Hindawi
Journal of Sensors
Volume 2019, Article ID 7209349, 16 pages
https://doi.org/10.1155/2019/7209349

Research Article

Sparse Sensor Placement for Interpolated Data Reconstruction Based on Iterative Four Subregions in Sensor Networks

Mingshan Xie[1,2,3] **Mengxing Huang**[1,2] **Yong Bai**[1,2] **Zhuhua Hu**[1,2] **and Yanfang Deng**[1]

[1]*College of Information Science and Technology, Hainan University, Haikou 570228, China*
[2]*State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou 570228, China*
[3]*College of Network, Haikou University of Economics, Haikou 571127, China*

Correspondence should be addressed to Yong Bai; bai@hainu.edu.cn

Received 12 May 2018; Revised 12 October 2018; Accepted 30 October 2018; Published 17 January 2019

Academic Editor: Carlos Ruiz

Data acquisition in large areas has issues of cost and data loss. When sensors are sparse in the physical field, it is critical to study the deployment methods to improve the accuracy of reconstructed data set and the precision of the recovery of lost data. It is desirable to place sensors at optimal locations to achieve higher precision of recovery. In this paper, we present a sparse sensor placement scheme for data interpolation reconstruction based on iterative four subregions using fractal theory. The results of our experiments demonstrate that the precision of our algorithm is higher than that with random placement in dispersion degree, coverage rate, and reconstruction accuracy.

1. Introduction

The sensor network is suitable for the information collection in large areas, for example, ocean monitoring [1], large-scale crop growth information monitoring [2], reservoir hydrology information acquisition, and weather forecast. However, when the monitored area is relatively large, if sensors are deployed in every corner, the cost will be very high in these application scenarios. Additionally, the full monitoring of an entire area causes several disadvantages—high cost of deployment, long transmission delay, slow response, and unnecessary data aggregation [3]. When the spatial correlation and time correlation of this monitoring information are high, we can reconstruct the data set of physical field information in the whole region with a small amount of information by effective reconstruction method, for example, the construction of the wind field and the pressure field and the application of the ocean hydrological information. The reasonable sparse sensor deployment is the prerequisite for improving the precision of the reconstruction. It is necessary for us to study the sparse deployment of sensors.

A number of prior works on sensor placement are focused on minimizing the number of sensors or maximizing the sensing quality provided by a network [2, 4–10]. Sparse deployment and data recovery involve data incompleteness. Now, many researchers are studying the reconstruction of incomplete data [2, 11–13]. Sensor deployment is a key to solve incomplete data reconstruction. When the monitored area is not very complex, such as marine hydrological acquisition, reservoir water quality monitoring, and large crop growth information acquisition, the physical quantity of the monitored area is more spatially relevant and the changes in the whole monitored area are relatively gentle. All these methods need to make use of the spatial correlation of data in the monitored area to achieve data reconstruction. The commonly used method is spatial interpolation.

Spatial data interpolation applications are becoming more and more widely used and highly valued by people. Interpolation of spatial data pertains to finding a function relation from a set of known spatial data, which can be a form of discrete points and can be a form of partitioned data, so that the relationship is best approximated by the

known spatial data and the values of other arbitrary points or arbitrary partitions within the range can be calculated according to the relation of the function [14]. The closer the points in space are, the more likely is that they are to have similar eigenvalues and the points that are farther away are less likely to have similar eigenvalues. This is the most basic theoretical assumption of spatial interpolation techniques [12].

In the spatial data interpolation applications, the sensor placement is very critical because inappropriate sensor placement can reduce the accuracy of the data reconstruction.

We study how to place the sensor placement problem based on a data interpolation reconstruction model that captures the development characteristics of target detection. The reconstruction accuracy of the interpolation method is affected by the reconstruction algorithm and sensor deployment. Among them, the optimization of sensor deployment is the primary problem that we have to solve. However, the deployment of sensors is greatly affected by the environment. How to solve the impact of environment on sensor deployment is what we need to consider. The deployment of sensors needs to consider dispersion and uniformity. Interpolation reconstruction algorithm is greatly affected by the distance—the farther away from the sensor, the higher the uncertainty of the reconstruction accuracy. We cannot make individual data too far away from sensors, so the probability of error is very high.

This paper is focused on developing fast sensor placement algorithms based on four subregions for the data reconstruction by interpolation. Our approach is inspired by fractal theory. In particular, we aim to improve the precision of reconstruction for achieving the data set of physical quantity in a large area. The main contributions of this paper are as follows:

(1) We are inspired by fractal theory. Considering the spatial characteristics of the monitored area and the scalability of sensor placement, we propose a sensor placement algorithm based on dividing four iteration subregions to improve the precision of interpolation reconstruction

(2) In the process of reconstructing data by interpolation, the dispersion of the sensors is not distributed enough, which will affect the precision of reconstruction. We put forward the concept of the dispersion degree of sensor placement. The dispersion of the sensors for data acquisition is critical. In order to access the dispersion degree of sensor placement, the concept of expansion area is introduced in this paper

(3) When the number of sensors cannot be completely divisible by 4, to allocate the remainder, this paper proposes a method of alternately allocating redundant sensors based on depth of assignment tree. We have solved the scalability of sensor deployment. Regardless of the number of sensors, they can be deployed according to a given method that is proposed in this paper. Uniform deployment is a special case of the proposed placement algorithm

2. Related Works

Sensor placement is the fundament of data processing for WSN or internet of things (IOT). The purpose of data processing is different, and the method of sensor deployment is also different. To improve the accuracy of signal reconstruction, Manohar et al. [11] explore optimized sensor placement based on a tailored library of features extracted from training data. Sparse point sensors are discovered using the singular value decomposition and QR pivoting.

To characterize or classify a high-dimensional system, in [15], Brunton and coauthors present an algorithm that can solve an ℓ_1 minimization to find the fewest nonzero entries of the full measurement vector that exactly reconstruct the discriminate vector in feature space; these entries represent sensor locations that best inform the decision task. They use the compressed sensing (CS) to find the key points, do not involve all sensor deployment, and do not deploy the sensors in the large area. Wu et al. proposed one scalable random placement algorithm with higher incoherence with the sparse representation basis [16]. They obtain the entire field's soil moisture value via the classical CS recovery algorithm.

The idea of compressed sensing is used in [15, 16]. PCA is used to find data of several key points to retrieve data from the whole monitored area. However, PCA has a constraint that requires sparse representation. In many cases, it is difficult to achieve this condition in actual projects. The situation in this paper is to use a common interpolation method to reconstruct the data of the whole monitored area.

In terms of the sensor placement for data fusion, Chang et al. present fast sensor placement algorithms based on a probabilistic data fusion model [17]. Simulation results show that their algorithms can meet the desired detection performance with a small number of sensors while achieving up to sevenfold speedup over the optimal algorithm.

In terms of the optimization method used in sensor sparse deployment, Chen et al. try to establish a theoretical framework for finding sensor positions to maximize the detection probability with a distributed sensor network [5]. They choose a 1-dimensional line deployment model and present the relevant numerical results. In [18], Moreno-Salinas et al. used tools from estimation theory and convex optimization to achieve the proper choice of the sensor positions, in order to determine the sensor configuration that yields the minimum possible covariance of any unbiased target estimator. In [19], Akbarzadeh and coauthors proposed an adaptation of the gradient descent method, which considers both the topography of the environment and a set of sensors with directional probabilistic sensing to optimize the position and orientation of sensors for the sensor placement problem. In [20], Zhu and coauthors used the quantum genetic algorithm (QGA) to optimize the corresponding sensor network on the upstream surface of a dam.

When the sensors cannot completely cover the monitored area, it can maximize the accuracy of the acquisition data in the whole area to optimize the sensor position according to the importance of different locations. Liu et al. [4] deduced that when the deployment is determined, the error based on Voronoi subdivision is the smallest. They define

the coverage weight of each node partition. Then a deployment optimization algorithm is proposed based on the perception node coverage weight and the virtual force idea. Liu and coauthors [21] proposed a node deployment strategy that combined coverage weight and virtual force for the prioritized event area in mobile sensor networks with low-density sensors, in which the coverage completely cannot be realized. However, they do not deal with situations where all nodes are fixed static nodes.

To cover a sensing area by deploying a minimum number of wireless sensors while maintaining the connectivity between the deployed sensors, Rebai et al. develop an integer linear programming model to solve the problem optimally in [22].

The deployment of sensors is also subject to the monitored environment. In view of the obstacles in the monitored area, in [3], Chang et al. propose an efficient sensor placement (ESP) approach for a sparse interested area with consideration of obstructers that block the data transmission among sensors. For the given number of sensors, in [23], Mukherjee and coauthors proposed algorithms to embed thermal sensors into a regular structure to minimize the number of sensors and determine sensor locations required to maintain a given accuracy in temperature sensing for a given design. In [24], Yoganathan and coauthors proposed a novel data-driven approach based on field measurements in an office building to derive the optimal (number and locations of) measuring points. Clustering algorithms, information loss approach, and Pareto principle were used to derive the optimal sensor placement strategy. There are many studies on the effective transmission of data in sparse deployments. In [25], Du and coauthors provided distanceless transmission to expand the communication range of sensor motes and devised a communication protocol to efficiently coordinate the distanceless link transmissions by leveraging rateless codes. In [26], the authors took the sensor sparse deployment strategy in the positioning system. They proposed FISCP, a fine-grained device-free positioning system for multiple targets working in sparse deployments. They do not study a specific method of sparse deployment.

