

Practical Node Deployment Scheme Based on Virtual Force for Wireless Sensor Networks in Complex Environment

Lu Wei¹, Yang Yuwang¹, Zhao Wei¹ and Wang Lei¹

¹ School of Computer Science and Engineering, Nanjing University of Science and Technology
Nanjing - China

[e-mail: yuwangyang@mail.njust.edu.cn]

*Corresponding author: Yang Yuwang

*Received August 2, 2014; revised October 10, 2014; revised November 21, 2014; revised January 19, 2015;
accepted January 26, 2015; published March 31, 2015*

Abstract

Deploying sensors into a target region is a key issue to be solved in building a wireless sensor network. Various deployment algorithms have been proposed by the researchers, and most of them are evaluated under the ideal conditions. Therefore, they cannot reflect the real environment encountered during the deployment. Moreover, it is almost impossible to evaluate an algorithm through practical deployment. Because the deployment of sensor networks require a lot of nodes, and some deployment areas are dangerous for human. This paper proposes a deployment approach to solve the problems mentioned above. Our approach relies on the satellite images and the Virtual Force Algorithm (VFA). It first extracts the topography and elevation information of the deployment area from the high resolution satellite images, and then deploys nodes on them with an improved VFA. The simulation results show that the coverage rate of our method is approximately 15% higher than that of the classical VFA in complex environment.

Keywords: wireless sensor network; node deployment; satellite image; virtual force algorithm

The work in this paper is supported by Assembly Research Fund (9140A05010113xxx), Key Technology R&D Program of Jiangsu, China (BE2012386, BE2011342), and the Agricultural Innovation Program of Jiangsu, China (CX(13)3054, CX(10)221, CX(11)2042), and Strategic Emerging Industry Development Program of Shenzhen, China (JCYJ20130331151710105).

1. Introduction

This paper is aimed to study the issue of node deployment in Wireless Sensor Networks (WSNs). Deployment is closely related to the accuracy, completeness and efficiency of sensory information retrieval. However, most researches on the sensor network deployment algorithms are confined to the laboratory or on paper because it is very expensive to deploy nodes in a real environment. Fortunately, with the rapid development of high-resolution satellite images, it is possible to deploy sensor nodes on them, which would be more practical than the previous approaches. The geographic information such as elevation and land cover types can be obtained from the satellite images. Therefore, this is a new approach to design and evaluate the deployment algorithms in the practical environment.

Satellite images generally refer to the pictures taken by satellites in space. Satellite images are used in different fields such as the military command, disaster relief, fire protection and other natural disaster monitoring. Senlet et al. [1] used satellite maps for precise localization of a mobile robot on sidewalks. In order to attain globally corrected localization results, they combined the stereo camera images, visual odometry, satellite map matching and a sidewalk probability transfer function obtained from the street maps. When the satellite images are used, the first issue is how to extract the information from the images. Nowadays, the classification and segmentation techniques for satellite images can be used to extract variant land cover types from the satellite images, such as water, grass, rock. Wilkinson et al. [2] presented a classical scheme for feature classification of the satellite images. Recently, there have also been researches [3][4] on the classification algorithms for high-resolution satellite images. In this paper, we extract the topography information from the satellite images, so some classical methods for classification and segmentation are used.

The deployment problem of mobile WSNs is also encountered during the deployment of mobile robots. Some universities and scientific research institutions have done lots of related work, and many algorithms [5] are proposed. The deployment methods can be roughly classified into three categories: the methods based on the geometric model [8], the methods based on the virtual potential field [9] and the methods based on the biological intelligence [10]. The method proposed in this paper belongs to the second category. This kind of algorithm uses the artificial potential fields for the self-diffusion of mobile nodes. Each node in the network is regarded as a virtual positive electric charge, which is repelled by the boundaries, obstacles and other nodes. All the nodes in the network area tend to spread to other places under the force of repulsion. In the meantime, these nodes need to avoid spreading out of the boundary. Finally, they reach an equilibrium state, which means the driven force equals to the resistance of each node. The advantage of this algorithm is that it is simple and straightforward to use, which can quickly achieve the purpose of spreading nodes to the whole sensing area. Moreover, each node has a short moving distance. On the other hand, the disadvantage is that it is easy to fall into local optimal solution [12-18].

One of the most classic literatures based on the virtual potential field method was written by Howard et al. [12], in which they provided a solution to the problem of deploying a mobile sensor network in unknown dynamic environments. Concretely, they described a potential-field-based approach for node deployment, in which nodes are regarded as virtual particles, and these nodes are subject to the virtual forces. These forces repel nodes from each other, and the obstacles. Moreover, these forces ensure that, from an initial compact state, nodes could spread out to maximize the coverage area of the network. In addition to these repulsive forces, nodes are also subject to a viscous friction force. This force is used to ensure that the network will eventually reach the state of static equilibrium. Similarly, the Virtual Force Algorithm (VFA) in [13] and the virtual spring force algorithm in [18] use both the repulsive and attractive force components to maximize coverage and uniformity for sensors.

In traditional VFA scheme, it is assumed that the sensing radius of each sensor node is the same. However, in some practical environment, the sensor nodes may locate in different terrains, so the transmission radius may be different. This paper proposes a deploy method for complex environment. Compared with previous methods, the main innovations and contributions of this paper are listed as follows:

- 1) We propose a deployment method based on the satellite images. Firstly, we extract the information of land cover types, area and height from the satellite images according to the locations at deployment fields. Then we convert the information into a deployment map, which is provided for the proposed virtual deployment method. In other words, the deployment can provide guidance for the practical deployment.
- 2) We propose an improved VFA suitable for deployment in the complex environment. The traditional virtual force algorithm is not suitable for deployment in such environment, so we put forward the improved VFA to implement the deployment. Then, through simulation, we evaluate the performance of the algorithm.

The remainder of this paper is organized as follows: Section 2 proposes the improved VFA suitable for virtual deployment in complex environment; Section 3 introduces the procedures for the classification and preprocessing of satellite images; Section 4 presents the virtual deployment scheme in detail. Moreover, we will evaluate both the proposed deployment algorithm and the existing classical algorithms through simulations; finally, Section 5 concludes the paper and points out the direction for future works.

2. VFA for Virtual-Deployment

In our virtual-force-based approach, the sensor nodes are modeled as points subject to the attractive or repulsive force according to the distance between every two sensors. By setting a threshold of desired distances between sensors, each sensor moves following the summation of the force vectors and achieves a uniform deployment eventually [19].

We try to overcome the limitations of previous approaches. In these approaches, a large communication range, a dense network, the obstacle-free fields and the full knowledge of the field layout are assumed. Based on the literatures [12], we put forward the new VFA. Each node in the deployment field is subject to four virtual forces:

- 1) Virtual force F_n among sensor nodes;
- 2) Attractive force F_m caused by uncovered area;
- 3) Repulsive force F_o between nodes and obstacles;
- 4) Repulsive force F_b between nodes and boundaries.

