SmartMoveX on a Graph – An Inexpensive Active Badge Tracker

John Krumm[1], Lyndsay Williams[2], Greg Smith[1]
{jckrumm, lyn, gregsmi}@microsoft.com

[1] Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA

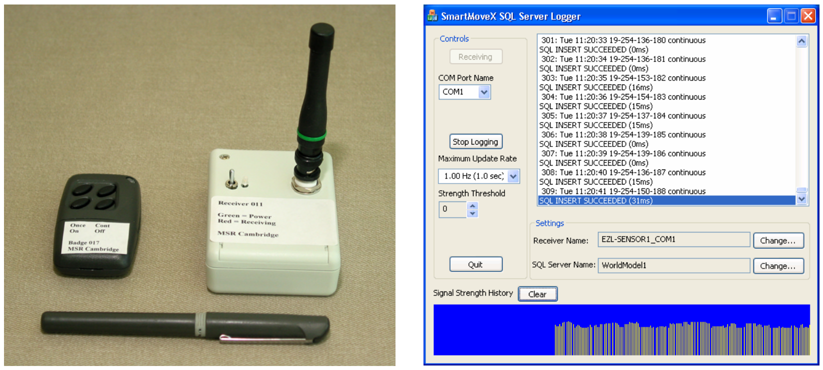
[2] Microsoft Research
7 J J Thomson Ave
Cambridge CB3 0FB, UK

Abstract. Measuring the locations of people in a building is an important part of ubiquitous computing. We present new hardware and software for this purpose. The hardware, called SmartMoveX, is an active badge system in which a small radio transmitter is attached to the person being tracked. Receivers placed in the building's existing offices, connected to existing PCs, transmit signal strength readings to a central PC using the building's existing computer network. Combined with the low cost of the hardware, using the existing network makes this active badge system much less expensive than many others. To compute locations based on signal strength, we gathered signal strength readings from predefined location nodes in the building. We defined a graph on these nodes, which allowed us to enforce constraints on computed movements between nodes (*e.g.* cannot pass through walls) and to probabilistically enforce our expectations on transitions between connected nodes. Modeling the data with a hidden Markov model, we used the Viterbi algorithm to compute optimal paths based on signal strengths over the node graph. The average location error was 3.05 meters, which compared favorably to a simple nearest neighbor algorithm's average location error of 4.57 meters.

1 Introduction

Knowing the indoor location of people is widely considered to be a key enabler of ubiquitous computing applications. This paper presents a new active badge system for this purpose. The two hardware components of our system are shown in Figure 1(a). The badge is worn by a person, and it transmits radio frequency (RF) signals to several receivers placed around the building. The receivers measure the RF signal strength from the badge transmissions. The signal strengths of the receptions are combined to compute the location of the badge in a tracking algorithm based on a hidden Markov model (HMM) defined on a graph of discrete locations in the building.

One of the system's features is its low cost, partly due to its low parts cost. The parts for a badge transmitter cost about US$ 6, for a receiver about US$ 16. The major cost saving, however, comes from reusing the building's existing computer network instead of requiring a custom network as many active badge systems do. Each of the receivers is connected to a normal PC via an RS-232 serial cable. When a receiver receives a badge transmission, it sends data to its host PC, which in turn forwards the data over the building's existing network to a central PC for storage and analysis. In a real implementation, the receivers would be placed in occupied offices and connected to existing PCs. These PCs would each run our very light-weight data-forwarding



(a) **(b)**

Fig. 1. (a) The badge transmitter is on the left, with buttons for on, off, one-time transmit and periodic transmit. The receiver, on the right, is powered externally and connects to a PC via RS-232. The receiver measures the signal strength of transmissions from the badge. The pen is just a pen. (b) This is a screen shot of our data logging program. Each receiver is connected to a host PC running an instance of this program, which retransmits data records to a central SQL Server database.

program in the background, shown in Figure 1(b). In our implementation, we used four receivers to cover 350 square meters. Not counting the cost of the badges nor any PCs, and considering the existing network infrastructure as free, this works out to a cost of about US$ 0.18/square meter.

The location of the badge is measured based on the signal strengths of the RF receptions at all the receivers. We match the live signal strengths with a set of calibration signal strengths taken from a set of known, discrete positions ("nodes") in the building, shown in Figure 2. We implemented a simple nearest-neighbor algorithm that matched the live measurements with the closest set of training measurements. This gave a mean location error of 4.57 meters. We implemented another algorithm that confined the badge's path to physically possible paths between the nodes as shown in Figure 2. Based on an HMM, this algorithm reduced the average error of our location measurements to 3.05 meters. While we show this algorithm working on signal strengths from our own badge hardware, it would apply to many other forms of sensor-based tracking in a building, including other active badge systems and signal strengths from a wireless network.