The deployment of sensors needs to consider the characteristics of monitored objects. For sensor deployment of water pipe leak location, in [27], the authors provide a methodology that incorporates uncertainties of different types and sources in the optimal sensor placement problem for leak localization shown by the example of the effect of demand uncertainties on potential pressure measurement points. In [28], for a performance evaluation of ageing infrastructure, Bertola et al. presented a measurement system design methodology to identify the best sensor locations and sensor types using information-from static load tests. A modified version of the hierarchical algorithm for sensor placement is proposed to take into account mutual information between load tests.

In summary, the purpose of our algorithm is to reconstruct data based on interpolation, which is suitable for large area monitoring. Since our idea comes from fractal theory, our algorithm has strong universality and expansibility.

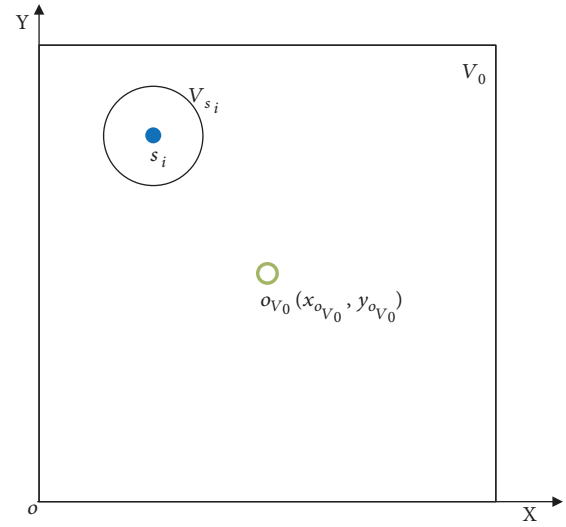

Figure 1: Monitored region with sensors.

3. Problem Formulations

In this paper, we assume that the monitored area V_0 is a rectangular region. A_{V_0} represents the area of the whole monitored area. We assume that the physical information is uniformly distributed in the monitored area, such as the temperature field, the wind field, and the humidity field. When these physical data are collected sparsely, the inverse distance interpolation algorithm can be used to reconstruct the physical quantity of the whole monitored area. The total number of sensors in the monitoring is given, which is represented by n_{V_0}.

In this paper, the lower left corner of the monitored area is set as the coordinate origin. $o_{V_0}(x_{o_{V_0}}, y_{o_{V_0}})$ is the center of the monitoring region, as shown in Figure 1.

The effective coverage area of the ith sensor s_i is V_{s_i}. $O_{V_{s_i}}$ is the central position of V_{s_i}, and its coordinate is $(x_{O_{V_{s_i}}}, y_{O_{V_{s_i}}})$. A_{s_i} is the effective coverage area of the ith sensor. If it is a sparse deployment, then

$$\sum_{i=1}^{n_{V_0}} A_{s_i} < A_{V_0}. \quad (1)$$

It is difficult to evaluate the quality of the sparse deployment by the sensors' connectivity and energy consumption. Because of the sparsity, the connection between sensors is difficult to achieve. For sparse deployment, energy consumption is no longer the main problem. Because of the limited number of sensors, the sleep strategy is not suitable for sparse deployment.

In the sparse deployment process, the monitored area cannot be fully covered when the number of sensors is limited. In the process of using the inverse distance interpolation [29] to reconstruct the data set, the node for the data acquisition is required to be as uniform as possible to avoid the large area of the monitoring void. A large area of monitoring void will not only affect the accuracy of the data reconstruction but

even lead to the leakage. Accordingly, we put forward the concept of the dispersion degree of sensor placement.

Definition 3.1 (dispersion degree of sensor placement). It reflects the degree of sensor deployment approaching uniform placement. When the sensor is uniformly deployed in the monitored area, the dispersion degree of the sensor deployment reaches maximum.

The sensor deployment with a high dispersion degree can obtain information in all directions when using interpolation to restore data, making the basis for recovery more comprehensive and thus more accurate. From the validation of the necessity of the dispersion degree in Section 5.1, we know that it is necessary to introduce the dispersion degree. We use C to represent the dispersion degree of the sensor deployment. In order to calculate the value of C, we also need to introduce the concept of the area of the monitored expansive region of a sensor.

$$A_{V'_{s_i}} = \frac{A_{V_0}}{n_{V_0}}. \tag{2}$$

Definition 3.2 (the area of the monitored expansive region of a sensor). It is the ratio of the area of the monitored region to the total number of sensors. We assume that V'_{s_i} is the monitored expansive region of the ith sensor s_i. $A_{V'_{s_i}}$ is the area of the monitored expansive region of the ith sensor s_i.

The expansion area V'_0 of all sensors in the monitored region is the union of each sensor's expansion area:

$$V'_0 = \bigcup_{i=1}^{n_{V_0}} V'_{s_i}. \tag{3}$$

The value of C can be obtained by the ratio of the area $A_{V'_0}$ of all the sensors' expansive region to the area A_{V_0} of the monitored region.

$$C = \frac{A_{V'_0}}{A_{V_0}}. \tag{4}$$

When the sensor can be added in the monitored region, the dispersion will decrease if each additional sensor is deployed at the same location each time. The dispersion degree is determined by the deployment position of all sensors.

Property 3.1. $(1/n_{V_0}) \le C \le 1$.

Proof. When all sensors are deployed at the same location, the effective monitoring regions and the expansive regions of all sensors are overlapped. At this time, the overlap area is maximal and C reaches minimum. Because

$$A_{V'_0} = \frac{A_{V_0}}{n_{V_0}}, \tag{5}$$

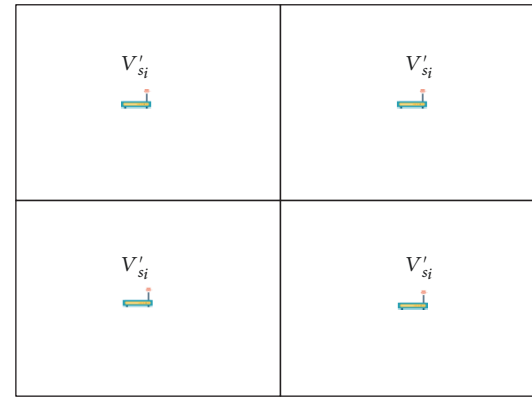

Figure 2: Sensor placement in which C is maximal.

then we have

$$C = \frac{A_{V'_0}}{A_{V_0}} = \frac{1}{n_{V_0}}. \tag{6}$$

When the sensor is deployed uniformly in the monitored area and $\cap_{i=1}^{n_{V_0}} V'_{s_i} = \phi$, the overlap area of the expansive region is almost nonexisting and C reaches maximum. Because

$$A_{V'_0} = A_{V_0}, \tag{7}$$

then we have

$$C = \frac{A_{V'_0}}{A_{V_0}} = 1. \tag{8}$$

In order to understand the concept of dispersion, let us give an example. We assumed that $A_{V_0} = 120$. If $n_{V_0} = 4$, then

$$A_{V_{s_i}}{}' = \frac{A_{V_0}}{n_{V_0}} = \frac{120}{4} = 30. \tag{9}$$

If the sensors are deployed as shown in Figure 2, then

$$\begin{aligned} A_{V'_0} &= 120, \\ C &= \frac{A_{V'_0}}{A_{V_0}} = \frac{120}{120} = 1. \end{aligned} \tag{10}$$

If the sensors are deployed as shown in Figure 3, then

$$\begin{aligned} A_{V'_0} &= 120, \\ C &= \frac{A_{V'_0}}{A_{V_0}} = \frac{30}{120} = \frac{1}{4}. \end{aligned} \tag{11}$$

It can be seen that it is a good method to utilize the area of the expansive region to obtain the dispersion degree. C reflects the overlap of the expansive region. In

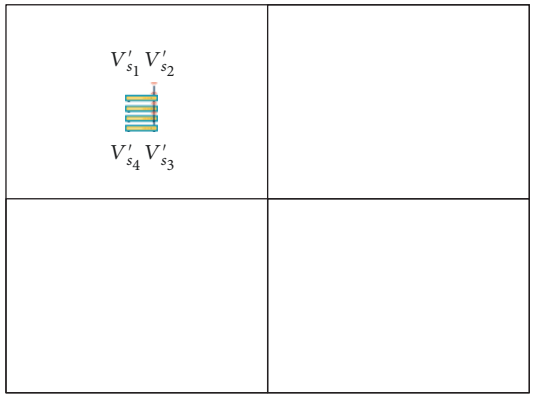

Figure 3: Sensor placement in which C is minimal.

sparse deployment, the higher dispersion degree reflects that the more uniformly the sensor is deployed and each cavity area is relatively smaller. The sparse deployment of sensor for the data reconstruction based on the inverse distance interpolation is transformed into solving the following optimization problem:

$$\begin{aligned} &\arg\max \quad C \\ &\text{s.t.} \quad \sum_{i=1}^{n_{V_0}} A_{s_i} < A_{V_0}. \end{aligned} \tag{12}$$

4. Sensor Placement Based on the Four Subregions

4.1. Sensor Placement Based on the Four Subregions. If $n_{V_0}=1$, then $n_{O_{V_0}}=1$, where $n_{O_{V_0}}$ represents the number of sensors at the center of the monitored region V_0. It is obvious that deploying 1 sensor at the center of the region is the best way. If $n_{V_0}>1$, it is needed that the monitored region is divided into subregions according to the size of the expansion region of the sensor and the sensor is deployed at the center of each subregion. So, the problem of placing the sensors has turned into a problem of how to divide the monitored region into grids. The division of the region corresponds to the distribution of the number of sensors.

The rectangle has the tetragonal property, that is, there are four directions in the space, the sum of rectangle's four angles is 360°, and the four edges can form a closed curve. The monitored region in the actual project usually has the tetragonal property. In this paper, the sensor placement with a schema based on dividing the region into four grids is used to obtain high degree of dispersion.

