Chapter XI
On a Class of Localization Algorithms Using Received Signal Strength

Eiman Elnahrawy
Rutgers University, USA

Richard P. Martin
Rutgers University, USA

ABSTRACT

This chapter discusses radio-based positioning. It surveys and compares several received signal strength localization approaches from two broad categories: point-based and area-based. It also explores their performance and means to improve it. It describes GRAIL - a sample positioning system. It finally concludes with a brief discussion of sensor applications that utilize location information.

INTRODUCTION

Location information is essential for many emerging applications, ranging from a diverse set of areas including asset tracking, workflow management, and physical security. Sensor networks offer an unprecedented potential for realizing many of these applications. Combined with a localization system, sensor nodes can be attached to objects and people and continuously track their locations. Those locations, when communicated back to a network backend, can then be utilized for functions such as controlling access to spaces, making decisions for workflow, or managing inventory.

Outdoors, the location information can be easily obtained using Global Positioning System (GPS) units. However, often it is not feasible to attach a GPS unit to each sensor, because of the additional cost to the sensor node, or because localization with GPS consumes considerable power. Also, GPS does not work well indoors because there is no clear line of sight to the satellites, and many applications must run in indoor environments.

For stationary sensors, a straightforward approach to localization is to simply store their positions during deployment. However, in many situations the node is mobile and the entire network is dynamic. Thus a localization system is necessary to track the positions of the sensor nodes and objects they are attached to.

This chapter will survey research on positioning the sensor nodes using the received signal strength (RSS) of wireless packet transmissions. Given that all modern radio chipsets include the hardware necessary to measure and report the signal strength of received packets, there is a tremendous cost and deployment advantage to re-using the existing RSS infrastructure of the communication network for localization purposes. However, additional hardware, such as ultrasound, can also be added to sensor nodes. The cost/performance tradeoffs and the impacts of additional localization resources in the sensor network are not well understood, and are the subjects of ongoing investigation.

We begin with a broad survey of existing localization approaches and algorithms. We then briefly discuss the causes of positioning uncertainty and how such uncertainty can be expressed in a meaningful and useful way for the higher-level applications. We will then describe two methods of location presentation along with their pros and cons: single-point localization and area-based localization. In the former, a single (x,y) spatial location is returned while in the latter a regular or irregular area is returned, for example an ellipse versus a set of tiles. We will elaborate on sample representative algorithms from each class.

We include a brief evaluation of the various approaches. We show how random and systematic variations in the signal strength affect the performance. Our basic performance metric is the accuracy of the returned position, i.e., how far it is from the true location of the sensor, along with variants of this metric. We also discuss methods and guidelines to improve the localization performance, such as optimally placing anchors in the environment.

We then describe a general infrastructure for indoor localization called the GRAIL system and show how the different approaches we discuss fit into such a system. We will briefly list some development and deployment issues. Finally, we conclude by discussing current and emerging applications of sensors that leverage the location information.

LOCALIZATION APPROACHES

The numerous approaches to localization defy a simple taxonomy. However, there are only a handful of overall strategies and approaches. In all cases *anchors* or *landmarks*, i.e., sensor or gateway nodes with known locations are needed at some point in the process.

Aggregate approaches position sensors using a collection of measurements from a large number of nodes. In contrast, *individual* approaches use information between a single sensor and a set of landmarks.

Orthogonal to the number of sensors participating in the process is the algorithmic approach used. *Lateration* approaches use some function of distance between the sensors and the landmarks. In contrast, *Scene matching* approaches match sensor observations to known maps and do not require any concept of direct physical distance in the algorithm.

When using lateration, the distances could be derived directly from a signal strength decay function, or more indirectly through hop counts (Niculescu et al., 2001). Using the actual sensed data, however, the resulting set of distance equations often has no exact solution, and so approximations must be found. Finding the best approximation is often difficult. A classic approach minimized the residual using least squares, as described in (Patwari et al., 2005). However, the problem can be generalized to viewing the system of distance constraints as an optimization problem, and then applying a range of optimization solvers (Dohertyl et al., 2001). Another approach views the sensor observations as existing in a high dimensional observational space and uses multi-dimensional scaling to estimate the positions in the lower-dimensional physical space (Shang et al., 2003). A further set of lateration approaches averages the coordinates of the landmarks observed by a sensor, either averaging the entire set, or selectively averaging overlapping regions (He et al., 2003; Stolero et al., 2004).

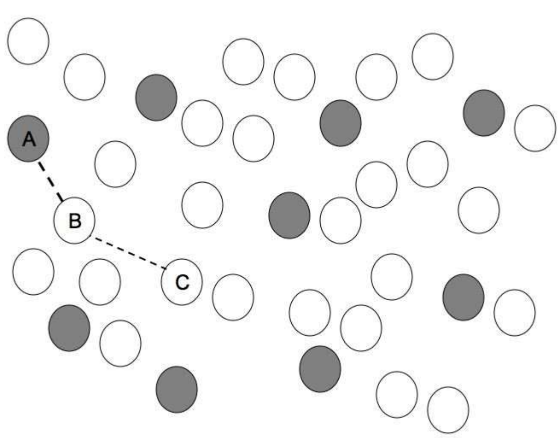
Matching algorithms can be generalized to a classic matching learning problem: given a known signal map and a set of observations, the localization system must derive the position on the map that best fits the observed data. Characterizing localization as matching thus opens the door to a wealth of machine learning approaches, including neural networks, Bayesian matching, and maximum likelihood estimation (Elnahrawy et al., 2004).

Figure 1 and Equation 1 show an example aggregate lateration approach that uses hop count. Node C computes its distance to landmark A, d, by multiplying the average hop distance, l_{avg}, over the network by 2 hops ($n = 2$). It repeats for all the landmarks (grey nodes) then uses lateration to compute its position.

$$d = n \times l_{avg} \tag{1}$$

Rather than using hop-counts, *individual* lateration must use another form of distance estimation to the landmarks. The following equation is a sample propagation model that translates signal strength, ss_{ij}, in dBm units, from sensor i to landmark j, to the distance between them, d_{ij}. The propagation parameters, a_i, and b_i, are unique for each environment and may even vary from one landmark j to another.

Figure 1. An aggregate localization scenario. Landmark nodes are plotted in grey. Node C is two hops away from node A.



$$ss_{ij} = a_i + b_i \times \log d_{ij} \tag{2}$$

Where

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3}$$

Typically, least squares estimation is then used to compute the unknown coordinates (x_i, y_i). Once at least 3 landmarks in two dimensions (4 in 3 dimensions) are known, a sensor node can trilaterate (i.e., use 3 landmarks) its position. Figure 2 shows an example network. Node *B* computes its distance to each landmark A_i using the measured signal to each landmark and Equation 2 with corresponding propagation parameters.

