

Research Article

Efficient Deterministic Anchor Deployment for Sensor Network Positioning

Yongle Chen,^{1,2} Ci Chen,³ Hongsong Zhu,¹ and Limin Sun¹

¹ State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

² University of the Chinese Academy of Sciences, Beijing 100190, China

³ The School of Software and Microelectronics, Peking University, Beijing 102600, China

Correspondence should be addressed to Limin Sun; sunlimin@iie.ac.cn

Received 10 January 2013; Accepted 24 February 2013

Academic Editor: Jianwei Niu

Copyright © 2013 Yongle Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensor network positioning systems have been extensively studied in recent years. Most of the systems share a common assumption that some known-position anchor nodes have existed. However, a more fundamental question is always being overlooked, that is, how to acquire the anchor's position. In general, GPS-based measures and the artificial calibration are two dominant methods to acquire anchor positions. Due to the high energy cost and failures in occlusion regions of the GPS modules, the artificial calibration method is adopted extensively. Nevertheless, numerous disadvantages of the artificial calibration, such as the expensive labor cost and error-prone features, also make it hard to be an efficient solution for the anchor positioning. For this reason, we design an efficient mapping algorithm between anchors and their positions (MD-SKM) to avoid the complicated artificial calibration. Additionally, we propose a best feature matching (BFM) method to further relax the restriction of MDS-KM where three or more calibrated anchors are needed. We evaluate our MDS-KM algorithm under various topologies and connectivity settings. Experiment results show that at a slightly higher connectivity level, our algorithm can achieve the exactly correct matching between anchors and their positions without any calibrated anchors.

1. Introduction

Sensor networks have been extensively studied due to their salient advantages of monitoring and controlling related applications, such as the battlefield monitoring, medical surveillance, and structure monitoring. Furthermore, in these applications, a basic service requirement implied is to determine the exact location of the happening event through pre-deployed sensors, in order that the operators are prone to execute appropriate control actions in response to the event. In this sense, the positioning technology of the sensor network is getting more and more attentions. Currently, most developed positioning systems depend on four metrics, including TOA, AOA, RSS, and the connectivity of the signals. In addition, these systems also have a common assumption; that is, some known-position nodes exist in the sensor network, which are also named as anchors. Based on the known-position anchors, they are able to localize

other unknown-position nodes. However, how to acquire the anchor's position is still an unsolved question.

At present, two methods are mainly used for acquiring anchors' positions. One is leveraging GPS modules, and the other is called the artificial calibration [1]. In practice, the placement ways to construct the sensor network decide which method to be adopted to acquire anchors' positions. The placement ways can be summarized in two types: the stochastic way and the deterministic way. In terms of the stochastic way, the most typical example is the battlefield monitoring. In the battlefield scenario, military sensors are randomly scattered from the air, where the anchors' positions are stochastic. In this case, to acquire anchors' positions, we have to rely on the GPS modules attached with sensors. The GPS modules are usually limited by a series of disadvantages, such as high energy costs and failures in occlusion regions, which make this method not applicable for the low-power sensor networks. Different from the stochastic way, in some

scenarios, the anchors' positions are deterministic according to predesigned placement blueprint, such as that in medical surveillance and structure monitoring. In these scenarios, the correspondence is recorded between anchors' physical positions in the blueprint and anchors' IDs, such that each anchor unambiguously knows its own position. We named this method the artificial calibration. Also, many works [2, 3] have pointed out that optimizing the anchor placement is able to accelerate the convergence of the positioning algorithm and improve the positioning accuracy. Nevertheless, this kind of methods always suffers from the complicated and error-prone mapping between physical locations and the node IDs, which is even more severe in a large sensor network.

In this paper, we design an efficient MDS-KM matching algorithm to avoid the artificial calibration cost in deterministic anchor placement. To the best of our knowledge, we are the first to consider solving the artificial calibration problem of anchors placement. Given sufficient calibrated anchors (C-anchors for short), we first design a distributed MDS-MAP(A) method to construct an absolute radiomap by using estimated distance or hop distance between anchors. In the absolute radiomap, each anchor has an absolute coordinate of its position, corresponding to the physical position in the blueprint. In order to map the radiomap with the blueprint, we use the kNN method to select the k -nearest physical positions in the blueprint away from the anchor absolute positions in the radiomap and then build a complete bipartite graph. Based on the bipartite graph, we adopt the Kuhn-Munkres (KM) algorithm to get a maximum weighted matching. Accordingly, we achieve the correspondence between anchor nodes in the radiomap and physical positions in the blueprint. Meanwhile, in order to relax our MDS-KM method for the cases without calibrated anchors, we design a best feature matching (BFM) method to actively map parts of anchors in the radiomap to positions in the blueprint. Our method will greatly improve the efficiency of anchor placement through avoiding the artificial calibration. The experiment in Section 6 shows that the mapping from the radiomap to the blueprint is exactly correct when the connectivity level of network is not excessively low.

The remainder of the paper is organized as follows. The related work is shown in Section 2. We formulate the problem in Section 3 while leaving the details of our algorithm design for Section 4. The further improved strategy is presented in Section 5. Then we show the experiment and simulation results of our MDS-KM algorithm in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

Many works have pointed out that anchor placement ways will help to improve positioning performance. The pioneer anchor placement ways are mainly based on the empirical evidence in positioning system. For example, Shang et al. [2] randomly place anchors in their experiment and find that a selection of collinear anchors in one test is rather unlucky. Recently, Akl et al. [3] study the anchor placement for passive positioning, and they find that the optimal placement is that

no three anchor nodes are collinear at the center of network. The authors of [4] point out that the optimal placement of anchors should be around the corners of the network and also find that the more nonlinearity results in the better positioning performance.

Doherty et al. [5] place the anchors at the corners of the network to acquire a better positioning results. However, the algorithm has a constraint requirements that all the unknown nodes should be placed within the convex hull of the anchors, which reduces the algorithm generality. Ash and Moses [6] analyse and prove that the anchors on the corners of network will help to improve positioning result when the network is a rectangle. Hara and Fukumura [7] also propose an anchor placement algorithm applied to the rectangle network, and that they point out the anchors must be placed in the centers of subrectangle regions divided from the rectangle network.

Some anchor placements focus on the effect of the environment. For example, the authors of [8] conduct some experiments where anchors are placed either on the ceiling or the floor. The study find that anchors on the floor are better for monitoring moving people in the room. Although many anchor placement works are developed, they only focus on how to improve the positioning performance based on anchor positions and ignore how to acquire the positions of anchors. This paper analyses the artificial calibration problem to acquire the anchor positions after deterministic placement. In order to efficiently acquire the anchor positions, we introduce MDS method to construct a radiomap corresponding to the blueprint. Then anchors physical positions can be self-calibrated by mapping the radiomap to the blueprint.

MDS method is a series of analysis techniques used for displaying the data proximity as a geometrical picture [9]. At present, there are many variants of MDS positioning algorithm, including classical metric MDS-MAP(C), distributed MDS-MAP(P), local MDS, and weighted dwMDS(G). Centralized MDS-MAP(C) [10] algorithm is the earliest usage of MDS techniques in sensor network positioning. Since MDS-MAP(C) uses the shortest hop distance as the estimate of the true Euclidean distance, it is not good for irregular network. A distributed MDS method, MDS-MAP(P) [11], is proposed to be applied to different network topologies. MDS-MAP(P) first constructs a 2-hop local map by executing MDS-MAP(C) method for nearby nodes then merges each local map into a global map based on the common nodes. Local MDS [12] is another distributed variant of MDS-MAP(C) improved for irregular topologies. The difference from MDS-MAP(P) is that the nearby nodes of constructing local map only include 1-hop neighbors and the weights are restricted to 0 or 1. Meanwhile, a least square optimization method is used for refining the local maps. The dwMDS(G) [13] is a weighted distributed MDS method, in which a weighted (Gauss kernel) cost function is adopted for adaptively emphasizing the most accurate range measurements. Besides, dwMDS(G) designs a neighbor selection method to avoid the biasing effects of noisy range measurements neighbors.