2.1 VFA for Virtual-Deployment

Assume that there exists attractive and repulsive virtual force between any pair of adjacent nodes in the sensor networks. The virtual force on node s_i due to s_j is F_{ij} ($j = 1, 2, \dots, k$ $s_j \in S_{Ni}$, and S_{Ni} is a collection of S_i 's neighbors).

$$F_{ij} = \begin{cases} 0, & d_{ij} = d_{th} \\ \omega_A(1/d_{th} - 1/d_{ij}), & d_{ij} > d_{th} \\ \omega_R(d_{th} - d_{ij}), & d_{ij} < d_{th} \end{cases} \quad (1)$$

In Eq. (1), d_{ij} denotes the Euclidean distance between nodes s_i and s_j (s_j is an adjacent node of s_i , so $d_{ij} < R_c$, R_c refers to communication radius), and d_{th} is the threshold distance. ω_A and ω_R denote the attraction and repulsion coefficients respectively [16]. The total force on node s_i due to its neighbors can be computed with Eq. (2).

$$\vec{F}_i = \sum_{s_j \in S_{Ni}} \vec{F}_{ij} \quad (2)$$

2.2 Attractive Force from Uncovered Area

To calculate the attractive force caused by an uncovered area, we divide the target area uniformly to form discrete virtual grids. Nodes are subject to an attractive force produced by those uncovered grids between the sensing radius R_s and the communication radius R_c . We define M as the total number of virtual grids in a region and u_m as the m -th grid. Consequently, u_m attracts sensor s_i if it is uncovered. Assuming that each virtual grid carries a unit of charge, and the charge number of sensor nodes is equal to the square of the sensing radius R_s , then,

$$F_{im} = \begin{cases} k \frac{q_i q_m}{d_{im}^2} = \frac{R_s^2}{d_{im}^2}, & R_s < d_{im} \leq R_c, u_m \notin \bigcup_{i=1}^N S_i \\ 0, & otherwise \end{cases} \quad (3)$$

In Eq. (3), q_i and q_m denote the electric charges carried by node s_i and grid u_m respectively. k is Coulomb's Constant and we set $k = 1$. R_s refers to the sensing radius, and

R_c stands for communication radius, where d_{im} denotes the Euclidean distance between grid u_m and node s_i . Note that only the uncovered grids have attractive force on sensor nodes. At the same time, we clip the virtual force to the domain $R_s < d_{im} \leq R_c$ for the purpose of distributed processing of local network. Finally, N stands for the total number of sensor nodes and S_i represents the effective coverage of node i .

2.3 Repulsive Force from Obstacles

The sensor nodes may be blocked by obstacles, which must be avoided during the deployment. A feasible approach is to regard obstacles as high potential fields. Then, sensor nodes could be repelled and kept away from the obstacles. Assume that the number of charges on per unit length of obstacles is equal to that of sensor node. Thus, F_{io} can be calculated with Eq. (4).

$$F_{io} = \begin{cases} \alpha \frac{q_i q_o}{d_{io}^2} (d_{oth} - d_{io}) = \alpha \frac{R_s^2}{d_{io}^2} (d_{oth} - d_{io}), & d_{io} < d_{oth} \\ 0, & d_{io} \geq d_{oth} \end{cases} \quad (4)$$

In Eq. (4), q_i and q_o denote the electric charges carried by node s_i and point o at the obstacles' edge respectively. The number of electric charges at point o is equal to R_s . d_{io} is the Euclidean distance between node i and point o . d_{oth} stands for the obstacle threshold distance, and nodes are subject to the repulsive force if $d_{io} < d_{oth}$. Finally, α represents the proportionality coefficient for force adjustment.

2.4 Repulsive Force from Boundaries

During the process of deployment, we need to constrain the size of deployment region to prevent nodes from spreading out of the fields. Similar to the approach for obstacles mentioned above, the region boundaries should also be taken as high potential fields so as to keep the nodes within the area. Assume that the number of charge on per unit length of boundaries is equal to that of the sensor nodes. Thus:

$$F_{ib} = \begin{cases} \beta \frac{q_i q_b}{d_{ib}^2} (d_{bth} - d_{ib}) = \beta \frac{R_s^2}{d_{ib}^2} (d_{bth} - d_{ib}), & d_{ib} < d_{bth} \\ 0, & d_{ib} \geq d_{bth} \end{cases} \quad (5)$$

In Eq. (5), q_i and q_o denote the electric charges carried by sensor node s_i and point b at the boundaries respectively. And the number of electric charges at point b is equal to R_s . d_{ib} is the Euclidean distance between node i and point b . d_{bth} stands for the boundary threshold distance. Finally, β represents the proportionality coefficient for force adjustment.

2.5 Coverage and Threshold Distance Optimization

The concept of coverage was introduced by Gage et al. [20]. Coverage is fundamentally

related to the effectiveness of monitoring an environment with a given set of sensors. Generally, coverage is considered as a metric of QoS for a sensor network.

According to the former VFA, we make a collection of discrete grids throughout the deployment area. Assume the total number of grids within deployment area is M_{all} , and the total number of grids covered by the sensor nodes is M_c . So coverage C can be computed by Eq. (6).

$$C = M_c / M_{all} \quad (6)$$

By setting the values of d_{th} , we can ensure that the nodes cover the deployment area in a certain density, or the deployment area coverage can be adjusted. However, if d_{th} is too small, the sensor node layout will be so tight that there is no guarantee to meet the coverage requirement. Contrarily, if d_{th} is too large, the sensor node will be sparse, and then it may cause coverage holes, as shown in Fig. 1(a).

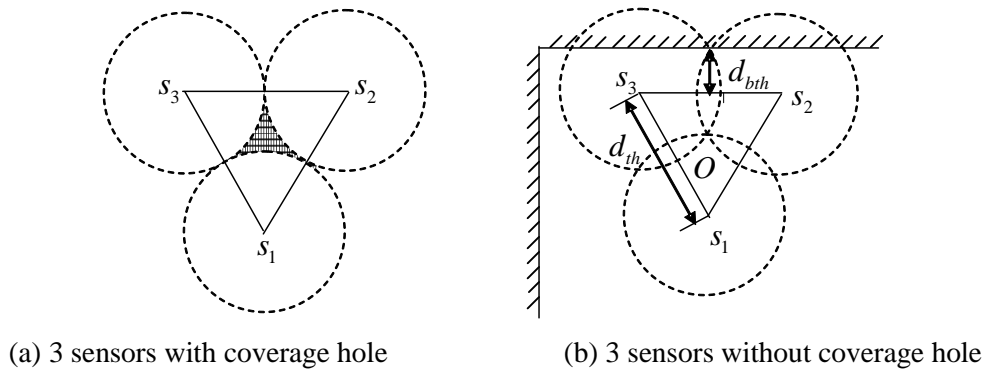


Fig. 1. Effect of d_{th} and d_{bth} on coverage

Consider Fig. 1(b), three sensor nodes s_1 , s_2 and s_3 overlap and intersect at point O . Although there is some overlapped area, Fig. 1(b) doesn't have coverage hole as that in the Fig. 1(a). After calculation, the distance between them in Fig. 1(b) is $\sqrt{3}R_s$. Similarly, the distance between nodes and boundaries is $0.5R_s$, so is the distance between nodes and obstacles. Thus, we set the threshold distance d_{th} to be $\sqrt{3}R_s$, the obstacle threshold distance d_{oth} to be $0.5R_s$, and the boundary threshold distance d_{bth} to be $0.5R_s$.

2.6 Virtual Force Analysis for Sensors

Consider the scenario shown in Fig. 2, there are four sensors s_1 , s_2 , s_3 and s_4 randomly deployed in a region, with obstacle O and boundary B . Then we analyze the virtual forces applied on these sensor nodes.

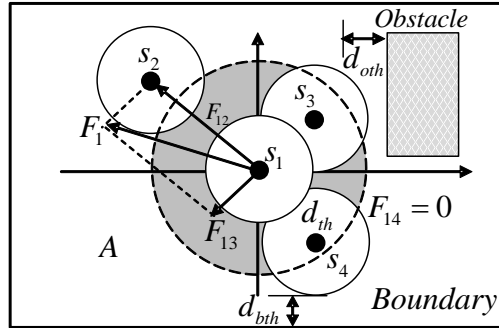


Fig. 2. Virtual force demonstration of 4 sensors

The total force of sensor node s_1 can be represented as $\vec{F}_1 = \vec{F}_n + \vec{F}_m + \vec{F}_o + \vec{F}_b$, where \vec{F}_n denotes the force on s_1 due to other nodes. Assume that the threshold distance $d_{th} = \sqrt{3}R_s$, $d_{12} > d_{th}$, $d_{13} > d_{th}$, and $d_{14} = d_{th}$. Thus, s_1 is subject to the virtual attractive force due to s_2 and the virtual repulsive force due to s_3 , and there is no force from s_4 . \vec{F}_m stands for the attractive force on s_1 due to the uncovered region represented by the shadow grids, which satisfies the condition $R_s < d_{lm} \leq R_c$. \vec{F}_o represents the repulsive force on s_1 due to obstacles, and it is zero because the distance is greater than $d_{oth} = 0.5R_s$. Finally, \vec{F}_b represents the repulsive force on s_1 due to the boundaries. For the same reason, it is also zero.

