Research Article

3D Self-Deployment Algorithm in Mobile Wireless Sensor Networks

Chunyu Miao,1,2 Guoyong Dai,1 Xiaomin Zhao,1 Zhongze Tang,1 and Qingzhang Chen1

1College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China
2College of Xingzhi, Zhejiang Normal University, Jinhua 321004, China

Correspondence should be addressed to Qingzhang Chen; qzchen@zjut.edu.cn

Received 25 April 2014; Revised 27 September 2014; Accepted 7 October 2014

Academic Editor: Zhangbing Zhou

Copyright © 2015 Chunyu Miao 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.

The sensor deployment problem of wireless sensor networks (WSNs) is a key issue in the researches and the applications of WSNs. Fewer works focus on the 3D autonomous deployment. Aimed at the problem of sensor deployment in three-dimensional spaces, the 3D self-deployment (3DSD) algorithm in mobile sensor networks is proposed. A 3D virtual force model is utilized in the 3DSD method. A negotiation tactic is introduced to ensure network connectivity, and a density control strategy is used to balance the node distribution. The proposed algorithm can fulfill the nodes autonomous deployment in 3D space with obstacles. Simulation results indicate that the deployment process of 3DSD is relatively rapid and the nodes are well distributed. Furthermore, the coverage ratio of 3DSD approximates the theoretical maximum value.

1. Introduction

The proliferation of wireless and mobile devices has fostered the applications of wireless sensor networks (WSNs) per- vaded from military to industry, agriculture, and other sce- narios [1]. The node deployment is a key issue in WSNs, since it seriously influences the feasibility and Quality of Service (QoS) for WSNs. Generally, the node deployment methods can be classified into static deployment scheme and dynamic deployment scheme [2]. And the static deployment schemes are mainly applied in WSNs with nonmobility nodes; it concludes determinate deployment methods and random deployment methods. In the random deployment scenario, sensor nodes are scattered via airplane or other aided measures into the Area of Interest (AOI) to which human being cannot get conveniently. Random static deployment methods ensure the supervision performance via sensor node redund- dancy. While the dynamic deployment methods are mainly used in a mobile WSNs (i.e., the sensor node has mobility) [3], after being scattered in AOI, the sensor nodes accomplish deployment autonomously. The autonomous deployment of mobile WSNs is a process in which the mobile sensor nodes adjust their positions dynamically according to a certain algo- rithm, until the predefined coverage requirement is achieved. By adopted dynamic deployment method, the coverage per- formance is improved, while the overhead of hardware is decreased. However, the self-deployment scheme designing in three-dimension scenario is difficult; aside from this, the obstacle and area boundary may disturb the deployment process. Aimed at solving these problems aforementioned, we proposed a 3D self-deployment (3DSD) algorithm to fulfill the nodes autonomous deployment in 3D area with obstacles. Virtual force model is introduced in 3DSD method. The network connectivity and nodes distribution density are also took into account to provide the feasibility of 3DSD. Extensive simulations are conducted, and experiment results indicate that the deployment duration of 3DSD is comparatively short, and the coverage ratio of 3DSD approximates the theoretical maximum value. The main contributions of our work can be summarized as follows: (1) the virtual force model is extended to three-dimensional space, and the optimal coverage ratio of 3D deployment is defined and calculated, (2) a density control tactic is explored, node density of each cluster is under control to ensure the node distribution of clusters is balance, (3) the network connectivity is concerned, and a negotiation protocol is designed to provide the practicability of 3DSD.

The rest of this paper is organized as follows. A brief introduction to the state of art of the sensor node deployment
methods is presented in Section 2. The three-dimensional sensing model and the virtual force model are defined in Section 3. In Section 4, the 3DSD method is described in detail. Performance analysis of the proposed algorithm is provided in Section 5, and conclusions are drawn in Section 6.

2. The State of the Art

For the node deployment in WSN, there are certain coverage performance criteria required to be satisfied. According to the application scenario of WSN, deployment coverage category can be classified into blanket coverage, barrier coverage, and scan coverage [4]. Our work belongs to the blanket coverage.

2.1. Methodology of Deployment. The node deployment is an enabling issue for WSN, so a large number of deployment methods are presented by scholars [5]. There are mainly three categories of research methods in node deployment: incremental deployment method [6], geometric analysis method [7], and virtue force based method [8].

