Distributed localization algorithm for wireless sensor networks using range lookup and subregion stitching

Farhan Khan | Sing Kiong Nguang

Department of Electrical, Computer, and Software Engineering, Faculty of Engineering, University of Auckland, Auckland, New Zealand

Abstract

One of the ways in which localization algorithms in wireless sensor networks (WSNs) have been categorized is whether they are range-based or range-free. Range-based algorithms use expensive hardware to measure one or more physical quantities and, in turn, use them to localize nodes with greater precision. In contrast, range-free algorithms use coarse-grained quantities like connectivity to localize nodes with limited precision. A middle way between these two approaches can be called a partial range-based approach that can utilize the existing received signal strength indicator (RSSI) readings from sensor nodes to improve the already existing coarse-grained localization methods. Another important consideration in WSNs is that a distributed localization algorithm is more computationally feasible as compared to its centralized counterpart. Keeping these two considerations in mind, a distributed localization algorithm is proposed here which falls in the aforementioned partial range-based category. The proposed algorithm called RangeLookup-MDS first creates subregions using connectivity information only. This is followed by the collection of RSSI readings from individual sensor nodes that are used to perform range lookup for inter-node distance estimates in a lookup table. After that, relative localization in every subregion is performed using multidimensional scaling, and then the relative maps are stitched together to create a consistent (but relative) coordinate system. The algorithm also has the capability to compute absolute coordinates in two-dimensional if the stitching step is executed with at least three non-collinear anchor nodes with known locations. Simulation results on uniform as well as irregular networks of various sizes show that the proposed algorithm provides improved localization accuracy and reduces localization error up to 25% in comparison to a previous partial range-based localization algorithm.

1 | INTRODUCTION

Wireless sensor networks (WSNs) are increasingly being deployed in the world for environmental monitoring (e.g. water monitoring [1]), workplace automation, healthcare monitoring, smart buildings, road safety applications, road network surveillance, vehicle platoon formation, vehicular tracking/networking, and target tracking. In many of the aforementioned applications, it is necessary to accurately localize the nodes with respect to a global coordinate system (or even a relative coordinate system) to report data that is geographically meaningful. Therefore, localization is seen as a basic service in WSNs. Furthermore, other services such as routing (e.g. geographical routing protocols) and, in some cases, topology management (e.g. geographic adaptive fidelity protocol) often rely on location information [2]. All these requirements call for the designing of efficient localization algorithms. The large scale of many of these sensor networks also requires that most of these algorithms or protocols will have to be inherently distributed with hierarchical architectures in order to provide scalability. One very naive solution to furnish location information is to equip each node with GPS but this is too costly, energy consuming, and prone to receiver noise or satellite signal blockages. Also, an ad hoc or random deployment of wireless sensor nodes may require nodes to self-organize themselves relative to each other in some relative coordinate...
system as opposed to an absolute coordinate system. In case if few anchor nodes (with known locations) are available, nodes can also compute absolute locations as well for all of the nodes.

Ascertaining the locations of sensors can be carried out in two main steps namely distance estimation and location computation [3]. There can be a third optional step also which involves applying different optimizations to the second step. The distance estimation step can use sophisticated hardware to measure different physical quantities like received signal strength indicator (RSSI), time of arrival (ToA), or time difference of arrival (TDoA) using ultrasound. Alternatively, distance estimation can also use simple parameters like connectivity resulting in course-grained localization. These two approaches to distance estimation give rise to two different categories of localization algorithms, namely range-based and range-free categories. A middle way between these two approaches will be a partial range-based approach. As every WSN node is fitted with radio communication module and WSN operating systems report RSSI as well as link quality indicator values, RSSI values can be employed in a practical way (like range quantization) to estimate the distance between nodes, and this estimate can be used as input to a distributed localization algorithm [4].

The work proposed here is a distributed localization algorithm called RangeLookup-MDS for computing the locations of sensor nodes in a two-dimensional (2D) map when only the RSSI values between one-hop neighbours in a small subregion are known. The algorithm has four major steps. The first step of the algorithm involves forming overlapping subregions which are composed of one hop-neighbours. The minimum overlapping requirement in this algorithm is three non-collinear common nodes between two subregions for a 2D map. This same step also designates one node as a head node for the subregion. This step can be accomplished using any of the available cluster establishment algorithms like ACE [5] or HEED [6]. In the simulation, a very simplified node degree-based algorithm is used to form clusters because it also provides a straightforward criterion for the selection of head node inside each cluster. The second step of the algorithm involves each subregion head node to collect all the inter-node RSSI measurements from the member nodes, use a simple lookup table to convert RSSI measurements into distance estimates, and create an inter-node distance matrix. The third step involves applying the multidimensional scaling (MDS) technique on the distance matrices to form local relative maps of the member nodes resulting in as many local maps as there are subregions. The fourth step of the algorithm involves map alignment using least-squares fitting in which all the individual local maps are aligned with each other step-by-step starting from an initial (starting) local map. This results in a consistent relative map of all the overlapping subregions. The fourth step uses rotation and translation to align the overlapping local maps with the initial local map. There is one worth-mentioning point for the fourth step of the algorithm and which is that if three non-collinear anchor nodes are present in the initial local map, a 2D absolute map can be obtained for all the overlapping subregions.

The contributions of this study are as follows:

- A lookup-based method has been proposed to establish the distance matrix in which the RSSI values are used to index into a predefined lookup table to give an estimate of range. Since this range lookup is being performed for a comparatively smaller subregion, the effects of shadowing will be less as compared to other methods that try to localize all the sensor nodes in a centralized scheme [7,8]. Second, the range lookup method is much faster and computationally economical for simple WSN nodes as compared to other methods that try to actually perform RSSI to distance conversion [9]. The authors believe that it is the first time such a lookup-based method is being used for location computation although it is common in other applications of WSNs like routing.
- The proposed algorithm first organizes all the nodes in subregions and elects a head node for every subregion. This organization is suitable for heterogeneous WSN applications like traffic monitoring and road network surveillance, where some WSN nodes have superior computation and storage capabilities in comparison to sensing-only nodes. But the proposed algorithm is equally suitable for fully homogeneous WSNs also.
- To reduce the computational complexity, MDS has been used in smaller subregions instead of a very large single network graph. This makes the application of MDS computationally feasible, allowing all the head nodes of subregions to compute the local maps in parallel.
- The proposed algorithm uses a least-squares based fitting algorithm to stitch small local maps together and yields a consistent relative map of the whole sensing domain. It is also the authors’ belief that Huang and Arun’s algorithm [10] is being used first time for such distributed stitching of the local maps. However, more complex algorithms have been used previously [11].
- Unlike previous relative localization methods which use the existing anchor nodes in the last step [7], the proposed algorithm can accommodate the available anchor nodes quite early and at the very start of the peer-to-peer alignment process which allows the alignment process to reveal the global map without any additional last step of correction. Similarly, one heuristic is proposed to incorporate all the anchor nodes as part of the same subregion early which can easily serve as the initial point of the alignment process.

The rest of this study is organized as follows. In Section 2, a very broad classification is presented for the most famous localization algorithms. Section 3 outlines the problem formulation very briefly. Section 4 discusses the four main steps of the proposed algorithm which includes construction of a local map of a subregion from estimated range (or distance) information followed by the calculations required to align two overlapping maps. In Section 5, simulation results of RangeLookup-MDS for the absolute case on two different network topologies are presented along with its comparison with RangeQ-MDS. Section 6 discusses performance analysis
for the relative case, followed by a discussion of some issues and refinements in Section 7. Experimental validation is discussed briefly in Section 8. Finally, the study is concluded in Section 9.

2 RELATED WORK

Localization is a very widely studied topic in WSN research [12]. Many centralized as well as distributed localization algorithms have been reported in the literature [13]. A very broad classification of localization algorithms, adapted from [12,13] is shown in Figure 1, which basically shows five main classes of localization algorithms which are namely centralized, range-based anchor-based distributed, range-free anchor-based distributed, anchor-free relaxation-based distributed, and anchor-free coordinate system stitching. The reader should note that there can still be overlapping among these classes of algorithms as this is not rigid but flexible classification. This classification is also not exhaustive as the number of algorithms is numerous.

The first class of algorithms, that is, centralized algorithms, require sensor nodes to send the collected inter-node ranging (or connectivity) information to a sufficiently powerful central controller which uses a central programme to solve for the locations. The outcome of the programme is absolute or relative locations of nodes which are then sent back to participating nodes. Two examples of centralized algorithms are MDS-MAP(C) proposed by Shang et al. in [7] which uses classical MDS to solve for relative locations and semidefinite programming approach proposed by Doherty et al. in [8]. The centralized algorithms do not scale well and require significant running times as the running time is dependent on the number $n$ of nodes participating in the localization process.

The second class of algorithms, that is, range-based anchor-based distributed algorithms, is the most basic and the most popular class of algorithms. These algorithms are called distributed because all the required location computation is done by the sensor nodes themselves. In this second class, a node with unknown location self-localizes itself using ranging information (distance or angle) to three or more sensor nodes that already know their absolute locations (called anchors/beacons) and subsequently using this ranging information to estimate its absolute location. The three most popular examples are trilateration, triangulation, and multilateration [3] as shown in Figure 2.

The ranging information is collected using sophisticated hardware and involves either ToA, TDoA, angle of arrival (AoA), or simply RSSI, which indicates basic connectivity. A very broad classification of the most popular localization-related measurements is shown in Figure 3. ToA involves measuring either one-way or round-trip propagation times and then using them to estimate the distance between the sender and receiver. TDoA requires specialized hardware because

![Figure 1](image1.png)

**FIGURE 1** A broad classification of localization algorithms. Adapted from [12]

![Figure 2](image2.png)

**FIGURE 2** Range-based anchor-based localization methods: (a) triangulation, (b) trilateration, and (c) multilateration. Adapted from [3]
sender and receiver exchange radio and sound messages to compute inter-node distance. AoA data is collected using specialized radio or microphone arrays which enables a receiver node to find out the direction of a sender node. As every sensor node is equipped with a radio transceiver, it can also be used to estimate inter-node distance. This is the idea behind RSSI ranging which utilizes the strength of the received radio signal to estimate inter-node distance. Another very intuitive way to use radio presence is to use it to gather simple connectivity information among neighbouring nodes, that is, which nodes are in the communication range of others. Also, another way in which range estimates can be inferred is to use phase difference measurements. The two most important techniques which employ basic laws of physics to determine ranging information are near field electromagnetic ranging (NFER) and radio interferometric measurements (RIMs). NFER infers range estimates by utilizing the near field phase relationship of electric and magnetic components, while RIM infers ranging information by exploiting radio frequency interference of two waves emitted from two reference nodes at slightly different frequencies by measuring the relative phase offset. In most of the range-based algorithms, after gathering sufficient ranging information (like ranging to three known anchors), the node which is interested in self-localization first creates an overdetermined system of equations and later solves that system of equations to determine an approximate location for the unlocalized node in question.