3. Satellite Image Classification and Preprocessing

In this section, **we introduce how to obtain the information from the satellite images.** First of all, these images are used to extract information such as the land cover types and elevation. Then, by using the information, we make a deployment map, and finish our deployment work on it, **as described in the next section.**



Fig. 3. Deployment area: Region A and Region B

The satellite images represent the real geographic features. Therefore, they can be used to obtain the information of the ground and learn such information as the geography, topography. This kind of information can also be applied for node deployment. The shape of deployment area is varied, and the terrain surface is complex and changeable. We chose Google satellite images as the source of regional information for our deployment work. Concretely, we need to preprocess the satellite images, calculate the deployment area, and extract the information, such as the land cover and elevation. Moreover, the obstacles and boundaries should be marked.

3.1 Selecting Deployment Area

We use latitude and longitude to mark the locations on satellite maps. We select two typical areas, as shown in [Fig. 3\(a\)](#) and [Fig. 3\(b\)](#) respectively. In region A, there is some area covered by water, which represents the regions with unreachable obstacles. Region B is a fluctuant terrain which includes a number of abrupt slopes. When sensor nodes try to move in such kind of terrain, they require more energy to overcome gravity. Therefore, these two regions are representative in our research.

The detailed parameters of these two regions are listed in [Table 1](#).

Table 1. Parameters of the deployment area

Parameters	Region A	Region B
Latitude (-90°~90°)	32.066835° N	32.092052° N
Longitude (-180°~180°)	118.939417° E	118.931519° E
Altitude (m)	500.00m	500.00m
Tilt (0~90°)	0	0
Azimuth (0~360°)	0	0
Image size (pixel)	720×576 pixel	720×576 pixel
Actual area (m ²)	539m×427m=230,153m ²	505m×421m=212,605m ²

3.2 Information Extraction from Deployment Area

We need to consider the land cover types of deployment area for the mobile node deployment. For example, the sensor nodes consume different energy in different terrains, and it is not suitable to deploy nodes in water. Therefore, in order to be closer to the real environment, we should classify the satellite images with features. Another important data is the height of the deployment area, which can be obtained through the satellite elevation maps. After getting these information, we can utilize them to make a deployment map for further deployment.

Step 1: Extract the Land Cover

In this section, the goal is to use a classification and segmentation algorithm to extract the land cover types, such as water, rocks, grass, mud.

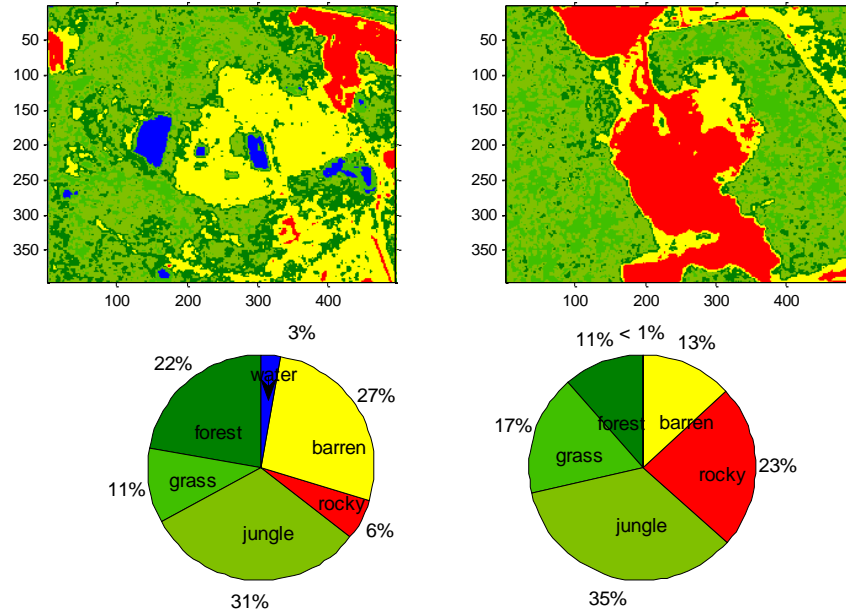


Fig. 4. Resulting plots of segmentation and classification after using K-means Algorithm

The algorithm we used is the K-means algorithm of ANN (Artificial Neural Network) [21]. It starts with some clusters of pixels in the feature space, each of them defined by its center. The first step is to allocate each pixel to the nearest cluster. In the second step, the new centers are computed with the new clusters. These two steps are repeated until convergence [22]. Fig. 4 shows the resulting plots of segmentation and classification on Fig. 3 after using the K-means algorithm.

Step 2: Obtain Elevation Map

The aim of obtaining the elevation information is to compute the nodes' energy consumption during the process of deployment. **The elevation information is obtained from GE (Google Earth). This software provides us with an interface to download the information of latitude, longitude, altitude, tilt, etc.** By specifying a target deployment area of screen coordinates (x, y) , we can obtain the elevation information of corresponding points on the satellite map. $X \times Y$ represents the size of the satellite image. *Pseudocode 1* shows the process:

Pseudocode 1. Input parameters: X, Y ; Output: Z

For $i=1$ to X

For $j=1$ to Y

$position = ge_app.GetPointOnTerrainFromScreenCoords(i, j);$

$Z(i, j) = position.Altitude;$

Endfor;

Endfor;

Where, $Z(i, j)$ denotes the elevation corresponding to the map coordinates (i, j) in

the picture. After obtaining all (X,Y,Z) coordinates, we can plot the 2D and 3D figures for Region A and Region B, as shown in Fig. 5 and Fig. 6 respectively.

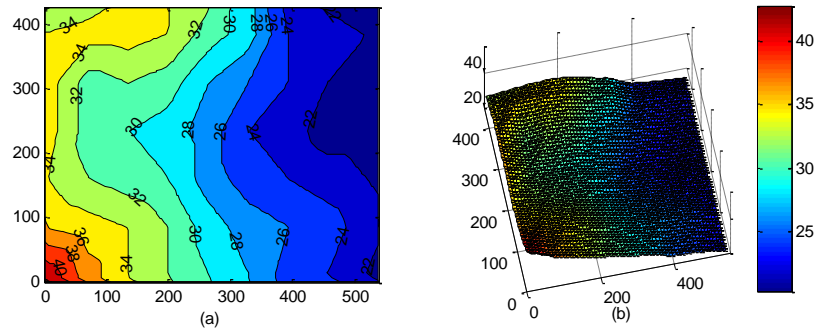


Fig. 5. 2D elevation maps (a) and 3D elevation maps (b) of Region A

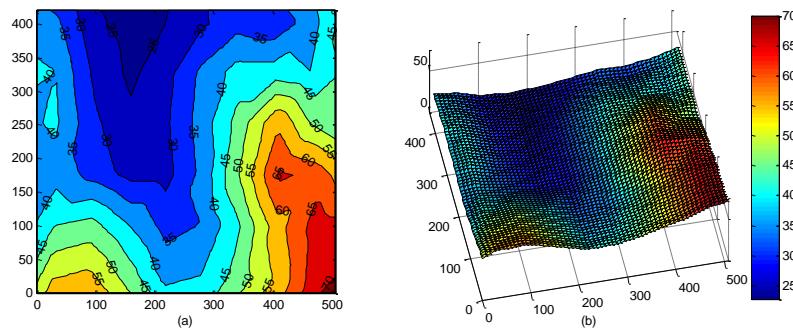


Fig. 6. 2D contour maps (a) and 3D elevation maps (b) of Region B

These 2D and 3D elevation maps can help us learn the difference in the terrains of the deployment area, which is useful when we plan to deploy nodes in those areas. In accordance with Fig. 5, the east terrain is high, while west terrain is low. The apex is approximately 40 meters high, and the lowest point is about 20 meters high. Thus, the drop height is about 20 meters. Fig. 6 shows that the highest elevation is about 70 meters, and the lowest is 25 meters, so the drop height is about 45 meters.

