

# An Algorithm for Fast, Model-Free Tracking Indoors

Aiyou Chen<sup>a</sup>, Cristina Harko<sup>b</sup>, Diane Lambert<sup>c</sup>, and Phil Whiting<sup>a</sup>

<sup>a</sup>Alcatel-Lucent, Murray Hill, NJ 07974

<sup>b</sup>Smith College, Northampton, MA 01063

<sup>c</sup>Google, New York, NY 10011

*Tracking people and objects indoors from signal strength measurements has applications as diverse as security monitoring, self-guided museum tours, and personalization of communications services. Accurate dynamic tracking in real-time has been elusive, though, because signal propagation in buildings and the paths that people follow are complex. This paper proposes a simple algorithm for indoor tracking that requires neither a propagation model nor a motion model. It is also simple to compute, requiring only standard tools: Delaunay triangulation, linear interpolation, moving averaging, and local regression. Experiments with real data and simulations based on real data show that the algorithm is not only simple, it is effective.*

## I. Introduction

The widespread deployment of wireless networks in buildings provides an opportunity to estimate the location of people and objects in real-time for emergency response, protection of corporate assets, and personalized, location-based communications. Many approaches to static location and dynamic tracking from wireless signal strength measurements have been proposed, some of which are reviewed in Section II. Typically, tracking involves updating static location estimates using a motion model.

This paper presents a new, simple, model-free algorithm for dynamic tracking that performs well with limited training data. Briefly stated, the proposed algorithm has three pieces.

**Map Training Data** For each of the  $N_{AP}$  access points (APs), map the mean received signal strength (RSS) across the region of interest by linearly interpolating sample means over a triangular partition of the space, ignoring height. Also map the standard deviation of RSS by linearly interpolating a robust estimate of standard deviation over the same triangles.

**Estimate Location** Estimate signal strength at the current location by taking a moving average of the  $m$  most recent RSS measurements. Then find the  $k$ -nearest neighbors of this average in the  $N_{AP}$ -dimensional mean signal strength map, and take a weighted average of their corresponding  $k$  locations. With Gaussian weights, this estimate

approximates the Bayes estimate of location for a uniform prior on location, Gaussian RSS measurements, and squared error loss.

**Smooth the Track** Take a locally linear smoothing of the moving average location estimates. This increases the stability of the estimated path, and as a result cuts down wall crossings.

Mapping is done off-line; location estimation and tracking are done on-line whenever RSS measurements for the tracked object become available.

Tracking, then, requires only Delaunay triangulation of the region, linear interpolation over triangles,  $k$ -nearest neighbors, weighted averaging, moving averaging, and locally linear smoothing. There are well-known fast algorithms for all these procedures (*e.g.*, [11] for Delaunay triangulation, [3], [13] for  $k$ -nearest neighbors, and [4], [5] for locally linear smoothing). We provide rough guidelines for choosing the number of nearest neighbors and the window of the moving average. While others (*e.g.*, [8]) have suggested using a few equally weighted nearest neighbors, we show that being bold and taking many neighbors, as many as 100, and weighting them probabilistically can be advantageous for static location estimation.

The rest of this paper is organized as follows. Related work is reviewed in Section II. Static location estimation is described in Section III, and the dynamic tracking algorithm is described in Section IV. Experiments with real data and simulations based on real paths are discussed in Section V.

## II. Related Work

Relating received signal strength (RSS) to position is key to estimating the location of either a static or moving object. Several parametric propagation models that describe path loss across a region have been proposed, but none is universally accepted [21],[16] and many require restrictive assumptions. The most common models are based on inverse exponent laws:  $10 \log_{10} L(d) = \gamma \log_{10} d + L_{\text{ref}}$  where  $d$  is distance,  $L(d)$  actual path loss and  $L_{\text{ref}}$  is path loss at a reference distance. Values for  $\gamma$  range from about 1.5 to 5 depending on channel conditions, such as whether the target location is line-of-sight down a corridor or whether the transmitter and receiver are on adjacent floors; see [23],[22],[25] as well as [16] and the references therein. There can also be additional log-normal shadowing loss with a standard deviation ranging from 3 to 15 dB that affects signal strength readings [16]. All these sources of variation need to be modeled for effective location estimation indoors.

More ambitious propagation models do not rely solely on distance but also include the loss incurred from walls, floors, partitions and other obstacles that lie between the transmitter and the receiver. Path loss can then be determined by adding up attenuation estimates for the various obstacles, see [26] for example. In practice, however, these extended path loss models offer little improvement over exponent models. Including detailed information about the composition of the walls, floors, and other structures in the building can help, but this information is difficult to obtain. Moreover, even sophisticated ray tracing models that take detailed building maps and materials into account, such as the WISE tool [12], have typical errors of 6 dB, which is far too high for accurate tracking. These tools may however be used in planning the location and number of APs as well as the distribution of training positions.

Nonetheless, some location estimation algorithms have been based on path loss models; for example, [2]. In that paper, the parameters of the path loss model are estimated from a set of training data with known locations using a regression model that includes wall attenuation and a path loss exponent. Location estimates are then based on the fitted model. Using simulated data, [2] shows that this procedure can be effective when signal propagation follows the assumed model. No experiments with real data are reported in that paper, however.

