

Algorithms for Location Estimation Based on RSSI Sampling

Charalampos Papamantou, Franco P. Preparata, and Roberto Tamassia

Department of Computer Science and Center for Geometric Computing
Brown University
{cpap,franco,rt}@cs.brown.edu

Abstract. In this paper, we re-examine the RSSI measurement model for location estimation and provide the first detailed formulation of the probability distribution of the position of a sensor node. We also show how to use this probabilistic model to efficiently compute a good estimation of the position of the sensor node by sampling multiple readings from the beacons (where we do not merely use the mean of the samples) and then minimizing a function with an acceptable computational effort. The results of the simulation of our method in TOSSIM indicate that the location of the sensor node can be computed in a small amount of time and that the quality of the solution is competitive with previous approaches.

1 Introduction

Estimating the location of a roaming sensor is a fundamental task for most sensor networks applications. For example, if a sensor network has been deployed to provide protection against fire (in this case, sensor nodes report a sudden increase in temperature), we want to know the location of the sensor that triggers an alert so that action can be taken accordingly. Additionally, some routing protocols for sensor networks, such as geographical routing [15, 44], make routing decisions based on the knowledge of the locations of the sensor nodes. Common location estimation protocols that are widely adopted in practice assume that there are some fixed nodes (base stations) that know their location which are called *beacons*. These nodes send a signal to the sensor nodes that want to determine their location. According to the intensity (or for example the angle) of this signal, the sensor node can have an estimate of the distance between them and the beacons.

After performing a certain number of such measurements for different beacons, the sensor node has to combine all this information (for RSSI (Received Signal Strength Indicator), this information is the power of each individual signal and the coordinates of the corresponding transmitter) in order to estimate its location. However one could ask the following question: Why cannot we use a Geographic Positioning System (GPS) to efficiently achieve the task of localization? The answer is that a GPS requires a strong computing platform which is not available in sensor networks. Sensor nodes are typically very low-computing

power units that can efficiently perform only basic arithmetic operations; Requiring the execution of complex arithmetic operations on a sensor node would entail a quick depletion of its battery which is not desirable for most practical applications. Finally, the localization problem gets even more difficult because the available power on the sensor node is limited: therefore no accurate measurements of the signal can be made (since an accurate measurement requires more computing power) which means that the measurements are prone to errors. This is something that should also be taken into consideration and treated accordingly.

Therefore, any location estimation algorithm should have the following requirements:

1. The sensor node should avoid complex and time consuming computations, which would deplete its energy supply (typically a low-cost battery) rapidly;
2. The computations should take into consideration the error in the measurements, which can be large.

1.1 Related Work and Observations

There are several proposed location estimation protocols for sensor networks, see, e.g., [5, 7, 8, 11, 23, 24, 25, 26, 27]. All these protocols use the same model, where some nodes *know* their location (either because they are fixed or by using GPS) and are called *beacons* or *anchor nodes*, and some other nodes, called *sensor nodes*, estimate their location using the information they receive from the beacons. This information consists of the beacons' coordinates and of features of the beacon signal, such as the *received signal strength indicator* (RSSI) or the *time difference of arrival* (TDoA). Also, other protocols (e.g., [26]) are based on the capability of the nodes to sense the angle from which a signal is received. Recently [38] presented a solution for aerial localization and [22] proposed a solution where the localization is based on adopting slightly different periodic signal frequencies (interferometric positioning). This solution [22] is very competitive and achieves very good precision. However the computations used are very intensive.

Several previous approaches use computationally demanding methods, such as convex optimization [8], systems of complex equations [26], *minimum mean square error* (MMSE) methods [6, 33], and Kalman filters [34]. In these approaches, the measurement model is not adequately analyzed and the error is assumed to be small, which is not the case in most real applications of sensor networks.

Other approaches, notably [17, 30, 36], estimate the location of a node using the RSSI method (analyzed in [31]), which is the most realistic model for sensor network communication. In [17], the authors evaluate the ability of a sensor network to estimate the location of sensor nodes. They assume that the location of the sensor node is known and develop arguments concerning the probability that the network will detect this location. They use the RSSI error model to analyze the problem of *evaluating* the ability of the sensor network to locate a

sensor node. However, they do not describe how their algorithms can be implemented on a sensor node to estimate its own location. Moreover, their method does not take into account the basic parameters of the RSSI model (standard deviation and path loss exponent) and thus gives incorrect results.

In this paper, we formulate the correct probability distribution of the position of a sensor node based on one reading produced with the RSSI model. Due to the errors implicit in the RSSI model, it is unrealistic to try to compute a good estimation of the location of the sensor node based only on a single measurement (or even few measurements) from each beacon. Such an approach would be so inaccurate to make the estimate practically worthless. Notwithstanding this difficulty, we show that a reliable estimation of the location can be achieved by processing a reasonably small number of readings of the signals.

