Coverage-Hole Trap Model in Target Tracking using Distributed Relay-Robot Network

Huawei Miao, Chia Ching Ooi, Xiaowen Wu, and Christian Schindelhauer
Computer Networks and Telematics, Department of Computer Science, University of Freiburg
Georges-Köhler-Allee 51, 79110 Freiburg, Germany
miaohuawei@hotmail.com, ooi@informatik.uni-freiburg.de,
balance_xiaowen@hotmail.com, schindel@informatik.uni-freiburg.de

ABSTRACT

Target tracking is an important issue in wireless sensor network applications. In this paper, we design a Coverage-Hole Trap Model (CTM) based on a system that contains one moving target, one moving pursuer and a distributed relay robot network with sensing coverage holes. Usually, the coverage hole is harmful for target tracking in wireless mobile robot network (WMRN). Many algorithms have been proposed to detect and avoid coverage holes. In this paper, we try to use coverage holes as traps to point out the region where the target is moving into, and help the pursuer to catch the target. After the coverage holes are discovered by multiple relay robots in the initialization phase, the pursuer calculates the target position and predicts where it should move to. We propose Distributed Coverage-Hole Detection Algorithm (DCDA), which is based on 3MeSH method to discover coverage holes and tackle this challenge by introducing the Coverage-Hole Based Pursuer Algorithm (CBPA). CBPA is a prediction-based algorithm for the pursuer using the information about the target and coverage holes obtained from the relays. Simulation results show that our methods address the limitation of the previous work, considerably improve the required tracking time and reduce the average total traveling distance of target.

1. INTRODUCTION

Target tracking is one of the killer applications for wireless sensor networks. Like other sensor networks, a network designed for target tracking has unattended nodes, wireless communications capability and the distributed system architecture. In addition, a lot of unique factors are considered. They include the number of moving objects, data precision, reporting frequency, sensor sampling frequency, moving speed of object and pursuer, and the location information of the moving objects. Normally, the existence of coverage holes in the wireless sensor network dedicated for target tracking is not a good case. Once the target runs into the coverage hole, relays can not detect its position and direction, which are important for the pursuer. When the pursuer runs into a coverage hole of certain size, relays can not communicate with it and can not provide the useful information about the target to the pursuer. A lot of algorithms have been proposed to detect and avoid coverage holes [3, 4, 5, 7, 2, 13, 14, 9].

However, we take a different approach. We build a new model named Coverage-Hole Trap Model (CTM) based on a system that contains one moving target, one moving pursuer and a distributed relay robot network that contains several coverage holes. Without knowing the coverage hole information, we can improve the required tracking time and reduce the average total traveling distance of target using CTM in the target tracking scenario compared with the same scenario using Distributed Relay-robot Algorithm (DRA) [8].

The rest of this paper is organized as follows: Section 2 describes the related work. Section 3 shows the problem formulation for mobile target tracking. In Section 4, we discuss the implementation issue for our CTM. This section includes Distributed Coverage-Hole Detection Algorithm (DCDA) for the relay robot network and Coverage-Hole Based Pursuer Algorithm (CBPA) for the pursuer. In Section 5, we compare our work with DRA, and finally, Section 6 provides some concluding remarks and future work.

2. RELATED WORK

Sensing coverage and connectivity has been a popular topic in wireless sensor networks since the past decade. Most of the papers focus on maintaining the networks connectivity, such as [5, 11, 12]. Other approaches, [3] explores the use of routing information for detecting holes. Coverage hole can be detected by a node which is far away and not directly affected by the damage that created the hole. The method does not require an initialization phase to learn the initial state of connectivity the network. [4] introduces a technique for detecting holes in coverage by means of homology, an algebraic topological invariant, while [7] introduces 3MeSH, a distributed coordinate-free hole detection algorithm for detecting holes that are introduced by sensor failure or mobility.

[10] considers a hybrid Wireless Sensor Network (WSN) deploying in a region with obstacles. Static sensors are used to monitor the environment. When detecting an event, those sensors will search nearby mobile sensors to move to their locations to conduct more in-depth analysis of events. In [6], the authors propose a location tracking architecture known as Scalable Tracking Using Networked Sensors (STUN). It is a hierarchical structure of sensors to track objects. The

leaves are sensors, the querying point as the root and the other nodes are communication nodes. This reduces the redundant messages. The message pruning in the hierarchy is the key to reduce the communication cost. But this method should build the structures of the tree.