The incremental deployment method is appropriate for the scenario that the prior information of the monitored area is unknown, but the deployment duration is too long. Geometric analysis method cannot guarantee network connectivity, and it is hard to be applied to some scenario with obstacles. Zou and Chakrabarty [8] first proposed deployment method using virtual force model (VFM) in WSN. In VFM, sensor nodes are supposed to be subjected to some kinds of forces which come from area border, other nodes, and so forth. The force balance is regarded as the final state of network. Due to the intuition and the descriptive ability of VFM, it is widely used in deployment issue of WSN [9]. There certainly are some hybrid methods used in node deployment [10].

2.2. Static Deployment and Dynamic Deployment. The node deployment issue can also be classified into static deployment and dynamic deployment according to whether nodes have moving ability. In dynamic blanket deployment, one subissue is how to mend deployment holes formed in static deployment process using mobile nodes [11]. Another subissue is self-deployment of some mobile enabling nodes after they are launched in AOI [12]. Our work focuses on the latter. A geometric analysis method is proposed in [13]; however, there is still the problem that it is hard to be employed in scenarios with obstacles. Zhang and Fei present a deployment scheme to enhance the coverage while keeping the network connected at each step of the deployment [14]. However, the obstacle is not taken into consideration. In [15], an autonomous deployment algorithm based on VFM is adopted in 2D scenario. But, for the VFM model with obstacle forces, some certain strategies should be adopted to avoid algorithm trapping into local optimum.

2.3. 3D Surface Deployment and 3D Space Deployment. According to the dimensionality of AOI, the node deployment method can be divided into two classes: deployment in 2D space and deployment in 3D space. In 3D space deployment, there are two subissues, one is 3D area coverage [16] (e.g., underwater sensor deployment) and the other is 3D surface deployment (e.g., mountain surface deployment). An underwater acoustic sensor self-deployment scheme is proposed in [17], which achieved relative good performance. Our work is about the former. A lot of applications of WSN are 3D scenario, so the autonomous deployment of mobile WSN in 3D space is of great significance [18]. It is hard to directly utilize 2D deployment method in 3D space [19]. One kind of solution for 3D coverage or 3D deployment is dimensionality reduction, such that [20] presented an efficient polynomial-time algorithm to solve the coverage problem in three-dimensional wireless sensor networks by translating the problem to 2D and 1D. However, it is not suitable for autonomous self-deployment algorithm. Another solution is to construct the 3D deployment model directly as [21]. However, this centralized method is lack of pervasiveness, because it can only be used in static deployment. Liu et al. [22] proposed a combined virtual force (CVF) algorithm for 3D autonomous deployment, and the performance of this algorithm is good, but the deployment duration is comparatively long.

2.4. The Gap between the Proposed 3DSD and Related Works. The topic of 3D node self-deployment in WSNs has already received considerable attention in the literature. However, our work is different from them because our proposed scheme is comprehensive. There are many practical requirements considered in our work, such as shortening the deployment process, avoiding the local optimum problem [23], handling the obstacles in AOI, and, meanwhile, maintaining the network connectivity.

3. Preliminaries

3.1. Problem Statement. The 3D self-deployment problem can be defined as follows: “Given a connected network of \( n \) sensors each of which has a fixed transmission and sensing range respectively, and are placed in a 3D space with a given area boundaries and some obstacles, our goal is to maximize the coverage area using these mobile sensors as rapidly as possible, meanwhile the network connectivity should be provided.” Taking the limited computation capacity into consideration, the deployment scheme should be lightweight. In the node self-deployment process, there may be some local optimum areas due to the obstacles; this situation must also be considered.

3.2. Three-Dimensional Sensing Model. We adopted a sphere perception model that node \( s \) is the center, and, \( R_i \) is the radius of the sphere [24]. As shown in Figure 1, within the sensing spherical region, the probability to be perceived is 1, while outside the sphere the probability to be perceived is 0. The perception function is expressed by

\[
C(s, p) = \begin{cases} 
1, & \text{if } d(s, p) \leq R_i \\
0, & \text{otherwise.}
\end{cases}
\]
In Figure 1, node $P$ is an arbitrary point in sphere, $C(s, p)$ represents the probability that $P$ can be perceived by sensor node $s$, and $d(s, p)$ denotes the Euclidean distance between the sensor node $s$ and the point $P$. Every sensor node has the same transmit power, that is, the same perception radius. In fact, the coverage issues based on the three-dimensional perception model are a problem, that is, how to set a ball with certain radius to cover a three-dimensional space completely. Taking the viewpoint of economic saving into consideration, we need to use fewer nodes to cover as many spaces as possible. Figure 2 is a schematic diagram of the ideal three-dimensional perception model in a three-dimensional space.

