Detecting Movement of Beacons in Location-Tracking Wireless Sensor Networks

Sheng-Po Kuo[1], Hsiao-Ju Kuo[1], Yu-Chee Tseng[1,2], and Yueh-Feng Lee[3]
[1]Department of Computer Science
National Chiao-Tung University, Taiwan
[2]Department of Information and Computer Engineering
Chung-Yuan Christian University, Taiwan
[3]Computer & Communications Research Laboratories
Industrial Technology Research Institute, Taiwan
E-mail:{spkuo, srkuo, and yctseng}@cs.nctu.edu.tw and yuehfeng_lee@itri.org.tw

Abstract—**Localization is a critical issue in wireless sensor networks. In most localization schemes, there are beacons being placed as references to determine the positions of objects or events appearing in the sensing field. The underlying assumption is that beacons are always static. In this work, we define a new *Beacon Movement Detection (BMD)* problem. Assuming that there are unnoticed changes of locations of some beacons in the system, this problem is concerned about how to automatically monitor such situations and identify these beacons. Removal of such beacons in the positioning engine may improve the localization accuracy. Two schemes are proposed to solve the BMD problem. Finally, we evaluate how these solutions can improve the accuracy of localization schemes in case that there are unnoticed movement of some beacons. Simulation results show that our solutions alleviate 53% the decrease of positioning accuracy caused by the exceptional beacon movement.**

Index Terms—**Context Awareness, Location-Based Service, Pervasive Computing, Positioning, Wireless Sensor Network.**

I. Introduction

Recently, we have seen significant progress in the areas of wireless ad hoc and sensor networks. Ad hoc networking technologies enable quick and flexible deployment of a communication platform. A wireless sensor network typically adopts the ad hoc network architecture and is capable of exploiting context information collected from sensors. Many applications of wireless sensor networks have been proposed [3], [5], [6].

Sensor networks are promising in supporting context-aware and location-aware services. The success of this area may greatly benefit human life. One essential research issue in sensor networks is *localization*, whose purpose is to determine the position of an object or event. For example, the sentient system Bat [2] is composed of a set of sensors for 3D localization purpose. Sensors are installed at known positions such as ceilings, to measure the signal traveling time from a user badge to them. Then, the location of the badge is calculated by a triangulation algorithm. Localization by signal's angle of arrival is addressed in [8], [9]. In [9], ultrasonic sensors are used to estimate the location and orientation of a mobile device with the Cricket compasses. In [1], a distributed positioning system called AHLoS (Ad Hoc Localization System) is proposed, where some beacons are aware of their own locations while others are not. The former are used to determine the positions of the latter. A similar work based on a probability model is proposed in [10]. The RADAR system [4] uses machine learning and pattern-matching techniques to estimate the locations of WiFi-enable mobile devices.

In all the above localization systems, there are a set of *beacon sensors* (or simply *beacons*), which are at fixed locations and periodically send out or receive short broadcast packets to estimate other objects' locations by either triangulation or pattern-matching schemes. Based on such an infrastructure, this paper points out a new *Beacon Movement Detection* (BMD) problem that may happen in most localization systems based on beacons. Beacon movements are usual cases especially in wireless sensor networks. The topologies of their ad hoc infrastructures are prone to be changed because of unexpected external forces, such as those caused by the animals being monitored. This problem is concerned about how to automatically determine the unexpected change of locations of some beacons in the system. Movement of some beacons may affect the accuracy of the localization results. For example, in Fig. 1(a), we show how three beacons determine a target's position in typical triangulation approaches. However, if beacon b_3 is moved to the location marked in gray without being noticed, the system may incorrectly estimate the target's location as shown in Fig. 1(b). Note that the circle centered at b_3 has a radius equal to the distance from the real location of b_3 to the target. Also note that the results proposed in this paper is applicable to not only unnoticed movement of beacons, but also unnoticed appearance of interference/obstacles in the sensing field, which may affect the localization results.