Aggregate approaches are practical only in dense networks, because they are often built on explicit proximity and isotropy assumptions (Niculescu et al., 2001). In most scenarios those assumptions do not hold. The sensor network can be sparse following irregular shape, which yields large errors in the position estimates. In the remainder of the chapter, we will therefore explore additional individual scene matching and lateration-based approaches.

UNCERTAINTIES AND OFFLINE TRAINING

Following Equations 2, 3 and assuming that signals travel with no obstructions, the unknown coordinates of any sensor node can be computed precisely and directly by translating radio ranges to distances using lateration. The environments where sensor networks are often deployed are however challenging, especially indoors. The radio signal suffers attenuations and distortions because of obstructions that absorb, reflect, refract or scatter the signals causing multipath effects. The distances to signals computed using equations 2, 3 are therefore inaccurate. Using lateration on those estimates directly will yield highly erroneous positions.

Scientists have thus opted for machine learning theory to map a set of radio signal strengths to a spatial position with high confidence along with a level of uncertainty. Next section describes an array of representative approaches.

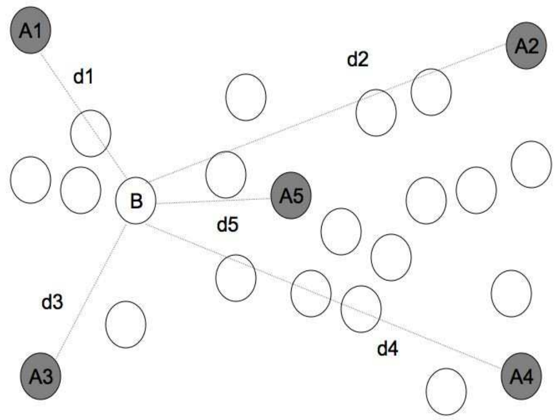
Broadly speaking, there are two families of strategies. The first family uses a *signal strength to distance* function. These rely on estimating the electromagnetic wave properties to compute the distance to known anchors from the observed signal strength, similar to Equation 3. The second family assumes the mapping can be completely arbitrary in that the received signal strength has no direct relationship with neighboring points or distance to the transmitting sensors, *i.e.*, strictly scene matching.

Both families follow what is so called *offline training* and *online localization* phases. Training data is collected during the offline phase and then applied during the online phase to infer the unknown position of the mobile sensor. The training data generally correlates the distance from the mobile sensor node to the anchor node (*i.e.*, landmark) with the strength of the (received/transmitted) signal, *i.e.*, the radio range. It can be in the form of a discrete training set or a continuous gridded map.

Training Set

A training set, *T*, consists of a set of empirically measured signal strength *fingerprints* from the *n* anchors in the network along with the *m* locations where they were collected. A fingerprint at a location,

Figure 2. An individual localization scenario. Landmarks are placed around the network, plotted in grey. Node B computes the distance d_i from each landmark A_i using the range to distance function.



i, with coordinates (x_i,y_i), is the set of average received signal strength, s_{ij}, from each anchor j. Typically, it is computed from a series of k signal strength *samples* collected at that location. A default value for s_{ij} is usually assigned in a fingerprint if no signal is received from anchor j at a location i. That is, $T = \{[(x_i,y_i), s_{i1}, s_{i2,\ldots,} s_{in}]\}, i = 1 \ldots m$.

The sensor node to be localized collects a set of *received signal strengths* (RSS). An RSS is similar to a fingerprint in that it contains a mean signal strength for each anchor j, $j = 1 \ldots n$. An RSS may also maintain a standard deviation of the sample set at each location i and anchor j, σ_{ij}. The collection of training points forms the training set.

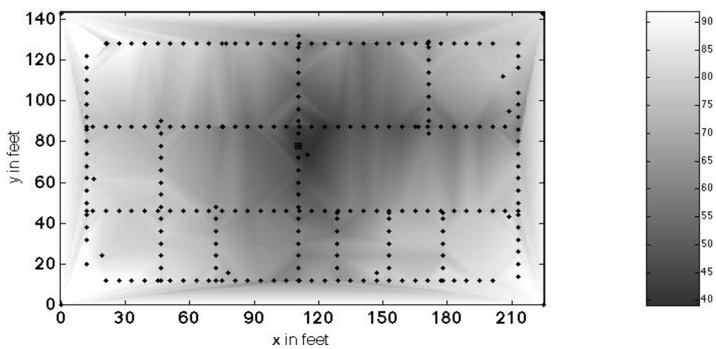
Gridded Map

Some localization approaches require a continuous map of fingerprints over the two-dimensional localization space. An approximation is to build a grid of regular simple shapes such as tiles that describe the expected fingerprint for the area described by the tile. The tiles can be tiny or coarse. A direct measurement of the fingerprint for each tile is expensive in terms of time and labor costs. Additional signal strength to location points may be interpolated, however, from a set of discrete training points in order to form the gridded map.

Specifically, a surface fitting approach is used to interpolate a fingerprint at each tile from a training set that would be similar to an observed one. Several approaches in the literature can be utilized for the interpolation such as splines. A map M_i, for each anchor node, i , $i = 1 \ldots n$, is built independently using the discrete observed training fingerprints for that anchor. It was found that the observed training points' spacing need only to follow a uniform distribution rather than have precise spacing (Elnahrawy et al., 2004).

Figure 3 shows a sample gridded map. This map was built using triangle-based linear interpolation in an indoor office building. In this approach the two-dimensional area is divided into triangular regions. The locations of the observed training points serve as the triangles' vertices. The expected signal strengths in intermediate locations (or tile) are then linearly interpolated using the "height" of the triangle at the center of the tile. This approach also naturally extends to volumes. Notice how the

Figure 3. A sample interpolated radio map for an anchor node. The dots show the actual observed training point locations while the square in the middle show the true location of the anchor.



signal is distorted and does not follow regular circular shapes as would be expected in a free space area where the signal decays uniformly with distance in all the directions.