In the two-dimensional plane, seamless maximum effective coverage area of three circles is acquired when the three circles intersect at one point and their centers constitute a regular triangle. Compared to the two-dimensional plane, the coverage problem in three-dimensional space is more complex, and the best coverage problem in three-dimensional space is proved to be an NP-hard problem [25], but the foundational idea of the best coverage in the three-dimensional space is the same as that in two-dimensional space. First, two definitions are proposed as follows.

**Definition 1** ($V_{ECA}$). For an arbitrary node $S_1$, its effective coverage area $V_{ECA}$ is the coverage space of node $S_1$ minus the volume of the overlapping area $V_Z$:

$$V_{ECA} = V - V_Z = \frac{4}{3}\pi r^3 - V_Z. \quad (2)$$

**Definition 2** ($C_{ECA}$). For an arbitrary node $S_1$, its effective coverage efficiency $C_{ECA}$ is the ratio of the $V_{ECA}$ and the volume of node coverage area $V$:

$$C_{ECA} = \frac{V_{ECA}}{V} = \frac{V - V_Z}{V} = 1 - \frac{3V_Z}{4\pi r^3}. \quad (3)$$

Next, let us discuss the intersection of four nodes.

**Theorem 3.** There are four seamless topological spheres $S_1, S_2, S_3,$ and $S_4$. When the four balls intersect at a point $O$, and the centers of the sphere constitute a tetrahedral which is a regular tetrahedron, the seamless maximum effective coverage space can be obtained. Its volume is $(28\sqrt{6} - 24)\pi r^3 / 9$.

**Proof.** Here we divided the positional relationship of the four sphere centers into two cases. The first one is shown in Figure 3(a); add a ball $S_4$ to the situation in which maximum effective coverage has been achieved by three balls. We have $S_4S_1 = S_4S_2 = S_4S_3 = \sqrt{2}r$, and the distances between $S_1, S_2,$ and $S_3$ are $\sqrt{3}r$; after calculations, the volume of the coverage space can be obtained:

$$V'_4 = \frac{30\sqrt{2} + 27\sqrt{3} - 32}{12}\pi r^3. \quad (4)$$

The second case is shown in Figure 3(b), and the tetrahedral made up of $S_1, S_2, S_3,$ and $S_4$ is a regular tetrahedron, and the four spheres intersect at the center of tetrahedron. $OS_1 = OS_2 = OS_3 = OS_4 = r$, $OQ = r/3$, so we can get $d = S_1S_2 = S_2S_3 = S_3S_4 = S_4S_1 = 2\sqrt{6}r/3$; after calculations, the volume of the coverage space under this positional relationship can be obtained:

$$V''_4 = \frac{28\sqrt{6} - 24}{9}\pi r^3. \quad (5)$$

Obviously, it can be seen that the maximum effective coverage space in four balls is obtained when the four sphere centers constitute a regular tetrahedron. According to formulas (2) and (3), the increment of $V_{ECA}$ and $C_{ECA}$ (named $V_{ECAI}$ and $C_{ECAI}$) that a new ball is added into space can be calculated as

$$V_{ECAI} = V_{ball} - 6 \times V_{crown} = \frac{4}{3}\pi r^3 - 2\pi h^2 (3r - h), \quad (6)$$

$$C_{ECAI} = \frac{V_{ECAI}}{V} = \frac{V - V_Z}{V} = 1 - \frac{3V_Z}{4\pi r^3}.$$
where \( h = (3 - \sqrt{6})r/3 \); when \( h \) is substituted into formula (6), we get

\[
V_{ECAI} = \frac{14\sqrt{6} - 24}{9} \pi r^3
\]

and the corresponding \( C_{ECAI} \) is

\[
C_{ECAI} = \frac{V_{ECAI}}{V_{ball}} = \frac{7\sqrt{6} - 12}{6} = 85.774\%.
\]

3.3. Three-Dimensional Virtual Force Model. The idea of virtual force model in three-dimensional space is that there are interactive forces between every node in the 3D space. According to the threshold \( d_{th} \) set in advance, when the distance of two nodes is less than the threshold \( d_{th} \), the force is expressed as repulsion, and the nodes will move away from each other; when the distance between two nodes is greater than the threshold, the force is expressed as the attraction, and the nodes will come close to each other. When the distance of two nodes is equal to the threshold, the two nodes are in force balance state and maintain stationary.

