Localization of Mobile Users Using Trajectory Matching

HyungJune Lee, Martin Wicke, Branislav Kusy, and Leonidas Guibas
Stanford University, Stanford, CA, USA
{abbado,wicke,kusy}@stanford.edu, guibas@cs.stanford.edu

ABSTRACT

We present an algorithm enabling localization of moving wireless devices in an indoor setting. The method uses only RF signal strength and can be implemented without specialized hardware. The mobility of the users is modeled by learning a function mapping a short history of signal strength values to a 2D position. We use radial basis function (RBF) fitting to learn a reliable estimate of a mobile node's position given its past signal strength measurements.

Even though we deal with extremely noisy measurements in a cluttered indoor setting, nodes are not required to be stationary during measurement or learning. We evaluate our algorithm in a real indoor setting using MicaZ motes, achieving an average localization accuracy of 1.3 m. In our experiments, using historical data improves the localization accuracy by almost a factor of two compared to using only the most current measurements.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Algorithms, Experimentation

Keywords

Localization, Mobility, RSSI, Sensor Network

1. INTRODUCTION

With the advent of ubiquitous wireless networks, supporting mobility of users has become a key topic in network research. Localizing users moving through a network is a fundamental problem in this area. Accurate estimates of users' locations enable more efficient routing strategies in the presence of mobile nodes. Location-dependent network services, with application examples ranging from building automation to targeted advertising or augmented reality, first and foremost require reliable localization techniques.

The field of localization has therefore been studied in a wide variety of research communities. Triangulation methods [12, 14] are among the most common. These methods estimate positions from a number of distance or angle measurements to beacon nodes, utilizing models that describe how acoustic or radio signals propagate in space (such as the inverse-square law). Even though relatively accurate models [20] for open areas exist, they are of limited use indoors, where model inaccuracies due to reflections and signal fading can lead to significant position errors. Since no accurate and efficient models of indoor signal propagation are available, a number of methods pre-compute a signal-strength map of the coverage area [1, 10]. These methods estimate position of a node by comparing the signal-strength signature of beacon nodes to the map. However, both acoustic and radio signals indoors tend to be highly variable over time, especially so for mobile users, resulting in reduced accuracy of these algorithms.

In this paper, we focus on the problem of using received signal strength indicator (RSSI) measurements to localize mobile users in indoor environments. As the structure of the environment is unknown, no good transmission model is available. However, in an indoor setting, the user's mobility is restricted by the environment (we cannot go through walls), and we can assume that not all possible movements within space are actually realized. Rather, the users move along a limited set of typical trajectories, suggesting that we can learn the structure of the space of possible movements from repeated observation. We can use this inferred knowledge to locate users, and extrapolate our observations to unknown trajectories.

One of the main problems when using signal strength data for localization is the large variance in these measurements. Our experimental data shows that the variance due to reflections is particularly severe when either transmitter or receiver are moving, even at low speeds. Systems that use RSSI readings for localization therefore use averages or require the nodes to be stationary during the measurement [1, 9, 10]. We propose to use a function fitting and interpolation scheme to learn a position function in the high-dimensional space of signal strength measurements. We not only use the current set of RSSI values for reachable nodes, but also a number of past samples, thereby matching a trajectory in signal strength space to a position.

The learning process handles noisy input data gracefully by computing a smooth approximation to the input samples. After a learning phase which requires position ground truth, queries to the localization subsystem

reduce to a simple function evaluation. We evaluate several versions of this algorithm, exploring different tradeoffs between the memory requirements, communication overhead, and localization accuracy. Our results clearly indicate benefits of user movement history, almost cutting the position error in half compared to using current measurements only. We also compare the accuracy of our algorithm for three different link quality estimators: received signal strength indicator (RSSI), link quality indicator (LQI), and packet reception ratio (PRR). Even though RSSI measurements are clearly superior, other link quality measures perform reasonably well.