ALGORITHMS

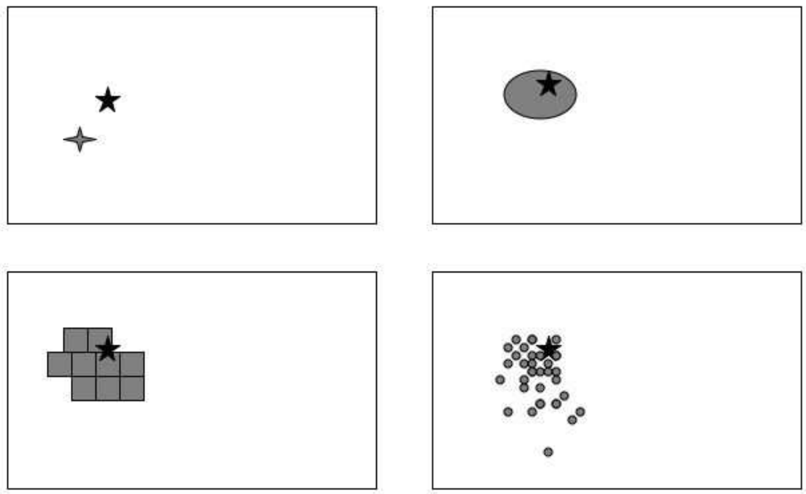
Localization algorithms are classified based on their output into two classes: point-based and area-based. In *point-based* localization, the goal is to return a single point for the sensor node while in *area-based* the goal is to return the *possible locations* of the sensor as an area or volume (areas and volumes are interchangeable from our perspective) (Elnahrawy et al., 2004). The area could be a regular shape such as a circle or an ellipse, or an irregular shape such as a set of tiles or a cloud of points. Figure 4 shows the various representations. The true location of the node is marked as a star.

This section gives an overview of various individual localization approaches from each class. There is generally a myriad of algorithms in literature on this topic, which stemmed out of long research over the past years. The algorithms selected here span broad techniques that proved practical and useful from surveying the references (Elnahrawy et al., 2004; Bahl et al., 2000; Youssef et al., 2003; Youssef et al., 2004; Battiti et al., 2002; Moore et al., 2004; Fang et al., 2005; Savvides et al., 2001; Lorincz et al., 2006; Hazas et al., 2003; Priyantha et al., 2000; Want et al., 1992; Krishnan et al., 2004; Ladd et al., 2002; Roos et al., 2002; Smailagic et al., 2002; Lim et al., 2006). They are intended to overview strategies rather than drill down into detail. The reader is therefore encouraged to pursue the references for further explanation.

Point-Based Algorithms

Point-based approaches can be further categorized as deterministic or statistical. In deterministic localization the sensor's RSS is matched against the training data set using a systematic deterministic strategy. The localization output is the coordinates of the closest matching point in the set. For statistical localization the matching is done probabilistically. This section sketches two major point-based algorithms, one from each category.

Figure 4. Different localization outputs: Point (top left), ellipse or circle (top right), tiles (bottom left), and cloud (bottom right). The ground truth is marked as a black star.



Deterministic Localization

A sample deterministic point-based algorithm we discuss here is the nearest neighbor approach. The strategy is to return the location of the closest fingerprint to the RSS fingerprints in the training data. It uses Euclidean distance in “signal space” as the deterministic measurement function. It views the fingerprints as points in an N-dimensions space, where each anchor node forms a dimension. Specifically, it computes the signal distance between each fingerprint vector in the radio signal map and the measured fingerprint for the localized sensor. It then picks the vector with the minimum distance and returns its coordinates as the estimated sensor’s location (Bahl et al., 2000).

This approach is sometimes referred to in literature as the RADAR approach. Other versions of this approach return the average position (centroid) of the top k closest vectors; i.e., averaged RADAR algorithm. For example if $k = 2$, it takes the closest two candidates and returns the mid-point on the floor between them. A disadvantage of deterministic approaches is that they require a large number of training points to perform adequately. To compensate for a small training set gridded signal maps are used as discussed before, e.g., as in a variant of RADAR called gridded RADAR (Bahl et al., 2000; Elnahrawy et al., 2004).

Statistical Localization

Many statistical approaches have been devised within the context of point-based localization. They range from applying simple Bayes’ rule and maximum-likelihood estimation to sophisticated support vector machines (Battiti et al., 2002; Youssef et al., 2003; Ladd et al., 2002; Roos et al., 2002). They generally map the problem of localizing sensors to a probability inference problem. First, a statistical model of the RSS as well as some system and environment parameters is constructed during the training phase. The model is then used during the online phase to infer the unknown sensor’s position.

Let us elaborate on an example probabilistic approach that applies Bayes' rule. Signals received from the different anchors are assumed to be independent. For each anchor j, j = 1... n, the received signal strengths at each (x_i, y_i) in the training data is s_{ij}. Using Bayes' rule, the probability of being at each fingerprint's location in the training data given the received signal vector of the sensor, $\overline{S_l} = (\overline{s_{lj}})$, is computed as follows.

$$P(L_i \setminus \overline{S_l}) = \frac{P(\overline{S_l} \setminus L_i) \times P(L_i)}{P(\overline{S_l})} \quad (4)$$

However, $P(\overline{S_l})$ is a constant c. Moreover, given there is no prior knowledge about the exact sensor's location, it is assumed that it is equally likely to be at any location, i.e., $P(L_i) = P(L_j), \forall i, j$. Therefore, Equation 4 is rewritten as:

$$P(L_i \setminus \overline{S_l}) = c \times P(\overline{S_l} \setminus L_i) \quad (5)$$

Without having to know the value c, the location in the training $L_{\max}, L_{\max = \arg\max_i}(P(\overline{S_l} \setminus L_i))$ is returned. Specifically, $P(\overline{S_l} \setminus L_i)$ is computed for every fingerprint i in the training set and the location of the highest probability candidate is returned as the predicted location. This approach hence inherently requires large enough training sets. Variants of the approach may return the midpoint of the top two or an average of top k candidates.