The formula of the virtual force in three-dimensional space can be described as

\[
\vec{F}_{ij} = \begin{cases} 
(\omega_A (d_{ij} - d_{th}), \alpha_{ij}, \beta_{ij}, \gamma_{ij}) , & \text{if } d_{ij} > d_{th} \\
0 , & \text{if } d_{ij} = d_{th} \\
(\omega_R \frac{1}{d_{ij}}, \alpha_{ij}, \beta_{ij}, \gamma_{ij} ) , & \text{otherwise.}
\end{cases}
\]

The vector \( \vec{F}_{ij} \) represents virtual force of \( S_i \) subject to \( S_j \) in three-dimensional space, \( d_{ij} \) is the Euclidean distance between \( S_i \) and \( S_j \) in three-dimensional space, \( d_{th} \) is the threshold value, and \( \alpha_{ij}, \beta_{ij}, \gamma_{ij} \) is the direction angle of the space vector \( \overrightarrow{S_iS_j} \). \( \omega_A \) and \( \omega_R \) are the virtual attraction coefficient and the virtual repulsion coefficient.

Force that a sensor node subject to come not only from neighbor sensor nodes but also from obstacles, and this kind of force is treated as repulsion. Due to the repulsion force between node and obstacle, the node is able to be deployed around the obstacle to the open area. \( \vec{F}_{IR} \) represents repulsion force that obstacles exert on \( S_i \). Furthermore, in order to take some areas needing to be covered a priori into account, attractions coming from these regions are introduced, and the jointed force of these regions is represented as \( \vec{F}_{IA} \). Then, the resultant force subjected by an arbitrary node \( S_i \) is

\[
\vec{F}_i = \sum_{j=1,j\neq i}^{k} \vec{F}_{ij} + \vec{F}_{IR} + \vec{F}_{IA},
\]

where \( \vec{F}_i \) stands for the resultant force exerted on \( S_i \) and \( k \) denotes the number of sensor nodes in a specific three-dimensional space.

4. Description of 3DSD Algorithm

The design scenario of 3DSD is presented in this section. The explanation and analysis of some design details are also illustrated, such as node density calculating, clustering strategy, and movement location calculation. Finally, we explain the concrete execution steps of 3DSD.

4.1. Assumptions. The 3DSD algorithm is a theoretical model in 3D space. All the sensor nodes have mobility. To highlight the emphasis, some assumptions are proposed as follows.

1. The sensing area of each sensor node is a sphere; furthermore, the sensing radius and communication radius of each node are equivalent.
2. Each sensor node has mobility and also has the ability to calculate distance and angle using received signal strength and direction.
(3) Each sensor node can detect obstacles in its communication scope and can estimate the relative position of obstacles.

(4) The travelling speed of each node is very fast, so the time consumption of single movement of node can be ignored.

(5) The sensor node movement in its route would not be hindered by other nodes.

4.2. Density Calculation. The magnitude of virtual force is determined by density of nodes and obstacles. The spherical communication area of the sensor node is divided into \( k \) subareas evenly, and the value of \( k \) is larger, and the expression of density is more accurate. A partition of certain sensor communication area \( (k = 8) \) is illustrated in Figure 4, wherein the red irregular object represents obstacle. The sensing radius of node S1 is \( r \), and the communication radius is \( R \), and its communication area is evenly divided into 8 regions. Region 1 to region 4 are arranged from front to rear and from right to left in upper hemisphere, as well as in lower hemisphere. Node S2 is in region 2, and node S3 is in region 8, and the obstacle is sensed in region 5 which is shown as the blue shaded area in the figure. The node density and obstacle density are presented as follows.

(1) The node density of S1 can be calculated as

\[
D = \frac{N_A}{N_t},
\]

where \( D \) denotes the node density, \( N_A \) is the number of nodes in a subarea of S1, and \( N_t \) is the theoretical minimal number of nodes which locates in the sensing area and its sense range can cover the whole subarea. \( N_t \) is a parameter that is related to the sensor node’s sensing radius and communication radius. In the three-dimensional perception model, \( N_t \) can be approximately calculated based on \( C_{ECA} \) as follows:

\[
N_t = C_{ECA} \frac{K_m}{V_{ball}} = \frac{3C_{ECA}K_m}{4\pi r^3},
\]

(12)

where \( K_m \) is the volume of a node’s single subarea and \( r \) is the sensing range of a sensor node.