The third class of algorithms, that is, range-free anchor-based distributed algorithms, is named because the nodes use their location information provided by anchors to estimate their own location and may or may not use expensive hardware/ranging methods other than connectivity and reception of location information from sensors. This class includes hop-count-based, area-based, and neighbourhood-based approaches. One example of a neighbourhood-based approach is the use of centroid algorithm in which a node estimates its location by computing the centroid of all the known anchor locations \[14\]. An example of an area-based approach is Approximate Point in Triangle (APIT) \[15\]. APIT uses a novel area-based approach, in which nodes are assumed to be able to hear a considerably large number of anchors. APIT does not assume that nodes can range to these anchors. Instead, every node creates a number of anchor triangles, where an anchor triangle is a triangle formed by three arbitrary anchors. The node then decides whether it is inside or outside a given triangle by comparing signal strength measurements with its nearby non-anchor neighbours. Once this process is complete, the node simply finds the intersection of the anchor triangles that contain it. The node chooses the centroid of this intersection region as its position estimate. The hop-count based approach uses the concept of gradient propagation in which each of the anchors propagates a gradient (which is the hop count to reach the designated anchor node) through the network which serves as the shortest path range among the known anchors and the still unlocalized nodes. The nodes later use this received gradient to compute their own locations using multilateration. Two different refinements have been proposed in \[16\] (called DV-Hop) and \[17\] (called gradient multilateration) for refining the hop count estimates to further improve localization accuracy. Both these refinements use iterative improvement algorithms.

The fourth class of algorithms, that is, relaxation-based anchor-free distributed algorithms, approach localization by using a coarse algorithm to roughly localize nodes in the network. This coarse algorithm is followed by a refinement step, which typically involves each node adjusting its location to optimize a local error metric. By doing so, these algorithms hope to approximate the optimal solution to a network-wide metric that is the sum of the local error metric at each of the nodes. An example of this approach is the spring relaxation method proposed in \[18\].

The fifth class of algorithms, that is, coordinate system stitching-based anchor-free distributed algorithms, divide the network into small overlapping subregions, each of which creates an optimal local map. Finally, the subregions use a peer-to-peer process to merge their local maps into a single global map. In theory, this global map approximates the global optimum map. There are four well-known algorithms for achieving coordinate system stitching in sensor localization. In \[7\], a distributed algorithm called MDS-MAP(P) has been proposed which is a distributed version of centralized MDS-MAP(C). Unlike the centralized MDS-MAP(C), MDS-MAP(P) allows individual nodes to first compute their own local maps called patches and then these patches are merged with the patches of other nodes using an incremental process. This incremental process of merging starts with a random node and then proceeds to merge with the node having the most number of common nodes with the initially chosen node. This process is continued one node at a time, growing the network to cover all the nodes in a single relative map. In \[11\], Meertens et al. propose to obtain a relative coordinate system of the whole sensing filed by following a two-step process. In the first step, each node creates a spatial map using one-hop neighbours by applying a simple trigonometry-based map formation heuristic using three nodes. After these three nodes are defined in a relative coordinate system, multilateration is used to iteratively add additional nodes to the map. In the second step, two spatial maps are aligned with each other using at least three non-collinear common nodes by computing an isometric coordinate transformation such that the sum of the differences in
locations of common nodes is minimized. The second step is then used in a peer-to-peer alignment process to establish a common coordinate system. One problem with this scheme is that it can render some of the nodes to be orphan because they do not have sufficient nodes common with other nodes. In [29], Moore et al. introduce the idea of robust quadrilaterals with the objective of avoiding flip ambiguities that affect localization calculations. Instead of using three arbitrary nodes to define a local map as in [11], Moore et al. start with robust quadrilaterals which is a fully connected set of four nodes where each sub-triangle is also robust. The idea is that the points of a robust quadrilateral can be placed correctly with respect to each other without flips (reflections). Once an initial robust quadrilateral is selected, simple multilateration can be used to add any node that is connected to three of the four vertices of the initial robust quadrilateral. In [20], Ji et al. use MDS to form local maps and apply an iterative version of MDS to compensate for missing inter-node distances. Their distributed localization algorithm initializes flooding from a starting anchor and ends at an ending anchor. The ending anchor passes its location back to the starting anchor.

The localization algorithm proposed here and which is being referred to as RangeLookup-MDS because of its use of MDS in creating local maps, also falls in this fifth class of algorithms which is the distributed coordinate system stitching class. In [12], Bachrach et al. have noted down the following two impressive reasons for coordinate system stitching-based localization.

- They are inherently distributed, and
- They allow the use of computationally expensive local map algorithms (like robust quadrilaterals or multidimensional scaling) which are difficult to use in the centralized algorithms belonging to the first class.

Additionally, a partial range-based approach called RangeQ-MDS is proposed in [21] by Li et al. for leveraging RSSI values to greatly improve the localization process. Li et al. use range quantization in a two-step process in which the RSSI information is ordered according to increasing range in the first step. The second step involves fragmenting the one-hop range into \( s \) quantities of smaller values and the appropriate range values are assigned to the appropriate node group. The second step may use a linear, max-min, or area-proportional model. In [4], a localization algorithm called SRang eQ is proposed which combines the range quantization method of [21] with the APS algorithm of [16] to improve localization accuracy. The scheme proposed here does not use range quantization but instead employs range lookup to estimate inter-node distances using RSSI readings to index into a lookup table maintained at every subregion head node. Although the use of MDS for localization is not new [4,7,21], it has garnered renewed interest among researchers because of its capability to localize nodes inside a relative coordinate system even in the toughest of the terrains. A very recent and comprehensive survey of MDS-based techniques for localization is presented by Saeed et al. in [22] which discusses how the various types of MDS-based localization methods have been applied over the years in WSNs, VANETs, and underwater sensor networks.

More recent trends in the localization area focus on three-dimensional (3D) localization, use of social network analysis, localization for mobile sensor nodes, and so on. For example, in [23], a 3D localization scheme for WSNs based on the well-known parametric loop division algorithm has been proposed which localizes a sensor node in a region bounded by a network of anchor nodes and iteratively shrink that region towards its centre point. In [24], Ahmad et al. have used social network analysis (SNA) to study one of the main properties of SNA which is being referred to as centrality. Ahmad et al. have proposed closeness centrality (CC) as a measure to denote the importance of the node inside the network due to its geo-location to other nodes and also to choose the node with the highest CC value as cluster head which then starts the multilateration process to localize with the help of anchor nodes. In [25], Hamouda et al. have proposed to localize a mobile sensor node by converting the problem into a zoning localization problem with the goal to zero-in to the zone where the mobile sensor resides at any instant. Their approach involves the usage of the belief functions theory to define an evidence framework for estimating the most probable zone for the sensor in question. In [26], Alishamaa et al. have proposed a node localization method based on a sine–cosine algorithm to deal with the flip ambiguity problem which occurs when the connectivity graphs of the sensor networks are not ‘rigid’ due to which the resulting localization of the sensor network graph is non-unique. Use of RSSI as an inexpensive ranging method is garnering new interest among researchers. For example, in [27], Pagano et al. have introduced a confidence parameter for improving accuracy which is based on the standard deviation of the RSSI values in addition to an anchor node selection strategy in order to deal with the impediment of non-line-of-sight communication between sensor nodes.

Two different approaches to localization have been proposed in [28,29]. In [28], Mishra et al. propose to calculate virtual coordinates of the sensor nodes deployed in a 3D area consisting of a single large obstacle. They propose a distributed approach with four anchors for providing localization of the deployed nodes. Their approach uses region division concept for locating virtual coordinates with respect to the obstacle. Each node performs self-localization and is responsible for computing its coordinates. They also consider the possibility of some nodes not receiving the minimum number of required signals from the anchors and propose to use one-off dissemination of the calculated data for each sensor node in their algorithm. In [29], an improved version of the centralized MDS-MAP(C) algorithm called IMDS is introduced which addresses the problem of using all-pairs shortest paths algorithm of MDS-MAP(C) by introducing a heuristic approach to solve the problem in the first step. Their approach uses simple geometry to find the approximate distance between two nodes without using shortest path distance between two nodes and instead relying on computing path using sines and cosines on a curve where the distant node might lie. For a rigid
Secure localization is getting sufficient attention from the researchers. In [30], Liu et al. propose a compressive-sensing-based localization approach to minimize the effect of hostile nodes on secure localization. Their approach uses a multi-region offline technique to detect hostile nodes and further employs a preprocessing method to weigh and balance the roles of anchors. Optimization methods have traditionally played a paramount role in relative/absolute localization in WSNs. In [31], Mohar et al. provide a brief review of different optimization algorithms like genetic, cuckoo search, bio-inspired, particle swarm, firefly, and so on, which have been used in the WSN localization.

The next section presents the problem formulation.

3 PROBLEM FORMULATION

The problem that is addressed here is as follows:

Given a collection of fixed sensor nodes in the (Euclidean) plane in which each node has received some of the RSSI readings between itself and its nearby neighbouring nodes within its communication radius $R$ and also between some pairs of its neighbouring nodes, the fixed sensor nodes are required to first arrange themselves in overlapping subregions based on one-hop neighbours designating one node as head node for each subregion. Each subregion consisting of sensor nodes within its communication radius is to construct a local spatial map showing relative locations of neighbouring nodes such that the map is approximately consistent. The subregions are also required to stitch their local relative maps in a globally consistent map using a peer-to-peer alignment process.

4 PROPOSED ALGORITHM

The proposed approach to solving the aforementioned localization problem has four main elements:

- **Subregion formation using one-hop neighbours:** Using the basic radio connectivity within their communication radius, the nodes first exchange connectivity information with other neighbouring nodes detailing which nodes are in the neighbourhood of which other nodes. This basic radio connectivity information is then used by nodes to organize themselves in small overlapping subregions (using any clustering algorithm depending on the computation complexity) consisting of one-hop neighbours and designate one node inside every subregion as a head node for the respective subregion (using any election algorithm/subregion characteristic or capability like node-degree etc.) [11].

- **Range lookup using RSSI values:** The collected RSSI readings from member nodes in every subregion are used to index into a lookup table at the subregion head node to estimate the inter-node distances and create distance matrices which can be used by the MDS process in the next step. The maintained lookup table is technology-specific, and in the proposed algorithm's case, it is calculated before deployment using the sensor node type and saved inside the sensor nodes [32]. Different sensor platforms will involve different lookup tables depending on the technology used for the radio transceiver.

- **Local map formation for each subregion using multidimensional scaling:** Each subregion head node uses the inter-node distance information to construct a map of its neighbourhood using MDS and passes the locations to other member nodes in the same subregion. For any given subregion, its map can only be determined up to a translation, rotation and reflection (unless other sources of information are used to fix the locations of some nodes) [7].