2 SmartMoveX Active Badge and Network Data Logger

The hardware for our active badge system is called SmartMoveX and was invented at Microsoft Research in Cambridge, UK. SmartMoveX consists of a small radio transmitter that transmits 433 MHz FM to multiple receivers as shown in Figure 1(a). Each transmission packet contains an ID number of the transmitter, a measured physical activity level, and an incrementing transmission counter to help detect missed transmissions. The transmitter uses a PIC microcontroller to read and control the functions of a tilt switch, tilt angle sensor, the transmitter itself, and four outside buttons. The tilt switch is used to shut off the transmitter after a minute of inactivity and to turn it on again when it moves, thus saving battery power. The physical activity of the transmitter is measured by how many times the tilt angle sensor measures an angle beyond a preset threshold. Although we don't use this information in our tracking application, it could be used to infer users' activities like running, walking, and sitting. In our experiments, we used the transmitter's periodic transmit mode which gave a new transmission every one second. The cost estimate of US$ 6 per transmitter is based on a transmitter without the tilt switch and tilt sensor, as they are not integral to the location-tracking problem we address in this paper.

The SmartMoveX receiver is a small box with connections for DC power and RS-232. It has a 15cm antenna with which to receive transmissions. Each transmission is demodulated by a receiver chip which also generates a digital radio signal strength indicator (RSSI). The transmission data (ID, physical activity level, transmission counter) and RSSI are sent to a serial communications chip which forwards the data out the RS-232 port to the host PC.

On each host PC we have a logging program listening to the receiver, as shown in Figure 1(b). Upon receipt of a transmission record from the receiver, this program forwards the data to a SQL Server database on a single central PC. Gathering all the records in a database allows us to run queries and computations on the transmission data and allows us to run offline experiments for tracking such as was done for this paper.

3 Spatial Representation and Calibration

Each RF transmission from a mobile transmitter is heard by all the receivers in the area, resulting in a column vector of signal strength readings, $\mathbf{s}$. In our case we used four receivers, so each signal strength vector had four scalar elements. In open space we might expect the signal strength to fall off with the square of the distance between the transmitter and receiver. Unfortunately, this simple relationship does not necessarily hold, as shown by an experiment in [1] using hardware similar to ours. In a continuation of this work, [2] went on to use the more sophisticated path loss model of Seidel and Rapport[3], which was found to fit the data better. This model was also used by [4] to account for attenuation due to walls. However, it is still difficult to predict the effect of furniture, devices, and people on signal strength, particularly if the locations of these things are unknown. In fact [4] found that their analytical model of signal strengths worked significantly worse than their pure empirical model for measuring locations within a multi-room building.

Based on the difficulty of analytically modeling signal strengths, we adopted an empirical approach to predicting signal strengths similar to the RADAR system of [4]. For this approach we took a series of calibration signal strength readings at predefined node locations in our building, shown in Figure 2. We manually picked the node positions, generally one for each office, one for each office-size rectangle in larger rooms, and one outside each door in the hallways. For each of the $N = 42$ nodes, we call the calibration signal strength readings $\mathbf{s}_i^{(j)}$, where i indexes the node ($i = 0 \ldots N-1$), and j indexes the N_i calibration samples at that node. Each $\mathbf{s}_i^{(j)}$ is a vector of four

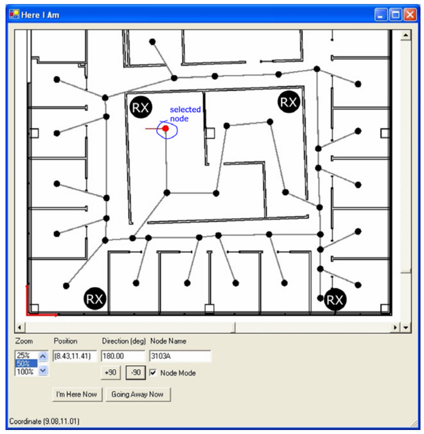


Fig. 2. This is a screen shot of our manual calibration program. It shows the layout of the rooms, the nodes, and connections between the nodes. The locations of the four receivers are shown with a RX.

signal strengths, one from each receiver. We took a total of 1256 calibration signal strength vectors. The number of calibration vectors taken for each node varied from 12 to 50, with an average of 30.