Especially for indoor positioning systems, this is an assumption that has been extensively used. For example, in [13, 14], the position estimation is based on a *location fingerprint* $\mathbf{t} = [t_1 \ t_2 \ \dots \ t_N]$, where N is the number of beacons and t_i ($i = 1, \dots, N$) is the mean value of the received signal strength over a certain time window. Also, in [3, 9, 20, 42], experiments with various sample sizes are presented where the samples are used to compute certain features of the signal strength such as the standard deviation and the path loss exponent. Finally, [12] presents simulations that use various number of samples, where more than 50 samples to *filter out* the errors in the probability distribution are used.

We show that using only the mean of the measurements is not a correct procedure, due to the lognormal distribution of the distance from the beacon (see Theorem 3). Instead of using directly the mean value, we use another value that is adequate according to the specific underlying probability distribution of the distance. The number of samples that are used vary from 20 to 60 and obviously the accuracy of the computed location grows with the number of samples. Finally, once the sampling has been performed, we show how to seek the minimum of a function that approximates the actual location with small computational effort.

1.2 Our Contributions

The main contributions of this paper are as follows:

1. We evaluate the probability that a sensor node lies within a certain region, given that the power received from the beacons is modeled with RSSI. To the best of our knowledge, this is the first detailed formulation of the probability distribution of the position of a sensor node. We show that unlike the normal distribution of the received power, the probability distribution of the actual position is lognormal. Thus, we give evidence to the role of the parameters σ and n in the probability distribution of the actual distance, where σ is the standard deviation of the normal variable that models the power received by the sensor and n is a parameter, called *path loss exponent*, that depends on the transmission medium. In previous approaches [17], the probability distributions used did not exhibit dependency on these two variables.
2. We present a method for estimating the location of a node from multiple sample power readings from the beacons. Our method computes the *expected*

value of the received power and combines it with the mean and the standard deviation of the sample readings using a steepest descent approach [37]. We show that our method is simple and efficient and provides a good estimation of the position. Note that using multiple sample readings is necessary for a reliable location estimation since the probability distribution of the location for a single sample implies that the domain within which the sensor lies with high probability has large area.

3. We describe an implementation of our location estimation algorithm that is suitable for execution on standard sensor hardware and we analyze the results of an extensive simulation of the execution of the algorithm in TOSSIM [10, 18]. Our simulation shows that our method has accuracy that is comparable to or better than that of previous methods.

1.3 Organization of the Paper

The rest of this paper is organized as follows. In Section 2, we overview the RSSI model, give formulas for the probability distribution of the position of a sensor node due to power measurements, and show how to estimate the actual distance given a set of sample power readings. We develop an efficient algorithm for location estimation and analyze its running time in Section 3. Finally, in Section 4, we report the results of the simulation and present a comparison (in terms of localization error) of our method with previous approaches. Concluding remarks are in Section 5.

2 Theoretical Framework

This section provides a formal probabilistic framework for estimation of the position of a sensor node from power measurements.

2.1 RSSI Model

Suppose we are given a region of the plane with k beacon nodes b_1, b_2, \dots, b_k (nodes of known location). The coordinates of the beacons are (x_i, y_i) for $i = 1, \dots, k$. The beacons transmit information about their location with a signal of normalized intensity to a sensor node s that does not know its location. Based on the locations of the beacons and the estimated distances from the beacons (computed from the received signals), the sensor is to compute its actual location.

Among the several models proposed for estimating the distance between a beacon and a sensor node, the most realistic and commonly used one is the *received signal strength indicator* model (RSSI) [31]. In this model, the beacon broadcasts signal to all sensors and the sensors can estimate the distance between them and the beacons on the basis of the strength of the signals they receive.

Let b_i be a beacon located at (x_i, y_i) and s a sensor node located at (x, y) . We define the *relative error* ϵ_i pertaining to b_i as follows. Suppose that s reads a distance \hat{r}_i , while the actual distance is r_i . The relative error is

$$\epsilon_i = \frac{\hat{r}_i}{r_i} - 1 \in [-1, +\infty). \quad (1)$$

The commonly accepted transmission model [31] expresses the received power p_i (in dBm) as

$$p_i = p_0 + 10n \log \left(\frac{r_i}{r_0} \right) \quad (2)$$

where p_0 is the received power in dBm at a reference distance r_0 and n is the path loss exponent which is a constant depending on the transmission medium (indoors, outdoors) and ranges typically from 2 to 4. In some environments, such as buildings, stadiums and other indoor environments, the path loss exponent can reach values in the range of 4 to 6. On the other hand, a waveguide type of propagation may occur in tunnels, where the path loss exponent drops below 2.

