ART: A NOVEL LOCALIZATION METHOD FOR MANAGING UNCERTAINTY REGIONS IN WIRELESS FADING-SIGNAL SENSOR NETWORKS*

Chao-Chun Chen[1], Da-Chung Mao[2], Chiang Lee[2]
[1]Department of Information Communication
Southern Taiwan University of Technology, Taiwan, R.O.C.
chencc@mail.stut.edu.tw
[2]Department of Computer Science and Information Engineering
National Cheng-Kung University, Taiwan, R.O.C.
{amao,leec}@imus.csie.ncku.edu.tw

ABSTRACT

Localization plays a critical role in sensor based systems. Sensors that are location aware broaden the types of sensor applications. Sensor environment contains interferences and background noises, which make the current distance estimation schemes difficult to the achieve high accuracy. This paper proposes an aggressive rectangle truncation (ART) localization algorithm for locating objects in a fading-signal sensor environment. The paper further presents a location verification mechanism to increase the accuracy and consistency of the estimated object location. Finally, we conduct real-world experiments to show that ART algorithm implemented on Mica2 motes outstands the general triangulation algorithm.

Keywords: Sensor networks, fading-signal, localization, verification mechanism.

1 INTRODUCTION

The emerging sensor technologies have pushed upward the trend of using a wide variety of sensor applications. Utilizing sensor helps to automate tasks and save a great deal of human resources. A typical wireless sensor comprises a processor, transmission components, sensing components, and battery [1]. There are many sensing component, e.g., light, temperature, sounds, pressure, and etc, being developed and used popularly in sensor networks to suit a large number of application's needs [11]. The data being sensed and stored by sensors vary from discrete to continuous and presence of targets to the measurements of the objects.

One of the leading research areas is localization, which is a fundamental technique in many sensor applications, such as moving object tracking, motion capturing systems, security surveillance, and etc [1]. Inside those sensor systems, sensor nodes must be location aware in order to specify the location of event occurrences [18].

Two major components for localization are the location aware beacons and distance estimation protocols. The existing protocols face the similar problems on distance estimation from the beacons, sensors that are location aware, to the sensed object. This is due to not only the limitation on the hardware accuracy, but also the existence of background noise, radio signal reflections and fading radio signals. Thus, designing high accuracy localization algorithms is a great challenge in sensor networks.

Many of the previous works on localization are based on the triangulation method. The main factors of triangulation localization are distance estimations from the sensors to the moving objects and location awareness of the sensor nodes in the sensor network. The triangulation algorithm combines these two pieces of information to compute the object location. Many distance estimation schemes are proposed based the received signal strength indicator (RSSI) that is used to measures the attenuation of the radio signal as an indicator for the corresponding distance measurement. However, localization algorithms using RSSI measurements suffer from untrustful distance measurements provided by sensor nodes within the radio range. When the distance estimation is applied to triangulation algorithm, (m+1) or more quadratic equations are applied to solve an m-dimensional point, which is the final location estimation. However, the distance estimations, from every sensor, on a stationary object can vary from time to time due to fading-signal environments. Thus, the uncertainty in the distance estimations could generate different deviation in distance from each beacon, leading to the quadratic equations obtaining absurd location estimates and even becoming unsolvable due to extravagant distance estimation from sensors. Consequently, obtaining a point from

*This work is supported by National Science Council of Taiwan (R.O.C.) under Grants NSC95-2221-E-006-208, and NSC95-2221-E-006-206-MY3.

these quadratic equations with fixed radius becomes highly unreliable. As a result, triangulated-based methods provide less accurate location estimations due the radio signal interferences and the backgrounds noises in the fading-signal wireless sensor network.

In this paper, we propose an uncertainty-region management scheme to estimate the location of a target based on the fading-signal characteristic of a wireless sensor network. The fading-signal environment produces unstable RSSI received by the sensors in the sensor network, which means that signal attenuates in a nonuniform way, incurring uncertainty converting from the RSSI to the distance estimation. We propose an Aggressive Rectangle Truncation (ART) localization algorithm that is intentionally designed for handling the uncertainty regions created by the RSSI fluctuation to estimate the target locations. The main idea of our algorithm is to manage and characterize the uncertainty region by creating a rectangle to represent the uncertainty region in order to provide the target location estimations. ART exploits the correlation between the uncertainty region (e.g., overlapping and nonoverlapping) by iteratively estimating the uncertainty region with higher probability of target residency from the nearby participated sensors. ART adopts an aggressive method to prune away the possible target resident regions that have smaller probability, which in turn helps the rectangle to shrink in a faster rate. Therefore, ART can still work efficiently with a small number of beacons. We further design a threshold-based verification mechanism to enhance localization reliability and consistency. In comparison with the general triangulation algorithm, our experiments show that ART algorithm achieves a lower localization error and more concentrated estimated locations in various beacons deployment spaces.

The rest of this paper is organized as follows. Section 2 overviews some works related to ranged based localization methods. Section 3 describes the RSSI fluctuation by measuring the RSSI of a beacon in different angles. Section 4 presents the generation of the uncertainty region, and the proposed approach to handle the uncertainty region. Section 5 presents the procedure of ART in detail. We then develop the location verification mechanism in Section 6. In Section 7, we evaluate the performance of the ART with the location verification mechanism. Finally, we conclude our work in Section 8.

2 RELATED WORK

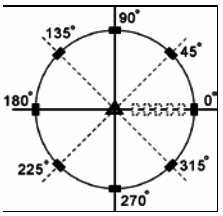
Many existing ranged-based localization algorithms have been proposed in wireless environments, and triangulated based methods are commonly used in the previous works [4, 18]. All of the localization methods require a distance estimation scheme, but the sensors do not provide components or functionalities that directly measure distance from the sensors to the objects. Instead, the sensors rely on other sensing capabilities that translate the sensed values to the distance estimation. Received signal strength indication (RSSI) [2] is one of the common distance estimation methods that measures the attenuation of the radio signal and translates the signal strength into distance. RADAR [4] that localizes the hotel guests in an indoor wireless environment has observed the nonuniform radiated pattern from the sensors, and the variation in the RSSI from different angles influences the distance estimation and consequently affects the accuracy of the object location estimation.

Galstyan et al. [9] proposed a localization method by imposing constraints on sensor nodes and iteratively narrarowing down the constraint to obtain the sensor location. The radio range of a moving beacon is known apriori, and the moving beacon broadcasts its locations to the sensor nodes in vicinity. No statistic model was employed on the nonuniform radiation pattern of the sensor nodes. Consequently, the uncertainty for the distance estimation is not considered and handled. Localization is carried through with a beacon node and thus, the information quality provided by the beacon cannot be differentiated.