We took the calibration readings in about 30 minutes by walking around with a laptop PC wirelessly connected to our building's network running the calibration program shown in Figure 2. At each node the walker rotated and moved in an effort to sample the likely positions and orientations of a person wearing a transmitter near that node. We note that our sampling of signal strengths at the nodes was not meant to measure the frequency of occurrence of signal strengths. Instead, we just tried to sample all the signal strength vectors that we would likely measure at each node. For this reason, it did not make sense to try to summarize the training data with a histogram or probability distribution function.

We gathered test data by walking on two prescribed paths through the nodes with a transmitter pinned to the front of the walker's shirt. The transmitter was set to periodically transmit at a one second interval, which resulted in a sequence of time-stamped signal strength vectors $\mathbf{s}(i)$. We computed the walker's speed by summing the distances between the visited nodes and dividing by the elapsed time of the walk. Using this speed and the distances between the nodes, we could estimate the walker's nearest node for each of the time-stamped test transmissions. In total, our ground truth data consisted of 140 transmissions, each characterized by the closest node and a vector of four signal strengths.

4 Nearest Neighbor Location Measurement

The general procedure for measuring the location of a transmitter is to compare its latest signal strength vector $\mathbf{s}$ against the calibration signal strength vectors described above. We implemented a simple nearest-neighbor location algorithm, similar to RADAR[4], as a baseline against which to test our more sophisticated graph-based algorithm described in the next section. The nearest neighbor algorithm simply finds the $\mathbf{s}_i^{(j)}$ with the minimum Euclidian distance to $\mathbf{s}$ and declares the transmitter to be at the node from which this $\mathbf{s}_i^{(j)}$ came. This computation is relatively fast with only 1256 calibration vectors to compare to, certainly fast enough to keep up with the 1 Hz transmission rate of the transmitter.

We quantified the results by computing the Euclidian distances between the computed and actual nodes. The average error for our 140 test nodes was 4.57 meters. The distribution of error distances is shown in Figure 3.

5 Tracking on a Graph

The nearest neighbor algorithm ignores any adjacency relationships between the nodes, so it allows instantaneous transitions between nodes that are separated by a wall and/or an arbitrarily large distance. We found we could improve the performance of the system significantly by adding path constraints that only allow physically realizable paths through the nodes. In fact, using this constraint reduced the mean error from 4.57 meters in the nearest neighbor case to 3.05 meters, using the same calibration data.

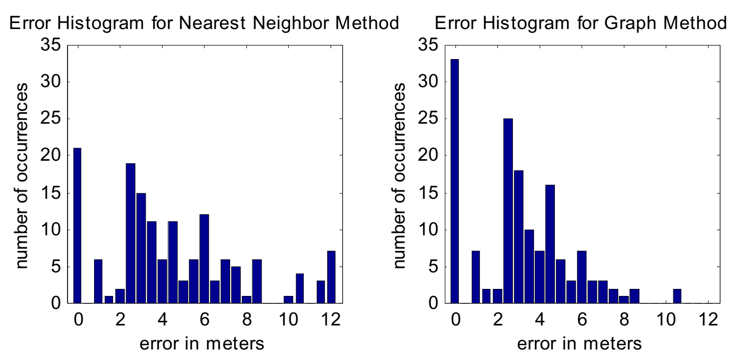


Fig. 3. These histograms show the distribution of errors measured in meters between the actual nodes and computed nodes for 140 test nodes. The Graph Method, using the Viterbi algorithm, makes fewer overall errors in picking the right node (0 column), and has a lower mean error of 3.05 meters compared to 4.57 meters for the Nearest Neighbor method.

5.1 Graph and Transition Probabilities

We instantiated the constraints with the manually constructed graph of nodes shown in Figure 2. The connections between the nodes show which paths we allow. This is an easy way of preventing paths from going through walls and of preventing superhuman transitions between distant nodes. We also attached transition probabilities between connected nodes as a soft constraint on the likelihood of moving between nodes or remaining at the current node. We assigned the transition probabilities manually based on our assumptions about people's behavior. For instance, we assumed that the probability of moving from a node in an office to the node outside the office's door was 0.05 in the one-second interval between transmissions, with the remaining 0.95 being the probability of remaining at the office node. We define $a_{i \to j}$ as the probability of transitioning from node i to node j, and we pick the transition probabilities such that $a_{i \to j} \geq 0$ and $\sum_{j=0}^{N-1} a_{i \to j} = 1$, where N is the number of nodes. We note that $a_{i \to i}$ indicates the probability of staying at node i, and it is never zero for our model. To eliminate the possibility of transitioning between node i and j, we simply set $a_{i \to j} = 0$.