In this paper, we design a distributed MDS-KM method to increase the efficiency of anchor placement. At first, we design an MDS-MAP(A) method focusing on the anchor

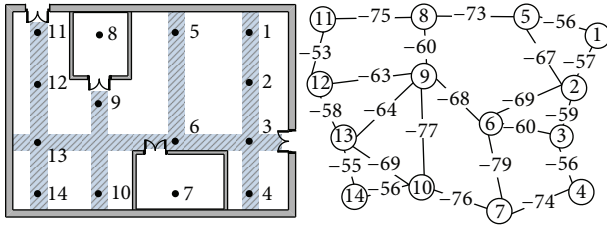


FIGURE 1: The blueprint and the corresponding radiomap.

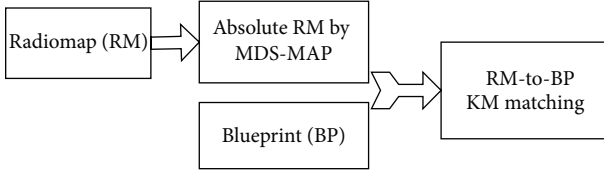


FIGURE 2: The MDS-KM algorithm framework.

positioning to construct a radiomap with absolute coordinates, which is not subject to the irregular anchors distribution. Afterwards, we use the KM algorithm to obtain the maximal weighted matching of complete bipartite graph constructed by the radiomap and the blueprint. Besides, our MDS-KM method can also avoid the error-prone mapping during the artificial calibration.

3. Problem Specification

The optimized placement of anchors has a very important impact for the positioning performance. For improving the positioning accuracy, a predefined blueprint is usually constructed to guide the anchor placement before deploying the positioning system. That is called deterministic anchor placement. For example, the left graph in Figure 1 is a blueprint, where the black cycles are the positions to place anchors. During the placement, anchor node ID will be one-to-one mapped to the anchor position marked on blueprint, which is called artificial calibration. This process will consume a higher labor cost and lead to error-prone mapping. In order to solve this problem, we build a radiomap using the connectivity in large sparse network or signal strength between anchors in small dense network and then adaptively map the radiomap to the blueprint with little or no artificial calibration. As shown in the right graph of Figure 1, the vertices in the radiomap represent anchor nodes and the edge weights represent the signal strength in the small network. In sparse network, many anchor nodes may have only few neighbor anchors or even none. Here, we make the shortest hop distance from one anchor to another anchor as the weight in the radiomap.

Accordingly, the problem to be solved becomes the exact mapping from the radiomap to the blueprint. Intuitively, the radiomap has similar characteristics with the blueprint. The radiomap-to-blueprint mapping should be graph isomorphism (GI) problem [14]. But in the small network, it does not

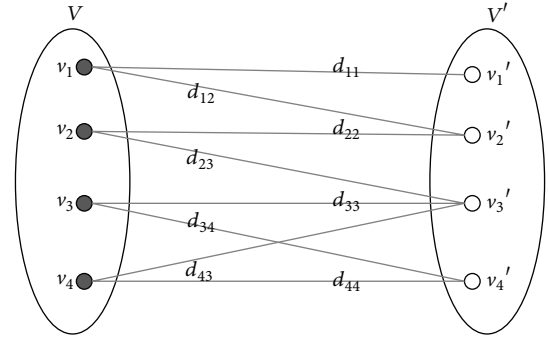


FIGURE 3: A complete bipartite graph.

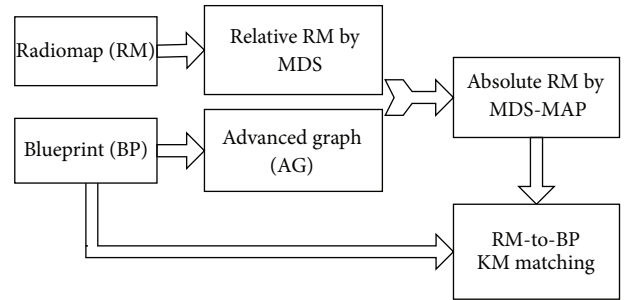


FIGURE 4: The improved MDS-KM algorithm framework.

strictly belong to graph isomorphism problem. Supposing the radiomap and the blueprint are isomorphic, each vertex and edge in both graphs must have a corresponding bijection. As the physical distance increases in the blueprint, the RSS in the radiomap damps and even disappears, but the physical distance can still be measured. Thus the blueprint is a complete graph, while the radiomap is a subgraph of the blueprint. Even though we limited the maximum measure distance in the blueprint, the edges in the blueprint may still not have a corresponding bijection to the edges in the radiomap due to the effect of the surrounding noise. The edges in the radiomap only have a corresponding bijection with the subset of the blueprint. This is a typical subgraph isomorphism problem [15].

However, subgraph isomorphism is an NP-complete problem [16]. Furthermore, the distances between vertices in the blueprint do not exactly reflect the RSS values in the radiomap subjected to the surrounding noise. Therefore, the existing heuristic subgraph isomorphism algorithm is not suitable for the radiomap-to-blueprint mapping. In this paper, we design an MDS-KM matching algorithm to solve this mapping problem in the small network or the sparse network. We introduce the multidimensional scaling (MDS) method in the anchor placement, which is well suited to compute a relative coordinates map in a low-dimensional space by one matrix representing distance information between nodes. Based on MDS method and sufficient known-position calibrated anchors (3 or more), we design a distributed MDS-MAP(A) method to construct the radiomap with absolute coordinates. Then the Euclidean distances of vertices in the

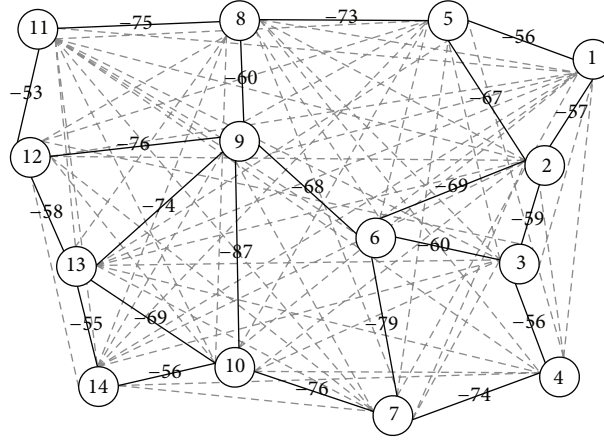
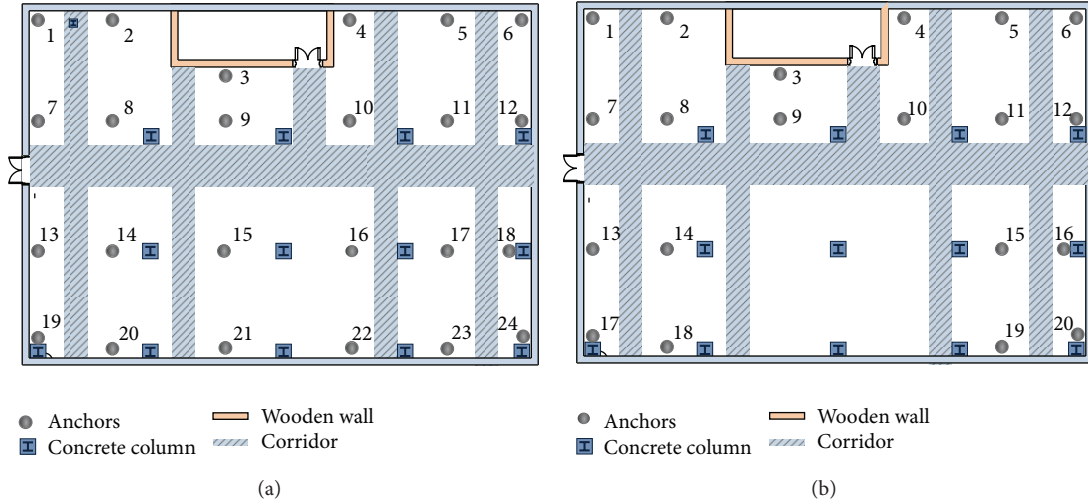


FIGURE 5: The advanced graph in the small network.

