IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 January 2024, accepted 24 January 2024, date of publication 26 January 2024, date of current version 16 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3358889

RESEARCH ARTICLE

A Component-Based Localization Algorithm for Sparse 3-D Wireless Sensor Networks

MAZHAR ISLAM [1], **MUHAMMAD IKRAM** [2], **MUSAED ALHUSSEIN** [3], **MUHAMMAD SOHAIB AYUB** [4], **MUHAMMAD ASAD KHAN** [5], **AND KHURSHEED AURANGZEB** [3], **(Senior Member, IEEE)**

[1]College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China
[2]Huawei Pakistan, Islamabad 44000, Pakistan
[3]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Computer Science, School of Science and Engineering, Lahore University of Management Sciences, Lahore 54792, Pakistan
[5]Department of Telecommunication, Hazara University Mansehra, Mansehra 21120, Pakistan

Corresponding author: Mazhar Islam (islammazhar2023@email.szu.edu.cn)

This Research is funded by Researchers Supporting Project Number (RSPD2024R553), King Saud University, Riyadh, Saudi Arabia.

ABSTRACT Node localization is one of the most essential features of wireless sensor networks (WSNs). Vavarious localization algorithms exist for densely deployed 3-D wireless sensor networks. However, for a sparse 3-D network, range-based localization is still a challenging task because it is difficult to find sufficient anchor nodes and distance information among nodes in a sparse 3-D network. To mitigate the sparseness issues in 3-D sensor networks, we present a component-based localization method in this paper in which we split the entire network into small overlapping sub-networks called components and assign local coordinates to each component. Then, we merge these small components to make a globally coordinated system. With a meager anchor ratio, we localize the whole network. We define merging conditions according to the number of common nodes, actual measured distances among nodes, and the calculated distance based on the local coordinates of the nodes. We assess how well our proposed algorithm performs by conducting extensive simulations. The outcomes confirm that the proposed algorithm works comparatively better in a sparse 3-D sensor network than in a densely deployed 3-D sensor network. Our algorithm localizes more than 83% of nodes at a node degree of 10 having 5% anchor ratio; however, other algorithms localize only 18%-79% in the same scenario.

INDEX TERMS Wireless sensor network, 3-D localization, component-based localization, patch and stitching localization.

I. INTRODUCTION

Internet of Things (IoT) applications are increasing gigantically in many fields [1], [2], [3]. In IoT applications, the most extensively used end devices are the nodes in Wireless sensor networks [4], [5]. A wireless sensor network (WSN) consists of several number of nodes that are randomly deployed. A wireless sensor network's objective is to keep an eye on the area's physical conditions, collect the data, and, through the collaboration of sensor nodes with each other, process the data to a central location [6], [7], [8]. WSNs can be extensively used in many fields, e.g., oceans monitoring [9], health monitoring [10] military surveillance [11], wildlife monitoring [12], and disaster management [13].

The information gathered from any sensor node in a WSN application is only useful if we know its source. The data without the location information is of no use. Therefore, localization is an important aspect in many applications of WSN. Similarly, while developing algorithms and protocols for WSNs, sensor node position plays an important role. For example, node location information is required in target tracking [14], indoor positioning systems [15], geographical routing [16] and coverage optimization [17]. There are two basic categories in which localization algorithms for wireless sensor networks can be broadly divided. These are range-based algorithms [18] and range-free algorithms [19].

The associate editor coordinating the review of this manuscript and approving it for publication was Chenshu Wu.

In range-based localization, a sensor node obtains its location by angle or distance information between itself and its neighbors. Meanwhile, in range-free algorithms, the determination of the location of an unknown sensor node relies merely on connectivity knowledge or hop count. The accuracy level of range-based algorithms is higher than range-free algorithms because special hardware is used in range-based localization algorithms to obtain angular or distance measurements, while no special hardware is used in range-free localization algorithms. Generally speaking, range-based algorithms are considered more costly than range-free algorithms due to the cost of the specialized hardware used. In many Internet of Things applications, very accurate position information of the node is required [20], so range-based algorithms are preferred.

In the past several years, researchers have introduced many range-based localization algorithms [21], [22]. Most of these algorithms are for 2-D networks in which the network's depth or height is not considered. But in a realistic environment, WSN is generally utilized in 3-D networks like underwater wireless sensor networks [23], [24], [25] in which the depth of the nodes is considered. So localization for 3-D networks remains a confronting task. Conventional range-based localization algorithms mostly use sequential localization techniques to localize the whole network nodes, in which sensor nodes are localized one by one in proper sequential order. In this case, each node must have enough angle or distance information of the nodes before it is in the sequence. The sequential localization strategy is successful when the network is dense or has a greater node degree. However, in the most of the 3-D networks, the fundamental sparseness poses a challenge in acquiring an appropriate sequential order of nodes for localization. Therefore, sequential localization algorithms face difficulties and are prone to failure in sparse 3-D network scenarios.

To mitigate the sparseness problem of sensor networks, component-based localization or patch and stitching-based localization techniques have been proposed [26], [27], [28], [29], [30], [31], [32], [33]. A component-based localization strategy divides the sparse sensor network into small sub-networks called components. Each component can be localized with a conventional sequential localization strategy. Each component is assigned a local coordinate system based on the sensor node's relative distance information. Then, these small components are amalgamated to form a single global component, and forming a global coordinate system. The component-based localization strategy has solved the sparseness problem for 2-D networks [27], [28], [29], [30], [31], [32], however, for 3-D networks, components-based localization is still a challenging task. One of the essential aspects of designing the localization algorithm for the component-based approach is to define the states under which two component patches can be merged. The existing component-based localization techniques are primarily for 2-D networks [27], [28], [29], [30], [31], [32], so these conditions can not be used to localize a 3-D network. For component-based localization of 3-D sensor networks, merging conditions still need to be clearly defined as there is not much work done for sparse 3-D sensor networks. Localization is also considered a very important aspect in the Near Field Communication [34], [35]. To this end, we solve the sparseness problem in 3-D sensor networks by component-based localization and define the merging conditions. The main contributions and innovations of this research article are summarized below.

- We propose a method for node localization in the sparse wireless sensor 3-D network by dividing the network into small sub-networks called components.
- We assign local coordinates within a single component facilitating better management and localization.
- We define the merging conditions based on the number of common nodes, actual distances, and the measured distances. The innovation lies here as it is difficult to have enough neighbor nodes in a sparse network. So based on the minimum neighbors and their distance information, we merge the components with the help of calculated distances and measured distances.
- We conduct extensive simulations to evaluate the performance of the proposed algorithm demonstrating that the proposed algorithm performs well in the sparse 3-D network.

The rest of the paper is structured as follows: Section II presents the literature review of the localization techniques in 3-D WSNs. We formulate our localization problem in section III. Section IV explains the framework of our proposed component-based localization algorithm in detail. Experimental evaluation is done in section V. Finally, we conclude the paper in section VI along with future directions.

II. LITERATURE REVIEW

The localization study in wireless sensor networks has earned meaningful attention from researchers in recent times [36], [37], [38], [39], [40], [41], [42]. The ancestor studies employ graph rigidity theory to determine the orders for the unique localization of the entire network. In 2-D sensor networks, attaining unique localization for the whole network is subject to the global rigidity of the network graph and the presence of a minimum of three anchor nodes [37]. If it is impossible to localize the whole network, such conditions are considered in which the individual nodes can be uniquely localized. In RRT-3B algorithm [39], [40], some states of unique localization are given for the nodes in a 2-D sub-network. These three conditions are 1) the sub-network topology should be rigid, 2) the presence of a minimum of three anchor nodes is essential, and 3) the sub-network should form a trilateration graph. In [43], the authors examined the localization conditions of nodes and grasped the upper limit for the number of nodes localized within a network.

Even for networks considered uniquely localizable, the complete realization of the entire network persists in a challenging endeavor, especially in the case of a very sparse network [37], [44]. The difficult task in sparse networks lies in the complexity of finding a trilateration order of nodes contributing to sequential localization [38]. To mitigate the problem of sparse networks, the Sweep algorithm is proposed in [41] and [42], which introduced the concept of finite localization. The wheel network is explicitly designed for an apparent category of sparse networks that relaxes the criteria of the trilateration order of nodes and introduces bilateration ordering, which results in more localized nodes than traditional sequential trilateration-based algorithms. For a very sparse network, it is still challenging to find bilateration order, with the worst-case scenario exhibiting exponential time complexity, so performance degrades significantly.

To overcome the shortcomings of sequential localization, component-based localization has been proposed for two-dimensional networks [27], [28], [30], [32]. In component-based algorithms, the information about common nodes and distance is used to localize nodes. In [30], certain conditions are given for merging two components if there are not enough common nodes. In [27] and [28], components do not overlap. Hence, they do not have any common node. So, the nodes are localized with only distance information between the neighboring components. As a result, only a few nodes are localized, limiting the localization process in a sparse network. In [32], the authors proposed two algorithms called basic common nodes-based localization algorithm (BCLA) and component-based localization algorithm with angle and distance information (CLAD). BCLA improved the component-based localization by exploiting the information of common nodes and the distance among neighbor components. CLAD further improved the localization process by utilizing angle information, distance information, and common nodes. In [29], a robust method called error tolerant component-based algorithm (ETOC) was introduced, which deals with ranging noises in component-based localization algorithms. However, all these component-based algorithms are for 2-D networks and can not work for 3-D networks.