The rest of this paper is organized as follows: after giving an overview of related work in Sec. 2, we formally define the problem that we are solving in Sec. 3. Sec. 4 describes our solution. We evaluate the performance of our algorithm in Sec. 5 before concluding in Sec. 6.

2. RELATED WORK

The problem of localization and tracking has been extensively studied in robotics literature. Several variants of the localization or tracking problem with or without landmarks, with or without access to a map, and using various forms of sensor input have been explored. A good overview of the field is given in the introduction of [19]. While most of these approaches use visual or range finding sensors, some are applicable to the problem of locating users within a wireless network.

Spatial relations between nodes can be found using a number of techniques, such as time of flight [14], angle of arrival of a signal [13], walking GPS [17], ultrawideband [15], or Doppler shift ranging [8]. The localization problem is then solved using optimization techniques, for example [2, 4]. However, these techniques often require sophisticated hardware. In contrast, we assume that nodes are equipped with very basic hardware that only allows us to measure some kind of radio link quality estimate.

Approximate point in triangle (APIT) [6] and DV-hop [11] are two representatives of range free algorithms. These algorithms estimate locations using simple radio connectivity information from multiple beacon nodes located in the proximity of the unknown node. APIT localizes nodes at an intersection of triangular regions defined around beacon nodes, whereas DV-hop triangulates the location using hop counts that estimate distances between nodes. These approaches, however, were designed for static networks and thus do not support mobility well. Moreover, their accuracy is often limited by requiring dense deployment of sensor nodes.

RSSI pattern matching algorithms (RADAR [1], Mote-Track [10]) are closely related to our approach. Users are localized inside a building using RSSI measurements from fixed beacon nodes. As no reliable radio propagation model is available, these techniques do not estimate Euclidean distances from the measured RSSI values. Instead, they learn the properties of radio signals for a particular position. The location of the unknown node is found by matching the current RSSI signature

of nearby beacons to this empirical model. This process was shown capable of eliminating multipath and shadowing effects, achieving meter-level localization accuracy.

Ladd *et al.* [9] proposed an indoor localization algorithm achieving an average error of 1.5 m. Similarly to our approach they sense RF signal strengths using standard hardware (Ethernet cards). However, their approach suppresses mobility related variation of radio signals, rather than utilizing it to its benefit: the training algorithm requires a person to stand still for a few moments or requires filtering to calibrate mobile users.

The LOCADIO system [7] explicitly models moving vs. stationary users in a probabilistic framework. Our method goes one step further: we are interested in mobile users and therefore explicitly use the history of the users movement in the localization algorithm.

3. PROBLEM FORMULATION

This paper describes an approach to the indoor localization problem using signal strength measurements as input. Given information about the connection quality to close-by infrastructure nodes, we infer the position of a mobile device. In our experiments, connection quality is measured by RSSI values that are provided by the wireless hardware. As some radio chips do not support RSSI measurements (e. g., the Nordic NRF903 used in [16]), it is important to note that our approach does not depend on the presence of this specific type of measurement. Packet reception ratios, which can be extracted in all wireless networks, can be used instead (see Sec. 5).

We assume that a mobile user moves through a network spanned by a set $\mathcal{N}, |\mathcal{N}| = N$ cooperating infrastructure nodes. The mobile node regularly broadcasts radio packets allowing for RSSI measurements. We assume that these packets are dedicated beacon packets, however, we believe that aggressive suppression of beacon packets will enable virtually zero-cost localization and tracking in practical applications, if regular network traffic to and from the mobile node is present.

