iLight: Indoor Device-Free Passive Tracking Using Wireless Sensor Networks

Xufei Mao*, ShaoJie Tang†, Xiaohua Xu†, Xiang-Yang Li†, Huadong Ma*

*Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, BUPT, Beijing, China
†Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
Emails: maoxufei@bupt.edu.cn, stang7@iit.edu, xxu23@iit.edu, xli@cs.iit.edu,mhd@bupt.edu.cn

***Abstract*—Target tracking is a main application of wireless sensor networks (WSNs), and has been studied widely [4], [10]. In this work, we study indoor passive tracking problem using WSNs, in which we assume no equipment is carried by the target and the tracking procedure is passive. We propose to use light to track a moving target in WSNs. To our best knowledge, this is the first work which tracks a moving object by using light sensors and general light sources. We design a novel probabilistic protocol (system) *iLight* to track a moving target and several efficient methods to compute the target's moving patterns (like height, moving speed etc.) at the same time. We implement and evaluate our tracking system *iLight* in a testbed consisting of 40 sensor nodes, 10 general light sources and one base station. Through extensive experiments, we show that *iLight* can track a moving target efficiently and accurately.**

I. Introduction

With respect to the tracking problems, there are two main directions. One direction (*e.g.*, [1], [2]) assumes that a target being tracked carries some assistant advice(s) (*e.g.*, wireless sensor nodes, RFID, PDAs) such that the target can be easily detected by other anchor sensor nodes. This is called device-based tracking. In contrast, the other direction assumes that the target being tracked is device-free, *i.e.*, the target carries no any assistant device and the tracking procedure is considered to be "passive". In this paper, we study the passive (device-free) tracking problem using WSNs. We propose to detect and track a target at an indoor environment using a group of light sensors with general light sources. Considering a security monitoring scenario in some place, *e.g.*, a museum, clearly, it is impossible to equip some device for an intruder.

Our main contributions are as follows. Firstly, to the best of our knowledge, this is the first work to use light sensors and general light sources to track a device-free target. Secondly, we propose a probabilistic tracking method to track a single target efficiently and accurately. Thirdly, we design several algorithms to compute the height and moving speed of a moving target with a surprisingly good accuracy and efficiency. Fourthly, we design and implement our device-free tracking system *iLight* using a WSN, consisting of 40 wireless sensor nodes, 10 general light sources and one base station (laptop with a wireless sensor node acting as a sink node). We conduct extensive experiments by testing our tracking algorithms on *iLight*. The experimental results show that *iLight* is not only able to compute the moving trajectory of a single target efficiently and accurately, but able to study the moving pattern (properties) of a moving target, like height, moving speed etc, which is the first WSN testbed that can achieve this. For example, our experimental results show that the average error of measured height of persons is around 2cm.

The rest of the paper is organized as follows. We first review related work in Section II. We define the problem and propose our main idea with a probabilistic approach in Section III. We conduct extensive experiments and present the experimental results Section IV. We conclude our work in this paper in Section VI.

II. Related Work

One direction of tracking problem using WSNs is called passive (Device-free) tracking problem (DfP) which was first defined by Youssef *et al.*, in [17]. They studied the feasibility of DfP and further discussed several research challenges regarding to the localization algorithms and infrastructure support. Tseng *et al.*, [15] studied the target tracking problem by using a mobile agent (a wireless sensor node) which can follow the target by hopping from sensor to sensor. Some similar work was dong by Kung *et al.*, in [11]. However, the work in [11], [15] assume every sensor node has a sensing range and can detect the existence of a target accurately as long as the target falls into this sensor node, which is not realistic. Later, Zhang *et al.* [18], [19] and Yao *et al.*, [16] proposed to use RF-based method to track transceiver-free targets. Their main idea is to detect targets based on Small-Scale Fading effect (SSF). Later, Moussa *et al.*, [13] studied the performance of two DfP techniques, moving average (MA) and moving variance (MV) in a real environment. He *et al.*, studied and designed VigilNet [10], a large-scale sensor network system consisting of 200 XSM motes which tracks, detects and classifies targets. Their main work concentrated on studying the tradeoff between the real-time performance and the energy consumption, and assumed that each wireless sensor can detect the existence of target with high probability when the target falls into the sensing range of the wireless node. In [5], Dutta *et al.* concentrated on the hardware design to save energy consumption, hence prolonged the life time of large scale wireless networks. The main idea of work in [9] by He *et al.*, is to let wireless sensor nodes alternatively work and let several sensors work together to exclude false alarm such that the energy consumption is decreased. Das *et al.*, [8] studied the problem to track moving objects using a smart sensor network. Their work was mainly based on two assumptions. One is that a sensor node is able to detect the existence of any moving object as long as the object falls in its sensing range and the reading of the sensor exceeds some threshold value. The other assumption is that the sensor has already learned the sensor reading to distance mapping.