We make two orthogonal lines whose intersection is the center of the rectangle and make each line perpendicular to the edge of the rectangle. This method makes the rectangle to be divided into 4 small rectangles of the same size, as shown in Figure 4. Obviously, the four little rectangles are similar to the original ones. With the increase in the number of sensors that need to be deployed, we need to subdivide these small rectangles in the same way. We are inspired by the fractal theory. We use the method of dividing the grid into four subregions iteratively to deploy the sensors, because four small rectangles can be combined into a large rectangle and the method is easy to implement. If the number of sensors to be placed in the monitored area is larger than 1, we have to divide the monitored region into 4 grids to place the sensors. If the number of sensors placed in a grid is larger than 1, then we have to divide this grid into 4 subregions in the same way. Iterate in the same way, until the number of sensors allocated in the subregions is 1. The algorithm in this paper takes advantage of the tetragonal property of the rectangle, so that the sensors can be fully dispersed in space.

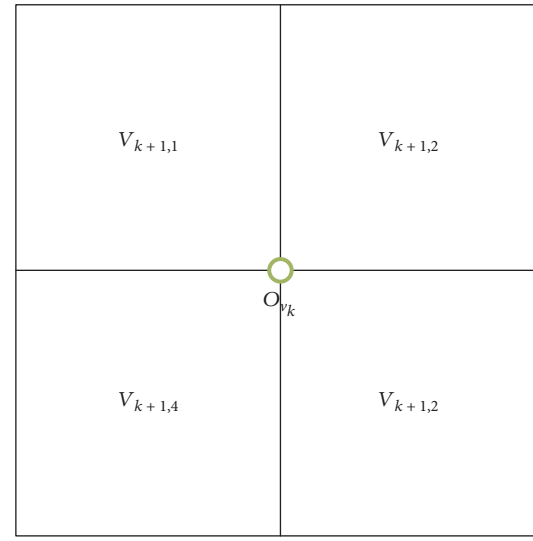

Figure 4: Four subregions of the monitored region.

$$\begin{aligned} &(V_{k+1,1}, V_{k+1,2}, V_{k+1,3}, V_{k+1,4}) = FO(V_k), \\ &V_k = V_{k+1,1} \cup V_{k+1,2} \cup V_{k+1,3} \cup V_{k+1,4}, \end{aligned} \tag{13}$$

where the function $FO(V_k)$ means to divide the grid into four equal subregions. The effect is shown in Figure 4.

In Figure 4, k represents the kth equal division and O_{V_0} is the center of the monitored region V_k. If the total number of sensors allocated to region V_k is 1, the sensor is placed at O_{V_0}. If the total number of sensors allocated to region V_k is larger than 1, we have to divide the region into four subregions and the sensor is placed at the center of each subregion. We make two orthogonal lines whose intersection is O_{V_0} and make each line perpendicular to the edge of region V_k. In this way, V_k is divided into four subregions by the two lines. $V_{k+1,1}$, $V_{k+1,2}$, $V_{k+1,3}$, $V_{k+1,4}$ represent the four subregions of region V_k. V_{k+1} indicates another four division on the basis of V_k. The subscript k reflects the number of times the region is divided into four equal parties.

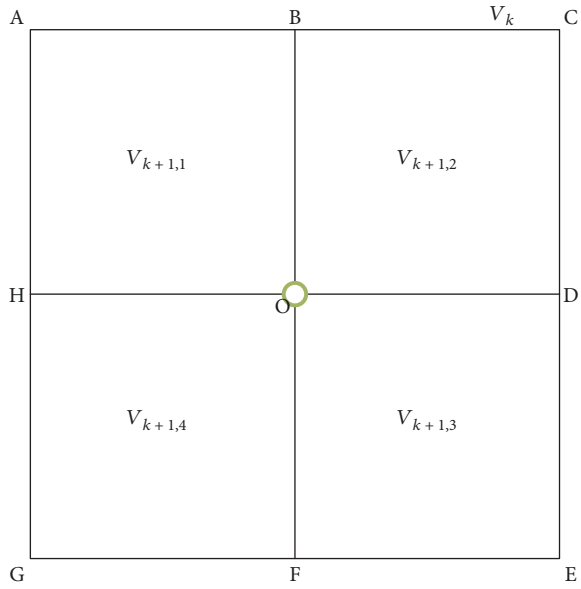

Figure 5: Four subregions for the proof.

$V_{k+1,1} = FO_1(V_k)$ represents the first subregion after dividing region V_k into four subregions; $V_{k+1,2} = FO_2(V_k)$ represents the second subregion after dividing region V_k into four subregions; $V_{k+1,3} = FO_3(V_k)$ represents the third subregion after dividing region V_k into four subregions; $V_{k+1,4} = FO_4(V_k)$ represents the fourth subregion after dividing region V_k into four subregions.

V_k is the parent region of V_{k+1}, and V_{k+1} is the subregion of V_k.

Property 4.1. Subregion V_{k+1} is similar to parent region V_k, i.e., $V_{k+1} \sim V_k$.

Proof. $V_k = \square ACEG$, $V_{k+1,1} = \square ABOH$, $V_{k+1,2} = \square BODC$, $V_{k+1,3} = \square ODEF$, and $V_{k+1,4} = \square HOFG$ are shown in Figure 5. O is the center of $\square ACEG$. $BF \perp HD$, $BF \perp AC$, $BF \perp GE$, $HD \perp AG$, and $HD \perp CE$. According to the definition of vertical intersection line, we have that $\angle BOH = 90°$, $\angle ABO = 90°$, and $\angle AHO = 90°$.

Because O is the center of $\square ACEG$, thus,

$$\frac{AB}{AC} = \frac{AH}{AG} = \frac{1}{2}. \tag{14}$$

Because $\square ABOH \sim \square ACEG$, thus, $V_{k+1,1} \sim V_k$. In the same way, $V_{k+1,2} \sim V_{k+1,3} \sim V_{k+1,4} \sim V_k$; therefore,

$$V_{k+1} \sim V_k. \tag{15}$$

If the number of sensors allocated into the subregion is larger than 1, we have to divide the subregion into 4 smaller subregions. Once the region is divided, the number of sensors assigned to the region is divided by 4. The number of sensors allocated to the subregion is a quarter of the number of sensors in the parent region. The number of sensors is allocated according to the four equal division method until the number of sensors allocated in the subregion is 1. Such iteration makes the division of the monitored region and sensor assignment form a quadtree, as shown in Figure 6.

When the number of sensors is divided by 4, it is possible to have the remainder. It is shown in Figure 6 that the number of redundant sensors is allocated to the first and third subregions which are on a diagonal line to the parent region. Because the distance between the centers of the two subregions on the diagonal line is greater than the distance between the two subregional centers on the same side, we choose to allocate redundant sensors to two diagonal subregions. The problem of redundant sensor number assignment will be explained in detail in Section 4.3.

In Figure 6, k also indicates the depth of the tree. The value of k is determined by n_{V_0} and V_0 is the whole monitored target region. If $n_{V_0} = 1$, then the sensor is placed at the center of region V_0 and $k = 0$. If $1 < n_{V_0} \leq 5$, then $k = 1$; if $5 < n_{V_0} \leq 21$, then $k = 2$. The larger the total number of sensors in the monitored region is, the larger k is. k is accumulating from 0. Once the monitored region is divided and the number of sensors n_{V_k} is divided by 4, k accumulates 1 time. Until n_{V_k} was reduced to 1, k stopped accumulating.

When the area of each subregion is equal and the central location of the parent region and the central location of subregion are placed with sensors, then the partition tree is a full quadtree which is shown in Figure 7.

As can be seen in Figure 7, when the partition tree is a full quadtree, adding more sensors will increase the depth k of the tree. If the partition tree becomes a full quadtree in the given k, the number of divided subregions reaches maximum.

Property 4.2. The relationship between the depth k of a partition tree and the maximum number $n_{\max}$ of sensors in the corresponding full quadtree is

$$n_{\max} = 4^0 + 4^1 + 4^2 \ldots + 4^k. \tag{16}$$

Proof. ① If $k = 0$, the monitored region does not need to be divided and the sensor can be placed at the center of the monitored region. If the total number of sensors is 2, it is necessary to divide the monitored region. So, the maximum number of sensors $n_{\max}$ in a quadtree with $k = 0$ is 1, $n_{\max} = 1 = 4^0$.

② Suppose that when $k = p$, this equation is set up, i.e., $n_{\max} = 4^0 + 4^1 + 4^2 \ldots + 4^p$.

Then when $k = p + 1$, we can reason according to the following steps.