From the last few years, researchers are more focused on the localization of 3-D [45], [46], [47], in which more focus is on underwater wireless sensor networks [48], [49], [50], [51]. In [52], the algorithm for an underwater sensor network is proposed, which is called the UPS algorithm. In this algorithm, the positions of nodes in three-dimensional space are projected onto two-dimensional space. Then, a standard 2-D localization algorithm is used to find the location of sensor nodes. In [48], the performance of different multilateration algorithms is studied by changing the deployment strategy of multiple anchor nodes. In [49], intruder localization is studied in UWSNs. In [50], the UWSN is divided into sub-networks called skeleton, and the number of localizable sensors is found. In [26], component-based localization for 3D networks is introduced in which certain conditions are derived for uniquely merging two sub-networks. The localization of sensor nodes depends on information concerning common nodes and connecting edges within the network. For transforming the coordinate system, translation parameters are treated as unknowns to form a set of uniquely solved equations. Most of the algorithms proposed require the dense deployment of nodes. Contrary to them, this proposed algorithm is more suited for sparse 3-D networks.

III. PROBLEM FORMULATION

We consider a large 3-D sensor network with N nodes. For this network, we generate a weighted distance graph [37], $G = [N, E, D]$. Each graph vertex represents a node n in the network. If there is a distance measurement between two nodes n_i and n_j, it is represented by the connecting edge $E(i, j)$. Each connecting edge has a weight called $D(i, j)$. The network has M anchor nodes ($M < N$) that are aware of their locations either through manual configuration or via GPS. We label the anchor nodes from 1 to M. The other nodes unaware of their positions are called unknown nodes and are labeled from $M + 1$ to N. Every unknown node needs to obtain its physical location by using the distance information of its neighbor nodes and the locations of the anchor nodes. The distance information between the sensor node and its neighbors is known. For any two nodes n_i and n_j, the distance measurement is given by d_{ij}. The physical location of node n_i is given by P_i or (x_i, y_i, z_i). A *component realization* is the mapping of component nodes to the coordinates in 3-D space, $P : N \rightarrow \mathbb{R}^3$. For all $(n_i, n_j) \in E$, $||P_i - P_j|| = d_{ij}$. $||P_i - P_j||$ represents the Euclidean distance of nodes n_i and n_j having positions P_i and P_j respectively.

Before explaining the details of the proposed algorithm, we present the patch and stitching localization, also known as component-based localization, with the help of an example. In Figure 1a, the distance graph of a sensor network is shown where the circles represent the unknown sensor nodes and squares represent the anchor nodes. Suppose we try to localize the unknown nodes with common localization techniques. In that case, we can not localize because, for three-dimensional localization, every unknown node must have distance information to at least four anchor nodes. From Figure 1a, we see no single node with sufficient distance information to the anchor nodes. We can localize all the nodes using component-based localization for the same network. We divide the whole network into two small sub-networks called components, represented by *ComponentA* and *ComponentB* as shown in Figure 1b. We see that the two components can not be realized as they do not have enough anchor nodes, so we have to merge both of the components A and component B with the help of common nodes and connecting edges between the components to form a single merged component represented by component (A, B) as shown in Figure 1c. After merging the components, we see that the merged component is now realizable because it has

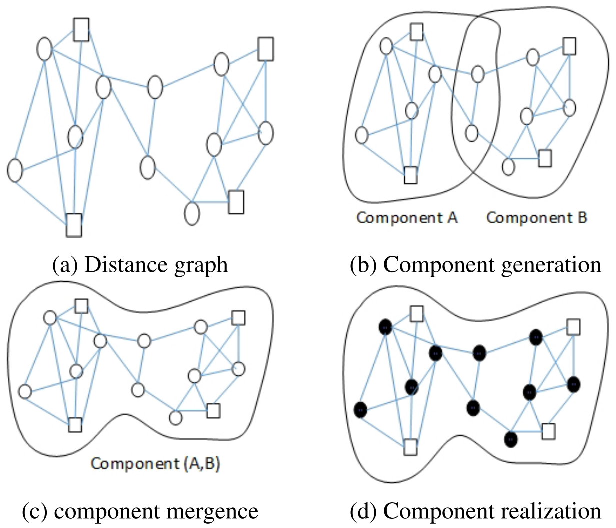

FIGURE 1. Component-based localization.

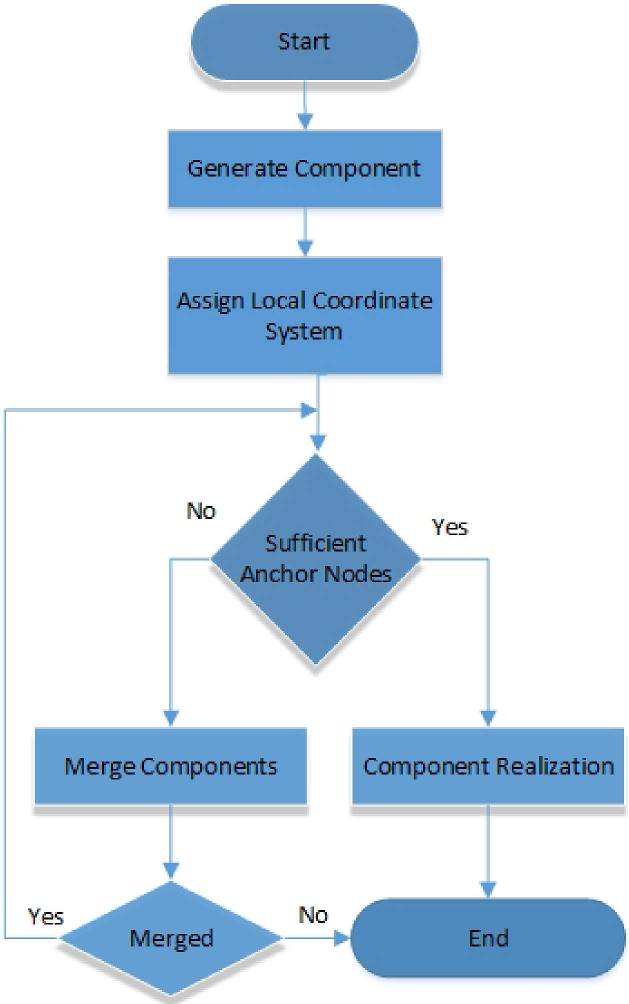

FIGURE 2. The overall illustration of the proposed algorithm.

enough anchor nodes in a whole component. All the nodes in the merged component have the coordinate so they can find the distance between them and anchor nodes, resulting in localization of all nodes in the component as shown by black circles in the Figure 1d.

IV. FRAMEWORK OF THE PROPOSED COMPONENT BASED LOCALIZATION ALGORITHM

The proposed algorithm is used to localize nodes in sparse 3-D sensor networks. It is based upon a patch and stitching technique in which the 3-D sparse wireless sensor network is divided into small parts called components [26]. Each component is assigned a local coordinate system based on the distance information of the nodes within a component. If the components meet the requirements of merging conditions, the components are merged to form the global coordinate system. Then the global coordinate system is converted to an absolute coordinate system based on anchor nodes' information. The algorithm can be divided into four basic steps: 1) component generation, 2) local coordinate system construction, 3) component mergence, and 4) component realization. The framework of the proposed algorithm is shown in the Figure 2. Each step is explained below.

A. COMPONENTS GENERATIONS

In the component generation phase, the sparse 3D network is divided into several small sub-networks called components. While generating the components, some techniques use a non-overlapping strategy in which the divided components do not overlap with each other [28], so they do not have any common node among the components. In the proposed work, we use an overlapping strategy in which one component can overlap with others, giving rise to some common nodes, which help merge the two components.

There are three main steps while generating the components. In the first step, we align all the nodes in the *unarranged* state. In the second step, we try to find four interconnected nodes, and among these four nodes, one of the nodes must be in the *unarranged* state. We call these four nodes a subgraph $G_0 = K_4$, and it is considered as the seed component to which other nodes can be added through multilateration. The seed component can be extended by finding any node n that is in *unarranged* state and not in G_0, but it is connected to at least four nodes in G_0, that node n is added to G_0 and G_0 is updated. In other words, we can say that this node n is added to the initial seed component, which has only four nodes. All the edges between this node n and subgraph $G_0 = K_4$ are also added to the seed component. This process is done until no more nodes can be added to G_0. This completes the generation of one component, so all the nodes in this component are marked as in *arranged* status. The same (step two) is repeated for generating other components until no more components can be formed. In the last (step three), the nodes still in *unarranged* status are labeled as in *isolated* state. This whole process is explained below in algorithm 1.

Algorithm 1 Component Generation

Input:The distance graph G=<N,E,D>
Output:Component with local coordinates

Mark all nodes with *unarranged* state.
Find four interconnected nodes making subgraph $G_0 = k_4$.
Mark the nodes in G_0 as in *arranged* state.
Mark subgraph $G_0 = k_4$ as seed Component.
repeat
  if there exists a node n with *unarranged* state and has at least four distances to subgraph k_4 **then**
    Add node n to G_0
    Mark the node as *arranged*
    Update G_0
    $G_0 = G_0 \cup n$.
    Update seed component
  end if
until No more node can be added to G_0
Mark the node having *unarranged* status as *Isolated* states
Construct the local coordinate system
Assign local coordinates to all nodes in G_0 and **return**

In the second step, while finding the initial subgraph $G_0 = K_4$, we put a restriction that at least one node should be in *unarranged* state. This restriction is because the number of components can be set to a minimum. Otherwise, without this restriction, many components can be generated, which increases exponentially as the number of nodes increases. By putting this restriction, the localization ratio increases because it produces overlapped components with enough common nodes and measured distances. In contrast, in [27] and [28], the components are nonoverlapping, thus they do not have common nodes and rely only on the measured distances between the components for the merging process, which results in a low localization ratio.

B. LOCAL COORDINATE ASSIGNMENT TO A COMPONENT

After the components are generated, and there are no further nodes that can be added to any component, we need to assign a local coordinate system to each component. A local coordinate system can be assigned to a component by selecting the initial four interconnected nodes, and local coordinates are assigned to these nodes. In the initial k_4 graph, we give coordinates to the first three nodes and then use the trilateration method to set the coordinates to the fourth node. After finding the coordinates of the initial four nodes, the rest of the nodes in a component can be assigned coordinates by using the multilateration technique.