In [15], authors propose a dynamic convoy tree-based collaboration (DCTC) framework to detect and track the mobile target and monitor its surrounding area. DCTC relies on a tree structure called convoy tree. The tree is dynamically configured to add some nodes and prune some nodes as the target moves. Considering the expensive investment of the real-life sensor deployment if require every point in the region to be covered (i.e., full coverage), [1] proposes a new mode of coverage called Trap Coverage that scales well with large deployment regions. But this method is only for deployment of relay robots.

No previous researches focus on treating coverage hole in the network as a good case in the target tracking scenario according to our best knowledge. In our work, when a pursuer receives the message from the relay robot network, it can get not only the target information but also the coverage hole information when the target is moving into the coverage hole.

3. PROBLEM FORMULATION

3.1 Symbols

We have several symbols as follows:

- R_c: communication range of robot
- R_s: sensing range of robots
- $G = (S, E)$: a communication graph if and only if $||s_i, s_j||_2 \leq R_c$
- $t_s = 0$: the time the target first enters the sensing area
- t_e: the time the pursuer captures the target
- $p_p(t)$: the position of the pursuer robot at time t
- $p_q(t)$: the position of the mobile target at time t
- When a target Q appears at $p_q \in R$, relay robot i can detect Q if and only if $||s_i, p_q||_2 \leq R_s$
- v_q: the target moving constant speed
- v_p: the pursuer moving constant speed
- $H = (b_1, b_2, b_3, \ldots, b_n)$: coverage hole with n boundary nodes

3.2 Assumptions

We have the following assumptions in this paper:

- Each relay robot has a unique id. It is aware of its position and is in the active state.
- All of relay robots are static and keep their states fixed.
- A relay robot knows its location and the locations of its neighbors through communication at the network deployment stage.
- The relay robots are able to self-organize to form an ad hoc robot network when a tracking event is triggered.
- No global information and no centralized infrastructure are provided.
- The network is not so dense, it exists coverage hole in the deployment stage.
- All the relay robots in the network act as multihop relays.
- The relay robots network sends messages to detect the coverage holes in the whole network in initialization.
- Each coverage hole is a polygon in geometry.
- Each boundary node for coverage hole is a polygon vertex in geometry.
- Target runs in the constant speed.

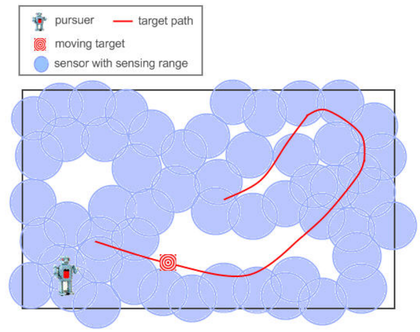


Figure 1: Mobile target tracking using wireless relay robots with coverage holes

- Pursuer runs in the constant speed.
- Relay robots in the network act as static multihop relays.
- When target appears in the pursuer's R_s, target tracking terminates.

3.3 Tasks

Our tasks focus on the multiple relay robot network that has sensing coverage holes. As shown in Figure 1, the WMRN contains one single pursuer, one single moving target and static multi-hop relay robots. After relay robots discovered the coverage holes during initialization, the WMRN uses our proposed scheme that is described in the following sections to conduct target tracking.

4. COVERAGE-HOLE TRAP MODEL(CTM)

Our Coverage-Hole Trap Model contains three parts. First, the coverage hole detection that uses DCDA in the initialization phase. When the target is moving in the tracking area without the coverage hole, relay robots send *tAdv* and trigger DRA mentioned in[8]. When the target moves into the area with some coverage hole, relays send *hAdv* and trigger Coverage-Hole Based Pursuer Algorithm (CBPA).

4.1 Distributed Coverage-Hole Detection Algorithm (DCDA)

Distributed Coverage-Hole Detection Algorithm (DCDA) detects coverage holes in the network initialization based on 3MeSH method. The goal of DCDA is to determine whether a relay robot is on the boundary of a coverage hole or not, and subsequently, determine the existence of coverage holes in the distributed relay robot network in CTM. DCDA is composed of three parts: 1-hop neighbors information collection, classification of relay robots and large hole detection. Each of them is described in the following subsections.

4.1.1 Neighbor Information Collection

Each relay robot sends out a ping requesting message to its 1-hop neighbors. Neighbors reply with their id and position. The pinging node considers itself at the origin of the Cartesian coordinates. According to the obtained information, it calculates the distance and the angle, θ of the relative position to its neighbors as illustrated in Figure 2.