III. Problem Formulation, System Architecture, and Our Approaches

A. Problem Formulation

Given an indoor area, we want to track a device-free moving target inside this area using wireless sensor nodes. We are also interested in obtaining a number of attributes of a moving target, such as the moving speeds, the moving trajectories, the height of the target (typically a human being). Here, we assume that each wireless node is equipped with 1) one light sensor: which can sense the level of light around it, and 2) at least one transceiver: which can communicate with neighbor nodes such that all wireless nodes construct a connected WSN. We assume that wireless sensors will be placed in a 3D domain and the geometric positions of sensor nodes can be obtained easily when we deploy light sources and wireless sensor nodes. For simplicity, from now on we call the sampled light level of a wireless node as photo value of this sensor node.

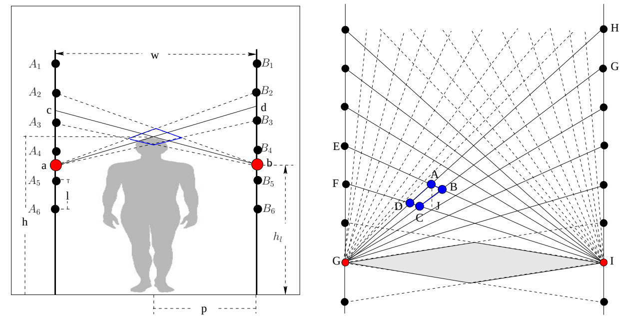
B. Finding Effective Detection Method

Through experiments, we found that a light sensor is very sensitive to the change of the light level of the environment around it. For example, the reading of the light sensor is around 50 when the only light source of the light sensor is a general 40w lamp which is 5 meters away. We further let a person go across between the light source and the lamp, and found that the reading of the light sensor drops to (around) 10 obviously. One thing needs to be mentioned that the Euclidean distance between the general light source and the light sensor should be not too large depending on the illumination intensity of the light source. For example, when the Euclidean distance between the lamp (with 40w used in our experiment) and a light sensor is more than 12 meters, the light sensor cannot tell exactly whether there is an obstacle between it and the light source due to the light attenuation and hardware constraints. Obviously, using special types of light sources (e.g. laser-like) can increase the valid distance of light beam and eliminate this kind of problem while increasing the cost as well. Our objective is to using general light sources to do tracking without much cost.

C. Computing Position and Height of the Target

In order to distinguish some target from others, we usually need to obtain some special characters of it, like height and so on. In this section, we will show how we compute the position and height of a target.

As we know, when a target stays or goes across the line between a light source and a sensor, the photo value of the sensor node will be affected such that we know the target will be somewhere along this line. Clearly, if we could find multiple such lines (going through some sensor node and some light source) at the same time, we may find the position of



(a) (b)

Fig. 1. (a) Black nodes denote two group of sensors and two red nodes (a and b) are two light sources. The height of the person and the altitude of light sources are h and h_l respectively. The distance between the person and the right side wall is p. The distance between each two adjacent sensor nodes in the same group is d. (b) The bottom left corner has coordinate $(0, 0)$ and the coordinates of four vertices (A, B, C, D) of marked quadrilaterals are (X_A, Y_A), (X_B, Y_B), (X_C, Y_C) and (X_D, Y_D) respectively.

the target more accurately by finding the intersection points of those lines. We call such an intersection point where the tracking system can locate a target accurately as a "catching point". In other words, the system is able to compute the position of a (moving) target when the target exists at a "catching point". We define the "catching point" of a target o as (x_o, y_o, t) where (x_o, y_o) is the position (in 2-dimension from a vertical view) of target o at time t. Actually, we will show that besides the position of the target, we can further compute the height of the target by carefully arranging the positions of light sensors and light sources.