Because layer $p + 1$ is the next layer of layer p in the quadtree, then the maximum number

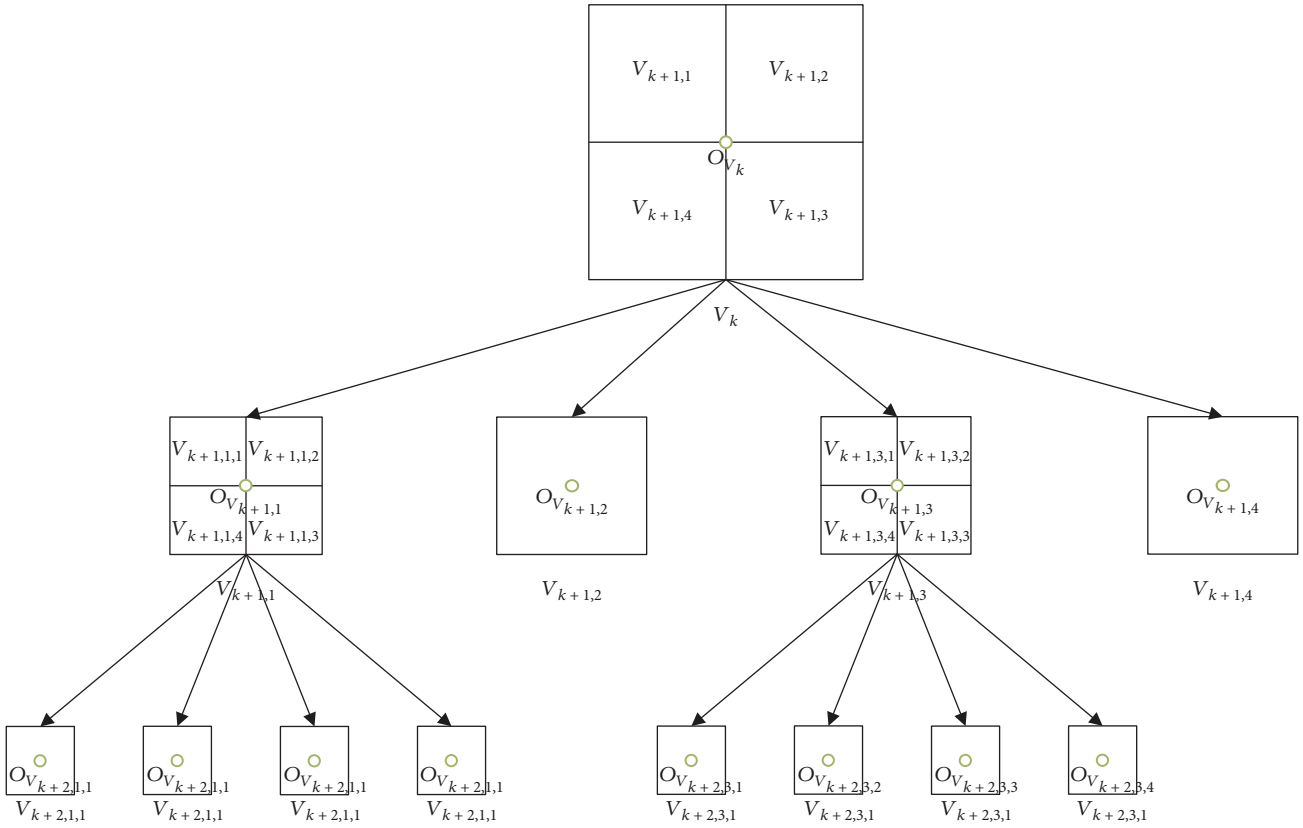

Figure 6: Division tree of subregions for the monitored region.

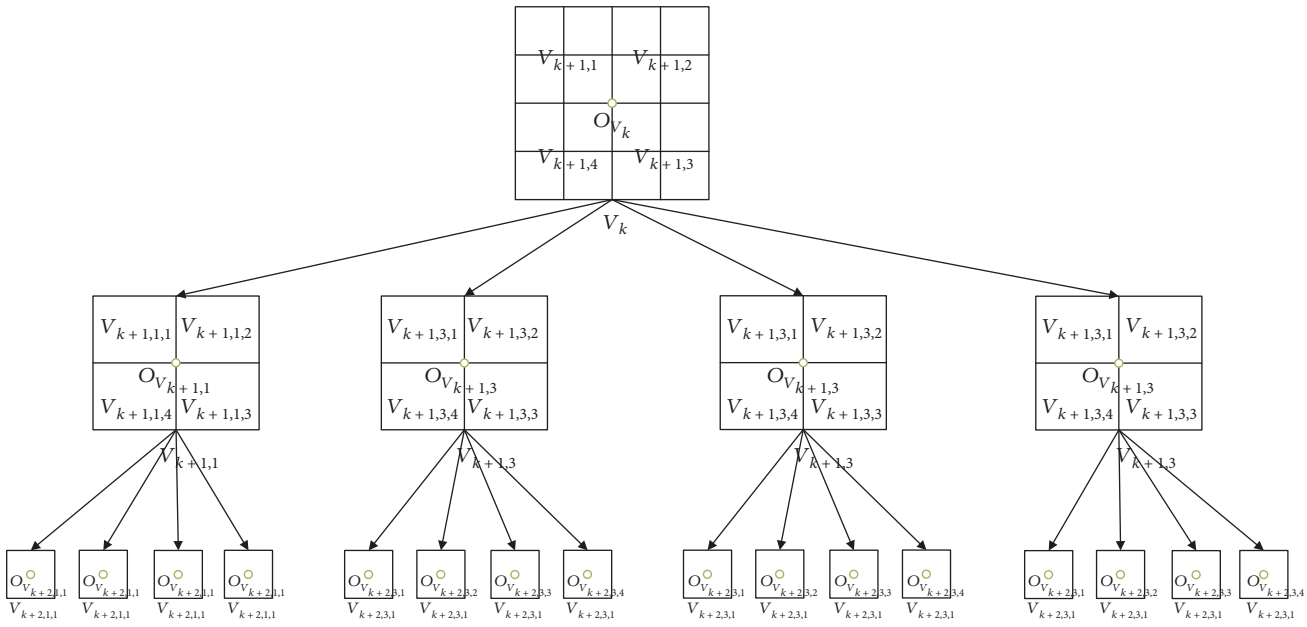

Figure 7: Division of full quadtree for the monitored region.

of sensors in layer $p+1$ in the quadtree is as follows:

$$n_{\max} = 4^0 + 4^1 + 4^2 \ldots + 4^p + 4^p \times 4 = 4^0 + 4^1 + 4^2 \ldots + 4^p + 4^{p+1}. \quad (17)$$

Therefore, when $k = p + 1$, this formula $n_{\max} = 4^0 + 4^1 + 4^2 \ldots + 4^{p+1}$ is established.

From steps ① and ②, we can see that the equation is established for all the natural numbers of k.

$$n_{\max} = 4^0 + 4^1 + 4^2 \ldots + 4^k. \quad (18)$$

Property 4.3. The maximum depth of the partition tree constrained by n_{V_0} is as follows:

$$\lceil \log_4 n_{V_0} \rceil = k. \quad (19)$$

Proof. Set the depth of the partition tree to k and the number of sensors to n_{V_0}.

Then it can be derived by Property 4.2 as follows:

n_{V_0} is larger than the number of sensors of a full quadtree in which the depth is $k-1$ and is less than or equal to the number of sensors of a full quadtree in which the depth is k, that is,

$$4^0 + 4^1 + 4^2 \ldots + 4^{k-1} < n_{V_0} \le 4^0 + 4^1 + 4^2 \ldots + 4^{k-1} + 4^k. \quad (20)$$

Then we have

$$0 < n_{V_0} \le 4^k. \quad (21)$$

The following inequality is obtained by performing the base 4 logarithmic operation on inequality (21):

$$\log n_{V_0} \le k, \quad (22)$$

because $\log n_{V_0}$ is not necessarily an integer but it is less than or equal to k and greater than $k-1$.

We have that k needs to round up $\log n_{V_0}$ and then $\lceil \log_4 n_{V_0} \rceil = k$.

Since the iterative dividing method uses the same method for each iteration, it gives similarity between subregions and entire regions and subregions and subregions. The iterative subregions have self-similarity, that is, the subregions divided at different depths have certain similarities.

4.2. Alternate Assignment of Remainders. The purpose of dividing the monitored area is to distribute the number of sensors and to determine the location of the sensors. The monitored region is divided into 4 subregions; correspondingly, the number of sensors assigned to the monitored region would be divided into four parts.

$$n_{V_k} = n_{V_{k+1,1}} + n_{V_{k+1,2}} + n_{V_{k+1,3}} + n_{V_{k+1,1}} + n_{O_{V_k}}, \quad (23)$$

where n_{V_k} represents the total number of sensors in region V_k. $n_{V_{k+1,1}}, n_{V_{k+1,2}}, n_{V_{k+1,3}}, n_{V_{k+1,1}}$ represent the number of sensors assigned to the four subareas. $n_{O_{V_k}}$ denotes the number of sensors at center O_{V_k} of region V_k. The value of $n_{O_{V_k}}$ is 0 or 1. $n_{O_{V_k}} = 0$ indicates that no sensor is placed at center O_{V_k} of V_k. $n_{O_{V_k}} = 1$ indicates that the sensor is placed at center O_{V_k} of V_k.

When the total number of sensors is divided equally, there will be cases where the total number of sensors cannot be completely divided into four equal parts, i.e., there may be a remainder divided by 4. In order to improve the dispersion of sensor deployment, we adopt the strategy of alternately allocating the remainder to the two diagonal subregions in this paper.

For example, if $k = p$, the remainder when n_{V_k} is divided by 4 is assigned to the first and third subregions.

Then if $k = p + 1$, the remainder when $n_{V_{k+1}}$ is divided by 4 is assigned to the second and fourth subregions.

If $k = p + 2$, the remainder when $n_{V_{k+2}}$ is divided by 4 is assigned to the first and third subregions.

With the increase of k, the remainder allocation is analogous to this method. Alternate allocation of remainder avoids being sensor intensive in the diagonal area.

We set $\text{quot}(n_k, 4)$ to represent the quotient when n_k is divided by 4. $\text{mod}(n_{V_k}, 4)$ represents the remainder when n_k is divided by 4. The value of the remainder is [0, 1, 2, 3]. We discuss the allocation of the remainder from the following four cases:

Case 1. If $\text{mod}(n_{V_k}, 4) = 0$, it means that the total number of sensors in region V_k is just divided into four equal parts. $n_{V_{k+1,1}} = \text{quot}(n_{V_k}, 4)$, $n_{V_{k+1,2}} = \text{quot}(n_{V_k}, 4)$, $n_{V_{k+1,3}} = \text{quot}(n_{V_k}, 4)$, and $n_{V_{k+1,4}} = \text{quot}(n_{V_k}, 4)$.

