Submitted to IJCAI 2003

A Learning Algorithm for Localizing People Based On Wireless Signal Strength That Uses Labeled and Unlabeled Data

Mary Berna, Brad Lisien, Brennan Sellner, Geoffrey Gordon, Frank Pfenning, Sebastian Thrun
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Abstract

We propose a probabilistic technique for localizing people through the signal strengths of a wireless IEEE 802.11b network. Our approach uses data labeled by ground truth position to learn a probabilistic mapping from locations to wireless signals, represented by piecewise linear Gaussian. It then uses sequences of wireless signal data (without position labels) to acquire motion models of individual people, thereby improving its ability to predict individuals' motions. As a result, the localizer becomes increasingly accurate while it is being used. The approach has been implemented in an office environment, and results are reported for a systematic study involving labeled and unlabeled data.

Recent developments in wireless technology have decreased the cost and increased the coverage of IEEE 802.11b wireless networking. Many businesses, hospitals, airports, schools, homes, etc. are now equipped with wireless digital networks. This trend opens up new possibilities for mobile computing, many of which demand location awareness of individual users. To attain such awareness, a number of research groups have begun to use wireless signal strength as a way to localize people. In the past months, a number of techniques for people localization has been proposed [1; 5; 9; 10; 11; 18; 20; 21].

In all these approaches, localization is achieved by monitoring the strengths of the signals emitted by access points placed at fixed locations, as those are received on a mobile device. By integrating such measurements over time, an estimate of a person's location can be obtained. Unfortunately, knowledge of the locations of the access points alone is generally insufficient to determine a person's location with accuracy. This is because building elements, such as walls, doors, and building geometry severely affect the spread of the wireless signal (see Fig.1). For this reason, the best present techniques use data-driven *learning* to generate maps of signal strength as a function of location.

For example, Ladd and colleagues [10; 11] have developed a technique in which a person carries a wireless receiver through an environment that records the signal strength in regular time intervals. To ground these measurements in real-world coordinates, the person also measures her own location (e.g., with tape measure). The resulting measurements are compiled into a geometric map of the environment, which describes the expected signal strength as a function of location. Once such

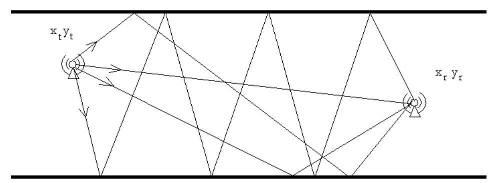

Figure 1: Radio waves from a IEEE 80211.b transmitter take multiple paths to the receiver. Because each path has a different length, the signal strengths will be different when detected by a mobile receiver. To accurately model these various paths, an accurate building model is required.

a map has been acquired, it can be used to localize people. Ladd and colleagues integrate measurements over time using Markov localization, a probabilistic localization technique originally developed in the context of mobile robot localization [8; 17]. Probabilistic localization techniques require a motion model for people; Ladd and colleagues use a random Brownian motion model in their approach.

Unfortunately, the process of data collection is tedious. Collecting samples of signal strength is relatively easy; however, labeling these samples with the correct position is an expensive process. This raises the question as to whether data *without* position labels can be used in any way to improve the system performance. Such unlabeled data can be collected continuously as people navigate through their environments. Thus, techniques that can exploit such data would make it possible to develop localization software that continuously improves its performance, and adapts to the building's users.

This paper proposes an algorithm that does just this: it continually improves the localization of people by learning from unlabeled data. Just as various previous localization techniques, our approach relies on labeled data to acquire a map of signal strength. However, our approach also exploits *unlabeled* data to continually improve the localization accuracy. It does this by continually refining a motion model of people walking about their environment. Learning motion models from *labeled* data is straightforward. Our approach learns this model from unlabeled data, using an online variant of the expectation maximization (EM) algorithm [6; 12]. In this way, it enables our algorithm to adapts to an individual person, exploiting regularities when navigating her environment. The idea of using EM to label unlabeled data is not new; it has been applied with great success in various domains, such as information retrieval and geosciences[3; 14; 16]. In the context of the present paper, the result of applying

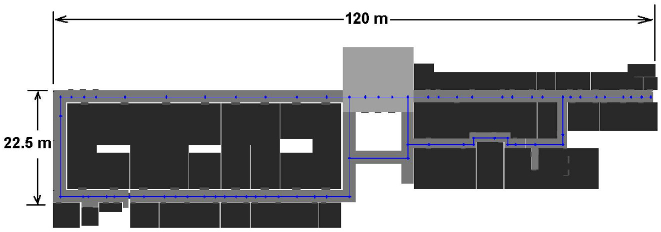

Figure 2: Map of the building with a 1-dim manifold describing the space of locations y.