Our main idea is as follows. We first divide sensor nodes into groups and consider each group as a cluster and the wireless sensor nodes inside a group will act as different roles (in order to do data collection and time synchronization, etc., which will be illustrated later). Next, we put two groups of sensors (face to face) on both sides of monitored area respectively. Here, a group of sensors will be hanged in a vertical line at one side of the monitored area. In addition, we put one light source at each side such that the light source can irradiate the group of sensors at the other side of the monitored area. See Fig. 1(a) for illustration. For simplicity, we assume the distance between two adjacent sensor nodes in the same group is same. Actually, our result will not be affected when the distance between any two adjacent sensor nodes in the same group is different.

In the case shown in Fig. 1(a), a group of sensors $\{A_1, A_2, \cdots, A_6\}$ are deployed on the left side and a group of sensors $\{B_1, B_2, \cdots, B_6\}$ are deployed on the right side. In addition, two general light sources which can irradiate the group of sensors at the other side are deployed on each of two sides (a and b respectively). For simplicity, we use the name of a wireless sensor node (resp. a light source) to denote its position as well. When a person with height h at position p comes across between these two groups of sensors, the photo values of sensors below point c (resp. d) will decrease extremely and the photo value of A_1, A_2, B_1, B_2 will remain almost same. Clearly, if we can find the position of the solid line segments (ad and bc) accurately, we will be able to compute the position and the height of the target (person) accurately. Unfortunately, it is not always possible to find the accurate position of this two solid lines in a real application scenario since the sensors are deployed discretely on both sides. By analyzing the photo values of sensor nodes, we only know that point c(resp. d) exists on the segment A_2A_3 (resp. B_2B_3) in the case shown in Fig. 1(a). Based on above analysis, we can make sure that the top point of the person will approximately exist in the quadrilateral which is the intersection (shown in the Fig. 1(a)) of four line segments, aB_2, aB_3, bA_2, bA_3. Notice here, if the target is not a person, this may be not true.

Let us consider a more general case, in which we assume there are enough number of sensors in each group such that any target cannot influence all wireless sensors of both sides. Next, we draw a line from each sensor to its light source, clearly, the section will be partitioned into small quadrilaterals. See Fig. 1(b) for illustration. In the example shown in Fig. 1(b), there are some triangles which includes the light source. This is due to the position of the light source and will not influence our results.

Let us consider the quadrilateral (with vertices A, B, C, D) which contains the top point of the target. Here, we assume A is the tallest point and A, B, C, D are in clockwise order. See Fig. 1(b) for illustration. Assume the coordinates of four vertices A, B, C, D of marked quadrilaterals are (X_A, Y_A), (X_B, Y_B), (X_C, Y_C) and (X_D, Y_D) respectively (those coordinates can be computed since the position and the Euclidean distance from each sensor node and each light source to the ground is known). Next, for all the candidate points in quadrilateral $ABCD$, we choose the point with coordinates $(\frac{X_B+X_D}{2}, \frac{Y_A+Y_C}{2})$ as the top point of the target being tracked. Clearly, the point $(\frac{X_B+X_D}{2}, \frac{Y_A+Y_C}{2})$ has the minimum error bound for both computed height and position.

Next, we show that our method to compute the height and position of a target has error bound $\frac{d}{2}$ and $\frac{w}{2}$ respectively where d is the distance between two adjacent sensor nodes in a group and w is the width of our monitored area. It is not difficult to prove the following lemma.

Lemma 1: For any quadrilateral $ABCD$ (example shown in Fig. 1(b)) where A is the tallest point and A, B, C, D are in clockwise order, if point A's x-coordinate $X_A < \frac{w}{2}$ then $X_C < X_A$; if $X_A = \frac{w}{2}$ then $X_C = X_A$; otherwise $X_C > X_A > \frac{w}{2}$.

Based on Lemma 1, it is not difficult to obtain the following Lemma 2.

Lemma 2: The error bound between $\frac{Y_A+Y_C}{2}$ and the real height h is bound by $\frac{d}{2}$ where d is the Euclidean distance between two adjacent sensors in the same group.

Clearly, when the top point of the target exists in the quadrilateral (grey area shown in Fig. 1(b)) which contains the light sources, the error bound could be up to $\frac{w}{2}$, which is the worst case. Unfortunately, our method cannot avoid such quadrilateral as long as d is not equal to 0. Fortunately, we can decrease the area of such quadrilateral significantly by decreasing the distance between the light source and each of its two adjacent sensor nodes (higher and lower than the light source respectively). Our experimental results show that the position error in iLight is bounded in $\frac{w}{9}$ when we put the light source into the middle of two adjacent sensor nodes (in the same group)with Euclidean distance 20cm.