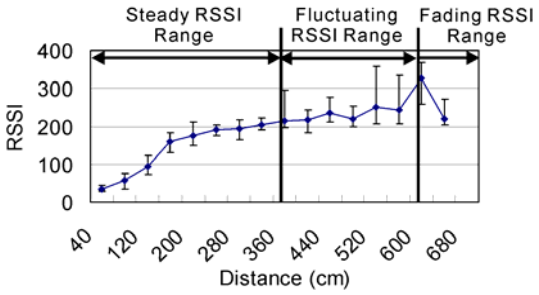
Most of the localization algorithm requires distance estimation. Other than RSSI measurements, several distance measuring schemes are also commonly employed to localize objects. Time of Arrival (ToA) [5, 12] or Time Difference of Arrival (TDoA) [10] techniques estimate the distance by measuring the propagation time of the messages from a set of reference points. Cricket [15, 17, 6] proposed a location support service based on TDoA. Cricket used both radio-frequency (RF) and ultrasounds to estimate the distance by measuring difference of arrival time between the RF and ultrasonic pulses. However, sound propagation speed changes relatively to the temperature, which also causes uncertainty in distance measurements. In a typical outdoor environment, temperature can vary over the day. The issue can be alleviated by observing statistical data of the temperature changes in days, and creating an adjustment for the sound propagation speed to improve the accuracy of the distance estimations. The Angle of Arrival (AoA) [14] approach incorporates directional sensors to collaboratively identify the target location by measuring the angle from the beacons to the target. The AoA approach could be interfered by the reflections of the signals. Therefore, the target location estimation could be greatly influenced in an environment where the multipath signal reflections are substantial.

3 PRELIMINARY

Before localizing an object, each sensor needs to have knowledge about its characteristic in signal receptions. Therefore, a training system is designed and one sensor was employed to measure and translate the signal to distance. Two Crossbow Mica2 sensors [7] were employed in the training process. Each Mica2 was equipped with a CC1000 device which can detect the RSSI when other sensors are within the communication range. Since the antenna of Mica2 has an unequal omnidirectional radiation pattern, a directional RSSI measuring scheme is employed to measure sensor RSSI in different angles [11, 16]. In order to demonstrate the granularity of RSSI variation in different angles, a beacon sensor that detects RSSI is placed in the middle of an open-ground indoor environment, and a training sensor is then placed in eight different angles in every 45◦, i.e., 45◦, 90◦, 135◦, 180◦, 225◦, 270◦ and 315◦, around the training sensor, shown by the dark gray rectangles in Figure 1(a). RSSI is measured every 40 cm until 680 cm for the sensor, as illustrated by the light gray dashed rectangles in the figure.



(a) Collecting Beacon RSSI Setup.



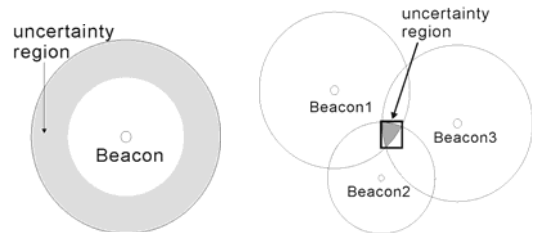
(b) The RSSI values in different distances.
Figure 1: RSSI fluctuation experiment.

Figure 1(b) shows the mapping function between the RSSI and the distances. Five hundred samples are withdrawn from every angle and every distance. The line in the graph shows the RSSI relative to the distance. The RSSI increases steadily from 40 cm to 440 cm, and hence, this range is identified as *Steady RSSI Range*. RSSI began to fluctuate greatly in the range after 440 cm, because while a sensor is using this training data to compare with the distance, for example, a sensed RSSI between 220 and 235 has four possible distance measurements. The possible distance ranges from approximately 400 cm to 520 cm and leads to a great deal of uncertainty in distance estimation. Also noticed that the difference between the maximum and the minimum RSSI at each position is enlarged after 360 cm, comparing to that before 360. The large deviation could result in the ranges of RSSI overlapping for different distance and a RSSI at a distance might be misrepresented by another. For instance, a RSSI at 520 cm could be indifferent to value at 600 cm. Therefore, RSSI ranging from 360 cm to 600 cm is identified as *Fluctuating RSSI Range*. After reaching the peak at 600 cm, the signal starts to decline and eventually becomes undetectable when the distance exceeds 680 cm. This RSSI range, identified as *Fading RSSI Range*, is generally unable to be used and should be discarded.

The RSSI of sensors could vary slightly. The antenna of a sensor differs from others of the same type, and creates unique RSSI while it is broadcasting radio signals. Due to this reason, each sensor is trained individually as opposed to using a set of training values for all the sensors.

4 UNCERTAINTY REGION MODELING

During localization, beacons work collaboratively to localize the object. In a radio-frequency (RF) based system, the distance measurements, from the beacons to the objects, depend on the correctness of the translation scheme from RSSI to distance. The distance estimation can be the visualized as the possible target locations lie on a perimeter around the beacon position with the radius of the estimated distance. The range of RSSI at a distance could partially overlap with the range of RSSI at another distance. If the range of a distance measurement overlaps with that of a lower distance, then the beacon could possibly overestimate the distance to the target, and vice versa. For example, the range of RSSI for distance at 400 cm overlaps with that at 480 cm in Figure 1(b). A RSSI with value 250 could potentially result in distance estimation of 480 cm, but the distance is actually 400 cm. Since the distance could be over- or underestimated, an upper bound of the estimated distance, for a received RSSI from a beacon, is establish to bind the possible target locations. Figure 2(a) shows the uncertainty region represents the possible target locations. The solid line indicates the upper bound for the received beacon information, but the actual target location is supposed to be some distance within the upper bound.



(a) Uncertainty region for one beacon. (b) Uncertainty region for multiple beacons.

Figure 2: Modeling uncertainty region.

As in the case of multi-beacon collaboration on localization, as shown in Figure 2(b), uncertainty region generated from multiple beacons could partially intersect. The smaller region generated by the intersection of the multiple uncertainty regions not only narrows down the possible target locations, but also increases the confidence of the target location, since the possible target locations within the uncertainty region is supported by three beacons in this case. The uncertainty region is formed by polygonal intersection area. As the RSSI could fluctuate in a fading-signal wireless sensor network environment, the polygonal uncertainty region is hard to be described in a practical sense. Therefore, it is feasible to adopt a rectangle to represent and provide an approximate uncertainty region.

