Ad Hoc Networks 11 (2013) 355–367

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

An incremental deployment algorithm for wireless sensor networks using one or multiple autonomous agents

Zhiyun Lin *, Sijian Zhang, Gangfeng Yan

Asus Intelligent Systems Lab, College of Electrical Engineering, Zhejiang University, 38 Zheda Road, Hangzhou 310027, PR China

ARTICLE INFO

Article history:
Received 31 December 2011
Received in revised form 30 May 2012
Accepted 26 June 2012
Available online 4 July 2012

Keywords:
Sensor network
Incremental deployment
Autonomous agent

ABSTRACT

The paper studies the deployment problem of wireless sensor networks using one or multiple autonomous agents. An online incremental algorithm based on Voronoi partition is proposed to solve the problem, for which each agent deploys sensors one-at-a-time with the objective of using less number of sensors to cover an area and maintain communication connectivity. A probabilistic sensor sensing model is applied for area coverage evaluation. The shape of target area is assumed to be known by the agents, but how the environment affects the communication is unknown a priori. Therefore, the agents are desired to autonomously place every new sensor at an appropriate location based on deployed sensors to ensure connectivity and coverage specifications. Both simulations and experiments using our self-made wireless sensors are conducted to validate the algorithm.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Wireless sensor networks have received significant attention during the last decade due to the fact that recent technological advances (e.g., sensing, computing, and IC manufacturing) have led to the emergence of small, powerful, smart sensors. Moreover, with the development of RF technology and wireless communication protocols, a wireless sensor network is possible to be deployed in a wide unknown area for the task of collecting information or in remote and hospitable areas for monitoring things of interests. Werner-Allen et al. [1] set up a sensor network consisting of 16 sensors to detect the seismic signals. The sensors are installed in a roughly linear configuration and the configuration ensures the radio connectivity. For habitat monitoring, a wireless sensor network is developed by Mainwaring et al. [2] and has been deployed on the Great Duck Island off the coast of Maine. This remote monitoring system is managed through Internet and is able to sample weather conditions such as temperature, barometric pressure, and humidity.

In wireless sensor networks, deployment is one of several fundamental issues that has been widely studied [3]. Random deployment is a typical technique for large-scale sensor networks. However, it may require much more redundant sensors to be deployed in order to achieve a given specification. Such a dense sensor network may not be desired due to cost, power, terrain commonality, and self-interference as suggested by Bulusu et al. [4]. More recently, several techniques called self-deployment are proposed assuming sensors own mobility. For example, potential fields [5] or virtual forces [6] based approaches are used to spread sensors out from a compact or random initial configuration to cover an unknown area. Mobile senors are subject to either attractive or repulsive forces according to the distances between their neighbors and themselves. The deployment process terminates when every sensor arrives at a location with zero composite artificial force from its neighbors. Appropriately designing the artificial force function would ensure that sensors are not overly clustered. However, it is unrealistic that every sensor is equipped with a high performance mobile platform and certain expensive devices (e.g., laser range-finder or

* Corresponding author. Tel.: +86 571 87951637; fax: +86 571 87952152.

E-mail addresses: linz@zju.edu.cn (Z. Lin), zsjzju@yahoo.com.cn (S. Zhang), ygf@zju.edu.cn (G. Yan).

1570-8705/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2012.06.009

omni-directional camera) to determine the range and bearing of nearby sensors and obstacles.

In this paper, we propose to use one or multiple autonomous agents to set up a static wireless sensor network in an area incrementally. A few autonomous mobile agents are assumed to be equipped with GPS or other localization devices, and are capable of navigating in the target area and deploying sensors autonomously. Thus, the equipment costs are relatively low, while a reasonable deployment can be achieved with less number of sensors compared with random deployment. The approach lies in the category of incremental deployment. An incremental deployment algorithm is proposed by Howard et al.[7], where sensors are deployed one-at-a-time based on the data gathered from previously deployed sensors. Nevertheless, it is assumed that all sensors should be able to "talk" with a remote base-station when the deployment algorithm is executed. An earlier work [8] on autonomous exploration is also related to the incremental deployment idea, where an occupancy map of the environment is built to find out the "frontiers" between open space and unexplored space and the map is updated continuously when a robot moves to the nearest frontier and gains new information of the unexplored space. We adopt the idea of exploring un-deployed regions and take into account the practical communication constraints of sensors in the deployment process. Two important metrics are considered to evaluate the performance of sensor deployment, namely, coverage and topology connectivity. According to the subject to be covered, the coverage types can be categorized as *area coverage, target coverage* and *barrier coverage* [9]. This paper mainly focuses on area coverage, which addresses the problem of how to cover the whole target area. The objective in this paper is to set up a sensor network with possibly the least number of sensors such that an area is fully covered and every sensor in the network is able to communicate with at least two neighbors. The first is for the coverage purpose like the Art Gallery problem [10] while the second ensues successful data collection from every sensor to the base station. Communicating with at least two neighbors (2-connectivity) offers robustness to single link communication failures. So the problem studied in this paper is different from the pure coverage problem like [11,12] where the main idea is to find out the coverage hole and then deploy a sensor to fill it. In [13], it is assumed that the detection probability of a target by a sensor is isotropic and diminishes with distance. Moreover, it is assumed that the sensors close to a surveillance spot make contributions to data fusion. We use the similar sensing model and define neighboring sensors close to a target point based on the Voronoi partition. In real setting environment, however, the communication field is irregular because of the effects of environmental disturbance and obstacle blocking. Thus, off-line pre-planning of deploying spots is not possible though the target area is known. A recently important work [14] searches the optimal placements, for which sensors are not only informative but also able to communicate efficiently. It proposes an algorithm to estimate predictive power and communication cost of unsensed locations. We adopt the idea of joint sensing quality, for which a point sensed by neighboring sensors is modeled as a joint probabilistic sensing model. But unlike the predictive algorithm, we propose to let each autonomous agent detect the communication effects when it moves and then deploy a sensor at an appropriate location to extend the covered region and ensure 2-connectivity. This method evaluates the communication effect in real environment.

The incremental deployment algorithm we propose consists of several steps. Initially, the first sensor is deployed in the target area according to certain heuristic rules for fast convergence. Secondly, each autonomous agent computes the Voronoi partition of the target area with the Voronoi center at the sensors already deployed. Thirdly, each autonomous agent computes its moving target point. If the Voronoi cell that the agent currently locates has not been completely covered, the agent computes the point with minimum detection probability by neighboring sensors in the Voronoi cell as its moving target point. Otherwise the agent moves to the Voronoi center of an adjacent Voronoi cell and computes its moving target point in the same way. To our knowledge the target point with minimum detection probability is close to the farthest vertex of the Voronoi cell. After determining the target point, the agent moves along the direction from current location to the target point as farther as possible based on real-time communication feedback so that it can still have 2-connectivity with deployed sensors and ensures a connected covered area. In our paper, a kind of communication performance metric (namely, loop packet loss probability) is used to check whether two sensors remain connected with a good communication quality. The agents iteratively deploy new sensors based on this scheme until the whole area is fully covered. The check of complete coverage of the target area is also easy as the agents only need to check whether all the existing Voronoi cells in their memories are covered or not. When more than one autonomous agents are running the procedure, they need to communicate their own Voronoi partition and negotiate their movements. Both simulations under a noisy communication model and experiments using our self-made sensors are conducted to validate our proposed algorithm.