For assigning a local coordinate system, we take three nodes A, B, C from the initial K_4 graph having nodes A, B, C, D. The coordinates of these three nodes can be defined in a plane by considering the node B at the origin of the coordinate system. We also consider that the node A is in the XY Plane and the node C is on positive $X-axis$ as shown in Figure 3.

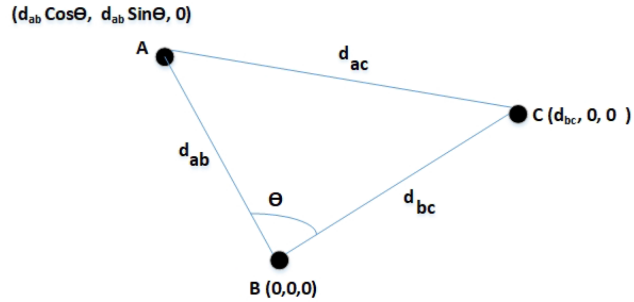

FIGURE 3. Coordinates of Nodes A, B, C in K_3.

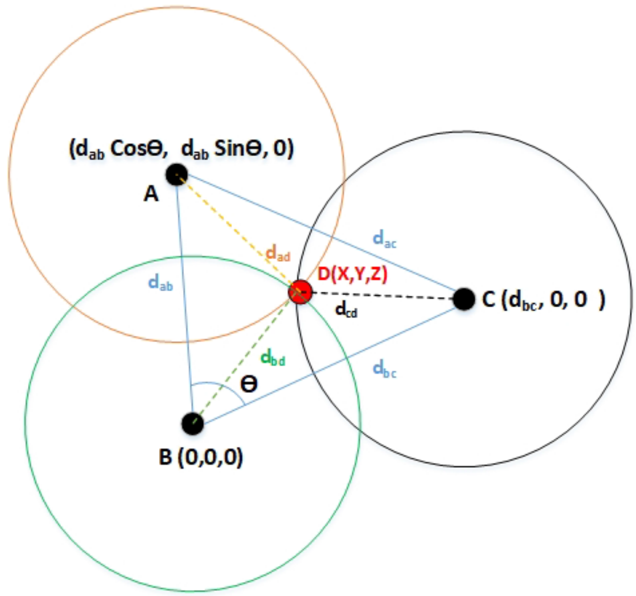

FIGURE 4. Coordinates of Node D in K_4.

As the nodes A, B, C are neighbors and have known measured distance information given by d_{ab}, d_{bc} and d_{ac}. According to the law of cosines, if the lengths of the sides are known, then the angles can be calculated by the formula below.

$$\theta = \cos^{-1}\left(\frac{d_{ab}^2 + d_{bc}^2 - d_{ac}^2}{2d_{ab}d_{bc}}\right) \tag{1}$$

After finding θ using the above formula (1), the coordinates of the nodes $A, B,$ and C can be estimated. To keep the simplicity, we suppose the node B to be at the origin having coordinates as $B = (0, 0, 0)$. We assume that the node C is on positive $X-axis$ having coordinates $C = (d_{bc}, 0, 0)$. As the node A is in the XY plane so the coordinates of node A is given by $A = (d_{ab}\cos\theta, d_{ab}\sin\theta, 0)$.

After finding the coordinates of $A, B,$ and C, we calculate the coordinates of the fourth node D in the initial K_4 graph by using the trilateration method as shown in Figure 4 below. In the trilateration method, we use the equations of three spheres centered at nodes A, B, C having a radius equal to d_{ad}, d_{bd} and d_{cd} respectively.

The equation of the sphere centered at node B with radius d_{bd} is given by the formula.

$$X^2 + Y^2 + Z^2 = d_{bd}^2 \tag{2}$$

The equation of the sphere centered at node C with radius d_{cd} is given by the formula

$$(X - d_{bc})^2 + Y^2 + Z^2 = d_{cd}^2 \tag{3}$$

Similarly the equation of the sphere centered at node A with radius d_{ad} is given by the formula

$$(X - d_{ab}\cos\theta)^2 + (Y - d_{ab}\sin\theta)^2 + Z^2 = d_{ad}^2 \tag{4}$$

By solving above three equations eqs. (2) to (4), having three unknowns X, Y and Z, we get the following

$$X = \frac{d_{bd}^2 - d_{cd}^2 + d_{bc}^2}{2d_{bc}} \tag{5}$$

$$Y = \frac{d_{bd}^2 - d_{ad}^2 + (d_{ab}\cos\theta)^2 + (d_{ab}\sin\theta)^2}{2d_{ab}\sin\theta} - d_{ab}\cot\theta \times \frac{d_{bd}^2 - d_{cd}^2 + d_{bc}^2}{2d_{bc}} \tag{6}$$

$$Z = \sqrt{d_{bd}^2 - X^2 - Y^2} \tag{7}$$

By using eqs. (5) to (7), we calculate the coordinates of node D, which completes the process of assigning local coordinates to the initial K_4 graph. Now, as the local coordinates are assigned to the K_4 graph, all the other nodes in the component can find their local coordinates through the multilateration method.

C. COMPONENT MERGENCE

In this step, if any of the two components satisfy the merging conditions discussed in section IV-F, the two components are merged to form a single component. After merging one component with the other, the coordinate system of one component should be converted to the coordinate system of another component so that both components have the same coordinate system. In this way, a global coordinate system is formed for the merged components. To merge the components, first of all, we select a seed component. There are many ways to choose the seed component. In [31], the seed component is selected randomly, so any component can be a seed component. In [33], the node having the highest degree is the seed node, and its respective component is selected as the seed component. We adopt the method of [26] and [32] and set the seed component having the highest number of nodes. Then, we add other components to the seed component so that the seed component can be extended and a single global network can be formed. Components merging can be either parallel or incremental [33]. In a parallel merging process, two components that satisfy the merging conditions are merged simultaneously. The incremental merging process selects a seed map, and other components are merged into the seed component in sequential order. In this paper, we use the incremental merging process [31]. Two ways exist to decide which component should be added first to the seed component. The first way is to take the component with the maximum number of nodes and add it to the seed component. The second way is to take the component having a maximum number of common nodes and add it to the seed component. Contrary to [26], we adopt the second method and select the component having a maximum number of common nodes and merge it to the seed component. If there is more than one component with the same number of common nodes, we break the tie by considering the component having a higher number of nodes. When two components are merged, the new merged component becomes a new seed, and we repeat the whole process till there is no further component to merge. After all the components are merged, we merge the isolated nodes. For isolated node merging, we try to find at least four connecting edges between the isolated node and the merged component, otherwise, the node remains isolated. The complete merging process is explained in algorithm 2

D. COMPONENT REALIZATION

The last step of the localization process is component realization. In this step, we use the coordinates of the anchor nodes and the measured distances to convert the global coordinates of the merged component to the absolute coordinates. At least four anchor nodes are required to calculate the absolute coordinates of the component. However, if the anchor nodes are not enough, only the relative positions of the nodes can be calculated.

E. COMPLEXITY ANALYSIS

We start with the component generation phase, finding a seed component (K_4) and extending it using the multilateration method. Finding complete subgraph with four edges K_4 takes $O(N^2)$ time [53]. After that, we find an ungrouped vertex with at least four distances to the seed component, which takes $O(N)$ time. We repeat this multilateration method until no further node can be added to the seed component taking $O(N)$ steps, i.e., multilateration extension takes a maximum $O(N^2)$ time. We assign local coordinates to all nodes in $O(N)$ steps. So, the component generation phase of our algorithm takes $O(N^2 + N^2 + N) = O(N^2)$. The time efficiency of our algorithm is much better than the worst-case time complexity due to the sparsity of our graph. For component mergence, we sort all the components according to their size to speed up the merging process due to a higher probability of satisfying merging conditions for larger components. For checking the component's mergence condition, we need to find common nodes and edges between these components, which takes $O(N^2/2)$ time, i.e., only two components containing half of the nodes each. We assign global coordinates to all the nodes in components in $O(N)$ steps. We need to check for a maximum $O(N/4)$ in the worst case, i.e., when there are only

Algorithm 2 Component Mergence

Input: C_i−Components generated as per algorithm 1
Output: Merged Component with global coordinate system

```
Find the seed Component C_s
if n(C_i) is maximum then
    C_s = C_i
end if
repeat
    Find next component to be merged C_n
    for all component C_i do
        Common = C_s ∩ C_i
    end for
    if n(Common) is maximum then
        if more than one component has maximum common nodes then
            Take (C_i) with maximum number of nodes
            C_n = C_i
        end if
    end if
    After satisfying merging conditions as per section IV-F
    Merge(C_s, C_n)
    Update C_s
    C_s = C_s ∪ C_n
until No more component can be merged
loop
    if there exists a node n with isolated state and has at least four distances to the final merged component C_s then
        Add n to the component C_s
        C_f = C_s ∪ n
    end if
end loop
Calculate global coordinate as per method given in section IV-D
```

seed components (K_4), and all satisfy merging conditions, making the worst-case running time of component mergance. $O(N/4 * N^2) = O(N^3)$. Overall, the runtime complexity of our algorithm becomes $O(N^2 + N^3) = O(N^3)$, but it would be much lower than this practically.