5 AGGRESSIVE RECTANGLE TRUNCATION (ART) ALGORITHM

In this section, the aggressive rectangle truncation (ART) algorithm is proposed to shrink the uncertainty region by applying the available nearby beacons collaboratively to estimate the target location. The beacon information is processed sequentially, such that the uncertainty region becomes smaller and the possible target locations are refined after each iteration. Rather than creating a loose bound of the uncertain region, ART commits a more aggressive approach to eliminate a large area of the uncertainty region while preserving a well target location prediction. Figure 3 depicts the flowchart of the ART algorithm. While performing localization, the target sensor gathers and stores the beacon information which is the beacon positions and the estimated distance from the beacons to the target using RSSI to lookup the distance table. Logically, each piece of beacon information can be visualized as a circle centered at the beacon position with the estimated distance being the radius. ART algorithm begins after the target has collected the information from all participated beacons. The sequence of flow of our proposed algorithm starts by finding and sorting the intersections between the circles to provide a more reliable estimation. After creating the first rectangle, the process moves onto a recursive step to truncate or pull the rectangle, which manages the circumstances of beacons over- or underestimating the distance to the target. Overall, the ART algorithm is summarized in two steps: (i) prioritizing the beacons and (ii) truncating and pulling the rectangle. They are presented in the following subsections.

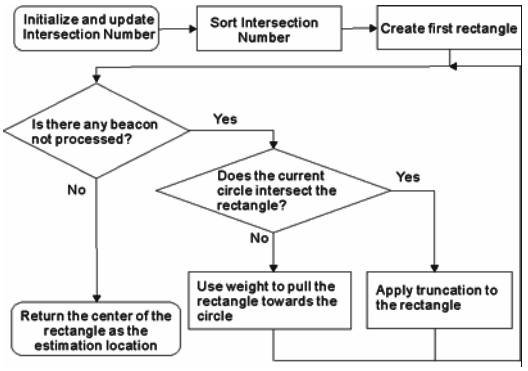


Figure 3: Flowchart for the aggressive rectangle truncation algorithm.

5.1 Beacon Priority

The basic concept of ART algorithm is to have each circle cutting the rectangle to retain the region of the rectangle that the participated beacons agree collaboratively on the possible target locations. Each beacon position is viewed as the center of a circle and the estimated distance can be visualized as the radius during the localization process. The intersection between a set of circles represents the possible locations of the target, that is, the uncertainty region generated from the participated beacons. The target maintains a list, called *Intersection Number List*, denoted as *INList*, which stores the number of intersection between a circle and other circles. The intersection number indicates the relative confidence of a circle that estimates the actual target location, because beacons are expected to intersect at a specific region during localization. Thus, a circle with a high intersection number provides a more reliable distance estimation. Since the beacon information in ART is processed in a sequential way, at the first step the *INList* is sorted such that more reliable circles can be used to estimate the target location with a higher priority. Among the beacons with the same number of intersections, the list is sorted by the radii in ascending order. Beacon information with smaller distance estimation generally offers a better distance estimation observed from Figure RSSI. Additionally, a smaller radius for the first rectangle creates a smaller uncertainty region with more confidence on the possible target locations. Figure 4 depicts an example of three beacons participating in the ART algorithm. The circle generated from the beacon B_2 intersects with the other circles and has a higher confidence in estimating target location. Beacon B_1

and B_3 both contain one intersection. B_3 has a smaller radius and thus, has a higher priority in localization process. The size of the uncertainty region correlates to the possible target locations and to the accuracy of the estimation. Subsequently, the beacon information is processed, one by one, following the sorted order to reduce the size of the rectangle so as to estimate the target location.

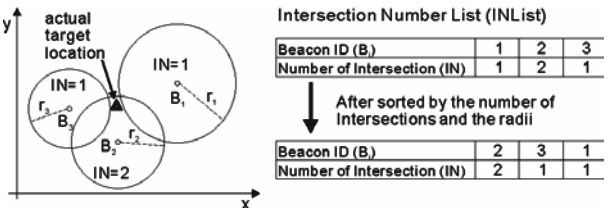


Figure 4: Intersection number list (INList).

5.2 Truncating and Pulling the Rectangle

After the *INList* has been sorted, the target location estimation begins by a recursive step that manages the rectangle. As each piece of the beacon information is processed in the *INList*, the area that most of the beacon uncertainty regions overlap represents a higher probability of the actual target location. Therefore, two operators, rectangle truncating operator (RTO) and rectangle pulling operator (RPO), are designed to progressively eliminate regions of the rectangle that have less chance of locating the target.

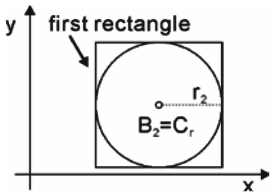


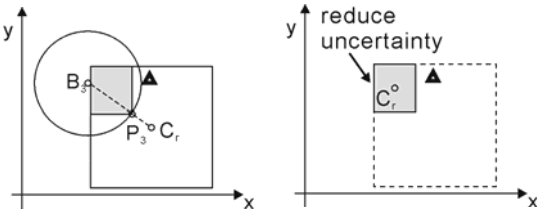
Figure 5: Uncertainty region for the first beacon in INList.

Initially, the first rectangle is established before the truncating and pulling rectangle step. The first rectangle is the minimum square that wraps around the first circle that is retrieved from the *INList*, as shown in Figure 5. The first piece of beacon information in the *INList* is most possible to capturing the actual target location within its uncertainty region, because the beacon has the highest intersection number among the participated beacons. Therefore, the rectangle wraps around the circle so that the whole area of uncertainty region is particularly reserved. In this case, the center of the rectangle, C_r, is equivalent to the center of the circle, B_2, and all edge lengths of the rectangle are equal to the diameter of the circle.

The distance received from the beacons could over- or underestimate the distance to the target due to the RSSI fluctuation, and results in circles that intersect or do not intersect with the rectangle. Therefore, two operators, which are applied to the rest of the beacons in the *INList*, are developed to handle these two cases of overestimation and underestimation:

• *Rectangle Truncating Operator (RTO)*: is used while the circle intersects the rectangle, which means that the possible target locations lie inside of the rectangle. The intersected region remains and the rest is removed from the rectangle to form a new rectangle.

• *Rectangle Pulling Operator (RPO)*: is used when the circle does not intersect the rectangle. In this case, the rectangle is dragged towards the beacon with a weighted distance.



(a) Applying rectangle truncating operator. (b) Formation of the new rectangle.

Figure 6: Truncating the rectangle.

Figure 6 illustrates an example of the rectangle truncation operator (RTO). When a beacon with the highest priority at the time is retrieved from the *INList*, a point on the circumference of the circle, called *circumference point*, P_i, that is closest to the center of the rectangle, C_r, is established, as shown in Figure 6(a). Circumference point P_i is determined by taking the intersection between the line, that passes through B_i and C_r, and the circumference of the circle. The edges of the new rectangle extend from P_i perpendicularly towards the edges of the current rectangle, in the direction of the beacon position, B_i, and thus the edges are perpendicular to the edges of the current rectangle. Finally, the new rectangle is represented by the shaded region and replaces the old rectangle as in Figure 6(b). If the rectangle is created by the last circle in the *INList*, the center of the rectangle is the estimated target location.