- **Alignment and stitching of local maps using least-squares fitting:** For any given pair of maps that have at least three non-collinear nodes in common, the locations of the nodes that appear in both maps can be used to calculate an isometric coordinate transformation (a translation + rotation) using least-squares fitting that approximately aligns the maps [10]. If this alignment is performed using three anchor nodes, absolute coordinates are obtained, otherwise relative coordinates are obtained.

Figure 4 shows the overall flow of the proposed algorithm in the form of a flowchart. As shown in the flowchart, the subregion formation step, which is the first step, is executed by all sensor nodes. After that, the range lookup step for RSSI to distance conversion, the local map formation step, and the stitching of local maps step is mainly carried out by the subregion head nodes. The stitching step also involves communication with nodes which are common between two subregions.

![FIGURE 4](image-url)
In the following four subsections, details of the aforementioned steps are provided.

4.1 | Subregion formation using one-hop neighbours

The first step of the localization algorithm involves establishing clusters (called subregions in this work) and designating one of the nodes as the head node of the subregion. To establish subregions, a new clustering algorithm is not proposed but a modified version of an existing node degree-based clustering algorithm called node degree algorithm has been used. The authors believe that proposing a new clustering algorithm can be an interesting future direction to follow.

Using the node degree algorithm results in establishing overlapping subregions. This overlapping of subregions allows the fourth step of the algorithm to initiate a peer-to-peer stitching process using the common nodes between two overlapping subregions. Using the degree of connectivity as a metric for which nodes to elect as cluster-heads has been proposed by Gerla et al. in [33]. The choice of using a node’s degree of connectivity allows nodes in a dense area of the network to be elected first as cluster-heads. The nodes maintain association with a primary subregion and send and receive information to the head node as their main contact but in addition all the sensor nodes who have overlapping with other subregions also maintain a secondary contact with the head nodes of those subregions too. This facilitates the stitching process of Step 4. The basic node degree-based algorithm works as follows:

Each node broadcasts the list of nodes that are in its communication range (including itself). Following are the steps that are taken to create subregions and elect head nodes:

- A node is elected as a subregion head node if it is the most highly connected node of all its ‘uncovered’ neighbour nodes (in case of a tie, lowest ID node is elected to break the tie).
- A node that has not elected its subregion head node yet is an ‘uncovered’ node, otherwise it is a ‘covered’ node.
- A node that has already elected another node as its subregion head node gives up its role as a subregion head node.

Based on the above steps/rules, overlapping subregions are created and the second step of the algorithm is initiated. One important point to note here is that the algorithm does not require to limit to only one type of clustering algorithm. Rather, any general cluster establishment algorithm like ACE [5] or HEED [6] which results in overlapping clusters can be used. The minimum requirement of overlapping is three non-collinear common nodes between subregions for 2D localization.

For the uniform network, a rectangular field of 30 × 6 m^2 was chosen. For the irregular network, the chosen sensing field of 15 × 20 m^2 included an obstruction of 9 × 6 m^2 which represents a physical obstacle in which no sensing nodes can be placed. This was done to create a C-shaped irregular network. After applying the node degree-based algorithm, five subregions were formed in both configurations. Similar networks were generated for the outdoor scenarios with sizes of 400 × 80 m^2 and 200 × 300 m^2. In case, if some of the generated locations are not flexible enough to get covered by the subregion formation algorithm, they might become orphan nodes. A simple solution to deal with orphan nodes is provided in Section 7.6.

The outcome of this subregion formation step is used as input to the next step.

4.2 | Range lookup using RSSI values

The second step of the algorithm is mainly concerned with the setting up of a distance matrix for each and every subregion which is given as input to the third step. The distance matrix is created in a very intuitive way by range lookup. This step is technology dependent because different technologies for radio interfaces involve different lookup tables. The assumption in this step is that all the sensor nodes in question are using the same radio interface, and to some extent, it is a homogenous network as far as the radio interface is concerned.

The rationale behind utilizing the RSSI for ranging is very simple because the energy of a radio signal is inversely proportional to the square of the distance between the source node and the destination node. Therefore, a node that is a recipient of a radio communication should ideally be capable of using the received signal strength to estimate (albeit coarsely) its distance from the transmitting node [12,13].

First, all the inter-node RSSI values are collected by the head node of the subregion. It involves some communication but since the size of the subregion is small, the number of exchanged measurements are limited in number. The head node first creates an \( n \times n \) RSSI matrix where all the RSSI readings are stored. If any RSSI reading is missing, the head node fills it up with twice the communication radius \( R \). Another way to tackle the missing reading problem is to use an all-pairs shortest path algorithm which will be discussed in Section 7.10. After the creation of this RSSI matrix, range lookup is performed using a lookup table. The small lookup table maintained at the head node of the subregion is used to substitute every RSSI value with a range estimate in units of metres. This converts the RSSI matrix into an approximate distance matrix which can be used as input to the third step.

The RSSI values received from neighbouring nodes can be greatly improved by receiving several RSSI readings from neighbours before finalizing an RSSI value, and then performing an averaging operation on the received RSSI values. This makes the distance estimation even more accurate and closer to real distance value as the effects of reflection are likely to be smoothed out during the averaging operation.

As this will be discussed later in Section 5, two different range lookup tables have been used in the simulations for indoor and outdoor scenarios. The one used for indoor scenario is shown in Table 1 and the one used for outdoor scenario is
shown in Table 2. After looking at both the tables, it is clear that one possibility is to use the interpolation technique if there are large intervals corresponding to a given distance. It will result in creating more range intervals and also in extending the size of the lookup table. Interpolation will always be an option if the values are spread too much but if the values for different distances are very close to each other, interpolation will be less beneficial. Another issue is that the use of interpolation will improve accuracy but at the cost of increased storage space requirements for sensor head nodes which will be problematic, especially for indoor sensor technologies like IEEE 802.15.4. The authors think that interpolation will be more beneficial in the case of outdoor technologies like dedicated short-range communication (DSRC), whose hardware does not have limited space like the indoor nodes and the range intervals are also suitably wide to accommodate interpolated values. In addition, extrapolation can also be an option at the two extreme ends of the lookup table.

### 4.3 Local map formation for each subregion using multidimensional scaling

Local map formation for a small subregion can be accomplished in a very straightforward manner by using MDS at every subregion head node. This step assumes that nodes have already exchanged inter-node RSSI measurements with each other in step 2, and the subregion head node has used those RSSI measurements to create a distance matrix of size $n \times n$ where $n$ is the number of sensor nodes in the subregion under consideration. In step 3, the subregion head node executes the MDS using the distance matrix of inter-node distances created in step 2. Since all the subregion head nodes have their respective distances matrices, therefore, this step 3 for all subregions can be executed in parallel taking time with a computational complexity of $O(1)$ theoretically, although MDS has a computational complexity of $O(n^3)$.

MDS is one of the most popular dimensionality reduction techniques for the analysis of dissimilarity (or similarity) data related to a set of data points so that the spatial structure between those data points is properly displayed. The main objective of MDS is to create a configuration of data points in a lower-dimensional space such that the dissimilarities between points in the higher dimensional space is depicted with some degree of exactness by the distances between points in the lower-dimensional space. This means that dissimilarity values that are closer to each other in a higher-dimensional space are depicted closer in the lower-dimensional space also. There are also other dimensionality reduction techniques available, for example, factor analysis.

Many aspects of MDS originated initially in social sciences research. The two most important types of MDS are metric type MDS (Classical or Scaling by MAjorizing a COmplicated Function [SMACOF]) or non-metric type MDS (Kruskal’s algorithm for isotonic regression). The point that is of highest practical importance is that if the dissimilarities in the actual space and the distances between points are assumed to be Euclidean, there is a closed-form solution that allows to depict the configuration of data points in a $d$-dimensional space. This is called the Classical MDS approach. As opposed to Classical MDS closed-form solution, SMACOF algorithm requires an iterative majorisation approach [34]. In contrast to metric MDS

---

### Table 1

| RSSI range (dBm) | Estimated distance     |
|------------------|------------------------|
| >= -54           | 5 ft (1.524 m)         |
| -55 to -56       | 7.5 ft (2.286 m)       |
| -57 to -58       | 10 ft (3.048 m)        |
| -59 to -60       | 12.5 ft (3.81 m)       |
| -61 to -62       | 15 ft (4.572 m)        |
| -63 to -64       | 17.5 ft (5.334 m)      |
| -65 to -66       | 20 ft (6.096 m)        |
| -67 to -68       | 22.5 ft (6.858 m)      |
| -69 to -70       | 25 ft (7.620 m)        |
| -71 to -72       | 27.5 ft (8.382 m)      |
| -73 to -74       | 30 ft (9.144 m)        |
| -75 to -76       | 32.5 ft (9.906 m)      |
| -77 to -78       | 35 ft (10.668 m)       |
| -79 to -80       | 37.5 ft (11.430 m)     |
| -81 to -82       | 40 ft (12.192 m)       |

### Table 2

| RSSI range (dBm) | Estimated distance (m) |
|------------------|------------------------|
| >= -24           | 10                     |
| -25 to -29       | 20                     |
| -30 to -34       | 30                     |
| -35 to -39       | 40                     |
| -40 to -41       | 50                     |
| -42 to -44       | 60                     |
| -44 to -47       | 70                     |
| -48 to -49       | 80                     |
| -49 to -50       | 90                     |
| -51 to -52       | 100                    |
| -53 to -54       | 110                    |
| -55 to -56       | 120                    |
| -57 to -58       | 130                    |
| -59 to -60       | 140                    |
| -61 to -62       | 150                    |

Abbreviations: DSRC, dedicated short-range communication; RSSI, received signal strength indicator.
type, non-metric type MDS is more appropriate in those cases where the original data are ordinal in nature and have been measured on a scale where the ranking of data is more crucial and not their dissimilarities [34].

In Classical MDS approach, the function \( f(.) \) which relates dissimilarities between points and Euclidean distances between data points is the identity function, so there exists a mapping such that between two given points \( i \) and \( j \), \( d_{ij} = \sigma_{ij} \) where \( \sigma \) represents dissimilarity and \( d \) represents Euclidean distance. This Classical MDS approach originated with Young and Householder [35] and later, Gower demonstrated the importance of Classical MDS and referred to it with the name principal coordinates analysis due to the reason that this also involves ideas very similar to those employed by principal components analysis approach [34]. Principal coordinates analysis and Classical MDS are one and the same approach, and they are now being categorized under the category of metric scaling [36].

The Classical MDS algorithm is based on the fact that the coordinate matrix \( X \) can be derived by eigenvalue decomposition from the scalar product matrix \( B = XX^T \). The problem of setting up matrix \( B \) from the dissimilarities matrix \( \Delta \) is solved by multiplying the squared dissimilarities with the centering matrix \( \bar{H} = I - n^{-1}11^T \). This step is referred to as double centering.

