Abstract: Mobile sensor has enabled many new applications in wireless sensor networks where coverage is one of the most important problems to support those applications. Previous research works of coverage issue mainly focus on the static sensor network, which can not handle the significant challenge brought by the node mobility. In this paper, we consider the coverage problem in mobile sensor networks under a very general framework, in which all the sensors are moving all the time with any specified movement model. We propose a scheme that can provide a Real-time Coverage-Map According to Probability (RC-MAP) of the mobile sensor networks. By introducing probability into coverage, RC-MAP provides three different types of coverage map in real-time, including probability to be covered, expected number of sensors to cover, and expected waiting time before being covered. Extensive simulation results demonstrate the high accuracy and effectiveness of the proposed scheme.

Keywords: probabilistic coverage, mobile sensor network, real-time coverage map

1. Introduction

The growing technique of mobile sensor has enabled many new sensor applications such as ZebraNet [1], CarTel [2] and CodeBlue [3], etc. It can be seen that for all the applications of mobile sensor networks, coverage is one of the most fundamental issues. Generally, coverage can be considered as the measure of quality of service of a sensor network [4]. The typical coverage problem can be described as follows: all sensors move uncontrollably in the region and the applications need a real-time coverage map of that region. The coverage map is a data matrix that every location in the region of sensor network has a value in the matrix to describe its coverage status (e.g., be covered or not, how many sensors cover it, how long will it wait until it is covered, etc.). One example in the scenario is that: in an ocean monitoring system (Fig. 1), sensors are deployed randomly on the sea to monitor the temperature, fish distribution, or the height of seabed. Due to the ocean wave, the sensors may float uncontrollably all the time and some coverage holes may appear or disappear dynamically. In this scenario, to get the knowledge of whole coverage situation of current networks, a real-time information map about coverage of those sensors is significant to the ocean monitoring system since the system can direct some yachts to fix those coverage holes basing on the coverage map.

Previous research works of coverage map are mainly focus on the static sensor networks [6][7][12]. The common idea is collecting position information from all the sensors and conducting analysis to calculate a coverage map. However, sensor mobility in mobile sensor networks brings significant challenge. Due to the sensor mobility, when the location information of a sensor is transmitted by multi-hops to the sink node, it may be outdated. This means in mobile sensor networks, the coverage map generalized basing on the outdated data transmitted by multi-hops is not real-time and not very accurate. Targeting at the above mentioned challenges, in this paper, our goal is to provide the real-time coverage map for a general framework that all the sensors are moving all the time with any specified movement model.

Figure 1. Yacht fills the coverage hole in sea monitoring.

In this paper, we introduce probability into coverage map in mobile sensor networks. Our basic idea is to leverage the node mobility to compute the probability of the position for the interested sensor nodes and generalize a real-time coverage map accordingly. We propose a scheme that contains a distributed protocol to collect information from individual sensors and a mathematical approach to generalize the Real-time Coverage-Map According to Probability (RC-MAP) of the sensor network. RC-MAP contains the basic descriptions of the coverage property: at any time instance, for each location in the region, we have the probability that it is to be covered by at least one sensor, the expected number of sensors to cover that location, and the expected waiting time before the location being covered. These basic descriptions of the coverage property in RC-MAP can fit the requirements of most sensor applications.

To the best of our knowledge, this is the first work that provides the real-time probabilistic coverage map for mobile sensor networks. In this paper, we propose a general platform to collect information and compute RC-MAP for mobile sensor networks. Our RC-MAP provides three basic descriptions of the coverage properties and many applications can use RC-MAP in their schemes.

The rest of the paper is organized as follows. We will introduce background and related works in Section 2. In Section 3, we will give our solution of RC-MAP: how to maintain history information table and leverage the mobility to compute the coverage probability. Section 4 gives a report on simulated experiments and analyzes the performance. We make a conclusion in Section 5.
2. Related Work

The concept of coverage is a measure of the quality of service (QoS) of the sensing functionality [4]. Due to different requirements and different sensing abilities, the coverage problem has different challenges. The first theoretical work on coverage problem of sensor networks was proposed by Meguerdichian, et al. [4]. It tried to use Voronoi diagram to describe the coverage problem in static sensor networks. Our target is different from [4] which assumes every sensor knows the coverage map, which is the eventually target in our work.

Different definition of quality of service of sensing functionality has brought new challenges. For example, the physical places are not desired to be covered by only one sensor but at least $K$ sensors so that the applications can perform sleep/wakeup scheduling to save energy. Focusing on the sensor deployment, Sasa Slijepcevic [5] presented the definition of SET $K$-COVER problem and proved that is NP-complete. An approximation solution has been proposed in [6]. These works focus on the deployment of sensors and they assume the mobile sensors are controllable in mobile sensor networks, which is quite different from our scenario.