Figure 7 shows an example of the rectangle pulling operator. When a beacon underestimates the distance, the estimated target location by the beacon does not reside in the rectangle, as shown in Figure 7(a). In this situation, the size of the rectangle remains unchanged when the estimated distance does not reach in the rectangle, but the beacon information is still valuable to the target estimation. RPO adopts a weighted shifting technique to pull the rectangle. Since the estimated distances received from the beacons are hypothetically correct, the rectangle is

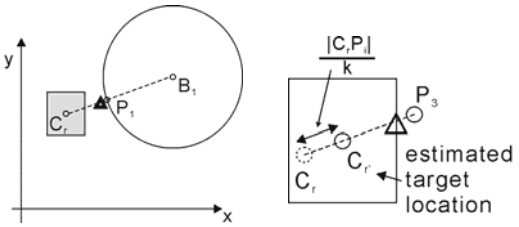
dragged toward the circumference point of the beacon. The rectangle is dragged toward the beacon with distance equal to the weight multiplied by the distance, between the center of the rectangle, *Cr*, and the circumference point, *Pi*, because the order of the beacon information indicates the relative importance and reliabilities of a piece of beacon information to other beacon information in the *INList*. Figure 7(b) shows the shift of the rectangle in the direction toward the beacon. The shift in the rectangle can be expressed as $1/k(|CrPi|)$, where k is the order of the beacon in the *INList*, $1/k$ is the weight, and $|CrPi|$ represents the distance between *Cr* and *Pi*. Figure 7(c) shows the rectangle shift after processing the beacon *B*1. The dashed rectangle and solid rectangle represent the position of the original rectangle and the new rectangle respectively. In this example, since the beacon *B*3 is the last beacon in the INList, the centroid of the rectangle becomes the estimated target location.

The ART algorithm sequentially imposes RTO and RPO on the sorted beacon information, and the centroid of the last rectangle is the estimated location of the target. Let *Bi* be the beacon information received from the nearby participated beacons sorted by the intersection number and the radius in the *INList*, where $i = 1, 2, \ldots, n$, and n is the number of the participated beacons. The ART algorithm can be expressed in a general mathematical equation as follows.

$$\begin{aligned} position &= centroid(B_1 \oplus B_2 \oplus B_3 \oplus \mathbf{K} \oplus B_n) \\ &= centroid(\overline{B}_1 \oplus B_3 \oplus B_4 \oplus \mathbf{K} \oplus B_n) \\ &= centroid(\overline{B}_2 \oplus B_4 \oplus B_5 \oplus \mathbf{K} \oplus B_n) \\ &\mathbf{K} \\ &= centroid(\overline{B}_i \oplus B_{i+2} \oplus B_{i+3} \oplus \mathbf{K} \oplus B_n) \\ &\mathbf{K} \\ &= centroid(\overline{B}_{n-2} \oplus B_n) \\ &= centroid(\overline{B}_{n-1}) \end{aligned} \quad (1)$$

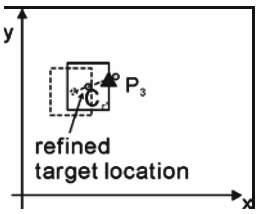
where $\oplus$ represents the RTO or RPO and is dynamically binded at run time. RTO is used when the circle intersects with the rectangle, and RPO is applied otherwise. *Bi* denotes the intermediate state of the rectangle after processing $(i + 1)$ beacons from *INList*, and can be calculated as follows.

$$\oplus = \begin{cases} \overline{B}_1 = B_1 \oplus B_2 & \text{when i=1} \\ \overline{B}_i = B_{i-1} \oplus B_{i+1} & \text{when i = 2, …, n-1} \end{cases} \quad (2)$$



(a) Instance before applying the rectangle pulling operator.

(b) Compute the pulling weight.

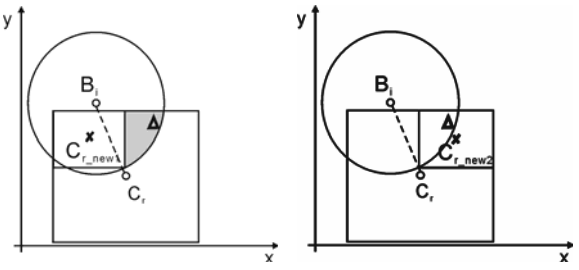


(c) Instance before applying the rectangle pulling operator.

Figure 7: Pulling the rectangle.

5.3 Enhancement for Implementing Rectangle Truncating Operator

When a RTO aggressively truncates away large sections of the intersected region between the rectangle and the circle formed by the beacon that could potentially contain the target, the possible locations in the rectangle can be far away from the target. In this case, ART algorithm could generate a less accurate localized point. Figure 8 depicts an example of a RTO generating a less accurate rectangle. In Figure 8(a), the original RTO truncates away the region (the gray region in the figure) that contains the location of the target. After processing the beacon *Bi*, C_{r_new1} is some distance away from the actual target location (the triangle in the figure), and unfortunately, the rectangle does not cover the actual target location. On the other hand, in Figure 8(b), a rectangle would provide a more accurate location estimation if the new rectangle is on the right hand side of the circumference point. However, the RTO presented in the previous section cannot generate the rectangle in Figure 8(b), because the rectangle in Figure 8(a) covers more intersected region and is believed to have a higher probability of capturing the



(a) Rectangle created by ART.

(b) Better estimation of rectangle.

Figure 8: Improvement on the rectangle generation.

actual target location. Therefore, an additional technique is needed to discover that the target could reside in the rectangle which represents a smaller intersected region, such as the instance in Figure 8(b).

In order to improve the RTO results, creating a new rectangle should consider a higher probability of capturing the target. Since a piece of beacon information is insufficient to construct new rectangles that probably contain the target, one step look ahead (OSLA) beacon is employed to assist the current beacon to determine the rectangle. The concept of applying OSLA beacons is to use two pieces of the beacon information simultaneously to identify the subrectangle in the current rectangle that potentially contain the actual target location. To achieve beacon collaboration, the current beacon creates rectangle candidates that cover the possible target locations estimated by the beacon. The OSLA beacon further determines the rectangle candidate that has higher probability by selecting the smallest distance between its uncertainty region and the rectangle candidate. The OSLA beacon is served for two purposes: (i) identifying the rectangles that could contain the target, and (ii) applying the distance estimation to determine the rectangle candidate that has a higher probability of capturing the target inside the rectangle. The improved RTO includes the following three steps to determine a new rectangle that has a higher probability.