Following are the steps of the Classical MDS method:

- Using the matrix of dissimilarities (also called distance matrix), \( \Delta \), set up the matrix of squared dissimilarities \( Q \), where each element of \( Q \) is given by:
  \[
  q_{ij} = \frac{1}{2} \sigma_{ij}^2 
  \]  
  (1)

- Find the centering matrix \( \bar{H} \) using the following:
  \[
  \bar{H} = I - n^{-1}11^T 
  \]  
  (2)

  where \( n \) is the number of sensor nodes in one subregion and \( I \) is the \( n \times n \) identity matrix, and one is a vector of \( n \) ones.

- Apply the double centering to find the matrix \( B \), as follows:
  \[
  B = \bar{H}Q\bar{H} 
  \]  
  (3)

- Determine the eigenvectors and eigenvalues of \( B \):
  \[
  B = A\Lambda A^T 
  \]  
  (4)

  From the above eigen decomposition of matrix \( B \), extract the \( d \) largest positive eigenvalues \( \lambda_1, \ldots, \lambda_d \) of matrix \( B \) and the corresponding \( d \) eigenvectors \( e_1, \ldots, e_d \) of matrix \( B \).

- A \( d \)-dimensional spatial configuration of the \( n \) sensor nodes is derived from the coordinate matrix \( X \) where the coordinates in the lower-dimensional space \( d \) are given by:
  \[
  X = A_dL_d^{1/2} 
  \]  
  (5)

  where \( A_d \) contains the eigenvectors corresponding to the \( d \) largest eigenvalues of matrix \( B \), and \( L_d^{1/2} \) contains the square root of the \( d \) largest eigenvalues of matrix \( B \) along the diagonal.

  Note that the resulting local maps will be relative maps with relative coordinates. Two possibilities exist after this step which will be further elaborated in the next section.

- The first possibility is to stitch all the relative maps together starting with a chosen initial subregion map and leading to a global relative map. The initial subregion selection can be based on choosing the most ‘rigid’ subregion.

- The second possibility is to use the three available anchor nodes (if present) in the initial subregion map and to first convert the initial subregion map into absolute coordinates followed by stitching of the other subregions in a peer-to-peer alignment process. The authors strongly believe that this peer-to-peer stitching/alignment process can also involve a degree of parallelization after the first conversion step of the initial subregion map into absolute coordinates is completed rendering the alignment process a faster convergence to the global absolute map. The selection criteria for the initial subregion is simple here and which is to choose the subregion which has the required three anchor nodes.

### 4.4 Alignment and stitching of local maps using least-squares fitting

This section considers the problem of stitching two relative maps (or one absolute and the other relative) that have at least three nodes in common with each other, but they differ in the locations of those common nodes.

In [11], Meertens et al. note that there are many factors that can contribute to discrepancies related to relative maps:

- Lack of absolute alignment: As explained in the previous section that the relative maps are computed using inter-node distances only, therefore they lack an absolute perception of orientation and the two relative maps may employ disparate coordinate systems due to the application of MDS. Even after this distinction, the two relative maps may be compatible in the sense that they are different only by some combination of translation, rotation and optional reflection. A coordinate transformation from one map to the other map can be calculated, provided that they have at least three common non-collinear nodes. This is exactly what is done in this step of the algorithm and described in this section.

- Disparate basis of information: The two relative maps may have been computed using somewhat different distance information. For example, the inter-node distance measurements may involve redundant estimations to reduce the adverse effects of ranging discrepancies. Furthermore, unreliable communication may wreak havoc as different
estimates may be forwarded in the network with the outcome of providing widely different estimates at different nodes, which may lead to nodes computing widely different averages of estimates. One possible way can be to average the location of the same node in the two given maps as the resulting forwarding error is small and can be averaged out.

- Effects of fading and multipath: Many physical processes that may be used for distance estimation are subject to multipath effects. For example, in the proposed RangeLookup-MDS algorithm, RSSI values are subject to fading and multipath effects. Another example can be the case where the distance between two nodes may be estimated using TDoA of one radio and other acoustic signals. The basic assumption in the ranging process is that the inter-node distance is a straight line, but the reflection can severely affect a signal deflecting it far away from its actual value. The proposed RangeLookup-MDS algorithm tries to solve the multipath problem in two ways. One way is to limit the subregion size to only its one-hop neighbours, and the second is to collect several RSSI measurements and perform an averaging operation.

This section focuses on the map alignment problem only and describes the adopted approach for calculating the requisite isometric transformation (translation + reflection). The adopted method is briefly described below.

The objective in this step is that if we are given two spatial relative maps \( M_1 \) and \( M_2 \), an isometric coordinate transformation can be computed such that the discrepancy between locations in \( M_1 \) and \( M_2 \) is minimized where the discrepancy is given by the Frobenius norm of the differences in locations of nodes that are common to both maps.

\[
\|D\|_F = \sqrt{\left( \sum_{i=1}^{M} \sum_{j=1}^{N} |\delta_{ij}|^2 \right)} \quad (6)
\]

where \( \delta \) represents the differences in locations of nodes in maps \( M_1 \) and \( M_2 \), \( M \) represents the number of common nodes between the two maps (e.g. minimum \( M = 3 \) is needed for 2D transformation), and \( N \) represents the dimensions (e.g. \( N = 2 \) for 2D). The Frobenius norm given above will also be used in the simulations to depict localization error for the whole sensing domain as well as in simulations, the differences between locations of nodes in computed map and coordinates in actual will be used. The map alignment problem is tackled in this step with the use of least-squares fitting algorithm as proposed by Arun and Huang [10]. The algorithm is suitable both for 2D as well as 3D point sets.

The transformation algorithm assumes that the two point sets \( p_i \) and \( p_i' \) where \( i = 1, 2, 3, \ldots, N \) in the two relative maps are related by:

\[
p_i' = Rp_i + T \quad (7)
\]

where \( R \) is a rotation matrix and \( T \) is a translation vector.

The SVD-based algorithm for finding \( R \) and \( T \) is as follows:

- From \( p_i \) and \( p_i' \), calculate \( p, p' \) and then \( q_i \) and \( q_i' \)

\[
p = \frac{1}{N} \sum_{i=1}^{N} p_i \quad (8)
\]

\[
p' = \frac{1}{N} \sum_{i=1}^{N} p_i' \quad (9)
\]

\[
q_i = p_i - p \quad (10)
\]

\[
q_i' = p_i' - p' \quad (11)
\]

- Calculate the \( 3 \times 3 \) matrix

\[
H = \sum_{i=1}^{N} q_i q_i'^T \quad (12)
\]

- Find the SVD of \( H \)

\[
H = U\Lambda V^T \quad (13)
\]

- Calculate \( R \)

\[
R = VU^T \quad (14)
\]

- The translation is found by:

\[
T = p' - Rp \quad (15)
\]

In [37], the SVD-based least-squares fitting algorithm was compared with three other algorithms, namely orthonormal matrices, unit quaternions, and dual quaternions, and it was found to provide the best overall accuracy and stability. Based on this finding, it has been adopted in the proposed RangeLookup-MDS methodology.

As briefly mentioned in the previous section, the above-mentioned algorithm for relative coordinates and absolute coordinates is slightly different.

The steps for stitching of two or more relative maps are as follows:

1. Select an initial relative map. The selection criterion will be to choose the most ‘rigid’ relative map as the initial point. This is discussed in Section 7.7.
2. Use the point sets of relative map 1 and relative map 2 to compute rotation \( R \) and translation \( T \) using the aforementioned algorithm with the outcome that the relative map 2 is rotated and translated into relative map 1.
3. After the rotation and translation are determined, the locations of all the other nodes in the relative map 2 can also
be adjusted using the computed $R$ and $T$ values as shown in Equation (7).

4. Update the locations of all sensor nodes in relative map 2.
5. Repeat the above steps for any other relative maps that have common nodes with either relative map 1 or relative map 2.

The steps for stitching of one absolute map and another relative map are slightly different from above and are given as follows:

1. Select an initial relative map. The selection criterion will be to choose the relative map with the most anchor nodes as the initial point. The minimum requirement for 2D is three non-collinear anchor nodes.
2. Use the coordinates of anchor nodes and relative coordinates of the same nodes in the relative map 1 to compute rotation $R$ and translation $T$ using the aforementioned algorithm with the outcome that the relative map 1 is rotated and translated into the coordinate system of anchor nodes.
3. After the rotation and translation are determined, the locations of all the other nodes in the relative map 1 can also be adjusted using the computed $R$ and $T$ values as shown in Equation (7).
4. Update the locations of all sensor nodes in relative map 1 which will now be promoted to the status of absolute map 1.
5. Use the point sets of absolute map 1 and relative map 2 to compute rotation $R$ and translation $T$ using the aforementioned algorithm with the outcome that the relative map 2 is rotated and translated into absolute map 1.
6. After the rotation and translation are determined, the locations of all the other nodes in the relative map 2 can also be adjusted using the computed $R$ and $T$ values as shown in Equation (7).
7. Update the locations of all sensor nodes in relative map 2 which will now be promoted to the status of absolute map 2.
8. Repeat the above steps for any other relative maps that have common nodes with either absolute map 1 or newly promoted absolute map 2.

There is one interesting heuristic that has been used for the above map stitching procedure to make this process easier to manage, which is presented in Section 7.5.

5 | PERFORMANCE ANALYSIS—ABSOLUTE COORDINATES

To study and evaluate the performance of proposed RangeLookup-MDS method, a number of experiments have been performed. Two different types of scenarios have been considered for the application of the proposed algorithm:

- The first RSSI range lookup scenario is an indoor scenario in which a data set related to MicaZ motes is used.
- The second RSSI range lookup scenario is an outdoor scenario in which a data set related to DSRC nodes is used while considering the movement of cars.

Both of the scenarios are also considered with two different networks. For indoor scenario, following are the details of the chosen rectangular fields:

- The first type of network is a uniform network without any physical obstacles which represents an isotropic placement of nodes in a rectangular shaped area of $30 \times 6 \text{ m}^2$.
- The second type of network is an irregular network with a C shape which represents the anisotropic placement of nodes in a rectangular shaped area of $15 \times 20 \text{ m}^2$ and includes a physical obstacle of $9 \times 6 \text{ m}^2$.

For outdoor scenario, following are the details of the chosen rectangular fields:

- A uniform network in a rectangular shaped area of $400 \times 80 \text{ m}^2$.
- An irregular network in a rectangular shaped area of $200 \times 300 \text{ m}^2$ with a physical obstacle of $126 \times 84 \text{ m}^2$.

In both scenarios, the shown results are for 32 sensor nodes because too many nodes occluded the pictorial representation of errors (shown as trails) in localization of the network. Since most earlier works on MDS-based localization like [7] presented results for fixed-size sensing fields, therefore to maintain consistency with the earlier works, fixed-size rectangular fields were chosen, but the number of sensor nodes was varied. The mentioned scenarios for the same size rectangular fields when repeated with 64, 128, and 256 nodes in the sensor network provided similar trends in results like those presented for 32 nodes. The obtained localization results are presented in Sections 5.1 and 5.2, respectively. All the simulations for indoor as well as outdoor scenarios were carried out using MATLAB version R2020a, and the simulations were executed on Dell Optiplex Core i7 machine.