2. Preliminaries and models

In this section we introduce the background of our work, which includes Voronoi partition, sensor sensing model and sensor communication metric.

2.1. Voronoi partition

We introduce a little bit background on Voronoi diagram. Given k points in a plane, the plane is partitioned into k sub-regions according to the *nearest-neighbor-rule* [15] such that every sub-region called *Voronoi cell* is dominated by a point called *Voronoi center* which is closest to all the points in this sub-region. As an example, a Voronoi diagram is shown in Fig. 1. We assume that the target area is a polygon, so that each voronoi cell is also a polygon. In our proposed incremental deployment algorithm, we will

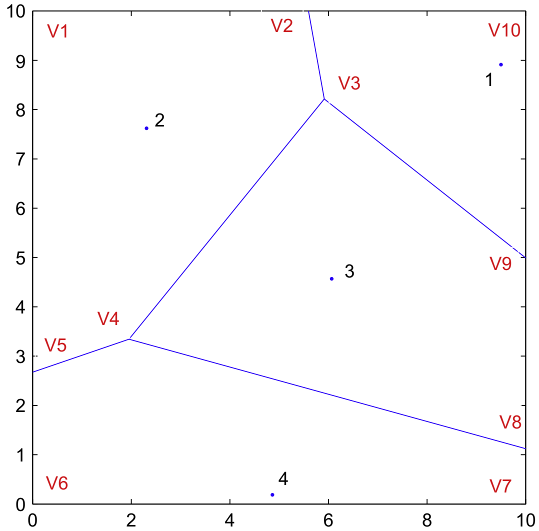


Fig. 1. A Voronoi partition with four points in a 10×10 area.

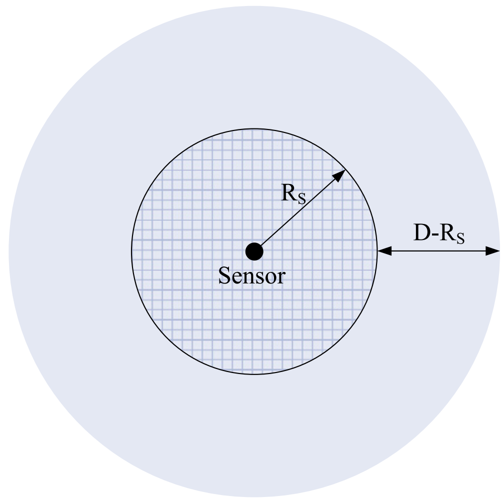


Fig. 2. The probabilistic detection model of a sensor.

apply Voronoi partition to find the farthest point in a Voronoi cell at each step.

2.2. Sensor sensing model

The sensing radius of all sensor is assumed to be identical, which is denoted by R_S. For a sensor s at (x_s, y_s) and any point p at (x_p, y_p), the Euclidean distance between them is denoted as $D(s,p)$. The binary sensor model that describes the detection rate of a sensor s at point p is given in the following equation (1) [16]

$$P_p(s) = \begin{cases} 1, & \text{if} \quad D(s,p) < R_S \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

However the binary sensing model is not accurate (e.g. the detection rate is impossible to vary sharply around the boundary of the sensing field). Therefore, the sensor detection model should be evaluated probabilistically[12,17]. Eq. (2) gives the probabilistic detection model of one sensor[6,12]. The sensing radius R_S expresses the detection field inside which a sensor is possible to detect a specific target without loss. And the parameter α is used to evaluated the quality of the sensor detection and the detection probability rate decreases with distance. Fig. 2 illustrates the probabilistic detection model of a sensor.

$$P_p(s) = \begin{cases} 1, & \text{if} \quad D(s,p) < R_S \\ e^{-\alpha \cdot (D - R_S)}, & \text{otherwise.} \end{cases} \tag{2}$$

We use the miss probability $M_p(s) = 1 - P_p(s)$ to evaluate the sensing performance indicating the probability of a point not being detected by sensor s. For the situation that several sensors surveil the same area, we make an implicit assumption that sensors detection are independent. Hence, we extend the probabilistic detection model of a sensor to the probabilistic sensing model of multiple sensors. Commonly, if there are more sensors near a point, it has lower rate not being detected. The miss probability for the target at point p is expressed in the following equation.

$$M_p = \prod_{i=p_1}^{p_n} M_p(s_i) \tag{3}$$

where $s_{p_1}, s_{p_2}, \ldots, s_{p_n}$ are the neighboring sensors which are close to point p. Generally, sensors far away from p make little contribution to target detection at p, and the target is mainly covered by neighboring sensors. Based on Voronoi partition, we only consider the closest sensor and adjacent sensors whose Voronoi cells are next to current Voronoi cell as neighboring sensors of point p. Taking Fig. 1 as an example, any point in the Voronoi cell dominated by sensor s_3 is supposed to have neighboring sensors, which means it is possibly covered by s_1, s_2, s_3, s_4. Similarly, any point in the cell dominated by sensor s_1 is

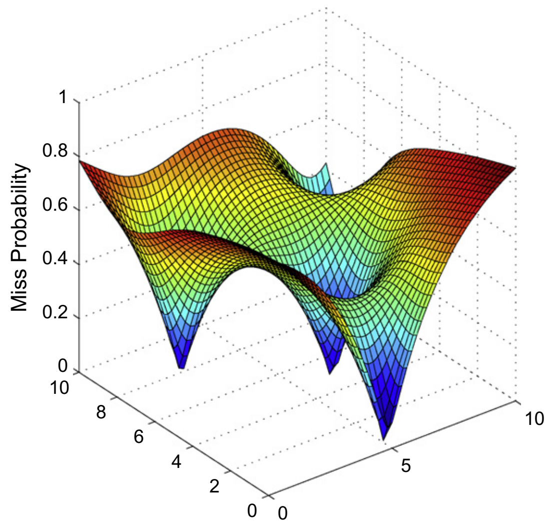


Fig. 3. The probabilistic sensing model of multiple sensors.

monitored by sensors s_1, s_2, s_3. Fig. 3 shows the miss rate distribution in the area with the setup in Fig. 1.

2.3. Sensor communication metric

In some early works it is assumed that communication range R_C of a sensor is much longer than its sensing range R_S [18], so that the communication model of a sensor is not an important issue. Some sensor communication models, which are similar with the sensing models such as binary model and probabilistic model, have been proposed [19,20]. However the communication quality between any two sensors may change drastically even with a small movement due to signal reflection, blocking, diffraction and other factors. On the other hand the communication quality may not decrease over distance with the same rate along any direction. Thus the binary model and probabilistic model are unable to state the communication situation in real environment perfectly.

In [14], the communication cost is evaluated as the expected numbers of retransmission. In this paper, we use a similar metric, the quality of communication connectivity between any two sensors is represented by the loop packet loss probability (*LPLP*) of their communication. In the case that packets are transmitted to the receiver and echoed back to the sender, the performance of the communication link between these two sensors can be measured by *LPLP*. Roughly, *LPLP* increases with distance, and reflects the transmitting power on both sides and disturbances from the surrounding environment. Sometimes the communication link between two sensors is asymmetric (e.g. one sensor can hear from the other sensor but the other sensor gets nothing from this sensor), and *LPLP* can address the quality of bidirectional communication. Moreover, *LPLP* does not depend on any radio signal estimation model, and hence it is robust. Thus *LPLP* is used as an evaluation metric of communication performance in the simulation and experiment.