FIGURE 6: Two experiments in uniform and n -sharp distribution.

radiomap and the blueprint are computed as the weights to construct a weighted bipartite graph, where one part of the bipartite graph includes all the vertices in the radiomap, and the other part of the bipartite graph includes all the vertices in the blueprint. Afterwards, we adopt the classical Kuhn-Munkres (KM) method [17] to carry out a maximum weight matching of the bipartite and then get a one-to-one mapping between anchor node IDs in the radiomap and positions in the blueprint.

4. Radiomap-to-Blueprint Mapping

4.1. Algorithm Overview. As mentioned above, the matching between the radiomap and the blueprint is our primary objective. The MDS-KM matching process is illustrated in Figure 2. In general, the MDS method utilizes the physical distance between anchors to construct a relative coordinate

radiomap. But the edge weights in the radiomap of the small network represent the RSS values. We need to transform RSS value to the estimated distance according to the signal propagation model. Then we use the MDS method to get a radiomap with relative coordinates. Having sufficient anchor node positions (3 for 2D networks and 4 for 3D networks), we can map the relative coordinates of anchors to absolute coordinates through a linear transformation [10]. Then we use KM algorithm to compute the optimal complete matching between the blueprint and the radiomap with absolute coordinates. Since the KM algorithm is applied to the weighted bipartite graph matching, we need to construct a bipartite graph utilizing the radiomap and the blueprint. Thus we design an error-torrent kNN vertex selection method to build a bipartite graph. Finally, we achieve the mapping from the radiomap to the blueprint through computing the maximum weighted matching of the bipartite. In Section 5,

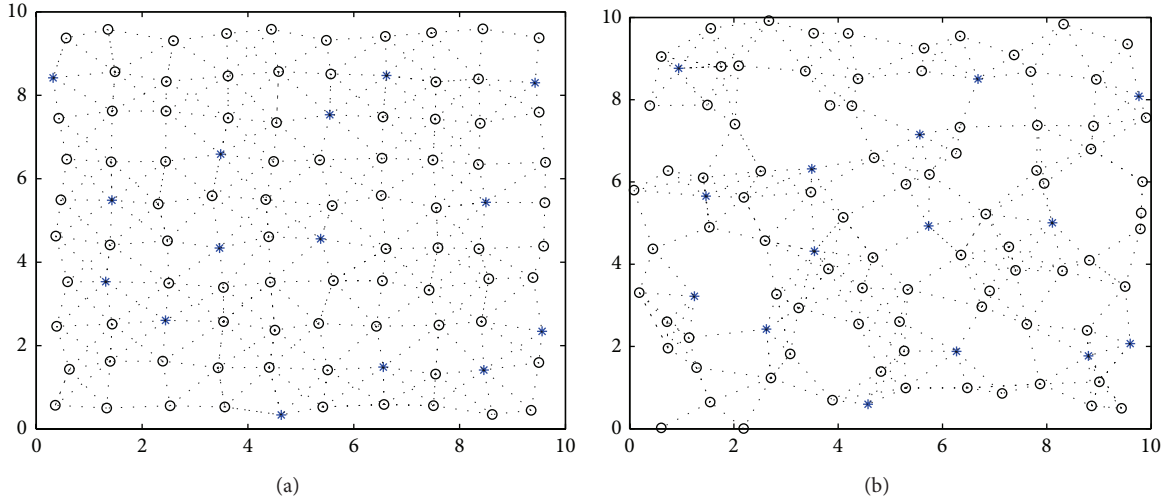


FIGURE 7: Two simulations in grid and random distribution.

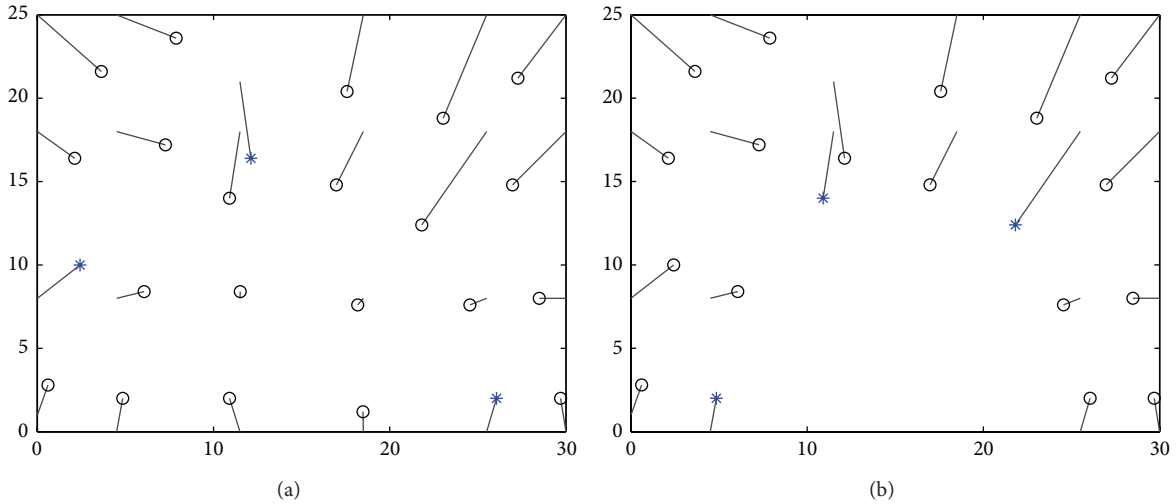


FIGURE 8: The average errors in both experiment scenarios.

we further design a best feature matching (BFM) method to relax the restriction of MDS-KM where three or more calibrated anchors are needed.

4.2. Absolute Radiomap Construction

4.2.1. Collecting Distance Information. In order to construct a radiomap, we need to compute the estimated Euclidean distance based on the RSS or hop distance between anchors. The RSS or hop distance of each pair of anchors should be obtained at first. Intuitively, flooding is a better selection. In the small dense network, each anchor node broadcasts the beacon packet periodically and keeps on receiving the beacon packets from other anchors then computes the RSSs of these beacon packets. After a while, each anchor will record an RSS sequence from other anchor nodes within its 1-hop communication range. Finally, each anchor sends its node

ID and RSS sequences to the backend positioning server for constructing the radiomap. In order to avoid the sending collision, we will make the broadcast cycle of each anchor different in our experiment.

Additionally, in large sparse network, many anchors may not be within the communication range of any other anchors. These anchors are isolated. We will use the shortest hop distances as the estimated distance. There are some intermediate unknown-position nodes scattered within the anchors. The shortest hop distance is defined as the minimum hop count between anchors multiplied by the average signal hop distance. In this process, each anchor will broadcast its beacon packet periodically. Each intermediate unknown-position node records the minimal hop value and adds itself to the value and then forwards the hop count continually with initial anchor ID until the beacon packet arrives to a new anchor or achieves our hop limit. In order to reduce the

communication cost, we set a hop upper limit (e.g., 10) to construct local map. Each anchor records all the minimum hop counts from nearby anchors and sends them and their node IDs to the positioning server.