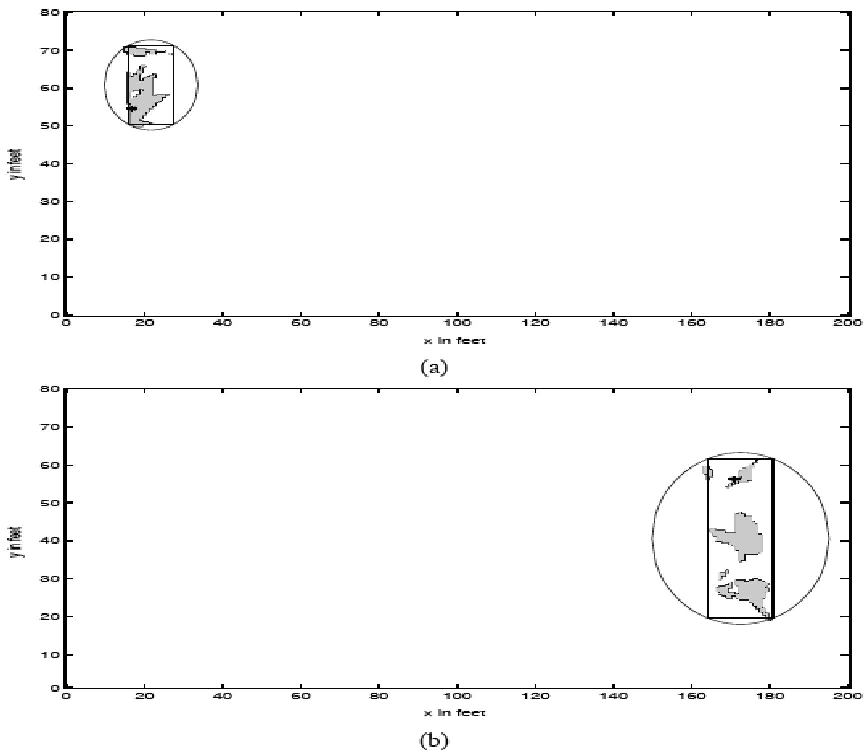
Area-Based Algorithms

We discussed that environmental effects impact localization and introduce fundamental uncertainty in the estimated position. Area-based approaches are better able to utilize and describe this uncertainty as compared to point-based approaches (Elnahrawy et al., 2004). Specifically, they provide an understanding of the localization confidence in a more natural and intuitive manner, where the term confidence is used loosely to refer to the positioning certainty. Hence, the larger the returned area, the less confidence we have in placement of the sensor in a particular location because many probable locations are included in the returned result. These approaches are also able to adjust the localization confidence by controlling the size of the returned area. Point-based approaches have difficulty describing such a trade-off systematically to the higher-level applications or users. A second advantage is that an area can naturally be mapped into a set of directions to search for the sensor in relation to the likelihood of its presence in the area, for example by beginning the search in the most likely area then continually expanding to the next most likely area and so on.

Figure 5 shows two example returned areas for a floor. The areas are shown by a dark color. The true location of the sensor is shown as a "*". The smallest circumscribing circles and rectangles are also shown. Figure 5(a) shows the localization can contain the sensor to an area the size of a single room while in Figure 5(b); the localization is more diffuse, in this case spanning two rooms.

The circumscribing circles show that augmenting a point with a distance to describe the uncertainty, in point-based localization, would likely return a much larger area than a strictly area-based approach. Returning rectangles, while reducing the inaccuracy of circles, no longer fits the definition of a point-based approach, however.

Figure 5. Sample areas returned by area-based presentation, specifically SPM, versus single-point based presentation. The true location is marked by a "".*



Simple Point Matching

Simple Point Matching (SPM) finds the set of those tiles whose signal strengths fall within a threshold of the received signal for each anchor node independently. Specifically, for each anchor *j*, *j* = 1... *n*, it "matches" all fingerprints $\overline{S_l} = (\overline{s_{l1}},...,\overline{s_{1n}})$ from the floor tiles (Elnahrawy et al., 2004). The matching tiles for each anchor *j* are computed by adding an expected "noise" level *q* to its received signal $\overline{S_{lj}}$, and then returning all the area tiles that fall within the expected threshold $\pm q$.

It then returns the area formed by intersecting all matched tiles from the individual anchors' tile sets. A gridded training map is used (Elnahrawy et al., 2004). The approach is eager, *i.e.*, it finds the fewest most probable tiles by starting with a very low *q*. It then incrementally increases it by trying *2q*, *3q*, .., until an intersection is found. Even in the worst case, a non-empty intersection will result, although *q* may expand to the dynamic range of signal readings. The value *q* is usually bounded by the standard deviation of the signals in the localization environment.

SPM is a Maximum Likelihood Estimation approach that assumes the anchor nodes are totally independent. Specifically, to localize a sensor *i*, the set of received signals between each anchor *j* and the sensor during a time window is estimated by a Gaussian distribution centered around the average measured signal $\overline{S_{lj}}$ with variance equals to $(\sigma_{lj})^2$.

Therefore, a $(1-\alpha)$ 100% confidence interval for the estimator is as follows, where $z_{\frac{\alpha}{2}}$ is a constant that depends on α, *e.g.*, it equals 1.96 for a 95% confidence interval of the estimator

$$\overline{s_{lj}} \pm z_{\frac{\alpha}{2}} \times \sigma_{lj}.$$

For single-mode distributions, such as Gaussian, increasing the confidence level, $(1 - \alpha)$, increases the width of the estimator's interval at the cost of adding less probable values to it. That is, less confidence is indeed better in our context; since higher probability values are the only ones included in the interval. Although the Gaussian approximation assumption may not be true in general, it has been proven effective in practice.

In SPM, the noise level $z_{\frac{\alpha}{2}} \times \sigma_{lj}$ corresponds to SPM eagerly attempts to find the appropriate (lowest) confidence level $(1-\alpha)$ for each anchor j that yields an overall non-empty area. It assumes that, for each j, the collected fingerprint follows a Gaussian distributions with a standard deviation q, equals to the highest σ_{ij}, among all the fingerprints in T. Therefore, it starts searching by adding a noise level of q_j, then $2q_j$, and so on, till a non-empty overall area is found.

Area Based Probability

The Area-Based Probability (ABP- α) algorithm returns a set of tiles bounded by a probability, α that the object is within the returned area. The probability is also called the *confidence*, and it is an adjustable parameter. ABP's approach to finding the tile set is to compute the likelihood of a received signal vector matching a fingerprint for each tile, and then normalizing these likelihoods given the prior conditions: (1) the sensor must be in the localization floor, and (2) all tiles are a-priori equally likely. ABP then returns the top probability tiles whose sum matches the desired confidence. The confidence controls the accuracy (error) versus the precision (size of returned area) tradeoff, both terms will be defined in more detail later in this chapter. ABP thus stands on a more formal mathematical foundation than SPM (Elnahrawy et al., 2004).

Similar to SPM, signals received from different anchors are assumed to be independent. Using Bayes' rule, ABP computes the probability of being at each tile's location, L_i, on the floor given the fingerprint vector of the sensor using Equations 4, 5 as before, $P(\overline{S_l} \setminus L_i)$. The exact probability is then computed for every tile/location rather than returning the location (tile) $L_{\max}, L_{\max = \arg\max_i} (P(\overline{S_l} \setminus L_i))$.