Step 1. Creation of rectangle candidates. The original rectangle is partitioned into four subrectangles. A subrectangle is a rectangle candidate if the subrectangle covers parts of the circle area generated from *Bi*. The actual target location hypothetically resides in the intersected region between the circles generated from the beacons. Therefore, each rectangle candidate has a probability of capturing the actual target location. Additionally, the candidates with low probabilities can be removed in order to increase the accuracy of the target location estimation.

Step 2. Insignificant rectangle candidate removal. This step removes the rectangle candidates that enclose a small intersected region. A smaller intersected region in a rectangle indicates that the probability of the target locating in the rectangle is low. Moreover, a smaller intersection region implies that the intersected region is farther away from the center of the rectangle, and thus the possible locations within the rectangle have lower probabilities of the predicting the target accurately.

Step 3. One step look ahead (OSLA) beacon enrollment. The OSLA beacon selects a rectangle candidate that is reachable by the estimated distance of the OSLA beacon. Since the target location estimated by an OSLA beacon hypothetically lies on the perimeter of the OSLA beacon, a shorter distance between the uncertainty region of the OSLA beacon and the rectangle candidates indicates that the target is more probable to lie inside the rectangle candidate.

Subsequently, the detail of the three steps is explained and illustrated by a run-through example. In the first step, a set of rectangle candidates are created for RTO by extending the edges vertically and horizontally, from the circumference point, across the current rectangle and cover sections of the intersected region simultaneously. Figure 9 shows an example of rectangle candidates. The current rectangle is partitioned into four subrectangles according to the circumference point. Three of the subrectangles centered at C_{r_new1}, C_{r_new2}, and C_{r_new3} are the rectangle candidates. Since these three subrectangles covers difference sections (the shaded region in the figure) of the uncertainty region of *Bi*, apart from the original RTO, two addition subrectangles centered at C_{r_new2} and C_{r_new3} are created.

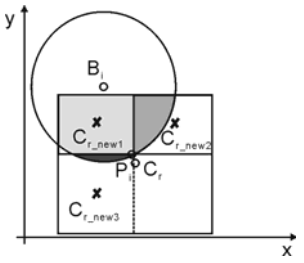


Figure 9: Rectangle candidates.

The second step removes insignificant rectangle candidates that could cause less accurate estimation. Although the new rectangle candidates centered at C_{r_new2} and C_{r_new3} represent the possible location of the target, the candidates can be deficient if the section of the intersected region that the candidate covers is relatively small. Figure 10 illustrates a circumstance where an OSLA beacon selects a rectangle candidate centered at C_{r_new3} and generates an imperfect estimation. The OSLA beacon B_i *next* would select the rectangle candidate centered at C_{r_new3}, because a larger area exists between the candidate centered at C_{r_new3} and the uncertainty region of the OSLA beacon B_{i_next}. Theoretically, containing the larger intersection area implies that the rectangle candidate has high probability of capturing the target. However, the intersection, between the uncertainty region of the *Bi* and *Bi next* in rectangle candidate centered at C_{r_new3} (the shaded region in the figure), is small. The insignificant rectangle that covers a small section of the intersected region has less probability to capture the

target and needs to be removed.

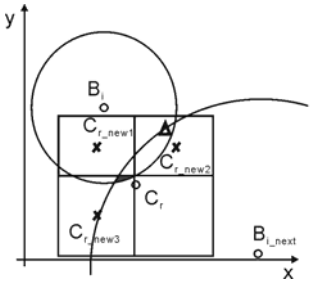
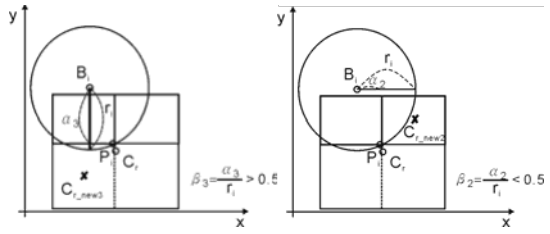


Figure 10: Imperfect rectangle candidate.

In order to identify the insignificant rectangle, the area of the uncertainty region covered by the candidate is compared to the uncertainty region of B_i. Let α_j be the distance measured for rectangle candidate centered at C_{r_newj} in the x or y coordinate, from the beacon B_i to the circumference point P_i. The α_j is measured in y coordinate when the intersected region covered by the candidate is split by the horizontal line extended from the circumference point P_i. Likewise, the α_j is measured in x coordinate when the intersected region is split by the vertical line. The ratio, denoted by β_j , between αj and the radius r_i of the circle determines the relative intersected region size. If $\beta_j \geq 0.5$, the ratio of intersected region is considered to be too small and the candidate is removed. Figure 11 illustrates instances of the significant and insignificant candidates. In Figure 11(a), α_3 is greater than one half of the radius ($\beta_3 > 0.5$), r_i, and thus, the rectangle is insignificant and is removed. Figure 11(b) demonstrates an alternative case where α_2 is less than one half of the radius r_i ($\beta_2 < 0.5$). As a result, the rectangle centered at C_{r_new2} is considered to be remains to be the rectangle candidate. Notice that although the example shows a rectangle candidate is removed, there exist situations that either of the two candidates are reserved or removed simultaneously.



(a) Insignificant rectangle. (b) Significant rectangle.

Figure 11: Insignificant rectangle removal constraint.

After the insignificant rectangle candidate is removed, the estimated distance from the OSLA beacon is applied to determine the candidate that overlaps the larger uncertainty region of B_{i_next}. The target location hypothetically lies on the perimeter of the OSLA beacon. Therefore, the shorter distance from the center of the candidates to the perimeter of the OSLA beacon indicates that the target is more possible to lie inside the candidate. Figure 12 shows an instance of involving an OSLA beacon to select a candidate that has a higher probability of capturing the target. Following the example discussed above, the OSLA beacon removes the candidate centered at C_{r_new1}, because the distance from C_{r_new1} to the perimeter is greater than that from C_{r_new2} to the perimeter (referring to the solid bold lines in the figure). Hence, the candidate centered at C_{r_new2} becomes the new rectangle.