D. Increasing the Number of Potential Catching Points

As we have introduced in last section, besides all the catching points, the monitored area may have some blind area within which the system can not catch the moving target since there are no any or (not enough) sensor nodes deployed. Obviously, the amount of potential catching points depends on not only the total number of wireless sensor nodes we deployed but also the positions of wireless sensor nodes and light sources. Although increasing the total number of wireless sensor nodes and light sources can decrease the blind area to some extent, this increases the total cost as well. Actually, we will show that by choosing different type of light sources while keeping the total number of sensor nodes, we can increase the potential catch points extremely without increasing much cost.

Through experiments, we found that the TSR $S1087-01$ light sensor is very sensitive to the light changing of the environment even it has multiple light sources at the same time. Hence, we proposed to use light sources with wider beam in our approach such that the number of possible catching points increases. Our experiment results also verified this.

However, this may be arise another problem since the reading of a sensor nodes could be affected by multiple light sources such that photo values of multiple sensors will be affected when a target exists. We further propose our probabilistic approach to solve this problem.

E. Our Probabilistic Approach

The main idea of our probabilistic method is as folllows. We first partition the monitored area into cells and assign probabilities to different cells based on the collected data such that for any time slot, the cell with highest probability will be considered the position of the moving target. See Fig. 2 for illustration. As we can see from Fig. 2, each cell could be gone through by one or more links (lines) between some light sensor and some light source. For instance, the cell (blue rectangle) in Fig. 2 is gone through by link $(2, 9)$ and $(3, 7)$. After partition, we are able to assign different probability to different cells by collecting enough readings of sensors. For instance, if there are some nodes from group $2, 3, 7, 9$ reporting "catching" event to the base station, it is more possible that the moving target is in the blue cell since all other cells being gone through by link $(2, 9)$ has smaller probability to cause the readings of sensors (in group 3 and 7) to change. For simplicity, we say that a link (a, b) is "active" at time slot t if there are sensors from both group a and group b reporting "catching" events at time slot t. Noticing that, there are some cells that are not

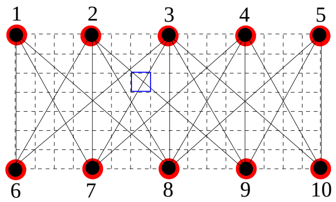


Fig. 2. A vertical view of partitioned area. Totally 10 (from 1 to 10) groups of sensors. Black (resp. Red) nodes denotes wireless sensors (resp. light sources). Solid lines denotes the possible links between light sensors and light sources. Slashed rectangles are partitioned cells.

gone through by any link, *i.e.*, these cells are in blind area in which a moving target cannot be detected. We can increase the number of sensor nodes to decrease such "blind" cells in order to increase the accuracy of "iLight" tracking system.

Based on the partition, we proposed the main idea of computing the trajectory of a moving target as follows. At the beginning, when no target exists, all cells have the same probability 0. Given any time t, if only two groups of sensor report "catching" events, then we use our proposed method (in Sec. III-C) to compute the location (and height etc.) of the target. Otherwise, when there are multiple links exist at time slot t, *i.e.*, there could be more than one potential possible position where the target could be. For each cell c and each link l, if l goes through cell c and l is active, we increase the probability for cell c (by 0.1 in our experiments) to contain the moving target. Next, we pick the center of the cell with the highest probability as the position of the moving target. Sometimes, two different cell may have the same highest probability. Through experiments we found that the main reason for two different cell have the same highest probability is due to the moving target's occupying two cells at the same time, in other words, these two cells are adjacent. In such case, we merge two cells into a big cell and consider the gravity center as the most possible position for the moving target. When two cells which are not adjacent to each other have the same highest probability (although this seldom happens according to our experimental results), we consider the cell which is closer to the position of the moving target at previous time slot has higher probability. The reason for us to do this is because we consider the moving target has regular moving speed, like the walking speed of a normal person. Since the average sample rate for a sensor is around 100 milliseconds and we consider the time period of a single time slot is 500 milliseconds, it is more possible that the cell which is closer to the position of previous time slot has higher probability. Actually, when the above assumption about the constraints of moving speed is not true, we can randomly pick up the center of one of two cells as the location of the moving object since we can continue to refine the position of the moving object by future readings of sensors. The probability of each cell will be reset to 0 after time slot t finishes. See Alg. 1 for details.