Pinging node stores the neighbor id, position, distance, angle θ in a neighbor node list (nnList). In the following part, we show how the type of pinging node can be classified by nnList.

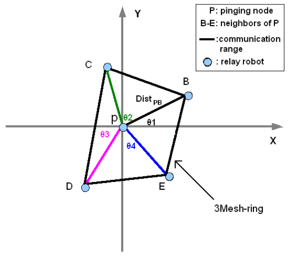


Figure 2: Distance, angle, 3MeSH-ring

Table 1: Type of relay robots

Type of relay robot	Definition	Abbr.
Active Node	Neighbor nodes can form a 3MeSH-ring that links connecting the pinging node and its neighbors partition the area within the ring into triangles	AN
Boundary Node	Neighbor nodes can not form a 3MeSH-ring	BN
Island Node	No neighbor	IN

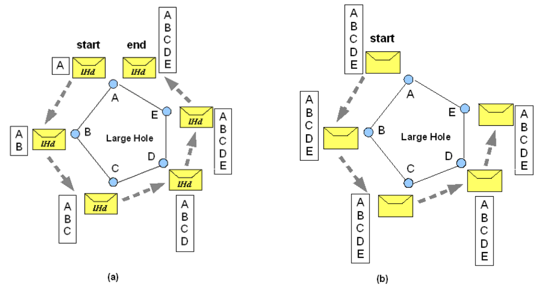


Figure 3: (a) Large hole detection by using *lhd* message (b) *A* informs other nodes about the large hole information

Table 2: Parameters of *lhd* message

Variable name	Type	Description
destAddr	int	Destination address of lhd message
srcAddr	int	Source address of lhd message
holeAccum	list	Boundary node list

4.1.2 *Classification of Relay Robots*

At the beginning of this part, we introduce three basic types of relay robots: Active Node, Boundary Node and Island Node which are listed in Table 1. According to the definition, pinging node in Figure 2 is an Active Node. After determining the definition above, we propose a method to identify the type of each relay robot. Here, |nnList| represents the size of the nnList.

- |nnList| = 0, P is an IN.
- |nnList| = 1 or 2, P is a BN.
- |nnList| > 2 and all neighbors are in one side of pinging node which means neighbors cannot form a 3MeSH ring. P is a BN.
- |nnList| > 2 and some of the neighbors can form a 3MeSH ring. P is AN.

4.1.3 *Large-Hole Detection*

Any large hole in WSN is formed by BN. Figure 3(a) illustrates our main idea on how to identify a large hole in a target area via message transmission. Node A sends a large hole detection message (*lhd*) to node B. *lhd* message records the nodes along the transmission path. Upon the end of message circulation, all the nodes A, B, C, D and E are added into the message. When node A receives *lhd* message, it knows that the large hole is found out. The parameters of *lhd* message are shown in Table 2.

In above case, the problem is that only the first node, A knows that a large hole has been detected. To solve such a problem, node A needs to send another message that contains the large hole information for nodes B, C, D and E. Then they save large hole information to be used by the tracking algorithm, Coverage-Hole Based Pursuer Algorithm (CBPA). Figure 3(b) illustrates this procedure.

4.2 Distributed Relay-robot Algorithm (DRA)

In DRA, multiple relay robots disseminate the information of target trajectory and the pursuer robot P do the path planning based on the future target position. The static relay robot that senses the target in its sensing range broadcasts a *tAdv* (target Advertisement) packet containing detection time, target position, direction and speed to build up a path to guide the pursuer. When P receives such *tAdv* packet, it uses a prediction-based algorithm to compute its path to predict the new target position and move to that position to approach target. Figure 4 shows the prediction of mobile target location using this strategy. More details on the algorithm can be found in [8].

4.3 Coverage-Hole Based Pursuer Algorithm (CBPA)

input : $p_q(t_s)$
output: Path Plan for P

$p'_q \leftarrow p_q(t_s)$
repeat
  move to p'_q at v_p
  if *receive tAdv* **then**
    run DRA
  if *receive hAdv* **then**
    run Algorithm 2
until Q *is captured*

Algorithm 1: Target Path Planning for Pursuer

When the target is moving into the tracking area with coverage hole, relays send *hAdv* (hole Advertisement) and trigger Coverage-Hole Based Pursuer Algorithm (CBPA). First, P calculates the path it should move to capture Q based on Algorithm 1. P knows the starting position of Q, which is $p_q(t_s)$ at initialization in the network. For example, it can be done through a centralized controller or base station. Subsequently, it should get such position information from the messages sent by relay robots. The pursuer P moves with constant speed v_p to the target position. If Q is running in the area without coverage hole, relay robot which detects it will send *tAdv* to P. When P receives *tAdv* message from relay robots, it runs Algorithm 1 to predict the target future position p'_q.