Several authors [1, 17, 8] have avoided path loss models and instead drawn an empirical mean RSS map for an AP by interpolating the mean signal strengths obtained at a set of known training locations. RSS measurements at an unknown location are then compared to the signal strength maps to estimate the receiver's location. Because there is no path loss model, the locations of the APs need not be known, and indeed are not used in any way. An unknown target location is then estimated by comparing its vector of  $N_{\text{AP}}$  signal strengths to the  $N_{\text{AP}}$  mean RSS maps computed from the training data.

For example, Krishnan *et al.* [17] describe a system in which training measurements are constantly generated by transceivers, called emitters, at fixed locations. The emitters allow the current propagation environment, which depends on factors that change over time, such as building occupancy, to be re-estimated routinely. A mean signal map is estimated by fitting a nonparametric smooth function to the training data using the coordinates of their locations as the covariates in the fitted model. To estimate a location, they compare its  $J$  strongest RSS readings to the corresponding  $J$  smoothed RSS maps, ignoring the  $(N_{\text{AP}} - J)$  weaker signals. The location of the nearest neighbor using Euclidean distance is declared to be the estimated location.

Elnahrawy, Li, and Martin [8] build a purely empirical mean signal strength map by partitioning the region ("floor space", ignoring height) with Delaunay triangulation and then applying triangle-based linear interpolation to the sample mean RSS at a set of training positions. (We use the same procedure to compute an empirical mean RSS map in this paper.) As in [17], a location is estimated by finding its nearest neighbor in the set of empirical maps, but now the maps for all APs are used, not just the maps corresponding to the  $J$  strongest signals at the location, and the average of  $k = 1, 2, \text{ or } 3$  nearest neighbors, not just the one closest neighbor, is used to estimate location. This approach is related to our own except that we have a heuristic for choosing  $k$  (described in Section III) and instead of averaging with uniform weights we use Gaussian weights with means and standard deviations taken from maps of the means and standard deviations of signal strengths. This probabilistic weighting gives a Bayes estimate of location for squared error loss when the prior distribution is uniform across the map and RSS measurements have normal distributions.

Dynamically estimating the location of a moving

person or object, or *tracking*, has also been studied extensively. Perhaps the most well-known algorithm for tracking is the Kalman filter, which has been used for indoor tracking in [15] and [14], for example. Kalman filters are based on linear motion, but people may not take linear paths indoors because there are doors, walls, and corners. Evennou, Marx and Novakov [9] use particle filters to accommodate more realistic paths. Signal measurements are assumed to be Gaussian, and motion likelihoods have a factor of one or zero depending on whether a wall is crossed. At each step, the posterior likelihood weight for a particle is determined by multiplying the motion likelihood and signal likelihood. Computations for this method are burdensome and not suited to online tracking. Voronoi filters that restrict movement to a graph that by design prohibits wall crossing have also been used in [18]. This procedure can be more accurate, but it also requires more computation.

Finally, other aspects of signal propagation besides mean path loss, such as round-trip timing [19] and delay [10], have been used to estimate location. Such techniques are potentially far more accurate than those based on signal strength, but they require high bandwidth (ultra wideband) signals and specialized hardware. For example, ranging involves detecting the return pulse reflected from a passive device on the target and measuring the round trip delay. For line-of-sight locations, this method can achieve centimeter accuracy over distances of about 20m [19]. The algorithm described in this paper could be applied with timing measurements, but we have not done so to date.

### III. Static Location Estimation

Following [8], the sample mean  $\hat{\mu}(x, y) = \overline{R}(x, y)$  of RSS measurements from an AP are computed at each location  $(x, y)$  in a training set. To take into account changing variability over the region, we also compute the MAD estimate of standard deviation  $\hat{\sigma}(x, y) = 1.4826 \text{ median}|R - \text{median}(R)|$  at each training location. (The factor 1.4826 is chosen so that the MAD is an unbiased estimate of the standard deviation for Gaussian measurements. The MAD estimate is used because it is robust against outliers and non-normality.) Empirical mean and standard deviation maps for each AP are obtained by linearly interpolating the sample means and MAD estimates over Delaunay triangles [11].

An unknown location is estimated by finding the

$k$ -nearest neighbors in the empirical mean maps  $\{\hat{\mu}_i(x, y)\}$  for  $i = 1, \dots, N_{\text{AP}}$  to the observed vector of RSS  $R = (R_1, \dots, R_{N_{\text{AP}}})$ . If the  $j^{\text{th}}$  nearest neighbor  $(x_j, y_j)$  has estimated mean  $(\hat{\mu}_{j1}, \dots, \hat{\mu}_{jN_{\text{AP}}})$  and estimated standard deviation  $(\hat{\sigma}_{j1}, \dots, \hat{\sigma}_{jN_{\text{AP}}})$ , then it is assigned weight

$$w_j = \prod_{i=1}^{N_{\text{AP}}} \frac{1}{\sqrt{2\pi}\hat{\sigma}_{ji}} \exp\left(-\frac{(R_i - \hat{\mu}_{ji})^2}{2\hat{\sigma}_{ji}^2}\right).$$

The weight is the Gaussian probability density function evaluated at  $R$ . Given  $n$  measurements at the unknown location, the average measurement is used for finding the  $k$  nearest neighbors and each  $w_j$  is the product of the  $n$  Gaussian probability densities evaluated at the  $n$  measurements respectively. The estimated location  $(\hat{x}, \hat{y})$  is defined to be the weighted average

$$\left(\frac{\sum_{j=1}^k w_j x_j}{\sum_{j=1}^k w_j}, \frac{\sum_{j=1}^k w_j y_j}{\sum_{j=1}^k w_j}\right).$$

Conventionally, only a few nearest neighbors, such as 1, 2, or 3, and uniform weights have been used, but using much larger  $k$  with Gaussian weights can be better. For example, using Gaussian weights over the entire mean map gives an approximate posterior mean for the unknown location when the RSS measurements are normally distributed with a common variance and the prior is uniform [24]. The RSS measurements do not have a common variance, so the portion of the map used to estimate the location should be large enough to capture a large fraction of the posterior distribution but not so large as to include many locations with very different distributions of RSS.