F. MERGING CONDITIONS

One of the most significant issues in the patch and stitching-based localization algorithms is merging the adjoining components. Generally speaking, components are generated in a patch and stitching-based localization, and local coordinates are assigned. After merging, the local coordinates are converted to global coordinates, then to absolute coordinates defined by the anchor nodes. This coordinate system conversion is done with the help of sufficient information about common nodes between the components, but how will this conversion be done if there is not an adequate number of common nodes between two components? To do the coordinate system conversion between components when there are not sufficient common nodes, we use the information of connected edges between the components and distances between the nodes in the components [30]. Two types of distance information can be used to convert the coordinate systems. One is the Measured distance, and the other is the calculated distance. The measured distance is the actual distance that two nodes have. When two nodes are in communication range with each other, they have a connected edge between them. This edge distance is called the measured distance. Calculated distance is based on the local coordinates of two nodes, and in this case, both nodes can not communicate with each other, and there is no connecting edge between both nodes.

$$||P_a - P_b|| = d_{ab} \tag{8}$$

where P_a and P_b are the 3-D coordinates of node a and node b respectively. we can further simplify it as

$$(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2 = d_{ab}^2 \tag{9}$$

In the conventional coordinate system conversion method, [54] three common nodes are required to merge two components in the 2-D networks and four common nodes in the 3-D networks. However, in a sparse 3-D sensor network, finding enough common nodes between the components is not easy. The authors in [27] and [28] proved that two components could be merged when at least four distance information exists for any 2-D network between two components. The algorithm also proved that components could be merged even when there is no common node between two components. This proposed work is related to a 3-D network, so more than four-distance information is required to merge the networks. We need some extra distance information between the two components. The proposed algorithm can merge two components with four or more common nodes without connecting edges, but when common nodes are less than four, we need the connecting edges to merge the network. The requirements of connecting edges depend upon the number of common nodes, as discussed in cases 1-5. Below are the conditions for merging two components when we have a different number of common nodes and connecting edges.

1) CASE 1: FOUR COMMON NODES

We suppose two components A and B have four or more non-co-linear common nodes between them. As discussed previously, if there are four or more common nodes, there is no need for any connecting edge between the components. The coordinate system conversion can be done only with the help of common nodes. For example, in Figure 5, all four common nodes have their coordinates in both coordinate systems of A and B. Our task is to convert the coordinate system of B into $A's$ coordinate system so that both the components have a single global coordinate system. Any node in B can calculate its distance to a, b, c, d using the $B's$ coordinate system because all the common nodes have

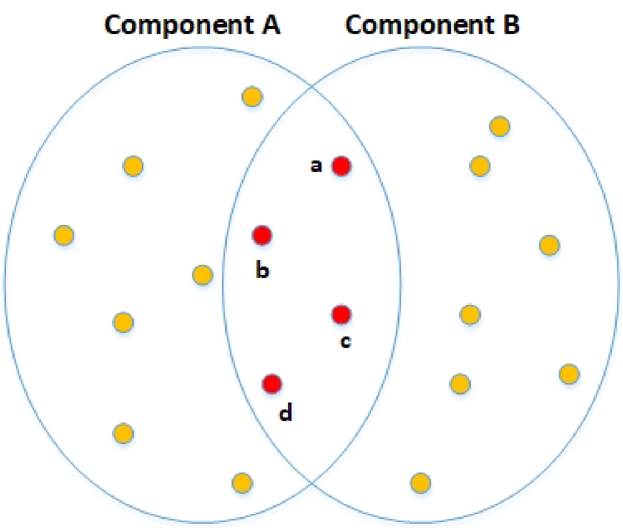

FIGURE 5. Case of four or more common node.

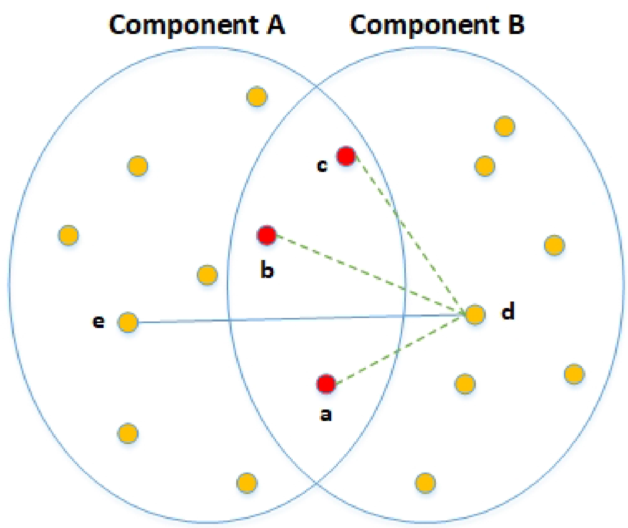

FIGURE 6. Case of three common nodes.

coordinates in B's coordinate system. If we find the distance between each common node and unknown nodes in B, then by multilateration method, the position of the unknown nodes in B can be calculated in $A's$ coordinates with the help of calculated distances.

2) CASE 2: THREE COMMON NODES

We suppose three common nodes exist between two components A and B. More than three common nodes are needed to convert the coordinate system of B to the coordinate system of A, so we have to rely on the information on connecting edges between the components. Two components with three common nodes can be merged as per theorem 1 below.

Theorem 1: Given three common nodes between two components A and B, they are uniquely mergeable if and only if at least one connecting edge exists between a non-common node in B and a non-common node in A.

Proof: Suppose three nodes a, b, c are common between two components A and B as shown in Figure 6. In addition to three common nodes, there is a connecting edge between non-common nodes in A and B. The connecting edge is the measured distance represented by the solid line ed.

We can see that the nodes a, b, c, d are in component B, and all these four nodes have the local coordinates of component B, so based upon the local coordinates, these four nodes can calculate the distance between them, which is represented by the dashed line ad, bd and ed in Figure 6 and also termed as the calculated distance. The nodes a, b, c are common, so they have the local coordinates of component A. Now, if we find the position of node d in the $A's$ coordinates, we have four nodes in component B that have coordinates of components A. This reduces our problem to case 1, and then by using multilateration, the position of all nodes in B can be calculated in $A's$ coordinates. Below is the method to calculate the position of node d in $A's$ coordinates system.

We denote the position of the common nodes a, b, c and non common node e by (x_a, y_a, z_a), (x_b, y_b, z_b), (x_c, y_c, z_c) and (x_e, y_e, z_e) respectively. These positions are according to $A's$ coordinate system. Now let us suppose the location of node d is unknown in $A's$ coordinates and is denoted by (x_d, y_d, z_d). From Figure 6, we can see that we have a total of four distance information, three are the calculated distances, and one is the measured distance, so we can obtain four distance equations below.

$$\begin{cases} (x_e - x_d)^2 + (y_e - y_d)^2 + (z_e - z_d)^2 = d_{ed}^2 \\ (x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2 = d_{ad}^2 \\ (x_b - x_d)^2 + (y_b - y_d)^2 + (z_b - z_d)^2 = d_{bd}^2 \\ (x_c - x_d)^2 + (y_c - y_d)^2 + (z_c - z_d)^2 = d_{cd}^2 \end{cases} \tag{10}$$

In equation 10, we have four distance equations with three unknowns x_d, y_d, and z_d, which can be solved uniquely. After finding the location of node d in the $A's$ coordinates, all the other nodes in B can easily calculate their position in the $A's$ coordinate, as explained earlier.

3) CASE 3: TWO COMMON NODES

We suppose two common nodes exist between two components A and B. Like case 2, in addition to two common nodes, we rely on the information of multiple connecting edges between the components. Two components with two common nodes can be merged as per theorem 2 below.

Theorem 2: Given two common nodes between two components A and B, they are uniquely mergeable if and only if at least five nodes exist in both components, and two nodes should be in each component A and B. In addition, there should be at least two connecting edges between two non-common nodes in A and B

Proof: Suppose nodes a and b are common between two components A and B as shown in Figure 7. In addition to two common nodes, there are two connecting edges between

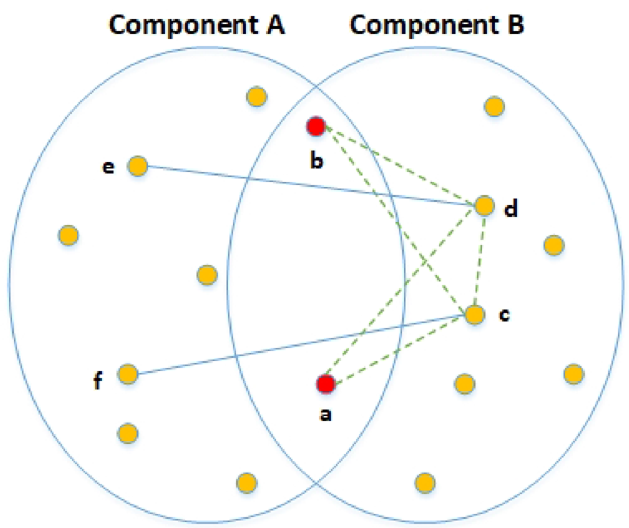

FIGURE 7. Case of two common nodes.

non-common nodes in components A and B. These are the measured distances represented by solid lines fc and ed. We can see that the nodes a, b, c, d are in component B, and all these four nodes have the local coordinates of component B, so based upon the local coordinates, these four nodes have calculated distances between them, which are represented by the dashed line ac, cd, bc, bd and cd as shown in Figure 7. As the nodes, a and b are common, they have the local coordinates of component A. Now, if we find the location of node c and d in the $A's$ coordinates, there are four nodes in component B with coordinates of components A. This will reduce our problem to case 1, and then by using multilateration, the position of all nodes in B can be calculated in $A's$ coordinates. The method is given below to calculate the position of node c and d in $A's$ coordinates.