We recall that if the received power in mW at a point k is P_k , and $P_{k'}$ is the received power at some reference point k' (again in mW), then the received power p_k in dBm at point k is defined as

$$p_k = 10 \log (P_k/P_{k'}). \quad (3)$$

The measured power, however, differs from that given in Equation (2); due to channel fading (variation of the received signal power caused by changes in transmission medium or path), the measured power is $\hat{p}_i = p_i + x$. The random variable x represents the medium-scale channel fading and is typically modelled as Gaussian zero-mean with variance σ^2 (in dBm). Typically, σ is as low as 4 and as high as 12 (this implies that the error may be large). Inserting \hat{p}_i and \hat{r}_i into (2), we get

$$\hat{p}_i = p_0 + 10n \log \left(\frac{\hat{r}_i}{r_0} \right) \quad (4)$$

where now the measured power \hat{p}_i in dBm relates to the measured distance \hat{r}_i by the sensor. By combining the above equations, we get that the relation between the measured distance and the actual distance is

$$\hat{r}_i = r_i 10^{\frac{x}{10n}} \quad (5)$$

which gives

$$\epsilon_i = 10^{\frac{x}{10n}} - 1. \quad (6)$$

2.2 Probability Distributions

In this section, we present the probability distribution of the position of the sensor node based on measurements of one or more beacons. We denote e^x with $\exp(x)$.

Theorem 1. *Let b_i be a beacon node located at (x_i, y_i) sending information to a sensor node under the RSSI model with standard deviation σ and path loss exponent n . Let \hat{r}_i be the measured distance from beacon node b_i at the sensor*

node. We have that the probability density function of the actual position (x, y) of the sensor node is given by

$$P_{X,Y}^{(i)}(x, y) = \frac{10n \exp \left(- \left(10n \log \frac{\hat{r}_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} \right)^2 / 2\sigma^2 \right)}{2\pi\sigma\sqrt{2\pi} \ln(10) ((x-x_i)^2 + (y-y_i)^2)}.$$

To simplify the notation, we denote the probability distribution due to beacon b_i with

$$\Phi_{b_i}(x, y) = P_{X,Y}^{(i)}(x, y).$$

The previous argument can be extended to a finite set of beacons $B = \{b_1, \dots, b_k\}$ yielding the following theorem:

Theorem 2. *Let $B = \{b_1, b_2, \dots, b_k\}$ be a set of beacons sending information to a sensor node under the RSSI model with standard deviation σ and path loss exponent n . If the measured distance from beacon node b_i at the sensor node is \hat{r}_i ($i = 1, \dots, k$), then the probability density function (due to all the beacons in B) of the actual position (x, y) of the sensor node is given by*

$$\Phi_{(B)}(x, y) = \frac{\prod_{i=1}^k \Phi_{b_i}(x, y)}{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(\prod_{i=1}^k \Phi_{b_i}(x, y) \right) dy dx}$$

where $\Phi_{b_i}(x, y)$ is the probability distribution due to beacon b_i , as defined in Theorem 1.

As we will see later, if we use only one measurement, we may end up with a density function having more than one maximum. Moreover, the computation of this maximum on a sensor node is a difficult task since there is no simple analytical expression for the maximum of the probability distribution. Also, there is no suitable analytical expression for the integrals needed to compute the probability for a certain region. Due to space limitations, the proofs of the two theorems can be found in the full version of the paper.

2.3 Samples of Measurements

In the previous sections, we have examined the probability distribution of the sensor's position based on a single measurement. This setting, however, can give rise to unacceptable errors for the values of σ ($\sigma = 4, 6, 8$) reported in the literature [31]. A consequence of this situation is that we may be unable to define a disk containing the sensor's location with an acceptable degree γ of confidence (say, $\gamma > 0.9$). Additionally, if our practice is based on only one reading, there is no way for the sensor to estimate the standard deviation σ (this is actually the standard deviation of the Gaussian random variable x introduced in Section 2.1) of each beacon. This parameter σ —conveniently assumed to be known in

Section 2.1—must be estimated in practice as it is needed in the computations (see below).

To overcome these difficulties, we show in this section how we can obtain a good estimate of the location of the sensor node based on small number of readings. Especially for indoor positioning systems, this is an assumption that has been extensively used. For example, in [13, 14], the position estimation is based on a *location fingerprint* $\mathbf{t} = [t_1 \ t_2 \ \dots \ t_N]$, where N is the number of beacons and t_i ($i = 1, \dots, N$) is the mean value of the received signal strength from the i -th beacon over a certain time window. Note that t_i denotes the *measured* power \hat{p}_i that appears in Equation 4. Hence, the “measured power” location fingerprint \mathbf{t} can be transformed to a location fingerprint \mathbf{r} of “measured radii” by using Equation 4, since the reference values p_0 and d_0 are known. The number of samples that are used vary from 20 to 60 and obviously this number affects the accuracy of the computed location. Also, in [3, 9, 20, 42], experiments with various sample sizes are presented where the samples are used to compute certain features of the signal strength such as the standard deviation and the path loss exponent.