When the target goes into a coverage hole, the boundary node that belongs to that coverage hole detects that Q is in its sensing range and hence, it sends *hAdv* to P. When P receives this *hAdv* message, it invokes Algorithm 2 to esti-

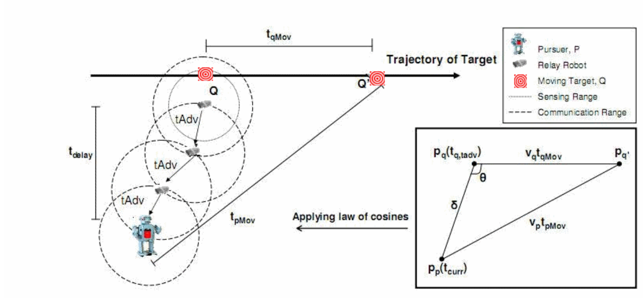
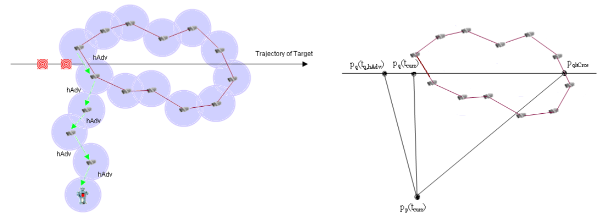
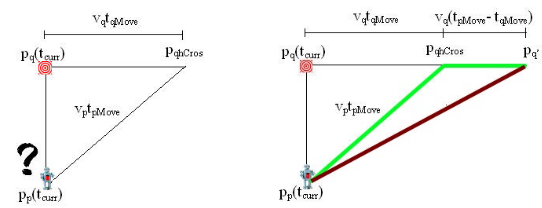


Figure 4: Prediction of mobile target location [8]



(a) (b)

Figure 5: (a)Scenario of prediction of mobile target location using *hAdv* (b) Geometry graph of scenario



(a) (b)

Figure 6: (a)Known and unknown variables for *P* (b) Case 1: $t_{pMove} > t_{qMove}$

input : previous p'_q, Q's latest detection time, $t_{q,latest}$, boundary list $H = b_1, b_2, b_3, \ldots, b_n$
output: Q's predicted position

retrieve $t_{q,hAdv}$ in $hAdv$
if $t_{q,hAdv} > t_{q,latest}$ **then**
  $t_{q,latest} \leftarrow t_{q,hAdv}$
  $t_{hdelay} \leftarrow t_{curr} - t_{q,hAdv}$
  retrieve $p_q(t_{q,hAdv})$,v_q and dir$_q$ in $hAdv$
  compute Dist$_{qc}$=$||p_q(t_{q,hAdv}), p_q(t_{curr})||_2$=$v_q \cdot t_{hdelay}$ to find $p_q(t_{curr})$
  compute p_{qhCros}
  get current position of P, $p_p(t_{curr})$
  compute t_{pMove} using $Dist_{pc}$=$||p_p(t_{curr}), p_{qhCros}||_2$=$v_p \cdot t_{pMove}$
  compute t_{qMove} using $Dist_{qc}$=$||p_q(t_{curr}), p_{qhCros}||_2$=$v_q \cdot t_{qMove}$
  if $t_{pMove} > t_{qMove}$ **then**
    p'_q=$p_q(t_{q,hAdv} + t_{qMove})$
  else
    $p'_q = p_{qhCros}$
else
  discard $hAdv$
return p'_q

Algorithm 2: Prediction of Target Position using *hAdv*

mate p'_q. *hAdv* contains such information not only in *tAdv* but also about the coverage hole Q moves into. The prediction of mobile target location by pursuer using coverage hole information is shown in Figure 5(a). We abstract a geometry graph from such scenario in Figure 5(b). In the graph, $p_q(t_{q,hAdv})$ is the target position at target detection time, $t_{q,hAdv}$ in *hAdv*. $p_p(t_{curr})$ is the pursuer position at the time P receives *hAdv*. $p_q(t_{curr})$ is the target position at the time P receives *hAdv*, and p_{qhCros} denotes the furthest segment intersection of the target trajectory and the polygon formed by sensing coverage hole.