We denote the position of two common nodes a, b and non common nodes f, e by (x_a, y_a, z_a), (x_b, y_b, z_b), (x_f, y_f, z_f) and (x_e, y_e, z_e) respectively. These positions are according to $A's$ coordinate system. Now let us suppose the position of nodes c and d is unknown in $A's$ coordinate system and is denoted by (x_c, y_c, z_c) and (x_d, y_d, z_d). From Figure 7, we can see that we have a total of seven distance information, five are the calculated distances and two are the measured distances, so we can obtain seven distance equations below.

$$\begin{cases} (x_f - x_c)^2 + (y_f - y_c)^2 + (z_f - z_c)^2 = d_{fc}^2 \\ (x_e - x_d)^2 + (y_e - y_d)^2 + (z_e - z_d)^2 = d_{ed}^2 \\ (x_a - x_c)^2 + (y_a - y_c)^2 + (z_a - z_c)^2 = d_{ac}^2 \\ (x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2 = d_{ad}^2 \\ (x_b - x_c)^2 + (y_b - y_c)^2 + (z_b - z_c)^2 = d_{bc}^2 \\ (x_b - x_d)^2 + (y_b - y_d)^2 + (z_b - z_d)^2 = d_{bd}^2 \\ (x_c - x_d)^2 + (y_c - y_d)^2 + (z_c - z_d)^2 = d_{cd}^2 \end{cases} \tag{11}$$

In equation 11, we have seven distance equations with six unknowns x_c, y_c, z_c, x_d, y_d and z_d, which can be solved uniquely. After finding the location of node c and d in the $A's$ coordinates, all the other nodes in B can easily calculate their position in the $A's$ coordinates, as explained earlier.

4) CASE 4: ONE COMMON NODE

We suppose one common node between two components A and B. Like previous cases, in addition to one common node, we have to rely on the information of multiple connecting edges between the components. Two components with one common node can be merged per theorem 3 below.

Theorem 3: Given one common node between two components A and B, They are uniquely mergeable if and only if there exist more than five nodes in both components, and there should be at least two non-common nodes in component A and three non-common nodes in component B. In addition, there should be at least four connecting edges between two non-common nodes in A and three non-common nodes in B.

Proof: Suppose the only common node a exists between two components A and B as shown in Figure 8. In addition to one common node, there are four connecting edges between non-common nodes in components A and B. These are the measured distances represented by solid lines fc, fb, ec, and ed. We can see that the nodes a, b, c, d are in component B, and all these four nodes have the local coordinates of component B, so based upon the local coordinates, these four nodes have calculated distances between them, which are represented by the dashed line ac, ab, ad, bc, bd and cd as shown in Figure 8. Only one common node a has coordinates in the $A's$ coordinate system, so if we find the position of node b, c and d in the $A's$ coordinate system, then we have four nodes in component B having coordinates of components A. This will reduce our problem to case 1, and then by using multilateration, the position of all nodes in B can be calculated in $A's$ coordinate system. Below is the method to calculate the position of node b, c, and d in $A's$ coordinates.

We denote the position of single common node a and non common nodes f, e by (x_a, y_a, z_a), (x_f, y_f, z_f) and (x_e, y_e, z_e) respectively. These positions are according to $A's$ coordinate system. Now let us suppose the position of nodes b, c and d is unknown in $A's$ coordinate system and is denoted by (x_b, y_b, z_b), (x_c, y_c, z_c) and (x_d, y_d, z_d). From Figure 8, we can see that we have a total of ten distance information; six are the calculated distances, and four are the measured distances, so we can obtain ten distance equations below

$$\begin{cases} (x_f - x_c)^2 + (y_f - y_c)^2 + (z_f - z_c)^2 = d_{fc}^2 \\ (x_f - x_b)^2 + (y_f - y_b)^2 + (z_f - z_b)^2 = d_{fb}^2 \\ (x_e - x_c)^2 + (y_e - y_c)^2 + (z_e - z_c)^2 = d_{ec}^2 \\ (x_e - x_d)^2 + (y_e - y_d)^2 + (z_e - z_d)^2 = d_{ed}^2 \\ (x_a - x_c)^2 + (y_a - y_c)^2 + (z_a - z_c)^2 = d_{ac}^2 \\ (x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2 = d_{ab}^2 \\ (x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2 = d_{ad}^2 \\ (x_b - x_c)^2 + (y_b - y_c)^2 + (z_b - z_c)^2 = d_{bc}^2 \\ (x_b - x_d)^2 + (y_b - y_d)^2 + (z_b - z_d)^2 = d_{bd}^2 \\ (x_c - x_d)^2 + (y_c - y_d)^2 + (z_c - z_d)^2 = d_{cd}^2 \end{cases} \tag{12}$$

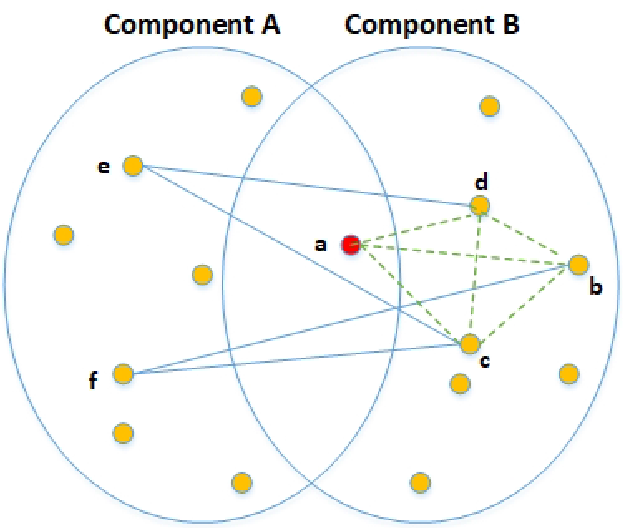

FIGURE 8. Case of one common node.

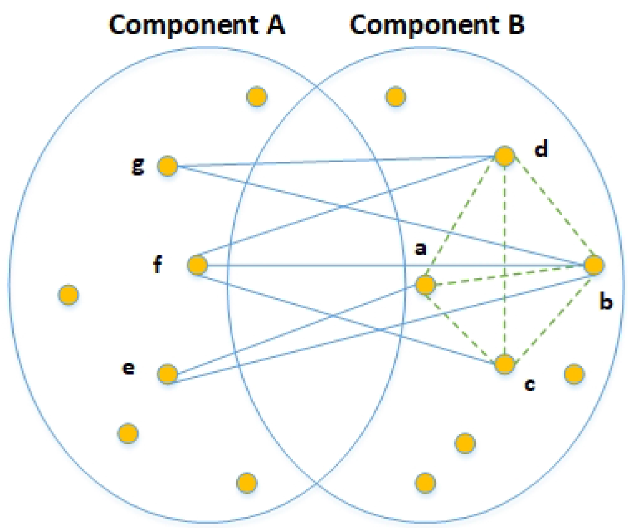

FIGURE 9. Case of no common node.

In equation 12, we have ten distance equations with nine unknowns $x_b, y_b, z_b, x_c, y_c, z_c, x_d, y_d$ and z_d which can be solved uniquely. After finding the location of node b, c and d in $A's$ coordinates, all the other nodes in B can easily calculate their locations in the $A's$ coordinate system as explained earlier.

5) CASE 5: NO COMMON NODES

We suppose two components A and B have one common node. These components can still be merged if there are a sufficient number of connecting edges between the two components. Two components without a common node can be merged as per theorem 4 below.

Theorem 4: Given no common node between two components A and B, They are uniquely mergeable if and only if there exist more than seven nodes in both components, and there should be at least three non-common nodes in component A and four non-common nodes in component B. In addition, there should be at least seven connecting edges between three non-common nodes in A and four non-common nodes in B.

Proof: Suppose no common node exists between two components A and B as shown in Figure 9. There are seven connecting edges between non-common nodes in components A and B. These are the measured distances represented by solid lines $ea, eb, gb, gd, fb, fc,$ and fd. We can see that the nodes a, b, c, d are in component B, and all these four nodes have the local coordinates of component B, so based upon the local coordinates, these four nodes have calculated distances between them which are represented by the dashed line ac, ab, ad, bc, bd and cd as shown in Figure 9. If we find the position of node a, b, c and d in the $A's$ coordinate system, we have four nodes in component B with coordinates of components A. This will reduce our problem to case 1, and then by using multilateration, the position of all nodes in B can be calculated in $A's$ coordinate system. Below is the method to find the location of node a, b, c and d in $A's$ coordinate system.

We denote the position of non common nodes e, f, g by (x_e, y_e, z_e), (x_f, y_f, z_f) and (x_g, y_g, z_g) respectively. These positions are according to $A's$ coordinate system. Now let us suppose the position of nodes a, b, c and d is unknown in $A's$ coordinate system and is denoted by (x_a, y_a, z_a), (x_b, y_b, z_b), (x_c, y_c, z_c) and (x_d, y_d, z_d). From Figure 9, we can see that we have a total of thirteen distance information; six are the calculated distances, and seven are the measured distances, so we can obtain thirteen distance equations below

$$\begin{cases} (x_e - x_a)^2 + (y_e - y_a)^2 + (z_e - z_a)^2 = d_{ea}^2 \\ (x_e - x_b)^2 + (y_e - y_b)^2 + (z_e - z_b)^2 = d_{eb}^2 \\ (x_g - x_b)^2 + (y_g - y_b)^2 + (z_g - z_b)^2 = d_{gb}^2 \\ (x_g - x_d)^2 + (y_g - y_d)^2 + (z_g - z_d)^2 = d_{gd}^2 \\ (x_f - x_b)^2 + (y_f - y_b)^2 + (z_f - z_b)^2 = d_{fb}^2 \\ (x_f - x_c)^2 + (y_f - y_c)^2 + (z_f - z_c)^2 = d_{fc}^2 \\ (x_f - x_d)^2 + (y_f - y_d)^2 + (z_f - z_d)^2 = d_{fd}^2 \\ (x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2 = d_{ab}^2 \\ (x_a - x_c)^2 + (y_a - y_c)^2 + (z_a - z_c)^2 = d_{ac}^2 \\ (x_a - x_d)^2 + (y_a - y_d)^2 + (z_a - z_d)^2 = d_{ad}^2 \\ (x_b - x_c)^2 + (y_b - y_c)^2 + (z_b - z_c)^2 = d_{bc}^2 \\ (x_b - x_d)^2 + (y_b - y_d)^2 + (z_b - z_d)^2 = d_{bd}^2 \\ (x_c - x_d)^2 + (y_c - y_d)^2 + (z_c - z_d)^2 = d_{cd}^2 \end{cases} \quad (13)$$

In equation 13 we have thirteen distance equations with twelve unknowns $x_a, y_a, z_a, x_b, y_b, z_b, x_c, y_c, z_c, x_d, y_d$ and z_d which can be solved uniquely. After finding the position of node a, b, c and d in $A's$ coordinate system, all the other nodes in B can easily calculate their position in $A's$ coordinate system as explained earlier.