3.3 Making Deployment Map

In this section, we will make a deployment map that not only contains the land cover information but also the elevation information, and at the same time, the unreachable regions are marked. After putting the land cover map and elevation map together, we could achieve the deployment map, as shown in Fig. 7.

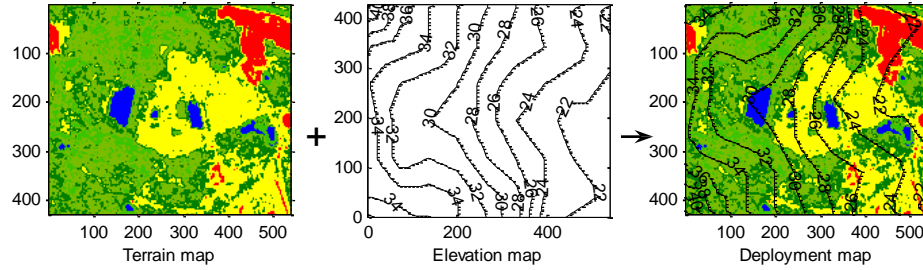


Fig. 7. Making a deployment map of Region A

Fig. 7 shows that the deployment map which contains both the land cover and elevation information, which also marks the unreachable areas (such as water). This deployment map will be used for the deployment in next section.

4. Deployment and Results

In this section, we will finish the deployment on the deployment map as shown in **Fig. 7**. The main contents of this section can be divided into 5 parts. 1) We need to estimate the total number of needed nodes before deployment according to the size of deployment area and the node sensing radius. 2) We introduce the node deployment steps for the VFA. 3) The sensors are deployed with a random deployment strategy and a manual deployment strategy, and the deployment results are compared. 4) The trajectory and the energy consumption of the node deployment are analyzed. 5) Finally, the deployment result is shown on a Google satellite map.

4.1 Calculate the Total Number of Needed Nodes

Coverage is critical for WSNs to monitor a region of interest and to provide a good QoS. In many practical scenarios, full coverage is required, which means that every point inside the region (excluding the obstacles) should be covered by at least one sensor. In this paper, one of our aims is to use the sensors to achieve maximal coverage for Region A and Region B.

Literatures [24] and [25] independently drew an important conclusion: if a convex area is completely covered by a set of nodes, these nodes must be connected when the condition $R_c \geq 2R_s$ is satisfied, and the coverage and connectivity must be guaranteed. Therefore, they assumed that $R_c = 2R_s$. In this paper, we also make the same assumption.

R. Kershner [26] proved the Disc Covering theorem: Let M denote a bounded plane point set and let $N(\varepsilon)$ be the minimum number of circles of radius ε which can cover M . Then,

$$\lim_{\varepsilon \rightarrow 0} \pi \varepsilon^2 N(\varepsilon) = (2\pi\sqrt{3}/9) \text{ meas } \bar{M} \quad (7)$$

Where, \bar{M} denotes the closure of M . Because the left side of Eq. (7) is simply the total area of the circles covering M , the constant $(2\pi\sqrt{3}/9=1.209)$ can be regarded as measuring the proportion of unavoidable overlapping. In this paper, the sensing radius R_s is equal to radius ε in Eq. (7).

Under the ideal propagation conditions, a radio wave propagates in free space and won't produce a reflection or scattering, so the energy cannot be absorbed by the obstacles. However, in the real environment, the communication radius R_c will change in accordance with different terrain surface. Assume that the sensing radius is proportional to the communication radius, thus the sensing radius R_s varies with different terrains. The actual distance of communication relates to the transmission power, receiver sensitivity, the working frequency and the environment, which can be expressed by Eq. (8).

$$FL(dB) = 20\lg d(km) + 20\lg f(GHz) + 92.44 + EL \quad (8)$$

Where, FL refers to the free space loss, d refers to the distance between two points, f is the working frequency, and EL is the attenuation caused by the environment. Assume that there are K land cover types in the deployment area, and according to Eq. (7), the number of nodes can be calculated by Eq. (9).

$$N = \sum_{i=1}^K \frac{(2\pi\sqrt{3}/9)A(i)}{\pi R_s^2(i)} \quad (9)$$

In Eq. (9), N is the total number of required nodes. $A(i)$ represents the area of each land cover types, and $R_s(i)$ denotes the sensing radius. We list the parameters in [Table 2](#).

Table 2. Deployment Parameters

Parameters	Value	Topography	Region A (%)	Region B (%)	EL
P_t (Transmit power)	0dBm	Forest	22.3%	11.2%	6.2dB
P_r (Received power)	-85dBm	Grass	10.6%	17.4%	4.5dB
f (frequency)	2.4GHz	Jungle	31.5%	34.9%	5.1dB
		Rock	5.9%	23.2%	2.2dB
		Barren	26.9%	13.2%	3.5dB
		Water	2.7%	0%	2.0dB

Now we provide an example of calculating the number of required nodes due to forest in Region A. Put parameters in [Table 2](#) into Eq. (8), and we obtain:

$$85dB = 20\lg d(km) + 20\lg 2.4 + 92.44 + 6.2dB \quad (10)$$

We can obtain $d = R_c = 86.7m$ with Eq. (10), so $R_s = 43.3m$ since $R_c = 2R_s$. According to [Table 1](#) and [Table 2](#), we know that the forest accounts for more than 22.3%, and $A=230,153 m^2$. Then, we could compute $N(\text{forest})=10.5 \approx 11$ with Eq. (8). Similarly, we could get the number of other corresponding nodes, which are listed in [Table 3](#).

Table 3. Sensor number and R_c , R_s

Topography	Nodes number of A	Nodes number of B	$R_c(m)$	$R_s(m)$
Forest	11	5	86.7	43.3
Grass	4	6	105.4	52.7
Jungle	12	12	98.4	49.2
Rock	2	4	137.3	68.7
Barren	7	3	118.3	59.1
Total Nodes	36	30		

4.2 Simulation Steps of Deployment

After getting the total number of required nodes for deployment, we can start the deployment work. For convenience, Region A is divided into $M_{all}=539 \times 427$ grids, and the division of Region B is $M_{all}=505 \times 421$, so that each small grid represents $1 m^2$. In accordance with the proposed improved VFA, we conduct simulation with Matlab. **Algorithm 1** shows the steps of the virtual force algorithm.

Algorithm 1. Virtual Force Algorithm suitable for virtual deployment

1. Initialize network, node location, threshold distance; set iteration number K ;
2. Calculate the coverage C , if not meet the goal, go Step 3, otherwise Step 10;
3. According to Eq. 3, calculate virtual force due to uncovered grids;
4. According to Eq. 1, calculate virtual force due to other nodes;
5. According to Eq. 4, calculate virtual force due to obstacles;
6. According to Eq. 5, calculate virtual force due to boundaries;
7. According to the results of Step 3,4,5,6, calculate total force;
8. Update nodes locations
9. Adjust sensing radius R_s according to nodes location;
if iteration is over, go Step 10, otherwise Step 2.
10. Simulation finishes. Output results.

We select Region A and Region B as our target areas, while the dots represent the sensor nodes, and the circles denote the sensing radius. K is set to be 500. The simulation parameters are shown in **Table 4**.