Case 2. If $\text{mod}(n_{V_k}, 4) = 1$, the remaining 1 sensor is deployed at the center of region V_k, i.e., $n_{O_{V_k}} = 1$.

Case 3. If $\text{mod}(n_{V_k}, 4) = 2$, in order not to destroy the symmetry of the monitored area in the subregions, we allocate the redundant sensors to two diagonal subregions of the quadrilateral. We allocate the redundant sensors to the first and the third subregions, i.e., $n_{V_{k+1,1}} = \text{quot}(n_{V_k}, 4) + 1$ and $n_{V_{k+1,3}} = \text{quot}(n_{V_k}, 4) + 1$, or allocate them to the second and the fourth subregions, i.e., $n_{V_{k+1,2}} = \text{quot}(n_{V_k}, 4) + 1$ and $n_{V_{k+1,4}} = \text{quot}(n_{V_k}, 4) + 1$. The odd-numbered subregions ($n_{V_{k+1,1}}$ and $n_{V_{k+1,3}}$) and the even-numbered subregions ($n_{V_{k+1,2}}$ and $n_{V_{k+1,4}}$) are rotated alternately to obtain the opportunity for accumulation.

Case 4. If $\text{mod}(n_{V_k}, 4) = 3$, one sensor is placed at the center of region V_k, i.e., $n_{O_{V_k}} = 1$, and the other two sensors are assigned using the method of Case 3.

$$
\begin{aligned}
n_{V_{k+1,1}} &= \begin{cases} \text{quot}\left(n_{V_k}, 4\right) + \max\left((-1)^{k+1} \times 1, 0\right), & \text{when mod}\left(n_{V_k}, 4\right) = 2 \text{ or mod}\left(n_{V_k}, 4\right) = 3, \\ \text{quot}\left(n_{V_k}, 4\right), & \text{when mod}\left(n_{V_k}, 4\right) = 0 \text{ or mod}\left(n_{V_k}, 4\right) = 1, \end{cases} \\
n_{V_{k+1,2}} &= \begin{cases} \text{quot}\left(n_{V_k}, 4\right) + \max\left((-1)^{k} \times 1, 0\right), & \text{when mod}\left(n_{V_k}, 4\right) = 2 \text{ or mod}\left(n_{V_k}, 4\right) = 3, \\ \text{quot}\left(n_{V_k}, 4\right), & \text{when mod}\left(n_{V_k}, 4\right) = 0 \text{ or mod}\left(n_{V_k}, 4\right) = 1, \end{cases} \\
n_{V_{k+1,3}} &= \begin{cases} \text{quot}\left(n_{V_k}, 4\right) + \max\left((-1)^{k+1} \times 1, 0\right), & \text{when mod}\left(n_{V_k}, 4\right) = 2 \text{ or mod}\left(n_{V_k}, 4\right) = 3, \\ \text{quot}\left(n_{V_k}, 4\right), & \text{when mod}\left(n_{V_k}, 4\right) = 0 \text{ or mod}\left(n_{V_k}, 4\right) = 1, \end{cases} \\
n_{V_{k+1,4}} &= \begin{cases} \text{quot}\left(n_{V_k}, 4\right) + \max\left((-1)^{k} \times 1, 0\right), & \text{when mod}\left(n_{V_k}, 4\right) = 2 \text{ or mod}\left(n_{V_k}, 4\right) = 3, \\ \text{quot}\left(n_{V_k}, 4\right), & \text{when mod}\left(n_{V_k}, 4\right) = 0 \text{ or mod}\left(n_{V_k}, 4\right) = 1, \end{cases}
\end{aligned} \tag{24}
$$

where the function of max $(\cdot, \cdot)$ is to take the maximum of two parameters.

We set O_{V_k} to represent the center of region V_k and set the sensor to be placed at the center of the region. If

$$\text{mod}\left(n_{V_k}, 4\right) = 1, \tag{25}$$

then

$$n_{O_{V_k}} = 1. \tag{26}$$

Theorem 4.1. *If the total number n of sensors that can be deployed satisfies $n = 4^x, x \in Z, x >= 0$, then uniform distribution can be achieved. Uniform deployment of sensors is a special case of our proposed placement scheme.*

Proof. ① If $x = 1$, then $n = 4$. According to our proposed placement scheme, the sensor deployment is shown in Figure 8.

Sensors are placed at the center of each subregion. It is clear that uniform deployment is achieved.

② Suppose that when $x = p$, this equation is set up. $n = 4^p$, then n can be divisible by 4, and uniform distribution can be achieved.

When $x = p + 1$, we can follow the following steps to reason, because

$$x = p + 1, \tag{27}$$

and then $n = 4^{p+1}$. When dividing the number n into 4 equal parts, the number of sensors assigned to each subregion is

$$\frac{4^{p+1}}{4} = 4^p. \tag{28}$$

When $x = p$, $n = 4^p$, Theorem 4.1 is set up. When $n = 4^p$, uniform distribution can be achieved. The number of sensors is uniformly distributed in each subregion and throughout the entire area.

Therefore, when $x = p + 1$, Theorem 4.1 is set up.

From steps ① and ②, we can see that Theorem 4.1 is set up.

4.3. Fine Adjustment of Sensor Position. When mod $(n_{V_k}, 4) \geq 2$, we assigned redundant sensors to two diagonal subareas in this paper. It makes the density of sensors in the four subregions different. It will be the reason that the dispersion degree of the two subregions that get the redundant sensors is smaller than that of the two subregions which do not acquire them. Since the dispersion degree of sensor placement is related to the positions of the sensors, we can move the sensors in the subregions assigned with the redundant sensors closer to the subregions that are not allocated with the redundant sensors. It allows the sensor placement in the parent regions to be distributed approximately uniformly.

We assume that the coordinates of the center position O_{V_k} of region V_k are $(x_{O_{V_k}}, y_{O_{V_k}})$ and the coordinates of the center position $O_{V_{k+1}}$ of subregion V_{k+1} are $(x_{O_{V_{k+1},i}}, y_{O_{V_{k+1},i}})$, where i is the index of the subregions. The coordinates of the center position of the child region and the parent region have the following relationship:

The coordinates of the center position of subregion $V_{k+1,1}$ are

$$\begin{cases} x_{O_{V_{k+1},1}} = x_{O_{V_k}} - \dfrac{d_{V_k}}{4}, \\ y_{O_{V_{k+1},1}} = y_{O_{V_k}} + \dfrac{d_{V_k}}{4}. \end{cases} \tag{29}$$

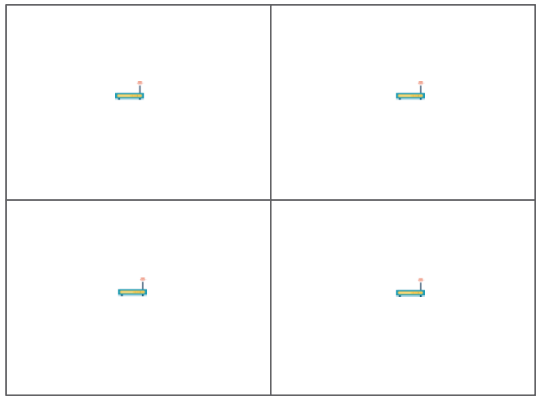

Figure 8: Uniform placement of sensors in the four subregions.

The coordinates of the center position of subregion $V_{k+1,2}$ are

$$\begin{cases} x_{O_{V_{k+1,2}}} = x_{O_{V_k}} + \dfrac{d_{V_k}}{4}, \\ y_{O_{V_{k+1,2}}} = y_{O_{V_k}} + \dfrac{d_{V_k}}{4}. \end{cases} \tag{30}$$

The coordinates of the center position of subregion $V_{k+1,3}$ are

$$\begin{cases} x_{O_{V_{k+1,3}}} = x_{O_{V_k}} - \dfrac{d_{V_k}}{4}, \\ y_{O_{V_{k+1,3}}} = y_{O_{V_k}} - \dfrac{d_{V_k}}{4}. \end{cases} \tag{31}$$

The coordinates of the center position of subregion $V_{k+1,4}$ are

$$\begin{cases} x_{O_{V_{k+1,4}}} = x_{O_{V_k}} + \dfrac{d_{V_k}}{4}, \\ y_{O_{V_{k+1,4}}} = y_{O_{V_k}} - \dfrac{d_{V_k}}{4}. \end{cases} \tag{32}$$

When $k = 0$, the coordinates of the center position O_{V_0} of region V_0 do not need to be updated. We set that d_{V_k} is the diameter of region V_k. We assume that the adjustment distance of the subregion is set to Δd.

$$\begin{aligned} 0 < \|\Delta d\| < \frac{d_{V_{k+1}}}{2}, \\ \|\Delta d\| = \frac{d_{V_{k+1}}}{4}. \end{aligned} \tag{33}$$

In this paper, we update the coordinates as follows:

Case 1. Sensors in the horizontal regions are adjusted close to each other.