Suppose now that we use a sample of k readings from beacon b_i . We have a sequence of radii $r_{i1}, r_{i2}, \dots, r_{ik}$. Let \bar{r}_i, \bar{s}_i denote the *unbiased* estimators of the value $\mathbf{E}[r_i]$ and of the standard deviation $\sqrt{\text{Var}(r_i)}$ of the underlying distribution of the measured radii \hat{r}_i ($i = 1, \dots, 3$) respectively. We have the following result that relates estimates of the actual distance and the standard deviation with reference to a beacon b_i with features of the lognormal distribution:

Theorem 3. *Suppose a sensor node reads k distance samples $r_{i1}, r_{i2}, \dots, r_{ik}$ from a beacon b_i that is modeled with the RSSI of path loss exponent n and standard deviation σ_i . If \bar{r}_i is the sample mean and \bar{s}_i is the sample standard deviation then we have the following:*

1. *The estimate of the square of the actual distance r_i^2 from beacon b_i is*

$$\frac{\bar{r}_i^4}{\bar{r}_i^2 + \bar{s}_i^2}.$$

2. *The estimate of the square of the standard deviation σ_i^2 is*

$$\frac{100n^2}{\ln^2(10)} \ln \left[1 + \left(\frac{\bar{s}_i}{\bar{r}_i} \right)^2 \right].$$

Note that the above theorem indicates that the quality of estimation of the actual distance is heavily dependent on the estimation of the distribution of the measured radii.

3 Location Estimation

In this section, we develop an algorithm for location estimation based on several samples. This algorithm does not involve any complex calculations (such as square roots) which is very important to consider when we develop algorithms to be executed on sensor nodes, due to the sensor’s modest computing power.

3.1 Algorithm

As we saw in the previous section, after completing the sampling procedure, we derive estimates for r_1^2, r_2^2, r_3^2 , given by Theorem 3. Our aim is to formulate a function whose minimum will yield a good approximation of the sensor's location. This function should be convex and also its derivatives should not include roots. Suppose now we have three beacons located at $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Let $f(x, y)$ be the function

$$f(x, y) = \sum_{i=1}^3 ((x - x_i)^2 + (y - y_i)^2 - r_i^2)^2. \quad (7)$$

Note that if all 3 circles intersect at the same point (x_0, y_0) , this function has minimum 0 at (x_0, y_0) . Unfortunately, minimizing that function is not an easy task, if we are restricted on the available primitives. Hence we are going to use methods that are based on the gradient of the function. The good feature about such methods is that we can get to a point very close to the minimum in a small number of computationally simple iterations. Indeed, let

$$\alpha(x, y) = \frac{\partial f(x, y)}{\partial x} = 4 \sum_{i=1}^3 (x - x_i)((x - x_i)^2 + (y - y_i)^2 - r_i^2) \quad (8)$$

and

$$\beta(x, y) = \frac{\partial f(x, y)}{\partial y} = 4 \sum_{i=1}^3 (y - y_i)((x - x_i)^2 + (y - y_i)^2 - r_i^2) \quad (9)$$

be the partial derivatives of f . Note that the above expressions are computable on a sensor node. The function $z = f(x, y)$ describes a convex solid surface with obvious definitions of “interior” and “exterior”. Initially, we make a guess for our point (this is required by all steepest descent methods [37]). Suppose, for uniformity, we choose as our initial point (x_0, y_0) the centroid of the beacon triangle. We compute the vector \mathbf{v} which is orthogonal to the tangent plane \mathcal{T} and pointing toward the exterior. Hence $\mathbf{v} = [\alpha(x_0, y_0) \beta(x_0, y_0) -1]^T$. Let now \mathcal{P} be the *vertical* plane containing \mathbf{v} applied to $(x_0, y_0, f(x_0, y_0))$. Since \mathcal{P} is a vertical plane, any normal vector \mathbf{w} of \mathcal{P} will have a zero z -component. Additionally, \mathbf{w} is orthogonal to \mathbf{v} and therefore may be chosen as $\mathbf{w} = [-\beta(x_0, y_0) \alpha(x_0, y_0) 0]^T$. We seek the vector \mathbf{q} pointing towards the minimum of the function. Such vector belongs to \mathcal{P} and is orthogonal to \mathbf{v} (\mathbf{q} is orthogonal both to \mathbf{v} and \mathbf{w}), i.e.,

$$\mathbf{q} = [\alpha(x_0, y_0) \beta(x_0, y_0) \alpha^2(x_0, y_0) + \beta^2(x_0, y_0)]^T.$$

Now we compute the intersection point (x'_0, y'_0) of the line (with the surface $z = 0$) passing by $(x_0, y_0, f(x_0, y_0))$ which is collinear with the direction \mathbf{q} and the xy -plane. The parametric equation of this line is $(x, y, z) = (x_0 + t\mathbf{q}_x, y_0 + t\mathbf{q}_y, f(x_0, y_0) + t\mathbf{q}_z)$ for all $t \in \mathbf{R}$. The new point (x'_0, y'_0) is then given by

$$(x'_0, y'_0) = \left(x_0 - \frac{f(x_0, y_0)\alpha(x_0, y_0)}{\alpha^2(x_0, y_0) + \beta^2(x_0, y_0)}, y_0 - \frac{f(x_0, y_0)\beta(x_0, y_0)}{\alpha^2(x_0, y_0) + \beta^2(x_0, y_0)} \right).$$