We use $LPLP^*$ to indicate the predefined threshold of loop packet loss probability, meaning that the communication is not acceptable if the loop packet loss probability is beyond this value too much. Generally, when $LPLP^*$ is low, the communication cost is low and the distribution of sensors is dense. Thus the setting of $LPLP^*$ is a tradeoff between the communication cost and sensor cost. The setting of $LPLP^*$ depends on the requirement of real-time performance by the task of wireless sensor networks. When real time communication is the critical concern, $LPLP^*$ should be lower. When the cost of wireless sensor networks is the main concern (e.g. using less number of sensors to cover the same target area), $LPLP^*$ should be higher.

3. Deployment algorithm

In our setup, all sensors are carried and deployed by one or multiple autonomous agents. The agents determine the positions and drop sensors one-at-a-time such that the communication connectivity can be guaranteed, i.e., every sensor can talk with at least two other deployed sensors. We assume that the target area is a polygon and the agents can localize and navigate autonomously in the area. In the following, we discuss two separate cases: single agent and multiple agents, respectively.

3.1. Single agent

The incremental deployment algorithm presented here includes three phases: initialization, target point calculation and execution. In what follows, the following notations will be used. We use a to represent the autonomous agent and use s_i to represent the ith sensor to be deployed by the agent. $D(s_i, a)$ denotes the distance between s_i and the agent's current location. M_a denotes the miss rate at the agent's current location. Moreover, we let $LPLP_i$ denote the loop packet loss probability of radio communication between s_i and the sensor that is carried by the agent and is going to be deployed. As the coverage requirement, we use M^* to indicate the predefined threshold of miss rate, meaning that a target signal is impossible to be detected by surrounding sensors if the miss rate is much higher than this value. For the target area, since it is assumed to be a polygon, we denote its vertex set by $AV = \{v_1, v_2, \ldots, v_n\}$. A Voronoi cell dominated by s_i is denoted as $\mathscr{A}_i$.

Initialization. Generally, the first sensor can be deployed randomly as a starting point and it is not subject to the communication connectivity constraint. Nevertheless, some heuristics may help reduce the deployment time. Basically, if the initial position of the agent is inside the target area but far away from the boundary, the first sensor can be dropped at current position. Otherwise, the first sensor should be deployed close to the nearest vertex of the area so that it is covered properly by the first sensor. After deploying the first sensor, the agent selects the farthest vertex $v_i \in AV$ and moves in the direction towards it until one of the constraints in Eq. (4) violates.

$$\begin{cases} LPLP_1 < LPLP^* \\ M_a < M^* \end{cases} \tag{4}$$

Then the second sensor is deployed. The reason of using these two constraints is that they guarantee communication connectivity between two sensors and the area coverage.

Target point calculation. Based on k deployed sensors, a Voronoi partition is made for the target area AV by the agent with each Voronoi cell dominated by a deployed sensor $s_i, i = 1, \ldots, k$. Denote $V_i, i = 1, \ldots, k$, the set of vertices of Voronoi cell dominated by sensor s_i. An example is shown in Fig. 1. The target area is divided into 4 sub-regions by 4 deployed sensors, in which

$$V_1 = \{v_2, v_3, v_9, v_{10}\}$$
$$V_2 = \{v_1, v_5, v_4, v_3, v_2\}$$
$$V_3 = \{v_3, v_4, v_8, v_9\}$$
$$V_4 = \{v_4, v_5, v_6, v_7, v_8\}.$$

As the coverage criteria, $\mathscr{A}_i$ is completely covered if and only if $\max_{p \in \mathscr{A}_i}(M_p) < M^*$. In order to cover as more uncovered area as possible with less deploying time, a heuristic idea is that the agent should move along the direction from the location of sensor s_k (the one just deployed) toward the point with maximum miss rate inside

$\mathscr{A}_k$ if $\mathscr{A}_k$ is still not completely covered. Thus, whenever $\mathscr{A}_k$ is completely covered, the agent will consider an adjacent Voronoi cell $\mathscr{A}_{i_*}$ $(i_* < k)$ and check whether $\max_{p \in \mathscr{A}_{i_*}}(M_p) < M^*$ to see the coverage status of $\mathscr{A}_{i_*}$. If it is not completely covered, then the agent updates its location to the Voronoi center of $\mathscr{A}_{i_*}$ and selects the point with maximum miss rate as its moving target point. Otherwise, the agent continues considering another adjacent Voronoi cell. The algorithm stops when all the Voronoi cells are completely covered. Finding the target point p_* with maximum miss rate inside $\mathscr{A}_{i_*}$ is a constrained nonlinear optimization problem. Some methods[21,22] are useful for programming to solve this problem. However these methods need an initial estimate. According to our experiences, p_* usually appears close to the farthest vertex of the Voronoi cell. Therefore the vertex $v_* \in \arg\max_{v_j \in V_{i_*}}(D(s_{i_*}, v_j))$ is selected as the initial estimate. The pseudo code of the procedure is given below.

$i_* = k; I = \{1, 2, \ldots, k\}$;
while $I \neq \emptyset$ **do**
  if $\max_{p \in \mathscr{A}_{i_*}}(M_p) < M^*$ **then**
    $I = I \setminus \{i_*\}$;
    $i_* = i_* - 1$;
  else
    select $v_* \in \arg\max_{v_j \in V_{i_*}}(D(s_{i_*}, v_j))$ as an initial estimate;
    select $p_* \in \arg\max_{p \in \mathscr{A}_{i_*}}(M_p)$ as a target point;
    return i_*, p_*;
  end if
end while

Execution. The agent moves to the location of sensor s_{i_*} with the index i_* returned from the algorithm of target point calculation. Then it moves along the direction from s_{i_*} toward p_* also returned from the algorithm of target point calculation and runs communication test. The agent decides to stop and drops off a new sensor s_{k+1} when either M_a reaches the upperbound M^* or only two communication links are left and the loop packet loss probability of one of them reaches the threshold $LPLP^*$.

Two kinds of performance, namely, connectivity and coverage, are the important issues in this paper. Based on the incremental deployment scheme, any deployed sensor can talk to the first sensor directly or through other sensors. Thus the connectivity of the sensor network is guaranteed and all deployed sensors maintain at least two communication links with other sensors (2-connectivity). For the convergence of the algorithm, we assume that the target area consists of finite grid points. Once a sensor is deployed, one or more grid points are covered, i.e., the miss rates at those grid points are less than the threshold. Therefore, when finite sensors are deployed after finite iterations of the algorithm, all grid points are completely covered. As long as the amount of grid points in the target area is sufficient, the whole area is completely covered by the deployed sensors.

3.2. Multiple agents

Using more autonomous deploying agents clearly increases the deployment efficiency and shortens the time if they collaborate in an effective way. As agents can be more powerful than sensors, we assume that the communication range of agents are much larger than that of sensors. The agent records the locations of all deployed sensors in the single-agent case. However, an agent may not own the location information of some unknown sensors which are not deployed by itself. For example, as shown in Fig. 4, sensors $s_{21}, s_{22}, s_{23}, s_{24}$ are deployed by agent a and therefore these locations are known by agent a, while $s_{11}, s_{12}, s_{13}, s_{14}, s_{15}$ are not deployed by agent a and so these locations are not known by agent a temporally. We assume that all deployed sensors have their own location information, and once an unknown deployed sensor s_{12} is inside the communication range of agent a, agent a can obtain the location of this sensor. Moreover, through communication, agent a can get the locations of other sensors $s_{11}, s_{13}, s_{14}, s_{15}$ which are able to communicate with s_{12} in a single hop or multiple hops. Finally any agent can know the location of all deployed sensors based on this scheme.