Most works on sensor coverage use a simple sensing model [11], that is, an event that occurs within the sensing radius of a node is detected with probability 1 while any event outside this circle of influence is assumed not detected. Nadeem [11] broke this idealized model and proposed a more realistic sensing model. This is the first time that probability is introduced into coverage problem. However, their work still focuses on the static sensor network. In this paper, we consider probability introduced by the movement model in mobile sensor networks.

All the works mentioned above have not considered the impact of mobile sensors. Some research works tried to leverage the mobile sensor to enhance the coverage of whole network. Xiaojiang and Fengjing [8] proposed several schemes to exploit mobile sensors to improve the coverage, provide better routing performance and enhance the network connectivity. Andrew [9] focused on the mobile sensor deployment and presented a deployment scheme for mobile sensors using potential fields: the sensors start from some compact initial configuration and spread out to cover the whole region. However, these works [9][13] on mobile sensor networks assume that the mobile sensor can be controlled which is quite different from our work.

We realized that the mobile sensors may not be controlled easily or even can not be controlled in many scenarios. In this paper, we will propose a real-time coverage map of uncontrollable mobile sensors.

3. Our Solution: RC-MAP

In this section, we describe the Real-time Coverage-Map According to Probability (RC-MAP) algorithm and the corresponding protocol.

3.1. Motivation

Assume that every sensor is aware of its real-time location from GPS or some localization algorithm such as MCL [10]. Due to node mobility, after several hops of data-exchanging, the position information of a sensor transmitted from far away to the monitor sensor/sink could be outdated. A simple example is shown in Fig. 2.

In Fig. 2, the transmission and monitor range of a sensor is a $3\times3$ grid region. Sensor C reports its initial coordinate position $(1,3)$ at time 0 to sensor A with sensor B as a relay-node. Assume every one-hop transmission costs one time slot. At time 2, the position $(1,3)$ of sensor C will be gathered by sensor A finally. However, at this time, the real coordinate position of sensor C is $(1,1)$ but not $(1,3)$ anymore. The

![Figure 2. Outdated Location Information Due to Sensor Movement.](image)

Table 1. Information Table of Sensor B at Time 2

<table>
<thead>
<tr>
<th>Sensor_id</th>
<th>Position</th>
<th>Time_delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3, 4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>3, 2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1, 3</td>
<td>0</td>
</tr>
</tbody>
</table>

![Figure 3. Probabilistic Coverage Matrix Based on the Movement Constraint.](image)
outdated information brings a misunderstanding to sensor A that position (1,3) is covered by sensor C. It is the key challenge of providing a real-time coverage map in mobile sensor networks.

Fortunately, there are also some constraints of sensor movement (e.g., constraint of maximum velocity or directions). We can exploit these constraints to help decide the real position of sensor. We can know the probabilities of some places that the sensor may move to.

Take sensor B at time 0 in Fig. 2 as an example. Sensor B is in (2, 2) at time 0 and we assume a four direction movement model that one sensor can move to the neighbor grid in four directions or stay without moving in one time slot. It can be calculated easily that at time slot 2, the grid (4, 1) has a probability, 20\%, to be covered. The complete coverage map of probability to be covered is also shown in Fig. 3. Though the position information maybe outdated after aggregation, we can still use the history information and movement model to constrain the sensor’s coverage region in the shadow area by probability value. This is the core idea of our solution.

In the rest of this section, first we will introduce a protocol to exchange the position information among sensors and maintain an information table on each sensor, and then define the way to calculate the probability of sensor position. Finally we will present some approaches to calculate three different types of coverage map.

3.2. The protocol and information table

In RC-MAP, every sensor tries to collect the position information from all the other sensors even these information are several hops delayed. Each sensor maintains an information table which contains the position information related to all the sensors. Every data item in the table is stored as a vector \(<sensor_id, position, time_delay>\). At each time slot, sensors will update their information table:

1. Self-update
   
   Sensor \(m\) gets its current position \(S_m\) from GPS or some other localization algorithm and adds item \(<m, S_m, 0>\) into its information table. All the position information items about other sensors increase their “time_delay” in the vectors by 1 since one time slot has passed.

2. Exchange information table with neighbors
   
   The sensor \(m\) broadcast its information table to its neighbors (1-hop of communication range), receives its neighbors’ information tables and combines them with its own table together.

3. Eliminate of duplication
   
   After exchanging and combination, the information table of sensor \(m\) may contain duplicated items which have the same sensor_id. For the items that have the same sensor_id, sensor \(m\) keeps the one with smallest time_delay. An example is shown in Table 1.