The BMD problem involves two issues. First, we need to determine those beacons that are unexpectedly relocated. Second, the result has to be forwarded to the positioning engine to improve the localization accuracy. To solve the first issue, we will allow beacons to monitor each other to determine those moved beacons. We will propose two schemes. In the first *location-based* scheme, we try to calculate each beacon's

1-4244-0264-6/07/$25.00 ©2007 IEEE

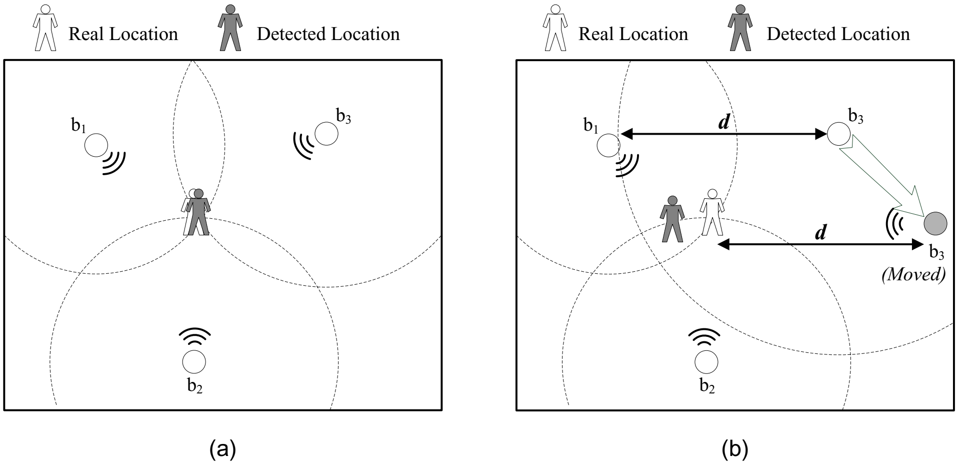


Fig. 1. An example of the Beacon Movement Detection (BMD) Problem.

current location and compares the result with its predefined location to decide if it has been moved. This scheme has high computation complexity, but is quite sensitive to noise. In the second *signal-strength-variation* scheme, the change of signal strengths of beacons will be exploited. This scheme has low complexity but perform much better.

The remainder of this paper is organized as follows: Section II gives a formal definition of the BMD problem. Section III presents our solutions to the BMD problem. We then evaluate the proposed algorithms and examine their capability to improve the localization accuracy in Section IV. Finally, Section V concludes on this work.

II. Problem Definition

We are given a sensing field, in which a set of beacons $B = \{b_1, b_2, ..., b_n\}$ are deployed for localization purpose. Periodically, each beacon will broadcast a HELLO packet. To determine its own location, an object will collect HELLO packets from its neighboring beacons and send a signal strength vector $\boldsymbol{s} = \langle s_1, s_2, ..., s_n \rangle$ to an external positioning engine, where s_i is the signal strength of the HELLO packet from b_i ($s_i = 0$ if it cannot hear b_i). The positioning engine can then estimate the object's location based on $\boldsymbol{s}$ (for example, in the case of RADAR [4], $\boldsymbol{s}$ is compared against a location database obtained in the training phase based on a similarity measurement).

Suppose that a set of *unreliable* beacons $B_M \subset B$ are moved or blocked by obstacles without being noticed. The *Beacon Movement Detection (BMD)* problem is to compute a detected set B_D that is as similar to B_M as possible. The result B_D may be used to calibrate the positioning engine to reduce the localization error (for example, in the case of RADAR, the entry s_i in $\boldsymbol{s}$ may be ignored if b_i is detected to be unreliable).

To solve the BMD problem, we will enforce beacons to monitor each other from time to time. Let's denote the local *observation vector* of b_i at time t by $O_i^t = \langle o_{i1}^t, o_{i2}^t, ..., o_{in}^t \rangle$, where o_{ij}^t is b_i's observation on b_j at time t. The content of an observation will depend on the corresponding BMD scheme (refer to Section III). We use the observation vector at time $t = 0$ to represent the original observation when no beacon is moved. The *observation matrix* is denoted by $O^t = \langle O_1^t, O_2^t, ..., O_n^t \rangle^T$. Given O^t, the BMD engine is responsible of calculating a set B_D. The result is then sent to the calibration algorithm in the positioning engine. Fig. 2 illustrates our system model.

Considering the following reasons, we define the *tolerable region* R_i of each beacon b_i as the geographic area within which movement is acceptable. First, radio signal tends to fluctuate from time to time. Second, ignoring the data of a beacon in the location database will decrease the localization accuracy due to less beacons helping the localization procedures. So the slight movement activities should be omitted. As a result, the unreliable set B_M only contains those beacons which are moved out of their tolerable regions. For simplicity, tolerable regions are assumed to be circles centered at beacons of the same radius. The size of tolerable regions is application-dependent, which is beyond the scope of this work.