Algorithm 1 Computing Position of The target at Time Slot t

Input: Given all the readings collected by the base station at time slot t, all cells in set $\mathcal{C}$
Output: The position of the moving targets.

```
Obtain all active links based on all readings collected by the base station, assume the active link set is L.
if Only one active link exists then
    Compute and return the position by our method proposed in Sec. III-C
else
    for each cell c ∈ C do
        for each active link l ∈ L do
            if l goes through cell c, increase the probability of cell c to contain the moving target
    while any two cells c1 and c2 have the same highest probability do
        If c1 and c2 are adjacent to each other, merge c1 and c2 into big cell c12, C = C ⋃ c12 \ {c1, c2}; Otherwise, increase the probability of the cell which is closer to the position of target at time slot t − 1.
    Return the position of the (gravity) center of the cell with the highest probability
```

IV. Performance Evaluation

To illustrate the feasibility and performance of our tracking system, we implemented our tracking system and tested our algorithms in a real-life wireless sensor network.

A. System Design

The *iLight* tracking system consists of 41 wireless sensor node (one node will be used as the sink node connected to the base station), 10 general light sources and one base station (laptop). We divide all 40 sensors into 10 groups and each group has 4 sensors respectively. As we have introduced before, each group of sensors has the same coordinates (from a vertical view) but different Euclidean distance to the ground. In our test bed, we simply sort all sensor nodes in a group in lexicographical order such that the group leader is tallest. Table I summarize the main parameters of our *iLight* tracking system.

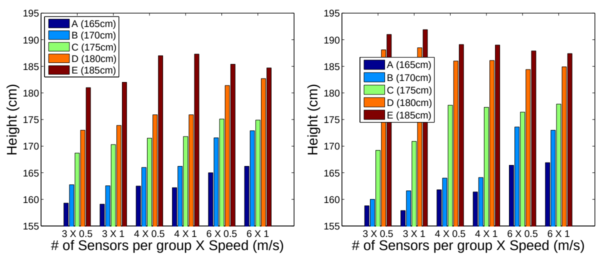
TABLE I
"iLight" System Parameters

Parameter	Value
# of TelosB sensor groups	10
Distance between two adjacent sensors in the same group	30 (resp. 20, 10) cm
# of sensors per group	3 (resp. 4, 6)
Distance between two groups at the same side	2.5 meters
Distance between two face-to-face groups	3 (resp. 4) meters
targets (height of people)	A(165cm), B(170cm), C(175cm), D(180cm), E(185cm)
Hight of a light source	160cm
Moving Speed	0.5 and 1 meter/sec
Size of a cell after partition	$0.5meter \times 0.5meter$

B. Experimental Results

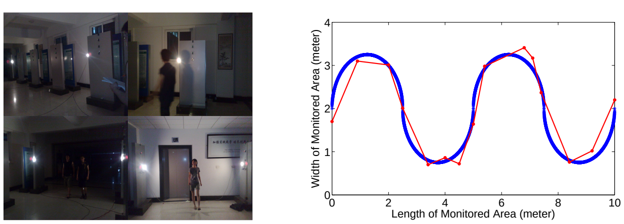
We choose 5 persons (namely A, B, C, D, E) with height 165cm, 170cm, 175cm, 180cm, 185cm respectively as our moving targets. We let each target go through the monitored area from different positions, *e.g.*, through the right middle or 1 meter to the right side respectively. We repeat each test case for 20 times and experimental results show the average values.

The following Fig. 3(a) (resp. 3(b)) shows our results for computing the heights of each of five targets when all targets move forward through middle (resp. 1 meter to the right side) with speed 0.5, 1 meter/sec respectively. As we can see in Fig. 3(a), with the increment of the number of sensor nodes per group, the error of each target's height decrease. When there is 3 sensors (with distance 30 cm) in a group, the maximum error is around 8cm. However, when we increase the number of sensor nodes to 6, the average error is refined to be within 2cm. In addition, the moving speed of a target does not influence the resultant height obviously since the sample rate of a wireless node in *iLight* is high enough to catch any passing target (human being in this case). The similar case happened in Fig. 3(b).



(a) (b)

Fig. 3. Experiment results for computing heights of different targets in different cases. (a) All targets move forward in the right middle. (b) All targets move forward in a line which is 1 meter to the right side.