Roughly, if the unknown location is estimated from the sample mean of  $n$  test measurements from the unknown location, then its estimated standard error is about  $\hat{\sigma}/\sqrt{n}$  where  $\hat{\sigma}$  is an average across locations and APs, and the uncertainty interval should contain at least one standard error centered at the location corresponding to the sample mean at the test location. Thus,  $k$  should be approximately the number of grid points that lie in a circle of radius  $a\hat{\sigma}/\sqrt{n}$ , where  $a$  is the error in distance caused by a 1-dB shift in RSS. When the grid points for a mean map are a distance  $d$  apart, this implies that

$$k \approx \frac{\pi a^2 \hat{\sigma}^2}{nd^2}. \quad (1)$$

In the experiments analyzed in Section 5, which have four APs,  $a$  is less than 1. We use

$a = 0.75$ . Furthermore, a threshold,  $w_{\text{thresh}} = (\sqrt{2\pi}\hat{\sigma})^{-nN_{\text{AP}}} \exp(-nN_{\text{AP}}/2)$ , is used to exclude neighbors with weights less than  $w_{\text{thresh}}$ .

#### IV. Model-free Online Tracking

Kalman filter and particle filter models used in [9], for example, assume a simple linear velocity model that is consistent with Brownian motion, while the Voronoi filter proposed in [9] adds constraints that prohibit impossible wall crossings. Linear motion models like these work well when the trajectory is simple, but not when people stop, turn, reverse direction, and change speeds abruptly, as they do indoors. On the other hand, most paths are at least smooth. Thus, we assume that the path itself is locally linear, with the linear coefficients changing over time.

It is important to note that the RSS measurements along a locally linear path need not have a locally linear mean because indoor propagation is complex. Thus, at every RSS reading, we update the path in two steps. We assume that the mean and standard deviation of RSS readings are roughly constant over short paths so we first estimate the current location from an average of RSS measurements over the recent past using the weighted  $k$ -nearest neighbor method described in Section III. We then assume that the motion path is locally linear over longer time spans and fit a smooth, locally linear model to a longer segment of the estimated locations in the path obtained so far. Note that the order of operations is important; the moving average controls for the variability in RSS measurements and the smoothing imposes regularity on the path. These two steps of are described in Sections IV.A and IV.B.

##### IV.A. Constant Location Estimation

If the target moves slowly relative to the rate at which the RSS measurements are taken, then the mean signal strength on its path should be roughly constant for short time spans. Thus, recent RSS measurements, not just the current RSS measurement alone, provide reasonable estimates of the current location. Since averaging RSS measurements reduces their variance, computing a location estimate for an average RSS over a short time span should give more stable location estimates without incurring much bias. Therefore, to estimate the location at a time  $t$  at which an RSS measurement is taken,

1. compute the average  $\bar{R}$  of the  $m$  most recent RSS measurements, and
2. find the weighted average of the locations of the  $k$  nearest neighbors of  $\bar{R}$  in the empirical mean signal strength map, as described in Section III

In words, the location estimate corresponds to a moving average of RSS measurements within a short window or span.

The bias introduced by using past RSS measurements in the moving average when the target is not stationary can be reduced by delayed tracking. That is, to estimate the location at time  $t$  consider not only  $m$  past and current measurements, but also the subsequent  $n$  measurements. The delay in tracking that results from using  $n$  future measurements may be acceptable if  $n$  is small and RSS readings are obtained frequently.

The optimal choice of  $m$  minimizes the mean squared error in estimated location, which depends on the variance  $\sigma^2$  in RSS measurements and the distance traveled over  $m$  measurements. If the moving average uses only the current and past measurements (no subsequent measurements) and the distance  $v$  between two adjacent RSS measurements is constant, then the average distance of the sites of the  $m$  measurements from the current location is  $mv/2$ . The mean squared error in the estimate of the current location is then

$$\text{MSE} = \text{variance} + \text{bias}^2 = a^2\sigma^2/m + m^2v^2/4, \quad (2)$$

where  $a$  is the average change in distance that results from a 1dB shift in RSS, as in Section 3. The above equation shows that 1) the variance of the error decreases linearly with  $1/m$ , and 2) the bias part increases linearly with  $m$ . If  $\sigma$  and  $v$  are roughly constant over the  $m$  measurements, then from equation (2) and by differentiating with respect to  $m$  we obtain that

$$\sqrt{\text{MSE}} \geq \left(\frac{27}{16}\right)^{1/6} (a^2\sigma^2v)^{1/3},$$

where the lower bound is achieved by taking

$$m = \left(\frac{\sqrt{2}a\sigma}{v}\right)^{2/3}. \quad (3)$$

For example, suppose  $a = 0.75$  m, the RSS measurements are received every 0.3 seconds,  $\sigma$  is about 3 dB along the recent path and the velocity is about 1 m/sec (i.e.  $v = 0.3$  m). Figure 1 shows how the minimal  $\sqrt{\text{MSE}}$  and the optimal  $m$  depend on the velocity.