3.3. The probability of sensor position

Due to the sensor mobility, the position of sensor varies overtime. There are two factors affect on the probability of sensor’s current position: one is the movement model and the other is the history information.

\[
g(s, u) = \begin{cases} 
1 & \text{distance}(s, u) \leq V_{\text{max}} \\
\frac{\pi(V_{\text{max}})^2}{\text{distance}(s, u)} & \text{otherwise}, 
\end{cases}
\]

For each time slot, a sensor can move within a certain range according to the certain movement model. We can use a function \(g(s, u)\) to represent the movement model. For example, assume the sensors are in the random movement model: each time the sensor can move from the current position \(s\) to any place \(u\) with in a maximum velocity limit \(V_{\text{max}}\) as shown in Fig. 4. Suppose all the places which have distances to \(S\) smaller than \(V_{\text{max}}\) will have the same probability to be moved to. The movement function is shown in Eq. (1-1).

![Figure 4. Different Movement Models.](image)

Different movement model has different movement function of \(g(s, u)\). For example, assume the sensors can only move according to the wind direction such as west-wind movement model as shown in Eq. (1-2):

\[
g(s, u) = \begin{cases} 
2 & \text{distance}(s, u) \leq V_{\text{max}} \\
\pi(V_{\text{max}})^2 & \text{& } u.x \geq s.x \\
0 & \text{otherwise}.
\end{cases}
\]

Using the movement function \(g(s, u)\), we can compute the probability that sensors may move to as time goes on since the probability of the movement destination at current time slot is based on the probability of the sensor position at previous time slot. Let \(f(s, u, t)\) be the probability that a sensor starts from position \(s\) and moves to position \(u\) after \(t\) time slots:

\[
f(s, u, t) = \int \int f(s, u', t-1) g(u', u) du' \quad t > 1
\]

Eq. (2) is computed by dynamic programming. We consider all possible positions \(u'\) of the sensor in the last time slot and sum the probabilities up. In the item vector \(<sensor_id, position, time_delay>\) of information table, we can get old position \(s\), time tag \(t\) for function \(f(s, u, t)\), so for any position \(u\) we can know the probability that the sensor is currently at \(u\). Thus, we can obtain a probability map of sensor distribution based on the information table.

We admit that leveraging precise movement model to improve the accuracy is not the main goal of our work. The up-layer applications can leverage the existing work of machine learning on the movement model such as HMM [14] to improve the accuracy of RC-MAP while we only focus on the platform scalability and our RC-MAP is suitable for different movement models.

3.4. The probabilistic coverage

Due to different applications have different coverage requirements, we provide three basic types of coverage maps.

3.4.1. Probability to be covered by at least one sensor

For a position \(x\) in the region and sensor \(i\) at location \(u\) with sensing range \(SR\), we can use function \(c(x, u)\) to demonstrate whether \(x\) can be monitored by sensor \(i\):
\[ c(x,u) = \begin{cases} 1 & \text{distance}(x,u) \leq SR \\ 0 & \text{distance}(x,u) > SR \end{cases} \] (3)

Equation (3) demonstrates a disk sensing model. But it is adaptable for different sensing model such as [11]. For the information item \(<i, s_o, t>\) from the information table, the probability of position \(x\) to be covered by sensor \(i\) can be calculated based on Eq. (2) and (3):

\[ p_i(x) = \int f(s_i, u, t) c(x,u) \, du \] (4)

According to Eq. (4), for position \(x\) in the region, the probability to be covered by at least one sensor is:

\[ p(x) = 1 - \prod_i (1 - p_i(x)) \] (5)

Based on Eq. (5), every position \(x\) in the region has a probability to be covered by at least one sensor. A coverage map (matrix of probability value) can be drawn.

### 3.4.2. The expected number of sensors to cover

For some applications, due to security or power related requirement, applications need the place of interest to be covered by at least \(k\) sensors which is also called \(k\)-coverage problem. In our RC-MAP, we could provide a coverage-map that every position \(x\) in the region has a number \(e(x)\) which is the expected number of sensors to cover that place. Based on basic probability theory, it is easy to achieve the expected number of sensors to cover as follows:

\[ e(x) = \sum_i p_i(x) \] (6)

### 3.4.3. The expected waiting time before being covered

We provide a simple way to predict the expected waiting time basing on the information table. Assume we have information table \(<i, s_o, t>\) currently. For a position \(x\) and sensor \(i\), we have the probability \(p(x)\) from Eq. (4) which is the probability that sensor \(i\) will monitor position \(x\) at the current time slot. Considering the worst case that at next time slot, we have no update on current information table so that the only change is \(t\) of every item in the table will increase by 1. For a position \(x\) and sensor \(i\), we will have \(p(x)\) which is the probability that sensor \(i\) will monitor position \(x\) at next time slot. We can also extend it to next \(j\) time slots:

\[ p^j(x) = \int f(s_i, u, t) c(x,u) \, du \] (7)

\[ p^j(x) = \int f(s_i, u, t + j) c(x,u) \, du \] (8)