(a) (b)

Fig. 4. (a) Test beds and test cases. (b) he blue curve is the real moving trajectory of target C and thin red line segments are the computed trajectories by iLight tracking system.

The Fig. 4(a) shows some pictures of our testbed and some test cases. In this case, we drew four tangent half circles with radius 1.25m on the ground in the monitored area and let each of five targets walk following the curve (blue curve shown in Fig. 4(b)). The Fig. 4(b) shows both the real moving trajectory (blue curve) and the computed moving trajectory (red line segments) of one of moving targets, C. As we can see, the computed moving trajectory of C has higher accuracy in the middle compared with the moving trajectory at the two ends. This is because there are more "catching points" in the middle of the monitored area such that *iLight* can refine the tracking trajectory better.

V. Acknowledgement

This work is supported by the National 973 Project of China under Grant No.2011CB302701, the National Science Fund for Distinguished Young Scholars under Grant No.60925010 and the National Natural Science Foundation of China (No.60903072).

VI. Conclusion

In this paper, we studied device-free passive tracking problem at an indoor environment. We proposed several algorithms to study the moving patterns of targets efficiently. We designed and implemented our tracking methods in a real-life WSN consisting of 40 wireless sensor nodes and one base station.

References

[1] Ahmed, N., Dong, Y., Bokareva, T., Kanhere, S., Jha, S., Bessell, T., Rutten, M., Ristic, B., and Gordon, N. Detection and tracking using wireless sensor networks. *ACM SenSys* (2007).

[2] Al-Ali, A. R., Aloul, F. A., Aji, N. R., Al-Zarouni, A. A., and Fakhro, N. H. Mobile rfid tracking system. *ICTTA* (2008).

[3] Aoun, M., Schoofs, A., and van der Stok, P. Efficient time synchronization for wireless sensor networks in an industrial setting. *ACM SenSys* (2008).

[4] Bahl, P., and Padmanabhan, V. An in-building rf-based user location and tracking system. *IEEE INFOCOM* (2000).

[5] Dutta, P., Grimmer, M., Arora, A., Biby, S., and Culler., D. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. *ACM/IEEE IPSN* (2005).

[6] Elnahrawy, E., Li, X., and Martin, R. The limits of localization using rssi. *SECON* (2004).

[7] Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., and Levis, P. Collection tree protocol. In *ACM SenSys* (2009).

[8] Gupta, R., and Das, S. R. Tracking moving targets in a smart sensor network. *Proc. of the VTC Fall Symposium* (2003).

[9] He, T., Krishnamurthy, S., Stankovic, J. A., and Abdelzaher, T. An energy-efcient surveillance system using wireless sensor networks. *ACM MobiSys* (2004).

[10] He, T., Vicaire, P. A., Yan, T., Luo, L., Gu, L., Zhou, G., Stoleru, R., Cao, Q., Stankovic, J. A., and Abdelzaher, T. Achieving real-time target tracking using wireless sensor networks. *RTAS* (2006).

[11] Kung, H. T., and Vlah, D. Efficient location tracking using sensor networks. *WCNC* (2003).

[12] Maroti, M., Kusy, B., Simon, G., and Ledeczi, A. The flooding time synchronization protocol. *ACM SenSys* (2004).

[13] Moussa, M., and Youssef, M. Smart cevices for smart environments: Device-free passive detection in real environments. In *IEEE PerCom* (2009).

[14] Patwari, N., and Agrawal, P. Effects of correlated shadowing: Connectivity, localization, and rf tomography. In *IPSN* (2008).

[15] Tseng, Y. C., Kuo, S. P., Lee, H. W., and Huang, C. F. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. *IPSN* (2003).

[16] Yao, Q., Gao, H., Liu, B., and Wang, F. Model: moving object detection and localization in wireless networks based on small-scale fading. In *ACM SenSys* (2008).

[17] Youssef, M., Mah, M., and Agrawala, A. Challenges: device-free passive localization for wireless environments. In *ACM MobiCom* (2007).

[18] Zhang, D., Ma, J., Chen, Q., and Ni, L. An RF-based system for tracking transceiver-free objects. In *IEEE PerCom* (2007).

[19] Zhang, D., and Ni, L. Dynamic Clustering for Tracking Multiple Transceiver-free Objects. In *IEEE PerCom* (2009).