For both the aforementioned scenarios, the localization errors obtained for RangeLookup-MDS have also been compared with RangeQ-MDS method. Later in this section, the RangeLookup-MDS method is also compared with connectivity-based and distance-based implementation of MDS-MAP(C) method and a centralized implementation of IMDS algorithm. In comparison of the proposed RangeLookup-MDS with distance-based MDS-MAP(C) and with centralized IMDS, the ranging error is modelled as actual range added with Gaussian noise. If it is considered that the range is represented by actual distance blurred with Gaussian noise and if it is supposed that the actual distance is $s^*$ and the range error is $e$, then the measured inter-node distance used in simulation scenario will be a random value obtained from a normal distribution $s^*(1 + \mathcal{N}(0, e))$ and it will simply be represented by $s$.

For the case of absolute coordinates, the performance metric used for comparing both RangeLookup-MDS with
RangeQ-MDS, and other MDS-based methods is the Frobenius norm of the matrix of differences between locations of nodes in computed map and coordinates of nodes in the actual map. In the figures and simulations, it is simply referred to as localization error. It is given in Equation (16).

\[ \|D\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |\delta_{ij}|^2} \]  

(16)

where \( \delta \) represents the differences in locations of nodes in actual map and computed map, \( n \) represents the number of sensor nodes in the whole sensor network, and \( m \) represents the dimensions (e.g., \( m = 2 \) for 2D and \( m = 3 \) for 3D).

In the next two subsections, the localization results for RangeLookup-MDS for indoor and outdoor scenarios are presented. A comparison of the proposed RangeLookup-MDS method with RangeQ-MDS [21] method with eight-level quantization is also presented for both scenarios.

5.1 Indoor scenario

For the indoor simulation scenario, 20 different uniform and irregular networks were generated, and all the four steps of RangeLookup-MDS algorithm were applied to them. Since the objective from the outset was computation of 2D locations, therefore, only three non-collinear anchor nodes were used. A small modification in the first step of the subregion formation algorithm was introduced, which allowed all the anchor nodes to be made part of the initial subregion. It should be noted that for computation of relative coordinates, this modification is not even necessary. All the results presented in this subsection use lookup table shown previously in Table 1. The table data has been derived from [32] and has the characteristics typical of IEEE 802.15.4 MicaZ sensor nodes by Crossbow. The localization in this scenario is sought for the fixed indoor sensor nodes. A modification of the proposed localization algorithm can be carried out for the mobile nodes as a future extension. Following are the characteristics of data used in this simulation.

- Crossbow’s MicaZ motes were used as the sender and receiver nodes.
- The receiver node was connected to the base station (BS) using MIB510 serial gateway.
- For BS, a Dell notebook computer equipped with Intel Core 2 Duo processor was used.
- Each MicaZ sensor node was programed using NesC in TinyOS 1 platform.
- For BS, a Java application collected the packets obtained through the serial port, extracted necessary information, and saved the packets into a database. The database stored the sender ID, counter, and RSSI values for range estimation.
- The signal strengths for fourteen different distances were measured, that is, from 7.5 to 40 ft with gaps of 2.5 ft.
- All the nodes were set to transmit at their default power level, and the locations of the sender and the receiver nodes were same in each case.

For a baseline performance for indoor scenario, the first step in the proposed algorithm was replaced by true Euclidean distances blurred with 2% range errors. The 32 nodes indoor uniform network provided a localization error of 1.2496 and the 32 nodes indoor irregular network provided a localization error of 2.9924. The baseline provided a comparison alternative for simulated networks.

5.1.1 Indoor uniform network

Figure 5 shows localization results after applying RangeLookup-MDS on uniform network of 32 nodes. The markers shown in green colour are the actual locations while the markers shown in blue are the computed locations. The trails shown in red for every sensor node graphically depict the differences between actual locations and computed locations. The anchor nodes are shown with red pentagrams. For this network, the RangeLookup-MDS provides a localization error of 1.4141. Figure 6 shows the application of RangeQ-MDS with eight-level quantization on uniform network of 32 nodes. The eight-level quantization allows the RangeQ algorithm to divide the radio range into eight quantization levels. For this network, RangeQ-MDS provides a localization error of 2.1840. Although RangeQ-MDS allows to use any stitching algorithm like MDS-MAP(P), but to make a fair comparison, same stitching approach has been used in both algorithms, which are least-squares fitting as explained in Section 4.4. Table 3 shows the comparison of RangeLookup-MDS and RangeQ-MDS on different-sized indoor uniform networks. It is clear from the comparison that on almost all the chosen sizes, RangeLookup-MDS provides up to a 35% reduction in localization error as compared to RangeQ-MDS.

5.1.2 Indoor irregular network

Figure 7 shows localization results after applying RangeLookup-MDS on a C-shaped irregular network. Normally, the C-shaped network is seen as more difficult for centralized localization algorithms because it represents a physical obstruction that occludes one side of the network from directly communicating with the other side. Since RangeLookup-MDS is a distributed localization algorithm, it performs equally well for the C-shaped network as well. For this network, the RangeLookup-MDS provides a localization error of 3.8735. Figure 8 shows the application of RangeQ-MDS with eight-level quantization on the same irregular network and the RangeQ algorithm provides a localization error of 5.2345. Table 3 shows the comparison of RangeLookup-MDS and RangeQ-MDS on different-sized indoor irregular networks. Unlike the uniform case, on almost all
the chosen sizes, RangeLookup-MDS provides up to a 25% reduction in localization error as compared to RangeQ-MDS.

5.2 | Outdoor scenario

For the outdoor simulation scenario, 10 different uniform and irregular networks were generated, and all the four steps of RangeLookup-MDS algorithm were applied to them. All the results presented in this subsection use lookup table shown previously in Table 2. The table data has been derived from [38] and has the characteristics typical of DSRC roadside units which communicate with on-board units of passing cars as well as with other roadside fixed infrastructure. The localization in this scenario is sought for the fixed roadside sensor nodes. A modification of the proposed localization algorithm can be carried out for the mobile nodes as a future extension. Following are the characteristics of data used in this simulation.
The DSRC nodes were placed in the middle of a two-way road in an urban environment.

Transmitter antenna heights were 1.5 m.

The measurements were made for distances between 20 and 150 m.

Typical DSRC frequency of 5.8 GHz was used.

Two sets of measurements were made, with and without cars. The data used in the reported simulations is for a fitting curve with the presence of cars as the urban area in consideration was a busy urban area.

The roadside units were set to transmit at 14 dBm power level.

For a baseline performance for outdoor scenario, the 32 nodes outdoor uniform network provided a localization error of 17.4943, and the 32 nodes outdoor irregular network provided a localization error of 41.8936.

5.2.1 Outdoor uniform network

Table 3 shows localization results after applying RangeLookup-MDS on outdoor uniform network of 32 nodes. For this network, the RangeLookup-MDS provides a localization error of 19.7980. Table 3 shows the application of RangeQ-MDS with eight-level quantization on outdoor uniform network of 32 nodes. For this network, RangeQ-MDS provides a localization error of 30.5760. Table 3 shows the comparison of RangeLookup-MDS and RangeQ-MDS on different-sized outdoor uniform networks. On almost all the chosen network sizes, RangeLookup-MDS provides up to a 35% reduction in localization error as compared to RangeQ-MDS.

| Network size | Localization error (RangeLookup-MDS) | Localization error (RangeQ-MDS) |
|--------------|--------------------------------------|----------------------------------|
| Indoor uniform |
| 32 nodes | 1.4141 | 2.1840 |
| 64 nodes | 1.9998 | 3.0886 |
| 128 nodes | 2.8281 | 4.3679 |
| 256 nodes | 3.9995 | 6.1771 |
| Indoor irregular |
| 32 nodes | 3.8733 | 5.2345 |
| 64 nodes | 5.4776 | 7.4027 |
| 128 nodes | 7.464 | 10.4689 |
| 256 nodes | 10.9551 | 14.8053 |
| Outdoor uniform |
| 32 nodes | 19.7980 | 30.5760 |
| 64 nodes | 27.9986 | 43.2409 |
| 128 nodes | 39.5959 | 61.1518 |
| 256 nodes | 55.9971 | 86.4817 |
| Outdoor irregular |
| 32 nodes | 54.2255 | 73.2830 |
| 64 nodes | 76.6864 | 110.7088 |
| 128 nodes | 108.4509 | 156.5658 |
| 256 nodes | 153.3727 | 221.4174 |

Table 3 shows localization results after applying RangeLookup-MDS on outdoor uniform network of 32 nodes. For this network, the RangeLookup-MDS provides a localization error of 19.7980. Table 3 shows the application of RangeQ-MDS with eight-level quantization on outdoor uniform network of 32 nodes. For this network, RangeQ-MDS provides a localization error of 30.5760. Table 3 shows the comparison of RangeLookup-MDS and RangeQ-MDS on different-sized outdoor uniform networks. On almost all the chosen network sizes, RangeLookup-MDS provides up to a 35% reduction in localization error as compared to RangeQ-MDS.
A question arises that the errors shown for the outdoor scenarios are very large and seem to be unitless. In fact, all the presented localization errors have the units of metres and they are presented without any type of normalization. A smaller localization error value specifies that the location algorithm is better in accuracy than the one it is compared with. The rationale for choosing the most straightforward performance metric is that it will greatly ease the process of reproducing the results by showing the actual non-normalized error values.

Another way in which the localization errors can be presented is to depict it as a percentage of the chosen communication radius $R$. For example, in the outdoor case, since $R = 70$ m, the localization error of 19.7980 m of the RangeLookup-MDS will be $28\% R$ and the localization error of 30.5760 m of RangeQ-MDS will be $43.68\% R$.

5.2.2 | Outdoor irregular network

Table 3 shows localization results after applying RangeLookup-MDS on an outdoor C-shaped irregular network. For this network, the RangeLookup-MDS provides a localization error of 54.2255. Table 3 shows the application of RangeQ-MDS with eight-level quantization on the same outdoor irregular network of 32 nodes, and the RangeQ-MDS algorithm provides a localization error of 73.2830. Table 3 shows the comparison of RangeLookup-MDS and RangeQ-MDS on different-sized outdoor irregular networks. Unlike the uniform case, on almost all the chosen sizes, RangeLookup-MDS provides up to a 25% reduction in localization error as compared to RangeQ-MDS.