Whenever such a beacon packet is received by the infrastructure nodes, we extract RSSI information, yielding values $r_i(t)$ that measure the signal strength of the packet sent at time t, received by the infrastructure node with ID i. For localization, we will not only consider the most recent RSSI values, but utilize historical data as well. For our computations, we assemble these values into a vector

$$\mathbf{r}(t) = \left[\begin{bmatrix} r_1(t - 0\Delta t) \\ \vdots \\ r_1(t - k\Delta t) \end{bmatrix}^T \cdots \begin{bmatrix} r_N(t - 0\Delta t) \\ \vdots \\ r_N(t - k\Delta t) \end{bmatrix}^T \right]. \quad (1)$$

$\mathbf{r}(t)$ defines a point in the space $\mathcal{X}_{k,N}$ of sampled trajectories through RSSI space, in which most of our computations will take place. The localization problem can then be described as finding a function $\mathbf{L} : \mathcal{X}_{k,N} \to \mathbb{R}^2$ which maps a trajectory from $\mathcal{X}_{k,N}$ to its end position in world space.

4. LOCALIZATION ALGORITHM

Our approach learns function $\mathbf{L}$ defined in the previous section from examples that we obtain in a training phase. During the training phase, ground truth locations of the mobile user are required, however, locations of infrastructure nodes are not needed. We use radial basis function fitting to compute a function that approximates the (noisy) input examples and is able to extrapolate from those examples in a smooth fashion in the localization phase.

4.1 Training Phase

In a training phase, a mobile node explores the physical space that is covered by the network. While moving around in space, both world space position $\mathbf{p}(t) = [p^x(t), p^y(t)]^T$ and signal strength measurements $r_i(t)$ are recorded at discrete times t_j. We then assemble the sampled trajectories $\mathbf{r}(t_j)$, and form N_s pairs $(\mathbf{r}(t_j), \mathbf{p}(t_j))$ that we use as training data. These pairs are samples of the function $\mathbf{L}$ that we are trying to find, and impose constraints of the form

$$\mathbf{L}(\mathbf{r}(t_j)) = \mathbf{p}(t_j). \tag{2}$$

We use RBF fitting to compute a smooth function that approximates these constraints.

4.1.1 RBF Fitting

Due to space constraints, we will only briefly review RBF fitting here, and refer the reader to [3] for an in-depth treatment of the subject.

We express the function $\mathbf{L} = [L^x, L^y]^T$ as a weighted sum of kernel functions ϕ plus a polynomial P^α in $\mathbf{r}$

$$L^\alpha(\mathbf{r}) = \sum_{i=1}^{N_c} w_i^\alpha \phi(\|\mathbf{r} - \mathbf{c}_i\|) + P^\alpha(\mathbf{r}), \tag{3}$$

where $\alpha = \{x, y\}$. The number, N_c, and the placement of the kernel *centers* $\mathbf{c}_i$ are free parameters. In our implementation, we use a linear polynomial $P^\alpha(\mathbf{r}) = \langle \mathbf{a}^\alpha | \mathbf{r} \rangle + b^\alpha$. The kernel function ϕ determines properties of the solution, such as its smoothness. We use the common multiquadric $\phi(d) = \sqrt{1 + d^2/\sigma^2}$, with $\sigma = 1$.

To compute weights w_i^α, we minimize the quadratic error at the constraints (2) by solving two linear systems of equations for the variables $[\mathbf{W}^\alpha, \mathbf{a}^\alpha, b^\alpha]^T$

$$\begin{pmatrix} \phi_{0,0} & \cdots & \phi_{0,N_c} & \mathbf{r}(t_0)^T & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \phi_{N_s,0} & \cdots & \phi_{N_s,N_c} & \mathbf{r}(t_{N_s})^T & 1 \\ 1 & \cdots & 1 & & \mathbf{0} \\ \mathbf{c}_1 & \cdots & \mathbf{c}_{N_c} & & \end{pmatrix} \begin{bmatrix} \mathbf{W}^\alpha \\ \mathbf{a}^\alpha \\ b^\alpha \end{bmatrix} = \begin{bmatrix} p^\alpha(t_1) \\ \vdots \\ p^\alpha(t_{N_s}) \\ \mathbf{0} \end{bmatrix}, \tag{4}$$

in a least-squares sense. Here, $\phi_{i,j} = \phi(\|r_i - c_j\|)$ and $\mathbf{W}^\alpha = [w_1^\alpha, \ldots, w_{N_c}^\alpha]^T$. If the number of constraints equals the number of degrees of freedom, the computed solution is exact and the function $\mathbf{L}$ interpolates the training values. For $N_c < N_s$, a least-squares approximation the training data is computed. In our case, the