ABP extends the statistical point based approach discussed above by its final step where it computes the actual probability density of the sensor for each tile given that the sensor must be at exactly one tile, i.e., $\sum_{i=1}^{L} P(L_i \setminus \overline{S_l}) = 1$. Using the resulting density, ABP returns the top probability tiles up to its confidence, α, i.e., the top probability tiles/locations such that their overall probability is $\geq$ *confidence*. Useful values of α have a wide dynamic range between 0.5 and less than 1. While a confidence of 1 returns all the tiles on the floor, picking a useful α is not difficult because in practice some tiles have a much higher probability than the others, while at the same time the difference between these high-probability tiles is small.

Bayesian Networks

Bayes nets are graphical models that encode dependencies and relationships among a set of random variables. The vertices of the graph correspond to the variables and the edges represent dependencies (Gelman et al., 2004). A Bayes net can be utilized to encode the relationship between the received signal and its location based on the signal-versus-distance propagation model described above. The initial parameters of the model are assumed to be unknown, and the training data is then used to compute a probabilistic model for each of the specific parameters. Various Bayesian networks have been designed and tested for localization. They differ in their complexity and assumptions. Here we describe a basic simple model as in Figure 6.

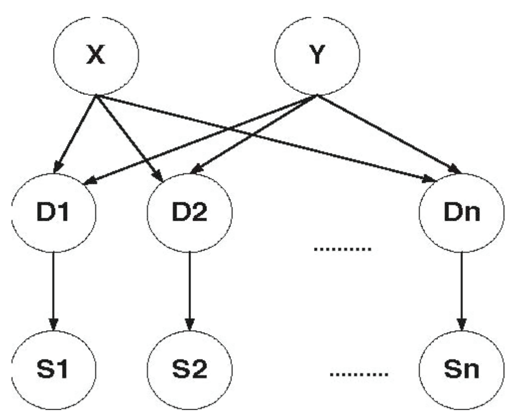
Each random variable s_j, $j = 1 \ldots n$ denotes the expected signal strength from the corresponding anchor node or landmark j. The values of these random variables depend on the Euclidean distance D_j between the landmark's location, (x_j, y_j), and the location where the signal s_j is measured (x,y). The baseline expected value of s_j follows a signal propagation model $s_j = b_{0j} + b_{1j} \times logD_j$, where b_{0j}, b_{1j} are the parameters specific to each landmark j in that environment. The distance $D_j = \sqrt{(x - x_j)^2 + (y - y_j)^2}$ in turn depends on the location (x,y) of the measured signal. The network accounts for noise and outliers by modeling the expected value, s_j, as a probabilistic distribution around the above propagation model, with some variance, τ_j.

Using the training fingerprints T and the fingerprint vector of the sensor, the network then learns the specific values for all the unknown parameters b_{0j}, b_{1j}, τ_j and the joint distribution of the (x,y) location of the sensor. In general, there is no closed form solution for the returned joint distribution of the (x,y) location. A simulator such as Markov Chain Monte Carlo is used to draw samples from the joint density for (x,y) (Madigan et al., 2005; Kleisouris et al., 2006; Heckerman et al., 1995). Samples that give, e.g., a 95% confidence on the density are mapped to tiles and returned as the estimated area. A substantive drawback of this approach is that it yields a large number of disconnected tiles. Although the tiles are concentrated around the most likely location, the scatter is substantial and can interfere with higher-level functions (Madigan et al., 2005; Elnahrawy et al., 2004).

LOCALIZATION PERFORMANCE

There have been many experiments that compare and contrast different individual localization algorithms and how they perform. Detail and in depth evaluations can be found in the references, e.g., (Elnahrawy et al., 2004; Battiti et al., 2002). Due to the limited space, the goal of this section is rather to conclude the performance one would expect when using such a positioning approach based on those extensive studies. It first describes the evaluation metrics usually used for assessment.

Figure 6. A simple Bayesian network used for localization



Evaluation Metrics

A traditional metric to assess the performance of point-based localization is *the localization error*. This is the distance from the true location, *i.e.*, the ground truth, of the sensor and the estimated location. There are many ways to express this metric. Scientists usually use the error CDF (Cumulative Density Function) to plot the error along with the corresponding probability of obtaining it. Figure 7 depicts a sample error CDF curve for a hypothetical approach. As shown the approach yields an error less than 10 feet 80% of the time. For area-based systems two metrics are used; accuracy and precision. *The accuracy* is a generalization of *localization error* to areas. It is the distance between the true position of the sensor and the returned area. It is quantified using CDFs of the order statistics such as the median distance. *Precision* describes the size of the area. A point is hence infinitely precise, but may not be very accurate. On the other hand, the area containing the entire scope of the localization system (e.g., a whole building) would have a high accuracy but poor precision. Accuracy and precision are useful utilities to quantitatively describe the performance of different localization approaches by observing the impact of increased precision (*i.e.*, less area) on accuracy (Bahl et al., 2000; Battiti et al., 2002; Madigan et al., 2005; Hand et al., 2001).

Localization Error

Outdoors, signals travel with no or little obstruction following the free space model. The variance and the attenuations are minimal and the localization error is negligible. Indoor environments, on the other hand are more challenging with obstacles everywhere. Researchers have hence focused on studying the performance of indoor localization rigorously. Figure 8 shows sample performance of those individual approaches described earlier in the Algorithms Section along with their variants in an indoor environment. The error CDFs are plotted. For area-based approaches the median accuracy CDF is used for the comparison. The individual curves are not labeled, as the goal is to show that they have similar performance. Although area-based approaches are better at describing uncertainty, their absolute performance is similar to point-based approaches. No existing approach has a substantial advantage in terms of localization performance.

A general rule of thumb is that using radio received signal strengths with much sampling one can expect a median error of roughly 10 feet and with relatively sparse sampling, every 20 feet, one can still get median errors of 15 feet[1]. Researchers therefore concluded that there are fundamental limitations in indoor localization's performance that cannot be transcended without qualitatively more complex models of the indoor environment, for example by modeling every wall, desk or shelf, or by adding extra hardware in the sensor node above that required for communication, for example, very high frequency clocks to measure the time of arrival (Battiti et al., 2002; Elnahrawy et al., 2004; Kaemarungsi et al., 2004).

Anchor Placement

Placement of the anchor nodes (*i.e.*, landmarks) in the environment also has an impact on the localization performance (Chen et al., 2006; Krishnakumar et al., 2005). The problem is generally an optimization problem with the goal of finding the anchor placement that minimizes the error between the true positions of the sensors and the estimated positions.