Table 4. Simulation parameters

Size of deployment area	Region A: 539m×427m; Region B: 505m×421m
Total nodes number (N)	Region A: 36; Region B: 30
Communication radius (R_c)	Vary with topography
Sensing radius (R_s)	Vary with topography
Grid size	1m×1m
Number of iterations (K)	500
Node mass (m)	1kg

d_{th}	$\sqrt{3}R_s$
d_{oth}	$0.5R_s$
d_{bth}	$0.5R_s$
ω_A, ω_B	2.0, 1.0
α, β	1.0, 1.0
Friction coefficient (μ)	0.1(rock); 0.25(barren); 1.0(grass); 2.5(jungle); 5.0(forest); 1.5(water)

4.3 Manual Deployment and Random Deployment

The scheme of deploying the sensor networks vary in practical application. It can be predetermined when the environment is sufficiently known and under control. The sensors can be strategically deployed manually. In this case, all the sensors will be deployed into an initial position, and then these nodes move to their destinations respectively. And the deployment can also be a priori undetermined when the environment is unknown or hostile. In this case, the sensor deployment cannot be pre-planned and must be performed manually. The sensors may be airdropped from an aircraft or deployed through other means. Such strategy can be considered as random deployment [27].

In this section, we perform the deployment of Region A and Region B with both the manual and random strategies respectively.

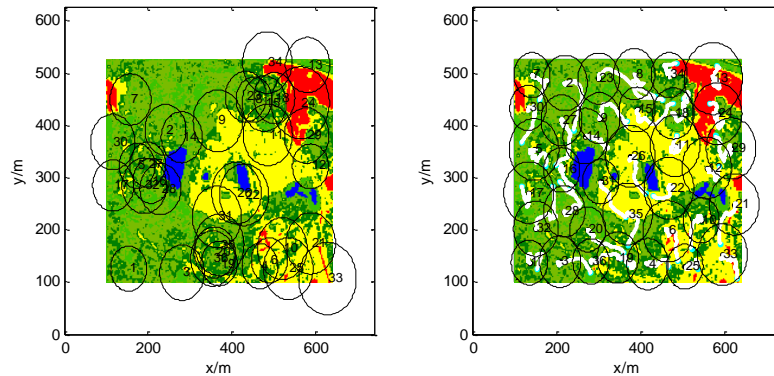


Fig. 8. Initial configuration (a) and final configuration (b) of Region A with Random Deployment strategy

Fig. 8(a) shows the initial configuration of nodes. All the nodes are randomly deployed in the field. After 500 iterations, all nodes reach the final configuration, as shown in **Fig. 8(b)**. In accordance with **Fig. 8(b)**, we observe that the node sensing radius R_s is different in different land cover types. The blue areas in the figure refer to water, marked as obstacles, which are not suitable for deploying nodes. Therefore, the nodes try to avoid these obstacles during the deployment process, and the distance between nodes and obstacles is determined by the threshold distance d_{oth} . Thus, the nodes well cover Region A.

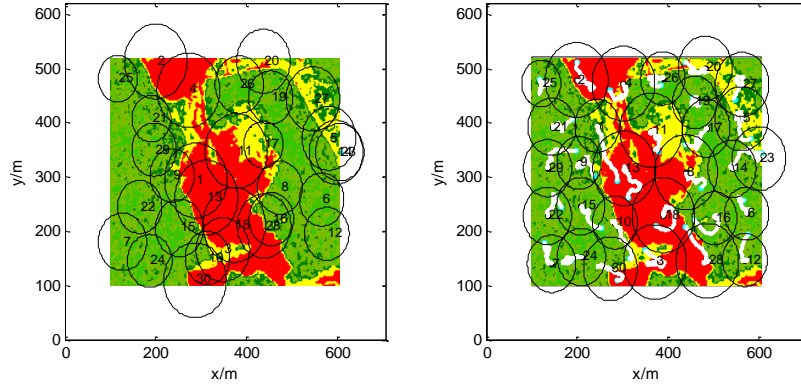


Fig. 9. Initial configuration (a) and final configuration (b) of Region B with Random Deployment Strategy

In accordance with **Fig. 9**, Region B does not have water in its terrain, and therefore, it has no obstacles compared to Region A. **Fig. 9(a)** is the initial state of the network, and **Fig. 9(b)** shows the final state after 500 iterations. From **Fig. 9(b)**, we observe that, because there is no obstacle, the nodes have more uniform distribution than those in Region A. By comparing **Fig. 8** with **Fig. 9**, we note that, although the two regions are of comparable sizes (Region A: $230,153 \text{ m}^2$, Region B: $212,605 \text{ m}^2$), Region A deploys 36 nodes while Region B only deploys 30 nodes. Region B has a larger scope of rocky terrain, and the nodes in the rocky area that sense a radius R_s are larger than those in other areas, so fewer nodes are needed.

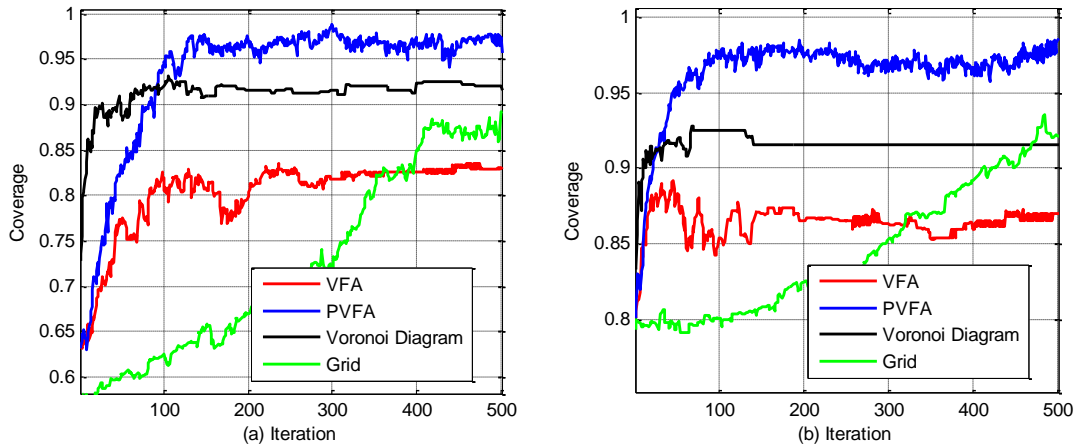


Fig. 10. Coverage of Region A (a) and coverage of Region B (b) with Random Deployment Strategy

Fig. 10 shows the coverage curves of Region A and Region B. The coverage rate increases as the iteration numbers increases. **Fig. 10(a)** is the coverage rate curve of Region

A. Because the nodes do not need to cover the water areas, the maximum coverage is close to 97% (see [Table 2](#), the water area accounting for 2.7%). [Fig. 10\(b\)](#) is the coverage rate curve of Region B, and coverage is close to 98% with the increasing iteration number. Compared to the traditional VFA and some classical deploy methods [\[28\]](#), the proposed VFA could provide higher coverage rate in both Regions. The method based on Voronoi Diagram could achieve high coverage rate when the iteration number is less. When the iteration number becomes much, the proposed method outperforms the method based on Voronoi Diagram.

Next, we deploy sensors in Region A and Region B with the manual deployment strategy. The initial locations of nodes can be arbitrarily chosen. In our experiments, we selected the lower left corner in the region as the initial position of nodes. With the progress of deployment, the nodes start spreading to other places from the lower left corner, which will cover the whole field at the end.

