Self-organization Strategies for Dynamic Context Coverage in Capability-Constrained Mobile Sensor Networks

Shiow-yang Wu, Chao-Hong Liu, Chen-Kuang Tzeng
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan, R. O. C.
showyang@mail.ndhu.edu.tw
Tel: +886-3-8634020 Fax: +886-3-8634010

Abstract

We propose and formally characterize a new problem named the dynamic context coverage problem in capability-constrained mobile sensor network environments. The goal is to move and adjust a network of mobile sensors with limited sensing capabilities to quickly achieve good coverage of dynamic changing contexts and continuously maintain the level of coverage. We propose several self-organization strategies and algorithms to solve the problem. Simulation results demonstrate that our techniques are highly effective in terms of dynamic context coverage, moving distance and self-organized deployment patterns.

Keywords: mobile sensor networks, heterogeneous sensors, dynamic context coverage, self-organization

1. Introduction

The advances in wireless communication and microelectronic technologies have made feasible a new promising area of research on wireless sensor networks [1, 2]. The coverage problem has been identified as one of the most important issue [8, 9, 10]. However, most existing works focus on static field, uniform context, homogeneous sensors and area-based coverage. For the purposes of our target applications such as environment monitoring, ecological protection, especially disaster search and rescue, we propose and formally characterize a new problem named *dynamic context coverage problem* in capability-constrained mobile sensor networks. The main goal is to quickly react to context intensity and environment changes, and automatically redeploy capability-constrained mobile sensors to continuously maintain maximal coverage. The problem is unique in several respects. First of all, the sensors are capability-constrained and therefore can only provide partial coverage limited by its sensing capability, not just sensing area [8, 13]. The same area may need more than one sensors to cover. Secondly, the context is non-uniform such that the intensity may vary in different areas. High intensity areas need more sensors to cover. Some areas may need no sensor at all if the intensity is too low to be considered significant. Most importantly, the contexts can change. Therefore the key issue is to cope with dynamism and continuously maintain coverage, rather than just the initial deployment or one time relocation [3, 11, 14, 16].

Our goals are as follows: (1) Based on local information, sensors can self-organize to cover the contexts as much as possible under capability constraints. (2) On detecting changes, leaks or failures, sensors can quickly self-adjust to respond to the dynamics. (3) When the contexts stay unchanged, sensors should stabilize with the best topology that consumes minimal energy to maintain longest lifetime. (4) The sensors should strike for the optimal balance on multiple context coverage. (5) Always operate with energy efficiency in mind. Only completely self-organized solutions are considered valid. Any reliance on global information such as the total number of sensors [13] is not allowed. Cluster-based or grid-based methods [4, 14, 16] are not appropriate either since the contexts can change rapidly.

We formally characterize the problem in Section 3, then propose three self-organization strategies and five algorithms accordingly in Section 4. Implementation and simulation results in Section 5 demonstrate that our techniques are highly effective in terms of dynamic context coverage, moving distance and sensor deployment patterns. Section 6 concludes the paper.

2. Related Work

The research on sensor networks have been growing in a very fast pace [1, 2]. Coverage problem has been attracting much attention [8, 9, 10]. In [5], a greedy algorithm is proposed to select connected sensors that fully cover the query area. The work in [12] employs the crossing angles of the communication ranges to count the coverage. Voronoi diagrams have been used extensively [11, 13]. Detail analysis of the intersection of communication range with nearby sensors can also be used on coverage [8]. There have been some efforts on mobile sensors for coverage problem. A deployment strategy to move sensors one at a time is proposed in [6]. Then *potential fields* is used to reduce deployment time [7]. In [15], a cluster-based virtual force algorithm is proposed to enhance the coverage after a random initial deployment. In [13], three movement-assisted protocols are proposed based on Voronoi diagrams.

To the best of our knowledge, the primary concern of previous works are centered around the geographical coverage of static and uniform context using homogeneous sensors with no capability constraint besides communication range and power. We consider a much more complex problem of multiple context coverage with heterogeneous and capability-constrained mobile sensors. Furthermore, most existing methods rely on various degree of global information such as field size, total number of sensors, potential fields, clusters, grid, etc. For applications in harsh environments such as disaster recovery and battle field, this is not desirable or even impossible. We are interested in completely self-organized solutions that rely only on local information. Because of the fundamental differences in the problem and allowable solutions, it calls for in depth problem characterization, new strategies and highly responsive algorithms to answer the challenges.

3. Problem Formulation

We use the notation $x.y$ to denote the y component of x. Given k contexts in a d-dimensional space, we formally characterize the problem as follows.

Definition 1 (Location)
A location $x = (x_1, x_2, \ldots, x_d) \in R^d$ is a point (or a vector) in the d-dimensional space.

Definition 2 (Context Node)
A context node is a triple (x, t, v) where x is the location, t is the context type with domain $D(t)$ and $v \in D(t)$ is the context value. Without loss of generality, we use the context number $i, 1 \leq i \leq k$ to denote its type.

Definition 3 (Context Field)
A context field is a pair (F, B) where F is a set of context nodes and B is the bounding object such that $\forall c \in F, c.x \in B$.