The described process gives a new point (x'_0, y'_0) . This point is expectedly closer to the point that corresponds to the minimum of f as we follow the direction of the gradient as long as the products $\alpha(x_0, y_0)\alpha(x'_0, y'_0) > 0$ and $\beta(x_0, y_0)\beta(x'_0, y'_0) > 0$. When this condition no longer holds, we have “overshot”; to remedy, we backtrack to the previous point referred here as (x, y) and apply a typical steepest descent method with very small rate λ . We therefore compute our new point (x', y') by setting

$$(x', y') = (x - \lambda\alpha(x, y), y - \lambda\beta(x, y)). \quad (10)$$

We continue this process until the gradients $\alpha(x, y)$, $\beta(x, y)$ change sign. At that point we stop and we report the final point as our estimation. Here we should emphasize the fact that it is important to take samples of adequate size. Taking samples implies a better behavior for f , meaning that there would be only one minimum and therefore the algorithm will quickly converge to the minimum. As far as the value of the variable λ is concerned, this variable is chosen to be small enough and inversely proportional to the size of the grid since these features of λ force the second repeat loop of the algorithm to converge quickly. This has been observed in the experiments. For the experiments, the value of λ is equal to $1000^{-m/100}$.

3.2 Complexity and Limitations

The most expensive part of the presented algorithm is the sampling procedure and the computation of the estimates \bar{r}_i^2 and \bar{s}_i^2 . These steps take time $O(k)$, where k is the number of samples. There is an obvious trade-off between accuracy and power consumption. Also, the computation executed on the sensor nodes depends on the time the gradient methods take to converge, which is generally small for a well-behaved function. For the other parts of the algorithm there are closed formulas, so we can assume that they take time $O(1)$. We can also see that the exact number of multiplications needed by the presented program is $(7k + 8) + 10n_1 + 4n_2$, where k is the size of the sample, n_1 is the number of iterations of the first repeat loop and n_2 is the number of iterations of the second repeat loop.

The size of the code of the program (ROM) written in NesC [10] is 47K whereas the amount of memory (RAM) needed to execute this program in TOSSIM [18] (see Section 4) is 637K. As far as the complexity of the closed formulas computation is concerned it is realistic to assume that the involved in closed-formula calculation can be executed on a sensor node (essentially floating point operations). For example, there are micro-controllers, such as the ATM-Mega128L [2] and MSP430 [39], which have very rich instruction sets. Finally, a hardware multiplier allows floating-point arithmetic to be carried out [21].

4 Simulation

In this section, we present and analyze extensive simulation results of our method. We have run our experiments with TOSSIM [10, 18], a widely used simulator of the TinyOS operating system for sensor networks.

We executed our simulations in a square of area $m \times m$ cells, where $m = 50, 100, 200$. The three beacons are placed in positions that form a well conditioned triangle (well-conditioning is synonymous with the fact that the function $f(x, y)$ has a single global minimum). Namely, the first beacon is placed at $(0, 0)$, the second beacon is placed at $(m, 0)$ and the third beacon is placed at $(m/2, 3m/4)$. The standard deviations of the three beacons $\sigma_1, \sigma_2, \sigma_3$ are set to 4 and the path loss exponent n is set to 2. We also recall that we set the variable λ that appears in Equation 10 equal to $1000^{-\frac{m}{100}}$, where m is the dimension of the grid. Finally the measured distance is computed using Equation 5.

We measure the execution time of the algorithm implemented in NesC [10] (that runs in TinyOS) and the average number of iterations of the **repeat** loops over 1000 runs. We also determine the mean of the distance d between the actual point and the computed point. We use the ratio $\frac{d}{m}$ to evaluate the quality of the solution computed by the algorithm. Note that this metric was proposed in [41].

The results obtained for different numbers of samples and different sizes of grids are shown in Table 1, where m is the dimension of the square region, k is the number of samples, the time is counted in milliseconds (we count the exact time that the simulated processor in TOSSIM takes to execute this program), n_1 is the number of iterations of the first **repeat** loop, n_2 is the number of iterations of the second **repeat** loop and d is the mean of the distance between the actual point and the computed point.

The simulation results with TOSSIM show that the sensor node can execute the algorithm in a small amount of time. This time is proportional to the number of samples we use each time, which indicates that the sampling procedure dominates the execution time on the sensor node. Only up to six iterations ($n_1 + n_2$) are enough to compute an estimation of the actual point and the quality of the estimation is dependent on the number of the samples. Additionally, note that for various grid sizes, the algorithm has a uniform behavior since the ratio $\frac{d}{m}$ is similar for different sizes of the grid. Finally, in all cases, the solution we get is better for larger sizes of samples. For example, though not practical for power consumption issues, for 200 samples the estimation gets even better ($d/m = 0.041$ for $m = 200$).