It is assumed that multiple autonomous agents work asynchronously, meaning that they do not need to deploy a new sensor and have a movement at the same time. Intuitively, it is not expected that two or more agents take the same moving direction in order to avoid the situation that one repeats another's path. We know that for the algorithm of single agent, the agent actually searches the path starting from a deployed sensor s_i to the target point with maximum miss rate to find an location for deploying new sensor. For the multi-agent case, a new target point (new path) has to be given up by an agent if this path has already been taken by some other agents who is searching on it now. Thus, the agent moves to another Voronoi cell and selects another target point when possible. Hence, whenever an agent places a new sensor and decides its new moving

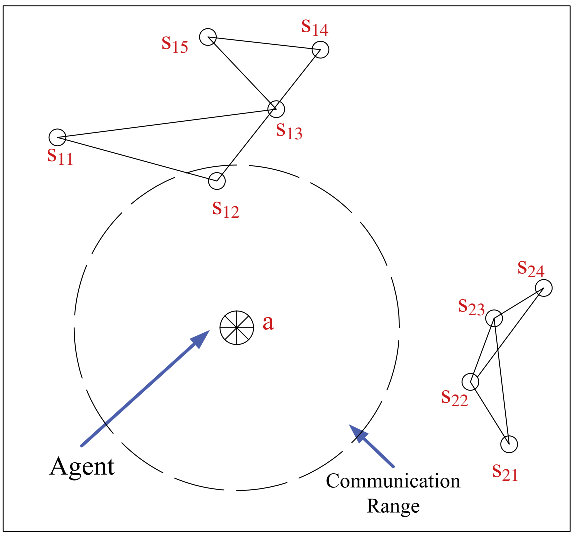


Fig. 4. An agent detects unknown sensors in multi-agent case.

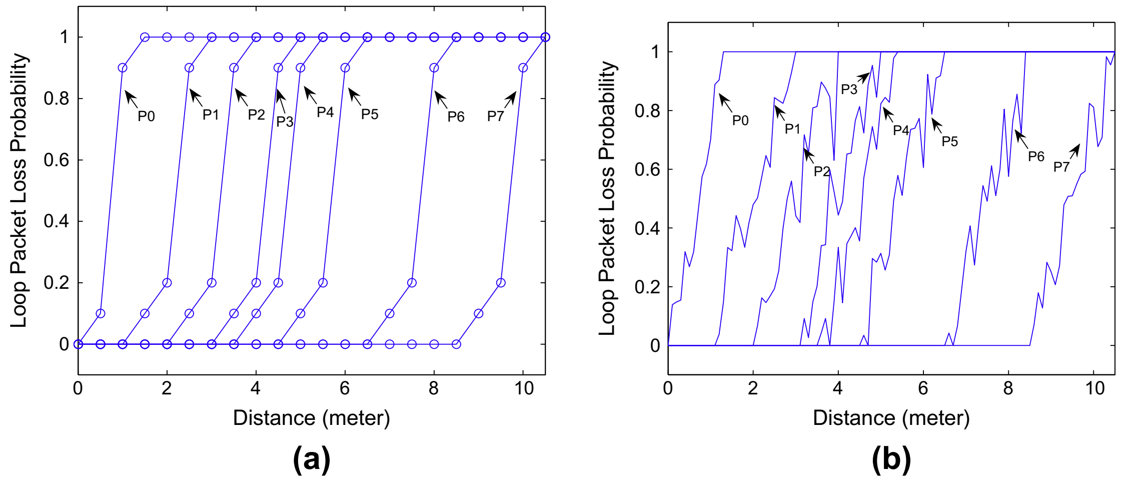


Fig. 5. (a) The radio signal model of our self-made sensors under eight different power levels, P0:$-25dBm$, P1:$-15dBm$, P2:$-10dBm$, P3:$-7dBm$, P4:$-5dBm$, P5:$-3dBm$, P6:$-1dBm$, and P7:$0dBm$. (b) The radio signal model with Gaussian noise.

direction, it should communicate the location of this new sensor and its new moving direction to its neighboring agents. The information will be used by neighboring agents to make decision on new movement after deploying a sensor. The information also prevents some agents from moving toward some very close target points.

In the multi-agent case, agents conduct the deployment process in parallel, which is the same as the case using single agent. The only difference is the algorithm of determining a new target point after deploying a sensor. The pseudo code of determining a target point by agent a_j is given below. The algorithm is run after a Voronoi partition is made based on all known deployed sensors whose locations are either localized by itself or received from other agents via communications. In the algorithm, the notation $p_t(t = 1, \ldots, n)$ is used to represent n target points toward which n neighboring agents are moving. D_T^* denotes the predefined threshold of distance between target points, which guarantees target points are not selected too close.

$i_* = k; I = \{1, 2, \ldots, k\};$
while $I \neq \emptyset$ **do**
  if $\max_{p \in \mathscr{A}_{i_*}}(M_p) < M^*$ **then**
    $I = I \setminus \{i\};$
    $i_* = i_* - 1;$
  else
    select $v_* \in \arg\max_{v_j \in V_{i_*}}(D(s_{i_*}, v_j))$ as an initial estimate;
    select $p_* \in \arg\max_{p \in \mathscr{A}_{i_*}}(M_p);$
    if $\min_{t=1}^{n}(D(p_*, p_t)) < D_T^*$ **then**
      $I = I \setminus \{i\};$
      $i_* = i_* - 1;$
    else
      return $i_*, p_*;$
    end if
  end if
end while

Once a target point p_* as well as its associated Voronoi cell dominated by sensor s_{i_*} are returned, agent a_j moves to the location of sensor s_{i_*} and searches the location for a new sensor on the path toward p_*, which is exactly the same as the case of single agent.

4. Simulation results

In practice, the algorithm proposed in Section 3 is applied for the deployment of sensor networks in a 20×20 meters target area. The communication quality between any two sensors may change drastically even with a small movement due to signal reflection, blocking, diffraction and other factors. In order to represent the character of the communication link, a radio signal model based on loop packet loss probability of our self-made sensors is built to simulate the communication environment. However, it should be pointed out that the radio signal model is only for communication environment simulation and our algorithm does not depend on any radio signal model. Then the simulation is conducted in two cases: single agent and two agents.

4.1. Model of radio signal

A radio signal model based on LPLP is used to simulate the communication environment. Fig. 5a shows a cluster of curves between LPLP and distance of two communicating sensors under different RF power levels in an outdoor environment. The RF power levels of two sensors may be different, and we assume that the LPLP between them depends on the sensor with lower RF power. In a real situation, the propagation of RF signals may be affected by the environment, which reflects, blocks or diffracts the signals. Thus, in our simulations, we add Gaussian noises $w \sim \mu(1, \sigma^2)$ to the signal model to reflect the environmental effects.

