Region-based Sensor Selection for Wireless Sensor Networks

Yoshiyuki NAKAMURA*, Kenji TEI†, Yoshiaki FUKAZAWA*, and Shinichi HONIDEN‡

*Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
†National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
‡The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
{nakayosi, tei0820, fukazawa}@fuka.info.waseda.ac.jp, honiden@nii.ac.jp

Abstract

In a sensor network, the technique that limits the number of sensors used for observation is effective to reduce the energy consumption of each sensor. To limit the number of sensors without sacrificing observation accuracy, an appropriate sensor combination must be selected by evaluating the observation effectiveness of various combinations. However, the computational workload for evaluating all the sensor combinations is quite large. We can define a parameter related to the optimal size of a region around an observation target by making a trade-off between accuracy and the computational workload. In region-based sensor selection, a combination of sensors is selected from that is near the observation target. Accuracy is better in a larger region with a lot of sensors, but the computational workload is heavier. In contrast, a smaller region with fewer sensors has poorer accuracy, but a lighter workload. The size of the region helps to control the trade-off between accuracy and the computational workload. We define a parameter related to the optimal size of a region, and use it to dynamically adjust the region's size. Our simulations confirmed that region-based sensor selection reduces the computational workload and improves accuracy in comparison to existing techniques.

I. Introduction

The main purpose of a wireless sensor network is to measure a target’s state (e.g., its temperature, sound, and light). Since noise is included in the observations made by a sensor, the results always include some errors. Therefore, to improve accuracy, the results from two or more sensors should be integrated. In general, accuracy of an observation improves with the number of observation sensors. Sensors have severe energy consumption limitations. The lifetime of each sensor is shortened when it is used too often, and this in turn shortens the network lifetime. Thus, a technique that minimizes the number of sensors used for observations would reduce the total energy consumption. However, when only the number of sensors is reduced, the number of observation errors increases. Now, to minimize the number of sensors without sacrificing observation accuracy, it is necessary to select appropriate combination of sensors by evaluating their observation effectiveness. However, the computational workload would be huge, if the evaluation of every combination of sensors was necessary. It is unrealistic to evaluate all the possible sensor combinations, because the computational resources of a sensor (CPU and memory) are usually very restricted. Therefore, we propose a sensor selection technique in this paper that we call a region-based sensor selection that takes the trade-off between accuracy and the computational workload into account. In general, a sensor that is near the target is the most effective choice for making an observation, because observation noise increases with the distance from the target. In region-based sensor selection, a combination of sensors is selected near an observation target. Only the sensors within a region around the observation target are considered candidate sensors. The size of the region helps to control the trade-off between accuracy and the computational workload. When the region is large and a lot of sensors exist within it, accuracy increases but so does the computational workload. When the region is small, the computational workload decreases but so does accuracy. We define a parameter related to the optimal size of a region in this paper, which we then use to dynamically...
adjust a region size.

The remainder of this paper is organized as follows. Section II describes spread-based heuristics. Section III describes region-based sensor selection in more detail. Section IV presents our simulation results. Section V describes the related works on the topic of sensor selection, and Section VI concludes this paper.

II. Spread-based heuristics

This section describes the spread-based heuristics [1] and its shortcomings. Section II-A describes the evaluation metrics and Section II-B describes the method used to reduce the computational workload when evaluating the suggested metrics. Section II-C describes the shortcomings of the spread-based heuristics.

A. Evaluation metrics

The metrics proposed in reference [1] are simple and can be evaluated without using a complex calculation. Thus, they are suitable for the computation by a sensor. The spread-based heuristics comprise three metrics for selecting sensors: collinearity, spread, and proximity. These metrics are used to evaluate a target’s position by using a three-point measurement and they select a combination of sensors that form a shape that is nearly an equilateral triangle centered on the target’s position. The evaluation formulas are defined as follows.

\[
\text{Collinearity} : \Phi = \sum_{i=1}^{n} \left( y_i - mx_i - c \right)^2 \quad (1)
\]

\[
\text{Spread} : \Delta = \sqrt{\sum_{i=1}^{n} \left( \frac{2\pi}{n} - \alpha_i \right)^2} \quad (2)
\]

\[
\text{Proximity} : \Psi = \sum_{i=1}^{n} d_i \quad (3)
\]

In these formulas, \( n \) is the number of selected sensors and \( x_i, y_i \) are the coordinates of sensor \( i \), \( m \) is the slope and \( c \) is the y-intercept of a straight line to fit \( y = mx + c \) through all of the sensors. Therefore, the value of formula (1) is small if the sensors are almost collinear. In formula (2), \( \alpha_i \) is the angle incident upon the target from the two sensors. When \( n \) is 3, formula (2) evaluates the difference between the angle of the equilateral triangle and the actual angle. Therefore, the value of formula (2) gets smaller as the sensors get closer to forming an equilateral triangle. \( d_i \) in formula (3) is the distance between sensor \( i \) and a target. Therefore, as this distance shortens, the value of formula (3) gets smaller. These three evaluation metrics are used to select the combination of sensors.