4.2.2. Estimated Distance. In the large sparse network, we can compute the Euclidean distance between the calibrated anchors. According to the minimum hop counts between them, we further compute the average single hop distance. Accordingly, we can compute the hop distance between each pairwise anchors as the estimated distance. In the small dense network, we need to use signal propagation model to compute the estimated distance based on the RSS value. According to whether the travel distance is short or large, the propagation models can be classified into large scale and small scale [18]. In general, the small-scale model needs to characterize the rapid fluctuations of RSS over short travel distance. It has a better accuracy than large-scale model, but it is very difficult to determine the model parameters. In this paper, we concentrate on the generality of the designed algorithm and do not consider a specific scenario. Hence, we select a good compromise between simplicity and accuracy, which is called the wall attenuation factor propagation model (WAF) [19]. This model provides flexibility when applied to indoor scenario while considering outdoor large-scale fading. This model is described as

$$p(d) [dbm] = p(d_0) [dbm] - 10\alpha \log\left(\frac{d}{d_0}\right) - \delta \quad (1)$$

$$\delta = \begin{cases} nw \times \text{WAF}, & nw < C \\ C \times \text{WAF}, & nw \geq C, \end{cases}$$

where d is the transmitter-receiver distance, $P(d_0)$ is the signal power at some reference distance d_0 , α indicates the rate at which the signal fades, C is the maximum number of obstacles up to which the attenuation factor makes a difference, nw is the number of obstacles between the transmitter and the receiver, and WAF is the obstacle attenuation factor. In general, the values of α and WAF depend on the specific propagation environment and should be derived empirically. Given the RSS value, we can further compute the estimated distance d_e as follows:

$$d_e = d_0 \times 10^{(p(d_0)[dbm] - p(d)[dbm] - \delta)/(10\alpha)}. \quad (2)$$

Additionally, there are some optimization methods to tune parameters of propagation model so that the RSS measurements can characterize the accurate distances [20–22]. In our algorithm, the MDS method can tolerate error gracefully due to the overdetermined nature of the solution [9]. Hence we do not need exactly RSS values depending on optimizing the propagation model.

4.2.3. Constructing Absolute Radiomap. In this part, we will use the MDS method to construct the absolute radiomap. At present, many types of MDS techniques have been developed [9]. In our algorithm, we design a distributed MDS-MAP(A) algorithm focusing on the anchor placement. The MDS-MAP(A) algorithm consists of four main steps as follows.

First, we use the above estimated distance to construct the 1-hop proximity matrix P for each anchor, where the 1-hop neighbors of anchors in large network will be the anchors in the range of hop upper limit. We denote the proximity measure between anchor i and j as p_{ij} . Then assuming an m -dimensional space, given the anchor i coordinates $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ and the anchor j coordinates $X_j = (x_{j1}, x_{j2}, \dots, x_{jm})$, the practical Euclidean distance between anchor i and j is denoted by d_{ij} which will construct a Euclidean distances matrix D as

$$d_{ij} = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2}. \quad (3)$$

In theory, the matrix P should be equal to the matrix D . But the estimated distance with errors makes them unequal. In this case, the MDS method can ensure P is approximate to D as far as possible.

Second, we run the MDS algorithm for each distance matrix P to get a local map with relative coordinates. In classical metric MDS, the proximity matrix P can be transformed to a double centered matrix B , which is symmetric and positive semidefinite matrix as

$$B = -\frac{1}{2} \left(P_{ij}^2 - \frac{1}{n} \sum_{j=1}^n P_{ij}^2 - \frac{1}{n} \sum_{i=1}^n P_{ij}^2 + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n P_{ij}^2 \right). \quad (4)$$

When we shift P to the center, B can also be expressed as follows:

$$B = XX^T = \sum_{k=1}^m x_{ik} x_{jk}. \quad (5)$$

We perform the singular value decomposition (SVD) on B to get $B = VAV^T$, which has complexity of $O(k^3)$, where k is the number of anchors in the local map. Thus, the complexity of computing n local maps is $O(k^3 n)$, where n is the number of anchors in the radiomap. The coordinate matrix is $X = VA^{(1/2)}$, where $A = \text{diag}(l_1, l_2, \dots, l_n)$ is the eigenvalue diagonal matrix in descending order. $V = [V_1, V_2, \dots, V_n]$ is the eigenvector corresponding to the eigenvalue. We select the first m eigenvectors to construct a coordinate matrix in lower dimension. This is the best low-rank approximation between matrix P and D in the least-squares sense.

Third, we merge all local maps to the whole relative radiomap. Each local map is a group of 1-hop neighbors. We randomly select a local map as the base map and then sequentially merge the neighbor local map according to the common nodes. Eventually, the base map grows to cover the whole radiomap. As known from [11], the complexity of this step is the same as step 2.

Finally, given sufficient calibrated anchors, we map the relative coordinates to the absolute coordinates of anchors through a liner transformation [10], which include scaling, reflection, and rotation. The radiomap with absolute positions can be achieved eventually. For r anchors, the complexity of this step is $O(r^3 + n)$.

4.3. Radiomap-to-Blueprint Matching. Since the surrounding noise and irregular topology affect the precision of estimated distance and lead to the inaccuracy absolute coordinates of anchors in the radiomap, the absolute coordinates in the radiomap are not completely consistent with the coordinates of anchor physical positions in the blueprint. Hence, the above two groups of coordinates cannot be corresponding completely. We only search for the most approximate matching of two coordinates. Therefore, the objective of the radiomap-to-blueprint matching turns into minimizing the sum of corresponding Euclidean distances between the physical positions in the blueprint and the absolute coordinate positions in the radiomap. We present a k -nearest neighbor (kNN) method to find the best approximate positions in two graphs. The k -nearest neighbor is a simple classification method in the data mining field. This algorithm can select the k -nearest ones through evaluating Euclidean distance between positions. For each anchor in the radiomap, we utilize the kNN method to find the k -nearest positions in the blueprint away from it. Then we can build a weighted bipartite graph, whose weights on edges are the Euclidean distances. An example with $k = 2$ is shown in Figure 3. Additionally, the value of parameter k is task specific. In our algorithm, we select the minimal k to guarantee that all the positions in the blueprint will be selected into V' when all anchors V in the radiomap have been carried out in the kNN operation. Thus the bipartite graph has a complete matching, where every vertex of the graph is exactly incident to only one edge.

Accordingly, the radiomap-to-blueprint matching problem will be transformed into a minimum weighted matching problem in a weighted bipartite graph, where the sum of the weight of all the edges in the bipartite matching is minimal. Such a matching is also known as the optimal assignment problem. It can be solved by Kuhn-Munkres (KM) algorithm in polynomial time. However, the KM algorithm just applies to solving the maximum weighted matching problem. We need to pick the minus of the weights in the bipartite so that the minimum weighted matching problem is further transformed into a maximum weighted matching problem. The KM algorithm will use vertex labeling method to transform the maximum weighted matching into complete matching in unweighted bipartite graph and then use the classical Hungarian algorithm to solve the maximum matching problem of unweighted bipartite graph.