The sensor in subregion $V_{k+1,1}$ is adjusted close to subregion $V_{k+1,2}$:

$$\begin{cases} x'_{O_{V_{k+1,1}}} = x_{O_{V_{k+1,1}}} + \|\Delta d\|, \\ y'_{O_{V_{k+1,1}}} = y_{O_{V_{k+1,1}}}, \\ \text{when } \max\left((-1)^{k+1} \times 1, 0\right) = 1, \\ \left(\text{mod}\left(n_{V_k}, 4\right) = 2 \text{ or } \text{mod}\left(n_{V_k}, 4\right) = 3\right). \end{cases} \tag{34}$$

The sensor in subregion $V_{k+1,3}$ is adjusted close to subregion $V_{k+1,4}$:

$$\begin{cases} x'_{O_{V_{k+1,3}}} = x_{O_{V_{k,3}}} - \|\Delta d\|, \\ y'_{O_{V_{k+1,3}}} = y_{O_{V3}}, \\ \text{when } \max\left((-1)^{k+1} \times 1, 0\right) = 1, \\ \text{mod}\left(n_{V_k}, 4\right) = 2 \text{ or } \text{mod}\left(n_{V_k}, 4\right) = 3. \end{cases} \tag{35}$$

Case 2. Sensors in the vertical regions move closer to each other.

The sensor in subregion $V_{k+1,2}$ is adjusted close to subregion $V_{k+1,4}$:

$$\begin{cases} x'_{O_{V_{k,2}}} = x_{O_{V_{k,2}}}, \\ y'_{O_{V_{k,2}}} = y_{O_{V_{k,2}}} + \|\Delta d\|, \\ \text{when } \max\left((-1)^{k+1} \times 1, 0\right) = 0, \\ \left(\text{mod}\left(n_{V_k}, 4\right) = 2 \text{ or } \text{mod}\left(n_{V_k}, 4\right) = 3\right). \end{cases} \tag{36}$$

The sensor in subregion $V_{k+1,4}$ is adjusted close to subregion $V_{k+1,1}$:

$$\begin{cases} x'_{O_{V_{k,4}}} = x_{O_{V_{k,4}}}, \\ y'_{O_{V_{k,4}}} = y_{O_{V_{k,4}}} - \|\Delta d\|, \\ \text{when } \max\left((-1)^{k+1} \times 1, 0\right) = 0, \\ \left(\text{mod}\left(n_{V_k}, 4\right) = 2 \text{ or } \text{mod}\left(n_{V_k}, 4\right) = 3\right). \end{cases} \tag{37}$$

4.4. Algorithm for the Sensor Placement Based on the Four Subregions. We define the following notation before we design the algorithm:

n represents the number of sensors that can be deployed. The coordinates of the central position of the monitored region are (x, y). The diameter of the monitored area is d. The vector of the location of the sensor deployment is $D = \{(x_i, y_i) \mid i = 1, 2, \cdots, n\}$. k is the depth of the division tree. The initial value of k is 0. p is the quotient when n is divided by 4. q is the remainder when n is divided by 4.

```
Sparsedeploy4 (int n, float x, float y, float d)
{intp, q;
p = n/4;
q = n \ 5;
k = k + 1;
Switch q
Case 0:
        {
        Sparsedeploy4(p, x − d/4, y + (d/4), d/2);
        Sparsedeploy4(p, x + (d/4), y + (d/4), d/2);
        Sparsedeploy4(p, x + (d/4), y − d/4, d/2);
        Sparsedeploy4(p, x − d/4, y − d/4, d/2);
        }
Case 1:
        {
        (x, y) → D;
        Return;
        }
Case 2:
        {
          If k == 1 then
            {
            Sparsedeploy4(p + max ((−1)^(k+1) × 1, 0), x − (d/4) + ||Δd||, y + (d/4), d/2);
            Sparsedeploy4(p + max ((−1)^k × 1, 0), x + (d/4), y + (d/4), d/2);
            Sparsedeploy4(p + max ((−1)^(k+1) × 1, 0), x + (d/4) − ||Δd||, y − d/4, d/2);
            Sparsedeploy4(p + max ((−1)^k × 1, 0), x − d/4, y − d/4, d/2);
            }
          Else
            {
            Sparsedeploy4(p + max ((−1)^(k+1) × 1, 0), x − d/4, y + (d/4), d/2);
            Sparsedeploy4(p + max ((−1)^k × 1, 0), x + (d/4), y + (d/4) − ||Δd||, d/2);
            Sparsedeploy4(p + max ((−1)^(k+1) × 1, 0), x + (d/4), y − d/4, d/2);
            Sparsedeploy4(p + max ((−1)^k × 1, 0), x − d/4, y − (d/4) + ||Δd||, d/2);
            }
          Endif
        }
Case 3:
        {(x, y) → D;
          Goto case 2;
        }
  }
```

Algorithm 1: The sensor placement based on the four subregions.

4.5. Fine Adjustment for Making Up for the Voids. In the actual deployment project, there will be some scenarios where subregions can not deploy sensors. This will affect the dispersion degree of sensors in the whole monitored region and further affect the precision of data reconstruction. It is necessary to perform fine adjustment so that the impact is as small as possible. We set the coordinates of center O_{V_Π} of subregion V_Π where sensors cannot be deployed, which is $(x_{O_{V_\Pi}}, y_{O_{V_\Pi}})$, as shown in Figure 9.

Although the sensor cannot be deployed in subregion V_Π, we can reconstruct the physical quantity of monitoring in this region by interpolation. In order to get high precision of reconstruction, the positions of sensors around it are fine adjusted. They were adjusted to get closer to it. In the section, we assume that the adjustment distance of subregion is set to $\Delta d'$ (Algorithm 2).

5. Experiments

In this section, we outline the tests used to investigate the performance of the proposed algorithm on MATLAB. Evaluation indexes include the dispersion degree of sensor placement, the precision of data reconstruction, and the coverage rate.

5.1. Verification of Reconstruction Precision. It is necessary to verify the impact of the deployment mechanism on the precision of data reconstruction. The reconstruction algorithm

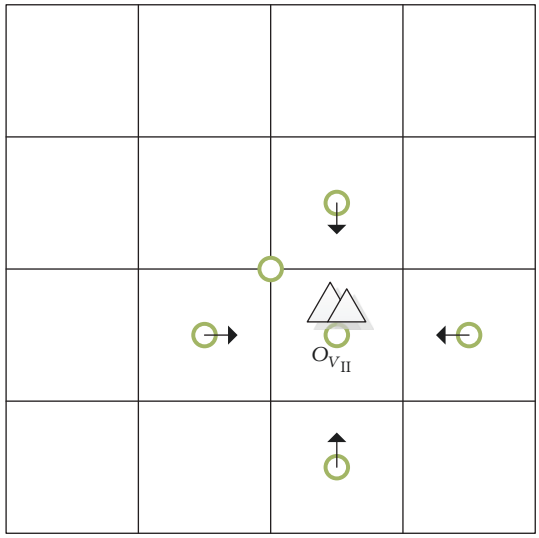

Figure 9: Subregion where sensors cannot be deployed.

used in this paper is an inverse distance-weighted (IDW) interpolation reconstruction algorithm, which is proposed by Shepard. Subsequently, IDW algorithm is widely applied in spatial analysis of various fields. Inverse distance-weighted interpolation is also known as "inverse distance-weighted average" or "Shepard method." Suppose that there are n known locations whose plane coordinates are (x_i, y_i) and the known data is z_i, where $i = 1, 2, 3, \ldots, n$, i.e., $\{z_{i(x_i,y_i)} \mid i = 123 \cdots n\}$. The interpolation function for the inverse distance-weighted interpolation is

$$f(x,y) = \begin{cases} \dfrac{\sum_{i=1}^{n} \left(z_i / d_i^p\right)}{\sum_{i=1}^{n} \left(1/d_i^p\right)} & \text{if } (x,y) \neq (x_i, y_i),\ i = 1, 2, \ldots, n, \\ z_i & \text{if } (x,y) = (x_i, y_i),\ i = 1, 2, \ldots, n, \end{cases} \tag{38}$$

where $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the distance between point (x, y) and point (x_i, y_i), $i = 1, 2, \ldots, n$. p is a constant greater than zero, called a weighted exponent. In this experiment, we set the value of p to 1.

The data set that we use is the measured data provided by the Intel Berkeley Research lab [30]. We select the data of each sensor at the same time in this experiment. We select this moment with a timing of 33. The sensor set, whose sensor index numbers are {1,3,7,9,14,15,18,19,21,22,24,26,28,29,30, 31,36,38,39,41,43,46,47,48,50,51,54}, provides the acquisition data at time 33. We test it by means of reconstructing the temperature value. The length of this monitored region is $kh = 41$ and the width is $kv = 32$. The x and y coordinates of the sensors are relative to the upper-right corner of the lab in meters. However, the x' and y' coordinates of the sensors in our algorithm are relative to the lower-left corner of the lab. It is necessary to transform the coordinates of the sensors provided by the Intel Berkeley Research lab to the coordinates in the coordinate system of our algorithm. We use formula (39) to transform them:

$$\begin{aligned} x' &= -x + kh, \\ y' &= -y + kv. \end{aligned} \tag{39}$$

The coordinates of the central location of the whole region in our coordinate system are (20.5, 16).

In order to compare the precision of the reconstruction, we choose the mean relative error (MRE). It reflects the precision of the estimated data relative to the measured data. The formula for the calculation of MRE is as follows [13]:

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{z(x_i) - \widehat{z}(x_i)}{z(x_i)} \right|, \tag{40}$$

where $z(x_i)$ is the actual acquisition value of the ith sensor. Correspondingly, $\widehat{z}(x_i)$ is the reconstructed value. n is the total number of sensors.

We first evaluate the impact of the dispersion degree of sensor placement. The definition of the dispersion degree is in Section 3. For the cases that the dispersion degree of sensor placement is not considered, we compare the placement of sensors concentrated at the bottom of the region and the placement of sensors concentrated on the right of the region with our placement. Because the placement proposed in this paper is designed based on the dispersion degree of sensor placement, they can be compared.