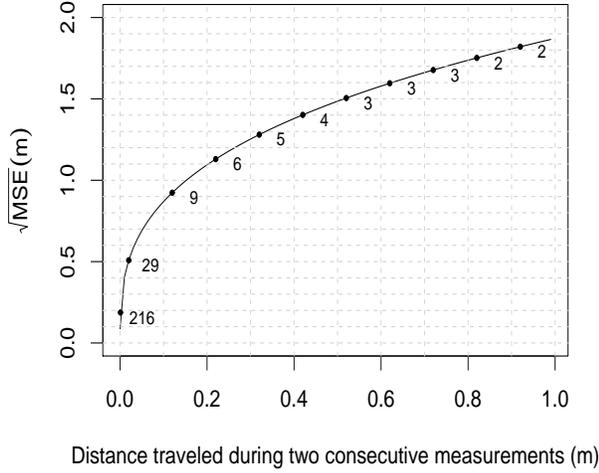


Figure 1:  $\sqrt{\text{MSE}}$  as a function of distance traveled between two consecutive measurements when  $a = 0.75$  m,  $\sigma=3$  dB. The solid line shows a cube root law fit to the data; the numbers beside the solid points show the optimal  $m$  for that velocity.

If at least one of the  $m$  RSS measurements is taken ahead of the current position, then the bias is reduced and the total number of measurements in the moving average can be larger than equation (3) implies.

#### IV.B. Locally Linear Motion

Computing a location estimate from a moving average of RSS measurements amounts to assuming that the target does not move over short time intervals. These estimates can be used to track a moving target, but they are likely to be noisy because they are based on only a few measurements. (Otherwise, the assumption that the target is not moving might be false.) To make use of older past information, we need to assume a motion model. Here we assume that the path is locally linear and use the *LOESS* smoother [4], [5].

The *LOESS* regression model is

$$x_i = g(t_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where  $g$  is an unknown, locally linear regression curve and the  $\epsilon_i$  are independent, constant variance Gaussian errors. The assumption of constant variance is approximately true over time spans that are not too large relative to the rate at which the target is moving. The *LOESS* model is fit by iteratively re-weighted least squares over a span of nearest neighbors (here we use 40), where the weight on  $(t_i, x_i)$  depends on

the distance of the fitted value  $(t_i, \hat{x}_i)$  from the observed. (See [4] for a detailed explanation of *LOESS* smoothing.) Because only a short delay in prediction is desirable in tracking, there are typically many more times  $t_i$  to the left of  $t$  than to the right of  $t$  in the span of  $t$ . A fast algorithm for fitting *LOESS* can be found in [5] and is available in R [27], an open source system for statistical computations and graphics. Besides not requiring an exact motion model, *LOESS* is robust to outliers and is not adversely affected by long-tailed, non-Gaussian distributions.

Finally, in the experiments that we have analyzed, there are often isolated measurements at the limit of detection (about  $-90$  dB in the data reported in Section V). That is, the previous and subsequent measurements are not close to the limit of detection, but the current measurement is. There may be several explanations for these outliers, such as passing obstacles, but in any case we correct for spurious “missing” readings by replacing each measurement at the limit of detection by the average of the previous two measurements, and then proceeding as if the replacement were the true reading. Replacement by the mean is not optimal, as replacement with a random draw from a normal distribution would be better, but it is easy.

#### IV.C. Dynamic Tracking Algorithm

For a nearest neighbors span  $k$ , moving averaging span  $m$ , (including  $n$  forward RSS measurements), *LOESS* spans of  $s$  and  $r$  past and future estimated positions, tracking has the following steps.

1. For a set of training data with  $N_{\text{AP}}$  APs,
  - (a) For each AP, create an empirical propagation model by triangle based linear interpolation of sample means of RSS.
  - (b) For each AP, create an empirical map of the standard deviation of RSS by triangle-based linear interpolation of MAD estimates.
2. For  $t = 1, 2, \dots$ , given an  $N_{\text{AP}}$ -dimensional vector of RSS measurements  $R_t$ ,
  - (a) If  $R_t$  is at the limit of detection, then replace it by the average of the two previous measurements.
  - (b) Compute the moving average of  $R_{t-m+1}, \dots, \dots, R_{t+n}$ , say  $ma(R)_t$ .

