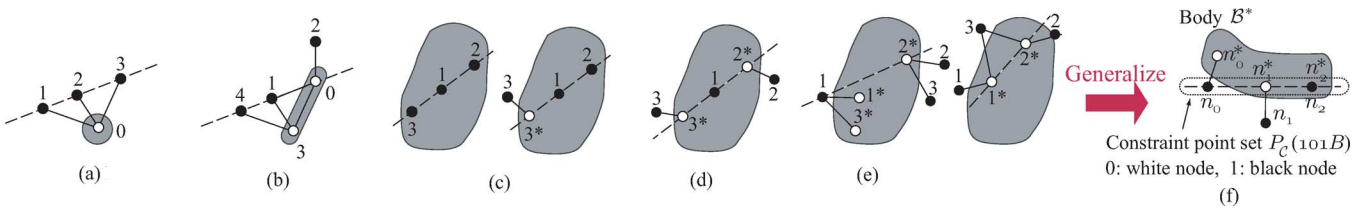


Fig. 9. Ambiguous body merging due to constraints collinearity. (a) Node merging: DOC $= 3$, DOF $= 2$. (b) Link merging: DOC $= 4$, DOF $= 3$. (c) Patch merging I: DOC $= 6$ or 5, DOF $= 3$. (d) Patch merging II: DOC $= 4$, DOF $= 3$. (e) Patch merging III: DOC $= 4$, DOF $= 3$. (f) Constraints collinearity. (f) depicts the general case that a line passes through the nodes of each constraint. Thus, body $\mathcal{B}^*$ can flip across the depicted lines and preserve constraint length.

$\mathcal{B}^*$ to share at least three nodes 1, 2, 3 to fix body $\mathcal{B}^*$. Different from a link that can provide one DOC, each shared node can provide two DOC.

Cases (d) and (e): CALL [13] proposed two other conditions that two patches can be merged uniquely, which are shown in Fig. 7(d) and (e). Fig. 7(d) depicts the condition of one shared node and two connecting links, which can provide four DOC. Fig. 7(e) shows the condition of four connecting links, which can also provide four DOC. If DOC is not redundant and is equal to DOF, then body $\mathcal{B}^*$ can have multiple spatial poses. In Fig. 8(d), body $\mathcal{B}^*$ has four ambiguities since it can be flipped across the depicted lines. Fig. 8(e) also shows that, when body $\mathcal{B}^*$ is connected to body $\mathcal{B}$ by only three links $[0, 0^*]$, $[1, 1^*]$, $[2, 2^*]$, it can have four ambiguities. In fact, the upper bound for the number of ambiguities for the case in Fig. 8(e) is 12, which will be proved in Section V-A.

B. Unique Body Merging Condition: Geometry of Constraints

Motivation: It is well known that, for multilateration, the three nodes 1, 2, 3 in Fig. 7(a) must be noncollinear. Otherwise, as shown in Fig. 9(a), the node 0 can flip across the depicted line without changing the length of each constraint. However, such noncollinearity checking is inadequate when applied to body merging. This is because multilateration considers the collinearity of only one end of the constraints (i.e., black nodes). However, it is possible that either ends (i.e., black nodes and white nodes) are collinear during body merging. For example, in Fig. 9(b), the nodes 0, 1, 4 are collinear, and thus node 3 can flip across the depicted line; in Fig. 9(c), the body $\mathcal{B}^*$ can flip across the line through the collinear nodes $1, 2, 3^*$.

For unique body merging, we propose a necessary condition that fully considers the collinear geometry of constraint set $\mathcal{C}$. The basic idea is that there does not exist a line to pass through one vertex of each constraint. Otherwise, body $\mathcal{B}^*$ can flip across this line and has ambiguous realizations as shown in Fig. 9. To describe the point set containing a vertex of each constraint, we define the concept of *constraint point set*.

Definition 4 (Constraint Point Set $P_{\mathcal{C}}(k)$): Assume that $\mathcal{C} = \{\ell_l\} = \{[n_l, n_l^*, d_l, w_l]\}$ is the constraint set during the merging of body $\mathcal{B}^*$ into body $\mathcal{B}$, as shown in Fig. 7(f). The constraint point set $P_{\mathcal{C}}(k)$ corresponding to $\mathcal{C}$ contains one of the nodes of each constraint ℓ_l ($\in\mathcal{C}$). The integer k, which is between 0 and $2^{|\mathcal{C}|}$ (exclusive), can indicate which node (n_l or n_l^*) is contained by $P_{\mathcal{C}}(k)$. If the lth bit of the integer k is 1, point set $P_{\mathcal{C}}(k)$ contains the node n_l in constraint ℓ_l; otherwise, point set $P_{\mathcal{C}}(k)$ contains the node n_l^* in constraint ℓ_l.

From the above definition, we know that the constraint point set is denoted by $P_{\mathcal{C}}(k)$, where integer k is a bitmap showing which end of each constraint will be included. For example, in Fig. 9(f), constraint set $\mathcal{C}$ contains three constraints: One of them is a shared node n_2 (equal to n_2^*), and the other two are links $[n_0, n_0^*]$, $[n_1, n_1^*]$. Then, the constraint point set $P_{\mathcal{C}}(k)$ with k=$101B$ will contain three node indices n_2, n_1^*, n_0 since only the middle digit is zero. Since binary $101B$ equals decimal 5, we often rewrite $P_{\mathcal{C}}(101B)$ as $P_{\mathcal{C}}(5)$.

With the concept of constraint point set, we can present a necessary condition for the unique merging of two bodies in

Theorem 1. This condition fully considers the collinearity of different constraint point sets by varying k value, which is illustrated in Fig. 9(f) as black or white nodes combinations.

Theorem 1 (Constraint Set Noncollinearity): For the unique merging of body $\mathcal{B}^*$ into body $\mathcal{B}$, a necessary condition is the noncollinearity of constraint set $\mathcal{C}$ defined as follows.

- When body $\mathcal{B}^*$ contains one node, constraint point set $P_{\mathcal{C}}(k)$ is noncollinear with $k = 2^{|\mathcal{C}|} - 1$ (or $11\ldots1B$).
- When body $\mathcal{B}^*$ contains two nodes, constraint point set $P_{\mathcal{C}}(k)$ is noncollinear for each $k \in [1, 2^{|\mathcal{C}|} - 1]$.
- When body $\mathcal{B}^*$ contains at least three nodes, constraint point set $P_{\mathcal{C}}(k)$ is noncollinear for each $k \in [2^{|\mathcal{C}|} - 2^{|\mathcal{C}|-m}, 2^{|\mathcal{C}|} - 1]$, where m is the number of shared nodes, and these nodes are placed at the end of constraint list $\mathcal{C}$.