A context field can have nodes of different types to model multiple contexts. *Static* contexts do not change. Otherwise, the contexts are *dynamic*. If there is only one context, it is called a *single context field*. If the context nodes are evenly distributed, it is called a *uniform context field*. Most existing works on coverage presume a single, static and uniform context field.

Definition 4 (Sensor Node and Network)
*A sensor node is a tuple (x, s, c, r, m, d) where x is the location. $s = (s_1, s_2, \ldots, s_k)$ is the sensor readings where s_i is the number of context nodes of type i observed by the sensor. $c = (c_1, c_2, \ldots, c_k)$ is the sensing capability where c_i is either the maximal number of context nodes of type i that can be covered or **null**, indicating a turned off or nonexisting capability. $r = (r_1, r_2, \ldots, r_k) \in R^k$ is the sensing radiuses. m is the communication radius. d is the maximum speed. A sensor network is a collection of sensor nodes.*

Definition 5 (Path)
A path in a sensor network S from node i to j is a sequence of nodes $i = i_1, i_2, \ldots, i_n = j$ such that $i_k \in S$ and $|i_k.x - i_{k+1}.x| \leq i_{k+1}.m, 1 \leq k \leq n-1$.

Definition 6 (Connected Sensor Network)
A connected sensor network S is a sensor network such that $\exists N \subset S$ which is the set of sinks, $\forall i, j \in N, i \neq j \implies \exists$ a path from i to j, and $\forall k \in S, \exists l \in N$ such that $\exists$ a path from k to l.

Definition 7 (Cover)
For a context node a and a sensor node b, we say that a is covered by b (denoted by $a \sqsubset b$) if $|a.x - b.x| \leq b.r_{a.t} \wedge b.c_{a.t} \geq b.s_{a.t}$, i.e. both the sensing radius and capability constraint are satisfied.

Definition 8 (Coverage)
For a context field F, a connected sensor network S and a subset P of the k contexts, the percentage of context nodes in F of types in P that are covered by the nodes in S is called the context coverage of F on P (denoted by $\mathcal{C}_P(F)$). When P is the set of all contexts, it is called the coverage of F (denoted by $\mathcal{C}(F)$).

Definition 9 (Problem Definition)
For a context field F, a set of sensor nodes S, k contexts and a desired level of coverage g, the dynamic context coverage problem is to continuously form a connected sensor network such that $\mathcal{C}(F) \geq g$.

The problem is challenging in several respects. Technically, to cover multiple contexts significantly complicates the problem since the sensor nodes need to strike for optimal balance. Furthermore, sensors are heterogeneous, mobile and capability-constrained. Existing solutions that assume homogeneous, static sensors and consider only communication radiuses can no longer be applied. The nodes must self-organize into a connected sensor network and continuously maintain the coverage. When all contexts remain unchanged, the sensors must stabilize themselves for energy efficiency. Theoretically, we want to determine the optimal coverage a set of sensor nodes can achieve, or the minimal number of nodes to reach a desired level of coverage. We also want to prove that, given a protocol, whether it will eventually stabilize if the contexts remain unchanged. For algorithm comparison, we want to know which one is more competitive in terms of context coverage, responsiveness, energy consumption and sensor topology. In this paper, we discuss our strategies and algorithms to quickly form a connected sensor network, achieve the desired level of coverage and continuously maintain the coverage.

4. Strategies and Algorithms

We devise three self-organization strategies and design several algorithms accordingly. We then build a simulation system which plot the sensor movement such that the network could be visualized. We present the strategies and algorithms in this section.

4.1. Algorithms based on Randomization

In randomization based strategy, sensor nodes move in random with context-sensitive control. The algorithm *context-sensitive random walk method (CSRWM)* is presented in Algorithm 1. Each node randomly choose the direction and speed of movement based on limited information from neighbors and the context reading. This is the simplest algorithm used as the base line for comparison.

Algorithm 1: Context-Sensitive Random Walk Method (CSRWM)

Data: $\mathcal{N}$ is the sensor node in question.
Result: The new location of $\mathcal{N}$.
if *(there is any neighbor lies within the sensing radiuses of* $\mathcal{N}$*)* $\vee$ *(*$\mathcal{N}$ *has no reading)* **then**
  $e \Leftarrow$ a random number within 0 and 1
  $\mathcal{V} \Leftarrow$ a random unit vector
  $\mathcal{N}.x \Leftarrow \mathcal{N}.x + e \times \mathcal{N}.d \times \mathcal{V}$
end

The algorithm *simulated annealing method (SAM)* presented in Algorithm 2 is CSRWM augmented with a context-sensitive *temperature* to speedup or slow down the random walk. The higher the temperature, the more active the sensor nodes which are more likely to spread toward uncovered areas but less likely to stabilize. Theoretically, if we start with a high temperature and gradually decrease it in a slow enough pace, then the sensor network will eventually reach an optimal stable state if all contexts remain unchanged.