Another *a priori* set of probabilities that we need is the probability of starting a path at node i as which we denote as π_i. Since we have no prior knowledge of where a path starts, we set $\pi_i = 1/N$, where N is the number of nodes.

5.2 Hidden Markov Model

The graph, initial state probabilities, and transition probabilities represent our *a priori* knowledge of people's behavior. The remaining element in tracking location is the signal strength data. The standard way to combine uncertain measurement data, discrete states, and transition probabilities between the states is a hidden Markov model (HMM)[5].

The "Markov" part of the HMM is manifested in our transition probabilities. We will say the sequence of nodes in a person's path up to time $i-1$ is $\{n_0, n_1, n_2, \ldots, n_{i-1}\}$. The first-order Markov assumption says that the probability of transitioning to some node n_i at time i is a function only of node n_{i-1} and not any of the other previous nodes. Stated as an equation:

$$P(n_i | n_{i-1}, n_{i-2}, \ldots, n_0) = P(n_i | n_{i-1}) = a_{n_{i-1} \to n_i} \tag{1}$$

The "hidden" part of the HMM has to do with the signal strength vectors which are probabilistically related to the node n_i via the probability distribution function $P(\mathbf{s} | n_i)$. Our approach to computing this was to use the training data we originally gathered for our nearest neighbor algorithm described in Section 4. We gathered this data by stopping at every node, turning and slightly moving, while recording signal strengths. This was intended to capture a series of plausible signal strength vectors, but not intended to capture their frequency of occurrence. To compute $P(\mathbf{s} | n_i)$ from this data, we first find the nearest neighbor calibration vector at the node n_i:

$$\mathbf{s}_{n_i}^{*} = \min_{\mathrm{j=0 \ldots N}_{n_i}-1} \left\| \mathbf{s} - \mathbf{s}_{n_i}^{(j)} \right\| \tag{2}$$

Where, as a reminder, N_{n_i} is the number of calibration vectors at node n_i, and $\mathbf{s}_{n_i}^{(j)}$ are the calibration vectors at node n_i. The assumption here is that the calibration vector from node n_i that most closely matches $\mathbf{s}$ corresponds to the physical pose of the walker at that node that produced $\mathbf{s}$. The probabilistic part of the observation probability function comes from the signal noise inherent in a series of signal strength readings from a stationary transmitter, which we model as Gaussian. An experiment with our hardware suggests that the noise standard deviation of signal strength is approximately one unit. Taking $\mathbf{s}^*_{n_i}$ as the mean and assuming statistical independence among the four receivers, the observation probability function is

$$P(\mathbf{s}|n_i) = \frac{1}{\left(\sqrt{2\pi}\right)^4 |\Sigma|^{0.5}} e^{-\frac{1}{2}\left(\mathbf{s}-\mathbf{s}^*_{n_i}\right)^T \Sigma^{-1} \left(\mathbf{s}-\mathbf{s}^*_{n_i}\right)} \tag{3}$$

where the covariance matrix Σ is the 4x4 identity matrix scaled by the standard deviation of the signal strength readings, which we measured to be approximately one.

With the initial state probabilities, transition probabilities, and observation probabilities, we can compute the probability of a given path through the nodes given the signal strength readings. The relevant parts are:

Transmission times	$\{t_0, t_1, t_2, \ldots, t_{T-1}\}$
Nodes along path	$\mathbf{N} = \{n_0, n_1, n_2, \ldots, n_{T-1}\}$
Measured signal strengths	$\mathbf{S} = \{\mathbf{s}(t_0), \mathbf{s}(t_1), \mathbf{s}(t_2), \ldots, \mathbf{s}(t_{T-1})\}$
Data likelihood	$P(\mathbf{S}\|\mathbf{N}) = \pi_{n_0} a_{n_0 \to n_1} a_{n_1 \to n_2} \ldots a_{n_{T-2} \to n_{T-1}} \prod_{i=0}^{T-1} P(\mathbf{s}(t_i)\|n_i)$

Given a set of signal strengths $\mathbf{S}$, we would like to find the path $\mathbf{N}^*$ that maximizes the data probability $P(\mathbf{S}|\mathbf{N})$. The Viterbi algorithm, described nicely in [5], gives an efficient way of finding $\mathbf{N}^*$ by considering all the data up to and including the current time.