Proof: When body $\mathcal{B}^*$ contains one node as depicted in Fig. 9(a), the three black nodes (noted as $P_{\mathcal{C}}(2^{|\mathcal{C}|}-1)$) should be noncollinear for unique merging. When body $\mathcal{B}^*$ contains two nodes as shown in Fig. 9(b), constraint sets $P_{\mathcal{C}}(k)$ with $k\in[1, 2^{|\mathcal{C}|}-1]$ must be noncollinear for unique merging. However, the two white nodes (noted as $P_{\mathcal{C}}(0)$) can be collinear because the flipping of body $\mathcal{B}^*$ across the line through $P_{\mathcal{C}}(0)$ cannot change the locations of the two nodes in $\mathcal{B}^*$. When $\mathcal{B}^*$ contains at least three nodes as depicted in Fig. 9(c)–(e), any sets $P_{\mathcal{C}}(k)$ with $k\in[2^{|\mathcal{C}|} - 2^{|\mathcal{C}|-m}, 2^{|\mathcal{C}|}-1]$ must be noncollinear. We do not need to check the collinearity for $k < 2^{|\mathcal{C}|} - 2^{|\mathcal{C}|-m}$ because we want to set the higher m bits to 1, which correspond to the m shared nodes between the two bodies. ■

Collinearity Testing: A key issue that arises from Theorem 1 is how to test the collinearity of a constraint point set $P_{\mathcal{C}}(k)$. However, the challenge is the interference of ranging noise, which can make constraint point set $P_{\mathcal{C}}(k)$ roughly collinear, as shown in Fig. 10. To test such rough collinearity, we propose the following method with two steps.

- First, we obtain a line ϕ fitting the constraint point set $P_{\mathcal{C}}(k)$ by orthogonal regression, as shown in Fig. 10.
- Second, we test whether the distance from each point to line ϕ is within a threshold that relates with ranging noise σ.

We use an example to show how we apply orthogonal regression to obtain a line ϕ that best fits a constraint point set $P_{\mathcal{C}}(k)$. The difficulty is that the node positions of $P_{\mathcal{C}}(k)$ may not be in the same coordinate frame. As shown in Fig. 10, the constraint point set $P_{\mathcal{C}}(k)$ contains seven node indices. Four of them $\{n_0, n_3, n_4, n_5\}$ are black nodes that are contained in body $\mathcal{B}$. Three of them $\{n_1^*, n_2^*, n_6^*\}$ are white nodes that are contained in body $\mathcal{B}^*$. Before applying orthogonal regression, we need to unify them in the same coordinate frame. This is achieved by our body merging optimization algorithm presented in Section III. But note that, if there exist flip ambiguities due to collinearity, this algorithm can obtain only one of possibilities to align their coordinate frames. After this merging, we can apply the following orthogonal regression algorithm to obtain the best-fit line ϕ.

Input: a set of data points $\{[x_l,\ y_l,\ w_l]\}$ in the same coordinate frame, where w_l is the weight of point x_l, y_l
Output: fitted line ϕ: $Ax + By + C = 0$ that minimizes the orthogonal distances from the data points to line ϕ

$\bar{x} = \sum w_l^2 x_l \,/\, \sum w_l^2$; $\bar{y} = \sum w_l^2 y_l \,/\, \sum w_l^2$

$x'_l = x_l - \bar{x}$; $y'_l = y_l - \bar{y}$; $W = \begin{bmatrix} \sum w_l^2 x'_l x'_l & \sum w_l^2 x'_l y'_l \\ \sum w_l^2 y'_l x'_l & \sum w_l^2 y'_l y'_l \end{bmatrix}$

Coefficients $[A, B]^{\mathrm{T}}$ is equal to the eigen vector of matrix W that has the minimum eigenvalue, and $C = -A\,\bar{x} - B\,\bar{y}$

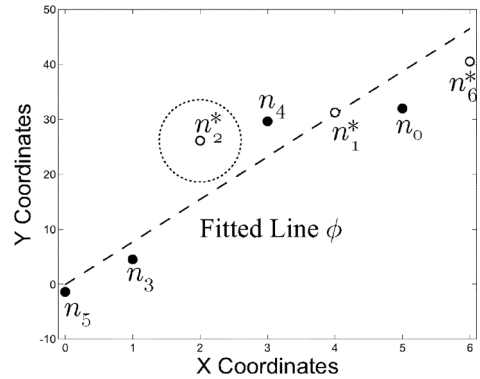


Fig. 10. Test collinearity of constraint point set $P_{\mathcal{C}}(k)$.

The second step is, with the best-fit line ϕ, to check whether the distance from each constraint vertex in $P_{\mathcal{C}}(k)$ to line ϕ is smaller than a threshold $2c\sigma/w_l$, where σ/w_l is the measurement noise of constraint ℓ_l, and c is an adjustable constant. If so, then it is a clear sign for rough collinearity of constraint point set $P_{\mathcal{C}}(k)$. Otherwise, the distance from some vertex (e.g., node n_2^* in Fig. 10) to the best-fit line ϕ must be larger than the threshold. Thus, the constraint point set is regarded as noncollinear. Only when there does not exist any collinearity as required by Theorem 1 can we merge two bodies uniquely.

V. Body Merging With Finite Ambiguities

In sparse networks, the condition for unique body merging may not always be satisfied, i.e., rule of redundant constraints DOC>DOF. This section will relax this condition and considers *non-underdetermined* constraints DOC≥DOF In such situations, there may exist multiple ambiguities to merge two bodies, which is shown in Fig. 8. In this section, we first prove that there exist only finite ambiguities if given non-underdetermined constraints between two bodies. We then present an algorithm to enumerate flip ambiguities, which works well in sparse networks and ensures location robustness.

We call our localization algorithm *Iterative Inflexible Body Merging* (IIBM), which permits finite ambiguities for body merging. Here, a body is called an "inflexible body" because it can have multiple but finite ambiguities. In contrast, a rigid body can have only one ambiguity, and a flexible body has a flexible part that can move and produce infinite ambiguities.

The idea of finite ambiguities is also adopted by the state-of-the-art SWEEPS [14] and CALL [13]. However, our algorithm still makes the following contributions.

1) Better robustness against collinear constraints: There are various cases of collinear geometry that can produce finite ambiguities as shown in Fig. 9. We can handle all the cases uniformly by the algorithm to present in Section V-B.
2) Higher percentage of localizable nodes: We equally apply all the DOC≥DOF cases shown in Figs. 7 and 8 to:
   - the merging of a local body into the global body;
   - the merging of a local body into another local body.

   In contrast, CALL [13] does not use the ambiguous cases in Fig. 8(c) and (d) for the merging of two local bodies.
3) Tighter upper bound for ambiguities: We derive an upper bound for the number of ambiguities when the two bodies are connected by only three links, as depicted in Fig. 8(e). The upper bound of 12 we derived (see Theorem 5) is much tighter than the bound of 24 provided by CALL [13].

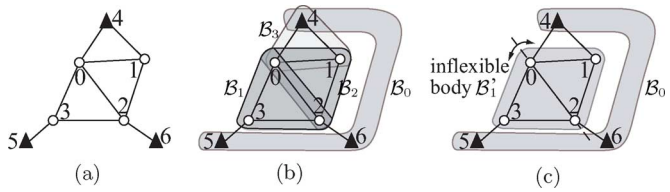


Fig. 11. Example of iterative inflexible body merging. (a) Input network. (b) Network division. (c) Merge body $\mathcal{B}_1'$ into $\mathcal{B}_0$.