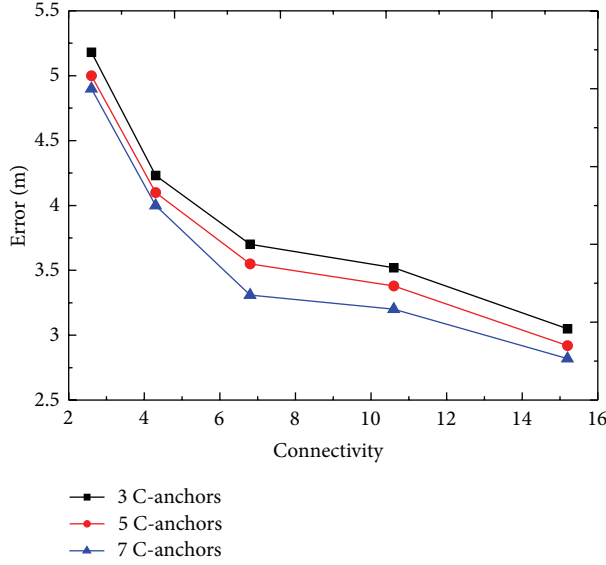
Algorithm 1 is a simplified KM algorithm procedure. We first initialize a feasible vertex labeling. Normally, each vertex in one side of the bipartite graph is labeled with the maximum weight of its incident edges connected to the vertices in the other side, and each vertex in the other side is labeled zero (line 2–6). The bipartite graph will become an unweighted bipartite graph. Then we seek a maximum matching using Hungarian algorithm and decide whether the maximum matching is a complete matching or not (line 7–8). If the maximum matching is a complete matching, we save the matching and return. Otherwise, we need to relabel the vertices following the KM algorithm rules and literately carry out the Hungarian algorithm (line 12–13). Finally, we can achieve a complete matching and get the mapping relationships between the radiomap and the blueprint.

5. Without Calibrated Anchors

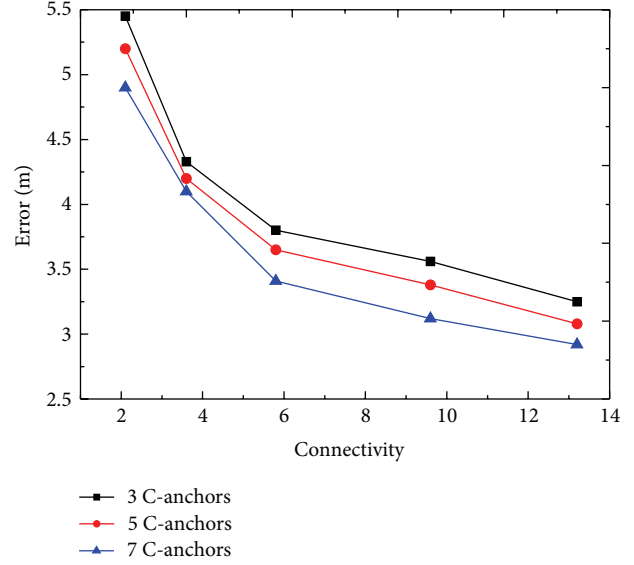
In this section, we try to relax our MDS-KM algorithm to be applied to the situation without any artificial calibrations. We design a best feature matching (BFM) method to actively get parts of mapping from anchors in the radiomap to positions in the blueprint without any artificial calibration. In order to distinguish the feature of vertices in the radiomap and the blueprint, we bring in the vertex weighted sequence as the feature metric, where the edge weight is RSS value or hop count. Then some vertices with best unique feature in the radiomap can be selected and their corresponding vertices are found in the blueprint by our BFM method. However, the edge weight in the blueprint is physical distance. The vertex weighted sequences in the radiomap are not comparable to those in the blueprint because of the different types of the edge weight. Hence, we transform the blueprint to an advanced graph (AG), whose vertex features are the RSS sequences in the small network and hop count sequences in the large network. The new matching process of MDS-KM algorithm is also changed to Figure 4. The advanced graph is used to seek the parts of anchors with a unique feature instead of the calibrated anchors to construct the absolute radiomap.

5.1. Blueprint to the Advanced Graph. In the small network, the distances between vertices in the blueprint are not exactly reflecting the RSSs in the radiomap due to the surrounding obstacles and noise. We first use the signal propagation model mentioned in the above subsection to transform the distances between vertices in the blueprint into the RSS values, which is constructed in an advanced graph denoted by $G_A = (V_A, E_A)$. These RSS values represent the weights of the edges in the advanced graph, and the number of vertices and edges in the advanced graph is the same as that of the blueprint. Since any two vertices in the blueprint have one edge, the advanced graph is also a complete graph. Figure 5 is an example of the advanced graph from the blueprint in Figure 1. In the large network, we compute the minimal hop counts between pairwise anchors in the blueprint after setting the communication range of node and then construct an advanced graph whose edge weights represent minimal hop counts. Similarly, the advanced graph in the large network is also a complete graph.

5.2. Best Feature Matching. Before executing the MDS-MAP(A) method, the radiomap $G_R = (V_R, E_R)$ has the vertex set V_R and edge set E_R . The edge weight represents the RSS or hop count. We first make the vertices distinguishable depending on their invariants, which are the fixed properties of vertices during matching. A simple invariant is the vertex degree. However, in a graph, the vertex degree is not unique. There is likely to be many vertices having the same degree. Therefore, we bring the weights into the vertex invariants, for example, $I(v_i, W) = (v_i, w_1, w_2, \dots, w_d)$ by following the arrangement $w_1 > w_2 > \dots > w_d$, and d is the degree of the vertex v_i . Similarly, we can formulate the corresponding vertex invariants of the advanced graph. For

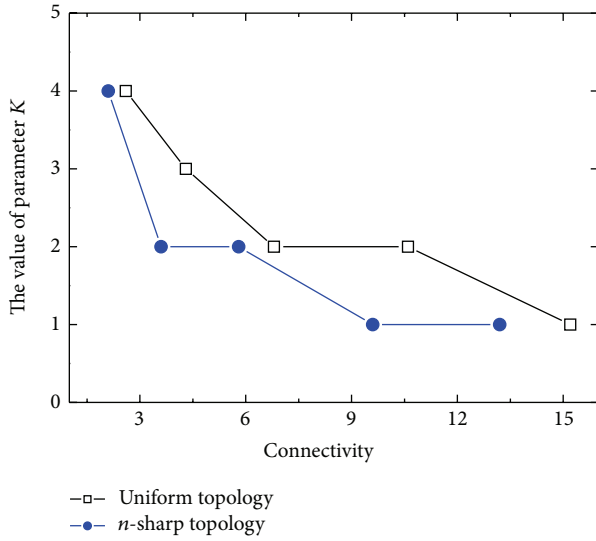


(a)



(b)

FIGURE 9: The error analysis in both experiment scenarios.

FIGURE 10: The selection of the K using for constructing bipartite graph.

example, $I'(v'_i, w') = (v'_i, w'_1, w'_2, \dots, w'_{n-1}), w'_1 > w'_2 > \dots > w'_{n-1}$. n is the number of all vertices. Each vertex degree is $n - 1$ since the advanced graph is a complete graph.

We will select the vertices invariants in the radiomap which are the most easy to distinguish. We noted that the degrees of many vertices in the radiomap are different so that the number of weights in some vertex invariants is inconsistent. This brings inconvenience to our feature comparison. Therefore, we need to normalize the vertex invariants of the radiomap. We first compute the maximal degree of all vertices $\text{Max}(d)$ in the radiomap then extend the vertex invariant $I(v_i, W)$ from (v_i, w_1, \dots, w_d) to $(v_i, w_1, \dots, w_d, w_{(d+1)}, \dots, w_{\text{Max}(d)})$, where

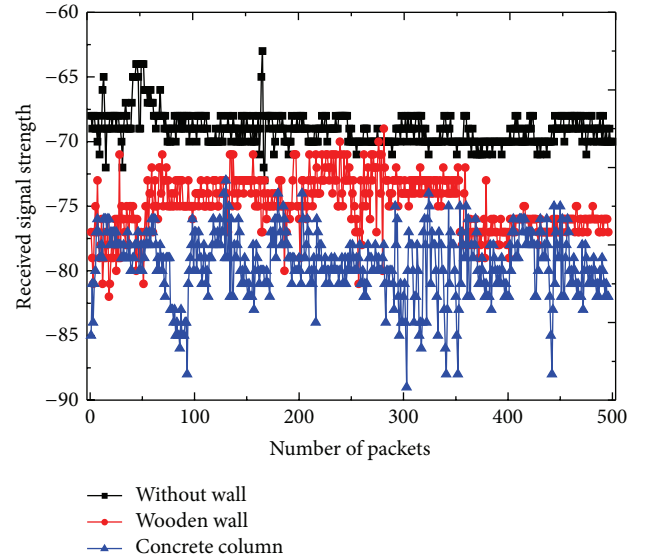


FIGURE 11: The obstacle affecting.