5.3 | Comparison of RangeLookup-MDS with MDS-MAP(C) for different range errors

In this section, RangeLookup-MDS is compared with connectivity-only MDS-MAP(C), distance-based MDS-MAP(C), and RangeQ-MDS for different percentages of range errors. The selected range errors are 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40%. Figure 9 shows a comparison of all the three techniques for indoor uniform network of 32 nodes. Both the versions of MDS-MAP(C) were centralized versions where centralized computation was performed at one central node with superior computation and storage capabilities to process a large 32$\times$32 distance matrix. In contrast, a distributed version of the RangeQ-MDS algorithm was used in this comparison. In addition to the two versions of the MDS-MAP(C) algorithm, Range-lookup-MDS is also compared with a more recent heuristic-based localization algorithm called IMDS (Improved-MDS). A centralized version of the original IMDS algorithm was implemented in which the use of all-pairs shortest path algorithm in the first step of MDS-MAP(C) was substituted with a geometric heuristic. The comparison as shown in Figure 9 confirms that the IMDS algorithm although performs better than both the distance-based and connectivity-based MDS-MAP(C) algorithms in terms of localization error, but does not perform better than RangeQ-MDS. RangeQ-MDS, in turn, performs worse than RangeLookup-MDS. Implementation of a distributed version of IMDS algorithm and its comparison with RangeQ-MDS and RangeLookup-MDS is a promising direction for future work.
6 | PERFORMANCE ANALYSIS—RELATIVE COORDINATES

One of the most appealing features of MDS-based localization methods is that they are able to recover locations when only inter-node distances are given. The computed locations are up to a translation, rotation, and sometimes reflection also. In this section, results of relative localization are presented for RangeLookup-MDS.

For the case of relative coordinates, since the computed locations are up to an isometric transformation, therefore, the difference between actual and computed locations cannot be used as a performance metric. Rather, the performance metric used for comparison is the Frobenius norm of the differences of pair-wise distances between nodes in actual map and relative map. In the figures and simulations, it is simply referred to as localization error, but a more suitable name will be relative localization error. It is given in Equation (16).

\[ \|D\|_F = \sqrt{\sum_{i=1}^{n} \|\delta_i\|^2} \]

where \( \delta \) represents the differences between pair-wise inter-node distances, \( n \) represents the number of sensor nodes in the whole sensor network, and \( n \frac{n-1}{2} \) represents the total number of pair-wise distances between nodes.

In the next two subsections, the relative localization results for RangeLookup-MDS for indoor and outdoor scenarios are presented.

6.1 | Indoor scenario

For the indoor simulation scenario, 20 different uniform and irregular networks were generated, and all the four steps of RangeLookup-MDS algorithm were applied to them. The first subregion was selected as a reference point from where the stitching process was initiated. There is a possibility that if the first subregion's distance matrix is not rigid enough, its own relative map will be reflected and when it serves as the reference subregion to other subregions stitches, the resulting global relative map will also have reflection. In Section 7.7, a simple method will be discussed to choose a sufficiently 'rigid' starting subregion with the goal to avoid reflection problem in the reference subregion so that global relative map does not have reflection problem either.

For a baseline performance for the indoor scenario, the 32 nodes indoor uniform network provided a relative localization error of 4.9857, and the 32 nodes indoor irregular network provided a relative localization error of 10.2070.

6.1.1 | Indoor uniform network

Figure 10 shows localization results after applying RangeLookup-MDS on uniform network of 32 nodes. For this network, the RangeLookup-MDS provides a relative localization error of 7.3435. For the same network, RangeQ-MDS provides a localization error of 10.8936.

6.1.2 | Indoor irregular network

Figure 11 shows localization results after applying RangeLookup-MDS on irregular network of 32 nodes. For this
network, the RangeLookup-MDS provides a localization error of 13.8339. For the same network, RangeQ-MDS provides a localization error of 18.2025.

6.2 Outdoor scenario

Similar to the absolute case, the outdoor scenario was simulated with 10 uniform and irregular networks of 32 nodes each. On all the networks, RangeLookup-MDS was applied with a fixed radio range of $R$, and localization errors were analyzed and then compared with the RangeQ-MDS method. The simulation was then repeated for bigger network sizes of 64, 128, and 256 nodes.

For a baseline performance for the outdoor scenario, the 32 nodes outdoor uniform network provided a relative localization error of 69.8004, and the 32 nodes outdoor irregular network provided a relative localization error of 142.8980.
6.2.1 | Outdoor uniform network

For the outdoor uniform network of 32 nodes, RangeLookup-MDS provides a localization error of 108.2837. In comparison, RangeQ-MDS provides a localization error of 142.4785.

6.2.2 | Outdoor irregular network

For the outdoor irregular C-shaped network, RangeLookup-MDS provides a localization error of 193.6753. In comparison, RangeQ-MDS provides a localization error of 251.5263.

These results confirm the assumption that the proposed algorithm can be suitably deployed in indoor as well as outdoor networks with uniform or irregular distribution of nodes. In the following sections, few recommendations are being discussed for some additional issues like RangeLookup-MDS or any localization algorithm can encounter.

7 | DISCUSSION

In this section, a number of important effects have been studied and some issues have been identified which, if not considered, may adversely affect the outcome of the localization process or the accuracy of computed locations. To deal with those issues, some suggestions and enhancements have been recommended.

7.1 | Effect of increasing number of anchor nodes on localization error

Most of the simulation results presented in Section 5 for absolute coordinates used three anchor nodes in the first subregion, which is the minimum required for aligning and stitching 2D spatial maps using least-squares fitting. The effect of increasing the number of anchor nodes in the first subregion was studied and all the simulations were repeated with four, five, six, seven, and eight anchor nodes. Figure 12 shows localization error for 32 nodes indoor uniform network with eight anchor nodes and Figure 13 presents localization error reduction in going from three to eight anchors. It is very clear that the error reduction is only 5% with an increased number of anchors. This should be seen as a strength of the proposed algorithm as it is able to perform stitching with the minimum number of non-collinear anchors, which is three for the 2D case.

In fact, one of the design considerations from the outset was to design the localization algorithm in such a way that the number of anchors needed is as small as possible. The results in Figure 12 confirm the fact that the proposed RangeLookup-MDS is able to provide suitable accuracy in localization even with only three anchor nodes in the 2D case. Additional simulations with more than eight anchor nodes proved that the increase in number of anchors provided no considerable improvement in the localization error of the given network. Hence, it can be said that the minimum number of anchors needed in a 2D case is three non-collinear anchor nodes in any subregion anywhere in the whole network. The optimum density of anchor nodes in a given network can be derived by simulations using an iterative method as follows:

1. Initialize \( m \) to 3.
2. Increment \( m \) by 1.
3. Place \( m \) anchors in the chosen initial subregion.
4. Perform stitching with \( m \) anchors and save the localization error in \( \text{loc} \).
5. Perform stitching with \( m + 1 \) anchors and save the localization error in \( \text{newloc} \).
   (a) If \( \text{newloc} < \text{loc} \), continue to step 2.
   (b) If \( \text{newloc} \geq \text{loc} \), stop and choose \( m \) as the most optimal value.

A future direction can be to find suitable localization with only two anchor nodes. Equally important in the proposed algorithm is the placement of anchor nodes which is discussed in the next subsection.

7.2 | Effect of placement of anchor nodes on localization error

Since the stitching process in RangeLookup-MDS is a peer-to-peer alignment process and it initiates from the first subregion, therefore, the placement of the anchor nodes has an important role in reducing the error emanating from the alignment process. To study this effect, the three anchors in the first subregion were placed in the corner of the sensing field which resulted in an increase in the localization error of the network, as shown in Figure 14. Additionally, a number of different shaped uniform and irregular networks like O-shaped, grid-shaped, and cross-shaped network were tried with different placement of anchors in a fixed-sized square-shaped sensing field of \( 20 \times 20 \) m\(^2\). It was discovered that the RangeLookup-MDS algorithm provides the least localization error on a cross-shaped network with three anchors in the centre of the cross (or sensing field), as shown in Figure 15. This observation bodes well to apply this proposed algorithm for outdoor scenarios like traffic intersection, traffic roundabout, or similar indoor scenarios.

7.3 | Effect of quantization levels on performance

For a fair comparison of proposed RangeLookup-MDS with the previous RangeQ-MDS algorithm, an eight-level quantization of the radio range \( R \) was adopted in the simulations as there were almost eight entries in the lookup table for the chosen radio range. An important aspect related to the number
of chosen quantization levels/number of given entries in the lookup table was studied and the same simulations were repeated for 6, 8, and 10 quantization levels. It was found that for the chosen radio range, both algorithms were unable to adequately construct 2D coordinates when the quantization level was less than six, for example, four-level quantization. In addition, beyond eight levels of quantization, there was little improvement observed in RangeQ-MDS. This observation is shown in Figure 16.

7.4 Effects of wireless propagation on RSSI values in outdoor scenario

Since RSSI values were used to construct a distance matrix and they can get affected by wireless propagation effects, therefore, the effects of wireless propagation on the RSSI values were studied.

To study the effects of wireless propagation in the outdoor scenario, a statistical wireless signal propagation model based on
was used. The model was used to obtain descriptive statistics for outdoor RSSI data based on the transmitter and receiver separation within the sensor network. Three salient phenomena were modelled:

1. Distance dependent path loss or attenuation in average received power;
2. Shadowing caused by obstructing structures like buildings, trees, and hills; and
3. Fine structures or small scale fading occurring due to multipath propagation whose statistics are captured by Nakagami-$m$ distribution.

Each of these effects was separately calculated and added together to produce a realization of received power in dBm on distance range of 10–150 m, at intervals of 10 m.

7.4.1 Path loss model

The following relationship was used to model average attenuation in signal power across large distances [31],

$$ P_r = P_t K \left( \frac{d_0}{d} \right)^\gamma $$

(17)
where $P_r$ is the received power, $P_t$ is the transmitted power, $d_0$ is the reference distance set to 10 m, $\gamma$ is the path loss exponent for outdoor propagation set to 3.7, $d$ is the $T_x$-$R_x$ separation in m, and $K$ is a unit-less constant that depends on the antenna characteristics.

The following relationship is used to calculate $K$:

$$K = \left( \frac{\lambda}{4\pi d_0} \right)^2$$

where $\lambda$ is the wavelength corresponding to 5.8 GHz.

**7.4.2 | Shadowing combined with path loss**

The net effect of shadowing caused by obstruction was modelled as a zero mean log-Normal random variable, $X_{\sigma_d}$ having a variance $\sigma_d^2 = 13.29$. This value is calculated from empirical data in [39] for outdoor propagation. The total average received power is thus equal to:

$$P_r \text{ dBm} = P_t \text{ dBm} + K \text{ dB} - 10\gamma \log_{10} \left( \frac{d}{d_0} \right) + X_{\sigma_d}$$

**7.4.3 | Narrow-band multipath fading**

A simple statistical model was assumed for the narrow-band, flat fading process that represents the fine structure in the received signal. The envelope of the faded signal is given by Nakagami-$m$ distribution with parameter $m = 0.5$. The probability density for the signal envelope $z$ is given as:

$$p_z(z) = \frac{2m^mz^{2m-1}}{\Gamma(m)} \left( \frac{P_r}{P_r} \right)^m \exp \left( -\frac{mz^2}{P_r} \right), \quad z \geq 0$$

The final statistics were obtained by using Monte-Carlo simulation through realization of 1000 received power versus distance curves and doing a box whisker plot of the aggregate data as shown in Figure 17.