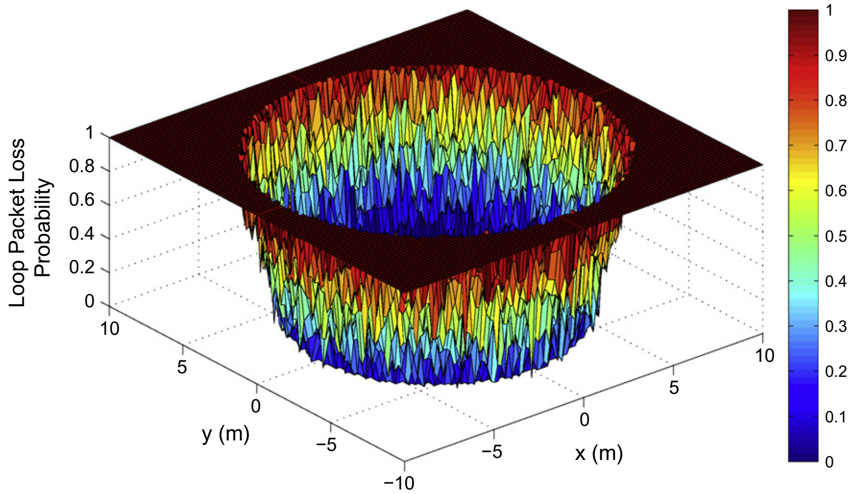


Fig. 6. The two-dimensional signal radio model when the output power is $-1dBm$.

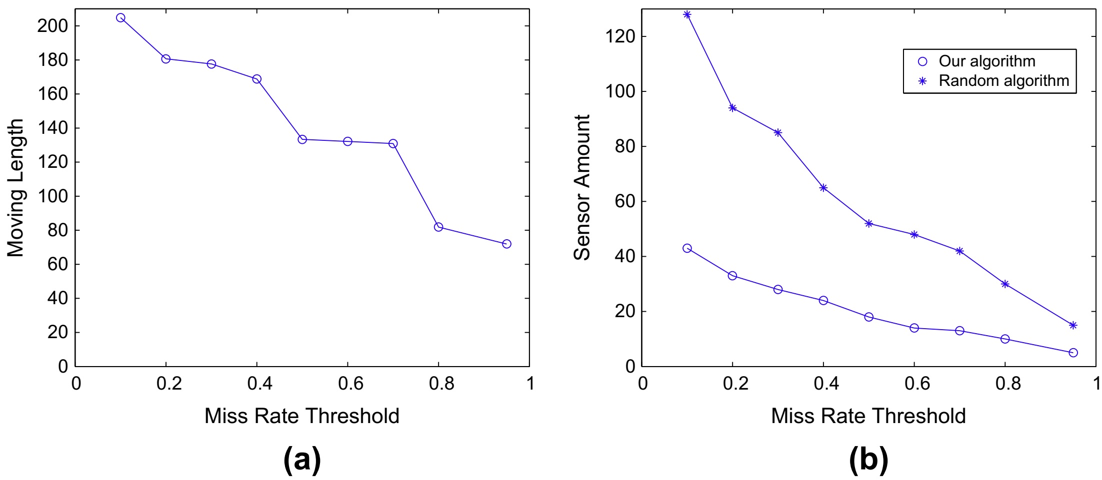


Fig. 7. (a) The moving length of the agent vs. the miss rate thresholds. (b) The amount of deployed sensors vs. the miss rate thresholds.

Fig. 5b depicts the radio signal model used in the simulation, whose variance σ^2 is 0.01. As an example, a two-dimensional radio signal model is presented in Fig. 6 with the RF power level being $P6(-1dBm)$, in which directional difference on communication is not medelled.

4.2. Single agent

Two cases of simulations are conducted to cover the whole area and guarantee 2-connectivity. In every case we compare the deployed sensor amount of our algorithm with the random deployment method. For the random deployment, we randomly deploy a few sensors in the target area and test the performance of connectivity and coverage. When the performance does not meet the requirement, we redeploy more sensors randomly and test the performance again. The sensor amount increases until the performance reaches the requirement.

In the first case, the RF power level is set to $-1dBm$, α is 0.5 and $LPLP^*$ is 0.2. The miss rate threshold M^* is under different values, which are 0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.95 respectively.

Moving length and sensor amount are considered as the performance metrics in the simulation. Fig. 7a shows the total moving length of the agent under different miss rate thresholds. Fig. 7b shows the amount of deployed sensors under different miss rate thresholds. The comparison result between our algorithm and the random deployment method is illustrated in Fig. 7b. We can find that the number of deployed sensors in our algorithm is much less than that in the random deployment algorithm under any miss rate threshold. We present the deployment results with Voronoi partition when M^* are 0.1,0.3,0.6,0.95 in Fig. 8. The distribution of sensors becomes sparser as anticipated when the miss rate threshold increases.

In the second case, α is 0.5, $LPLP^*$ is 0.2 and M^* is 0.5. The power level of each sensor is under different values, which

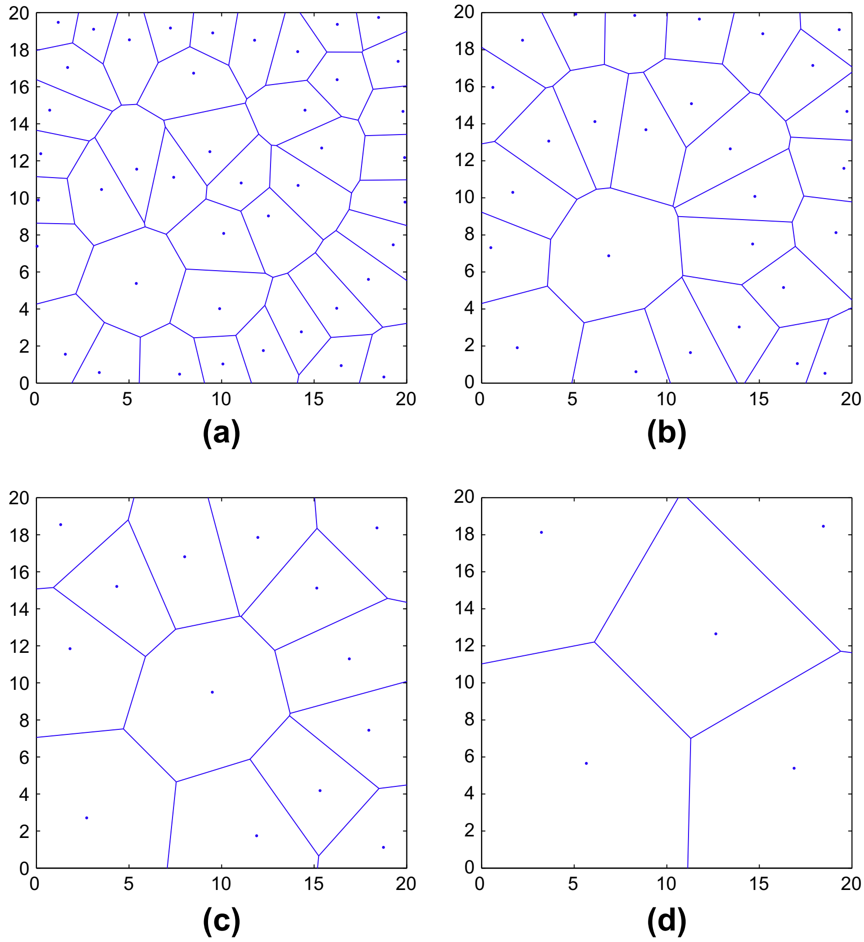


Fig. 8. Deployment results by one agent under different miss rate thresholds M^*. (a) 43 sensors are deployed when M^* is 0.1. (b) 28 sensors are deployed when M^* is 0.3. (c) 14 sensors are deployed when M^* is 0.6. (d) Five sensors are deployed when M^* is 0.95.