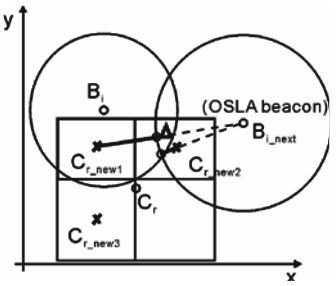


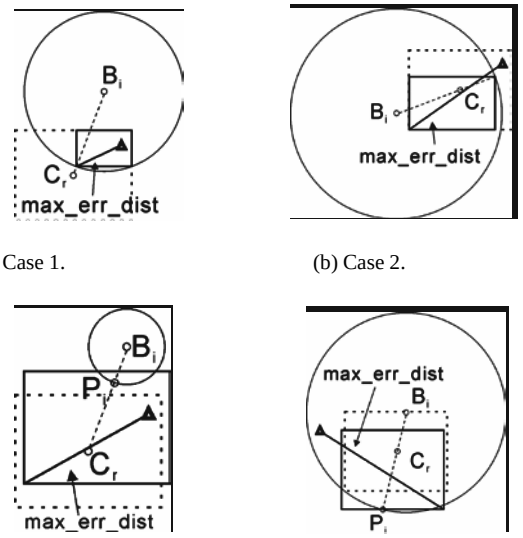
Figure 12: Discovering the closest center of the rectangle candidates from the perimeter of the OSLA beacon.

5.4 Why is ART Algorithm Better?

ART algorithm can control the localization error by minimizing the increase in the error during the localization process. However, if the distance estimation techniques offer less accurate estimated distances in the fading-signal sensor environment, none of the localization algorithm can generate accurate results. Although RSSI is unstable in a wireless sensor network and an estimated distance can be inaccurate, the actual target location is usually some distance away from the perimeter of the circle provided by the beacon. The probability of the actual target location locating closely to the perimeter is higher than that locating far away from the perimeter. Based on the characteristics of estimated distance uncertainty, the ART algorithm is designed to manages the beacon information, and identify the possible target locations. Therefore, maintaining the uncertainty region is a key aspect to prevent the localization error from increasing.

In order to manage the localization error, a *maximum error distance*, denoted as *max_err_dist*, is used as a metric to measure the localization error for the ART algorithm. The maximum error distance is the maximum distance between the target and the possible target locations, which represents the worst possible error distance in ART. The *max_err_dist* measures the maximum localization error in the intermediate state of the rectangle when the beacons are sequentially processed. If the *max_err_dist* does not largely increase during a localization process, the

estimated location is believed to be reasonably accurate.



(a) Case 1. (b) Case 2.

(c) Case 3. (d) Case 4.

Figure 13: Influences of RTO and RPO operations on the *max_err_dist*.

The localization can be influenced by two operations, RTO and RPO, in the ART algorithm. RTO is able to avoid expanding the *max err dist*. RPO can cause the *max_err_dist* to change, but in most cases, *max err dist* reduces after employing. An increase in the *max err dist* is resulted from beacons providing exaggerated distance estimation such that the rectangles is pulled away from the target. However, these beacons usually have smaller priorities in *INList* and weighted shift reduce the severity on localization error. *Max_err_dist* changes in the ART algorithm can be summarized into four cases, the corresponding examples are illustrated in Figure 13. The dashed rectangle represents the current rectangle and the solid rectangle is the new rectangle after processing the beacon B_i. The following four cases describe the influences of RTO and RPO on *max_err_dist*.

Case 1: A RTO causes the center of the new rectangle closer to the target. The *max err dist* is reduce in this case, because the farthest corner of the rectangle from the target has been truncated and the new corner that has the *max err dist* must be closer to the target than the previous one. Figure 13(a) illustrates an instance of such case where the region of the rectangle outside of the circle is eliminated. Since the size of the rectangle and the possible locations for the target has been reduced, the localization error also decreases.

Case 2: A RTO causes the center of the new rectangle farther from the target. The *max err dist* remains the same because the new rectangle remains in the previous rectangle. The corner of the rectangle that has the maximum error distance from the target remains unchanged. An instance of the center of the rectangle moves farther away from the target after executing a RTO is shown in Figure 13(b). In this case, the RTO is less effective in location estimations, since the *max err dist* is not reduced.

Case 3: A RPO causes the center of the new rectangle move closer to the target. The *max err dist* is reduced because the rectangle is dragged towards the target, as shown in Figure 13(c). Thus, the farthest corner of the rectangle from the target has been shifted toward the target, leading the new rectangle a closer estimation of the target. The distance of the shift is the maximum reduction in the *max_err_dist*. Since the RPO applies a weighted shift based on the priority of the beacon information in the INList, the *max err dist* is reduced at most $1/k(|C_rP_i|)$ distance.

Case 4: A RPO causes the center of the new rectangle to move farther from the target. The *max_err_dist* increases because the rectangle and the farthest corner of the rectangle is dragged away from the target. The weighted shift applied in RPO could cause the max err dist to increase at most $1/k(|C_rP_i|)$. Figure 13(d) depicts an example of the RPO dragging the rectangle away from the target. The case occurs when the beacon provides a exaggerated distance estimation. The circle with exaggerated estimated distance usually has less intersection number, because the beacon does not estimate the target to a common uncertainty region.

Notice that *max err dist* could increase if the distance between the target and the beacon is far different from the estimated distance, i.e., the target locates near to the beacon, but the estimated distance is relatively large. The case is generally caused by the estimated distance generated from fluctuated RSSI, and only takes place in the RPO. Consequently, the less accurate location estimation is not a defect of the ART algorithm.

6 LOCATION VERIFICATION MECHANISM (LVM)

A fading-signal sensor environment causes large distances between localized points on a static object. While the localization operation is executed once, the accuracy of the estimated target location can never be verified. The noises and interferences in the wireless sensor network can cause drastic change in the RSSI and leads to a large deviation of the target location estimation comparing to the location of the

actual target location. The problem can be overcome by finding a set of location estimations and computing an average location from the set of the location estimations. The purpose of finding the average position is to reduce the impact of the huge localization error generated from excessive RSSI deviations. However, finding an average coordinate from a large set of locations can be expensive for wireless sensors.

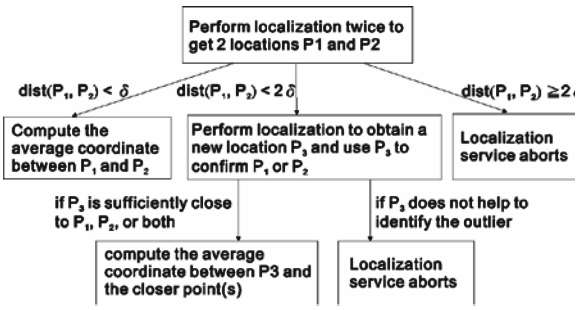
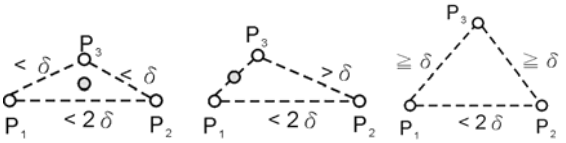


Figure 14: Flowchart of the location verification mechanism.