Another problem related to multipath fading is frequency-selective fading. Since DSRC uses Orthogonal Frequency-Division Multiplexing (OFDM) scheme which shows robustness to frequency selective fading as long as the delays within guard intervals are introduced in multipath channels, the effect of frequency-selective fading has not been studied here. A very interesting future direction will be to study the inter-symbol interference problem caused when the delays beyond the guard intervals are introduced in the multipath channels.

The updated RSSI values and distance estimates when used for the outdoor localization provided similar trends in localization accuracy as obtained for the normal unaffected RSSI values. It is the authors’ belief that it is the first time such an impact of wireless propagation on distance estimation is being presented for any MDS-based localization algorithm.

**7.5 | A simple process for initializing the map alignment/stitching process**

A question arises that when two subregions start their alignment process using common sensor nodes, how should they initialize and proceed. In [11], Meertens et al. describe a process for obtaining a shared coordinate system.

In the proposed RangeLookup-MDS algorithm, alignment can proceed as follows:
On initialization, each head node determines for itself a random priority with which it tags its own map when it is broadcast. Ideally, each head node would receive a unique priority and priorities would be randomly distributed throughout the network.

When a head node receives a map, it processes the map differently according to whether its own priority is higher or lower than the received map’s priority:

- If the receiving head node’s priority is lower than the received map’s, then the receiving head node aligns its own local map with the received absolute/stitched map and after the alignment is completed, it increases its own priority to make it equal to the received map.
- If the receiving head node’s priority is higher than the received map’s priority, then the head node aligns the received map with its own map.
- If the receiving head node’s priority is same as that of the received map’s priority, then the head node averages the coordinates of all common nodes with the received map’s common nodes in order to make the coordinates closer to the actual values.

In this way, the coordinate system of the head node with the highest priority acts as a fixed reference system to which the other head nodes align themselves. In the proposed algorithm, the subregion with three anchor nodes will assume the highest priority. In the case of relative coordinates, the subregion with the most rigid distance matrix will assume the highest priority so that the flip ambiguity can be avoided.

### 7.6 Case of orphan nodes

Although in the simulations, there were no unlocalized nodes because of controlled radio range, but in real sensor networks, there will be orphan nodes that due to their remoteness fail to get covered by any subregion when step 1 of the algorithm which creates subregions is completed. Following is a very simple way to deal with the case of orphan nodes.

- If the orphan node comes in contact with only one localized node, it can query the newly found node and temporarily assume the coordinates of that node.
- If the orphan node comes in contact with two localized nodes, it can query the newly found nodes for their coordinates and temporarily declare itself to be at the midpoint of those nodes by averaging their locations and assuming the averaged coordinates of its own.
- If the orphan node comes in contact with three localized nodes, it can query the newly found nodes for their coordinates, ensure that those three points are non-collinear and declare itself to be the head node a new subregion and execute only, steps 2, 3 and 4 of RangeLookup-MDS and calculate coordinates of its own.

### 7.7 Case of flip ambiguity

In the computation of the relative coordinates, it was observed that if the distance matrix of the starting subregion is not ‘rigid’ enough, it can create a local map that will be reflected across y-axis. Additionally, when all the other subregions use that starting subregion as a reference to stitch to, the resulting global relative map will also be reflected. To avoid this problem in relative coordinates, a simple heuristic can be used. Instead of randomly picking any subregion as a starting point, the picked subregion can be checked for rigidity by using the Pebble Game algorithm [40] and can be given a higher priority only if it is found rigid. Otherwise, another subregion can be picked.
and checked. This heuristic will result in having a global relative map without any reflection problem.

7.8 | Case of change in operating voltage

The localization algorithm is based on the assumption that the operating voltage of all the batteries in all of the sensor nodes is same. But, if sensor nodes have different battery levels or are using power optimization schemes, they may have different voltage levels, and assuming that the major portion of power is consumed in communication instead of computation, the transmit power will be non-uniform across the network leading to incorrect range measurements.

One possible solution to this problem is simply to encode the \( T_c \) power level for each sensor node and use a modulated feed-forward signal to transmit that power level value which is received by the neighbouring receiver sensor node and demodulated at the receiver sensor node. The demodulated value can then be used to re-calibrate the measured RSSI value to the correct one by multiplying the power level with the RSSI value at the receiver node.

Additionally, the power levels can be quantized and instead of sending a number to the receiver node, their indices can be sent to the receiver node which can perform a table lookup to find out the correct power level and use it for re-calibration.

7.9 | Computational complexity

The computational complexity of the RangeLookup-MDS can be given by individual complexities involved in the four steps:

- Each local map in step 3 can be created with a complexity of \( O(v^3) \) where \( v \) is the number of nodes in a subregion. If there are \( c \) subregions in the field, then the computation of \( c \) maps will take \( O(cv^3) \). But all the local maps can be computed in parallel, so this step will take a fixed amount of time.
- Step 4 will take the longest time if the alignment process is serial. If there are \( r \) shared nodes between two peer subregions, then each subregion's alignment/stitching process will require \( O(r^3 + (v - r)) \) where \( v \) is the number of nodes in a subregion. The total time required will be \( c \cdot O(r^3 + (v - r)) \) but this alignment process can be easily parallelized if the starting subregion is not in the corner of the sensing field but in the middle.

Typical time in seconds taken for the chosen networks in simulation for the absolute case is shown in Table 4. The time taken in seconds for the relative coordinates is slightly less than the absolute case because of some time saving in the alignment step. Figure 18 shows an example of execution time taken by centralized distance-based MDS-MAP(C), distributed version of RangeQ-MDS, and RangeLookup-MDS. In terms of execution time, RangeLookup-MDS performs better than MDS-MAP(C) and slightly better than RangeQ-MDS also. This is because the centralized version handles very large dissimilarity matrices as compared to distributed versions where a comparatively smaller-sized subregion ensures faster computation. Also, in distributed versions, all subregions are able to compute their local maps in parallel, and the longest component of time taken is in the serial peer-to-peer alignment process in worst-case scenario. In case if some steps of the alignment process in distributed versions are parallelized, the time taken by distributed versions will become significantly less.

7.10 | Refinements

A number of refinements related to the different limitations of the RangeLookup-MDS algorithm are possible.

- The first major refinement can be to utilize DV-Hop algorithm [16] for providing three non-collinear nodes in the first subregion which has to serve as the starting point for stitching phase of absolute coordinates. While computing absolute coordinates in Sections 5.1 and 5.2, it was described that three non-collinear nodes are part of the first subregion from where the stitching step can begin. If one or all of these three nodes are not present in the subregion or if all

| Network size | MDS-based local maps (s) | Alignment/stitching (s) |
|--------------|--------------------------|-------------------------|
| 32 nodes     | 0.0774                   | 0.562                   |
| 64 nodes     | 0.1161                   | 0.843                   |
| 128 nodes    | 0.1742                   | 1.2645                  |
| 256 nodes    | 0.2612                   | 1.8968                  |

**FIGURE 18** Comparison of execution time taken by MDS-MAP(C), RangeQ-MDS, and RangeLookup-MDS
those three anchor nodes are scattered at three different locations away from the first subregion, DV-Hop or any gradient propagation method can be used so that the alignment starting subregion can first compute absolute coordinate for three of its nodes before starting the peer-to-peer alignment step.

- As the alignment process proceeds further from the initial subregion, the effect of fitting also starts to affect subsequent location computations. To minimize this effect, the alignment process for the same subregion can be repeated more than once to match the rotation and translation more closely to the desired coordinates. Additionally, refinement algorithms like damped least-squares method can also be used, but these methods will be more computationally expensive and are more suitable for centralized methods as compared to distributed ones like RangeLookup-MDS. MATLAB provides `lsqnonlin()` function which can be used to try an application on local maps or on peer-to-peer alignment step or globally after the localization process is completed.

- Other different types of MDS methods like iterative SMA-COF and isotonic regression can be tried in step 3 of the proposed algorithm with the objective of expediting local map computation.

- In a smaller subregion, most of the sensor nodes can communicate with each other using direct line-of-sight. If a larger subregion is formed in such a way that all the nodes in a subregion cannot directly communicate with each other, the missing inter-node distances at the subregion head node can be obtained by using an all-pairs shortest path algorithm like Floyd-Warshall’s or Johnson’s algorithm. This operation will be computationally expensive as the computational complexity of all-pairs algorithm is $O(v^3)$, where $v$ is the number of nodes in a given subregion.

### 7.11 3D RangeLookup-MDS algorithm

One aspect of a sensing field is that it is physically a 3D scenario. Most localization algorithms have focussed mainly on the 2D scenario because of simplicity and computational ease. In outdoor scenarios like traffic monitoring or road networks where there is not much variation in the heights of the installed sensors, 2D coordinates will be sufficient to provide a good picture of location of sensors in the field being monitored. But, in either indoor or outdoor scenarios where there is a significant change or variation in the height or depth parameter of the installed nodes, the computed 2D coordinates will not be able to truly depict the locations of sensors in the sensing field. Rather, the computed 2D coordinates can sometimes be a little confusing.

In this section, a modified version of RangeLookup-MDS localization algorithm is being presented to compute 3D coordinates of sensors in the sensing field. The proposed modification uses the ideas similar to those presented for 2D version in Section 4 with the difference that instead of a minimum of three non-collinear common nodes between subregions, there is a stringent requirement to have four common nodes between overlapping subregions with the condition that the four nodes should be non-coplanar.

#### 7.11.1 Modified step

The 3D version of RangeLookup algorithm has almost the same first three steps as RangeLookup-MDS with the only difference in step 4. In the following, updated step 4 is being presented.

- **Alignment and stitching of local maps using least-squares fitting:** For any given pair of maps that have at least four non-coplanar nodes in common, the locations of those four nodes that appear in both maps can be used to calculate an isometric coordinate transformation (a translation + rotation) using least-squares fitting that approximately aligns the maps. If this alignment is performed using four anchor nodes with known absolute coordinates in 3D in any subregion, the absolute coordinates are obtained, otherwise relative coordinates are obtained.

#### 7.11.2 Performance analysis

To study and evaluate the performance of 3D RangeLookup-MDS method, a number of experiments were performed. The chosen network was a uniform network without any physical obstacles which represents an isotropic placement of nodes in a rectangular-shaped indoor area in which a data set related to MicaZ motes is used. The shown results are for a network of 44 sensor nodes with two subregions of 22 nodes each. The 44 node network has been divided into two subregions of 22 nodes each. The network has four anchor nodes for the absolute case. The radio range $R$ is fixed.