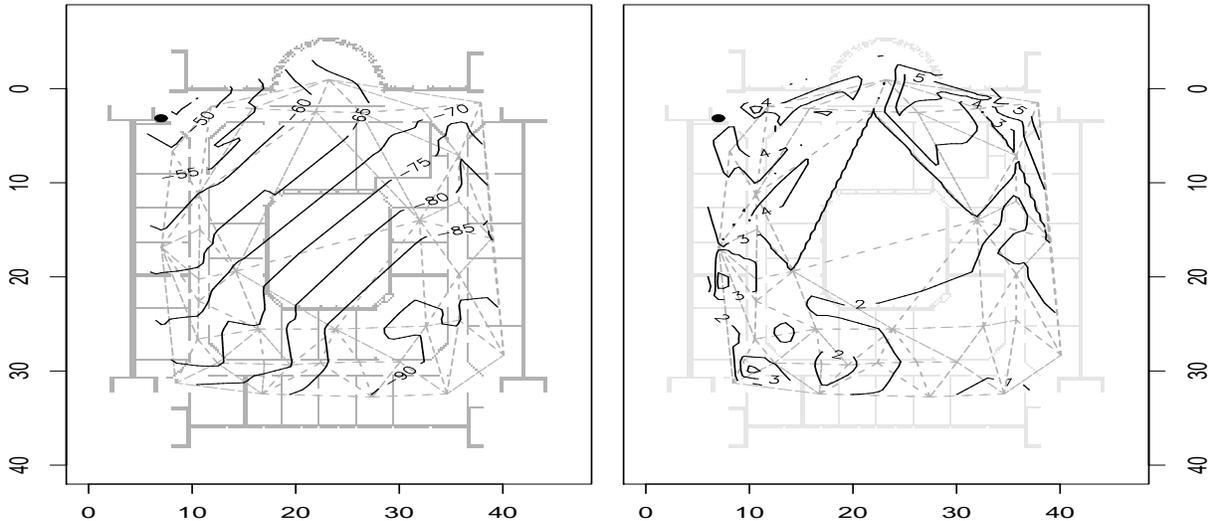


Figure 2: Contours of the estimated mean signal strength (left) and estimated standard deviation (right) of signal strength for one AP for the data discussed in Section 5. The AP, shown by a filled circle, is located in the northwest stairwell; the  $40\text{ m} \times 40\text{ m}$  building map is shown by solid lines in the background. The dotted lines show the Delaunay triangulation based on 30 training positions.

- (c) Estimate the location  $(\hat{x}_t, \hat{y}_t)$  of  $ma(R)_t$  by finding the Gaussian-weighted average of the locations of its  $k$  nearest neighbors in the empirical mean signal strength map.
- (d) Fit a LOESS model to  $(\hat{x}_{t-s+1}, \dots, \hat{x}_{t+r})$  to estimate  $\hat{X}_t$ .
- (e) Fit a LOESS model to  $(\hat{y}_{t-s+1}, \dots, \hat{y}_{t+r})$  to produce the estimate  $\hat{Y}_t$ .
- (f) Return  $(\hat{X}_t, \hat{Y}_t)$  as the estimated position at time  $t$ .

The first major step constitutes training and can be computed off-line; the second major step defines tracking and can be computed on-line. The final positions  $\{(\hat{X}_t, \hat{Y}_t)\}$  are smooth with respect to  $t$  and serve as the estimated moving path. It would be preferable, but more complicated, to smooth the bivariate  $(x_t, y_t)$  in one step, taking into account the correlation in the two dimensions. We believe that this would further reduce impossible wall-crossings.

## V. Experimental Results

### V.A. Signal Strength Maps

Our experiments use WIFI data provided by Frédéric Evnou of France Telecom Division R&D and re-

ported in [9]. About 100 RSS measurements from four APs were obtained at each of 88 positions in a space of about  $40\text{ m} \times 40\text{ m}$  that includes corridors and offices around a walled central courtyard with stairwells in the corners (see Figure 2). An AP was located in each of the four corners. We use RSS measurements from 30 fixed positions that are roughly uniformly spaced over the region of interest (ignoring the courtyard, for example) for training. As in [8], we found that using more training positions, which is costly, has little effect on the quality of location estimation.

The estimated mean and standard deviation (MAD) signal strength maps based on all the measurements for the 30 training positions for the northwest AP and an interpolation grid size of  $0.25\text{ m}$  are shown in Figure 2. (The mean and standard deviation maps for the other access points are similar, with obvious rotations that depend on the location of the AP.) As would be expected, the mean signal strength diminishes from about  $-50\text{ dB}$  to about  $-90\text{ dB}$  (apparently, the limit of detection) with distance from the access point. Surprisingly, the mean signal strength falls off with the square root of distance in both the corridors and offices, with an additional constant loss for locations in offices. This model does not fit all APs “behind the

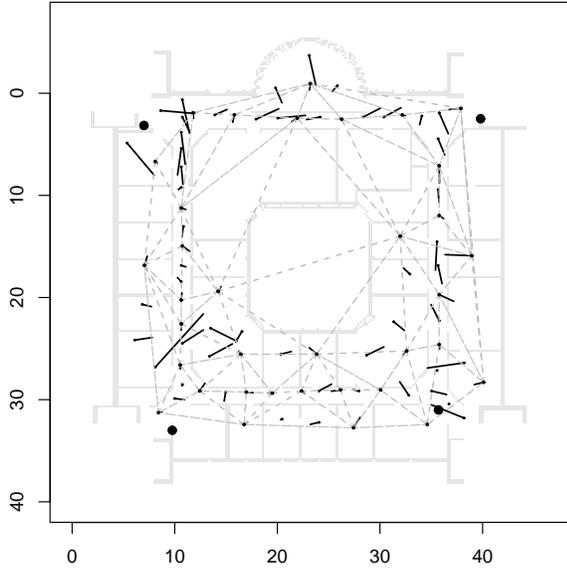


Figure 3: Errors in location estimates for an experiment with the mean and standard deviation maps shown in Figure 2. The maps were based on 100 measurements at each of the 30 training positions; locations were estimated from 10 randomly chosen RSS measurements from each location. The circles at the end of the line show the true location; the other end of the line denotes the estimated location.

courtyard wall”, though, so we do not use it here.

The most prominent features of the standard deviation maps are the hills in each corner. For the access point in the NW corner, the standard deviation is highest in the northeast corner and lowest in the southwest corner, probably because the signal there is weak and bounded below by -90. Over the entire space, the estimated standard deviation ranges from 1 to 5 dB, with a mean and median standard deviation over the region of about 2.8 dB and 3.0 dB respectively.