Algorithm 2: Simulated Annealing Method (SAM)

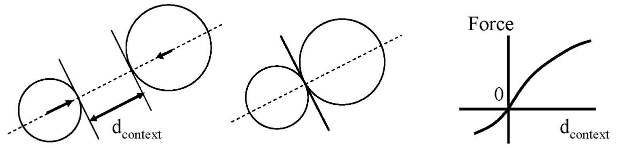
Data: $\mathcal{N}$ is the sensor node in question.
$\mathcal{B} = b_1, b_2, \ldots, b_n$ are neighbors of $\mathcal{N}$.
e is the ratio of annealing effect.
Result: The new location of $\mathcal{N}$.
$\mathcal{A} \Leftarrow \sum_{i=1}^{n} b_i.s + \mathcal{N}.s$
$\mathcal{K} \Leftarrow \sum_{i=1}^{n} b_i.c + \mathcal{N}.c$
$\mathcal{V} \Leftarrow$ a random unit vector
if $\mathcal{N}.s$ *has changed since the last round* **then**
  $f \Leftarrow \frac{|\mathcal{N}.c| - |\mathcal{N}.s|}{|\mathcal{N}.c|}$
  $g \Leftarrow \frac{|\mathcal{K}| - |\mathcal{A}|}{|\mathcal{K}|}$
  /* *Update the temperature.* */
  $t \Leftarrow \frac{f}{g} - 1$ if $g \neq 0$, otherwise $t \Leftarrow f$
  if $t > 1$ **then**
    $t = 1$
  else if $t < 0$ **then**
    $t = 0$
else
  $t \Leftarrow$ (previous value of t) $\times$***e***
$\mathcal{N}.x \Leftarrow \mathcal{N}.x + t \times \mathcal{N}.d \times \mathcal{V}$

4.2. Algorithms based on Virtual Forces

In virtual forces based strategy, we model either the context readings or the variations with respect to neighboring nodes as virtual forces. By combining all the forces affecting a sensor node, we can obtain a net force to drive the sensor node toward its ideal location. Theoretically, if the contexts remain unchange, the sensor network will eventually stabilize. Algorithms based on virtual forces are naturally self-organized. We have designed two algorithms based on virtual forces: the *subjection method (SM)* and the *collision/diffusion method (CDM)*.

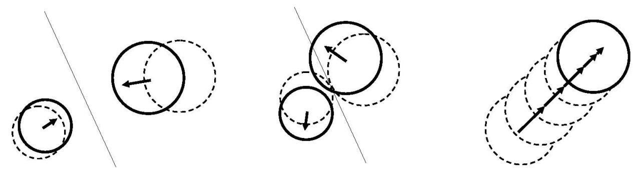
SM is based on distance function (geographical distance, context distance, etc.) to determine the force. Attraction force is useful in healing the uncovered breaches while repulsion force is good for expanding the coverage to previously unattended area. Two forces can be combined to define a function of subjection force against virtual distance as the example in Figure 1. The circles denote the sensing radiuses. The difference in context readings ($d_{context}$) is used to determine inter-subjection. (b) demonstrates the state of zero subjection. The subjection can become negative for repulsion force as in (c). It is permissible for different sensor nodes to use different distance functions. Details of the SM is presented in Algorithm 3.

CDM is to simulate the interaction among particles. Each sensor node is treated as a particle with its context readings as its size. Figure 2(a) and (b) demonstrate the collision effect. We introduce a friction effect to eventually stabilize the sensors, as shown in (c), which is also used to model the diminishing effect of diffusion. Algorithm 4 describes the procedure CDM.



(a) Subjection by context distance (b) Zero subjection (c) Subjection-distance function

Figure 1. Subjection-Distance function.



(a) Before collision (b) After collision (c) Friction Effect

Figure 2. The collision and friction effect.

Algorithm 3: Subjection Method (SM)

Data: $\mathcal{N}$ is the sensor node in question
$\mathcal{B} = b_1, b_2, \ldots, b_n$ are neighbors of $\mathcal{N}$.
Result: The new location of $\mathcal{N}$.
$\mathcal{E} \Leftarrow$ a zero vector
`Preferred_Distance`($\mathcal{N}$)
foreach *neighbor* b_i *in* $\mathcal{B}$ **do**
  `Preferred_Distance`(b_i)
  $p \Leftarrow \mathcal{N}.p + b_i.p$
  $\mathcal{V} \Leftarrow b_i.x - \mathcal{N}.x$
  if $|\mathcal{V}| \neq 0$ **then**
    /* *Add the force from* b_i. */
    $\mathcal{F}_i \Leftarrow \frac{\mathcal{V}}{|\mathcal{V}|} \times (\frac{|\mathcal{N}.s|}{|\mathcal{N}.c|} - \frac{|b_i.s|}{|b_i.c|}) \times (\frac{p}{|\mathcal{V}|})^2$
    $\mathcal{E} \Leftarrow \mathcal{E} + \mathcal{F}_i$
end
if $|\mathcal{E}| > \mathcal{N}.d$ **then** $\mathcal{E} \Leftarrow \frac{\mathcal{E}}{|\mathcal{E}|} \times \mathcal{N}.d$
$\mathcal{N}.x \Leftarrow \mathcal{N}.x + \mathcal{E}$