(2) The density of an obstacle is defined as

\[
D_O = \frac{O_m}{K_m},
\]

(13)

where \( O_m \) is the volume of obstacles that a sensor node can detect in single subarea and \( K_m \) is the whole volume of one subarea. Take Figure 4, for instance, assuming that the communication range of S1 is 10 m and the sensing range is 5 m. Here, suppose that \( O_m = 200 \), and from formula (8), we get that the \( C_{ECA} \) is 85.774%. The node density and obstacle density can be calculated as follows.

The node density of area 2 (or area 8):

\[
K_m = \frac{(4/3) \pi r^3}{k} = \frac{(4/3) \times \pi \times 10^3}{8} = 523.599
\]

\[
N_A = 1
\]

\[
N_t = \frac{3C_{ECA}K_m}{4\pi r^3} = \frac{3 \times 85.774\% \times 523.599}{4 \times \pi \times 5^3} = 0.858
\]

\[
D = \frac{N_A}{N_t} = \frac{1}{0.858} = 1.166.
\]

In area 5:

\[
K_m = \frac{(4/3) \pi r^3}{k} = \frac{(4/3) \times \pi \times 10^3}{8} = 523.599
\]

\[
O_m = 200
\]

\[
D_O = \frac{O_m}{K_m} = \frac{200}{523.599} = 0.382.
\]

There is one node in area 2 and area 8, respectively, so the node density of these two areas is equivalent. While the number of nodes in the other six areas is 0, the node density is 0. Similarly, obstacle only can be perceived in area 5, so the obstacle density of other areas is 0.

4.3. Clustering Strategy and Density Control. Sensor nodes need clustering to fulfill density control, and the clustering strategy should be simple and efficient to decrease the deployment duration. So, the minimum ID clustering algorithm is adopted here [26].

With nodes movement, there may appear a situation that the member nodes and head node need reselect. As depicted in Figure 5, nodes form two clusters in a certain moment; that is, S1, S2, S6, and S7 and S3, S4, and S5 are clustering...
into two groups, respectively, as in Figure 5(a). S1 and S3 are head node, and the others are member node. In the next moment (i.e., Figure 5(b)), S3 moves to the new location; the two clusters must be reclustered. The clustering operation carry out only once per iteration to reduce computational complexity.

Due to the presence of obstacles, the phenomenon “regional segmentation” may emerge; namely, the force that the nodes are subject to is balanced in local area. To solve this problem, cluster density control is adopted. After clustering is finished, the neighboring head nodes exchange the in-cluster node density. The cluster with higher density schedules some nodes to move in the cluster with lower density, in case the density difference is greater than a certain threshold. To do so, the local force balance is broken, and also the node distribution is more even. The threshold can be determined by the amount of nodes in advance.

4.4. Decision of Target Location. There is a key issue, that is, how to calculate the target location that the node needs to move to. The distance and direction information can be calculated according to the received signal strength and received signal direction, based on the virtual force model proposed in Section 3.2, and we define a virtual force vector \( \vec{F} \):

\[
\vec{F} = \begin{cases} 
(D(G - d_{th}), \alpha, \beta, \gamma), & \text{if } G > d_{th} \\
0, & \text{if } G = d_{th} \\
\left(\frac{D}{G}, \pi - \alpha, \pi - \beta, \pi - \gamma\right), & \text{otherwise,} 
\end{cases}
\]

(16)

where \( D \) stands for the node density of a certain subarea of a sensor node (1/k communication area) and \( G \) denotes the Euclidean distance between source node and its neighbor in the subarea. \( (\alpha, \beta, \gamma) \) represents the bearings on the direction in which source node points to its neighbor in case virtual force is expressed as attraction; to repulsion case, the direction angle should be \( (\pi - \alpha, \pi - \beta, \pi - \gamma) \). Here, \( d_{th} = 2\sqrt{r}/3 \), \( r \) is the sensing radius of nodes. The force introduced by an obstacle is calculated by another vector \( \vec{F}_O \) as

\[
\vec{F}_O = \left(\frac{D_O}{G_O}, \pi - \alpha, \pi - \beta, \pi - \gamma\right),
\]

(17)

where \( D_O \) is the density of an obstacle in a certain subarea and \( G_O \) denotes the Euclidean distance between the sensor node and the centroid of obstacle. After calculating the values of \( \vec{F} \) and \( \vec{F}_O \), we can get the value of the resultant force \( \vec{F}_k \) in each subarea: \( \vec{F}_k = \vec{F} + \vec{F}_O \). Let \( \vec{F}_{xk} \) represent the component of \( \vec{F}_k \) on the x-axis, let \( \vec{F}_{yk} \) denote the component of \( \vec{F}_k \) on the y-axis, and let \( \vec{F}_{zk} \) be the force component of \( \vec{F}_k \) on the z-axis. Then, the resultant force that comes from all partitions of a certain node is

\[
|\vec{F}_{con}| = \sqrt{\left(\sum_{m=1}^{k} |\vec{F}_{xk}|\right)^2 + \left(\sum_{m=1}^{k} |\vec{F}_{yk}|\right)^2 + \left(\sum_{m=1}^{k} |\vec{F}_{zk}|\right)^2},
\]