Ideally, we would expect $B_M = B_D$. However, for many reasons this cannot be achieved. For ease of discussion, we define two events. A *hit event* is obtained on a beacon b_i if $b_i \in B_M$ and the BMD engine also determines that $b_i \in B_D$. A false event is obtained on b_i if $b_i \notin B_M$ but $b_i \in B_D$.

III. Beacon Movement Detection Algorithms

To solve the BMD problem, we propose two detection schemes, namely *location-based* and *signal-strength-variation* schemes. These schemes differ in their local processing rule of beacons and the decision algorithm at the BMD engine. In the location-based scheme, each beacon reports its actually received signal strengths, which are used by the BMD engine to compute each beacon's current location and to compare against its original location. In the signal-strength-variation scheme, each beacon locally decides if some neighboring beacons have moved according to a threshold of signal strength change and reports its binary observation to the BMD engine.

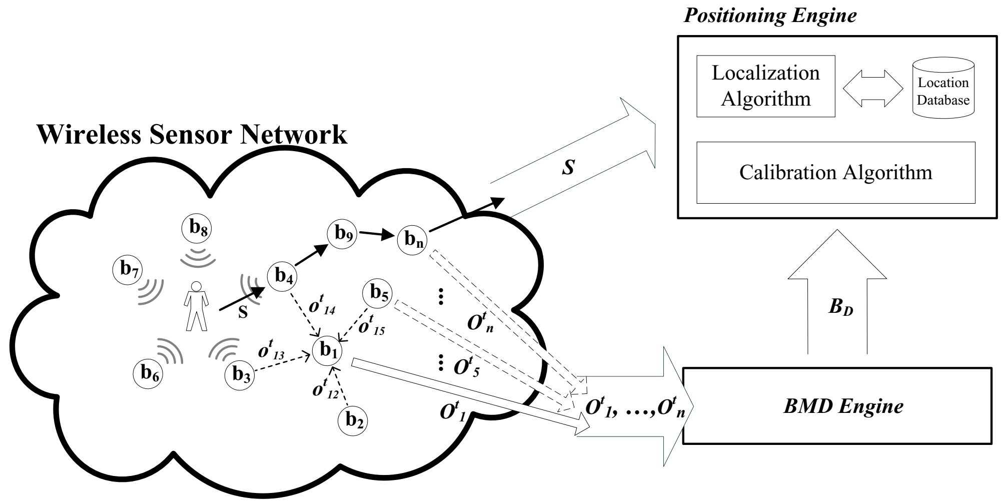


Fig. 2. The system model.

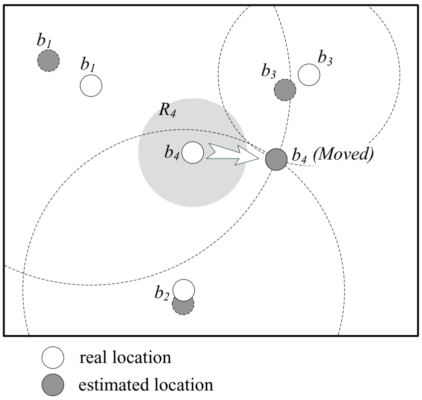


Fig. 3. An example of movement detection in the location-based scheme where b_4 is the only beacon being moved. A trilateration technique is used in this example.

A. Location-Based (LB) Scheme

The location-based scheme utilizes localization techniques to monitor the movements of beacons. Techniques such as trilateration or pattern-matching can be used in the BMD engine. Each beacon is in charge of reporting their observed signal strength values to the BMD engine. Let the observed signal strength by b_i on b_j at time t be s^t_{ij}. The observation o^t_{ij} of a beacon b_i on b_j at time t is defined as $o^t_{ij} = s^t_{ij}$. In this scheme, we assume the BMD engine knows the initial location of each beacon. The engine then estimates the position of each beacon through any localization technique. Let the estimated location of b_j at current time t be ℓ^t_j. If ℓ^t_j is out of the tolerable region R_j, then b_j is determined to be unreliable.