V. EXPERIMENTAL EVALUATION

A. SIMULATION SETUP

We provide a thorough study of our proposed component-based localization in this section. For simplicity, we named our algorithm Component-Based 3D Algorithm (CB3L)in the rest of the paper. We use an Intel Core I7 machine with 8GB RAM for performance analysis. Two primary metrics assess the algorithm's performance: the localization ratio, indicating the successfully localized nodes, and the localization error. The localization ratio is the proportion of localized nodes to all nodes in the network. On the other hand, the average difference between all sensor node's calculated positions and their ground truth positions is known as the localization error. For a fair comparison with [26], we express localization error as a multiple of the communication radius R. Our analysis is limited to static networks, where all nodes are stationary. For consistent evaluation, we adopt the same performance parameters as [26], including anchor ratio, node degree, and distance measurement error. In a randomly deployed 3-D sensor network spanning an area of $10R \times 10R \times 10R$, nodes are uniformly distributed. A node can only measure the distance from other nodes if it is placed in its communication range and considered as neighbors of each other. Node degree is controlled by changing the number of deployed nodes in our simulations, ranging from 8 to 13. To assess the proposed CB3L algorithm's performance, we perform extensive simulations. Each simulation involves 100 random network generations for a given set of parameters, ensuring the reported results are averages from multiple trials.

B. IMPACT OF NODE DEGREE

To probe the effect of varying node degrees on the localization ratio of the networks, we fix the anchor ratio at 5% and 10%, respectively, and vary the node degree from 8 to 13. The simulation results are plotted in Figure 10 and Figure 11, respectively, which show that the localization ratio increases with the increase of average node degree. This is because when the average node degree is less, there is insufficient distance information among the nodes, so many nodes remain isolated without joining any component. As a result, the localization ratio is affected. When the node degree increases, more nodes are in the node's communication radius, which increases distance information. Most nodes join any components, reducing isolated nodes to a minimum, which results in an increased location ratio.

The CB3L algorithm outperforms the other state-of-the-art algorithms. In Figure 10, when the anchor ratio is fixed at 5%, we can see that the CB3L algorithm's localization ratio is higher. For a node degree 10, the CB3L algorithm's localization ratio is more than 83% while SQ, UPS, and CBL can only localize 18%, 39%, and 79% nodes, respectively, for the same network simulation parameters.

In Figure 11, the anchor ratio is increased from 5% to 10%, and we can see that the ratio of localized nodes in SQ and UPS increases more rapidly than CBL and CB3L algorithms.

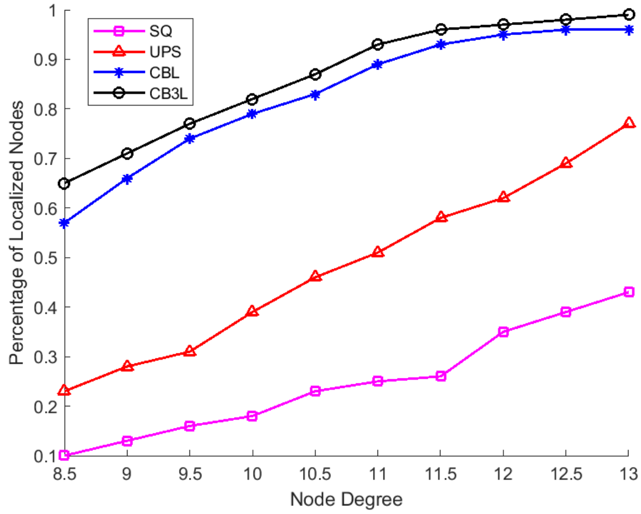

FIGURE 10. Impact of node degree on localization ratio with 5% anchor.

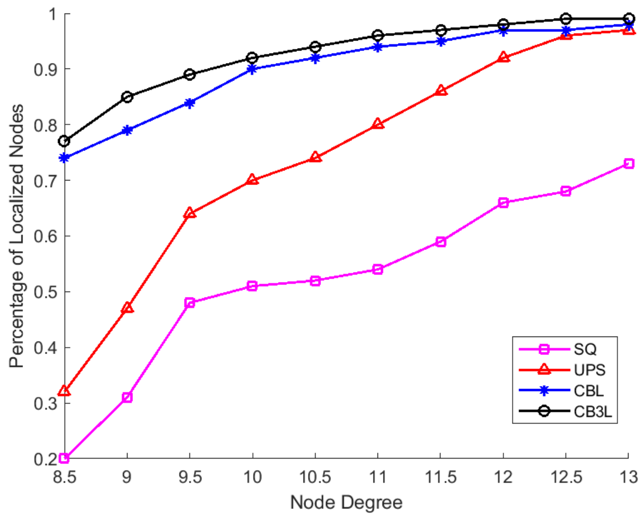

FIGURE 11. Impact of node degree on localization ratio with 10% anchor.

The reason is that in any component-based localization algorithm, only four anchor nodes are sufficient to localize the whole network, provided that none of the anchor nodes is in isolation. So, increasing the anchor nodes above a certain number has no significant effect on any component-based localization algorithm. From both Figure 10 and Figure 11, we can see that when the network is sparse, there is a significant performance difference among the algorithms. However, as the network becomes denser, the performance difference becomes insignificant. This concludes that the CB3L algorithm is more suited for a sparse than a dense network.

Figure 12 describes how the size of a component varies by changing the node degree. We can see that when the network is dense, or the node degree is varied from 8 to 13, there are more nodes in the largest component, which might have a sufficient number of anchor nodes. Hence, that component

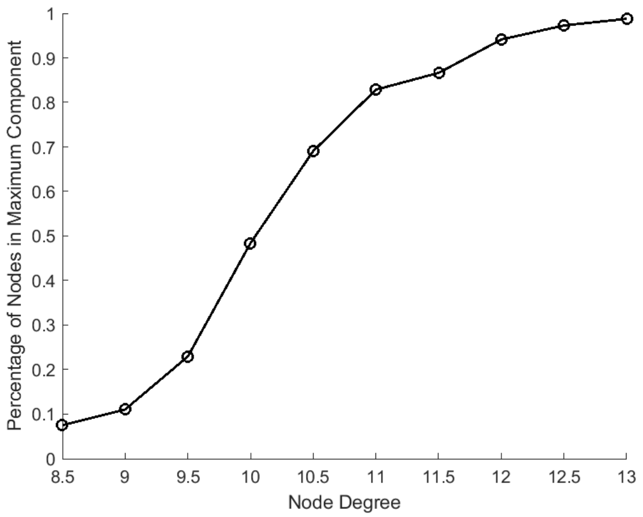

FIGURE 12. Percentage of nodes in largest component.

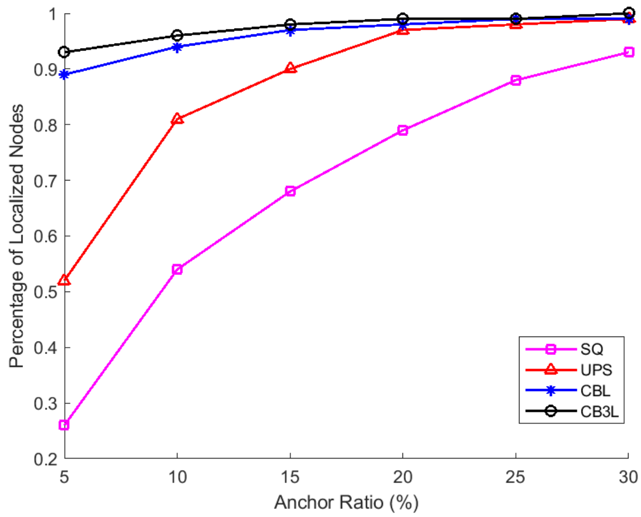

FIGURE 14. Impact of anchor ratio on localization ratio for node degree 11.

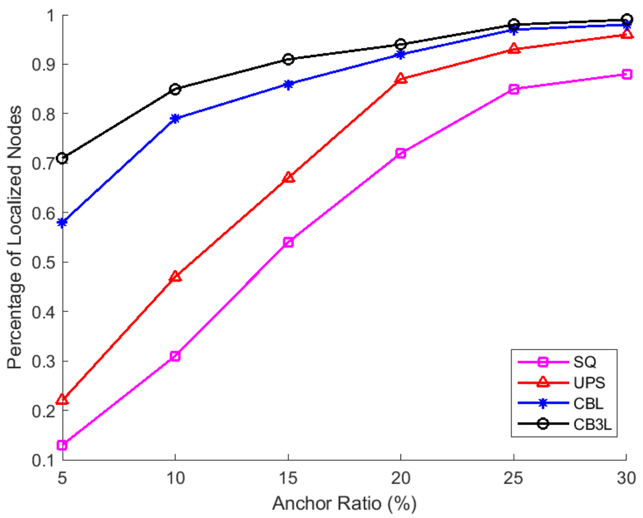

FIGURE 13. Impact of anchor ratio on localization ratio for node degree 9.