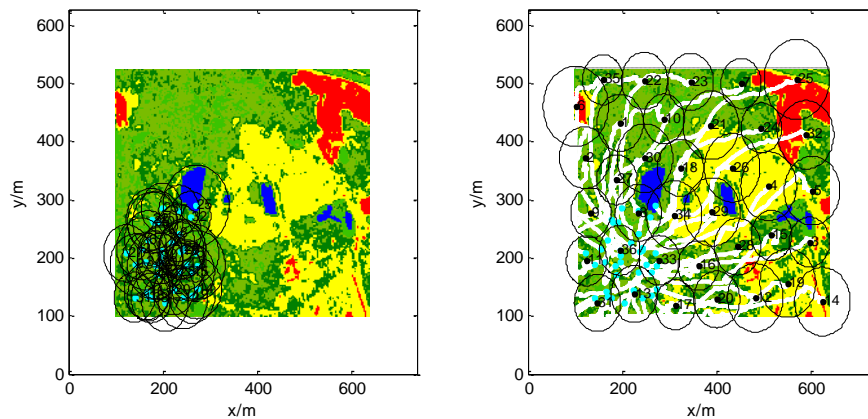


Fig. 11. Initial configuration (a) and final configuration (b) of Region A with Manual Deployment Strategy

From [Fig. 11\(a\)](#), we can see that the nodes are densely distributed in the lower left corner of Region A. At the end of the deployment process, the nodes reach the state as shown in [Fig. 11\(b\)](#). Compared to [Fig. 8\(b\)](#), the nodes' moving tracks are much longer because they need to move farther from the initial position to spread to other areas. Also we can see that during the process of deployment, the nodes can successfully avoid obstacles (water).

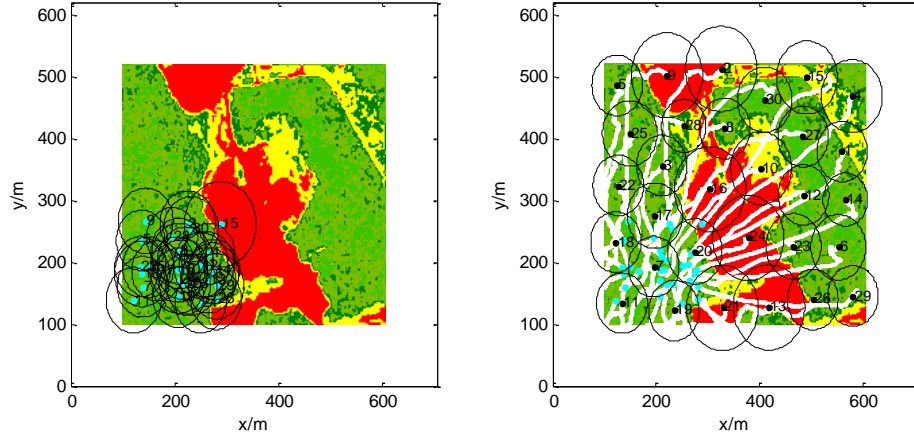


Fig. 12. Initial configuration (a) and final configuration (b) of Region B with Manual Deployment Strategy

In Region B, the initial position of nodes is also located at the left bottom corner, as shown in **Fig. 12(a)**. With the deployment process, the nodes gradually diffuse to the other parts of the region and reach the final state, as shown in **Fig. 12(b)**. Because there are no obstacles in Region B, compared to **Fig. 11(b)**, the mobile trajectories of nodes are straighter. In addition, the nodes do not need to avoid the obstacles and change their directions.

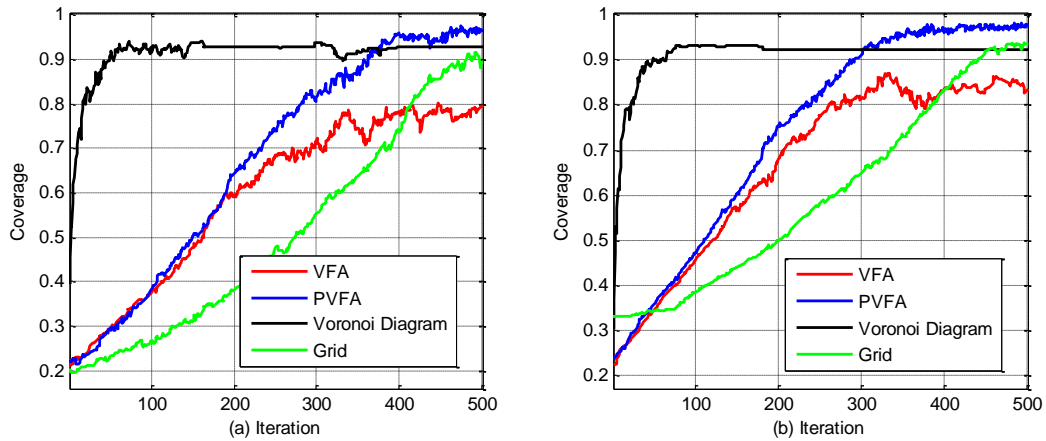


Fig. 13. Coverage of Region A (a) and coverage of Region B (b) with Manual Deployment Strategy

Fig. 13(a) and **Fig. 13(b)** show the coverage curves of Region A and Region B respectively with the manual deployment mode. Compared to the random deployment mode shown in **Fig. 10**, it requires more time to reach the maximum coverage state, because the nodes need to move farther to get to their target place.

Through analysis and comparison of the two deployment modes, we can draw the following conclusion: the manual deployment mode requires longer time to reach the maximum coverage than the random deployment mode, i.e., the nodes need to move a longer distance to reach their final position under the manual deployment mode. In addition, if the initial location of nodes can be more evenly distributed in the deployment area, the deployment time can be reduced.

4.4 Trajectory and Energy Consumption Analysis

In this paper, we only focus on the energy consumption of sensor nodes during the deploy process. The proposed scheme and some other similar schemes are centralized schemes. After a powerful computer calculates the destination for each sensor node, these nodes will move to its position. Therefore, the energy required to move will be studied in this subsection.

Through deployment on the satellite map, we can estimate the energy consumption of nodes according to the actual terrain and height during the deployment process. The analysis of energy consumption can provide a valuable reference for practical deployment.

During the process of deployment, the energy consumed by nodes comes from two aspects: firstly, the nodes must consume energy to overcome the ground friction; secondly, the nodes must overcome gravity. The deployment area's land cover is different, so the friction coefficient will be different. The friction coefficient used during the virtual deployment process is listed in [Table 4](#) and the consumption of friction can be expressed by Eq. (11).

$$E_{\mu} = \mu mgL \quad (11)$$

Where, μ is the friction coefficient, m is the node mass, and g is the acceleration of gravity. L denotes the length of the nodes' moving trajectory. The friction coefficient μ varies with different land cover types. The energy required to overcome gravity is as follows.

$$E_g = \begin{cases} mg\Delta H, & \text{for } \Delta H > 0 \\ 0, & \text{for } \Delta H \leq 0 \end{cases} \quad (12)$$

In Eq. (12), ΔH stands for height difference between the starting position to the end position. When a node moves from high position to low position, ΔH is negative. From a practical perspective, E_g should be set to be zero when ΔH is negative. If the path from the starting position to the end position includes multiple uphill parts, E_g is the sum of height differences of each uphill part.

We assume that the i -th node's total energy consumption is E_t^i , then $E_t^i = E_{\mu}^i + E_g^i$, where, E_{μ}^i and E_g^i refer to the energy required by the node i to overcome the friction and gravity respectively. In the random deployment mode, because the initial position of each node i is random, so every distance it moved is not necessarily equal, and E_{μ}^i is not

necessarily equal either. Assuming that the initial energy of each node is equal, the minimum value is E_{\min} , and we must satisfy Eq. (13) to ensure successful deployment.

$$E_{\min} \geq \max \{E_t^i \mid i = 1, 2, \dots, N\} \quad (13)$$

In Eq. (13), N is the total number of nodes within the target region. E_{\min} cannot represent the real situation through one virtual deployment, so we should calculate the average E_{\min} through multiple virtual deployments. And the more times we do, the closer E_{\min} is to the real case. For example, as in Region A, Fig. 14 shows the results after carrying out 500 times of random virtual deployment.