B. Reducing the computational workload

The metrics proposed in spread-based heuristics can be calculated without complex processing. However, the computational workload becomes huge if it evaluates them to the all sensor combinations. Spread-based heuristics reduces the computational workload by decreasing the number of selected sensors. For each selection, only one sensor that seems to be ineffective for the observation is eliminated from the combination, and a new one is selected from the other sensors in the network. Let the total number of sensors be \( N \) and the number of selected sensors be \( n \). Then the computational workload to select the sensors becomes \( O(N^n) \). With the technique mentioned in [1], the computational workload can be reduced to \( O(N) \).

C. Problem with spread-based heuristics

The sensors that remain in the combination are selected according to the position of the target in the past. Therefore, accuracy might decrease if the selected sensors are ineffective. If the target’s speed is high, this shortcoming is significant. When the target’s speed is high, the sensors that remain in the combination that are far from the target and the error included in the observation result increase.

![Fig. 1. Overview of spread-based heuristics](image)

An example of such a situation is shown in Fig. 1, where a target’s position \( T(k) \) is observed by sensors \( s_1, s_2 \) and \( s_3 \) at time \( k \), and the sensors chosen to observe \( T(k+1) \) are selected by the predicted target’s position \( T'(k+1) \). With the technique mentioned in [1], sensor \( s_3 \) is removed from the combination and sensor \( s_4 \) is then selected. Sensors \( s_1, s_2, \) and \( s_4 \) are then used to observe the target’s position \( T'(k+1) \). However, in this case, the combination of sensors \( s_2, s_4 \) and \( s_5 \) is more suitable according to the metrics described in Section II-A. Therefore, the appropriate sensors are not selected and accuracy of the target’s position decreases. This problem occurs because \( s_1 \) and \( s_2 \) are not selected according to the present target’s position.
III. Region-based sensor selection

Region-based sensor selection is a sensor selection technique that makes a trade-off between accuracy and the computational workload. In this scheme, all the sensors are selected for each selection. In this case, the increase in the amount of processing becomes a problem. Therefore, the sensors chosen to observe a target are selected from the candidate sensors that are nearest to the observation target. The candidate sensors are limited to a circular area called a region. The computational workload of the sensor selection is substantially reduced by evaluating the effectiveness of only the candidate sensors. Moreover, energy consumption is reduced by limiting the number of sensors related to the sensor selection processing. Section III-A describes the procedure of region-based sensor selection. Section III-B describes the method of dynamically adjusting the region size.

A. Flow of sensor selection

The behaviors of the sensors are shown in Fig. 3. Each sensor can take a leader role or a regular sensor role. At time $k$, the leader sensor selects the other necessary sensors, estimates the target’s position $T’(k+1)$, and predicts the target’s position $T’(k + 1)$. Since the leader sensor always collects the positions of the candidate sensors, it does not need to know all the sensor positions. This makes it easy to account for breakdowns and the addition of necessary sensors to the network. Moreover, there are no special sensors, so sensors can easily exchange roles. In our simulations, the leader sensor at time $k + 1$ is one of the sensors selected at time $k$.

Fig. 3. Flow of sensor selection

The first thing the leader sensor does is to broadcast the estimated target’s position $(T_x, T_y)$ and the region’s radius ($r$). Each sensor replies to the leader sensor with its own position $(x, y)$ if it is within the region. If it is not within the region, a sensor returns to the stand-by mode without replying. After a fixed period, the leader sensor evaluates the combinations of the responding sensors and selects a combination to observe the target. After selecting the sensors, the leader sensor broadcasts the selected sensors’ IDs. The selected sensors observe the target and send their observation results back to the leader sensor. The leader sensor estimates the target’s position from the observation results and predicts the target’s position. This sensor selection procedure is then repeated.

Since our technique selects sensors from around the target’s position, the metrics, which evaluate each sensor’s position and the target’s position, are suitable. For this research, we used the evaluation metrics proposed by spread-based heuristic (Section II). Let the sets of sensor combinations within a region be $S$, and the threshold of $\Phi$ be $\Phi_{thr}$. $\Phi_{thr}$ is an average of $\Phi$. First, the combinations with $\Phi > \Phi_{thr}$ are removed from $S$. Then the combination with the least $\Delta$ is selected for the observation. The selected sensors form a shape similar to an equilateral triangle centered on the target’s position.