Procedure `Preferred_Distance(`n`)`

---

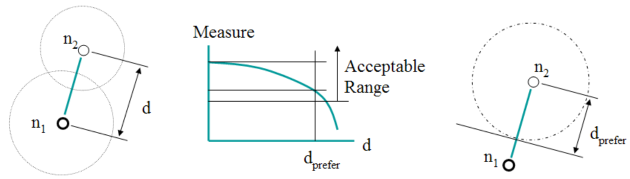
if $|n.s| > |n.c|$ **then**
  $n.p \Leftarrow \frac{\sqrt{3}}{2} \times |n.r| \times \sqrt{\frac{|n.c|}{|n.s|}}$
else
  $n.p \Leftarrow |n.r|$

Algorithm 4: Collision/Diffusion Method (CDM)

Data: $\mathcal{N}$ is the sensor node
$\mathcal{B} = b_1, b_2, \ldots, b_n$ are neighbors of $\mathcal{N}$
$\boldsymbol{f}$ is the ratio of friction effect.
Result: The new location of $\mathcal{N}$.
$\mathcal{N}.D \Leftarrow$ The last displacement of $\mathcal{N}$
if $\mathcal{B} \neq \emptyset$ **then**
  $\mathcal{I} \Leftarrow \mathcal{O} \Leftarrow$ a zero vector
  `Preferred_Distance`($\mathcal{N}$)
  foreach b_i *in* $\mathcal{B}$ **do**
    `Preferred_Distance`(b_i)
    $p \Leftarrow \mathcal{N}.p + b_i.p$
    $\mathcal{V} \Leftarrow b_i.x - \mathcal{N}.x$
    if $|\mathcal{V}| \neq 0$ **then**
      $\mathcal{U} \Leftarrow \frac{\mathcal{V}}{|\mathcal{V}|} \times (|\mathcal{V}| - p)$
      /* *Compute the collision force.* */
      if $|\mathcal{V}| < p$ **then**
        $b_i.D \Leftarrow$ The last displacement of b_i
        $\mathcal{I} \Leftarrow \mathcal{I} + (\frac{|\mathcal{N}.s| - |b_i.s|}{|\mathcal{N}.s| + |b_i.s|}) \times \mathcal{N}.D + (\frac{2|b_i.s|}{|\mathcal{N}.s| + |b_i.s|}) \times b_i.D$
      else
        $\mathcal{O} \Leftarrow \mathcal{O} + \mathcal{U} \times (\frac{p}{|\mathcal{U}|})^2$
  end
  if $|\mathcal{I}| \neq 0$ **then** $\mathcal{O} \Leftarrow \mathcal{I}$
  if $|\mathcal{O}| > \mathcal{N}.d$ **then** $\mathcal{O} \Leftarrow \frac{\mathcal{O}}{|\mathcal{O}|} \times \mathcal{N}.d$
  $\mathcal{N}.x \Leftarrow \mathcal{N}.x + \mathcal{O}$
else
  /* *Apply friction effect when no collision.* */
  $\mathcal{N}.x \Leftarrow \mathcal{N}.x + \mathcal{N}.D \times \boldsymbol{f}$

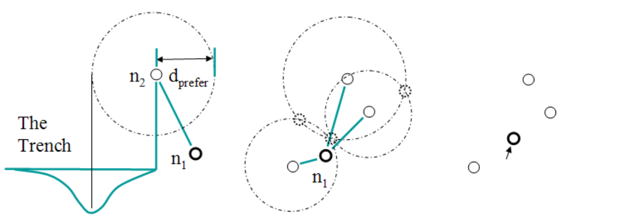
4.3. Algorithm based on Spatial Information

This strategy exploits relative spatial relationships of neighboring sensors to determine the right position of a node. We have designed the *trenching method (TM)* based on spatial information. Each sensor calculates the position that it should move to based on its relationships with neighbors, and thereby establishing the best local topology (in terms of coverage). The key is to define a quality measure (the trench) to derive the most appropriate relative positions of two nodes, as shown in Figure 3. The quality measure can be any thing varies with context distance such as the communication quality or coverage level. In general, the most appropriate distance is the point before the drastically dropping of quality or the minimal acceptable QoS to have the broadest coverage. Figure 4(a) demonstrates an example of a quality measure. Node n_1 is expected to move to the preferred distance of n_2. (b) shows a more generalized case where node n_1 is affected by three neighbors. The best position is the deepest location that meets all neighbors' expectations as much as possible. TM (Algorithm 5) is very general and flexible since any reasonable quality measure can be adopted. By using different measures, we can employ different strategies based on the same algorithm.



(a) Distance and sensed area (b) Definition of quality index (c) Preferred distance of n_1 to n_2

Figure 3. Example of spatial relationship measure in trenching method.



(a) The trench and preferred distance (b) Trenching results of various nodes (c) Move to the most appropriate position

Figure 4. The trenching method.