(18)

where \( |\vec{F}_{con}| \) is the resultant force that a node is subject to. The norm of \( |\vec{F}_{con}| \) denotes magnitude of force. Let \( (\alpha, \beta, \gamma) \) denote the direction of virtual force; these three parameters present the included angle formed by \( \vec{F}_{con} \) and the positive direction of x-axis, y-axis, and z-axis, respectively. In order to facilitate the calculation, three unit vectors are introduced here: they are \( \vec{e}_x = (1, 0, 0), \vec{e}_y = (0, 1, 0), \) and \( \vec{e}_z = (0, 0, 1), \) which stand for unit vector on x-axis, y-axis, and z-axis, respectively. Here, \( \theta_x \) is the angle between \( \vec{e}_x \) and \( \sum_{m=1}^{k} \vec{F}_{xk} \), \( \theta_y \) is the angle between \( \vec{e}_y \) and \( \sum_{m=1}^{k} \vec{F}_{yk} \), and \( \theta_z \) is the angle
between \( \vec{e}_z \) and \( \sum_{m=1}^{k} \vec{F}_{zk} \). Then, the direction angle between the old position and the new one is

\[
\alpha = \cos^{-1} \left( \frac{\sum_{m=1}^{k} \vec{F}_{zk} \cdot \cos \theta_z}{\sqrt{\left( \sum_{m=1}^{k} \vec{F}_{zk} \right)^2 + \left( \sum_{m=1}^{k} \vec{F}_{yk} \right)^2 + \left( \sum_{m=1}^{k} \vec{F}_{zk} \right)^2}} \right).
\]

(19)

where \( k \) is the number of subareas. \( \beta \) and \( \gamma \) also can be calculated using formula (19). The distance between the new position and the old position is \( d = D_{\text{cons}} / |\vec{F}_{\text{cons}}| \), where \( D_{\text{cons}} = D + D_O \); it denotes the sum of node density and obstacle density in the node’s communication range.

The movement range of a node calculated from resultant force may be very large, and the direction of each node is different. So, after moving, some nodes may be isolated. Here, a negotiation strategy is adopted to ensure network connectivity.

(1) Node \( n_i \) (who has decided to move a certain distance) sent an “ask” message to its neighbors, and the movement range and direction are included in the “ask” message.

(2) Neighbor node \( n_j \) (\( j = 1, 2, \ldots, n & \& j \neq i \)) estimates the network connectivity after \( n_i \) moving, and there are three cases as follows.

(a) If there is no effect on network connectivity after movement of \( n_i \), then \( n_j \) sends back “ok” message.

(b) If the new position of \( n_i \) is out of their communication range and \( \text{neighbor} (n_i) > 2 \), \( n_j \) sends back a “suggest” message which includes a suggested moving distance \( \vec{d} \).

(c) If the new position of \( n_i \) out of their communication range and \( \text{neighbor} (n_i) \leq 2 \), \( n_j \) sends back an “alert” message which includes a suggested moving distance \( \vec{d} \) as well.

(3) Node \( n_i \) determines the moving distance according to the acknowledgment message:

(a) If there is no “alert” message sent back and \( \text{numbers} \) (ok) \( \geq 2 \), the origin distance is maintained, in case number (ok) = 1; the maximum suggested distance \( \vec{d} \) is adopted; otherwise, the secondary maximum \( \vec{d} \) is accepted.

(b) In case that there is some “alert” message sent back, the minimum suggested distance \( \vec{d} \) within these “alert” messages is accepted.

To do so, there are at least two one-hop neighbors of \( n_i \). The negotiation introduces some communication overhead, but the 2-connected property is provided.