Figure 19 shows application of 3D RangeLookup-MDS on an indoor uniform network of 44 nodes for the absolute case. In contrast, 3D RangeQ-MDS provided a localization error of 4.9223. A comparison of localization errors shows that 3D Range-Lookup-MDS reduces the localization error up to 30% in this uniform indoor network.

Figure 20 shows application of 3D RangeLookup-MDS on an indoor uniform network of 44 nodes for the relative case. In contrast, 3D RangeQ-MDS provided a relative localization error of 27.7453. A comparison of localization errors shows that the 3D RangeLookup-MDS reduces the relative localization error up to 33% in this indoor uniform network.

#### 7.11.3 Limitations

Similar to the 2D case where three non-collinear nodes were needed for stitching step, the limitation of the 3D RangeLookup-MDS also arises from the stitching step which
requires that the four common nodes between any two sub-regions which need to stitch themselves should be non-coplanar, otherwise if all the common nodes are coplanar, it will end up in a degenerate case and a required unique rotation will not be obtained as required by the stitching step.

Another limitation of the proposed algorithm is that for most distance matrices, 2D coordinates will almost always be recovered using the two largest eigenvalues, but the 3D coordinates will be obtained only for those point sets in which three eigenvalues are available.

8 | EXPERIMENTAL VALIDATION

To experimentally validate the proposed algorithm, a nine node single subregion indoor uniform network was created with eight Crossbow MicaZ motes and one Dell computer acting as the head node where all computations were performed. All nine nodes were used in an indoor environment and the indoor scenario specifications mentioned in Section 5.1 were completely adopted. Two experiments were carried out using the indoor scenario.
The first experiment involved verification of the lookup table shown in Table 1 after placing the eight motes at eight different distances from 5 to 22.5 ft with gaps of 2.5 ft. This was done to verify the part of the lookup table which was mainly used in the indoor scenario.

The second experiment involved placing the nodes in a rectangular area which was marked with 2D absolute coordinates. Three of the motes were given the status of anchor nodes and the other five were member nodes whose absolute coordinates had to be computed by the subregion head node PC. The head node PC was not assigned any 2D coordinates. The same Java application which collected packets from the motes for the purpose of RSSI to distance conversion was modified to calculate their 2D coordinates also. Both RangeLookup-MDS and RangeQ-MDS computations were performed.

For the proposed RangeLookup-MDS, the nine nodes indoor uniform network provided a localization error of 0.1064. In contrast, the RangeQ-MDS provided a localization error of 0.1939. This denotes a 45% improvement in the localization error which is higher than expected. It is understood that a larger-sized network with 32 or more nodes will provide improvements in the localization error from 25% to 35% as compared to RangeQ-MDS algorithm. In addition, the authors suggest that the use of Cohda Wireless DSCR nodes for experimental validation of outdoor scenario which can be an interesting future direction to follow.

9 CONCLUSION

Here, a partial range-based distributed localization algorithm has been proposed for the fixed sensors nodes in a 2D sensing domain. The proposed algorithm organizes sensors in small subregions and then creates inter-node distance matrices utilizing range lookup for RSSI to distance conversion. The algorithm employs MDS technique from the data visualization field to construct local relative maps for subregions using inter-node distance matrices. These small local maps are stitched together using the least-squares fitting technique with the outcome that the stitched map approximates the absolute global map. Simulation results on indoor and outdoor uniform as well as irregular networks of various sizes show that the proposed algorithm provides improved localization accuracy and reduces localization error up to 25% in comparison to a previous partial range-based localization algorithm.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support provided by the Systems & Control Group at the Department of Electrical, Computer, & Software Engineering of the University of Auckland.

CONFLICT OF INTERESTS

The authors declare that they have no conflict of interests.

ORCID

Farhan Khan https://orcid.org/0000-0002-9805-9187

REFERENCES

1. Carminati, M., et al.: A self-powered wireless water quality sensing network enabling smart monitoring of biological and chemical stability in supply systems. Sensors, 20(4), 1125 (2020)
2. Akyildiz, I., Vuran, M. C.: Wireless Sensor Networks, 1st ed. John Wiley & Sons, Inc., New York (2010)
3. Boukerche, A.: Algorithms and Protocols for Wireless Sensor Networks, 1st ed. John Wiley & Sons, Inc., Hoboken, NJ (2009)
4. Li, X., et al.: A sorted RSSI quantization based algorithm for sensor network localization. In: Proceedings of the 11th international conference on parallel and distributed systems, 557–563, Fukoaku(2005)
5. Chan, H., Perrig, A.: ACE: an emergent algorithm for highly uniform cluster formation. In: Proceedings of the 1st European workshop on wireless sensor networks, pp. 154–171, Berlin (2004)
6. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. 3(4), 366–379 (2004)
7. Shang, Y., et al.: Localization from connectivity in sensor networks. IEEE Trans. Parallel Distrib. Syst. 15(11), 961–974 (2004)
8. Doberty, L., et al.: Convex position estimation in wireless sensor networks. IEEE, Anchorage, AK (2001) Proceedings of IEEE INFOCOM 2001 Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (2001)
9. Parwari, N., Hero, A.: Using proximity and quantized RSS for sensor localization in wireless networks. In: Proceedings of the 2nd IEEE/ACM international conference on wireless sensor networks and applications, 892–898, Columbus (2005)
10. Arun, K., et al.: Least-squares fitting of two 3-d point sets. IEEE Trans. Pattern Anal. Mach. Intell. 9(5), 698–700 (1987)
11. Meerum, L., Fitzpatrick, S.: The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances, pp. 1–1, Kestrel Institute Technical Report, Palo Alto, CA (2004).
12. Bachrach, J., Taylor, C.: Localization in sensor networks. In: Stoimenovic, I. (ed.) Handbook of Sensor Networks: Algorithms and Architectures, 1st ed., pp. 277–310, John Wiley & Sons, Inc., Hoboken, NJ (2005)
13. Dagher, R., Quilez, R.: Localization in wireless sensor networks. In: Mitton, N., Simplot-Ryl, D. (eds.) Wireless Sensor and Robot Networks From Topology Control To Communication Aspects, 1st ed. pp. 203–247, World Scientific (2014)
14. Bulusu N., et al.: Scalable, ad hoc deployable RF-based localization. In: Proceedings of Grace Hopper Celebration of Women in Computing Conference, pp. 1–5, Vancouver (2002)
15. He, T., et al.: Range-free localization schemes for large scale sensor networks. In: Proceedings of the 9th annual international conference on mobile computing and networking (MobiCom), pp. 81–95, 2003, New York, NY
16. Niculescu, D., Nath, B.: Ad hoc positioning system (APS). In: Proceedings of the IEEE global telecommunications conference (GLOBECOM), pp. 2926–2931, San Antonio (2001)
17. Nagpal, R., et al.: Organising a global coordinate system from local information on an ad hoc sensor network, In: Proceedings of the 2nd international conference on Information processing in sensor networks, pp. 333–348, Palo Alto (2003)
18. Priyantha, N., et al.: Anchor-free distributed localization in sensor networks. In: Proceedings of the 1st international conference on embedded networked sensor systems, 340–341, Los Angeles, CA (2003).
19. Moore, D., et al.: Robust distributed network localization with noisy range measurements. In: Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, pp. 50–61 (2004)
20. Ji, X., Zha, H.: Sensor positioning in wireless ad hoc networks using multidimensional scaling. In: Proceedings of the 23rd annual joint conference of the IEEE computer and communications societies, Hong Kong, China, pp. 2652–2661 (2004)
21. Li, X., et al.: A partial-range-aware localization algorithm for ad-hoc wireless sensor network. In: Proceedings of the 29th annual IEEE international conference on local computer networks, pp. 77–83. Tampa (2004)

22. Saeed, N., et al.: A state-of-the-art survey on multidimensional scaling-based localization techniques. IEEE Commun. Surv. Tutor. 21(4), 3565–3583 (2019)

23. Ahmad, T., et al.: Parametric loop division for 3d localization in wireless sensor networks. Sensors. 17(7), 1697 (2017)

24. Ahmad, T., et al.: Social network analysis based localization technique with clustered closeness centrality for 3D wireless sensor networks. Electronics. 9(5), 738 (2020)

25. Hamouda, E., Abohamama, A-S.: Wireless sensor nodes localiser based on sine–cosine algorithm. IET Wirel. Sens. Syst. 10(4), 145–153 (2020)

26. Alshamaa, D., et al.: An evidential framework for localization of sensors in indoor environments. Sensors. 20(1), 318 (2020)

27. Pagano, S., et al.: Indoor ranging and localisation algorithm based on received signal strength indicator using statistic parameters for wireless sensor networks. IET Wirel. Sens. Syst. 5(5), 243–249 (2015)

28. Mishra, S., et al.: Localization in 3D wireless sensor network. In: Chaki, R. et al. (eds.) Advanced Computing and Systems for Security. Advances in Intelligent Systems and Computing, vol. 667, pp. 121–131. Springer, Singapore (2018)

29. Stojkoska, B.R.: Nodes localization in 3d wireless sensor networks based on multidimensional scaling algorithm. Inter. Sch. Res. Not. 14(1), 1–10 (2014)

30. Liu, C., et al.: Multiregional secure localization using compressive sensing in wireless sensor networks. ETRI J. 41(1), 739–749 (2019)

31. Mohar, S., et al.: A survey of localization in wireless sensor network using optimization techniques. In: Proceedings of the 4th international conference on computing communication and automation (ICCCA), Greater Noida, India, pp. 1–6, (2018)

32. Hussain, S., Rahman, M.: Using received signal strength indicator to detect node replacement and replication attacks in wireless sensor networks. In: Proceedings of SPIE 7344, data mining, intrusion detection, information security and assurance, and data networks security, pp. 150–160, Orlando (2009)

33. Gerla, M., Tsai, J.: Multicluster, mobile, multimedia radio network. Wirel. Netw. 1(3), 255–265 (1995)

34. Torgerson, W.: Multidimensional scaling: I. Theory and method. Psychometrika. 17, 401–419 (1952)

35. Young, G., Householder, A.: Discussion of a set of points in terms of their mutual distances. Psychometrika. 3, 19–22 (1938)

36. Gower, J.: Some distance properties of latent root and vector methods in multivariate analysis. Biometrika. 53, 325–338 (1966)

37. Larusso, A., et al.: A comparison of four algorithms for estimating 3-D rigid transformations. In: Proceedings of the 1995 British conference on machine vision, pp. 24.1–24.10, Birmingham (1995)

38. Al-Hourani, A., et al.: Propagation measurements in 5.8GHz and path-loss study for CEN-DSRC. In: Proceedings of the 2014 international conference on connected vehicles and expo (ICCVE), Vienna, pp. 1086–1091 (2014)

39. Goldsmith, A.: Wireless communications, 1st ed. Cambridge University Press, Cambridge (2005)

40. Jacobs, D. J., Hendrickson, B.: An algorithm for two-dimensional rigidity percolation: the pebble game. J. Comput. Phys. 137, 346–365 (1997)