Algorithm 5: Trenching Method (TM)

Data: $\mathcal{N}$ is the sensor node
$\mathcal{B} = b_1, b_2, \ldots, b_n$ are neighbors of $\mathcal{N}$.
Result: The new location of $\mathcal{N}$.
$\mathcal{I} \Leftarrow \mathcal{O} \Leftarrow$ a zero vector
`Preferred_Distance`$(\mathcal{N})$
foreach b_i *in* $\mathcal{B}$ **do**
  /* *Compute the best trench between* $\mathcal{N}$ *and* b_i. */
  `Preferred_Distance`(b_i)
  $p \Leftarrow \mathcal{N}.p + b_i.p$
  $\mathcal{V} \Leftarrow b_i.x - \mathcal{N}.x$
  if $|\mathcal{V}| \neq 0$ **then**
    $\mathcal{U} \Leftarrow \frac{\mathcal{V}}{|\mathcal{V}|} \times (|\mathcal{V}| - p)$
    if $|\mathcal{V}| < p$ **then**
      $\mathcal{I} \Leftarrow \mathcal{I} + \mathcal{U}$
    else
      $\mathcal{O} \Leftarrow \mathcal{O} + \mathcal{U} \times (\frac{p}{|\mathcal{U}|})^2$
end
if $|\mathcal{I}| \neq 0$ **then** $\mathcal{O} \Leftarrow \mathcal{I}$
if $|\mathcal{O}| > \mathcal{N}.d$ **then** $\mathcal{O} \Leftarrow \frac{\mathcal{O}}{|\mathcal{O}|} \times \mathcal{N}.d$
$\mathcal{N}.x \Leftarrow \mathcal{N}.x + \mathcal{O}$

---

5. Performance Evaluation

We develop a Java based simulation environment as depicted in Figure 5 and Figure 6. We use a 300 by 300 context field, 200 sensor nodes and single context with 800 context nodes randomly distributed across the field. Initially, all sensor nodes are placed in the circle of initial placement

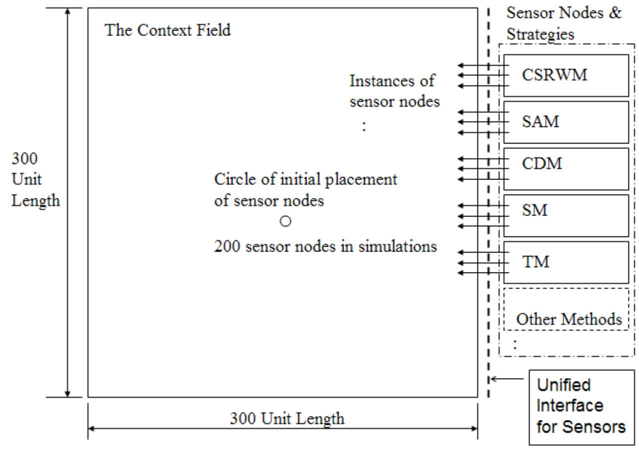


Figure 5. The simulation environment.

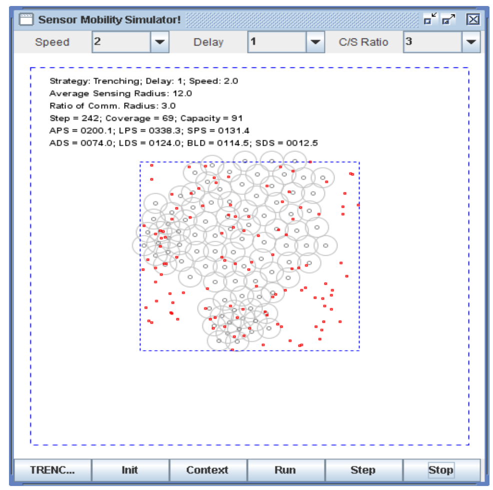


Figure 6. A screen shot of the simulation environment and a simulation in progress.

at the center. The context field object is responsible for providing all contextual information and keeping track of global statistics such as the coverage level. Sensor parameters and settings are: **speed**(1)–maximum moving length per step; **capability**(4); **sensing radius**(12); **communication radius**(36); **accumulated moving distance**; **delaying factor**(1)–number of steps between movement decisions; **strategy**–CSRWM, SAM, SM, CDM, or TM. Both the coverage and the speed of achieving it are evaluated. Moving distance is an indication of energy consumption. Coverage pattern demonstrates how well a method covers the field.

Figure 7 shows the results on CSRWM. Randomization effect is evident by the variation of coverage. It takes about 1000 rounds to reach 30% on the average. CSRWM can not maintain the coverage, and the network is neither stable nor organized. SAM is similar to CSRWM as shown in Figure 8.

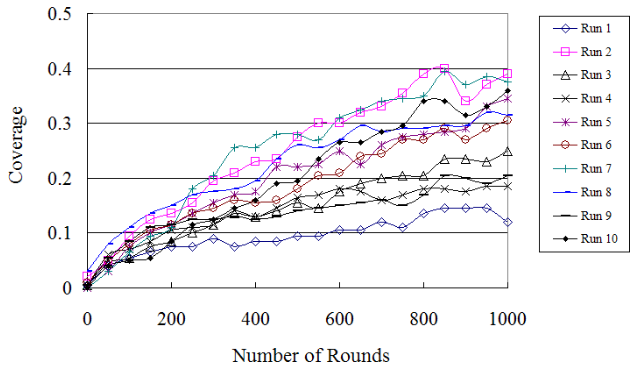


Figure 7. Coverage adaptation of CSRWM.