Example: We use an example to explain how the IIBM algorithm localizes the sparse network depicted in Fig. 11(a).

First, the network topology is split into a list of bodies. As shown in Fig. 11(b), the body list includes the following:

- the global body $\mathcal{B}_0$ containing the three GCF anchors 4, 5, 6 with known coordinates in global coordinate frame;
- a local body $\mathcal{B}_1$ corresponding to the triangle 0, 2, 3;
- a local body $\mathcal{B}_2$ corresponding to the triangle 0, 1, 2;
- a local body $\mathcal{B}_3$ corresponding to the triangle 0, 1, 4.

Second, merge the bodies iteratively by the following steps. Since none of the local bodies $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ have enough connections to global body $\mathcal{B}_0$ that can satisfy $\mathrm{DOC}\geq\mathrm{DOF}$, we can only select two local bodies and merge them. The local bodies $\mathcal{B}_1$ and $\mathcal{B}_2$ share two nodes 0 and 2 as shown in Fig. 11(b), which can satisfy $\mathrm{DOC}\geq\mathrm{DOF}$ and corresponds to the ambiguous merging case in Fig. 8(c). This merging can generate the inflexible body $\mathcal{B}_1'$ in Fig. 11(c) with two ambiguities since it can be folded across the line through nodes 0, 2. Now, the new inflexible body $\mathcal{B}_1'$ and the global body $\mathcal{B}_0$ have four connecting links, which can satisfy $\mathrm{DOC} > \mathrm{DOF}$ and corresponds to the unique merging case in Fig. 7(e). Thus, body $\mathcal{B}_1'$ can be localized uniquely in GCF. Note that the wrong ambiguity of $\mathcal{B}_1'$ (i.e., nodes 1, 3 on the same side relative to line 0, 2) can be rejected during this merging. This is because the wrong ambiguity cannot be aligned with the global body $\mathcal{B}_0$. Specifically speaking, no matter how we adjust the spatial pose of the wrong ambiguity of body $\mathcal{B}_1'$ by Algorithm 1, it is impossible for us to reduce the error of all constraints below a defined threshold (see line 4 of Algorithm 4 that returns $\emptyset$).

A. Condition of Non-Underdetermined Constraints

In this section, we prove the theoretical foundation of inflexible body merging, i.e., when merging body $\mathcal{B}$ with body $\mathcal{B}^*$, there exist only finite merging ambiguities, if the constraint set $\mathcal{C}$ is non-underdetermined and satisfies $\mathrm{DOC}\geq\mathrm{DOF}$. Our proof is based on solving quadratic equation systems, and we find that the maximum number of ambiguities is 12.

Equation System: We present the governing equations in (5), which have three unknowns in 2-D space. Two unknowns are contained in translation p, and one unknown is in rotation R. For this parallel equation set, its real-number solutions are the ambiguities that the two bodies can be aligned. Note that when body $\mathcal{B}^*$ contains only one node (i.e., all nodes in $\mathcal{B}^*$ are equal to zero $p_l^*\equiv\mathbf{0}$), (5) can degrade to multilateration and lose its variable R. Thus, for the following analysis, we assume the nodes p_l^* do not overlap, which means at least one of $\|p_{l_1}^*-p_{l_2}^*\|(0\leq l_1<l_2<n)$ is nonzero.

Definition 5 (Governing Equations of Body Merging Problem): During the merging of body $\mathcal{B}^*$ into body $\mathcal{B}$, we have the constraint set $\mathcal{C}'=\{\ell_l'\}=\{[p_l, p_l^*, d_l, w_l]\}$ as defined in Definition 3. The purpose of merging is to find all the real-number solutions of the following equation system:

$$d_l^2 = \|(p+Rp_l^*)-p_l\|^2 (0 \leq l < |\mathcal{C}'|). \tag{5}$$

For simplicity, the above equation omits the reflection of body $\mathcal{B}^*$, which can later be implemented by left multiplying each position p_l^* of body $\mathcal{B}^*$ by reflection matrix $\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$.

Lemma 2: We can convert (5) to the following equation system about translation p and rotational tangent value t:

$$\mathbf{0} = [\,A_l+B_l \quad 2C_l \quad A_l-B_l\,][\,t^0 \quad t^1 \quad t^2\,]^{\mathrm{T}} \tag{6}$$

where t is equal to $\tan\frac{\gamma}{2}$ and γ is the rotational angle of the rotation matrix R. Symbols A_l, B_l, C_l are defined as follows:

$$A_l = \|p\|^2-2p_l\cdot p+\|p_l\|^2+\|p_l^*\|^2-d_l^2$$
$$B_l = 2p_l^*\cdot p-2p_l^*\cdot p_l \qquad C_l=2\overset{\curvearrowright}{p}_l^*\cdot p-2\overset{\curvearrowright}{p}_l^*\cdot p_l$$

where $\overset{\curvearrowright}{p}_l^* = \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} p_l^*$. Note that $p_l^*\cdot\overset{\curvearrowright}{p}_l^*\equiv 0$, $\|p_l^*\|\equiv\|\overset{\curvearrowright}{p}_l^*\|$.

Proof: First, we can substitute rotation R in (5) by rotation matrix with angle γ

$$\begin{aligned} d_l^2 &= \left\|\begin{bmatrix}\cos\gamma & \sin\gamma\\ -\sin\gamma & \cos\gamma\end{bmatrix} p_l^*+(p-p_l)\right\|^2\\ &= \left\|\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} p_l^*\cos\gamma+\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} p_l^*\sin\gamma+(p-p_l)\right\|^2\\ &= \left\|p_l^*\cos\gamma+\overset{\curvearrowright}{p}_l^*\sin\gamma+(p-p_l)\right\|^2\\ &= \|p_l^*\|^2+\|p-p_l\|^2+2p_l^*\cdot(p-p_l)\cos\gamma+2\overset{\curvearrowright}{p}_l^*\cdot(p-p_l)\sin\gamma. \end{aligned}$$

Simplification is achieved by introducing symbols A_l, B_l, C_l.

$$0 = A_l+B_l\cos\gamma+C_l\sin\gamma.$$

Then, the above equations can be transformed by tangent half-angle formulas to a polynomial system about $t=\tan\frac{\gamma}{2}$.

Root Finding: By solving the equation system in (6), we can give out the upper bounds of the number of ambiguities for different cases satisfying $\mathrm{DOC}\geq\mathrm{DOF}$. Theorem 3 is for the case in Fig. 8(d). Theorems 4 and 5 are for the case in Fig. 8(e). We do not need to prove the upper bounds for the cases in Fig. 8(a)–(c) because they correspond to bilateration and patch stitching and have been proved by previous work.