An example using the trilateration technique is shown in Fig. 3. Beacon b_4 is moved out of its tolerable region R_4. Since beacons b_1, b_2, and b_3 are unmoved, they can help to determine b_4's new location. One thing worthy of mentioning is that because of b_4's movement, the estimated location of b_1, b_2, and b_3 may also be changed by a certain degree. So the tolerable regions need to be defined carefully. As shown by our simulation results, the location-based scheme is too sensitive to any movement, and thus does not perform well.

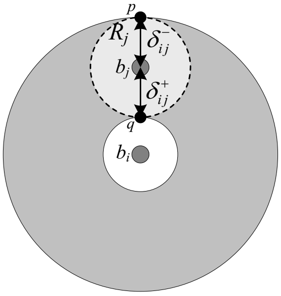


Fig. 4. Determining thresholds δ^+_{ij} and δ^-_{ij} by the tolerable region R_j of b_j in the signal-strength-variation scheme.

B. Signal-Strength-Variation (SSV) Scheme

In the location-based scheme, we report the observations according to the received signal strengths directly. It is sensitive to any slight movement. Hence, the signal-strength-variation scheme hides the information of signal strengths and just reports binary observations to the BMD engine. Similar to the location-based scheme which assumes that beacons can measure the signal strengths of `HELLO` packets from their neighbors, the signal-strength-variation scheme asks each beacon b_i to evaluate the amount of signal strength change of each neighboring beacon b_j. Let the observed signal strength by b_i on b_j at time t be s^t_{ij} ($t = 0$ means the initial observed signal strength). The observation o^t_{ij} of b_i on b_j is

$$o^t_{ij} = \begin{cases} 1, & \text{if } (s^t_{ij} - s^0_{ij} \geq \delta^+_{ij} \text{ or } s^0_{ij} - s^t_{ij} \geq \delta^-_{ij}) \text{ or} \\ & (b_j \text{ moves into or out of } b_i\text{'s coverage}) \\ 0, & \text{otherwise,} \end{cases}$$

where δ^+_{ij} and δ^-_{ij} are the pre-defined thresholds of signal strength variations.

The thresholds δ_{ij}^{+} and δ_{ij}^{-} of each beacon b_i can be determined by the tolerable region R_j of b_j. Let locations p and q be the farthest and nearest locations in R_j with respect to b_i (refer to Fig. 4). If the expected signal strengths of HELLO packets from a beacon at p and q are s_p and s_q, respectively, then $\delta_{ij}^{-} = s_{ij}^{0} - s_p$ and $\delta_{ij}^{+} = s_q - s_{ij}^{0}$. As can be seen, as long as b_j moves within the belt-like gray region, b_i will not report a movement event.

Because we assume that beacon movements are regarded as accidents, we try to select a set B_D that contains as few beacons as possible. First, we transform matrix O^t to an undirected *observation graph* $G_O = (V, E)$, where $V = \{b_j | O^t[i,j] = 1 \text{ or } O^t[j,i] = 1, 1 \leq i \leq n, 1 \leq j \leq n\}$ and $E = \{(b_i, b_j) | O^t[i,j] = 1 \text{ and } O^t[j,i] = 1\}$. Note that G_O is not an empty graph which means $V \neq \emptyset$. Second, observe that if $O^t[i,j] = 1$ and $O^t[j,i] = 1$, then at least one of b_i and b_j has been moved. Therefore, the BMD problem can be regarded as a *vertex cover* problem [7], whose goal is to find the smallest set $V' \subseteq V$ such that for each $(b_i, b_j) \in E$, $b_i \in V'$ or $b_j \in V'$.

In this scheme, two algorithms are proposed. The first one, called *Enumerate-SSV*, is only presented here for reference purpose. From graph G_O, we first construct all minimum vertex covers (since this problem is NP-complete, this step could be very costly). Among all solutions, the one with the lowest *stability* level is selected. The stability of a beacon b_i is

$$stability(b_i) = \frac{NPos(i)}{Pos(i)},$$

where $NPos(i)$ is the set of neighboring beacons of b_i that positively report that b_i has not been moved, i.e., $NPos(i) = \{b_j | o_{ji}^{t} = 0, b_j \text{ can hear HELLO packets from } b_i \text{ originally}\}$, and $Pos(i)$ is the set of all beacons that positively report that b_i has been moved, i.e., $Pos(i) = \{b_j | o_{ji}^{t} = 1\}$. The stability level of a vertex cover is the sum of stability values of all beacons in the cover set. Then the vertex cover with the lowest stability level is selected as our B_D.