B. Region Setting

An appropriate region size is decided by the simulation. TOSSIM [2] was used as the simulator. Simulations were carried out for 140 sensors randomly placed within $200m \times 200m$ area. The number of target is one, and the target’s movement model was a straight line or a circle. The region radius $r$ took on values from $\{30, 40, 50, 60 and 70\}$, the
target’s speed $v$ took on values from $\{1, 5, 10, 15 \text{ and } 20\}$, and we took the averages from 100 simulations. A movement history was needed to predict the moving target’s position. Therefore, it was assumed that it was possible to accurately pursue the target and to maintain a correct movement history. The AVERAGE heuristic proposed by reference [3] was used to predict the target’s position. In this heuristic, it is assumed that the distance and direction of a target’s movement is the mean value of the past movement.

The results from the simulation are shown in Figs. 4 and 5, which specifically show results for straight line and circle target movement models, respectively. In these cases, the number of errors increases not only when the region is small, but also when the region is too large. The sensors were first evaluated according to the distance from the target, because our technique selects sensors from the region. When the region is large, the sensors are selected by evaluating their arrangement (collinearity and spread). Therefore, the number of errors included in the observation results increase because a sensor that was far from the target was selected. Therefore, the region only needs to be small for an appropriate size to exist.

Figs. 4 and 5 also show that the target’s speed influences the appropriate region size. If the target is moving fast, even a small prediction error becomes significant. The number of observation errors increases when selecting sensors from a small region. Therefore, the appropriate region size increases as the target’s speed increases. However, if the region size is decided only based on the target’s speed, when it is slow moving it becomes very small. In that case, there are not enough sensors within a region and an appropriate arrangement of the necessary sensors can not be found. Therefore, a minimum size is needed. This is decided by the number of sensor within the region. However, a sensor is sometimes added or deleted in wireless sensor networks, and the density of the sensor is not constant. Therefore, the density of sensors at time $k+1$ is unknown. Since the target’s position at time $k$ will be close to the one at $k+1$, the densities of the sensors nearest the target will be close to being the same at these times. Thus, by letting the density of the region at time $k+1$ be $d_{k+1}$, we can approximate $d_{k+1}$ by using $d_k$, i.e., $d_{k+1} \approx d_k$. The simulation results indicated that the appropriate region size is related to the target’s speed and sensor density. Thus, we can calculate the region radius $r_{k+1}$ in the following way:

$$r_{k+1} = \alpha v_{k+1} + \frac{\beta}{\pi d_{k+1}} \quad (4)$$

where, $v_{k+1}$ is the target’s speed at time $k+1$, and $\alpha$ and $\beta$ are constants. $v_{k+1}$ can be calculated using the target’s moving history, and $d_{k+1}$ can be approximated by $d_k$. Three sensors or more are needed within the region so that our technique evaluates the position by three point measurement. Thus, $\beta = 3$ and $\alpha = 2$ from Figs. 4 and 5.

We can calculate the region radius $r_{k+1}$ as follows

$$r_{k+1} = 2v_{k+1} + \frac{3}{\pi d_k} \quad (5)$$

IV. Performance Evaluation

In this section, accuracy, computational workload, and energy consumption of region-based sensor selection are evaluated. The simulation setting and the prediction technique are the same as that described in Section III-B. Given the above settings, we evaluated the number of errors when the total number of sensors $N$ and the region radius $r$ are varied. $N$ took on the values from $\{100, 120, 140, 160 \text{ and } 180\}$, and we averaged the values from 100 simulations.

The errors in the actual target’s position and the estimated target’s position were used for the evaluation metrics of accuracy. The processing time for the sensor selection was used as an evaluation metric of the computational workload. PowerTOSSIM [4] was used as a simulator for
TABLE I. Energy consumption

<table>
<thead>
<tr>
<th>$v$ (m/s)</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>spread-based [mA]</td>
<td>297</td>
<td>301</td>
<td>287</td>
<td>308</td>
<td>303</td>
</tr>
<tr>
<td>region-based [mA]</td>
<td>110</td>
<td>139</td>
<td>186</td>
<td>239</td>
<td>285</td>
</tr>
</tbody>
</table>

TABLE II. Processing time

<table>
<thead>
<tr>
<th>$N$</th>
<th>100</th>
<th>120</th>
<th>140</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>select 3 sensors [s]</td>
<td>3.80</td>
<td>10.70</td>
<td>20.21</td>
<td>36.52</td>
</tr>
<tr>
<td>region-based [s]</td>
<td>0.84</td>
<td>1.21</td>
<td>1.35</td>
<td>1.72</td>
</tr>
<tr>
<td>spread-based [s]</td>
<td>0.81</td>
<td>0.85</td>
<td>0.90</td>
<td>0.96</td>
</tr>
</tbody>
</table>

the energy consumption evaluation. It is difficult to measure the exact processing time of a simulator. Therefore, we used a real sensor to measure the processing time. The sensor was a MICAz [5] made by the Crossbow company.