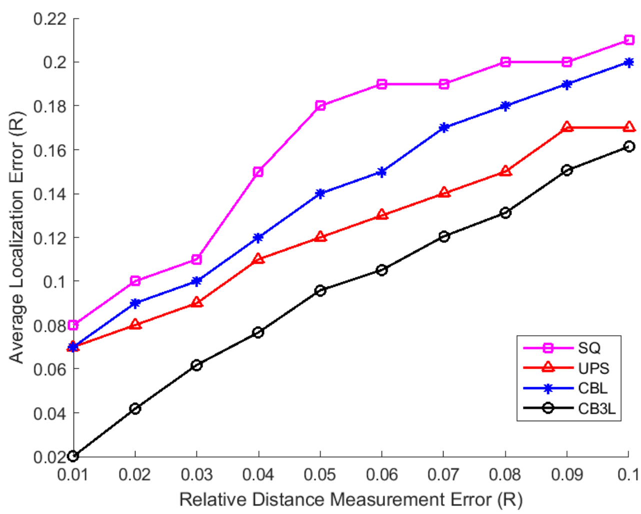

FIGURE 15. Impact of distance measurement error on average localization error.

can be easily localized without any merging process. This also eases the merging and localization process of other remaining components.

C. IMPACT OF ANCHOR RATIO

To probe the effect of varying anchor ratios on the localization ratio of the networks, we fix the node degree at 9 and 11, respectively. The anchor ratio is varied from 5% to 30%, and the simulation results are plotted in Figure 13 and Figure 14, respectively, which show that the localization ratio increases as the anchor ratio increases. The increased ratio is significant in the case of regural 3-D localization algorithms like SQ and UPS. The localization ratio is meager when the anchor ratio is 5%, and it keeps increasing with an increase in the anchor ratio. On the other hand, for the localization ratio of component-based algorithms like CBL and CB3L algorithms, the increase ratio is insignificant. When the anchor ratio is 5% in the CB3L algorithm, the algorithm performs better with a localization ratio of 89%. By increasing the anchor ratio to 30%, there is a slight increase in the localization ratio, which is 91%. We can see that when the node degree is less (figure 13), the localization ratio of the CB3L algorithm is increased with the increase in anchor ratio, but when the node degree is high (figure 14), and by increasing the anchor ratio, there is a negligible effect on localization ratio of the CB3L algorithm. This is because when the node degree is high, most nodes are in a single largest component with enough anchors to localize all the nodes.

D. IMPACT OF MEASUREMENT ERROR

Figure 15 presents the analysis of the localization error with relative distance measurement errors. Due to substantial error propagation caused by the sequential localization process, we notice that the localization error in the sequential algorithms is significant. UPS achieves the best localization error result among sequential algorithms because it is presumed that the depth of the sensor node is known thus, there is zero error in z-axis coordinates. This changes the 3-D localization of UPS into a normal 2-D localization process, which performs better regarding localization error. In contrast to the sequential techniques, the error propagation is suppressed by the CB3L algorithm, because it generates a small number of components and merges them afterward, which experiences much fewer component mergences.

VI. CONCLUSION

In this paper, we address the significant challenge of node localization in sparse 3-D wireless sensor networks and propose a component-based localization method that effectively overcomes the limitations of insufficient anchor nodes and distance information. By dividing the whole network into small overlapping components, and on the basis of common nodes and merging conditions, we merge these components into a large network having a global coordinate system. The proposed approach demonstrates remarkable performance in localizing nodes with a minimum anchor ratio. The extensive simulations validate our algorithm's superiority in sparse 3-D environments. The ability of our algorithm to localize over 83% of nodes with minimum localization error shows its potential as a significant effort in the field of localization of wireless sensor networks. The results also concluded that the proposed algorithm is more appropriate for a sparse network than a dense network. In the forthcoming work, we plan to enhance the work by improving merging conditions and exploiting angle information in the sparse network when we do not have sufficient distance information among nodes.

ACKNOWLEDGMENT

This Research is funded by Researchers Supporting Project Number (RSPD2024R553), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] L. L. D. Oliveira, G. H. Eisenkraemer, E. A. Carara, J. B. Martins, and J. Monteiro, "Mobile localization techniques for wireless sensor networks: Survey and recommendations," *ACM Trans. Sensor Netw.*, vol. 19, no. 2, pp. 1–39, May 2023.

[2] D. W. Wajgi and J. V. Tembhurne, "Localization in wireless sensor networks and wireless multimedia sensor networks using clustering techniques," *Multimedia Tools Appl.*, vol. 83, no. 3, pp. 6829–6879, Jan. 2024.

[3] P. Yadav and S. C. Sharma, "A systematic review of localization in WSN: Machine learning and optimization-based approaches," *Int. J. Commun. Syst.*, vol. 36, no. 4, p. e5397, Mar. 2023.

[4] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Imran, and M. Ur-Rehman, "Challenges, applications, and future of wireless sensors in Internet of Things: A review," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5482–5494, Mar. 2022.

[5] H. Chaitra, G. Manjula, M. Shabaz, A. B. Martinez-Valencia, K. Vikhyath, S. Verma, and J. L. Arias-Gonzáles, "Delay optimization and energy balancing algorithm for improving network lifetime in fixed wireless sensor networks," *Phys. Commun.*, vol. 58, Jun. 2023, Art. no. 102038.

[6] X. Gao, X. Zhu, and L. Zhai, "AoI-sensitive data collection in multi-UAV-assisted wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5185–5197, Aug. 2023.

[7] M. Chaudhary, N. Goyal, A. Benslimane, L. K. Awasthi, A. Alwadain, and A. Singh, "Underwater wireless sensor networks: Enabling technologies for node deployment and data collection challenges," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3500–3524, Feb. 2023.

[8] S. R. Jondhale, R. Maheswar, J. Lloret, S. R. Jondhale, R. Maheswar, and J. Lloret, "Fundamentals of wireless sensor networks," in *Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks*. Cham, Switzerland: Springer, 2022, pp. 1–19.

[9] M. Krishnapriya, G. A. Prasanna, and S. Anbarasu, "Rank-based energy-efficient key management routing for wireless sensor network-based IoT medical sensors," *Wireless Pers. Commun.*, vol. 130, no. 3, pp. 2175–2196, Jun. 2023.

[10] S. Rani, H. Babbar, P. Kaur, M. D. Alshehri, and S. H. Shah, "An optimized approach of dynamic target nodes in wireless sensor network using bio inspired algorithms for maritime rescue," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2548–2555, Feb. 2023.

[11] Z. Lv, D. Chen, H. Feng, W. Wei, and H. Lv, "Artificial intelligence in underwater digital twins sensor networks," *ACM Trans. Sensor Netw.*, vol. 18, no. 3, pp. 1–27, Aug. 2022.

[12] J. Bravo-Arrabal, P. Zambrana, J. J. Fernandez-Lozano, J. A. Gomez-Ruiz, J. S. Barba, and A. García-Cerezo, "Realistic deployment of hybrid wireless sensor networks based on ZigBee and LoRa for search and rescue applications," *IEEE Access*, vol. 10, pp. 64618–64637, 2022.

[13] P. S. Farahsari, A. Farahzadi, J. Rezazadeh, and A. Bagheri, "A survey on indoor positioning systems for IoT-based applications," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7680–7699, May 2022.

[14] M. Elsanhoury, P. Mäkelä, J. Koljonen, P. Välisuo, A. Shamsuzzoha, T. Mantere, M. Elmusrati, and H. Kuusniemi, "Precision positioning for smart logistics using ultra-wideband technology-based indoor navigation: A review," *IEEE Access*, vol. 10, pp. 44413–44445, 2022.

[15] A. R. Basha, "A review on wireless sensor networks: Routing," *Wireless Pers. Commun.*, vol. 125, no. 1, pp. 897–937, 2022.

[16] X. Li, S. Xu, H. Zhao, S. Han, and L. Yan, "An adaptive multi-zone geographic routing protocol for underwater acoustic sensor networks," *Wireless Netw.*, vol. 28, no. 1, pp. 209–223, Jan. 2022.

[17] W. Osamy, A. M. Khedr, A. Salim, A. I. A. Ali, and A. A. El-Sawy, "Coverage, deployment and localization challenges in wireless sensor networks based on artificial intelligence techniques: A review," *IEEE Access*, vol. 10, pp. 30232–30257, 2022.

[18] A. Achroufene, "RSSI-based geometric localization in wireless sensor networks," *J. Supercomput.*, vol. 79, no. 5, pp. 5615–5642, Mar. 2023.

[19] H. Sun and M. Tian, "Improved range-free localization algorithm based on reliable node optimization and enhanced sand cat optimization algorithm," *J. Supercomput.*, vol. 79, no. 18, pp. 20289–20323, Dec. 2023.

[20] C. Ren and L. Liu, "Towards full passive Internet of Things: Symbiotic localization and ambient backscatter communication," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 19495–19506, Nov. 2023.

[21] F. B. Sorbelli, C. M. Pinotti, S. Silvestri, and S. K. Das, "Measurement errors in range-based localization algorithms for UAVs: Analysis and experimentation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 4, pp. 1291–1304, Apr. 2022.

[22] A. Elzanaty, A. Guerra, F. Guidi, D. Dardari, and M.-S. Alouini, "Towards 6G holographic localization: Enabling technologies and perspectives," *IEEE Internet Things Mag.*, vol. 6, no. 3, pp. 138–143, Sep. 2023.

[23] S. Kumari, P. K. Mishra, and V. Anand, "Coverage and connectivity aware deployment scheme for autonomous underwater vehicles in underwater wireless sensor networks," *Wireless Pers. Commun.*, vol. 132, no. 2, pp. 909–931, Sep. 2023.