In order to balance the considerations of a high confidence in location and low energy consumption, a threshold-based location verification mechanism (LVM) is designed to improve the consistency of location estimations. If two consecutive localized points are located closely together, the estimate locations are believed to be fairly correct. The closeness between the points is represented by a threshold δ. Due to different environments and limitation of the hardware, the closeness of the localized points is application dependent. Since the actual target location is not known in the real world environment, the accuracy of the localized points depends on the consistency of the localized points. Figure 14 shows the flow of the LVM algorithm. Initially, two points are retrieved to confirm the target location. The distance between the two points, $P1$ and $P2$, are summarized into following three cases. In the first case that the distance between the two localized points is less than the threshold δ (that is, dist($P1$, $P2$) < δ), these positions are considered to be relatively close estimation to the actual target location. The average coordinate between $P1$ and $P2$ is computed and returned. In the second case that the distance is greater than δ and less than two times δ (that is, *dist*($P1$, $P2$) < 2δ), the location of a localized point is believed to be creditable and the other location is affected by the noises. In such cases, a second chance is given to verify the correct location and a third point is employed to identify the creditable localized point. The details of this case will be further explained later. In the last case that the distance between the two localized points is apart for more than two times δ (that is, *dist*($P1$, $P2$) $\geq 2\delta$), the localized points are less creditable and discarded because the environment is predominantly noisy for further calculation. As a result, the localization procedure aborts and no target location is generated.



(a) $P3$ is sufficiently close to the other two points.

(b) $P3$ is sufficiently close to one of the two points.

(c) $P3$ cannot confirm the correct point.

Figure 15: Target location estimations in various algorithms ($4m^2$).

In the second case above, where only one of the two localized points is affected by sudden amplified noises, a third point $P3$ is generated to compare with the previous two localized points. If $P3$ locates closely to one of the two points, $P3$ and the point are considered to be fairly accurate estimations. Figure 15 depicts three scenarios using a third localized point to confirm the more accurate point. The cases for the third point, $P3$, are managed as follows:

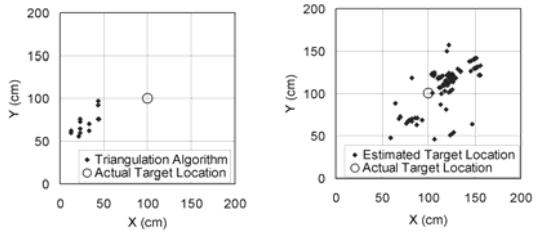
- **dist(**$P1$**,** $P3$**)** < δ **and dist(**$P2$**,** $P3$**)** < δ**:** since the $P3$ is sufficiently close to two points, all three points are considerably reasonably accurate. Therefore, the average coordinate between $P1$, $P2$ and $P3$ is calculated as the estimated location, as shown in Figure 15(a). An instance of this case would be the points, $P1$ and $P2$, locating on either ends of the actual target location, and $P3$ lies between the two points.
- **dist(**$P1$**,** $P3$**)** < δ **or dist(**$P2$**,** $P3$**)** < δ**:** the more accurate localized is confirmed by the $P3$, the average coordinate between $P3$ and the closer point $P1$ or $P2$ is calculated. Figure 15(b) illustrates an example of the case where $P3$ is closer to $P1$. The average coordinate between $P3$ and $P1$ is computed as the final location estimation. $P2$ is identified as an outlier and discarded.
- **dist(**$P1$**,** $P3$**)** $\geq \delta$ **and dist(**$P2$**,** $P3$**)** $\geq \delta$**:** the third point $P3$ is considered to be a misrepresentation which cannot reinforce the confirmation of the more accurate point. Figure 15(c) depicts this case when $P3$ is far away from the two points. $P3$ is a biased point itself and is a second chance to aid the location estimation. A new point is obtained to replace $P3$ and the procedure in case 2 is executed again. If the new $P3$, which falls in this case in the second time, is still not close enough to determine the more accurate point, the sensor environment is considered to be too noisy for the sensors to offer reasonably reliable distance estimations. Therefore, the points are

discarded and the estimated location is generated.

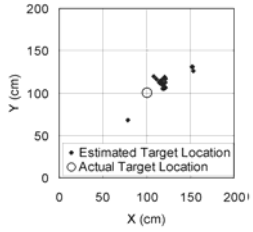
*P*3 plays a role of the eliminating the outlier points by having a close estimation with the more accurate point. If *P*3 is unable to identify the outlier between the two localized points, *P*1 and *P*2, then the estimated locations are considered to be unreliable and the localization process terminates.

7 EXPERIMENT STUDY

In this section, we conduct a set of real-world experiments to evaluate the performance of our algorithms. Five Mica2 motes are employed to run the localization algorithm in an indoor environment. Four of the motes are set as the beacons which know their own positions, and the fifth one is used to represent the target that requests for localization. Beacons are deployed in a 1m x 1m, a 2m x 2m, a 3m x 3m, and a 4m x 4m area which form the four corners of the square, and the 4m x 4m area is the default setting for measuring different metrics. The threshold δ in the localization verification mechanism is set to 30 cm. The location verification mechanism is initialized every two seconds. The target is placed at the center of the square where it is equal distant to the four beacons at the corners. For reader's interest, the source codes can be found at our project site [13].



(a) Triangulation localization points. (b) ART localization points.



(c) ART+LVM localization points.

Figure 16: Target location estimations in various algorithms ($4m^2$).

In the first experiment, we conduct target location estimations using the triangular method [8], ART, and ART with LVM, as shown in Figure 16. Figure 16(a) shows the result of using the triangulation algorithm. Apparently, the gravity of the estimated location of using the ART method is much closer to the actual target location. In Figure 16(a), the localized points generated from triangulation algorithm are approximately 50 cm away from the actual target location, since the localization is affected by the fading-signal environment. Comparing to the triangulation algorithm, ART alleviates the fading-signal effect and generates estimated locations mostly within 50 cm, as shown in Figure 16(b). In Figure 16(c), the location verification mechanism (LVM) further improves the ART results, by eliminating the outlier estimations. There are apparently much more estimated locations that are closer to the actual target location than the other two can achieve. Concentration of estimated locations is a nice feature of LVM that greatly assists to identify the actual target location.

The second experiment measures the localization error using various algorithms. Figure 17 shows the localization error of using our methods comparing with that of using the triangulation algorithm. The localization error is measured by taking the average of the Euclidean distance from the estimation target locations to the actual target location. Notice that the greater a deployment area, the greater the localization error would appear. In all cases, localization error of the ART algorithm with LVM always outperforms the other two methods. The localization error of triangulation algorithm increases larger than ART and ART with LVM as the deployment space increases. This is because the effect of the fading-signal has a greater impact when the distance between the beacon and the target increases.