## V.B. Location Experiments

First, the mean and standard deviation maps in Section V.A are used to estimate all 88 locations using a random subset of 10 RSS measurements from each AP for each location. Thus, the standard error at any point in the empirical mean signal strength map is about  $2.8/\sqrt{10} = 0.95$ . Because the grid size is 0.25 m, equation (1) gives  $k \approx 22$ . Thus, we estimate location with a Gaussian weighted average of the 25 nearest neighbors.

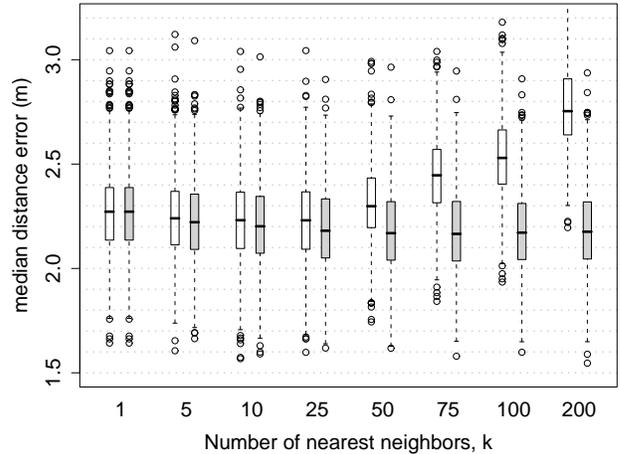


Figure 4: Simulated distribution of median location error over the region with a training set of 100 RSS measurements from each of 30 locations and each of the 88 locations estimated from one randomly chosen RSS measurement from the location. The white boxplots are for uniform weights; the gray are for Gaussian weights with means and standard deviations taken from the empirical mean and standard deviation maps. Each pair of boxplots represents a different number  $k$  of nearest neighbors.

The errors for one trial are shown in Figure 3. The errors are generally small, except outside the convex hull of the training positions. Moreover, the errors are generally larger along the north-south corridors than they are along the east-west corridors. Also, the errors tend to be larger in the corners, *i.e.*, very close to the access points. The 0.1, 0.5 and 0.9 quantiles of the errors are 0.35, 0.96 and 2.86 meters respectively. Similar maps are obtained for other trials under the same conditions (10 RSS measurements from each location to be estimated).

The effect of using Gaussian weights rather than uniform weights can be seen in Figure 4. There, static location estimates for the 88 locations are based on one randomly selected RSS measurement from each location, using the training maps shown in Figure 3. The optimal number of nearest neighbors  $k$  from equation (1) is now 220, or roughly 200. In Figure 4, the boxplots are based on computing the median error over the map for 1000 replications of this experiment. The box shows the 0.25, 0.5 and 0.75 quantiles, the open circles show outliers, and the dashed lines show the tails of the data without the outliers. As

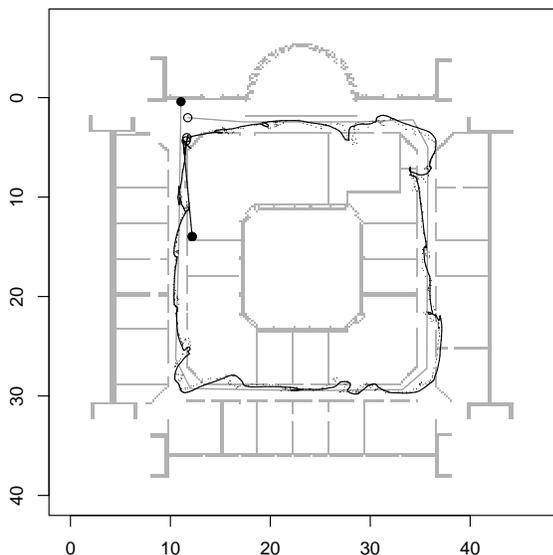


Figure 5: The true path (light gray line along corridors), location estimates based on the moving average (dotted line), and smoothed, tracking estimates (solid line). The filled circles denote the start of the path; the open circles denote the end of the path. (Axes are the same as in Figure 2.)

would be expected, Gaussian weights are not much better than uniform weights if  $k$  is small, but their importance increases with  $k$ . There is about a 5% reduction in the average median error for  $k = 200$  and Gaussian weights. Note that the difference between using  $k = 1$  and  $k = 5$  is negligible, with or without weights.

## VI. Tracking Experiments

We will analyze the performance of our tracking algorithm for the path data (RSS measurements and locations) in [9] using the training maps from Section V.B, but first we must evaluate whether the training maps are appropriate for the tracking data. The tracking data were collected days after the training data, and the signal strength map might have changed in the interim. In fact, comparing the mean observed RSS in the tracking experiments with the mean observed RSS in the training data shows that adjusting the RSS measurements for APs 1,2,3, and 4 by 1 dB, 1.5 dB, 1.5 dB