$d \leq \text{Max}(d), w_{(d+1)} = w_{(d+2)}, \dots, w_{\text{Max}(d)} = w_{\min}$. w_{\min} is the minimum RSS value measured from anchor device in the small network or hop count of zero in the larger network. We can compute the Euclidean distance d_{RA} between vertices invariants in two graphs as follows:

$$d_{RA} = \sqrt{\sum_{i=1}^{\text{Max}(d)} (w_i - w'_i)^2}. \quad (6)$$

We still adopt the k -nearest neighbor ($k = 2$) method to find the two minimum d_{RA} between vertices in the radiomap and vertices in the advanced graph. For each vertex in the

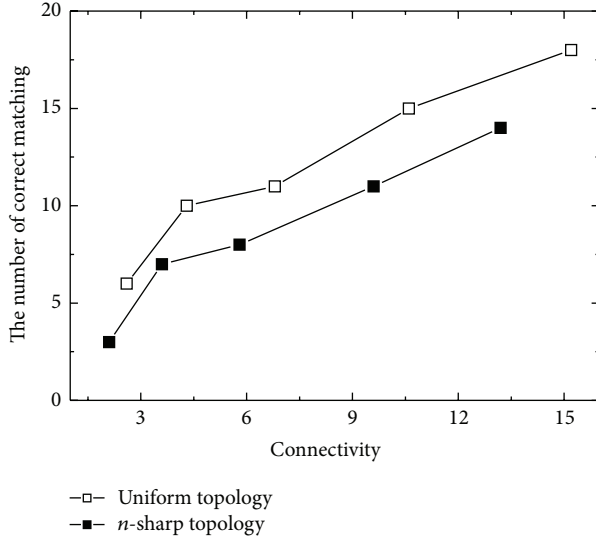


FIGURE 12: The results of BFM method in the small network.

radiomap, the absolute value of the difference of the two minimum Euclidean distances can be computed and sorted in descending order. The bigger the absolute value of the difference, the more unique the vertex features. So the vertices in front of the order are the most possible unique and distinguishable ones. They can actively catch their corresponding minimum Euclidean distance vertices in the advanced graph. To some extent, this method is subject to the symmetry of anchors in the blueprint. But we can artificially design the blueprint keeping asymmetric. Meanwhile, the irregular environment also affects the symmetry of the blueprint. Therefore, in practice, the weights of anchors in the blueprint are hardly perfectly symmetric.

6. Implantation and Experiment

6.1. Experiment Design. In our experiment, we will run MDS-KM algorithm on a variety of anchor topologies in the small and large networks. In the $30\text{ m} \times 25\text{ m}$ room, the anchors are installed on the ceiling or concrete columns. (1) Figure 6(a) is the placement blueprint, where there are 24 positions to place anchor nodes. Concrete columns and wooden walls in the room are the principal obstacles affecting communication quality between anchors. (2) We simplify the topology of Figure 6(a) into an n -sharp topologies of 20 positions as shown in Figure 6(b).

In the large network, we simulate the anchors in the MATLAB placed with grid distribution and random distribution, respectively, as shown in Figures 7(a) and 7(b). A number of 100 nodes are placed uniformly and randomly in a $10\text{ r} \times 10\text{ r}$ multihop network, where 85 nodes are intermediate unknown-position nodes denoted by the circle, and 15 nodes are anchors denoted by the stars (*). For the purpose of facilitating the comparison of positioning error, we select the similar anchor positions in both topologies to construct the radiomap.

It should be noted that the complicated office room is more sensitive to the noise than outdoors. Meanwhile, the most indoor positioning systems are usually deployed deterministically according to the placement blueprint. Therefore, we choose the indoor environment as the case of the small network, which is more powerful to verify the MDS-KM performance.

6.2. The Small Network. During the radiomap construction, we set each anchor ID number multiplied by 100 milliseconds as its broadcast cycle to avoid the sending collision. After running 2 minutes, we compute the average RSS values between anchors. We use our MDS-MAP(A) method in the topologies (1) and (2) for constructing the absolute radiomap based on 3 random calibrated anchors, denoted by the stars (*) as shown in Figure 8. The circles represent the estimated absolute positions, and the solid lines represent the errors between the estimated positions and the true positions. The longer the solid line, the larger the positioning error. The transmitting power of TelosB in TinyOS system is classified into 1 to 31 levels. With the level rising, the transmitting power becomes higher. We set the highest level of transmitting power in this group of experiments. The results show that we have the average estimation errors of 3.05 m and 3.25 m in two topologies.

Figure 9 shows the average performance of MDS-MAP(A) positioning affected by connectivity and numbers of calibrated anchors. Figures 9(a) and 9(b) show the results of MDS-MAP(A) positioning of two topologies, respectively. We set the transmitting power levels as 11, 17, 21, 26, and 31, respectively, in our experiments. Three, five, and seven calibrated anchors are used. Then we get the connectivity levels of 2.6, 4.3, 6.8, 10.6, and 15.2 in the uniform topology, and 2.1, 3.6, 5.8, 9.6, and 13.2 in the n -sharp topology. With the lowering of the connectivity level, the positioning performance declines significantly. When the connectivity level is less than 3, the average error will be achieved to around 5.5 m. Besides, the positioning error becomes lightly lower with the increasing of C-anchors. Meanwhile, the different numbers of calibrated anchors also have very close positioning errors. Therefore, a certain range of a number of variations of calibrated anchors has no significant influence on positioning performance.

We obtain a radiomap with absolute coordinates after MDS-MAP(A) operation. Before running the KM matching, we need to set the parameter k for constructing a bipartite graph. In our experiment, we show the minimal k to producing a complete bipartite graph in Figure 10. With the connectivity level rising, the value of k reduces gradually. When the connectivity level is 15.2 in uniform topology and 9.6 and 13.2 in n -sharp topology, the value of k is 1. That means that the bipartite graph is already a one-to-one mapping complete graph. Then we can obtain the optimal matching between the blueprint and the radiomap without the KM method. Meanwhile, we find that this mapping is also exactly correct. Under other connectivity levels, we must use the KM method to find the optimal matching. We find that the rate of correct matching between anchors in the

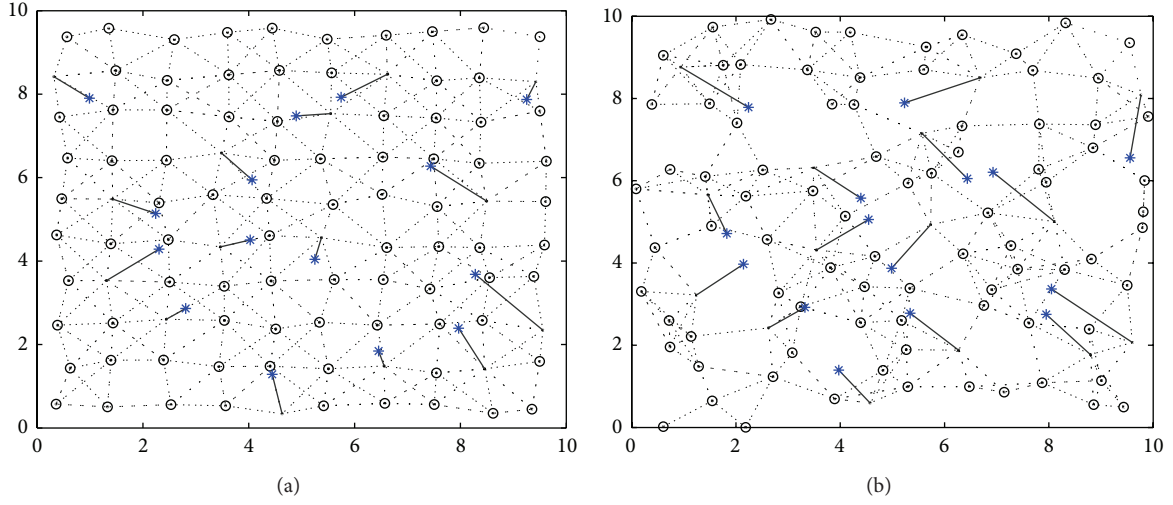


FIGURE 13: The average errors in both simulation scenarios.