We take 10 sensor nodes in the bottom region. Their index numbers are {24, 26, 28, 29, 30, 31, 36, 48, 39, 41}, which are the subset of the sensor set with a timing of 33. We only deploy sensors at the location of some sensors out of these 10 sensors to reconstruct temperature values at other sensor locations. Ten tests were conducted for the comparison. We calculate the average MRE corresponding to the different total sensors number of the 10 tests. We compare the average MRE of the proposed placement and the placement without consideration of dispersion, as shown in Figure 10.

Then we take 10 sensor nodes in the right side of the region again. Their index numbers are {14, 15, 18, 19, 21, 22, 24, 26, 28, 29}. We did ten tests. The comparison result is shown in Figure 11.

From Figures 10 and 11, we can see that the dispersion degree of sensors does affect the precision of data reconstruction. It is obvious that the more sensors are deployed, the smaller the reconstruction error is. Our placement is based on the dispersion degree of the sensor deployment. The MRE of our placement is lower than that of the placement without considering the dispersion degree. From the results of many experiments, we can see that in the process of reconstructing data using interpolation, the dispersion degree affects the precision of reconstruction. The physical quantity in the space has spatial correlation; the reconstructed data is based on data from all directions, which is more comprehensive, so the reconstruction precision will be higher.

In terms of reconstruction accuracy, our algorithm has made some improvements based on other algorithms. Ten

For all (x, y) in D
If $x > x_{O_{V_{\Pi}}} - d/2^{\lceil \log_4(n-1) \rceil}$ then $x = x + \Delta d'$ end if
If $x < x_{O_{V_{\Pi}}} + (d/2^{\lceil \log_4(n-1) \rceil})$ then $x = x - \Delta d'$ end if
If $y > y_{O_{V_{\Pi}}} - d/2^{\lceil \log_4(n-1) \rceil}$ then $y = y + \Delta d'$ end if
If $y < y_{O_{V_{\Pi}}} + (d/2^{\lceil \log_4(n-1) \rceil})$ then $y = y - \Delta d'$ end if
End for

Algorithm 2: The fine adjustment algorithm for making up for the voids.

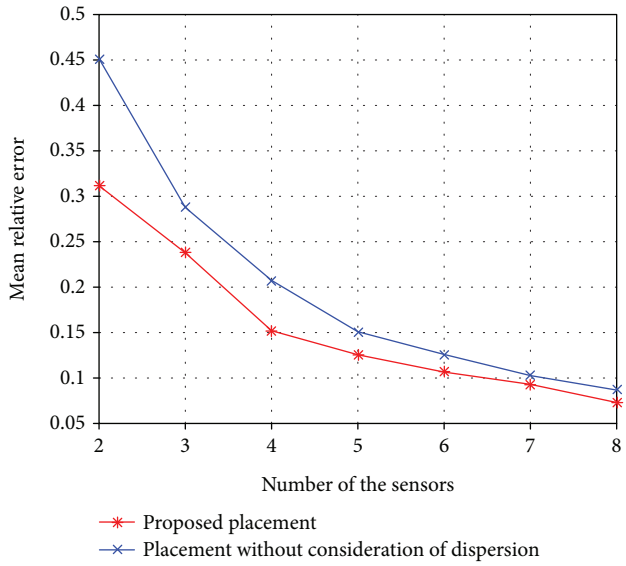

Figure 10: MRE comparison of the placement where the sensors are concentrated at the bottom and the proposed placement where the sensors are dispersed throughout the region.

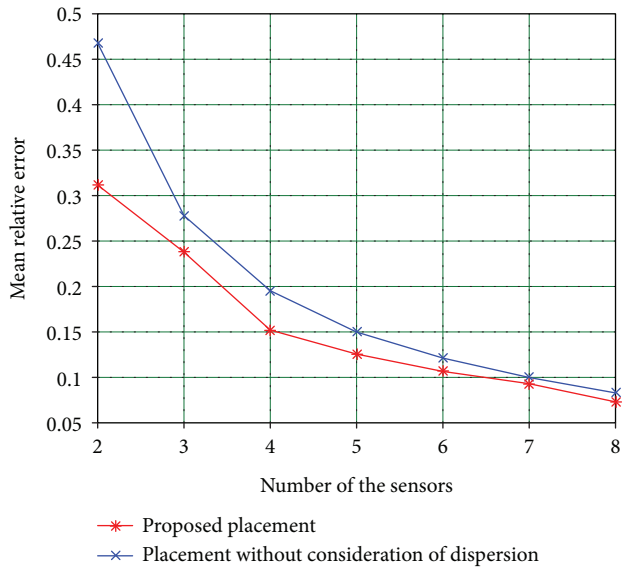

Figure 11: MRE comparison of the placements where the sensors are concentrated on the right of the region and the proposed placement where the sensors are dispersed throughout the region.

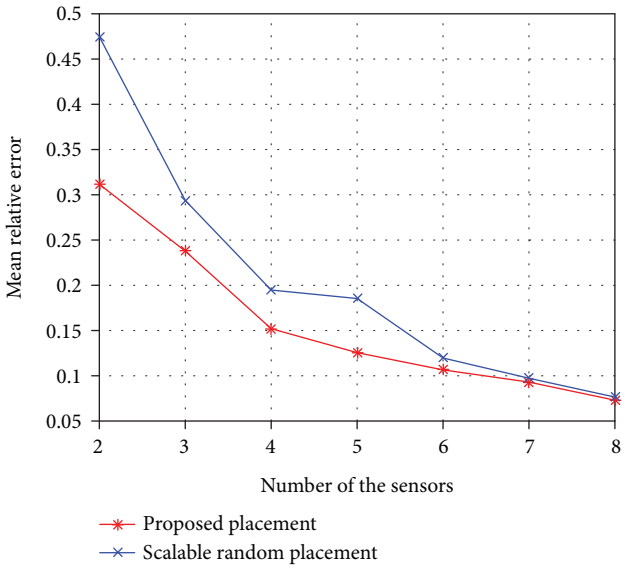

Figure 12: Result of the comparison of the proposed placement and scalable random placement.

tests were conducted for the comparison. We calculate the average MRE corresponding to the different total sensor numbers of the 10 tests. The proposed method is compared with the scalable random placement [16]. The result is shown in Figure 12.

After several experiments, we found that when the number of sensors is small, the gap in the mean relative error between the two algorithms is very large. The MRE of our placement algorithm is much smaller than that of the scalable random placement algorithm. This is because our placement algorithm takes into account the dispersion degree during deployment. The sensors that we deployed have an impact on the physical quantity of the location where the data needs to be reconstructed in all directions, resulting in high recovery accuracy. Our algorithm has obvious advantages in sparse deployment. As the number of sensors that can be deployed increases, the sensors assigned to each direction are available and the MRE of scalable random deployment is also decreasing but the magnitude of reduction is not much larger than that of our placement algorithm. And, we can see from Figure 12 that my algorithm is showing a steady decline.

5.2. Verification of Dispersion and Coverage. We set that the coordinates of the central position of the monitored area are (500 m, 500 m). The diameter of the monitored region

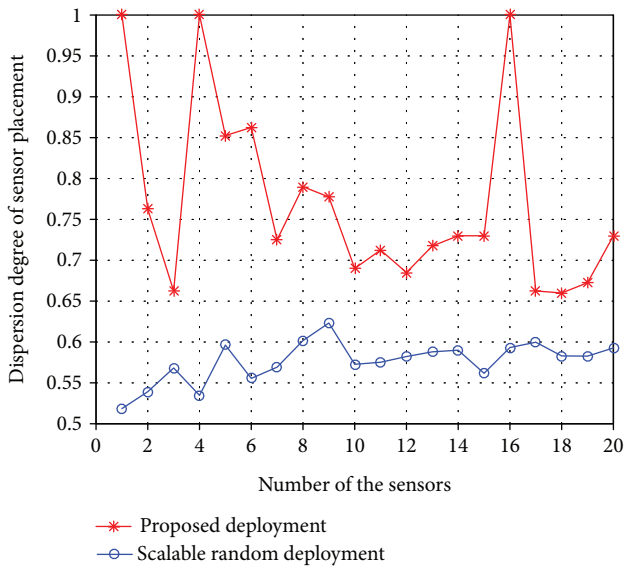

Figure 13: Dispersion of the two different placements varies with the number of sensors.

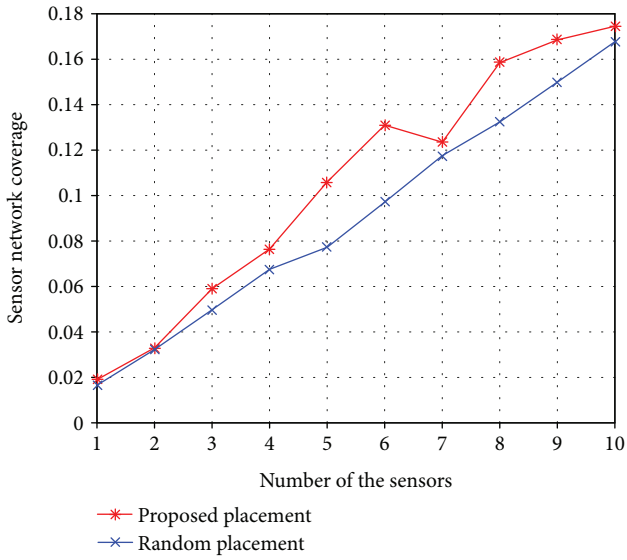

Figure 14: Comparison of coverage rates between proposed placement and random placement with the number of sensors.

is 400 m. The diameter of the monitoring range of the sensor is 40 m. The communication radius of the sensor nodes is assumed to be 40 m. The dispersion degree of sensor placement can be used as a measure of the quality of the deployment.