Theorem 3: There are at most four ambiguities during the merging of body $\mathcal{B}^*$ into body $\mathcal{B}$, if:

- they can satisfy the condition of $\mathrm{DOC}\geq\mathrm{DOF}$;
- they share at least one node as shown in Fig. 8(d).

Proof: The following two parallel equations can be established from the two constraints in Fig. 8(d):

$$0 = d_0^2=\|p-p_0\|^2 \qquad (\text{with } d_0=0 \text{ and } p_0^*=\mathbf{0})$$
$$0 = [\,A_1+B_1 \quad 2C_1 \quad A_1-B_1\,][\,t^0 \quad t^1 \quad t^2\,]^{\mathrm{T}}.$$

The first equation corresponds to the zero-length link with $d_0=0$, and it is additionally assumed $p_0^*=0$, which can be realized by a translation of body $\mathcal{B}^*$'s coordinate frame. From the first equation, it can be known that translation p is equal to p_0. From the second equation that follows (6), two roots can be

derived for variable t since p is already known and the second equation is quadratic in t. With a known t, we have only one γ, and thus only one rotation R, since $t=\tan\frac{\gamma}{2}$. Considering the reflection of body $\mathcal{B}^*$, which can double the number of ambiguities, variables p, R can have at most four solutions, and thus Fig. 8(d) can have four ambiguities. ■

Theorem 4: There are at most eight ambiguities during the merging of body $\mathcal{B}^*$ into body $\mathcal{B}$ if:

- they can satisfy the condition of $\mathrm{DOC}\geq\mathrm{DOF}$;
- they do not share a node as shown in Fig. 8(e);
- a node of body $\mathcal{B}^*$ is incident to two links, e.g., $p_0^*=p_1^*$.

Proof: The following two parallel equations can be established from the three links in Fig. 8(e), however with two links $l=0,1$ incident to a same node in $\mathcal{B}^*$, i.e., $p_0^*=p_1^*$:

$$\begin{aligned} d_l^2 &= \|p-p_l\|^2 \qquad (\text{with } l=0 \text{ or } 1 \text{ and } p_0^*=p_1^*=\mathbf{0}) \\ 0 &= [\,A_2+B_2 \quad 2C_2 \quad A_2-B_2\,]\,[\,t^0 \quad t^1 \quad t^2\,]^{\mathrm{T}}. \end{aligned}$$

The first equation corresponds to these two links with $p_0^*=p_1^*=\mathbf{0}$, which can be realized by moving the original point of body $\mathcal{B}^*$ to this shared sensor. The first equation has two roots for p since two circles have two intersection points. The second equation directly follows (6), by which a known p, we can derive two roots for rotation R. Considering the reflection of body $\mathcal{B}^*$, $[p, R]$ can have eight ambiguities. ■

Theorem 5: There are at most 12 ambiguities during the merging of body $\mathcal{B}^*$ into body $\mathcal{B}$ if:

- they can satisfy the condition of $\mathrm{DOC}\geq\mathrm{DOF}$;
- they are connected by three links as shown in Fig. 8(e).

Proof: The following derivations assume that body $\mathcal{B}$ contains at least three nodes and the number of links n equals 3, to make it a determined system with $\mathrm{DOC}=\mathrm{DOF}$, which is shown in Fig. 8(e). It is also assumed that there are no zero-length links $\forall l\in[0,n), d_l\neq 0$, and there do not exist two links sharing a sensor $\forall l_1 l_2\in[0,n), l_1\neq l_2\rightarrow p_{l_1}^*\neq p_{l_2}^*$.

Before the elimination of variable t in (6), note that B_l, C_l are linear in p, and A_l is quadratic in p. The next step therefore is to reduce the degree of A_l by eliminating the quadratic term $\|p\|^2$; otherwise, the resultant, after the elimination variable t in (6), can have a high degree in p.

First, it is assumed that in (5) p_0^* equals zero, which can be realized by a temporary translation of body $\mathcal{B}^*$'s coordinate frame, and thus the following equation can be established:

$$d_0^2 = \|p-p_0\|^2=\|p\|^2-2p_0\cdot p+\|p_0\|^2. \tag{7}$$

Therefore, the degree of A_l $(l=1,2)$ can be reduced to one in (8) by substituting $\|p\|^2$ with $d_0^2+2p_0\cdot p-\|p_0\|^2$

$$\begin{aligned} \mathbf{0} &= \begin{bmatrix} A_1+B_1 & 2C_1 & A_1-B_1 \\ A_2+B_2 & 2C_2 & A_2-B_2 \end{bmatrix} [\,t^0 \quad t^1 \quad t^2\,]^{\mathrm{T}} \\ A_l &= -2(p_l-p_0)\cdot p+(\|p_l\|^2-\|p_0\|^2)+\|p_l^*\|^2-(d_l^2-d_0^2) \\ B_l &= 2p_l^*\cdot p-2p_l^*\cdot p_l \quad C_l=2\overset{\curvearrowright}{p}_l^*\cdot p-2\overset{\curvearrowright}{p}_l^*\cdot p_l. \end{aligned} \tag{8}$$

By elimination of variable t in (8), (9) is established. The elimination technique used is linear elimination since the sufficient and necessary condition for the two polynomials in (8) to have a common root for variable t is that the determinant of their associated Sylvester matrix M should vanish

$$\begin{aligned} 0 &= \det(M) \\ M &= \begin{bmatrix} A_1+B_1 & 2C_1 & A_1-B_1 & 0 \\ 0 & A_1+B_1 & 2C_1 & A_1-B_1 \\ A_2+B_2 & 2C_2 & A_2-B_2 & 0 \\ 0 & A_2+B_2 & 2C_2 & A_2-B_2 \end{bmatrix}. \end{aligned} \tag{9}$$

This common root, by eliminating t's quadratic term, is

$$t=\frac{(A_1-B_1)(A_2+B_2)-(A_1+B_1)(A_2-B_2)}{2C_1(A_2-B_2)-(A_1-B_1)2C_2}. \tag{10}$$

The polynomial without t in (9) is quartic in p because each row of M is linear in p. We, however, can further reduce (9) to a cubic polynomial by substitutions of $\|p\|^2$ with $d_0^2+2p_0\cdot p-\|p_0\|^2$. Equation (9) can be expanded to

$$\begin{aligned} 0 &= (B_1C_2-C_1B_2)^2+2A_1A_2(B_1B_2+C_1C_2) \\ &\quad -A_1^2(B_2^2+C_2^2)-A_2^2(B_1^2+C_1^2). \end{aligned} \tag{11}$$