Figure 1: A map of the experiment area. Its size is approximately 30 m × 25 m. Shown are the infrastructure node locations. Areas that were accessible to us are shaded.

training data is very noisy, and a smoother, approximate solution is desirable. The optimal number of centers depends on the complexity of $\mathbf{L}$, in our case, 3000 centers were optimal (see Sec. 5.2).

Since our kernel functions ϕ have global support and no singularities, the exact position of the centers $\mathbf{c}_i$ has little impact on the quality of the solution as long as the relevant parts of $\mathcal{X}_{k,N}$ are well sampled. We place the centers along the path by using a subset of the training sample positions as centers.

4.2 Localization

Once training is complete, we can use $\mathbf{L}$ to compute the position of the mobile node given measured data $\mathbf{r}(t)$ along the trajectory of the node. The RSSI measurements that are the components of $\mathbf{r}(t)$ are noisy, and therefore the computed position $\mathbf{p} = \mathbf{L}(\mathbf{r}(t))$ is not reliable enough. We remedy this by prefiltering the RSSI measurements, in particular, by replacing $\mathbf{r}(t)$ by a low-pass filtered version $\bar{\mathbf{r}}(t)$.

However, this introduces a bias in the localization: the computed position trails the true position as changes in the RSSI values $\mathbf{r}(t)$ gradually impact the low-pass filtered version $\bar{\mathbf{r}}(t)$. We solve this by requiring the mobile node to send a burst of b packets in a rapid succession (we use $b = 5$ packets in 50 ms) and using low-pass filter of width b. In our implementation, we use a simple box filter. This ensures that $\bar{\mathbf{r}}(t)$ is rapidly updated, keeping the position bias to a minimum.