Figure 7. Sample Error CDF. The x-axis is the error while the y-axis is the probability. The crossing lines mean that 80% of the time the distance error is less than or equals to10 feet.

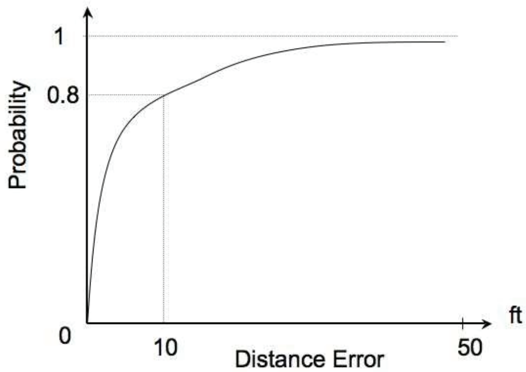
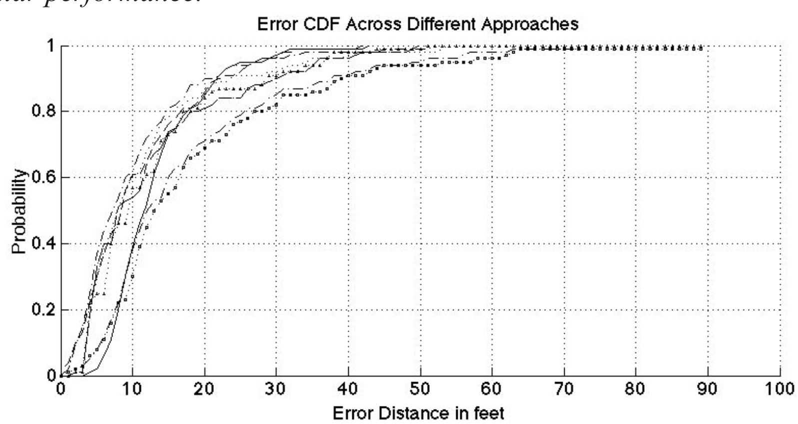


Figure 8. Error across a wide variety of point-and area-based approaches. The CDFs are clustered which shows similar performance.



Linear placement of anchors is the worst layout because the confusion along the other dimension will never get resolved, causing large errors. A uniform deployment in contrast is intuitive and practical. Comparing the error between the two scenarios showed that the latter improves the localization error and makes indoor localization less sensitive to the environment noise and biases. In terms of localization metrics the error CDFs, *e.g.*, similar to the one shown in Figure 7, shift up and to the left compared to their counterparts when using linear anchor placement.

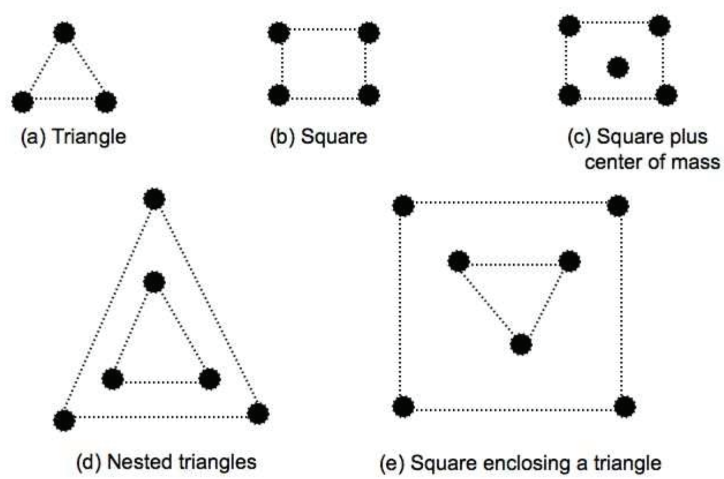
Researchers found that optimal anchor placement follows some simple and symmetric patterns that are easy to achieve. They derived some guidelines that are easy to follow when deciding about placing anchors. The placement patterns should follow simple shapes such as squares and equilateral triangles, or enclosing of them. Complex shapes such as pentagons or hexagons have been shown not to be optimal. Figure 9 shows the patterns for 3, 4, 5, 6, and 7 anchors. Generalization to higher number of anchors is straightforward. It is important to take into consideration the physical constraints of the environment where the network is deployed. A slight deviation from the guidelines in the form of stretching or shrinking the shapes has been shown to be tolerable.

A SAMPLE LOCALIZATION SYSTEM

We showed that individual radio range-based localization approaches have similar limited performance with respect to the localization error. They generally differ however in their applicability to a certain localization scenario or an environment. For example, they vary in how much training they require to achieve a decent performance, how long it takes to compute the estimates (*i.e.*, complexity), how many sensors can be localized simultaneously (*i.e.*, scalability), the ability to compute consistent positions (*i.e.*, outliers), and tolerance to measurement errors and biases. In order to understand these deployment issues it helps to think of the core localization approach as a piece of the higher-level applications (LaMarca et al., 2005; Savvides et al., 2001; Chen et al., 2008). We therefore briefly describe a sample core localization system called GRAIL (General purpose Real-time Adaptable Localization) (SourceForge, 2008; Chen et al., 2008). GRAIL can be integrated seamlessly into any application that utilizes radio positioning via simple Application Program Interfaces (APIs). It has been used to simultaneously localize multiple devices running 802.11 (WiFi), 802.15.4 (ZigBee) and special customized RollCall™ radios (InPoint Systems, 2008).

GRAIL has the following key properties: (1) General Purpose, it supports positioning of a variety of physical modalities, networks or radios, devices and algorithms. Specifically, it localizes any wireless device that transmits packet data. It adopts a centralized approach in order to be able to localize a diverse set of radios. Specifically, the inherent anchor-based localization strategy eliminates the need to install special software on the devices to be localized, e.g., sensors, and therefore enables rapid integration of any radio device. (2) Real-time, it can localize stationary and mobile sensors in real time. (3) Adaptable to indoor noise and multi-path effects. (4) Indoors, GRAIL was originally designed to scale in indoor environments. It naturally works in outdoor environments however.

Figure 9. Layout for optimal anchor placement. As the number of anchors increases from 3 up the placement takes simple triangle or square shapes and then enclosing of them.