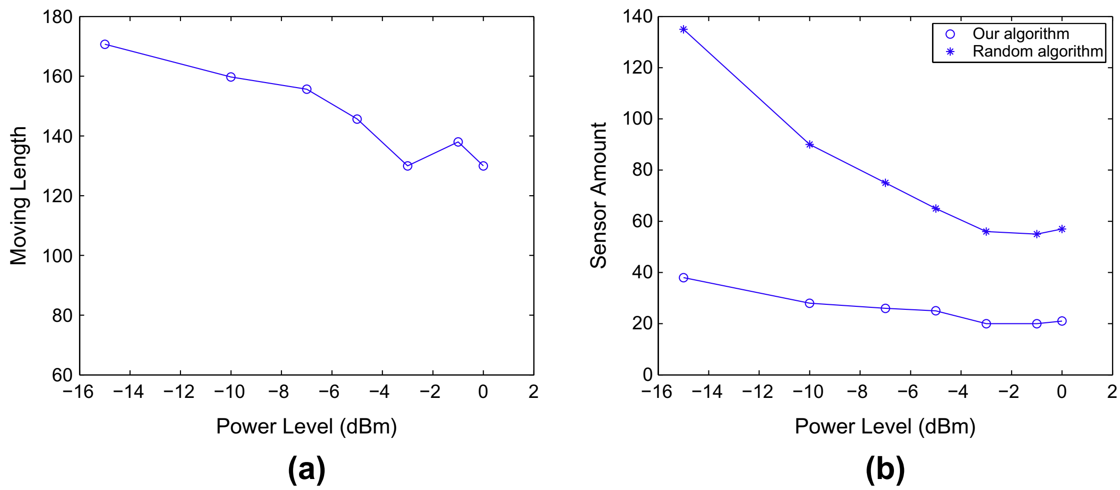


Fig. 9. (a) The moving length of the agent vs. the power level of sensors. (b) The amount of deployed sensors vs. the power level of sensors.

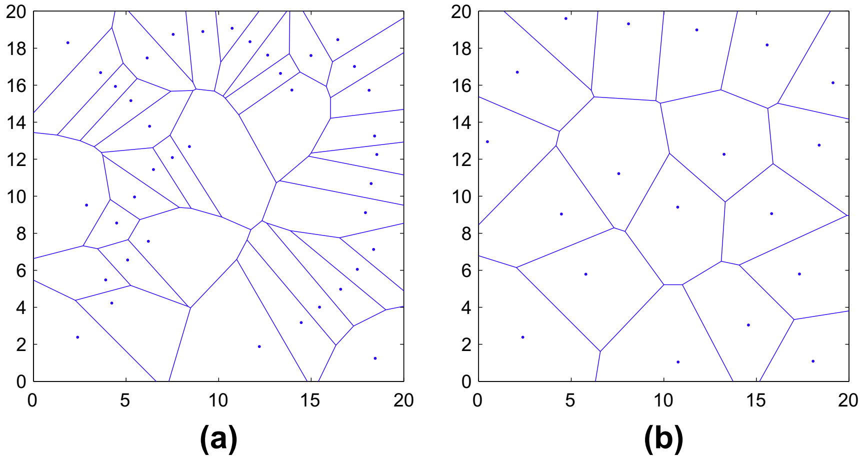


Fig. 10. Deployment results by one agent under different RF power levels. (a) 39 sensors are deployed when power level is $-15dBm$. (b) 19 sensors are deployed when power level is $-3dBm$.

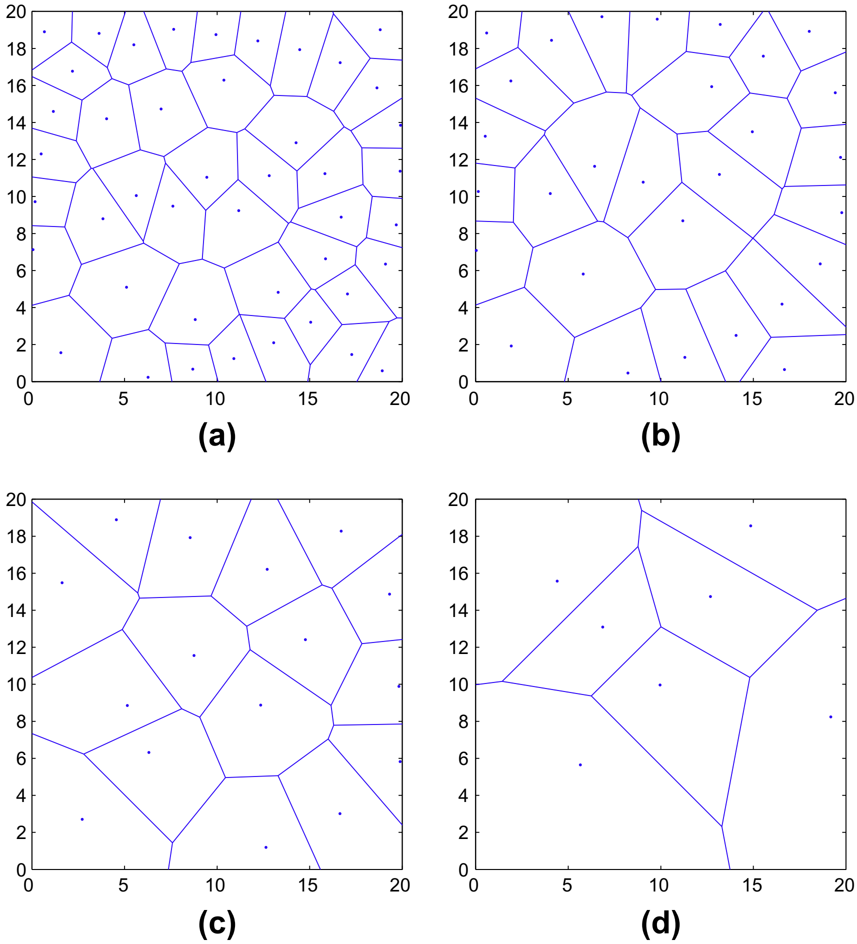


Fig. 11. Deployment results by two agents under different miss rate thresholds M^{*}. (a) 44 sensors are deployed when M^{*} is 0.1. (b) 29 sensors are deployed when M^{*} is 0.3. (c) 16 sensors are deployed when M^{*} is 0.6. (d) 7 sensors are deployed when M^{*} is 0.95.