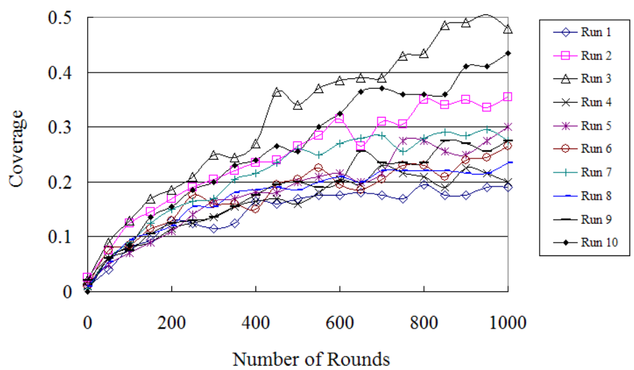


Figure 8. Coverage adaptation of SAM.

We can speed up or slow down the sensors by controlling the annealing process. Theoretically, the network pattern will stabilize if the annealing is performed adequately, and the sensor nodes will not move since then. SM is much more effective than the randomization based methods, as demonstrated in Figure 9. It takes only about 100 rounds to reach 80% to 90% coverage. Once achieved, it can effectively maintain the level without much vibration. Figure 10 demonstrates the results on CDM. Average of 95% coverage is achieved at around 200 rounds with high stability and low variation. All measures indicate that CDM is a very effective method. Figure 11 shows the coverage adaptation of TM. It takes about 300 rounds to reach its high level of coverage with high stability and low variation.

Another dimension of evaluation is the average moving distance. For achieving the same coverage level, the lower the distance, the better the method. Figure 12, Figure 13 and Figure 14 show the results along 1000 rounds of executions of SM, CDM and TM, respectively with those of CSRWM and SAM for comparison. Randomization based methods perform worst since all sensors move with no or limited consideration of the contexts. Both CDM and TM perform well. In most cases, a node takes no more than 25% of its maximal speed and still reach a high coverage with low variation. SM also performs well but both the moving

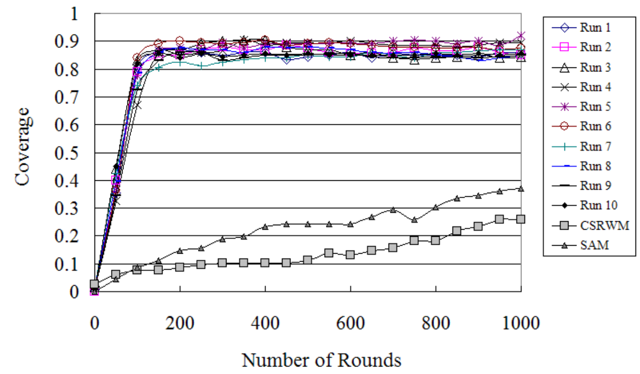


Figure 9. Coverage adaptation of SM.

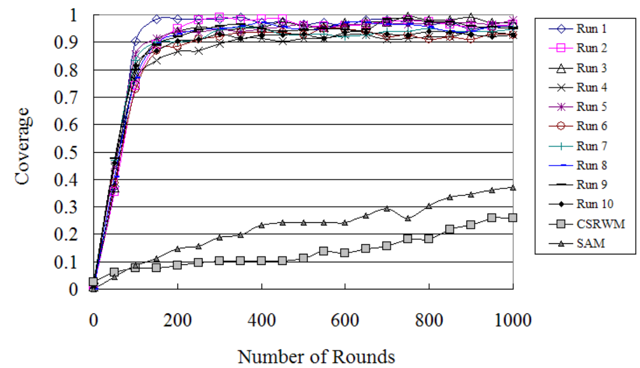


Figure 10. Coverage Adaption of CDM

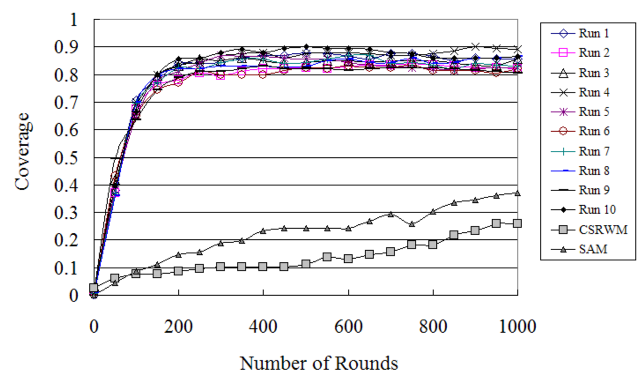


Figure 11. Coverage Adaption of TM

distance and the variation are larger than CDM and TM.

Figure 15 shows the coverage adaption of all methods. Figure 16 presents the range of coverage. CSRWM and SAM can achieve only about 30% coverage in 1000 rounds. SM, CDM and TM reach high coverage in about 200 to 300 rounds, and maintain the coverage. CDM has the highest coverage while SM performs slightly better than TM.