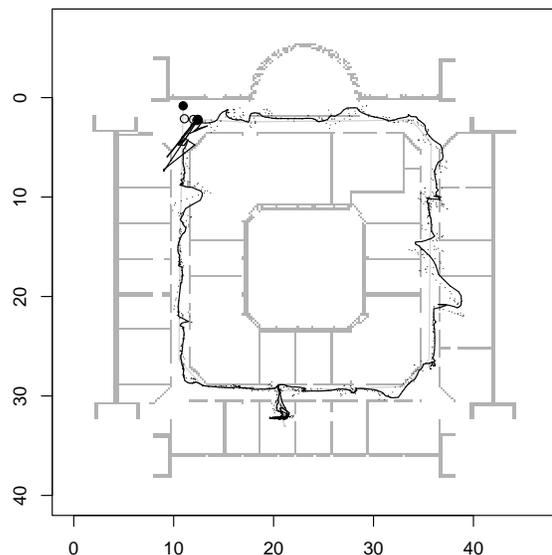


Figure 6: Estimated path using a moving average with a span of 8 (dotted) followed by loess with a span of 40 (solid line); the true path is the light gray line shadowed by the estimated path which goes into one room off the corridor. (Axes are the same as in Figure 2.)

and 1 dB respectively corrects for the average drift in the mean. More generally, drift in calibration could be evaluated using data routinely collected from known locations, as proposed in [17].

Figure 5 shows the results of applying our tracking algorithm to the path data reported in [9] with  $k = 100$  nearest neighbors and a moving average of  $m = 8$  past and current measurements. The 0.1, 0.5, and 0.9 error quantiles over the map are 0.3, 1.3 and 2.5 m respectively for the moving average estimates; the values after smoothing are 0.5, 1.4 and 2.6 m. Thus, smoothing in this case does not reduce errors, but the smooth path has fewer wall crossings and unrealistic jumps. In comparison, [9] reports a median error of 2.0 m for the Kalman filter and 1.6 m for the Voronoi filter that avoids wall crossing. That is, our tracking method reduces the median error of the Kalman filter by about 30% and the median error of the much more complicated Voronoi filter by about 14% in this example.

Finally, we construct simulated paths by fixing the locations along an actual path and simulating RSS

measurements by sampling from normal distributions with means and standard deviations determined by the empirical RSS maps. The same training data used in Figure 5 were used here. Estimates for a simulated path are shown in Figure 6. Here the 0.1, 0.5 and 0.9 quantiles of location error are 0.27, 0.9 and 1.8 m respectively after the moving average step. After smoothing the path, the 0.1, 0.5 and 0.9 quantile errors are 0.4, 0.8 and 1.8 m, and the paths are again more realistic.

Table 1 shows the effect of using  $k$  nearest neighbors and a moving average over  $m - 1$  past RSS measurements, the current measurement, and either no or one future measurement. The table reports median error over the path, averaged over 100 simulated paths. As would be expected, using one future ( $t = n + 1$ ) RSS measurement to estimate location at time  $n$  reduces the median error along the path noticeably – by about 10%. Moreover, with one future RSS measurement, the optimal choices are a moving average window of  $m = 8$  and  $k = 25$  nearest neighbors. Equation (1) would suggest  $k = \pi 0.75^2 \hat{\sigma}^2 / (8/16) \approx 28$  in this case, which is quite close to the simulated optimum for both the no look-ahead and one look-ahead cases in the table.

## VII. Conclusions

We have shown that low complexity tracking algorithms that use only propagation loss measurements and do not require a motion model can be used for tracking people and objects indoors. In experiments, the proposed algorithm does better than previous algorithms: our MSE is about 14% smaller than the error of the Voronoi filter, and about 30% smaller than the error of the Kalman filter. Root mean squared errors are on the order of 1 m. The algorithm is built from simple components, Delaunay triangulation, linear interpolation, moving averaging, and local linear regression, which are fast to compute. The performance of these algorithms can be enhanced by taking measurements more frequently or using more APs, without requiring much more computation. This approach does need good training data, though. Training data should be recent, or else it must be adjusted (calibrated) to accommodate changes in propagation resulting from changes in the building, the objects within it, or the state of windows and doors, for example.

For static or constant location estimation, we have proposed averaging with Gaussian weights. This

weighted average approximates the posterior local mean of location if RSS measurements are Gaussian and the prior is uniform over the space. The posterior local mean is the optimal estimate for Gaussian errors. We have also proposed a method for choosing the span  $k$  for  $k$ -nearest neighbors.

For tracking, we have shown that greater accuracy is obtained by *first* averaging over the RSS values and *then* estimating the location of the average RSS instead of first estimating the locations and then averaging over these. The choice of the moving average span has to balance bias and variance: generally shorter spans are less biased but also less precise. The best trade-off can be determined by using an estimate of the tracked object’s speed, e.g. integration of the path length divided by time. When speed is estimated accurately, our results show that the error using the optimal  $m$  increases only as the cube root of speed. Finally, we note that using no motion model, but rather taking local moving averages and locally linear regressions, is preferable to using either a simple approximate motion model, as Kalman filtering does, or a complicated motion model, as particle filters do.

## Acknowledgments

We thank Frédéric Evennou, Dominique Barthel, and Francois Marx for many helpful and delightful discussions about tracking and for providing the data used in this paper. We also thank Eiman Elnahrawy, David James and Thomas Bengtsson for helpful discussions.

## References

- [1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE Infocom*, 2000.
- [2] Y. Chen and H. Kobayashi. Signal strength based indoor geolocation. *Proceedings IEEE Conference on Communications*, New York, 2002.
- [3] K. L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete Comput. Geom.*, 22, 63–93, 1999.
- [4] W. S. Cleveland and S. J. Devlin. Locally-weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 12, 596–610, 1988.