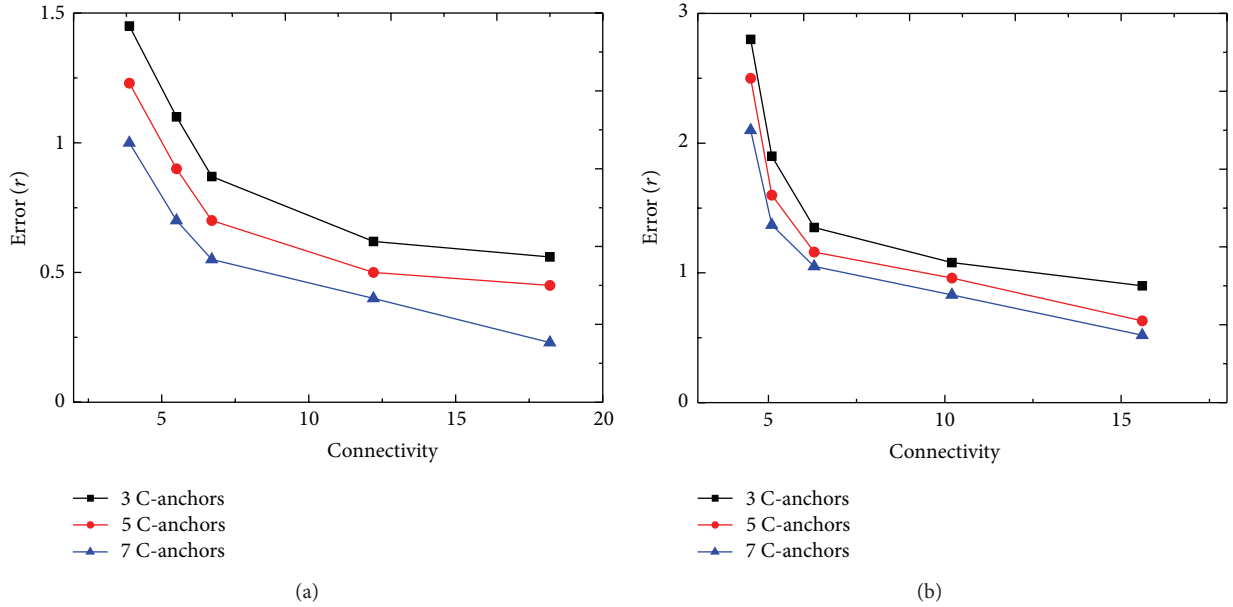


FIGURE 14: The error analysis in both simulation scenarios.

radiomap and positions in the blueprint can achieve 100% when connectivity level is over 3. Only when the connectivity level is less than 3, there are two anchor nodes with error mapping in both topologies, where the node IDs are 3 and 10, respectively. This is because both nodes are close to each other. The positioning error from the MDS-MAP(A) method will make their positions confused so that the maximum weighted matching of the KM method is not exactly the mapping from the radiomap to the blueprint. Meanwhile, we also observe that the more calibrated anchors cannot help the accuracy of the KM matching unless the anchors with error matching are calibrated anchors.

In order to validate the performance of our BFM algorithm, we need to exactly transform the physical distance of the blueprint into RSS value of the advanced graph. At first,

we make a measurement test for determining the parameters WAF and α in (1). During our experiment, we test two types of obstacle materials, 40 cm width wooden wall and 60 cm \times 60 cm width concrete column. Two TelosB nodes lie in two sides of obstacle and 2 m away from the obstacle. One node broadcasts beacon packet every 10 seconds, while another node receives the packet and computes the RSS value. We spend 80 minutes to get the results shown in Figure 11. We find that the wooden wall and concrete column can approximately reduce RSS 5 db and 10 db, respectively. Based on the measurement, we further compute the fading factor α in our environment, which is approximate to 3. Then we use the experimental values to construct the advanced graph.

Figure 12 is the number of correct matching anchors with connectivity increasing during the BFM process. The number

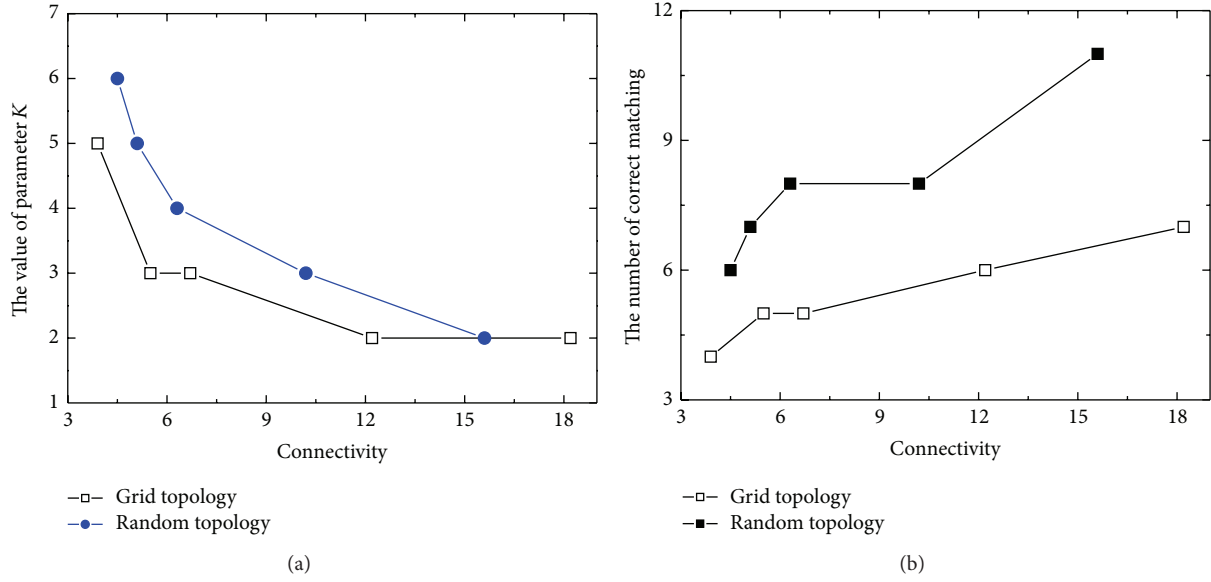


FIGURE 15: The results of BFM method in the large network.

```

1:  $G(X, Y, W)$  /*  $G$  is a bipartite Graph,  $W$  is the Weight. */
2: for all  $(x \in X \text{ and } y \in Y)$  do
3:   /* Initialize all vertices labeling */
4:    $l(x) = \text{Max}\{w(x, y), y \in Y\}$ 
5:    $l(y) = 0$ 
6: end for
7:  $M = \text{Hungarian}(G(X, Y, l))$ 
8: if ( $M$  is complete matching of  $G$ ) then
9:    $\widehat{M} = \text{save}(M)$ 
10:  return  $\widehat{M}$ 
11: else
12:  relabeling( $l$ ) /* as KM rules */
13:  goto 7
14: end if

```

ALGORITHM 1: The Kuhn-Munkres Algorithm.

in the uniform case is lightly more than that in the random case, which is mainly due to more quantity of anchors in uniform topology. Meanwhile, we find that there are three or more anchors at least with correct matching even when the connectivity is lower than 3 in two topologies. Therefore, we can run our MDS-KM method in all the above experiments without any calibrated anchors, which further reduces the labor cost. But unfortunately our BFM method cannot help to solve the error mapping of the MDS-KM method under the lower connectivity.