Figure 17 is the comparison of average moving distance and Figure 18 shows the range. TM achieves similar coverage as SM with a smaller moving distance and lower variation. CDM stands out with the smallest moving distance

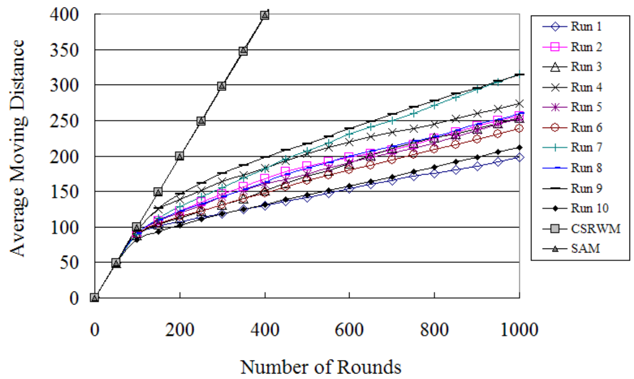


Figure 12. Moving Distance of SM

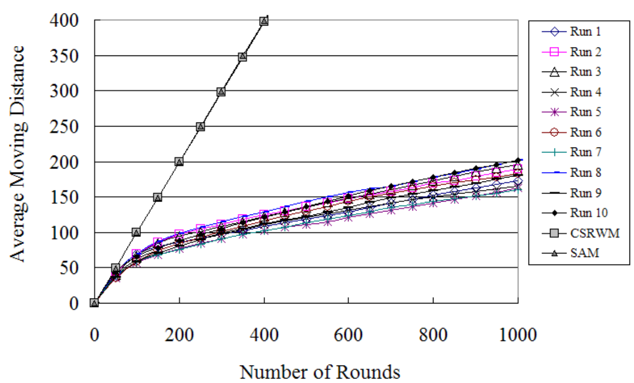


Figure 13. Moving Distance of CDM

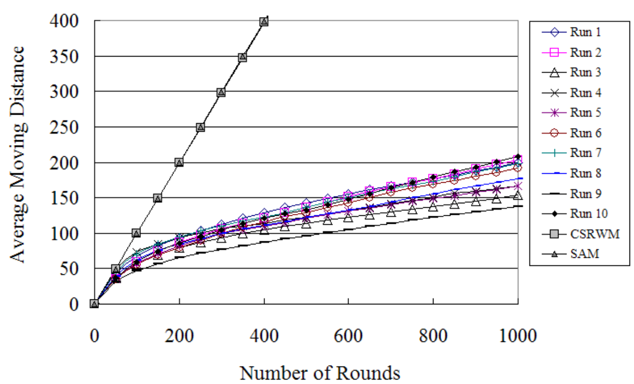


Figure 14. Moving Distance of TM

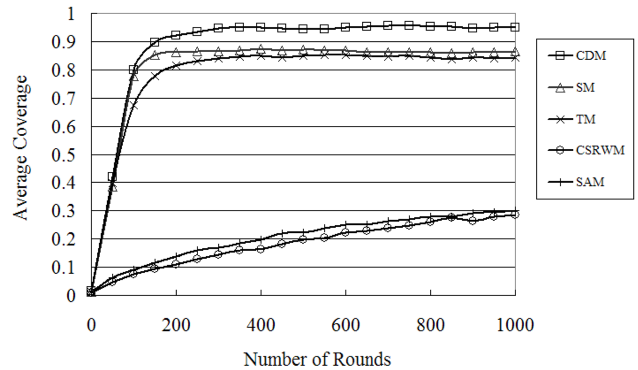


Figure 15. Comparison of Coverage Adaption

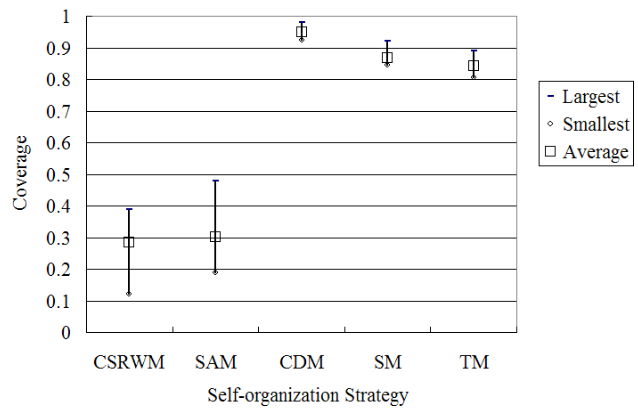


Figure 16. Range of Coverage

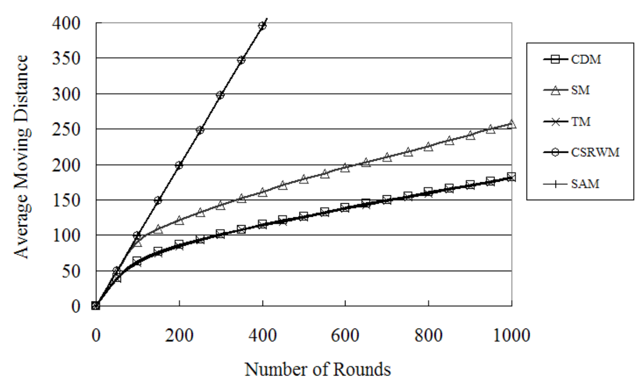


Figure 17. Comparison of Moving Distance

and lowest variation.