Figure 6(a) shows the abstract geometry graphs of known and unknown variables for pursuer P. Let t_{qMove} be the time Q running from $p_q(t_{curr})$ to p_{qhCros}. We formulate the followings:

$$t_{qMove} = ||p_q(t_{curr}), p_{qhCros}||_2/v_q \tag{1}$$

Let t_{pMove} be the time P running from $p_p(t_{curr})$ to p_{qhCros}. We get

$$t_{pMove} = ||p_p(t_{curr}), p_{qhCros}||_2/v_p \tag{2}$$

Then, comparing t_{qMove} and t_{pMove}, there are three possible cases:

- $t_{pMove} > t_{qMove}$: when Q arrives p_{qhCros}, the exit point of the coverage hole, P is still on the way to p_{qhCros}. In this case, just let P move to the new position p'_q, which is calculated by function *rayToPoint*() given the direction and Cartesian Coordinate position of Q, and

$$||p_{qhCros}, p_q||_2 = v_q(t_{pMove} - t_{qMove}) \tag{3}$$

  P plans its path to reach, like illustrated in Figure 6(b), P will not go along the red path instead of the green path to get to p'_q.
- $t_{pMove} = t_{qMove}$: when Q arrives p_{qhCros}, the exit point of the coverage hole, P also arrives p_{qhCros}. It is a simple case, and let P move to p_{qhCros} as shown in Figure 6(b).
- $t_{pMove} < t_{qMove}$: when P arrives p_{qhCros}, the exit point of the coverage hole, Q is still inside the hole. When this happens, P just stays at p_{qhCros} until it receives new information about Q.

5. EVALUATION

We use the network simulation tool OMNet++ and its mobility extension Mobility Framework (MF) to do the CTM simulation and performance evaluation. Former sections have already described the relay robot network strategy DCDA and the pursuer strategy CBPA in the target tracking. The pursuer needs to get *tAdv* and *hAdv* messages from the relay robots to calculate the its own predicted path.

At first, we establish a one-hole scenario that the network only has one coverage hole with one moving target Q, one moving pursuer P and several relay robots that can relay message to P if necessary. In the network, Q runs into the coverage hole and P either only uses *hAdv* to predict path to

Table 3: Simulation setup for one-hole scenario

Parameter	Value(s)
Field Size (m^2)	800*800
Relay robot distribution	uniform
Number of pursuer	1
Number of relay robot	20
Number of target	1
Max. sensing range (m)	50
Max. transmission range (m)	100
Path loss exponent	3
Speed of P, v_p (m/s)	2
Speed of Q, v_q (m/s)	1,2,3,4
ttl: Initial, Incremental, Max.	1,1,20
Physical,Mac,Network layer header size (bits)	192,272,160

catch Q, or uses *tAdv* sent by relay robots without knowing coverage hole information mentioned in DRA. We compare the results produced by running both of them. Second, we establish a multi-hole scenario, which contains 50 static relay robots, one moving P and one moving Q. Some parts of the network contain coverage holes. When Q is running into the coverage hole, the boundary nodes belong to that hole send *hAdv* to P. Otherwise, they send *tAdv* to P. We compare the results of the proposed combined approach with the approach running only DRA.

The most important criterion of our evaluation is the time P consumes to capture Q. The others are the tracking failure percentage and the total traveling distance of Q. Both are important in target tracking scenario.

5.1 One-hole Scenario

The simulated scenario are run based on a special distributed CTM that use either *tAdv* or *hAdv* only. First, the simulation setup is described. Then, the results are presented and discussed by comparing the same scenario run separately twice using either only *hAdv* or *tAdv*. Some simulation setup information is summarized in Table 3. In this scenario, the simulations run on the 800m by 800m field with several relay robots uniformly deployed as in the previous section. The relay robots are static. There is only one coverage hole as shown in Figure 5(a). The maximum number of the boundary nodes is 17. Pursuer position is randomly deployed. It uses at most 3 relay robots to connect the pursuer to the one of the boundary nodes. Most of the time, it connects directly with coverage hole boundary node. All relay robots, pursuer and target apply IEEE 802.11 as their MAC layer protocols. Relay robot network uses flood as network layer protocol, no special routing protocol is used.