A Localization Scenario

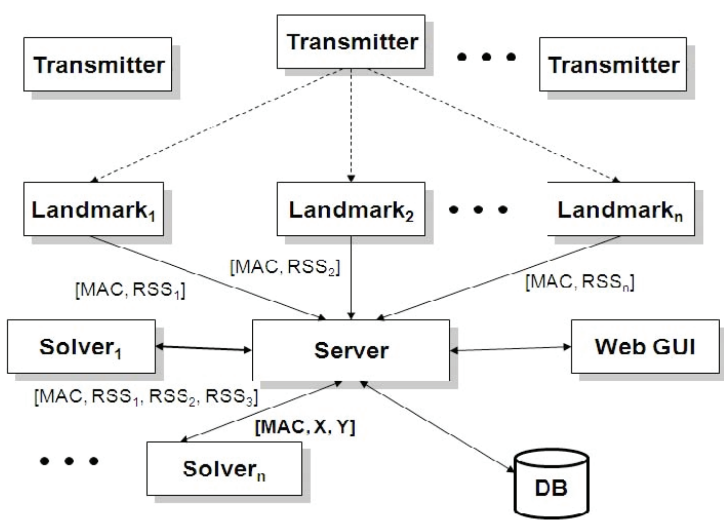
To understand how a specific localization algorithm fits into a system let us describe a typical localization scenario within GRAIL. The localization process starts once a transmitter (*i.e.*, a sensor) in the network transmits packets. The anchor nodes continuously monitor the radio traffic at the packet-level. They timestamp the observed packets and extract the value of the observed physical property, i.e., the received signal strength in this case. The anchors then forward these values along with other header information to a central entity called the server. The server collects traffic data from all the anchor nodes in real time and aggregates those values into fingerprints. It then sends those aggregates to an instance of a solver. The solver entity utilizes a "localization approach" to estimate the locations. It uses an implementation of a localization approach along with some training data and information about the localization environment (anchor placement, for example). Once the localization estimates are computed it sends them to the server. The process ends when the locations are stored in the database or disseminated back to the network nodes.

GRAIL Components

Figure 10 shows the main components of GRAIL: transmitters, landmarks, the server, solvers, the database, and the web server. The transmitters, landmarks and solvers correspond to the sensor nodes, anchors, and localization approaches in our context. We give a brief overview of each of the components and their functionality next.

Much like the Hypertext Transfer Protocol (HTTP) used on the web and the Transaction Language 1 (TL1) used as a standard protocol in the telecommunications equipment industry, all communications between the system components use a simple text-based protocol over TCP sockets. The reader may refer to the references for a detailed description.

Figure 10. Overall architecture for the GRAIL system



- **Transmitter:** Any device transmitting radio packets that needs to be localized.
- **Landmark:** These are the anchor nodes. They passively monitor the existing network traffic and forward the raw data or a summary to the server. Limiting the role of the anchors in the system or network to traffic monitoring enables scalability because regular unsophisticated nodes can easily work as anchors.
- **Database:** The database is a repository for storing the hard state of the GRAIL system. Specifically, it maintains the localization results, fingerprints computed from data samples, anchor information, environment information, and the transmitters in the network.
- **Web Server:** The web server provides a front-end to the GRAIL system. It provides simple authentication and means to interact with users, e.g., to view location estimates, adjust the system parameters or settings. A set of APIs is also provided to support any potential higher-layer application built on top of the core localization.
- **Server:** The localization server is a centralized moderator that collects data samples from the landmarks, summarizes and cleans them, and then passes the data to solvers to compute the unknown positions. The server interacts with a web server as a user interface. It is responsible for storing all the related traffic information and the estimated positions in the database.
- **Solver:** A solver is an implementation of one or more localization approach. It computes the location estimates and sends the results back to the server. GRAIL's architecture is flexible in that multiple different solvers can run simultaneously against a single server, which can improve the overall localization accuracy and allow for load balancing. The server or the user can control which approach to use at every localization attempt depending on the environment, the number of nodes, the training data and so on.

CONCLUSION AND APPLICATIONS

In this chapter we surveyed a set of localization approaches and algorithms for sensor networks. An important conclusion we can draw is that no existing approach or algorithm has been shown to be the best, or even good enough, for most applications. The primary reason is that the location-based services and applications built on top of the localization system are still in their infancy. Until there are more widespread and longer deployments of applications using sensor networks, the performance requirements and resulting cost/performance tradeoffs will not be well understood.

Helping to fill the applications gap are commercial products and solutions that have recently emerged targeting different markets, including wireless security, access control, and workflow management in health-care and industrial plants (Aeroscout, 2008; Ekahau Inc., 2008; Kordinate, 2008; Newbury Networks, 2008; Airtight Networks, 2008). While the mapping from the spatial location to application function is straightforward in security-related applications it is not as intuitive in the latter ones. The general idea is to improve the cost of operation by attaching sensors to employees, inventory and equipment in factories, and additionally caregivers and patients in hospitals. Activities are detected using proximity information from the estimated position and higher-level decisions are taken or actions are made accordingly. For example, if a doctor and a nurse are both localized in the same room as a patient then it will be concluded that this patient is getting treated. If a high traffic of workers has moved within proximity of a factory machine then it might be an indication of a machinery breakdown, and so on.

As an example illustrating the unknown cost-performance tradeoffs, we can consider a healthcare application measuring the productivity of caregivers by measuring activity using location as the base input. In this application, sensors attached to caregivers are mobile and may not form a dense network. Individual lateration approaches are hence very applicable and have better cost-performance, than say, an aggregate approach requiring a high sensor density. Also, the sensor lifetime might only be on the order of a few hours, for example, the length of one shift. However, sensor cost and form factors are critical variables to obtaining good data, as people misplace sensors, or fail to wear them if they are too bulky or look strange.

We are still in exciting times with regards to sensor networks and their applications. Location is one critical piece of the puzzle that has yet to be solved.

REFERENCES

Aeroscout. (2008). *Aeroscout enterprise visibility solutions, white paper*. Retrieved October 10, 2008, from http://www.aeroscout.com

Airtight Networks. (2008). *Airtight networks, white paper.* Retrieved October 10, 2008, from http://www.airtightnetworks.net

Bahl, P., & Padmanabhan, V. (2000). RADAR: An in-building RF-based user location and tracking system. *In proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 775-784).

Bahl, P., & Padmanabhan, V., & Balachandran, A. (2000*). Enhancements to the RADAR user location and tracking system.* Microsoft Research Technical Report.

Battiti, R., & Brunato, M., & Villani, A. (2002). *Statistical Learning Theory for Location Fingerprinting in Wireless LANs* (Tech. Rep. No. DIT-02086). University of Trento, Informatica e Telecomunicazioni.

Chen, Y., Chandrasekaran, G., Elnahrawy, E., Francisco, J. A., Kleisouris, K., Li, X., Martin, R. P., Moore, R. S., & Turgut, B. (2008). GRAIL: A general purpose localization system. *Sensor Review* (pp. 115-124).