Figure 19 demonstrates the simulation of context change at every 100 rounds interval. CDM, SM and TM all respond quickly within about 20 to 50 rounds.

CDM outperforms the others because it always considers its nearest neighbors to avoid unnecessary movement that may be introduced by considering more neighbors. It also reaches the highest coverage by responding to the nearest neighbors immediately, rather than averaged by the influence of other neighbors. TM outperforms SM in moving distance because TM directly calculates a *destination*. TM and SM are comparable in coverage because they both take all neighbors into account, and therefore unlikely to respond to the urgent need of nearest neighbors.

6. Conclusions and Future Work

We have identified and formulated the problem of dynamic context coverage in capability-constrained mobile

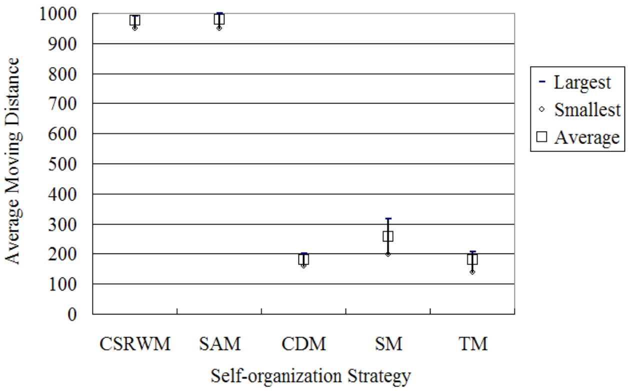


Figure 18. Range of Moving Distance

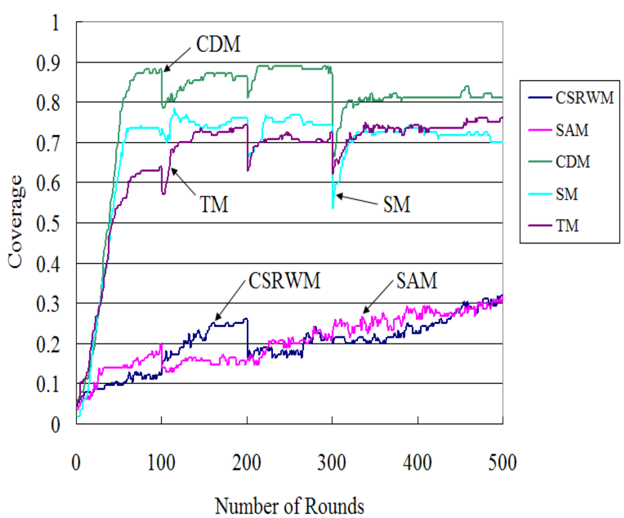


Figure 19. Comparison of Coverage Adaption in Dynamic Context

sensor network which is expected to have broad applications in domains such as environment monitoring, ecological protection, disaster search/rescue, and military applications. Based on the inspiration from mother nature, we propose several self-organization strategies and algorithms for the responsive adaptation of sensor nodes to the coverage of a field with multiple and dynamically changing contexts. Simulation and performance evaluation results demonstrate that the proposed methods can quickly achieve and maintain the desired level of coverage.

We plan to further scrutinize our algorithms to improve their responsiveness while minimize their energy consumption. The design of dynamic data management and query processing schemes is also on the top of our list. The theoretical aspects of the proposed problem and solutions are both interesting and practically useful. We are now building mobile carrier for the MICA2 sensor network. It is our goal to eventually apply the technology on target applications.

References

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[2] C.-Y. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8), August 2003.

[3] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. k. Saluja. Sensor deployment strategy for target detection. In *WSNA2002: 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, 1999.

[5] H. Gupta, S. R. Das, and Q. Gu. Connected sensor cover: Self-organization of sensor networks for efficient query execution. In *4th ACM International Symposium on Mobile Ad hoc Networking and Computing*, pages 189–200, 2003.

[6] A. Howard, M. J. Mataric, and G. S. Sukhatme. An incremental self-deployment algorithm for mobile sensor network. *Autonomous Robots*, 13(2):113–126, Sep 2002.

[7] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *DARS02: 6th International Symposium on Distributed Autonomous Robotics Systems*, 2002.

[8] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. *ACM Mobile Networks and Applications (MONET), special issue on Wireless Sensor Networks*, 2005. (to appear).

[9] X.-Y. Li, P.-J. Wan, and O. Frieder. Coverage in wireless ad hoc sensor networks. *IEEE Trans on Computers*, 52(6):753–763, Jun 2003.

[10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *IEEE Infocom*, volume 3, pages 1380–1387, 2001.

[11] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *MobiCom'01: 7th Annual International Conference on Mobile Computing and Networking*, pages 139–150, 2001.

[12] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor network. In *1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 32–41, 2002.

[13] G. Wang, G. Cao, and T. L. Porta. Movement-assisted sensor deployment. In *IEEE INFOCOM*, 2004.

[14] G. Wang, G. Cao, T. L. Porta, and W. Zhang. Sensor relocation in mobile sensor networks. In *IEEE INFOCOM*, 2005.

[15] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *IEEE INFOCOM*, 2003.

[16] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems*, 3(1):61–91, 2004.