Table 1. Results of the simulation in TOSSIM for an $m \times m$ square grid ($m = 50, 100, 200$): execution time, average number of iterations of the program (n_1, n_2), localization error (d) and ratio $\frac{d}{m}$ for various samples sizes over 1000 runs

$m \times m$	k	time (ms)	n_1	n_2	d	d/m
50×50	20	0.140	4.045	1.081	5.018	0.1003
50×50	40	0.230	4.226	1.061	3.774	0.0745
100×100	20	0.130	3.670	1.142	9.986	0.0986
100×100	40	0.240	3.817	1.046	7.634	0.0763
200×200	20	0.120	3.640	2.736	19.977	0.0998
200×200	40	0.240	3.820	2.323	14.957	0.0747

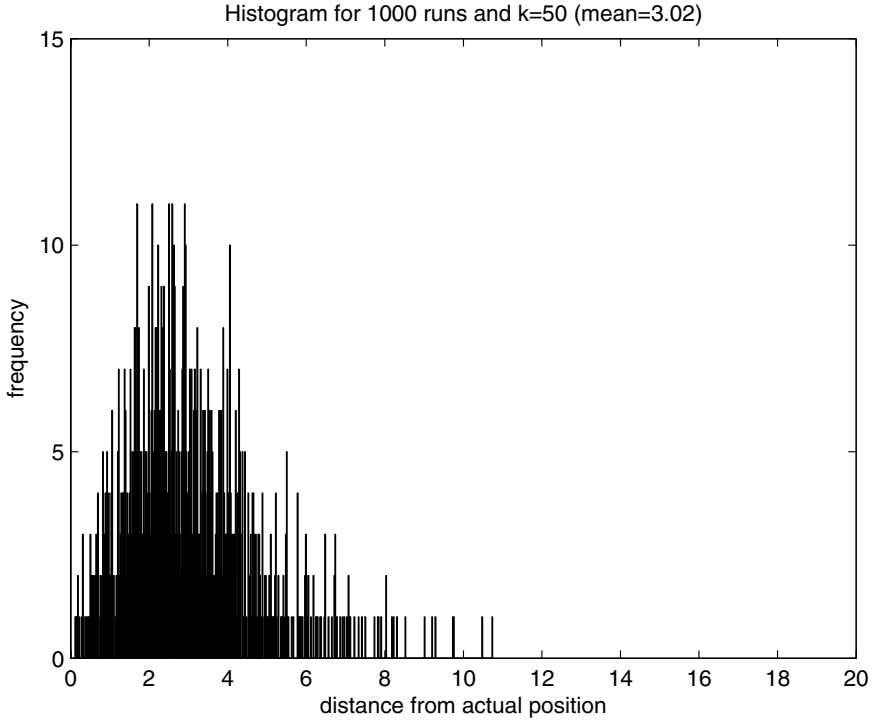


Fig. 1. Histogram for 1000 runs and $k = 50$

We show in Figure 1 the probability distribution of the distance of the computed point from the actual point derived for 1000 runs and for $m = 50$. In particular, we show the plot of the distribution of the error of the estimation (which we define as the distance of the computed point from the actual point) for sample size $k = 50$. The mean of the error for these measurements is about 3 for $k = 50$. Also, we observe that the distribution of the error appears to follow a lognormal distribution.

Finally, Table 2 compares location estimation algorithms. We present the average localization error d , the area A of the field where the experiments are executed and the ratio $\frac{d}{\sqrt{A}}$. We use as a comparison measure the quantity $\frac{d}{\sqrt{A}}$, which for our algorithm is bounded by 0.1 (this is what we get for the smallest number of samples $k = 20$). However, for a number of samples $k = 300$ (though not so practical) we can get even smaller values (for example for $k = 300$ and for an area 1000×1000 we get $\frac{d}{\sqrt{A}} = 0.026$).

From Table 2, we see that our method gives always better or as good results as the results obtained by the existing methods (except for [38] where, however, the time needed for localization ranges from 10 milliseconds to 2 minutes — something that is also observed in [22]— where the precision is even better). Also, if we slightly increase the number of the samples we use, we get very good results and the ratio drops substantially (for example for $k = 60$ we get a ratio

Table 2. Comparison of existing work. In each row, we display the bibliographic reference and the respective average localization error (d), the size of the area of the experiments A , the ratio $\frac{d}{\sqrt{A}}$ and finally the number of samples used by each method. Note that it is not always feasible to compare between different methods since the settings used can be different. N.A. stands for “not applicable” and it means that the certain method does not refer explicitly to the number of samples used or that the sampling technique is not used.

reference	error d	simulation area A	d/\sqrt{A}	number of samples
[3]	3	22.5×45.5	0.090	20
[32]	3	16×40	0.118	N.A.
[4]	4	35×40	0.107	250
[29]	7.62	13.71×32	0.360	40
[1]	3	500	0.130	N.A.
[30]	6	60×60	0.100	N.A.
[5]	1.83	10×10	0.183	20
[35]	0.8	6×6	0.130	50
[19]	13	18751	0.094	N.A.
[43]	10	26×49	0.280	N.A.
[38]	3.5	60×120	0.058	N.A.
[16]	0.82	5×5	0.164	N.A.
our scheme	4.350	50×50	$\simeq 0.087$	25
our scheme	3.020	50×50	$\simeq 0.064$	50

$\frac{d}{\sqrt{A}} = 0.06$). Note that previous methods use more than three beacon nodes (see for example [28] where $O(m)$ beacons are placed in the area of localization for an $m \times m$ grid).