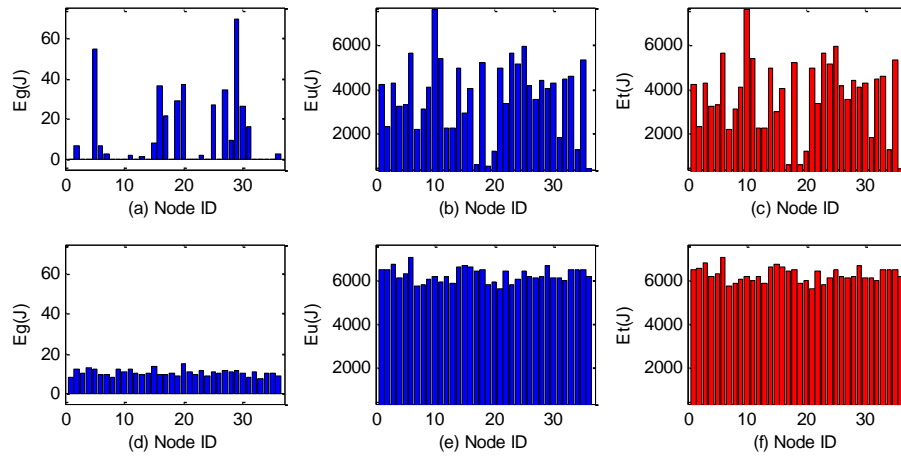


Fig. 14. Energy consumption of Region A with random deployment

Fig. (14) shows the energy consumption of nodes in Region A by using the random deployment mode, where Fig. 14(a) and Fig. 14 (b) represent the energy required to overcome the gravity and friction respectively. Fig. 14(c) shows the total energy consumption during the random deployment. Fig. 14(d), Fig. 14(e) and Fig. 14(f) represent the nodes' average energy consumption histogram after iterating for 500 times. In accordance with Fig. 14(a), the energy required to overcome gravity is positive when a node moves from a low altitude to a high altitude. Moreover, the energy required to overcome gravity is zero when a node moves from a high altitude to a low altitude.

The selection of E_{\min} should refer to the E_t mean in Fig. 14 (f). It is meaningful to determine the value of E_{\min} before real deployment, which can provide valuable reference to determine how much energy the nodes should carry initially.

Unlike the random deployment mode, the initial position of nodes is known and fixed in the manual deployment mode. The initial position invariant means that the result of each deployment is the same, i.e., the final location of the nodes is changeless. Then the path of each node traversed during the deployment process is also the same, which leads to constant power consumption. We choose Region B as the target area, and the trajectory and power

consumption of the node in the manual deployment mode are shown in **Fig. 15**.

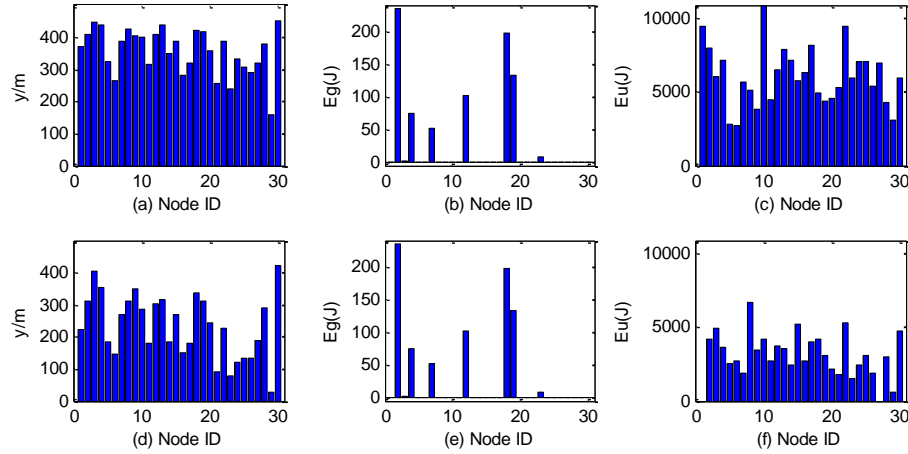


Fig. 15. Energy consumption of Region B with manual deployment

Fig. 15(a) shows the nodes' moving trajectory and moving distance; **Fig. 15(b)** shows the energy required for each node to overcome the gravity; **Fig. 15(c)** shows the energy consumption for each node to overcome the friction; **Fig. 15(d)** shows the straight-line distance between the nodes' initial position and final position. It is easy to observe that the straight-line distance is shorter than the actual moving trajectory.

When the nodes are pre-deployed, its initial position and final position are prior known and fixed, so we can use the path optimization algorithm to optimize the node's moving trajectory. This can realize the purpose of saving energy consumption. We use Lumelsky and Stepanov's path planning algorithm [32] called BUG2 to help a sensor move from a starting point to a destination point in a region with obstacles. And then the friction energy consumption E_μ can be optimized. By comparing the energy consumption before optimization in **Fig. 15(c)** with the energy consumption after optimization in **Fig. 15(f)**, we observe that E_μ has significantly decreased after optimization. Finally, the selection of E_{\min} should refer to the maximal value of $E_g + E_\mu$. After calculating the energy required for each node, the minimum initial energy E_{\min} could be determined. This result guarantees that the deployment method could finish with a high probability.

4.5 Deployment Effect on Satellite Image

The final effect of deploying nodes on satellite images is showed in **Fig. 16** and **Fig. 17**.

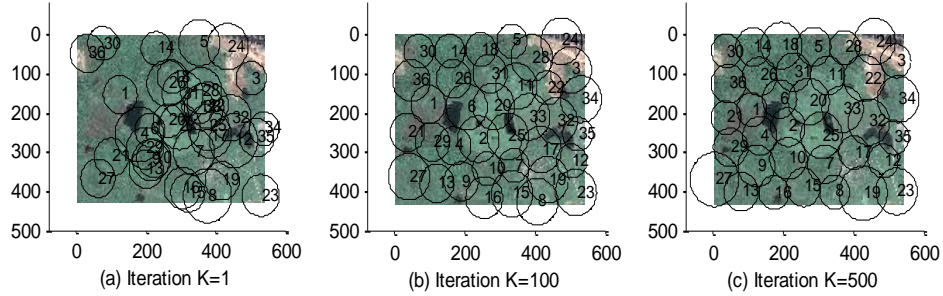


Fig. 16. Virtual deployment effect of Region A by using Random Deployment Mode

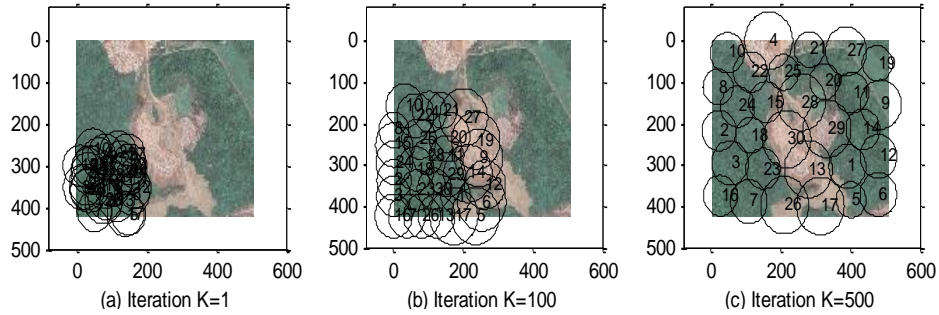


Fig. 17. Virtual deployment effect of Region B by using Manual Deployment Mode

Fig. 16 shows the initial state, middle state and final state of Region A, and the employed strategy is the random deployment mode. **Fig. 17** shows the initial state, middle state, and final state of Region B, and the employed strategy the manual deployment mode.

6. Conclusion and Future Work

In this paper, we provide an improved VFA for node deployment in the complex environment. Moreover, we use the Google satellite maps to extract practical information such as the land cover and elevation. Then, based on these practical information, we evaluate the proposed algorithm. In accordance with the experimental results, the proposed algorithm could provide 15% higher coverage compared to the traditional VFA.

Compared to the real environment, the information obtained through satellite maps still has a few shortages. For example, the newest satellite map can only provide dated photographic images instead of real-time images. But the proposed method could provide a relatively real deployment environment, which is advanced than the software simulation. The future research includes constructing a more accurate wireless signal transmission model. In the new model, the influence of weather should be considered.