[24] H. Liu, B. Xu, and B. Liu, "A novel predictive localization algorithm for underwater wireless sensor networks," *Wireless Netw.*, vol. 29, no. 1, pp. 303–319, Jan. 2023.

[25] K. K. Gola, M. Dhingra, B. Gupta, and R. Rathore, "An empirical study on underwater acoustic sensor networks based on localization and routing approaches," *Adv. Eng. Softw.*, vol. 175, Jan. 2023, Art. no. 103319.

[26] X. Liu, J. Yin, S. Zhang, B. Ding, S. Guo, and K. Wang, "Range-based localization for sparse 3-D sensor networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 753–764, Feb. 2019.

[27] 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 Int. Conf. Netw. Protocols*, Oct. 2008, pp. 288–297.

[28] X. Wang, J. Luo, Y. Liu, S. Li, and D. Dong, "Component-based localization in sparse wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 540–548, Apr. 2011.

[29] X. Wang, Y. Liu, Z. Yang, K. Lu, and J. Luo, "Robust component-based localizationin sparse networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 5, pp. 1317–1327, May 2014.

[30] O.-H. Kwon and H.-J. Song, "Localization through map stitching in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 1, pp. 93–105, Jan. 2008.

[31] O.-H. Kwon, H.-J. Song, and S. Park, "The effects of stitching orders in patch-and-stitch WSN localization algorithms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 9, pp. 1380–1391, Sep. 2009.

[32] S. Zhang, S. Yan, W. Hu, J. Wang, and K. Guo, "A component-based localization algorithm for sparse sensor networks combining angle and distance information," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 3, pp. 1014–1034, Mar. 2015.

[33] J. Kashniyal, S. Verma, and K. P. Singh, "A new patch and stitch algorithm for localization in wireless sensor networks," *Wireless Netw.*, vol. 25, no. 6, pp. 3251–3264, Aug. 2019.

[34] J. An, C. Yuen, C. Xu, H. Li, D. W. K. Ng, M. Di Renzo, M. Debbah, and L. Hanzo, "Stacked intelligent metasurface-aided MIMO transceiver design," 2023, *arXiv:2311.09814*.

[35] J. An, C. Yuen, L. Dai, M. Di Renzo, M. Debbah, and L. Hanzo, "Toward beamfocusing-aided near-field communications: Research advances, potential, and challenges," 2023, *arXiv:2309.09242*.

[36] G. Oliva, S. Panzieri, F. Pascucci, and R. Setola, "Sensor networks localization: Extending trilateration via shadow edges," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2752–2755, Oct. 2015.

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

[38] B. D. O. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang, "Graphical properties of easily localizable sensor networks," *Wireless Netw.*, vol. 15, no. 2, pp. 177–191, Feb. 2009.

[39] T. Eren, O. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *Proc. IEEE INFOCOM*, vol. 4. Hong Kong, China, Mar. 2004, pp. 2673–2684.

[40] D. K. Goldenberg, A. Krishnamurthy, W. C. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. D. O. Anderson, "Network localization in partially localizable networks," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 1. Miami, FL, USA, Mar. 2005, pp. 313–326.

[41] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse, and Y. R. Yang, "Localization in sparse networks using sweeps," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2006, pp. 110–121.

[42] J. Fang, M. Cao, A. S. Morse, and B. D. O. Anderson, "Localization of sensor networks using sweeps," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 4645–4650.

[43] M. M. Vijay, J. Sunil, V. G. A. G. Vincy, M. IjazKhan, S. S. Abdullaev, S. M. Eldin, V. Govindan, H. Ahmad, and S. Askar, "Underwater wireless sensor network-based multihop data transmission using hybrid cat cheetah optimization algorithm," *Sci. Rep.*, vol. 13, no. 1, p. 10810, Jul. 2023.

[44] M. Arabsorkhi, H. Zayyani, and M. Korki, "3-D hybrid RSS-AoA passive source localization with unknown path loss exponent," *IEEE Sensors Lett.*, vol. 7, no. 6, pp. 1–4, Jun. 2023.

[45] R. Mani, A. Rios-Navarro, J.-L. Sevillano-Ramos, and N. Liouane, "Improved 3D localization algorithm for large scale wireless sensor networks," *Wireless Netw.*, pp. 1–16, Mar. 2023.

[46] J. Wang, L. Cheng, Y. Tu, and S. Gu, "A novel localization approach for irregular wireless sensor networks based on anchor segmentation," *IEEE Sensors J.*, vol. 22, no. 7, pp. 7267–7276, Apr. 2022.

[47] R. Álvarez, J. Díez-González, P. Verde, R. Ferrero-Guillén, and H. Perez, "Combined sensor selection and node location optimization for reducing the localization uncertainties in wireless sensor networks," *Ad Hoc Netw.*, vol. 139, Feb. 2023, Art. no. 103036.

[48] J. Luomala and I. Hakala, "Adaptive range-based localization algorithm based on trilateration and reference node selection for outdoor wireless sensor networks," *Comput. Netw.*, vol. 210, Jun. 2022, Art. no. 108865.

[49] S. Sikdar, P. Mirgal, and S. Banerjee, "Low-velocity impact source localization in a composite sandwich structure using a broadband piezoelectric sensor network," *Compos. Struct.*, vol. 291, Jul. 2022, Art. no. 115619.

[50] H. Ping, Y. Wang, X. Shen, D. Li, and W. Chen, "On node localizability identification in barycentric linear localization," *ACM Trans. Sensor Netw.*, vol. 19, no. 1, pp. 1–26, Feb. 2023.

[51] X. Luo, W. Zhong, X. Li, and X. Guan, "Bearing rigidity-based localizability analysis for wireless sensor networks," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 6, pp. 526–539, 2020.

[52] C. Lin, G. Han, M. Guizani, Y. Bi, J. Du, and L. Shu, "An SDN architecture for AUV-based underwater wireless networks to enable cooperative underwater search," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 132–139, Jun. 2020.

[53] P. Liu and R. Geldmacher, "An O(max(m,n)) algorithm for finding a subgraph homeomorphic to K4," in *Proc. 11th Southeastern Conf. Combinatorics, Graph Theory Comput.*, vol. 29. Boca Raton, FL, USA: Florida Atlantic Univ., Mar. 1980, pp. 597–609.

[54] M. Bernas, S. Toyonaga, D. Kominami, M. Sugano, L. Mesin, and B. Hasani, *Advances in Wireless Sensor Network*. Wuhan, China: Scientific Research, 2018.

MAZHAR ISLAM received the bachelor's degree in telecommunication engineering from the National University of Computer and Emerging Sciences (FAST-NUCES), Islamabad, Pakistan, in 2010, and the master's degree in telecommunication engineering from the University of Engineering and Technology Taxila, Taxila, Pakistan, in 2016. He is currently pursuing the Ph.D. degree in information and communication engineering with Shenzhen University, China. He is also a Lecturer with the Department of Telecommunications, Hazara University Mansehra, Pakistan. His research interests include wireless sensor networks, information security, and data communication.

MUHAMMAD IKRAM received the bachelor's degree in telecommunication from Hazara University Mansehra, Pakistan, in 2017, and the master's degree in information and communication from the University of Science and Technology Beijing, China, in 2021. Currently, he is a RAN Engineer with Huawei Pakistan. His research interests include wireless communication, wireless sensor networks, and image processing.

MUSAED ALHUSSEIN received the B.S. degree in computer engineering from King Saud University (KSU), Riyadh, Saudi Arabia, in 1988, and the M.S. and Ph.D. degrees in computer science and engineering from the University of South Florida, Tampa, FL, USA, in 1992 and 1997, respectively. He is currently a Professor with the Department of Computer Engineering, College of Computer and Information Sciences, KSU, where he has been a Faculty Member of the Computer Engineering Department, College of Computer and Information Science, since 1997. Recently, he has been successful in winning a research project in the area of AI for healthcare, which is funded by the Ministry of Education, Saudi Arabia. He is the Founder and the Director of the Embedded Computing and Signal Processing Research (ECASP) Laboratory. His research interests include typical computer architecture and signal processing topics with an emphasis on big data, machine/deep learning, VLSI testing and verification, embedded and pervasive computing, cyber-physical systems, mobile cloud computing, big data, eHealthcare, and body area networks.

MUHAMMAD SOHAIB AYUB received the M.S. degree in computer science and the Ph.D. degree from the Lahore University of Management Sciences, Pakistan, in 2015. His research interests include data science, distributed systems, and program analysis.

MUHAMMAD ASAD KHAN received the Ph.D. degree in information and communication engineering from the University of Electronic Science and Technology of China (UESTC). He is currently with the Department of Telecommunication, Hazara University Mansehra, Pakistan. His research interests include mobile network optimization (content caching, network slicing, and small cells), social network analysis, and network immunization.

KHURSHEED AURANGZEB (Senior Member, IEEE) received the B.S. degree in computer engineering from the COMSATS Institute of Information Technology Abbottabad, Pakistan, in 2006, the M.S. degree in electrical engineering (system on chip design) from Linkoping University, Sweden, in 2009, and the Ph.D. degree in electronics design from Mid Sweden University, Sweden, in June 2013. He is currently an Associate Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University (KSU), Riyadh, Saudi Arabia. He has authored and coauthored more than 90 publications, including IEEE/ACM/Springer/Hindawi/MDPI journals, and flagship conference papers. He has obtained more than 15 years of excellent experience as an Instructor and a Researcher in data analytics, machine/deep learning, signal processing, electronics circuits/systems, and embedded systems. He has been involved in many research projects as a Principal Investigator and a Co-Principal Investigator. His research interests include the diverse fields of embedded systems, computer architecture, signal processing, wireless sensor networks, communication, and camera-based sensor networks with an emphasis on big data and machine/deep learning with applications in smart grids, precision agriculture, and healthcare.