5 Conclusions

In this paper, we have analyzed the RSSI model for location estimation in sensor networks. Given a normal distribution for the error in dBm, we compute the correct probability distribution of the sensor’s location and then we adopt this probability distribution in a theoretical analysis of sampling the measurements for location estimation. We finally give a simple algorithm that can be executed on sensor nodes; its complexity, for a constant number of beacons, is proportional to the size of the sample.

Location estimation in sensor networks presents several trade-offs. If higher accuracy is desired, one has to deploy more beacons or use more samples. Using a large number of beacons and samples causes significant energy consumption. The energy-optimal case occurs when only three beacons are deployed and an estimation of the actual point is based on the probability distribution computed by taking into consideration only one measurement. This solution, however, gives unacceptable errors. Additionally, performing computations with the exact probability distribution is unrealistic, since it involves complex formulas. Hence, were we to depend on few measurements, off-line computed data must be stored as

tables within the sensor, which immediately creates a storage problem. However, one can use more samples, thus increasing energy consumption.

Acknowledgments

This research was supported by the U.S. National Science Foundation under grants IIS-0324846 and CCF-0830149 and by the Center for Geometric Computing and the Kanellakis Fellowship at Brown University. The views in this paper do not necessarily reflect the views of the sponsors. We thank Goce Trajcevski for useful discussions.

References

- [1] Alippi, C., Vanini, G.: A RSSI-based and calibrated centralized localization technique for wireless sensor networks. In: Proc. IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PERCOMW), pp. 301–306 (2006)
- [2] Atmel Corporation. ATM128 Datasheet, Revised 2461-09/03 (2003)
- [3] Bahl, P., Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. In: Proc. IEEE Conf. on Computer Communications (INFOCOM), pp. 775–784 (2000)
- [4] Brunato, M., Battiti, R.: Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks* 47(6), 825–845 (2005)
- [5] Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine* 7(5), 28–34 (2000)
- [6] Capkun, S., Hubaux, J.-P.: Secure positioning of wireless devices with application to sensor networks. In: Proc. IEEE Conf. on Computer Communications (INFOCOM), pp. 1917–1928 (2005)
- [7] Dil, B., Dulman, S., Havinga, P.: Range-based localization in mobile sensor networks. In: Römer, K., Karl, H., Mattern, F. (eds.) *EWSN 2006*. LNCS, vol. 3868, pp. 164–179. Springer, Heidelberg (2006)
- [8] Doherty, L., Pister, K.S.J., Ghaoui, L.E.: Convex optimization methods for sensor node position estimation. In: Proc. IEEE Conf. on Computer Communications (INFOCOM), pp. 1655–1663 (2001)
- [9] Faria, D.B.: Modeling signal attenuation in IEEE 802.11 wireless LANs. vol. 1. Technical Report TR-KP06-0118, Stanford University (2005)
- [10] Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., Culler, D.: The nesC language: A holistic approach to networked embedded systems. In: Proc. ACM Conf. on Programming Language Design and Implementation (PLDI), pp. 1–11 (2003)
- [11] He, T., Huang, C., Blum, B.M., Stankovic, J.A., Abdelzaher, T.: Range-free localization schemes for large scale sensor networks. In: Proc. of the Int. Conf. on Mobile Computing and Networking (MOBICOM), pp. 81–95 (2003)
- [12] Hu, L., Evans, D.: Localization for mobile sensor networks. In: Proc. of the Int. Conf. on Mobile Computing and Networking (MOBICOM), pp. 45–57 (2004)
- [13] Kaemarungsi, K., Krishnamurthy, P.: Modeling of indoor positioning systems based on location fingerprinting. In: Proc. IEEE Conf. on Computer Communications (INFOCOM), pp. 1012–1022 (2004)