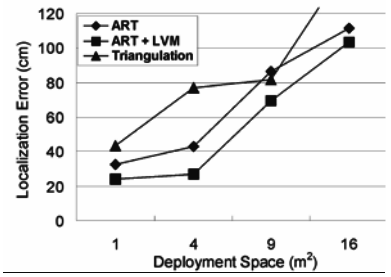


Figure 17: Localization error compared with Triangulation Algorithm.

The third experiment measures the failure ratio of the location verification mechanism. A failure is encountered when the verification mechanism generates no location. The failure ratio indicates the number of the localization times when LVM generates no result over the total number of the localization times. Figure 18 shows the failure ratio for different threshold δ. The ART algorithm with LVM is applies to generate localized point for every 10 cm of threshold δ. LVM (2-pt abort) and LVM (3-pt abort) show the failure ratio for LVM using 2 localized points and LVM using 3 localized points,

respectively. LVM (total) is the sum of the failure ratio of LVM (2-pt abort) and LVM (3-pt abort). As the δ increases, the localized points are more distributed. This is a tradeoff between a localization failure ratio and the consistency of the location estimation.

A third point is seldom employed when δ equals 10 cm and 50 cm, as shown by the curve of LVM (3-pt abort). For $\delta = 10cm$, the distance between a pair of the localized points is frequently far apart in the fading-signal sensor environment such that the LVM discards the points before a third point is applied. For $\delta = 50cm$, ART is able to estimate the location within the threshold, and thus, third points are less requested to assist the location estimations. As more of the third points are enrolled in localization (i.e., $\delta = 20 \sim 40cm$), the failure ratio is reduced nearly 20%. Hence, using a third point in LVM has significantly avoided a large amount of the localization failures.

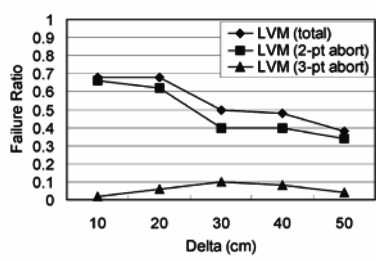


Figure 18: Failure ratio vs. the threshold δ.

8 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel efficient localization algorithm in a wireless fading-signal sensor environment. From the observed RSSI fluctuation, an uncertainty region model is designed to manage the uncertainty regions generated by the beacons. We then develop the aggressive rectangle truncation (ART) algorithm based on the uncertainty region model. Due to the fading-signal environment, precise distance estimations are difficult to achieve. ART accommodates pruning the low-probability region to gradually shrink the possible location region, rather than computing locations based on the direct distance relationship, such as triangulations. We also designed a location verification mechanism to provide more concentrated location estimation. The concentration of estimated locations provides a strong confidence on the estimated target location. Furthermore, the verification framework can also be applied to other existing localization methods. Comparing to the triangulation algorithm, the proposed algorithm has as large as 61% of improvement in reducing localization errors. For the future work, we are currently extending our work to track moving objects in wireless sensor network. Another direction of our future work is to extend the ART algorithm to assist to capture human motions for multimedia simulation applications.

ACKNOWLEDGEMENT

The authors would like to thank Professor Ko-Hsin Liang for the insightful advices on efficiently measuring RSSI and systematical translation between RSSI and distance. The authors also acknowledge Shi-Jei Liao for the support in building experiment infrastructures and sharing the source codes of the triangulation algorithm.

REFERENCES

[1] lan F.Akyildiz, WwllJan Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," in *IEEE Communications Magazine*, 40(8): pp. 102-114, 2002.
[2] Cesare Alippi, Giovanni Vanini, "A RSSI-Based and Calibrated Centralized Localization Technique for Wireless Sensor Networks", in *Proceedings of IEEE PERCOMW*, pp. 301-305, 2006.
[3] Mark de Berg, Marc van Kreveld, Mark Overmars, Otfried Schwarzkopf "Computational Geometry," Springer-Verlag Berlin Heidelberg, 2000.
[4] Paramvir Bahl and Venkata Padmanabhan, "RADAR:An In-Building RF-Based User Location and Tracking System," in *Proceedings of IEEE INFOCOM*, pp. 775-784, 2000.
[5] Woo Cheol Chung, Dong Sam Ha, "An Accurate Ultra Wideband Ranging for Precision Asset Location," in *Proceedings of UWB Systems and Technologies*, pp. 389-393, 2003.
[6] Cricket Project, "Cricket v2 User Manual," MIT Computer Science and Artificial Intelligence Laboratory, 2004.
[7] Crossbow Technology Inc., http://www.xbow.com/.
[8] Chao-Chun Chen, Yung-Chiao Tseng, Shi-Jei Liao, and Chiang Lee "Design and Implementation of Shooting Simulation System over Wireless Sensor Networks," *Communication of IICM*, 9(2): pp. 123-138, 2006.
[9] Aram Galstyan, Bhaskar Krishnamachari, Kristina Lerman1 and Sundeep Pattem, "Distributed Online Localization in Sensor Networks Using a Moving Target," in *International Symposium on IPSN*, pp. 61-70, 2004
[10] K. C. Ho, Wenwei Xu, "An Accurate Algebraic Solution for Moving Source Location Using TDOA and FDOA Measurements", in *IEEE Transactions on Signal Processing*, pp. 2453-2463, 2004.
[11] Xiang Ji, Hongyuan Zha, "Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling," in *Proceedings of IEEE INFOCOM*, pp. 2652-2661, 2004.
[12] Joon-Yong Lee, Robert A. Scholtz, "Ranging in a Dense Multipath Environment Using an UWB Radio Link," in *Proceedings of IEEE J. Selected Areas in Communications*, pp. 1677-1683, 2002.
[13] Da-Chung Mao and Chao-Chun Chen, ART at SourceForge.net, https://sourceforge.net/projects/artlvm/, 2006. [14] Dragos Niculescu and Badri Nath, "Ad Hoc Positioning System (aps) Using AoA," in *Proceedings of IEEE INFOCOM*, pp. 1734- 1743, 2003.
[15] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The Cricket Location-Support System," in *Proceedings of MOBICOM*, pp. 32-43, 2000.
[16] Niels Reijers, Gertjan Halkes, Koen Langendoen, "Link Layer Measurements in Sensor Networks," in *Proceedings of IEEE MASSP*, 2004.
[17] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha, "Tracking Moving Devices with the Cricket Location System," in *Proceedings of MobiSys*, pp. 190-202, 2004.
[18] Jason Small, Asim Smailagic, Daniel P. Siewiorek, "Determining User Location for Context Aware Computing through the Use of a Wireless LAN Infrastructure," Technical Report, December 2000.