5. EVALUATION

We have validated our localization method in a small indoor testbed. Fig. 1 shows a map of the space. We distributed 11 MicaZ motes [5] in the area. A mobile mote sends beacon packets in regular intervals. Infrastructure nodes receive these packets and record sequence numbers, timestamps, and various signal strength indicators. MicaZ's CC2420 radio chip [18] allows us to measure RSSI and LQI for each received packet. Additionally, we compute a packet reception ratio (PRR) by counting the number of received packets in each burst.
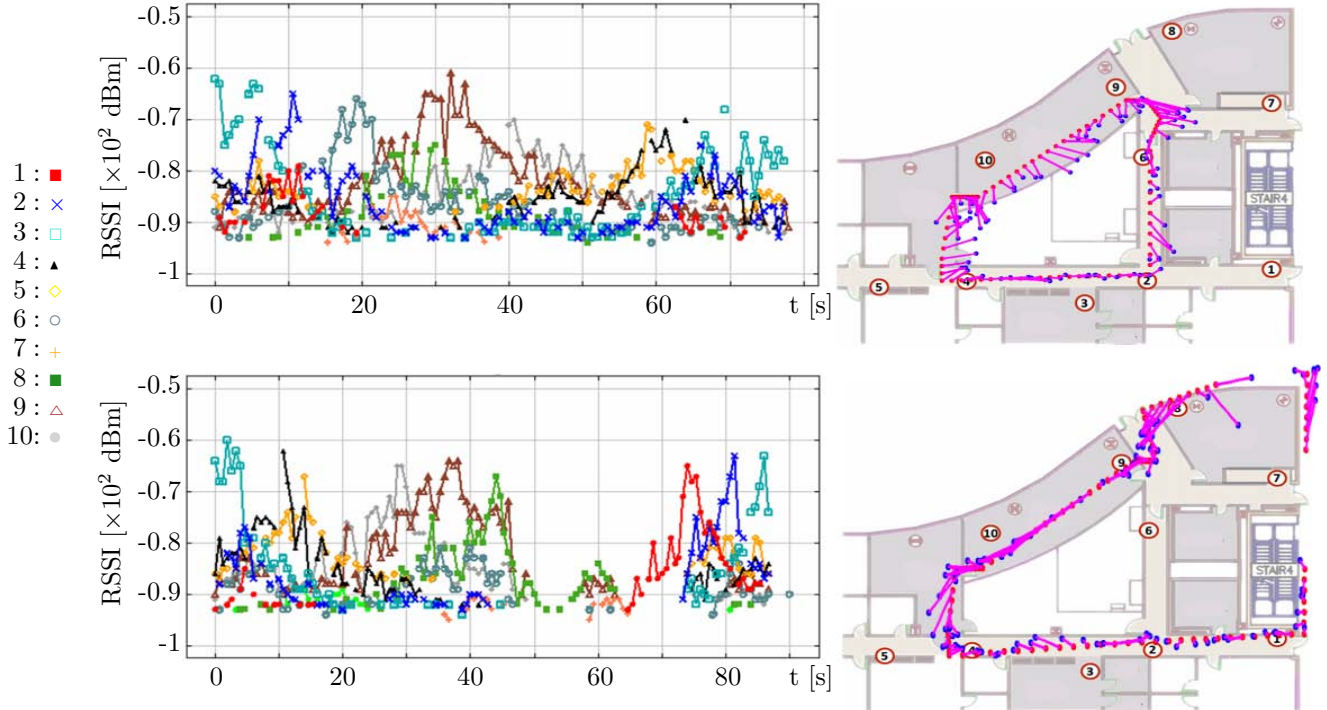
Figure 2: Localization for routes (f) and (g). Figures on the left show recorded RSSI measurements for all nodes, figures on the right show estimated and ground truth locations (blue and red dots, respectively), corresponding pairs connected by a line.

As mentioned before, our method can be applied to any indicator, as long as it gives some information about the distance between transmitter and receiver. To study the effect of measurement noise on the localization accuracy, the mobile node transmits a burst of 30 packets, rather than a single beacon packet. This allows us to experiment with different filtering techniques to improve the input signal. Further, we chose Δt=0.6 s in our experiments, i. e., the mobile node transmits one burst of packets every 0.6 s.

In the training phase, we require pairs $(\mathbf{r}(t), \mathbf{p}(t))$ as described in Sec. 4.1. Therefore, we move the mobile node along a predefined route through the experiment area. We store timestamps for predefined waypoints, while continually recording RSSI and LQI at all nodes. The training positions $\mathbf{p}(t)$, which will also be used as ground truth, are estimated using linear interpolation between the waypoints. Fig. 2 shows some of the routes and associated RSSI measurements. In our experiments, we never use the same route for both training and evaluation.

5.1 Error Measures

In order to evaluate our algorithm, we compute two error measures along the path. The position error

$$e_{\text{pos}}(t_i) = \|\mathbf{L}(\mathbf{r}(t_i)) - \mathbf{p}(t_i)\| \tag{5}$$

gives us information about the general performance of the localization algorithm. We also compute the distance to the closest point on the path

$$e_{\perp}(t_i) = \min_{\mathbf{p} \in \mathcal{P}} \|\mathbf{L}(\mathbf{r}(t_i)) - \mathbf{p}\|, \tag{6}$$

where $\mathcal{P}$ is the ground truth path that we are testing against. This error measure does not contain errors due to imperfect timing information used for ground truth computation. It also contains information on how accurately a world-space trajectory can be reconstructed using the localization data.