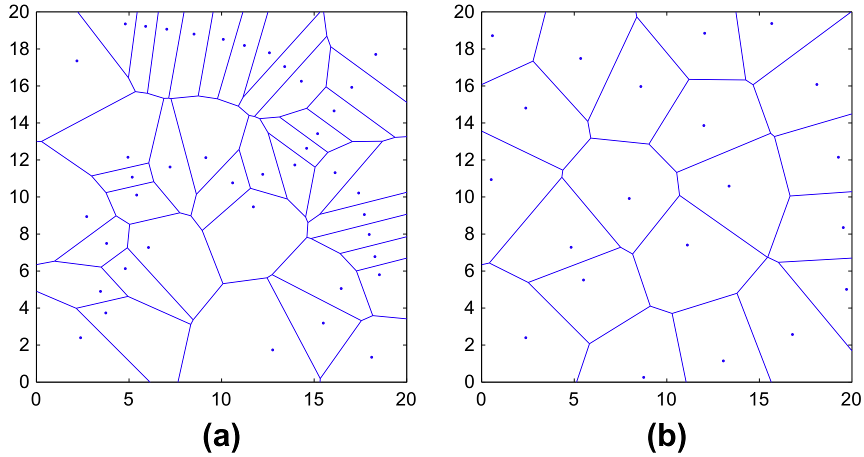


Fig. 12. Deployment results by two agents under different RF power levels. (a) 41 sensors are deployed when power level is $-15dBm$. (b) 21 sensors are deployed when power level is $-3dBm$.

are $-15dBm, -10dBm, -7dBm, -5dBm, -3dBm, -1dBm$, $0dBm$ respectively. Fig. 9a shows the total moving length of the agent under different RF power levels. Fig. 9b shows the amount of deployed sensors under different RF power levels. The comparison result between our algorithm and the random deployment method is shown in Fig. 9b too. We can find that the number of deployed sensors in our algorithm is much less than that in the random deployment algorithm under any RF power level. We present the deployment results with Voronoi partition when RF power level are $-15dBm, -3dBm$ in Fig. 10. The distribution of sensors becomes sparser with power level increases as anticipated. In Fig. 10a, the sensing range is larger than the communication range because of the low RF power level. Thus the communication connectivity is the dominating constraint and the area of each Voronoi cell is not uniform. However, since the RF power level is high in Fig. 10b, the miss probability threshold becomes the dominating constraint. Therefore the sensor deployment result approaches uniform.

4.3. Two agents

The simulation of two agents deploying sensors is similar to that of single-agent case. The only difference is that two agents execute the deployment algorithm in parallel. Again, two cases are conducted.

In the first case, the simulation parameters are almost the same as the single-agent case. The RF power level is $-1dBm$, α is 0.5, $LPLP^*$ is 0.2 and D_T^* is 1. Fig. 11 shows the deployment results with Voronoi partition when M^* are 0.1,0.3,0.6,0.95.

In the second case, α is 0.5, $LPLP^*$ is 0.2 and M^* is 0.5. The power level of each sensor is under different values, which are $-15dBm, -10dBm, -7dBm, -5dBm, -3dBm$, $-1dBm, 0dBm$ respectively. Fig. 12 shows the deployment results with Voronoi partition when RF power level are $-15dBm, -3dBm$. The sensing range is larger than the communication range because of the low RF power level in Fig. 12a. Thus the communication connectivity is the dominating constraint and the area of each Voronoi cell is not uniform. However, since the RF power level is high in Fig. 12b, the sensor miss probability threshold becomes the dominating constraint. Therefore the sensor deployment result approaches uniform. For the double-agent case, the amount of deployed sensors is similar with that in single-agent case, while the average moving length of each agent is reduced by nearly a half. The goal of completely covering the target area and maintaining 2-connectivity is achieved.

Fig. 13. Our self-made sensor.

Fig. 14. Experiment place.

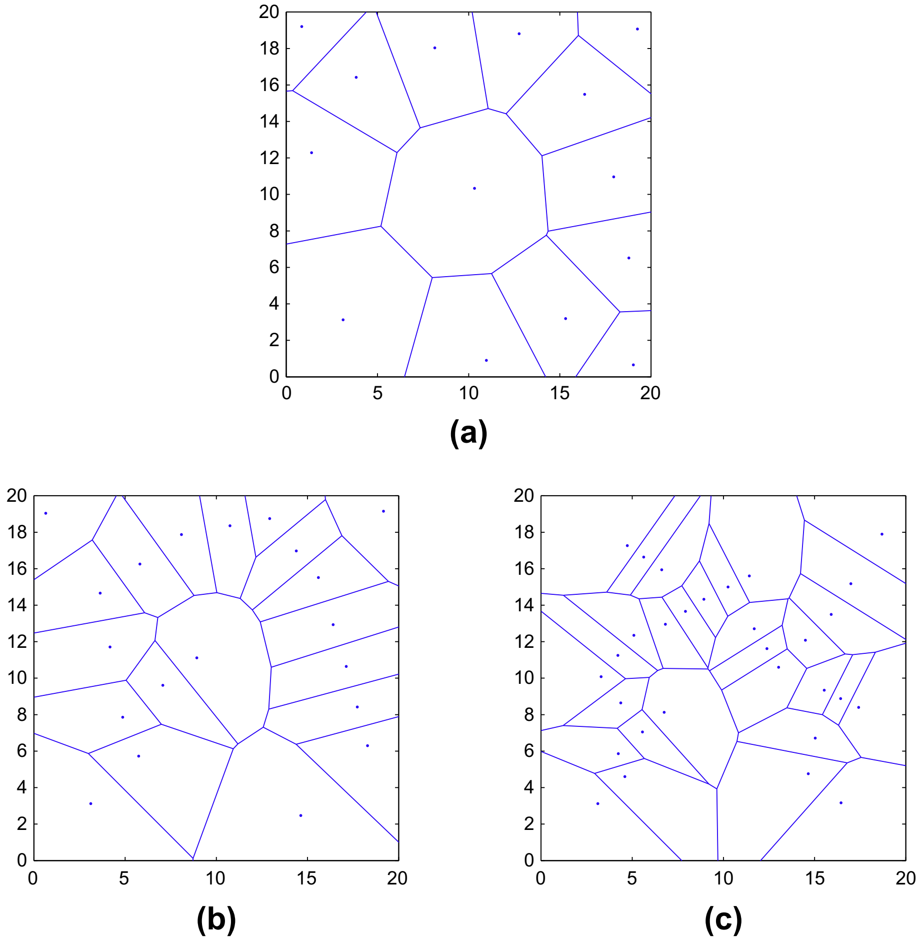


Fig. 15. Deployment results in the experiments. The target area is $20m \times 20m$. (a) 14 sensors are deployed when RF power is $0dBm$. (b) 20 sensors are deployed when RF power is $-5dBm$. (c) 30 sensors are deployed when RF power is $-15dBm$.

5. Experimental results

In order to verify the proposed algorithm, we designed several low-cost sensors and tested them in a 20×20 meters area. The deployment experiment on a lawn in the campus of Zhejiang University demonstrates the reliability and applicability of the deployment method.

5.1. Hardware design of sensors

It is designed that every sensor has a 2.4 GHz transceiver which supports the IEEE 802.15.4 radio. The sensor features a Texas Instruments CC2430 which is a true system-on chip solution for 2.4 GHz IEEE 802.15.4/Zigbee. We chose CC2430 because it combines the excellent CC2420 RF transceiver with an enhanced 8051 MCU which provides 128 KB flash, 8 KB RAM and many other powerful features (e.g. temperature sensor). A custom hardware board has been built to integrate a CC2430, a PCB antenna and USB data interface. A pair of AA batteries power each sensor. A picture of such a sensor is shown in Fig. 13.