The following analysis shows that (11) is cubic in p. With $(p_1^*\cdot p)(\overset{\curvearrowright}{p}_2^*\cdot p)-(\overset{\curvearrowright}{p}_1^*\cdot p)(p_2^*\cdot p)=(p_1^*\cdot\overset{\curvearrowright}{p}_2^*)\|p\|^2$, we have

$$\begin{aligned} (B_1C_2-C_1B_2)/4 &= (p_1^*\cdot\overset{\curvearrowright}{p}_2^*)\|p\|^2 \\ &\quad -(p_1^*\cdot p)(\overset{\curvearrowright}{p}_2^*\cdot p_2)-(p_1^*\cdot p_1)(\overset{\curvearrowright}{p}_2^*\cdot p)+(p_1^*\cdot p_1)(\overset{\curvearrowright}{p}_2^*\cdot p_2) \\ &\quad +(\overset{\curvearrowright}{p}_1^*\cdot p)(p_2^*\cdot p_2)+(\overset{\curvearrowright}{p}_1^*\cdot p_1)(p_2^*\cdot p)-(\overset{\curvearrowright}{p}_1^*\cdot p_1)(p_2^*\cdot p_2). \end{aligned}$$

Moreover, we have $(p_l^*\cdot p)^2+(\overset{\curvearrowright}{p}_l^*\cdot p)^2=\|p_l^*\|^2\|p\|^2$ and $(p_l^*\cdot p)(p_l^*\cdot p_l)+(\overset{\curvearrowright}{p}_l^*\cdot p)(\overset{\curvearrowright}{p}_l^*\cdot p_l)=\|p_l^*\|^2(p_l\cdot p)$. Therefore

$$(B_l^2+C_l^2)/4=\|p_l^*\|^2(\|p\|^2-2p_l\cdot p+\|p_l\|^2).$$

Because $(p_1^*\cdot p)(p_2^*\cdot p)+(\overset{\curvearrowright}{p}_1^*\cdot p)(\overset{\curvearrowright}{p}_2^*\cdot p)=(p_1^*\cdot p_2^*)\|p\|^2$

$$\begin{aligned} (B_1B_2+C_1C_2)/4 &= (p_1^*\cdot p_2^*)\|p\|^2 \\ &\quad -(p_1^*\cdot p)(p_2^*\cdot p_2)-(p_1^*\cdot p_1)(p_2^*\cdot p)+(p_1^*\cdot p_1)(p_2^*\cdot p_2) \\ &\quad -(\overset{\curvearrowright}{p}_1^*\cdot p_1)(\overset{\curvearrowright}{p}_2^*\cdot p)-(\overset{\curvearrowright}{p}_1^*\cdot p)(\overset{\curvearrowright}{p}_2^*\cdot p_2)+(\overset{\curvearrowright}{p}_1^*\cdot p_1)(\overset{\curvearrowright}{p}_2^*\cdot p_2). \end{aligned}$$

By the substitution of $\|p\|^2$ with $d_0^2+2p_0\cdot p-\|p_0\|^2$, $B_1C_2-C_1B_2$, $B_l^2+C_l^2$, $B_1B_2+C_1C_2$ can be reduced to be linear in p, and thus reduce (11) to be cubic in p.

The quadratic curve about p in (7) and the cubic curve about p in (11), according to Bézout's theorem, has $2\times 3=6$ intersection points in complex space. In fact, given special inputs, these six intersection points can all stay in real space in our experiments. Given a translation p, there is only one t value as indicated by (10) and thus only one rotation R with $t=\tan\frac{\gamma}{2}$. Considering the reflection of body $\mathcal{B}^*$, the number of solutions for $[p, R]$ is at most 12. ■

B. Flip Ambiguity Enumeration

Motivation: For inflexible body merging, a key point is how to enumerate the finite ambiguities during merging of two bodies. The existence of multiple ambiguities can be caused by the following:

- the lack of redundancy in constraints as shown in Fig. 8;
- or the collinearity of constraints as shown in Fig. 9.

In Section V-A, we presented a method to enumerate ambiguities, which however is a symbolic algorithm that solves polynomial equation systems. It is difficult to enumerate the flip ambiguities that are caused by the rough collinearity of constraint point set due to interferences of ranging noise.

In this section, we propose a numerical algorithm to enumerate flip ambiguities with the presence of ranging noise. Our observation is that a symmetry property exist for the flip ambiguities in Fig. 8(a)–(d) and in Fig. 9(a)–(e). That is, if we can obtain one ambiguity and flip it across the collinear nodes, then we can get the remaining ambiguities. For example, in Fig. 9(d), the other ambiguity can be obtained by flipping the depicted ambiguity across the line through nodes $1, 2^*, 3^*$. The most difficult case is in Fig. 8(d), where there are two flip lines ϕ_0, ϕ_1 and four flip ambiguities. In this situation, the second ambiguity is obtained by flipping the first ambiguity across ϕ_0, the third ambiguity is obtained by flipping the first ambiguity across ϕ_1, and the fourth ambiguity is obtained by flipping the first ambiguity across ϕ_0 then across ϕ_1.

We present in Algorithm 3 the pseudocode of our algorithm to enumerate flip ambiguities, which can improve the robustness against collinearity. First, we test the collinearity of each constraint point set $P_{\mathcal{C}}(k)$ at lines 2–6. Our collinearity detection at line 6 can tolerate the presence of ranging noise, thanks to the ability of orthogonal regression in detecting rough collinearity of a constraint point set. Second, we use detected flip lines to enumerate flip ambiguities at lines 7 and 8.

Algorithm 3: EnumerateFlipAmbiguities

Input: An ambiguity of body $\mathcal{B}$ (call it body $\mathcal{B}$ for short);
An ambiguity of body $\mathcal{B}^*$ (call it body $\mathcal{B}^*$ for short);
Constraint Set $\mathcal{C} = \{\ell_l\} = \{[n_l, n_l^*, d_l, w_l]\}$
Output: All flip ambiguities when merging two ambiguities $\mathcal{B}$,$\mathcal{B}^*$

1 Get the first ambiguity by invoking Algorithm 2
// Steps to obtain other ambiguities
2 **switch** *number of nodes in body* $\mathcal{B}^*$ **do** // Theorem 1
3 **case** 1*:* $k = 2^{|\mathcal{C}|}-1$; **case** 2: $k = 1$; **otherwise**: $k = 0$
4 **for** $k \mathrel{+}= 2^{|\mathcal{C}|} - 2^{|\mathcal{C}|-NumOfSharedNodes}$; $k < 2^{|\mathcal{C}|}$; $k{+}{+}$ **do**
5 **if** *the nodes in constraint point set* $P_{\mathcal{C}}(k)$ *overlap at one position* **then return** empty set $\emptyset$ to report failure
6 **if** *constraint point set* $P_{\mathcal{C}}(k)$ *is collinear around a line* ϕ *obtained by orthogonal regression* **then** record line ϕ
7 **if** *a flip line* ϕ_0 *has been found* **then** get the second ambiguity by flipping the first ambiguity across line ϕ_0
8 **if** *a second flip line* ϕ_1 *has been found* **then** get the third ambiguity by flipping the first ambiguity across ϕ_1; get the fourth ambiguity by flipping the first one across ϕ_0 and ϕ_1

C. Algorithm Complexity and Ambiguity Explosion Problem

We analyze the computational complexity of our proposed algorithms from two perspectives. One is the cost of body merging primitive. The other is the complexity of our localization algorithm that iteratively invokes the merging primitive.