5.2 Parameters

Our algorithm accepts several parameters that can be tuned for optimal performance. The number of past measurements k determines how much historical information about the trajectories is available. As can be seen in Fig. 3 (a), a value of $k = 4$ is optimal in our case. Note that this significantly outperforms the case $k = 0$, in which our algorithm reduces to a traditional localization method based on averaged RSSI measurements.

The properties of the fitted RBF function depend on the number of kernel functions (centers) used. Varying the number of kernel functions, we see that using around 3000 kernels is optimal in our setting (Fig. 3 (b)). Decreasing the number of centers degrades the quality of the RBF approximation, while increasing it further yields a function that interpolates the constraints too faithfully. Since the input data is very noisy, interpolation is not desirable. The optimal number of RBF centers depends on the complexity of the environment in which localization is attempted.

As described in Sec. 4.2, the function $\mathbf{L}$ is queried with a low-pass filtered RSSI vector $\bar{\mathbf{r}}$. Fig. 3 (c) shows the influence of the width b of the box filter on the local-
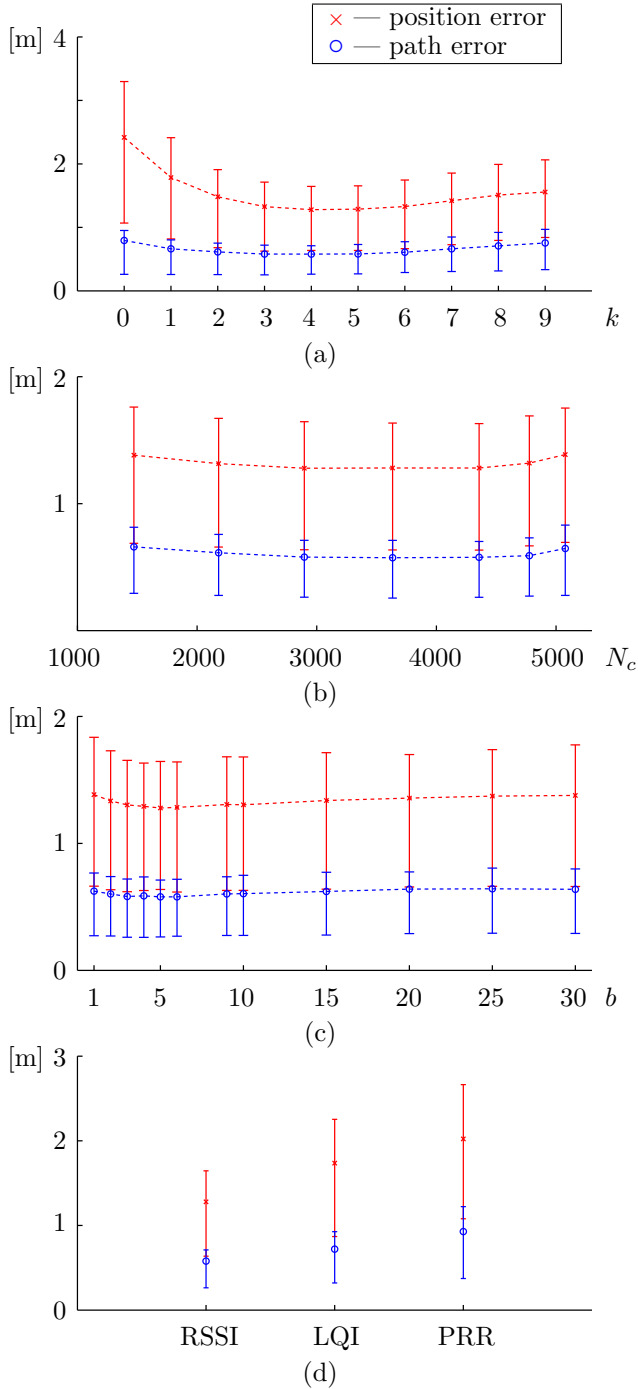
(a)