Chen, Y., Francisco, J., Trappe, W., & Martin, R. P. (2006). A practical approach to landmark deployment for indoor localization. In *proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Network* (pp. 365-373).

Dohertyl, L., Pister, K. S. J., & ElGhaoui, L. (2001). Convex position estimation in wireless sensor networks. In *proceedings of the IEEE International Conference on Computer Communications* (pp. 1655–1663).

Ekahau, Inc. (2008). *The Ekahau Positioning Engine*. Retrieved October 10, 2008, from http://www.ekahau.com

Elnahrawy, E., Li, X., & Martin, R. P. (2004). The Limits of Localization Using Signal Strength: A Comparative Study. *In proceedings of IEEE International Conference on Sensor and Ad hoc Communications and Networks* (pp. 406–414).

Elnahrawy, E., & Li, X., & Martin, R. P. (2004). Using area-based presentations and metrics for localization systems in wireless LANs. *In LCN's Fourth International IEEE Workshop on Wireless Local Networks* (pp. 650–657).

Fang, L., Du, W., & Ning, P. (2005). A beacon-less location discovery scheme for wireless sensor networks. *In proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 161–171).

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian Data Analysis.* Chapman and Hall.

Hazas, M., & Ward, A. (2003). A high performance privacy-oriented location system. *In proceedings of the IEEE International Conference on Pervasive Computing and Communications* (pp. 216).

Hand, D., Mannila, H., & Smyth, P. (Ed.) (2001). *Principles of Data Mining.* The MIT Press.

He, T., & Huang, C., & Blum, B., & Stankovic, J., & Abdelzaher, T. (2003). Range-free localization schemes in large scale sensor networks. *In proceedings of the ACM International Conference on Mobile Computing and Networking* (pp. 81–95).

Heckerman, D. (1995). *A tutorial on learning with Bayesian networks* (Tech. Rep. No. TR-95-06). Microsoft Research.

InPoint Systems. (2008). *White paper.* Retrieved October 10, 2008, from http://inpointsys.com

Kaemarungsi, K., & Krishnamurthy, P. (2004). Modeling of indoor positioning systems based on location fingerprinting. *In proceedings of Twenty Third Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 1012-1022).

Kleisouris, K., & Martin, R. P. (2006). Reducing the computational cost of Bayesian indoor positioning systems. *In proceedings of the Third IEEE International Conference on Sensor and Ad hoc Communications and Networks* (pp. 555-564).

Kordinate LLC. (2008). http://www.kordinate.com

Krishnakumar, A., & Krishnan, P. (2005). On the accuracy of signal strength-based location estimation techniques. In *proceedings of Twenty Fourth Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 642-650).

Krishnan, P., & Krishnakumar, A. S., & Ju, W., & Mallows, C., & Ganu, S. (2004). A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks. *In proceedings of the IEEE International Conference on Computer Communications* (pp. 1001–1011).

Ladd, A. M., Bekris, K. E., Rudys, A., Marceau, G., Kavraki, L. E., & Wallach, D. S. (2002). Robotics-based location sensing using wireless Ethernet. *In proceedings of The Eighth ACM International Conference on Mobile Computing and Networking* (pp. 227–238).

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., & Schilit, B. (2005). Place lab: Device positioning using radio beacons in the wild. *In proceedings of Pervasive Computing* (pp. 116-133).

Lim, H., Kung, L., Hou, J., & Luo, H. (2006). Zero-Configuration, robust indoor localization: Theory and experimentation. *In proceedings of the IEEE International Conference on Computer Communications* (pp. 1-12).

Lorincz, K., & Welsh, M. (2006). Motetrack: A robust, decentralized approach to RF-based location tracking. *Springer Personal and Ubiquitous Computing, Special Issue on Location and Context-Awareness*.

Madigan, D., Elnahrawy, E., Martin, R. P., Ju, W. H., Krishnan, P., & Krishnakumar, A. S. (2005). Bayesian indoor positioning systems. *In proceedings of Twenty Fourth Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 1217-1227).

Moore, D., Leonard, J., Rus, D., & Teller, S. (2004). Robust distributed network localization with range measurements. *In proceedings of Conference on Embedded Networked Sensor Systems* (pp. 50–61).

Newbury Networks. (2008). *Newbury networks, white paper.* Retrieved October 10, 2008, from http://www.newburynetworks.com

Niculescu, D., & Nath, B. (2001). Ad hoc positioning system (APS). In *proceedings of the IEEE Global Telecommunications Conference* (pp. 2926–2931).

Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. O., Moses, R. L., & Correal, N. S. (2005). Locating the nodes. *IEEE Signal Processing Magazine*.

Priyantha, N., Chakraborty, A., & Balakrishnan, H. (2000). The cricket location-support system. *In proceedings of the ACM International Conference on Mobile Computing and Networking* (pp. 32–43).

Roos, T., Myllymaki, P., & Tirri, H. (2002). A Statistical Modeling Approach to Location Estimation. *IEEE Transactions on Mobile Computing* (pp. 59–69).

Savvides, A., Han, C. C., & Srivastava, M. (2001). Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *proceedings of the Seventh Annual ACM International Conference on Mobile Computing and Networking* (pp. 166–179).

Shang, Y., W. Ruml, W., Zhang, Y., & Fromherz, M. P. J. (2003). Localization from mere connectivity. In *proceedings of the ACM International Symposium on Mobile Ad-Hoc Networking and Computing* (pp. 201–212).

Smailagic, A., & Kogan, D. (2002). Location sensing and privacy in a context aware computing environment. *IEEE Wireless Communications* (pp. 10-17).

SourceForge.net. (2008). The GRAIL Real Time Location Service, documentation and source code. Retrieved October 10, 2008, from http://grailrtls.sourceforge.net.

Stoleru, R., & Stankovic, J. (2004). Probability grid: A location estimation scheme for wireless sensor networks. *In proceedings of IEEE International Conference on Sensor and Ad hoc Communications and Networks*.

Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. In *proceedings of ACM Transactions on Information Systems* (pp. 91–102).

Youssef, M., & Agrawala, A. (2004). Handling samples correlation in the HORUS system. *In proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 1023–1031).

Youssef, M., Agrawal, A., & Shankar, A. U. (2003). WLAN location determination via clustering and probability distributions. *In proceedings of IEEE International Conference on Pervasive Computing and Communications* (pp. 143–150).

ENDNOTE

[1] The 802.11 Wireless Local Area Network (WLAN) technology was used for the evaluations, however the results apply to all radio-based localization.