5.2. Deploying experiment

We use our self-made sensors to do the experiment. As the main purpose of the experiment is to verify the proposed incremental deployment algorithm, a person takes the role of autonomous agent. When a sensor is placed on the ground, it is also marked on the map in laptop. The laptop computes the target point. Then a person holding a new sensor moves along the direction towards the target point. During the motion, the new sensor to be deployed consecutively tests *LPLP* with at least two deployed sensors. Once *LPLP* reaches the threshold $LPLP^*$ or M_a is beyond M^*, the new sensor is placed at the current location. The process continues until the whole area is covered. The experiment took place in a football field shown in Fig. 14 and the target area is $20m \times 20m$. There is no obstacles in the area, so the RF power becomes the main factor affecting communication quality.

Three experiments are conducted with the RF power being set $0dBm, -5dBm$ and $-15dBm$, respectively. The M^* is $0.7, LPLP^*$ is 0.2 and α is 0.5. Fig. 15 records the deployment results of these experiments. In Fig. 15a, the sensor miss probability threshold becomes the dominating constraint because the RF power level is high. Therefore the deployment result approaches uniform. However, in Fig. 15c, the communication connectivity is the dominating constraint and the area of each Voronoi cell is not uniform. Under these three different RF power levels, 14, 20 and 30 sensor are deployed eventually. The total distances traveled by the agents for deploying the sensor network in the three experiments are 145 m, 154 m, and 144 m, respectively. By this approach, it shows that though communication between sensors may be affected by the real-setting environment, the specifications of coverage and topology connectivity can still be satisfied.

6. Conclusions

In this paper, an incremental deployment algorithm is proposed to deploy a static sensor network by taking into account practical communication constraints and probabilistic sensor sensing model. The goal is to use possibly the least number of sensors to cover an area and maintain communication connectivity in a real setting environment. For this objective, deploying agents check how a real-setting environment affects the communication and deploy sensors accordingly. Voronoi partition is applied to help determine the moving direction during deploying. The probabilistic sensor sensing model is used to evaluate the area coverage and loop packet loss probability is used as a metric to evaluate communication quality. A bunch of sensors based on CC2430 have been designed and used in our deployment experiments. Both simulations and experiments demonstrate the success of the algorithm and applicability in practical situations. However, how to optimize the deploying path to reduce the total time has not been studied yet.

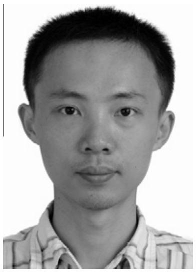
Acknowledgement

The work was supported by National Natural Science Foundation of China under Grant 60875074.

References

[1] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, J. Lees, Deploying a wireless sensor network on an active volcano, IEEE Internet Computing 10 (2) (2006) 18–25.
[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, Wireless sensor networks for habitat monitoring, in: Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications. ACM, 2002, pp. 88–97.
[3] J. Chen, E. Shen, Y. Sun, Information Technology Journal 8 (3) (2009) 293–301.
[4] N. Bulusu, D. Estrin, J. Heidemann, Adaptive beacon placement, in: Proceedings of the International Conference on Distributed, Computing Systems, 2001, pp. 489–498.
[5] A. Howard, M. Mataric, G. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, Distributed autonomous robotic systems 5 (2002) 299–308.
[6] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in: Proceedings of the Annual Joint Conference of the Computer and, Communications, 2003, pp. 1293–1303.
[7] A. Howard, M.J. Matarić, G.S. Sukhatme, An incremental self-deployment algorithm for mobile sensor networks, Autonomous Robots 13 (2) (2002) 113–126.
[8] B. Yamauchi, A frontier-based approach for autonomous exploration, in: Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation. IEEE, 1997, pp. 146–151.
[9] B. Wang, H. Lim, D. Ma, A survey of movement strategies for improving network coverage in wireless sensor networks, Computer Communications 32 (13) (2009) 1427–1436.
[10] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press Oxford, New York, NY, 1987. vol. 4.
[11] L. Vieira, M. Vieira, L. Beatriz, A. Loureiro, D. da Silva Junior, A. Fernandes, Efficient incremental sensor network deployment algorithm, in: Proceedings of Brazilian Symposium on Computer Networks, Gramado/RS, Brazil.
[12] S. Dhillon, K. Chakrabarty, Sensor placement for effective coverage and surveillance in distributed sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC03), New Orleans, LA, March 2003.
[13] Z. Yuan, R. Tan, G. Xing, C. Lu, Y. Chen, J. Wang, Fast sensor placement algorithms for fusion-based target detection, in: Proceedings of the Real-Time Systems Symposium. IEEE, 2008, pp. 103–112.
[14] A. Krause, C. Guestrin, A. Gupta, J. Kleinberg, Near-optimal sensor placements: Maximizing information while minimizing communication cost, in: Proceedings of the 5th International Conference on Information Processing in Sensor Networks. ACM, 2006, pp. 2–10.
[15] F. Aurenhammer, Voronoi diagrams: a survey of a fundamental geometric data structure, ACM Computing Surveys 23 (3) (1991) 345–405.
[16] K. Chakrabarty, S. Iyengar, H. Qi, E. Cho, Grid coverage for surveillance and target location in distributed sensor networks, IEEE Transactions on Computers 51 (12) (2002) 1448–1453.
[17] C. Wu, K. Lee, Y. Chung, A delaunay triangulation based method for wireless sensor network deployment, in: Proceedings of the 12th International Conference on Parallel and Distributed Systems. IEEE, 2006.
[18] M. Younis, K. Akkaya, Strategies and techniques for node placement in wireless sensor networks: a survey, Ad Hoc Networks 6 (4) (2008) 621–655.
[19] K. Kar, S. Banerjee et al., Node placement for connected coverage in sensor networks, in: Proceedings of the Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03), Sophia Antipolis, France, 2003.
[20] J. Bredin, E. Demaine, M. Hajiaghayi, D. Rus, Deploying sensor networks with guaranteed capacity and fault tolerance, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing. Urbana-Champaign, Illinois: ACM, 2005, pp. 309–319.
[21] R. Fletcher, M. Powell, A rapidly convergent descent method for minimization, The Computer Journal 6 (2) (1963) 163–168.
[22] D. Golfarb, A family of variable metric updates derived by variational means, Mathematics of Computing 24 (1970) 23–26.

Zhiyun Lin received his Ph.D. degree in electrical and computer engineering from University of Toronto, Canada, in 2005. From 2005 to 2007, he was a postdoctoral researcher in the Department of Electrical and Computer Engineering at University of Toronto, Canada. In 2007 he joined the College of Electrical Engineering at Zhejiang University, China, as a Professor. His recent research is concentrated on large scale networked multi-agent systems, wireless sensor networks, switched and hybrid systems, and feedback control of biped robots.



Sijian Zhang received the B.S. degree in electrical engineering from Zhejiang University, China in 2008. He is currently working towards the PhD degree in electrical engineering at Zhejiang University, China. His current research interests include sensors and sensor networks.

Gangfeng Yan received the B.S. and M.S. degrees in Control Theory and Control Engineering from Zhejiang University, China, in 1981 and 1984, respectively. He is currently a professor in the Department of Systems Science and Engineering, Zhejiang University. His research interests include hybrid systems, neural networks, and cooperative control.