Our body merging primitive is to merge inflexible body B with inflexible body B^* to generate the resultant body B^+ and enumerate the flip ambiguities. Its computational complexity is $O(n{\times}n^*{\times}(|B^+|{+}2^{|C|-m}))$, where

n number of ambiguities in body B.
n^* number of ambiguities in body B^*.
$|B^+|$ number of nodes in the merged body B^+.

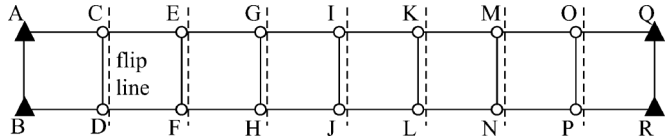


Fig. 12. Example of ambiguity explosion.

$|C|$ number of constraints between bodies B,B^*.
m number of shared nodes between bodies B,B^*.

There are totally $n{\times}n^*$ combinations, for ambiguities in body B and ambiguities in body B^*. For each combination, the two ambiguities are treated as two rigid bodies and are merged by Algorithm 2. The complexity of Algorithm 2 is $O(|B^+|)$ because its most expensive operation is the refinement step that fine-tunes the locations of each node in the merged body B^+. After Algorithm 2, we need to further invoke Algorithm 3 to enumerate flip ambiguities. Since Algorithm 3 needs to check each constraint point set for collinearity, its computational cost is $O(2^{|C|-m})$, where $|C| - m$ is the number of links between two bodies which is often smaller than $\sqrt{|B^+|}$.

Our localization algorithm invokes the body merging primitive iteratively. Its computational cost is difficult to express as a function of network size, because it is also affected by network sparsity level and specific network topology. For example, we can deploy the network topology in Fig. 12 on a bridge to monitor structural health. In this special topology, the computational cost of our localization algorithm grows exponentially with the network scale. This is because when we merge two bodies (e.g., ABCD and CDEF), the number of ambiguities of the merged body will double due to the flip line CD. Since there are totally seven flip lines, the number of ambiguities will grow exponentially to 2^7 after we merge all the local bodies in the network. Such a phenomenon is called "ambiguity explosion." As described before, the computational cost of our merging primitive is proportional to the number of ambiguities. Thus, when ambiguity explosion exists, the cost of our localization algorithm can grow exponentially. However, the depicted topology is just an extreme case to have such a long chain of ambiguous body merging. In our experimental networks in rectangular field and in O-shaped field, the chain of exponential growth is frequently broken after three or four times of body merging. The subsequent redundant body merging can help reject parts of ambiguities accumulated.

To mitigate ambiguity explosion problem, we apply an additional rule to the merging order of iterative merging process.

- When multiple pairs of bodies are available that can satisfy the merging condition $\mathrm{DOC}{\geq}\mathrm{DOF}$, we prefer the body pair whose constraints have the largest $\mathrm{DOC}{-}\mathrm{DOF}$.

The intuition for this rule to prefer large $\mathrm{DOC}{-}\mathrm{DOF}$ value is that the larger this value is, the smaller the chance for body merging to increase the number of ambiguities, and also the greater the chance to reject incorrect ambiguities.

However, the above rule may collide with the previously mentioned rule that prefers the merging of global body with local body (call it embedding). We resolve the collision as follows. Name the rule that prefers embeddings as rule 1, and name the rule that prefers largest $\mathrm{DOC}{-}\mathrm{DOF}$ value as rule 2.

- First, we try to apply rules 1 and 2 jointly. It means that we use rule 1 to get a list of body embeddings, and apply

TABLE I
SYSTEM PARAMETERS TO BE TESTED

NtwkDegree	Network degree is, for a sensor, the expected number of neighbors to which the distances can be measured.
AncPercentage	Anchor percentage is the percentage of GCF anchors deployed in the network, among all the network nodes.
NtwkRegion	Network region is the region to deploy sensor network, which can be of rectangular shape, O Shape, or U Shape.
GCF Anchor LocError	Localization Error of a GCF anchor is the distance between its true location and its declared location.
LocPercentage	Localization percentage is defined as the percentage of nodes that can be localized with finite ambiguities.
LocError	Localization error of a node is the distance between its true location and its unique location estimate.
Body Merging Accuracy	Body merging accuracy is the mean-squared error of all constraints between two bodies, as formalized in Eq. (3).

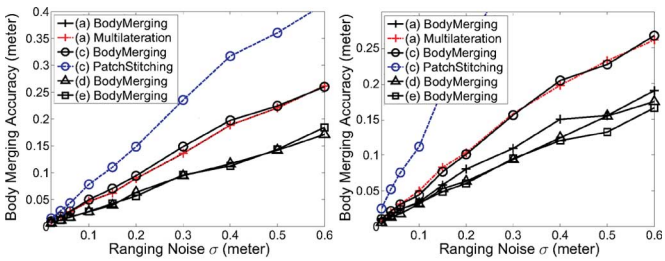


Fig. 13. Comparisons in body merging accuracy. (a) GCF anchor $\mathrm{LocError} = 0$. (b) GCF anchor $\mathrm{LocError} = \sigma$.

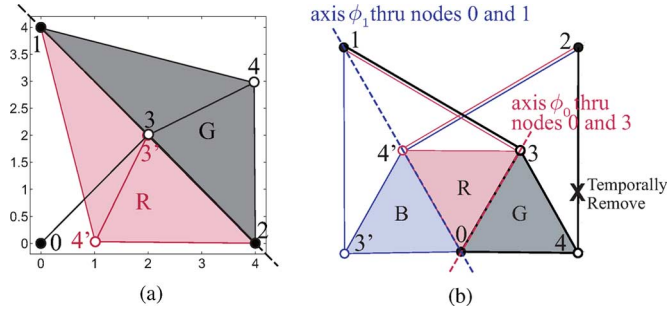


Fig. 14. Flip ambiguity enumeration during body merging. (a) Test case (c). (b) Test case (d).

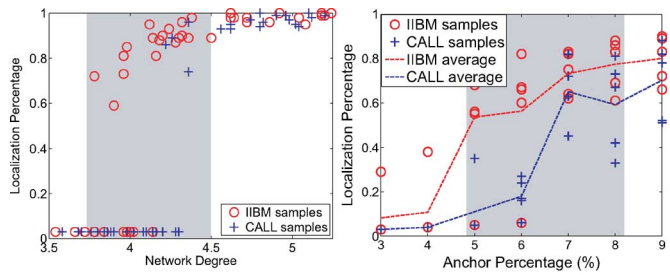


Fig. 15. Comparisons in localization percentage. (a) Fixed anchor number $=3$. (b) Fixed network degree ≈ 3.8.

rule 2 to this list to find the body embedding with the largest $\mathrm{DOC}-\mathrm{DOF}$ value. If this body pair can satisfy the condition of $\mathrm{DOC}>\mathrm{DOF}$, then output it directly.

- Otherwise, we abandon rule 1 and adopt rule 2 alone.