(b)

(c)

(d)

Figure 3: Position error ($e_{\mathbf{pos}}$, red) and path error ($e_\perp$, blue) depending on algorithm parameters. The graphs show the mean error, error bars represent 75% and 25% quantiles. (a) History size k. (b) The number N_c of RBF centers. (c) Low-pass filter width b. (d) Measurement type: RSSI, LQI, or PRR with 5 burst packets.
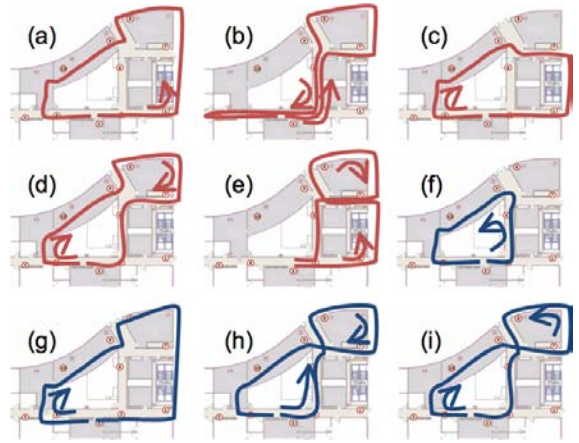

Figure 4: Routes used. (a)–(e) for training, (f)–(i) for testing.

ization results. Our experiments show that a moderate filter width of $b = 5$ performs best. While lower values of b yield noisy measurements, high values of b smooth out important details and degrade the localization accuracy.

Unless otherwise noted, all results were obtained using $k = 4$, $N_c = 3000$, and $b = 5$, as described above.

5.3 Results

We use a training set of 5 routes (a) - (e), each of which we recorded 5 times. All errors are computed for 4 different routes (f) - (i), each recorded several times to obtain a total of 24 test routes. Although the routes used for testing are composed of path segments seen during training, none of the testing and training routes are the same (see Fig. 4).

Table 1 summarizes the localization results. Overall, our method is able to localize a mobile node with a mean position error of less than 1.3 m. If radio contact is lost entirely, we do not predict a location. Note that by linearly interpolating position values between waypoints, we implicitly assume that the motion of the mobile node is of constant speed between waypoints. This assumption is generally not true in our experiments. We moved the mobile device by carrying it while walking, and the routes involved opening doors, and at one point, entering a 6 digit access code. Given perfect ground truth, our results would probably improve further.

We can see in the experimental data that errors are more likely to be along the path than perpendicular to the path (c.f. Fig. 2). We attribute this in part to the systematic timing errors in the ground truth. Table 1 also contains the path error $e_\perp$, which is not affected by timing problems with the ground truth. However, it is also invariant to other errors.

As mentioned above, our algorithm does not require RSSI measurements, and can operate using LQI or approximate PRR measurements (see Fig. 3 (d)). Using these measures, the mean position error, measured across all test routes, then increases to 1.7 m and 2 m respectively. Comparing the three measures, the reason

e_{pos}	2.5%	25%	50%	75%	97.5%	max	avg
(f)	0.18	0.67	1.07	1.64	4.40	5.63	1.32
(g)	0.14	0.58	0.98	1.58	3.40	4.88	1.20
(h)	0.19	0.68	1.09	1.70	3.76	6.92	1.32
(i)	0.20	0.58	0.97	1.64	4.30	8.22	1.25
All	0.18	0.64	1.03	1.65	4.09	8.22	1.28

$e_{\perp}$	2.5%	25%	50%	75%	97.5%	max	avg
(f)	0.09	0.30	0.52	0.86	2.24	4.97	0.68
(g)	0.07	0.23	0.34	0.51	1.46	4.03	0.44
(h)	0.09	0.28	0.47	0.74	1.75	3.03	0.57
(i)	0.08	0.26	0.38	0.65	2.47	4.36	0.56
All	0.08	0.26	0.43	0.71	2.13	4.97	0.58