### Table 1: Input arguments of 3DSD.

| Arguments | Meaning |
|-----------|---------|
| \( k \)   | The number of partitions |
| \( \lambda \) | Duration of waiting for acknowledgement |
| \( \tau \) | Upper limit of waiting for resent “ok” message |
| \( \omega \) | The minimum distance threshold of node movement |
| \( \varepsilon \) | The gap threshold of average density |
| \( r \) | Sensing radius of sensor node |
| \( R \) | Communication radius of sensor node |

**Algorithm 1: Node deployment.**

4.5. Description of 3DSD. Each node maintains two types of states; one is cluster state, and the other is node state. The cluster state has three substates: “undecided state,” “cluster head,” and “membership,” while the node state has two substates: “undeployed” and “deployed.” The node state is used to distinguish whether the nodes have been deployed in present round. Arguments of 3DSD are listed in Table 1.

The pseudocode of 3DSD is provided in Algorithm 1. Here, \( \omega \) is an input argument used as a threshold, and it plays a critical role in the terminating of deployment. Its value is set as the 0.1 times communication radius of sensor node, such that the deployment duration and the coverage ratio are balanced, according to simulation experiment.

5. Simulation and Analysis

5.1. Simulation Environments. The size of the AOI is set as \( 80 \, \text{m} \times 80 \, \text{m} \times 80 \, \text{m} \). In order to calculate the volume of the obstacles and the coverage ratio, the AOI is divided into grids, and the region area is \( 1 \, \text{m} \times 1 \, \text{m} \times 1 \, \text{m} \). We assume that the sensing range is 10 m, and the communication range is 20 m. In the simulation, the success rate of sending and receiving messages is 100%. There are several obstacles in this three-dimensional space, and they are randomly placed, and in order to facilitate the calculation of obstacle density, the
shapes of the obstacles are set as rectangular in simulation. The boundaries of the three-dimensional space are also treated as an obstacle.

5.2. Effect and Evaluation

(1) In the three-dimensional space with obstacles, 120 nodes are scattered randomly in the center of the area. Then, the deployment commences. These 120 nodes finally are deployed evenly to the whole space under the functions of attraction and repulsion, and the effect of the algorithm implementation is shown in Figure 6.

(2) The ID distribution of cluster head node during the algorithm execution is shown in Figure 7.

In the figure, the abscissa represents the ID number, and the ordinate indicates the times the node was selected as cluster head in the total 30 deployment rounds. Algorithm has performed twice as illustrated in Figures 7(a) and 7(b), we can see the difference of the two results of execution, this is because the nodes are randomly placed in initialization stage, and different initial position will lead to forming different cluster during the algorithm execution.

(3) Density control strategy is used in 3DSD. Changes of the average density during the implementation are shown in Figure 8. The abscissa is deployment rounds, and the ordinate is the average density of clusters. From the beginning, there is high density because nodes are concentrated, slow nodes start to spread out, density becomes lower and lower, and node density between clusters tends to be balanced.

(4) Two most important indicators to assess deployment algorithm are coverage ratio and deployment rounds. Figures 9 and 10 are the results of the execution of 3DSD. Still in the previous environment of 3D space, the number of nodes increased from 5 to 120, the abscissas are number of nodes, and the ordinate is the coverage ratio and deployment rounds, respectively. We can see that with the gradual increase of the number of nodes, both coverage and deployment rounds show ascendant trend. The number of nodes is in a linear relationship with the coverage, and the
number of nodes is proportionate to the deployment rounds. When there are 120 nodes in the AOI, the highest coverage rate can reach 82%. But the theoretical calculated value of the highest coverage rate is 88%; nearly 6% of the error is caused by the system parameter setting and the too big grid.

5.3. Comparison with CVF. The CVF algorithm [22] also focuses on how to maximize the coverage ratio in 3D space by autonomous tactic. Comparisons are conducted between the CVF and our 3DSD algorithm. The two comparison indicators are coverage ratio and deployment time (i.e., deployment duration), as shown in Figures 11 and 12, respectively. The network coverage performances of these two methods are similar. The deployment rounds of these two methods are both increasing with the number of sensor nodes increased. But, the deployment time of 3DSD is less than the CVF. The reasons are that clustering and density control
5.4. The Flexibility of 3DSD. In previous subsection, the rapid deployment feature of 3DSD is indicated. The terminal criteria of deployment are related to value of $\omega$. When the amount of sensor nodes is enough to fully cover the AOI, we can adjust the value of $\omega$ to speed up the deploying process as illustrated in Figure 13. The amount of nodes is 200 in the identical area, and the value of $\omega$ varies from 10% of sensing radius to 30% of it. As we can see in Figure 13, in all of these five cases with different $\omega$, the coverage is the same, that is, 100%. However, the deployment duration is shortened along with $\omega$ increase. This result shows that 3DSD has flexibility, and it can provide rapid deployment for different initial condition.