VI. SIMULATION RESULTS

In this section, we use simulation results to verify the contributions we declared. We first verify that our body merging optimization algorithm is more generalized than the traditional algorithms (i.e., multilateration and patch stitching) in tolerating ranging noise. We then show that our flip ambiguity enumeration algorithm can improve the robustness of body merging. We further show that our generalized merging condition $\mathrm{DOC}\geq\mathrm{DOF}$ can achieve higher localization percentage than with state-of-the-art CALL [13]. Finally, we present the integrated testing results showing that our IIBM algorithm can localize concave networks with uneven density distribution.

A. Experimental Settings

We adopt ultrasonic TOA for ranging whose radius is 6 m and noise is 2 cm. Similar ranging parameters are adopted in citation [15]. In Table I, we list the system parameters that can be adjusted for sensor networks, e.g., network degree, anchor percentage, shape of network deployment region, and the error in deployed GCF anchors. We will investigate their impact on the localization performance metrics, including localization percentage, and average localization error. A nonconventional performance metric is the accuracy of body merging that is quantified by the mean squared error of constraints between two bodies. It can regarded as the average deformation of springs between two bodies.

B. Body Merging Accuracy by Tolerating Ranging Noise

This experiment is to show that our body merging optimization algorithm in Algorithm 1 can handle all the cases in Fig. 7(a)–(e) while achieving high body merging accuracy that is proportional to ranging noise. In contrast, the traditional multilateration can only handle the case in Fig. 7(a), and patch stitching is designed for the case in Fig. 7(c).

In Fig. 13, we assume the merging is between the global body and a local body, and all the nodes in the global body are GCF anchors. We then assume that the error in GCF anchor positions is zero in Fig. 13(a) and is equal to range noise σ in Fig. 13(b). Our purpose is to simulate the effect of error accumulation in Fig. 13(b), i.e., larger localization error in the global body due to the increased number of nodes.

Fig. 13(a) shows that our body merging optimization algorithm can tolerate ranging noise and achieve high accuracy for all the cases of body merging (see Fig. 7). Fig. 13(a) also shows that when handling case (a), we can achieve accuracy comparable to multilateration; when handling case (c), we can achieve higher accuracy than patch stitching. This is because when local body $\mathcal{B}^*$ contains multiple nodes, it will have localization noise in its own coordinate frame. Patch stitching will not modify these node positions after merging. However, we will refine the merged body $\mathcal{B}^+$ using AFL spring relaxation (see Algorithm 2). The most outstanding advantage is that all the cases can be handled in a unified manner since the two groups of nodes are aligned based on rigid body dynamics.

Fig. 13(b) shows that our body merging optimization algorithm can mitigate the error accumulation problem, i.e., the error in node positions within a body get increased in repeated merging process. In Fig. 13(b), we simulate this effect by increasing the location error of global body from zero to

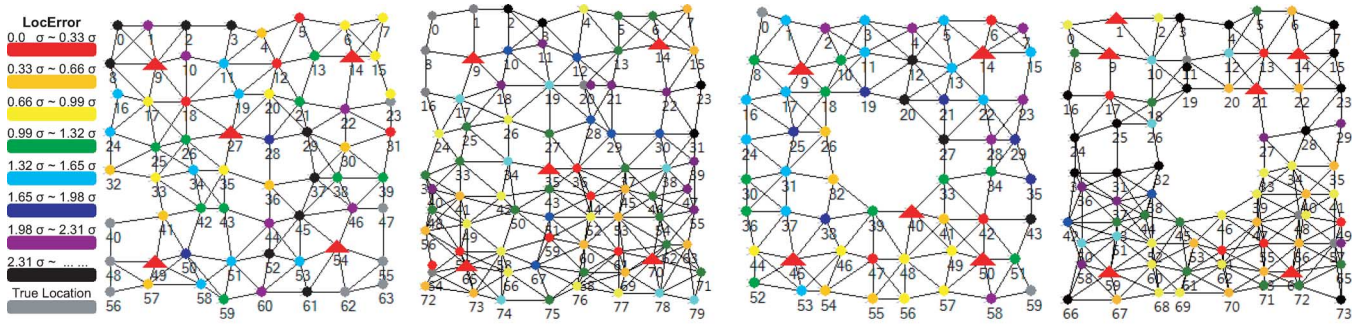


Fig. 16. Integration test results: Localization of whole networks by IIBM algorithm. (a) NtwkDegree=4.81, LocError=1.58σ, LocPercentage=85%. (b) NtwkDegree=6.4, LocError=1.26σ, LocPercentage=94.6%. (c) NtwkDegree=5.13, LocError=1.16σ, LocPercentage=96.7%. (d) NtwkDegree=6.35, LocError=1.45σ, LocPercentage=100%.

ranging noise σ. Then, our body merging optimization algorithm outperforms multilateration when handling case (a) and outperforms patch stitching when handling case (c). Such an advantage is because we apply refinement to merged body $\mathcal{B}^+$ to tolerate the positioning uncertainties in body $\mathcal{B}$.

C. Robust Body Merging Against Unexpected Flipping

For body merging, it is important to discover the collinear geometry in constraint set $\mathcal{C}$ and enumerate flip ambiguities, as discussed in Sections IV-B and V-B. We use the experiments in Fig. 14 to show our flip ambiguity enumeration algorithm in Algorithm 3 can handle all the collinear cases in Fig. 9.

Fig. 14(a) shows that our algorithm can enumerate flip ambiguities for patch merging. A patch with nodes 0, 1, 2 is merged with another patch 1, 2, 3, 4. They share two nodes 1, 2 and have one connecting link $[0, 3]$. Since the three nodes 1, 3, 2 are collinear, this second patch has two possible realizations that are marked with gray and red colors separately.

Fig. 14(b) depicts an abnormal case whose number of ambiguities is three. Our algorithm can handle such a case. The patch with nodes 0, 1, 2 is merged with the other patch 0, 3, 4. Their constraint set contains one shared node 0 and two links $[1, 3]$, $[2, 4]$. Since the three nodes 0, 3, 2 are collinear, there exist multiple ambiguities during patch merging. However, there exist three ambiguities (marked by gray, red, and blue colors) rather than two. For enumeration, we first reduce the constraint set to be determined (i.e., $\mathrm{DOC} = \mathrm{DOF}$) by removing a link, e.g., $[2, 4]$. Then, the first ambiguity can be obtained (e.g., with the gray color) together with the two illustrated flip axes ϕ_0 and ϕ_1. The red ambiguity is obtained by flipping the gray ambiguity across axis ϕ_0. The blue ambiguity is obtained by flipping the gray ambiguity firstly across ϕ_0 then across ϕ_1.

D. Localization Percentage in Sparse Networks

In this experiment, we verify that our IIBM algorithm can achieve high localization percentage in sparse networks with sparse anchor distribution and low network degree. We also show that, with the unified body merging condition DOC>DOF, we can localize more nodes than CALL [13].