Using the Viterbi algorithm along with the transition probabilities and observation probability function, we achieved an average location error of 3.05 meters compared to an average error of 4.57 meters for the nearest neighbor algorithm described previously. The distribution of errors is shown in Figure 3. We attribute this improvement to three factors:

1. The graph limits paths to only those that are physically possible.
2. The transition probabilities encourage paths that conform to our *a priori* expectations of people's behavior.
3. The Viterbi algorithm uses *all* the data up to and including the current time to compute the most likely current node. The nearest neighbor algorithm using only the current signal strength vector.

6 Comparisons to Similar Systems

Active badge location systems date back to the work of Olivetti Research Laboratory in 1989[6], which used diffuse infrared to measure proximity. Other location measurement systems since then have used ultrasound, RF signal strength, and RF time of flight. An excellent taxonomy and survey of these systems can be found in [7].

We will compare our system against technologies with similar hardware or software. The most similar existing hardware is from the company RFIDeas Inc. Their “AirID” product allows users approaching a PC to be automatically logged on by virtue of their wearing a badge transmitting RF[8]. Although this system was not designed to measure location in a building, Hightower *et al*.[1] investigated its use for location measuring in a room. They measured 3D location based on multiple receivers and an empirically derived function giving signal strength as a function of distance to the receiver. The hardware limited signal strength measurements to two bits, which in turn limited the system’s resolution to a cube of three meters on a side.

One of our system’s advantages is its low infrastructure cost. There are systems with theoretically even less expensive infrastructure costs, such as the SpotOn hardware by Hightower *et al.* [1] and the “Positioning by Diffusion” idea from Spratt[9]. Both these systems can theoretically operate with no fixed base stations, although real world test results have not been published yet. MIT’s Cricket[10] location-support system uses non-networked, ceiling-mounted ultrasonic transmitters whose cost per unit is about the same as our RF receivers, making its infrastructure cost similar to ours. Randell and Muller[11] describe a similar system with much higher spatial resolution and low cost. Both the infrared Active Badge and ultrasonic Active BAT[12] from AT&T Cambridge require their own dedicated network to connect statically mounted base stations, which is expensive.

Another closely related system, both in terms of hardware and software, is RADAR[4], which comes from our colleagues at Microsoft Research. The Nibble[13] system also uses 802.11 signals to compute locations in a building. Interestingly, it uses the measured signal-to-noise ratio instead of absolute signal strength. Nibble is based on a Bayesian network to compute the probability of being at any of a set of discrete locations in the building, much like our system. Nibble’s Bayesian network also supports the inclusion of transition probabilities between nodes.

7 References

1. Hightower, J., G. Borriello, and R. Want, SpotOn: An Indoor 3D Location Sensing Technology Based on RF Signal Strength, UW-CSE 2000-02-02, University of Washington, 2000.
2. Hightower, J., et al., Design and Calibration of the SpotOn Ad-Hoc Location Sensing System, http://www.cs.washington.edu/homes/jeffro/pubs/hightower2001design/hightower2001design.pdf.
3. Seidel, S.Y. and T.S. Rapport, *914 MHz Path Loss Prediction Model for Indoor Wireless Communications in Multi-floored Buildings.* IEEE Transactions on Antennas and Propagation, 1992. **40**(2): p. 207-217.
4. Bahl, P. and V.N. Padmanabhan. *RADAR: An In-Building RF-Based Location and Tracking System*. in *IEEE INFOCOM 2000*. 2000. Tel-Aviv, Israel.
5. Rabiner, L.R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-285.
6. Want, R., et al., *The Active Badge Location System.* ACM Transactions on Information Systems, 1992. **10**(1): p. 91-102.
7. Hightower, J. and G. Borriello, *Location Systems for Ubiquitous Computing.* Computer, 2001. **34**(8): p. 57-66.
8. Proximity Acitivated Identification, http://www.pcprox.com/Support/AIRID/airid.html.
9. Spratt, M., An Overview of Positioning by Diffusion, HPL-2001-207, HP Laboratories Bristol, 2001.
10. Priyantha, N.B., A. Chakraborty, and H. Balakrishnan. *The Cricket Location-Support System*. in *6th ACM International Conference on Mobile Computing and Networking*. 2000. Boston: ACM Press.
11. Randell, C. and H. Muller. *Low Cost Indoor Positioning System*. in *Ubicomp 2001*. 2001. Atlanta, GA, USA: Springer.
12. Harter, A., et al. *The Anatomy of a Context-Aware Application*. in *5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*. 1999. Seattle, WA: ACM Press.
13. Castro, P., et al. *A Probabilistic Room Location Service for Wireless Networked Environments*. in *Ubicomp 2001*. 2001. Atlanta, GA, USA: Springer.