- [14] Kaemarungsi, K., Krishnamurthy, P.: Properties of indoor received signal strength for WLAN location fingerprinting. In: Proc. Int. Conf. on Mobile and Ubiquitous Systems (MOBIQUITOUS), pp. 14–23 (2004)
- [15] Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks. In: Proc. of the Int. Conf. on Mobile Computing and Networking (MOBICOM), pp. 243–254 (2000)
- [16] Krohn, A., Hazas, M., Beigl, M.: Removing systematic error in node localisation using scalable data fusion. In: Langendoen, K.G., Voigt, T. (eds.) EWSN 2007. LNCS, vol. 4373, pp. 341–356. Springer, Heidelberg (2007)
- [17] Kuo, S.-P., Tseng, Y.-C., Wu, F.-J., Lin, C.-Y.: A probabilistic signal-strength-based evaluation methodology for sensor network deployment. In: Proc. Int. Conf. on Advanced Information Networking and Applications (AINA), pp. 319–324 (2005)
- [18] Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: accurate and scalable simulation of entire TinyOS applications. In: Proc. Int. Conf. on Embedded Networked Sensor Systems (SENSYS), pp. 126–137 (2003)
- [19] Lorincz, K., Welsh, M.: MoteTrack: A robust, decentralized approach to RF-based location tracking. *Personal and Ubiquitous Computing* 11(6), 489–503 (2007)
- [20] Lymberopoulos, D., Lindsey, Q., Savvides, A.: An empirical characterization of radio signal strength variability in 3-D IEEE 802.15.4 networks using monopole antennas. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 326–341. Springer, Heidelberg (2006)
- [21] Lynch, C., Reilly, F.O.: Processor choice for wireless sensor networks. In: Proc. ACM Workshop on Real-World Wireless Sensor Networks (REALWSN), pp. 52–68 (2005)
- [22] Maróti, M., Völgyesi, P., Dóra, S., Kusý, B., Nádas, A., Lédeczi, Á., Balogh, G., Molnár, K.: Radio interferometric geolocation. In: Proc. Int. Conf. on Embedded Networked Sensor Systems (SENSYS), pp. 1–12 (2005)
- [23] Moore, D., Leonard, J.J., Rus, D., Teller, S.J.: Robust distributed network localization with noisy range measurements. In: Proc. Int. Conf. on Embedded Networked Sensor Systems (SENSYS), pp. 50–61 (2004)
- [24] Nagpal, R., Shrobe, H.E., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 333–348. Springer, Heidelberg (2003)
- [25] Nasipuri, A., Li, K.: A directionality based location discovery scheme for wireless sensor networks. In: Proc. ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA), pp. 105–111 (2002)
- [26] Niculescu, D., Badrinath, B.R.: Ad hoc positioning system (APS) using AOA. In: Proc. IEEE Conf. on Computer Communications (INFOCOM), pp. 1734–1743 (2003)
- [27] Niculescu, D., Nath, B.: DV based positioning in ad hoc networks. *Telecommunication Systems* 22, 267–280 (2003)
- [28] Ochi, H., Tagashira, S., Fujita, S.: A localization scheme for sensor networks based on wireless communication with anchor groups. In: Proc. Int. Conf. on Parallel and Distributed Systems (ICPADS), pp. 299–305 (2005)
- [29] Prasithsangaree, P., Krishnamurthi, P., Chrysanthis, P.K.: On indoor position location with wireless LANs. In: Proc. IEEE Int. Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 720–724 (2002)
- [30] Ramadurai, V., Sichitiu, M.L.: Localization in wireless sensor networks: A probabilistic approach. In: Proc. Int. Conf. on Wireless Networks (ICWN), pp. 275–281 (2003)

- [31] Rappaport, T.S., Rappaport, T.: *Wireless Communications: Principles and Practice*, 2nd edn. Prentice-Hall, Englewood Cliffs (2001)
- [32] Roos, T., Myllymaki, P., Tirri, H., Misikangas, P., Sievanen, J.: A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks* 9(3), 155–166 (2002)
- [33] Savvides, A., Han, C.-C., Strivastava, M.B.: Dynamic fine-grained localization in Ad-Hoc networks of sensors. In: *Proc. of the Int. Conf. on Mobile Computing and Networking (MOBICOM)*, pp. 166–179 (2001)
- [34] Savvides, A., Park, H., Srivastava, M.B.: The bits and flops of the n-hop multilateration primitive for node localization problems. In: *Proc. ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 112–121 (2002)
- [35] Shen, X., Wang, Z., Jiang, P., Lin, R., Sun, Y.: Connectivity and RSSI based localization scheme for wireless sensor networks. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC 2005*. LNCS, vol. 3645, pp. 578–587. Springer, Heidelberg (2005)
- [36] Sichertiu, M., Ramadurai, V.: Localization of wireless sensor networks with a mobile beacon. In: *Proc. IEEE Conf. on Mobile Ad-hoc and Sensor Systems (MASS)*, pp. 177–183 (2004)
- [37] Snyman, J.A.: *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer, Heidelberg (2005)
- [38] Stoleru, R., Vicaire, P., He, T., Stankovic, J.A.: StarDust: a flexible architecture for passive localization in wireless sensor networks. In: *Proc. Int. Conf. on Embedded Networked Sensor Systems (SENSYS)*, pp. 57–70 (2006)
- [39] Texas Instruments. MSP430C13x1 Datasheet (Revised September 04, 2004)
- [40] Wackerly, D., Mendenhall, W., Scheaffer, R.: *Mathematical Statistics with Applications*, 6th edn. Duxbury Advanced Series (2002)
- [41] Whitehouse, K., Karlof, C., Culler, D.: A practical evaluation of radio signal strength for ranging-based localization. In: *ACM Mobile Computing and Communications Review*, pp. 41–52 (2007)
- [42] Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J., Gao, X.: A wireless LAN-based indoor positioning technology. *IBM J. Res. Dev.* 48(5/6), 617–626 (2004)
- [43] Yedavalli, K., Krishnamachari, B., Ravula, S., Srinivasan, B.: Ecolocation: a sequence based technique for RF localization in wireless sensor networks. In: *Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN)*, p. 38 (2005)
- [44] Yu, Y., Govindan, R., Estrin, D.: Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, UCLA Computer Science Department (2001)