Table 1: Average (over 100 replications) of the median error using simulated RSS measurements along the true path in Figure 5 as a function of the number of nearest neighbors  $k$ , the number  $m$  of past and current RSS measurements in the moving average, and whether 0 or 1 future RSS measurements were used in the moving average.

$m$	$k$							
	1	5	10	25	50	100	200	500
	No look ahead							
1	2.02	1.99	1.97	1.94	1.91	1.87	1.83	1.82
2	1.45	1.43	1.41	1.39	1.36	1.35	1.34	1.33
4	1.06	1.04	1.03	1.01	1.00	0.99	0.99	0.99
6	0.93	0.91	0.90	0.89	0.88	0.88	0.88	0.88
8	0.91	0.89	0.88	0.87	0.87	0.87	0.87	0.86
10	0.95	0.93	0.92	0.92	0.91	0.91	0.91	0.91
12	1.04	1.02	1.01	1.00	1.00	1.00	1.00	1.00
	One step look ahead							
1	2.02	1.99	1.97	1.94	1.91	1.87	1.83	1.82
2	1.46	1.43	1.41	1.39	1.36	1.35	1.34	1.33
4	1.05	1.02	1.01	0.99	0.98	0.97	0.97	0.97
6	0.89	0.87	0.86	0.85	0.84	0.83	0.83	0.83
8	0.85	0.82	0.81	0.80	0.80	0.79	0.79	0.79
10	0.86	0.84	0.83	0.82	0.82	0.82	0.82	0.82
12	0.92	0.90	0.89	0.89	0.88	0.88	0.88	0.88

- [5] W. S. Cleveland and E. H. Grosse. Computational Methods for Local Fitting. *Statistics and Computing*, 12, 47–62, 1991.
- [6] N. Dyn and S. Rippa. Data-dependent triangulations for scattered data interpolation and finite element approximation. *Applied Numeric Mathematics*, 12, 89–105, 1993.
- [7] Ekahau, Inc. The Ekahau Position Engine 2.1, July 2003, whitepaper at [www.ekahau.com](http://www.ekahau.com).
- [8] E. Elnahrawy, X. Li and Martin, R. The limits of localization using signal strength: a comparative study. In *Proceedings of IEEE Infocom*, 2005.
- [9] F. Evennou, F. Marx and E. Novakov. Map-aided indoor mobile positioning system using particle filter. *IEEE Wireless Communications and Networking Conference*, New Orleans, LA, March 2005.
- [10] F. Evennou, F. Marx, S Nacivet. Sensor fusion for UWB and Wifi indoor positioning systems. *The 8th Annual International Symposium on Advanced Radio Technologies*, Boulder, Colorado, March 2005.
- [11] S.J. Fortune, Voronoi diagrams and Delaunay triangulations, *CRC Handbook of Discrete and Computational Geometry*, 377–388, 1997.
- [12] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright. WISE design of indoor wireless systems: practical computation and optimization. *IEEE Computational Science and Engineering*, 2, 58–68.
- [13] S. J. Fortune and J. Hopcroft, A note on Rabin’s nearest neighbor algorithm. *Information Processing Letters*, 8, pp. 20–23, 1979.
- [14] D. Fox, J. H Hightower, H. Kauz, L. Liao and D. Patterson. Bayesian techniques for location estimation. *IEEE Pervasive Computing*, vol. 2, number 3, 2003.
- [15] I. Guvenc, C. Abdallah, R. Jordan, and R. Dedeoglu. Enhancements to RSS based indoor tracking systems using Kalman filters. *Internat-*

- tional Signal Processing Conference and Global Signal Processing Expo*, Dallas, TX, 2003.
- [16] H. Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81, 943–968, 1993.
- [17] P. Krishnan, A.S. Krishnakumar, W. Ju, C. Mallows and S. Ganu. A system for LEASE: System location estimation assisted by stationary emitters for indoor RF wireless networks. In *Proceedings of IEEE Infocom*, 2004.
- [18] L. Liao, D. Fox, J. Hightower, H. Krautz and D. Schulz. Voronoi tracking: location estimation using sparse and noisy sensor data. *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE/RSJ, 2003.
- [19] Z. N. Low, J. H. Cheong, C. L. Law, W. T. Ng, and Y. J. Lee. Pulse detection algorithm for line-of-sight (LOS) UWB ranging applications. *IEEE Antennas and Propagation Letters*, 4,63–67, 2004.
- [20] D. Madigan, E. Elnahrawy, R. P. Martin, W. Ju, P. Krishnan and A.S. Krishnakumar. Bayesian indoor positioning systems. In *Proceedings of IEEE Infocom*, 2005.
- [21] D. Molkdar. Review on radio propagation into and within buildings. *IEE Proceedings-H*, 138, 61–73, 1991.
- [22] T. Rappaport. Indoor radio communications for factories of the future. *IEEE Communications Magazine*, 15–24, May 1989.
- [23] T. Rappaport and C. McGillem. UHF fading in factories. *IEEE Journal Selected Area of Communications*, 7, 40–48, 1989.
- [24] T. Roos, P. Myllymaki, and H. Tirri. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, Jan-March 2002.
- [25] A. Saleh and R. Valenzuela. A statistical model for indoor multipath propagation. *IEEE Journal Selected Area of Communications*, 5, 128–137, 1987.
- [26] S. Seidel and T. Rappaport. 900 Mhz path loss and prediction techniques for in building communications system designs. *Proceedings 41st Vehicular Technology Conference*, St. Louis Missouri, VTC 91, 1991.
- [27] <http://cran.r-project.org>