Table 1: Summary of localization results (in meters).

for the inferior performance of LQI and PRR measurements is easy to see: the LQI values remain high whenever there is radio coverage and drop sharply as soon as connectivity is about to be lost. Therefore, these measurements are good as an indicator for link quality, but contain only little information about distance of the neighboring node. PRR measurements have similar problems: as long as no packets are lost, PRR provides limited information about distance. As packets start being lost, the measurement is necessarily discrete. In our case, we used averaging over only five packets, giving us only a very rough idea of how good, or bad, the link really is. In this light, the results for PRR are astonishingly good.

6. CONCLUSION AND FUTURE WORK

We have presented a novel method for locating mobile users within a network using RSSI measurements. The algorithm uses not only the current RSSI measurement for localization, but takes advantage of past values. Using historical values significantly increases the stability of the localization. This is particularly important in indoor settings, where RSSI measurements are extremely noisy, in particular for mobile nodes. Using trajectory information in RSSI space allows us to locate mobile users even while they are moving.

This research leaves ample room for future work. The dimension of the RSSI trajectory space grows linearly with the number of nodes. Therefore, exploiting locality is crucial to ensure the scalability of the algorithm. Local, overlapping localization areas can be used to solve this problem.

Since we use past measurements at fixed time intervals, we implicitly assume that the speed of the mobile user at a given position is similar during training and localization. Although the method can be trained for different speeds, explicitly handling speed differences, for example using dynamic time warping, would be an interesting extension.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User-Location and Tracking System. In *Proc. InfoCom*, volume 2, pages 775–84, 2000.

[2] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, 2004.

[3] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.

[4] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye. SpaseLoc: An adaptive subproblem algorithm for scalable wireless sensor network localization. *SIAM J. Optimization*, 17(4):1102–1128, 2006.

[5] Crossbow Technology Inc., http://www.xbow.com. *MICAz wireless measurement system*, June 2004.

[6] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *Trans. on Embedded Computing Sys.*, 4(4):877–906, 2005.

[7] J. Krumm and E. Horvitz. LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In *Proc. MobiQuitous*, 2004.

[8] B. Kusy, A. Ledeczi, and X. Koutsoukos. Tracking mobile nodes using RF Doppler shifts. In *Proc. SenSys*, pages 29–42, 2007.

[9] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavrati. On the feasibility of using wireless ethernet for indoor localization. *IEEE Trans. Robotics and Automation*, 20(3):555–559, 2004.

[10] K. Lorincz and M. Welsh. MoteTrack: a robust, decentralized approach to RF-based location tracking. *Personal Ubiquitous Comp.*, 11(6):489–503, 2007.

[11] D. Niculescu and B. Nath. DV Based Positioning in Ad Hoc Networks. *J. Comm. Sys.*, 22(1):267–280.

[12] N. Patwari, A. III, M. Perkins, N. Correal, and R. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Trans. on Sig. Proc.*, 51(8):2137–2148.

[13] R. Peng and M. L. Sichitiu. Angle of Arrival Localization for Wireless Sensor Networks. In *Proc. Conf. Sensor and Ad-Hoc Comm. and Networks*, 2006.

[14] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *Proc. Infocom*, pages 172–183, 2005.

[15] Z. Sahinoglu, S. Gezici, and I. Guvenc. *Ultra-Wideband Positioning Systems- Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press, 2008.

[16] P. Sikka, P. Corke, and L. Overs. Wireless Sensor Devices for Animal Tracking and Control. In *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pages 446–454, 2004.

[17] R. Stoleru, T. He, and J. A. Stankovic. Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks. In *Proc. LCN*, pages 480–489, 2004.

[18] Texas Instruments, Data Sheet, Rev. 1.4. *Chipcon CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*.

[19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[20] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, 2006.