Similarly to Eq. (5), for next \(j\) time slots, we can compute the probability \(p^j(x)\) of position \(x\) could be covered by at least one sensor show in Eq. (9):

\[ p^j(x) = 1 - \prod_j (1 - p^j(x)) \] (9)

Consider different \(j\), we can calculate the expected waiting time \(w(x)\) to be covered of position \(x\) as follows:

\[ w(x) = \sum_{i=1}^{x} \left( i \times p^j(x) \times \prod_{j=0}^{i-1} (1 - p^j(x)) \right) \] (10)

In our solution, we propose a protocol to collect all the information data even they maybe outdated. Based on the information table, we can calculate the probability of a sensor’s position using the pre-determined movement model. It will support us to analyze the probability of a position to be covered by a sensor and different types of coverage map.

### 4. Evaluation

The key metric for evaluating a coverage technique is the accuracy of the coverage estimates versus the communication overhead and movement constraints. For all of our experiments, 60 sensor nodes are randomly distributed in a 200mx200m rectangular region. We assume a fixed sensing range \(SR = 20\) and the communication range \(CR = 40\). We randomly pick a sensor and use its information table to compute RC-MAP. The network settings we vary are: Maximum Velocity and Communication Overhead(TTL).

#### 4.1. Visualization

In traditional coverage description such as 0/1 approach, we usually use 0 to represent the uncovered area and 1 stands for covered area. This coverage map can be easily turned to a black-white map (e.g. black region is the uncovered area and white region is the covered area). In RC-MAP, we use a gray-scale map to demonstrate the coverage situation (e.g. the lighter area has higher probability to be covered and the darker area has lower probability to be covered).

We have some interesting observations from Fig. 5: the gray-scaled map in Fig. 5 is generally similar to the standard answer—the black-white map; in black-white map, the most two big coverage holes at top and right bound of the region still keep big in gray-scale map. However, the small hole at right-bottom nearly disappears in the gray-scale map. That means RC-MAP can keep the big important coverage holes and may miss the small holes.

#### 4.2. Accuracy of RC-MAP

For every location \(x\) in the region, we can achieve a probability value \(p(x)\) to describe the coverage situation on this location. Compared with the real coverage situation \(q(x)\), we can define the estimated error \(\epsilon\):

\[ \epsilon^r = \frac{\sum_{x \in \text{region}} \left| (p(x) - q(x)) \times (D - \text{distance}(x,s)) \right|}{\sum_{x \in \text{region}} (D - \text{distance}(x,s))} \] (11)

In Fig. 6, the estimate error decreases in a startup phase. It is because sensors need to take some time for gathering the information from multi-hops. From Fig. 6, we notice that three types of RC-MAP have the same decreasing trend and they all converge finally. For convenience, we evaluate the impact of different parameter settings just for the first type of RC-MAP, the probability to be covered by at least one sensor, in all of the following evaluations.

#### 4.3. Impact of Maximum Velocity

In RC-MAP, the core idea is to leverage the node mobility. From Fig. 7, it illustrates that the estimate error grows higher while the maximum velocity becomes larger. It is because that with lower maximum velocity, we can predict the position of sensor in a smaller area with higher probability which could bring stronger constraints. Small maximum velocity can help us exploit the history position information well and in an extreme case that the maximum velocity is zero, which means a static sensor network, RC-MAP provides good accuracy but
estimate error is not zero since it considers the isolated nodes which are disconnected to the viewpoint as uncovered area.

4.4. Tradeoff between Estimate Error and Communication Overhead

As mentioned in Section 3, we set a TTL for every position item in the information table to discard the old data and avoid large overhead. Small TTL can reduce the size of information table and the corresponding overhead. However, discarding items in the information table by TTL may increase the estimated error. Fig. 8 gives a basic knowledge that the estimated error decreases when TTL increases. Small TTL such as 5 or 6 can support enough accuracy.

5. Conclusion

By introducing probability into coverage, in this paper, we exploit node mobility to provide a real-time coverage map for mobile sensor networks. Our core observation is coverage probability. We propose a platform that can provide Real-time Coverage Map According to Probability (RC-MAP). We provide a mathematical approach to compute the probability of sensor position and three types of coverage maps. The protocol requires low overhead and the probabilistic coverage map has a high accuracy that can fit most mobile applications. Based on RC-MAP, different upper-layer protocols can be implemented.

Acknowledgement

This research was supported in part by the 863 Program under Grant No. 2006AA012228, the NSFC Overseas Young Investigator Grant 60629203, by HKUST Nansha Research Fund, by Hong Kong RGC Grants HKUST 622407, N_HKUST609/07 and RPC06/07. EG05.

References