References

- [1] T. Senlet, A. Elgammal, "Satellite image based precise robot localization on sidewalks," in *Proc. of 2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2647-2653, 2012. [Article \(CrossRef Link\)](#)
- [2] G. G. Wilkinson, "Results and implications of a study of fifteen years of satellite image classification experiments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 433-440, 2005. [Article \(CrossRef Link\)](#)
- [3] Z. Meng, B. Xiao, "High-resolution satellite image classification and segmentation using Laplacian graph energy," *Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 605-608, 2011. [Article \(CrossRef Link\)](#)
- [4] T. Nguyen, J. Han, D. C. Park, "Satellite image classification using convolutional learning," *11th International Conference of Numerical Analysis and Applied Mathematics 2013*, vol. 1558, no.1, pp. 2237-2240, 2013. [Article \(CrossRef Link\)](#)
- [5] G. Tuna, V. C. Gungor, K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters," *Ad Hoc Networks*, vol. 13, pp. 54-68, 2014. [Article \(CrossRef Link\)](#)
- [6] S. Sengupta, S. Das, M. D. Nasir, B. K. Panigrahi, "Multi-objective node deployment in WSNs: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity", *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 405-416, 2013. [Article \(CrossRef Link\)](#)
- [7] V. Gallart, S. Felici-Castell, M. Delamo, A. Foster, J. J. Perez, "Evaluation of a real, low cost, urban wsn deployment for accurate environmental monitoring," in *Proc. of IEEE 8th International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, pp. 634-639, 2011. [Article \(CrossRef Link\)](#)
- [8] M. R. Ingle, N. Bawane, "An energy efficient deployment of nodes in wireless sensor network using Voronoi diagram," in *Proc. of 3rd International Conference on Electronics Computer Technology (ICECT)*, 6, pp. 307-311, 2011. [Article \(CrossRef Link\)](#)
- [9] N. Aitsaadi, N. Achir, K. Boussetta, G. Pujolle, "Artificial potential field approach in WSN deployment: Cost, QoM, connectivity, and lifetime constraints," *Computer Networks*, vol. 55, no. 1, pp. 84-105, 2011. [Article \(CrossRef Link\)](#)
- [10] C. Ozturk, D. Karaboga, B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm," *Sensors*, vol. 11, no. 6, pp. 6056-6065, 2011. [Article \(CrossRef Link\)](#)
- [11] W. Yiyue, L. Hongmei, H. Hengyang, "Wireless sensor network deployment using an optimized artificial fish swarm algorithm," in *Proc. of 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, vol. 2, pp. 90-94, 2012. [Article \(CrossRef Link\)](#)
- [12] A. Howard, M. J. Matarić, G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," *Distributed autonomous robotic systems 5*. Springer Japan, pp. 299-308, 2002. [Article \(CrossRef Link\)](#)
- [13] Y. Zou, K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, 2003. [Article \(CrossRef Link\)](#)
- [14] Y. Zou, K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61-91, 2004. [Article \(CrossRef Link\)](#)
- [15] S. Poduri, G. Sukhatme, "Constrained coverage for mobile sensor networks," in *Proc. of ICRA'04. IEEE International Conference on Robotics and Automation, 2004. IEEE*, 1, pp. 165-171, 2004. [Article \(CrossRef Link\)](#)
- [16] N. Heo, P. K. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," *Wireless Communications and Networking. WCNC 2003*. vol. 3. 2003. [Article \(CrossRef Link\)](#)
- [17] N. Heo, P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks,"

- IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 1, pp. 78-92, 2005. [Article \(CrossRef Link\)](#)
- [18] B. Shucker, J. K. Bennett, "Scalable control of distributed robotic macrosensors," *Distributed Autonomous Robotic Systems 6*, Springer, Japan, pp. 379-388, 2007. [Article \(CrossRef Link\)](#)
- [19] M. C. Akewar, N. V. Thakur, "A Study of Wireless Mobile Sensor Network Deployment," *International Journal of Computer Networks and Wireless Communications*, vol. 2 pp. 533-541, 2012.
- [20] D. W. Gage, "Command control for many-robot systems," *Naval Command Control And Ocean Surveillance Center RDT And E DIV*, San Diego CA, 1992.
- [21] A. T. Sapkal, C. Bokhare, N. Z. Tarapore, "Satellite Image Classification using the Back Propagation Algorithm of Artificial Neural Network," *Technical Article*, pp. 1-4, 2006.
- [22] X. Tan, S. Chen, Z. H. Zhou et al, "Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 875-886, 2005. [Article \(CrossRef Link\)](#)
- [23] D. Lu, Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823-870, 2007. [Article \(CrossRef Link\)](#)
- [24] X. Wang, G. Xing, Y. Zhang, et al. "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. of the 1st international conference on Embedded networked sensor systems. ACM*, pp. 28-39, 2003. [Article \(CrossRef Link\)](#)
- [25] H. Zhang, J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1 no. 1-2, pp. 89-124, 2005. [Article \(CrossRef Link\)](#)
- [26] R. Kershner, "The number of circles covering a set," *American Journal of Mathematics*, pp. 665-671, 1939. [Article \(CrossRef Link\)](#)
- [27] J. Lee, A. D. Dharne, S. Jayasuriya, "Potential field based hierarchical structure for mobile sensor network deployment," in *Proc. of American Control Conference, 2007. ACC'07. IEEE*, pp. 5946-5951, 2007. [Article \(CrossRef Link\)](#)
- [28] G. Wang, G. Cao, T. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640-652, 2006. [Article \(CrossRef Link\)](#)
- [29] N. Bartolini, G. Bongiovanni, T. L. Porta et al, "Voronoi-based deployment of mobile sensors in the face of adversaries," in *Proc. of IEEE International Conference on Communications (ICC). IEEE*, pp. 532-537, 2014. [Article \(CrossRef Link\)](#)
- [30] C. H. Wu, K. C. Lee, Y. C. Chung, "A Delaunay triangulation based method for wireless sensor network deployment," *Computer Communications*, vol. 30, no. 14, pp. 2744-2752, 2007. [Article \(CrossRef Link\)](#)
- [31] R. C. Luo, O. Chen, "Mobile sensor node deployment and synchronous power management for wireless sensor networks," *IEEE Transactions on Industrial electronics*, vol. 59, no. 5, pp. 2377-2385, 2012. [Article \(CrossRef Link\)](#)
- [32] V. J. Lumelsky, A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, 2, no. 1-4, pp. 403-430, 1987. [Article \(CrossRef Link\)](#)



Lu Wei received his bachelor degree from the department of computer science, Shenyang Institute of Aeronautical Engineering, China in 2000, and master degree from Nanjing University of Science & Technology (NUST) in 2008. Currently, he is studying for his Ph.D degree in NUST. His research interests include wireless communication and information security.



Yang Yuwang received his bachelor degree from NorthWestern Polytechnical University in 1988, master degree from University of Science and Technology of China in 1991, and Ph.D degree from Nanjing University of Science and Technology (NUST) in 1996. From 2000 to 2001, he was a visiting scholar in the Department of Computer Science, South Bank University, London. From 2002 to 2009, he is an associate professor of Computer Science Department at NUST. Currently, he is a professor of Computer Science Department at NUST. His research interests are wireless sensor network, industry control network, and intelligent system. His email address is yuwangyang@mail.njust.edu.cn.



Zhao Wei received master degree in computer science from Nanjing University of Science & Technology (NUST) of China in 2006. Currently he is studying for Ph.D. degree in NUST. His research interests are wireless sensor network, routing protocol, network coding. His email address is zhaoweinjst@163.com.



Wang Lei received his bachelor degree from Anhui University of China in 2008. He has been a Master-Doctor combined program graduate student in computer science of Nanjing University of Science & Technology (NUST), China Since 2008. Currently, he is a visiting student in the department of computer and electrical engineering in Michigan State University, USA. His research interests are network coding, wireless sensor network, multipath routing, distributed storage system. His email address is kingstones@live.cn.