Considering the above algorithm is quite costly when the problem scales up, the second algorithm *Greedy-SSV* adopts a heuristic approach as follows. If a beacon b_i's degree in G_O is higher, it is more suspicious to be moved. So the algorithm sorts the vertices in G_O according to their degrees of uncovered edges in descending order, and then examines them one by one. A node is included in B_D if any edge incident to it has not been covered.

IV. Simulation Results

In this section, we present our simulation results to evaluate the proposed schemes. The performance metrics include the probabilities of hit and false events. We also use the results to calibrate the positioning engine and measure the localization error when a localization scheme is applied (refer to our system model in Fig. 2). Experiments are conducted under different conditions, such as the ratio of moved beacons, the maximum movement distance, and the noise degree of the environment.

A. Simulation Model

The sensing field is a 500m×500m square area. There are 25 beacons randomly deployed on the field with a distance restriction of at least 20 meters [1]. Moved beacons are chosen randomly and a parameter *moved ratio (MR)* is used to control the number of moved beacons. The moving distance is uniformly distributed between 0 and a parameter *moved degree (MD)*. The tolerable region of each beacon is a circle centered at the beacon with a radius of 20 meters. Note that due to the definitions of tolerable regions, only part of the moved beacons will be considered moved.

The signal propagation of HELLO packets are modeled by a *log-distance path loss model* [11], where the path loss of a distance d is

$$PL(d) = PL(d_0) + 10\alpha log_{10}(\frac{d}{d_0}) + X_\sigma,$$

where d_0 is a reference distance, α is the path loss exponent typically ranging from 2 to 6, and X_σ is a zero-mean Gaussian random variable with a standard deviation σ. Also, the receiver sensitivity is -100 dBm (signal lower than this value is not detectable by a receiver). The default parameter settings are: transmit power $P_t = 17$ dBm, reference path loss $PL(d_0) = 41.5$ dBm, path loss exponent $\alpha = 3.5$, and $\sigma = 2.13$ dBm. All results are from the average of 50 experiments. To reduce the influence of noise, signal strength is calculated from the average of 10 HELLO packets.

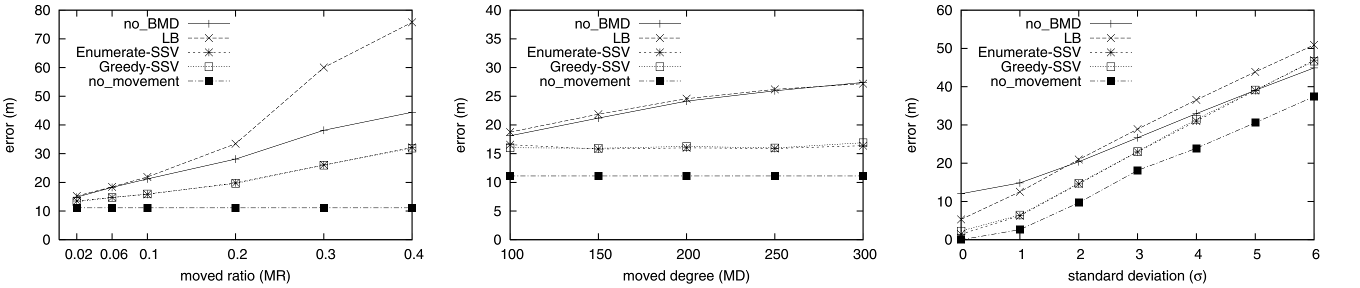
B. Evaluation of Localization Accuracy

After determining the moved set B_D, the set will be sent to the positioning engine to calibrate the location database. In the following, we will assume the pattern-matching-based localization algorithm [4], where the location database contains the signal vector $\langle v_1, v_2, \ldots, v_n \rangle$ of each training location in the sensing field, where v_i is the averaged signal strength of beacon b_i, $i = 1, 2, \ldots, n$. For calibration purpose, we will ignore the element v_i for each b_i that is determined to be in B_D during the localization procedure. Clearly, this will reduce the accuracy of localization. However, if the moved beacons are not ignored, the error may be even higher. In the following, we will evaluate how our schemes can reduce localization errors due to moved beacons. In our experiment, there are 25×25 training points and 25×25 test points. Then, the average positioning error of the 625 test points is recorded.