6.3. The Large Network. We run MDS-MAP(A) method for the grid and random topologies of the large network to construct the absolute radiomap based on 3 random calibrated anchors as shown in Figure 13. The circles represent unknown-position intermediate nodes. The stars represent the anchor nodes, and the solid lines represent the errors

between the estimated positions and the true positions. In the $10r \times 10r$ area, we set the communication range as $1.5r$ and $2r$, respectively, in the grid and random topologies. The average connectivity levels of both topologies are 6.7 and 6.3, respectively. Although both connectivity levels are similar, the positioning errors have a big difference. After running the MDS-MAP(A) method for the radiomap, we have the corresponding average estimation errors of $0.87r$ and $1.35r$ in both topologies. This is because the connectivity level of nodes in the random case is uneven so that its estimated error of hop distance is significantly bigger than that in the grid case. Therefore, the corresponding absolute radiomap in the random case has also a bigger average estimation error.

Additionally, we compare the performance of the MDS-MAP(A) method in different connectivity levels and calibrated anchors. In both topologies, we select 3, 5, and 7 calibrated anchors randomly to construct the absolute radiomap

during every trail. In the grid topology, the radio ranges are from $1r$ to $2r$, with an increment of $0.25r$, which result in the connectivity of 3.9, 5.5, 6.7, 12.2, and 18.2, respectively, as shown in Figure 14(a). We find that the higher connectivity level will bring about a better positioning result, and the more calibrated anchors also improve the positioning performance. When connectivity level is lower than 6.7 especially, the average estimated error will increase significantly. In the random topology, the radio ranges are from $1r$ to $3r$, with an increment of $0.5r$, which lead to average connectivity of 4.5, 5.1, 6.3, 10.2, and 15.6, respectively, as shown in Figure 14(b). This design is to compare the performance of the MDS-MAP(A) algorithm under the similar connectivity levels of both topologies. We can see that the positioning performance in the random topology has a significant reduction than that in the grid topology. The maximum average estimated error is even twice that in the grid topology. That is mainly because the estimated hop distance in the random topology is rather inaccurate.

Figure 15(a) is the k -value selection of both topologies. We can find that the k in the random topology has a higher value than that in the grid topology. This is because the higher errors of the estimated hop distance in the random topology produce the bigger position errors of the absolute radiomap. Thus the anchors in the radiomap cannot exactly correspond with the positions in the blueprint. In order to get a complete bipartite graph, k -value must be increased. Afterwards, we find that the KM method can reach a 100% rate of correct matching except that there are 3 and 2 error-matching anchors, respectively, under the connectivity of 4.5 and 5.1 in the random topology. It is further suggested that the MDS-KM algorithm is well suited to the higher connectivity network.

Figure 15(b) reflects the BFM method performance in both topologies of the large network. In the random topology, the BFM method can obtain a better feature matching result. This is because many vertices in the grid topology have the same hop count sequences subjected to the symmetry of anchor distribution. Therefore, the vertices invariants in the grid topology are hard to be distinguished, while in the random topology there are more distinguishable vertices with unique invariants. But in both topologies, we can also find that there are more than three anchors with correct feature matching. In other words, the MDS-KM method can run successfully in two simulation scenarios of the large network without any calibrated anchors.

7. Conclusion

In this paper, we consider the anchor self-positioning problem in detail. During the deterministic anchor placement, we design an efficient mapping algorithm between anchors and positions (MDS-KM) to avoid the expensive labor cost and error-prone features of artificial calibration. Additionally, we propose a best feature matching (BFM) method to obtain some mappings between anchors and positions in advance so that any calibrated anchors are not needed. Experimental results show that the MDS-KM algorithm can achieve the

100% correct matching between anchors and positions under a higher connectivity level. Meanwhile, in our experiments and simulations, the BFM method can obtain sufficient known-position anchors to support the successful running of the MDS-KM method.

Acknowledgments

This work is supported by the General Program of National Natural Science Foundation of China (NSFC) under Grant no. 61073180 and the National Key Basic Research Program of China (973) under Grant no. 2011CB302902.

References

- [1] H. S. AbdelSalam and S. Olariu, "Towards enhanced RSSI-Based distance measurements and localization in WSNs," in *Proceedings of the IEEE INFOCOM Workshops 2009*, pp. 1–2, April 2009.
- [2] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.
- [3] R. Akl, K. Pasupathy, and M. Haidar, "Anchor nodes placement for effective passive localization," in *Proceedings of the International Conference on Selected Topics in Mobile and Wireless Networking (iCOST '11)*, pp. 127–132, October 2011.
- [4] T. Kunz and B. Tatham, "Localization in wireless sensor networks and anchor placement," *Journal of Sensor and Actuator Networks*, vol. 1, no. 1, pp. 36–58, 2012.
- [5] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1655–1663, April 2001.
- [6] J. N. Ash and R. L. Moses, "On optimal anchor node placement in sensor localization by optimization of subspace principal angles," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 2289–2292, April 2008.
- [7] S. Hara and T. Fukumura, "Determination of the placement of anchor nodes satisfying a required localization accuracy," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 128–132, October 2008.
- [8] R. Zemek, M. Takashima, S. Hara et al., "An effect of anchor nodes placement on a target location estimation performance," in *Proceedings of the IEEE Region 10 Conference (TENCON '06)*, pp. 1–4, November 2006.
- [9] I. Borg and P. Groenen, "Modern multidimensional scaling: theory and applications," *Journal of Educational Measurement*, vol. 40, no. 3, pp. 277–280, 2003.
- [10] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 201–212, ACM, New York, NY, USA, June 2003.
- [11] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, vol. 4, pp. 2640–2651, March 2004.

- [12] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, vol. 4, pp. 2652–2661, March 2004.
- [13] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 39–64, 2006.
- [14] D. C. Schmidt and L. E. Druffel, "A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices," *Journal of the Association for Computing Machinery*, vol. 23, no. 3, pp. 433–445, 1976.
- [15] Sansone and M. Vento, "Subgraph transformations for the inexact matching of attributed relational graphs," *Computing*, vol. 12, pp. 43–52, 1998.
- [16] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the 3rd annual ACM symposium on Theory of computing (STOC '71)*, pp. 151–158, ACM, New York, NY, USA, 1971.
- [17] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [18] T. S. Rappaport, *Wireless Communications: Principles and Practice*, IEEE Press, Piscataway, NJ, USA, 1st edition, 1996.
- [19] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proceedings of the 9th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, vol. 2, pp. 775–7784, 2000.
- [20] K. Benkič, M. Malajner, P. Planinsic, and Z. Cucej, "Using RSSI value for distance estimation in wireless sensor networks based on ZigBee," in *Proceedings of the 15th International Conference on Systems, Signals and Image Processing (IWSSIP '08)*, pp. 303–306, June 2008.
- [21] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "Virtual calibration for RSSI-based indoor localization with IEEE 802.15.4," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.
- [22] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proceedings of the 3rd Workshop on Embedded Networked Sensors (EmNets '06)*, 2006.