Ten tests were conducted for the comparison. We calculate the average dispersion degree corresponding to the different total sensor numbers of the 10 tests. The dispersion degree of sensor placement varies with the number of sensors, as shown in Figure 13. It is obvious that the dispersion of the algorithm is higher than that of the scalable random algorithm with different total numbers of sensors that can be deployed. When the total number of sensors is 4^x, where $x \in \mathbb{Z}, x \geq 0$, the sensor can be uniformly distributed and the dispersion degree is 1. From Figure 10, we can see that the dispersion degree of the sensor placement in our algorithm is 1, if the total numbers of sensors are 1, 4, and 16, i.e., 4^x when $x = 0$, 1, and 2.

We introduce sensor coverage efficiency (CE) [31] to measure the utilization rate of sensor coverage. CE is defined as the ratio of the area of the effective coverage union of all sensors in the region to the area of the monitored region, as shown in

$$CE = \frac{\cup_{i=1}^{n} A_i}{A}, \tag{41}$$

where A_i is the coverage area of the ith sensor.

In this experiment we, define the sensor sensing radius as 3. Ten tests were conducted for the comparison. We calculate the average CE corresponding to the different total sensor numbers of the 10 tests. The result is shown in Figure 14. With the increase of the number of sensors, the sensor coverage rate is on the rise. In the case that the number of sensors is less than 4, the effective coverage of our algorithm and that of random deployment are not much different. As the number of sensors increases, the concentration of sensors in some local regions may increase in random deployment but our algorithm ensures that there are no regions where the sensors are placed densely. As can be seen from Figure 14, with the increase in the number of sensors, in most cases, our algorithm coverage is higher than the coverage of the random deployment algorithm. From Figure 14, we can see that with the increase of sensor number, in most cases, the coverage rate of our algorithm is higher than that of random deployment algorithm. In the first test, it has 9 cases that the coverage rate of our algorithm is higher than that of random placement, accounting for 90%.

6. Conclusions

In large area information acquisition, it is desirable to collect information by means of sparse sensor coverage to acquire data information. In this paper, we proposed a sensor deployment algorithm based on four subregions to improve the precision of data reconstruction by IDW. The placement was implemented with a realistic data set of the Intel Berkeley Research lab for testing the reconstruction precision and simulated for testing of the dispersion and coverage of the sensor network. We compared our placement with the scalable random placement. In comparison, we found that the MRE of our sensor placement is lower than that of the random placement, while the number of sensors is not very large. Another advantage of our placement lies in its coverage, as on the tested maps, its performance is better than the random placement. The final advantage of the algorithm is related to its distributed nature. Our placement was inspired by the fractal theory, and it possesses good scalability.

Data Availability

The measured data that support the findings of this study are supplied by the Intel Berkeley Research lab. It can be available from http://db.csail.mit.edu/labdata/labdata.html.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was financially supported by a project of the Natural Science Foundation of Hainan Province in China (Grant no. 617033), the Innovative Research Projects for the Graduate Students of Hainan Higher Education Institutions (Grant no. Hyb2017-06), the National Natural Science Foundation of China (Grant no. 61561017 and Grant no. 61462022), Hainan Province Major Science and Technology Project (Grant no. ZDKJ2016015), Open Project of the State Key Laboratory of Marine Resource Utilization in South China Sea (Grant no. 2016013B), and the Key R&D Project of Hainan Province (no. ZDYF2018015).

References

[1] J. Luo, *Research on Key Technologies of Marine Monitoring Sensor Networks*, Ocean University of China, Shandong, China, 2010.

[2] M. Nian and X. Zhao, "Deployment of wireless sensor networks for farmland surface humidity monitoring in Xinjiang," *Computer Systems & Applications*, vol. 20, no. 7, pp. 42–46, 2011.

[3] B.-J. Chang, J.-B. Peng, and Y.-H. Liang, "Minimizing transmission delay and deployment cost for sensors placement in sparse wireless sensor networks," in *Proceedings of 2007 IEEE Wireless Communications and NETWORKING Conference*, pp. 2757–2761, Hong Kong, China, March 2007.

[4] J. Liu, L. L. Cheng, T. Wang, and J. H. Wang, "Nodes deployment optimization algorithm for reliable sense in sensor networks," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 45, no. 6, pp. 1941–1945, 2015.

[5] Z. Chen, W. Xu, and H. Chen, "Distributed sensor layout optimization for target detection with data fusion," in *Proceedings of the 11th ACM International Conference on Underwater Networks & Systems - WUWNet '16*, p. 50, Shanghai, China, October 2016.

[6] J. Li, L. Andrew, C. Foh, M. Zukerman, and H. H. Chen, "Connectivity, coverage and placement in wireless sensor networks," *Sensors*, vol. 9, no. 10, pp. 7664–7693, 2009.

[7] H. Xu, J. Zhu, and B. Wang, "On the deployment of a connected sensor network for confident information coverage," *Sensors*, vol. 15, no. 5, pp. 11277–11294, 2015.

[8] J.-P. Argaud, B. Bouriquet, F. de Caso, H. Gong, Y. Maday, and O. Mula, "Sensor placement in nuclear reactors based on the generalized empirical interpolation method," *Journal of Computational Physics*, vol. 363, pp. 354–370, 2018.

[9] S. Henna, "Energy efficient fault tolerant coverage in wireless sensor networks," *Journal of Sensors*, vol. 2017, Article ID 7090782, 11 pages, 2017.

[10] Z. Sun, C. Li, X. Xing, and H. Wang, "A new energy-efficient coverage control with multinodes redundancy verification in wireless sensor networks," *Journal of Sensors*, vol. 2016, Article ID 2347267, 11 pages, 2016.

[11] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction," 2017, arXiv.

[12] Q. Zhu, W. Zhang, and J. Yu, "The spatial interpolations in GIS," *Journal of Jiangxi Normal University (Natural Science)*, vol. 28, no. 2, pp. 183–188, 2004.

[13] S. Jang, L. Ren, B. Yong, and Y. He, "Comparison of spatial interpolation methods for the precipitation in Laoha River basin," *Journal of Arid Land Resources and Environment*, vol. 24, no. 1, pp. 80–84, 2010.

[14] X. Huang, J. Ma, and T. Qin, *Introduction to Geographic Information System*, Higher Education Press, Beijing, China, 2001.

[15] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse sensor placement optimization for classification," *SIAM Journal on Applied Mathematics*, vol. 76, no. 5, pp. 2099–2122, 2016.

[16] X. Wu, Y. Wu, M. Liu, and L. Zheng, "In-situ soil moisture sensing: efficient random sensor placement and field estimation using compressive sensing," in *Proceeding of 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pp. 1–6, Wuhan, China, September 2011.

[17] X. Chang, R. Tan, G. Xing et al., "Sensor placement algorithms for fusion-based surveillance networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1407–1414, 2011.

[18] D. Moreno-Salinas, A. Pascoal, and J. Aranda, "Optimal sensor placement for multiple target positioning with range-only measurements in two-dimensional scenarios," *Sensors*, vol. 13, no. 8, pp. 10674–10710, 2013.

[19] V. Akbarzadeh, J. C. Lévesque, C. Gagné, and M. Parizeau, "Efficient sensor placement optimization using gradient descent and probabilistic coverage," *Sensors*, vol. 14, no. 8, pp. 15525–15552, 2014.

[20] K. Zhu, C. Gu, J. Qiu, W. Liu, C. Fang, and B. Li, "Determining the optimal placement of sensors on a concrete arch dam using a quantum genetic algorithm," *Journal of Sensors*, vol. 2016, Article ID 2567305, 10 pages, 2016.

[21] J. Liu, L. Cheng, T. Wang, and J. Wang, "Sparse deployment scheme in mobile sensor networks with prioritized event area," *International Journal of Communication Systems*, vol. 29, no. 4, pp. 760–771, 2016.

[22] M. Rebai, M. Le berre, H. Snoussi, F. Hnaien, and L. Khoukhi, "Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks," *Computers & Operations Research*, vol. 59, pp. 11–21, 2015.

[23] R. Mukherjee, S. Mondal, and S. O. Memik, "Thermal sensor allocation and placement for reconfigurable systems," in *Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, pp. 437–442, San Jose, CA, USA, November 2006.

[24] D. Yoganathan, S. Kondepudi, B. Kalluri, and S. Manthapuri, "Optimal sensor placement strategy for office buildings using clustering algorithms," *Energy and Buildings*, vol. 158, pp. 1206–1225, 2018.

[25] W. Du, Z. Li, J. C. Liando, and M. Li, "From rateless to distanceless: enabling sparse sensor network deployment in large areas," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2498–2511, 2016.

[26] B. Xie, D. Fang, T. Xing et al., "FISCP: fine-grained device-free positioning system for multiple targets working in sparse deployments," *Wireless Networks*, vol. 22, no. 5, pp. 1751–1766, 2016.

[27] D. B. Steffelbauer and D. Fuchs-Hanusch, "Efficient sensor placement for leak localization considering uncertainties," *Water Resources Management*, vol. 30, no. 14, pp. 5517–5533, 2016.

[28] N. Bertola, M. Papadopoulou, D. Vernay, and I. Smith, "Optimal multi-type sensor placement for structural identification by static-load testing," *Sensors*, vol. 17, no. 12, article 2904, 2017.

[29] Z. Xu, J. Guan, and J. Zhou, "A distributed inverse distance weighted interpolation algorithm based on the cloud computing platform of Hadoop and its implementation," in *Proceeding of 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2412–2416, Zhangjiajie, China, August 2015.

[30] http://db.csail.mit.edu/labdata/labdata.html.

[31] J. Wang, X. Ni, and Y. Cheng, "Underwater sensor deployment based on grid division and virtual forces," *Computer Technology and Its Applications*, vol. 42, no. 2, pp. 102–105, 2016.