6. Conclusion

A distribution autonomous deployment algorithm 3DSD is proposed in this paper. Aimed at the node deployment issue in 3D space, the virtual force model is adopted in 3DSD. The forces coming from boundaries of AOI, obstacles in sensing area, and the sensitive area are totally considered. The negotiation strategy is introduced to provide network connectivity. The node density control scheme is adopted such that the density between clusters is balanced. Simulation results show that the performance of 3DSD is preferable, and the coverage ratio of 3DSD is quite high.

The work focus on 3D autonomous deployment for WSN situates initially theoretical stage yet. In the future, the following issues should be considered: (1) to construct the 3D irregular sensing model, (2) to introduce error eliminating methods for computing and moving, and (3) to evaluate the feasibility of 3D node deployment algorithm in field experiment.

Conflict of Interests

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

Acknowledgment

This research was supported by The National Natural Science Foundation of China (61379023).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, pp. 393–422, 2002.
[2] R.-S. Ko, “Analyzing the redeployment problem of mobile wireless sensor networks via geographic models,” Wireless Communications and Mobile Computing, vol. 13, no. 2, pp. 111–129, 2013.

[3] Y.-Y. Shih, W.-H. Chung, P.-C. Hsiu, and A.-C. Pang, “A mobility-aware node deployment and tree construction framework for ZigBee wireless networks,” IEEE Transactions on Vehicular Technology, vol. 62, no. 6, pp. 2763–2779, 2013.

[4] D. W. Gage, “Command control for many-robot systems,” in Proceedings of the 19th Annual Technical Symposium and Exhibition on Expanding Missions, Unmanned Systems (AUVS ’92), Huntsville, Ala, USA, June 1992.

[5] R. V. Kulkarni and G. K. Venayagamoorthy, “Bio-inspired algorithms for autonomous deployment and localization of sensor nodes,” IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews, vol. 40, no. 6, pp. 663–675, 2010.

[6] Z. Lin, S. Zhang, and G. Yan, “An incremental deployment algorithm for wireless sensor networks using one or multiple autonomous agents,” Ad Hoc Networks, vol. 11, no. 1, pp. 355–367, 2013.

[7] G. Wang, G. Cao, and T. L. Porta, “Movement-assisted sensor deployment,” in Proceedings of the 23rd Conference of the IEEE Computer and Communications Societies (INFOCOM ’04), pp. 2469–2479, Hong Kong, March 2004.

[8] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies, vol. 2, pp. 1293–1303, IEEE Computer Society, San Francisco, Calif, USA, April 2003.

[9] Z. Jing and Z. Jian-Chao, “A virtual centripetal force-based coverage-enhancing algorithm for wireless multimedia sensor networks,” IEEE Sensors Journal, vol. 10, no. 8, pp. 1328–1334, 2010.

[10] X. Yu, W. Huang, J. Lan, and X. Qian, “A novel virtual force approach for node deployment in wireless sensor network,” in Proceedings of the 8th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS ’12), pp. 359–363, May 2012.

[11] H. Mahboubi, J. Habibi, A. G. Aghdham, and K. Sayrafian-Pour, “Distributed deployment strategies for improved coverage in a network of mobile sensors with prioritized sensing field,” IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 451–461, 2013.

[12] K. Mougou, S. Mahfoudh, P. Minet, and A. Laouiti, “Redeployment of randomly deployed wireless mobile sensor nodes,” in Proceedings of the 76th IEEE Vehicular Technology Conference (VTC Fall ’12), IEEE, Quebec City, Canada, September 2012.

[13] N. Bartolini, T. Calamoneri, T. F. La Porta, and S. Silvestri, “Autonomous deployment of heterogeneous mobile sensors,” IEEE Transactions on Mobile Computing, vol. 10, no. 6, pp. 753–766, 2011.

[14] C. Zhang and S. Fei, “Connectivity-preserved and force-based deployment scheme for mobile sensor network,” Wireless Personal Communications, vol. 77, no. 1, pp. 463–475, 2014.

[15] J. Li, B. Zhang, L. Cui, and S. Chai, “An extended virtual force-based approach to distributed self-deployment in mobile sensor networks,” International Journal of Distributed Sensor Networks, vol. 2012, Article ID 417307, 14 pages, 2012.

[16] X. Yu, N. Liu, W. Huang, X. Qian, and T. Zhang, “A node deployment algorithm based on van der Waals force in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 505710, 8 pages, 2013.