Fig. 15(a) depicts the relation between network degree and anchor percentage. It shows that both our algorithm and CALL can achieve close to 100% localization percentage when the network degree is larger than 4.5. When the network degree is reduced between 3.7 and 4.5, our IIBM algorithm can outperform CALL by roughly two times. This is because CALL does not consider ambiguous cases (c) and (d) in Fig. 8 for body merging, which would reduce the chance for local bodies to expand. In contrast, IIBM algorithm can apply all the ambiguous merging cases (a)–(e) equally to the merging of two local bodies and the merging of global body with a local body.

Fig. 15(b) depicts the relation between anchor percentage and localization percentage. It shows that IIBM algorithm strongly outperforms CALL when the anchor percentage is between 5% and 8%. This is because IIBM algorithm gives local bodies greater chance to expand, and a larger local body has better chance to be localized in GCF.

E. Performance in Convex and Concave Networks

This experiment is an integration test of our IIBM algorithm to localize a whole network, which can be of different densities and deployed in concave or convex regions. As shown in Fig. 16, we have tested both squared region and O-shaped region. In either region, the network density can be uniform or nonuniform. For nonuniform networks, the density of lower parts is increased by 50%.

In Fig. 16, GCF anchors are drawn as triangles, and sensors are drawn as dots. If a sensor can be localized uniquely, then a colored dot is used to show its estimated location, and its color corresponds to the localization error [see the legend on the left of Fig. 16(a)]. The nodes that cannot be uniquely localized are shown as gray dots at their true locations, e.g., nodes 55, 62, 63 in Fig. 16(a) and 0, 1, 8, 16 in Fig. 16(b).

In all the four test cases, the average error of localized nodes is kept below 2σ. This good accuracy is because: 1) we can provide good body merging accuracy by our body merging optimization algorithm; and 2) we can ensure body merging robustness by enumerating the flip ambiguities. For example, in Fig. 16(a), nodes 40, 48, 56 cannot be localized uniquely due to the collinearity of nodes 41, 49, 57. Moreover, for the sparse networks in Fig. 16, the localization percentage is kept above 80%, which is consistent with the results in Fig. 15.

VII. Conclusion

This paper focused on fine-grained iterative localization of wireless sensor networks. We have proposed an optimization algorithm that can tolerate ranging noise when aligning two bodies. This algorithm is very generalized and can replace the traditional multilateration and patch stitching as the new primitive for network localization. We also proposed a flip ambiguity enumeration algorithm to improve the robustness of localization. This algorithm can discover the rough collinearity of constraints due to the interference of ranging noise. Our other

contribution is that our body merging condition DOC≥DOF can achieve higher localization percentage than state-of-the-art SWEEPS and CALL. Our final demonstration shows that our IIBM localization algorithm can work well, even in concave networks with nonuniform network density.

REFERENCES

[1] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. ACM MobiCom*, 2003, pp. 81–95.

[2] L. Dimitrios, L. Quentin, and S. Andreas, "An empirical characterization of radio signal strength variability in 3-D IEEE 802.15.4 networks using monopole antennas," in *Proc. EWSN*, 2006, pp. 326–341.

[3] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *J. Telecommun. Syst.*, vol. 22, no. 1–4, pp. 267–280, 2003.

[4] H. Lim and H. J. C. , "Localization for anisotropic sensor networks," in *Proc. IEEE INFOCOM*, 2005, pp. 138–149.

[5] C. Wang and L. Xiao, "Locating sensors in concave areas," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[6] B. Xiao, L. Chen, Q. Xiao, and M. Li, "Reliable anchor-based sensor localization in irregular areas," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 60–72, Jan. 2010.

[7] Q. Xiao, B. Xiao, J. Cao, and J. Wang, "Multihop range-free localization in anisotropic wireless sensor networks: A pattern-driven scheme," *IEEE Trans. Mobile Comput.*, vol. 9, no. 11, pp. 1592–1607, Nov. 2010.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. ACM MobiCom*, 2000, pp. 32–43.

[9] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," in *Proc. ACM SenSys*, 2003, pp. 340–341.

[10] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proc. IEEE INFOCOM*, 2004, vol. 4, pp. 2652–2661.

[11] A. Savvides, H. Park, and M. B. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mobile Netw. Appl.*, vol. 8, no. 4, pp. 443–451, 2003.

[12] B. K. P. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Amer. A*, vol. 5, no. 7, pp. 1127–1135, 1988.

[13] X. Wang, J. Luo, S. Li, D. Dong, and W. Cheng, "Component based localization in sparse wireless ad hoc and sensor networks," in *Proc. IEEE ICNP*, 2008, pp. 288–297.

[14] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. Anderson, A. S. Morse, and Y. R. Yang, "Localization in sparse networks using sweeps," in *Proc. ACM MobiCom*, 2006, pp. 110–121.

[15] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. ACM SenSys*, 2004, pp. 50–61.

[16] L. Meertens and S. Fitzpatrick, "The distributed construction of a global coordinate system in a network of static computational nodes from internode distances," Tech. Rep. KES.U.04.04, 2004.

[17] F. W. H. , "Position-location solutions by Taylor-series estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-12, no. 2, pp. 187–194, Mar. 1976.

[18] A. Witkin, D. Baraff, and M. Kass, "Physically based modeling: Principles and practice," ACM SIGGRAPH Course Notes, 1997 [Online]. Available: http://www.cs.cmu.edu/~baraff/sigcourse/

Qingjun Xiao (M'12) received the B.Sc. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, the M.Sc. degree from Shanghai JiaoTong University, Shanghai, China, in 2007, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2011, all in computer science.

Currently, he is a Postdoctoral Researcher with the University of Florida, Gainesville, FL, USA. His research interests include protocols and distributed algorithms in wireless sensor networks.

Bin Xiao (M'02–SM'11) received the B.Sc. and M.Sc. degrees in electronics engineering from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from the University of Texas at Dallas, Dallas, TX, USA, in 2003.

Currently, he is an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include mobile cloud computing, data management, network security, wireless sensor networks, and RFID systems.

Dr. Xiao is a member of the Association for Computing Machinery (ACM). He is an Associate Editor for the *International Journal of Parallel, Emergent and Distributed Systems*. He is a recipient of the Best Paper Award of IEEE/IFIP EUC 2011.

Kai Bu received the B.Sc. and M.Sc. degrees in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2006 and 2009, respectively, and is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

His research interests include RFID and wireless networks.

Mr. Bu is a member of the IEEE Communications Society. He is a recipient of the Best Paper Award of IEEE/IFIP EUC 2011.

Jiannong Cao (M'93–SM'05) received the B.Sc. degree from Nanjing University, Nanjing, China, in 1982, and the M.Sc. and the Ph.D. degrees from Washington State University, Pullman, WA, USA, in 1986 and 1990, respectively, all in computer science.

He is currently a Chair Professor and the Head of the Department of Computing with The Hong Kong Polytechnic University, Hong Kong. His research interests include mobile and pervasive computing, computer networking, parallel and distributed computing, and fault tolerance. He has published over 280 technical papers in these areas.

Dr. Cao is a Senior Member of the China Computer Federation, the IEEE Computer Society, and the IEEE Communication Society, and is a member of the Association for Computing Machinery (ACM).