Table I compares the energy consumptions of region-based sensor selection and the spread-based heuristics [1] in the processing shown in Fig. 3. The energy consumption was influenced by the number of sensors related to the sensor selection. Since region-based sensor selection changes the region size to correspond to the target’s speed, the target’s speed changed and $N$ fixed at 140. The spread-based heuristics has constant energy consumption because it did not take into account the target’s speed. In contrast, compared to the spread-based heuristics, region-based sensor selection reduced the energy consumption by 35% for $v = 10$. This is because the number of sensors for the sensor selection is small, but the computational workload is large.

Table II shows the processing times, including those of a technique that selects from all of the sensors. Region-based sensor selection needs a processing time of about 9% of existing techniques for $N = 140$. However, it takes longer than the spread-based heuristics. This is because the computational workload of region-based selection is $O(N^2)$, whereas the computational workload of the spread-based heuristics is $O(N)$. However, because region-based selection reduced $N$, the increase in processing time is small. The combination of sensors can be selected before the target actually moves, because the sensor is selected on the basis of the predicted target’s position. Therefore, the influence of the processing time is small when the target is observed at intervals greater than the processing time. The processing time might influence the energy consumption. However, Table I shows that region-based selection reduces the energy consumption, because it uses only a few sensors for the sensor selection processing.

Figs. 6 and 7 compare the errors of region-based sensor selection and the spread-based heuristics [1]. Fig. 6 shows the relationship between $N$ and the average error when the target’s speed is assumed to be 10 and Fig. 7 shows the relation between $v$ and the average error when the number of sensors is assumed to be 140. In Fig. 6 we can see that region-based sensor selection reduces the number of errors by 45% when compared to the spread-based heuristics. This proves that it is able to select an effective combination for target observation. The error cannot be decreased even if there are a lot of sensors, because the spread-based heuristics selects only one sensor again. Similarly, the error increases in spread-based heuristics as the target’s speed increases. Since our technique selects all the sensors again according to the moving destination of the target, the increase in error due to the target’s high speed can be decreased. However, the error increases when the target moves at a slow speed, so the error of a positional prediction may increase. Region-based sensor selection becomes more effective as the target moves faster. Since the spread-based selection selects only one sensor, the sensors that remain in the combination may quickly become distant from the target. Therefore, the error included in the observation result increases.

The simulation shows that region-based sensor selection improves accuracy and reduces the energy consumption. Since our technique changes the region size in response to the target’s speed changes, it becomes more effective as the target moves faster.
V. Related works

Previous works [6], [7] proposed evaluation metrics for sensor selection. The evaluation metrics proposed in [7] involve a lot of computation calculations and are not suitable when a sensor calculates them. References [8], [9], [10], [11], and [12] researched other sensor selections. References [9] and [11] assumed that a sensor was a directional sensor, like a camera, and these approaches are not suitable for sensors that observe at a distance from the target. References [8] and [12] assumed that a sensor has the ability to move, and these approaches are not suitable for common sensors that have no mobility. Reference [10] took into consideration the reduction of information due to communication delays. A sensor was selected based on the present target’s position. However, because the actual target’s position was uncertain, it was necessary to predict the target’s position from past movements. References [3], [13], and [14] proposed heuristics to predict a target’s position. The AVERAGE heuristics proposed in reference [3] and [13] were used for simulation purposes in Section IV. Reference [15] was another approach that reduced the energy consumption of a sensor. In reference [15], the energy consumption was reduced by changing the radio strength according to the distance from the receiver, because the energy consumption was larger when the data was transmitted.

VI. Conclusion and future work

We proposed region-based sensor selection, which takes into account the trade-off between accuracy and the computational workload. Region-based sensor selection selects an appropriate combination of sensors for a target observation. The computational workload is reduced by selecting only the candidate sensors that are within the region around the target. We showed that the number of sensors within the region and the target’s speed are related to the trade-off between accuracy and the computational workload, and therefore, we proposed a method to dynamically adjust the region. The simulation results showed that region-based sensor selection reduces the computational workload and energy consumption in comparison with the spread-based heuristics, and its accuracy improves even more when the target moves slowly. In region-based sensor selection, sensors are selected by referring to the predicted target’s position. Therefore, if the predicted target’s position differs from the actual target’s position, the selected sensors will not be effective enough for an observation. The AVERAGE heuristics predicts the target’s position from a target’s past movement. Therefore, when a target moves at random and a target’s velocity greatly changes, region-based sensor selection will be ineffective. Finding a way to reduce the influence of prediction accuracy will be a future work. One approach is to make multiple predictions and to select the best sensor combination. In that case, it will be necessary to reconsider the trade-off between observation accuracy and the computational workload and the power consumption.

References