We compare our results against the *no_movement* case, where no beacon is moved, and the *no_BMD* case, where there are unnoticed movements of some beacons but no special action is taken. The former is only used as a reference.

Fig. 5(a), (b), and (c) show the average positioning errors under different *MR*, *MD*, and σ, respectively. The results demonstrate that the signal-strength-variation scheme

[1] The restriction is to avoid some beacons being placed too crowded, thus reducing the detection capability of the network. When a scenario is generated not satisfying the restriction, it will be discarded and we will regenerate another scenario.



(a) (b) (c)

Fig. 5. Comparison of average localization errors by varying: (a) *MR* (*MD*=150m), (b) *MD* (*MR*=0.1), (c) σ (*MD*=150m, *MR*=0.1).

generally performs better than the other schemes. Surprisingly, due to its high occurrences of false events, the *LB* algorithm's errors are quite unacceptable, sometimes even worse than the *no_BMD* case. Considering the *Enumerate-SSV* and *Greedy-SSV* algorithms proposed in the signal-strength-variation scheme, we can observe that the former does not always perform better than the latter. This is because we assume that only a small number of beacons are moved, and thus the *Enumerate-SSV* algorithm will try to find the smallest B_D. However, this assumption is not true any more when a lot of beacons are moved. Furthermore, considering that the *Enumerate-SSV* algorithm are computationally infeasible in a large scale network, the *Greedy-SSV* is more preferable in general cases.

To model the error recovery capability of a BMD algorithm, we propose the following metric:

$$ERC(algorithm) = \frac{\text{error}_{\text{no\_BMD}} - \text{error}_{algorithm}}{\text{error}_{\text{no\_BMD}} - \text{error}_{\text{no\_movement}}} \times 100\%.$$

Ideally, an *ERC* of 100% is expected. However, this is unlikely to be achieved because some data are ignored in the location database. For example, when $MR = 0.1$, $MD = 150$, and $\sigma = 2.13$, the *ERC* values are -6.71%, 53.53%, and 53.28% for *LB*, *Enumerate-SSV*, and *Greedy-SSV*, respectively.

V. Conclusions

In this paper, we define a new beacon movement detection (BMD) problem in wireless sensor networks for localization applications. This problem describes a situation that some beacon sensors which participate in the localization procedure are moved unexpectedly. The result is a reduced localization accuracy if we disregard this situation. We propose to allow beacons to monitor each other to identify the moved beacons. Two schemes are presented to solve the BMD problem. Some heuristics are proposed by mapping the BMD problem to the vertex-cover problem. Localization accuracy of the proposed schemes are obtained through simulations. It is showed that the best heuristic, *Enumerate-SSV*, has an error recovery capability of 53%. For general cases, the *Greedy-SSV* balances the tradeoff between the localization accuracy and computation complexity. As to future work, it deserves to further investigate the BMD problem if there is some trust model among beacons.

VI. Acknowledgement

Y. C. Tseng's research is co-sponsored by Taiwan MoE ATU Program, by NSC grants 93-2752-E-007-001-PAE, 96-2623-7-009-002-ET, 95-2221-E-009-058-MY3, 95-2221-E-009-060-MY3, 95-2219-E-009-007, 95-2218-E-009-209, and 94-2219-E-007-009, by Realtek Semiconductor Corp., by MOEA under grant number 94-EC-17-A-04-S1-044, by ITRI, Taiwan, under grant number 6352BA5131, by Microsoft Corp., and by Intel Corp.

References

[1] A. Savvides, C.C. Han, and M.B. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM/IEEE MOBICOM*, 2001.

[2] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *Computer*, 34(8):50 – 56, 2001.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[4] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *IEEE INFOCOM*, pages 775–784, 2000.

[5] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38V45, 2004.

[6] A. Cerpa, J. Elson, D. Estrin, L. Girod, and M. Hamilton. Habitat monitoring: Application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communications*, Apr. 2001.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[8] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS) using AoA. In *IEEE INFOCOM, San Francisco, CA.*, 2003.

[9] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. J. Teller. The cricket compass for context-aware mobile applications. In *ACM/IEEE MOBICOM*, pages 1–14, 2001.

[10] V. Ramadurai and M. L. Sichitiu. Localization in wireless sensor networks: A probabilistic approach. In *Int't Conf. on Wireless Networks (ICWN)*, pages 275–281, June 2003.

[11] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.