In order to do the performance comparison of simulation runs using *tAdv* and *hAdv* respectively, we deploy networks using the appropriate diameters. Figure 7(a) and Figure 7(b) illustrate the comparison results of the proposed Coverage-Hole Trap Model approach, CTM and Distributed Relay-robot Algorithm, DRA. In CTM, we assume that pursuer can only receive *hAdv* message in this one-hole scenario. The difference between using *tAdv* and *hAdv* is the usage of coverage hole information. We use average total time consumption by pursuer and average total traveling distance of target as the comparison metrics.

First, Figure 7(a) compares the average total time consumed by pursuer to detect target. Based on the successful cases performed by DRA and CTM, DRA performs better than CTM, especially at the ratio is 0.5 considering that pursuer runs further to the intersection on the coverage hole using CTM. The average total time reduces over higher speed ratio, for example, from 0.5 to 1, since the target's trajectory is a rectangle. P moves to the intersection, which is the other end of target trajectory segment. Obviously, if Q does not change its direction, the faster it runs, the sooner it will arrive at that intersection point. So pursuer can catch the target with less tracking time.

On the other hand, Figure 7(b) shows the average total moving distance of Q. If the target travels at longer distance at the same speed, P needs more tracking time to catch it. The increasing total distance over higher target speed is shown in both CTM and DRA.

Next, Figure 7(c) shows the percentage of tracking failure. For DRA, the reason of the failure is that it has no strategy for P that runs into a coverage hole and is disconnected from relay robots. CTM fails because CBPA is totally based on the information of the coverage hole. If relay robots can not detect the coverage holes correctly, and send *hAdv* to P, P will not be able to calculate the intersection point and can not predict target position. We can observe that when v_q=1,2,3 and 4, CTM performs without any failure while DRA has 3.33%, 3.33%, 16.67% and 10% of failure respectively. CTM is suitable under certain condition with the existence of coverage holes.

Table 4: Simulation setup for multi-hole scenario

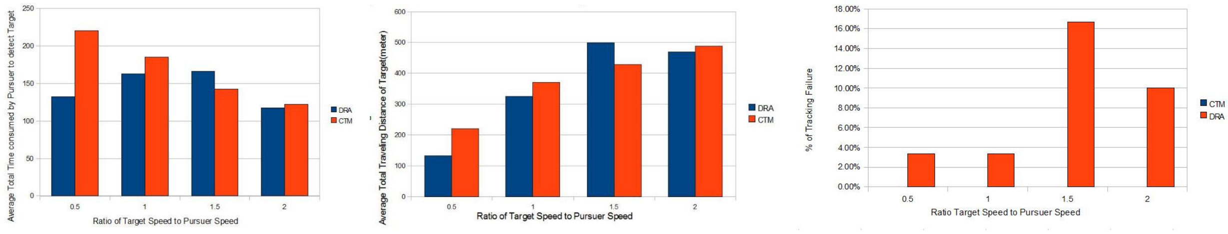
Parameter	Value(s)
Field Size (m^2)	800*600
Relay robot distribution	uniform
Number of pursuer	1
Number of relay robot	50
Number of target	1
Number of coverage holes	6

5.2 Multi-Hole Scenario

In this section, we set up manually the simulation scenario to evaluate our CTM-based approach, which contains DCDA, CBPA, and DRA. We compare the results with another approach, in which CTM only runs DRA. As shown in Figure 8(a), the simulated scenario has 50 relay robots with 6 coverage holes. Some simulation setup information are the same as that summarized in Table 3. The difference are shown in Table 4. Similarly, all relay robots, pursuer and target apply IEEE 802.11 as their MAC layer protocols. Relay robot network uses flood as network layer protocol, no special routing protocol is used. Figure 8(b) shows the tracking path using CTM, while Figure 8(c) shows the tracking path using DRA. Comparing the tracking strategies using CTM and DRA, we can observe that the total target moving distance in CTM is smaller than in DRA in this scenario. Hence, P can catch Q with a much shorter time using CTM.

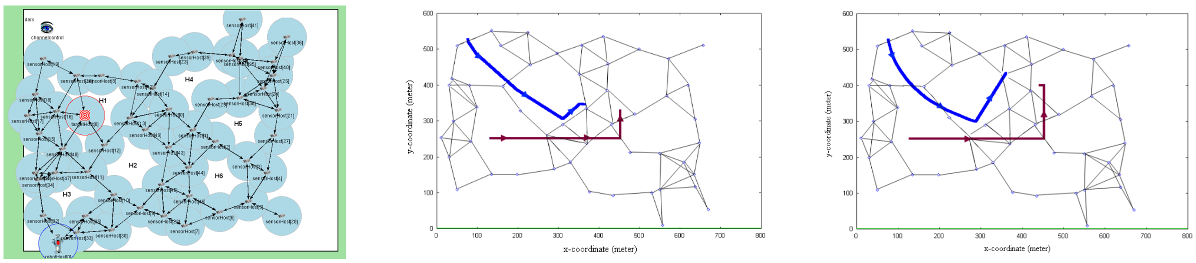
6. CONCLUSIONS

In this paper, we define the problem of target tracking in a multiple relay robot network that contains coverage holes. To effectively track the moving target, or to minimize the total time consumed by pursuer and total traveling distance of target, we create an efficient strategy that makes a coverage hole as a trap for the target. We propose a model for such scenario named Coverage-Hole Trap Model (CTM). We use DCDA to discover the coverage holes and introduce a pursuer tracking method called Coverage-Hole Based Pursuer Algorithm (CBPA). CBPA is a prediction-based algorithm



(a) (b) (c)

Figure 7: (a)Average total time consumed (b)Average total traveling distance of target (c)Percentage of tracking failure



(a) (b) (c)

Figure 8: (a)Simulated network with 50 relay robots with 6 coverage holes (b)Tracking path using CTM: Pursuer (blue) and Target (red) (c)Tracking path using DRA: Pursuer (blue) and Target (red)

for the pursuer utilizing the information about the target and the coverage holes received from the relays. The results of the simulations performed show that it is feasible to use CTM in a sensor network with coverage holes.

As the relay robots used in our work are static, it is promising to develop an application-specific model that let some relay robots move to create a new coverage hole based on the target moving speed, direction, and the pursuer's acknowledgment to help pursuer relocate target position. In this paper, the coverage hole area surrounded by relay robots and the border is not considered in the hole detection part. The border discovering strategy for tracking in such coverage holes should be taken into account in the future work. Lastly, it is also a challenge to increase the number pursuer and target in the network.

7. REFERENCES

[1] P. Balister, Z. Zheng, S. Kumar, and P. Sinha. Allowing coverage holes of bounded diameter in wireless sensor networks. In *IEEE INFOCOM*, 2009.

[2] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar. Coverage preserving redundancy elimination in sensor networks. In *IEEE SECON*, 2004.

[3] P. Corke, R. Peterson, and D. Rus. Finging holes in sensor networks.

[4] R. Ghrist and A. Muhammad. Coverage and hole-detection in sensor networks via homology. 2005.

[5] C. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *In Proceedings of the 2nd ACM WSNA03*, 2003.

[6] H. T. Kung and D. Vlah. Efficient location tracking using sensor networks. In *WCNC*, 2003.

[7] X. Li, D. K. Hunter, and K. Yang. Distributed coordinate-free hole detection and recovery. In *IEEE GLOBECOM*, 2006.

[8] C. C. Ooi and C. Schindelhauer. Energy-efficient distributed target tracking using wireless relay robots. In *In DARS08: Proc. of the 9th International Symposium on Distributed Autonomous Robotic Systems*. Springer-Verlag, 2009.

[9] S.Meguerdichian, F.Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor network. In *IEEE INFOCOM*, pages 1380–1387, 2001.

[10] A. Verma, H. Sawant, and J. Tan. Selection and navigation of mobile sensor nodes using a sensor network. In *IEEE International Conference on Pervasive Computing and Communications*, pages 41–50, 2005.

[11] G. Wang, G. Cao, and T. L. Porta. A bidding protocol for deploying mobile sensors. In *In 11th IEEE International Conference on Network Protocol ICNP 03*, pages 315–324, 2003.

[12] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *In Proceeding of the ACM SenSys 03*, pages 1615–1620, 2003.

[13] J. Wu and S. Yang. Coverage issue in sensor networks with adjustable ranges. In *ICPPWorkshops*, pages 61–68, 2004.

[14] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. UIUC, 2003.

[15] W. Zhang and G. Cao. Dynamic convoy tree-based collaboration for target tracking in sensor networks. In *IEEE Transcactions on Wireless Communications*, volume 3, 2004.