EM to label data can be interpreted as the forward phase of the Baum-Welsh algorithm used to train hidden Markov models [15]; but here it is applied to a continuous state space.

In an empirical evaluation, we compare our results to the most accurate published results. An empirical evaluation in [10; 11] in shows that an accuracy of 1.5 meters is attained in 70% of all cases; the accuracy in the remaining 30% is not reported. By the same measure, our approach achieves a comparable error rate of 2.1 meters, but it requires a factor of 50 less labeled training data. These numbers are to be taken with a grain of salt, since the experimentation was carried out in a different building, with a different density of wireless access points, etc. We also show an improvement of the average localization error by 3.2 meters through continual learning from unlabeled data.

1 Learning Signal Strength Maps from Labeled Data

In a first step, our algorithm learns a signal strength map from labeled data. The input to this learning process is a set of labeled data items, consisting of a vector of signal strength measurements and a location at which these measurements were recorded:

$$\mathbf{x_k} \quad = \quad (\theta_{1,k}, \ldots, \theta_{N,k}, y_k)^T \tag{1}$$

Here $\mathbf{x}_k$ is the k-th training example, $\theta_{n,k}$ is the measurement of signal strength of the n-th access point (from N different access points), and y_k is the position at which the measurement is taken.

Our approach then learns a piecewise-linear Gaussian model (with fixed variance) of the conditional probability

$$p(\theta \mid y) \tag{2}$$

which related locations y in the environment to signal strengths θ. Our representation of this probability is based on a map of the environment, like the one shown Fig. 2. From this map, we extract a one-dimensional submanifold (with intersections) that corresponds to the thresholded Voronoi diagram described in [4].[1] This manifold defines the piecewise linear-Gaussian representation. In particular, our approach represents $p(\theta \mid y)$ by a function $\mu(y)$ plus an additive, zero-mean

[1]We note that such an approach works best in corridor environments to which our approach is indeed tailored, but it may fail in open environments. However, we do not view this as a principle limitation of our approach, as our representation is easily extended to handle those as well.

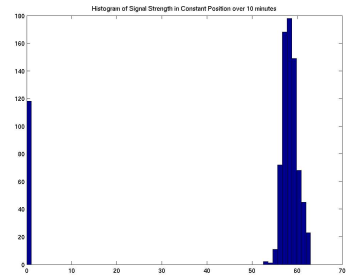

Figure 3: Distribution of signal strength of a single access point accumulated over a 10 minute period. The measurements are approximately normal distributed, with occasional maldetection events. because of the latter, zero-strength measurements are ignored.

Gaussian noise term with variance σ^2. The function μ is a piecewise linear function that

$$\mu(y) \quad = \quad \frac{1}{\sum_l d(y_l, y)} \sum_l d(y_l, y)\; a_{n,l} \tag{3}$$

where l indexes over a finite collection of "nodes" on the submanifold (whose values define the piecewise linear function $\mu(y)$), and d is a thresholded negative L_1 distance function (a distance function that is c at distance zero and is bounded below by 0):

$$d(y_l, y) \quad = \quad \max\{0, c - |y_l - y|\} \tag{4}$$

The nodes y_l are spaced a distance c apart from each other. According to (14), the function μ is defined via the values at the nodes $a_{n,l}$. Each such value corresponds to the mean signal strength of the access point n at node l.

The values $a_{n,l}$ are now conveniently obtained from the training data using the maximum likelihood estimator:

$$a_{n,l} \quad = \quad \frac{1}{\sum_k d(y_l, y_k)} \sum_k d(y_l, y_k)\; \theta_{n,k} \tag{5}$$

The variance σ^2 of $p(\theta \mid y)$ is learned globally; it is assumed to be equal for all locations y and all access points n. This was found empirically to be a good approximation, and training a global covariance requires orders of magnitude less data than training variance values that depend on specific locations and access points. The global covariance is recovered by the obvious maximum likelihood estimate:

$$\sigma^2 \quad = \quad \frac{1}{NK} \sum_k \sum_n (\theta_{n,k} - \mu(y_k))^2 \tag{6}$$

where K is the total number of training examples. We note that μ is a function defined everywhere in 2D space, even though its support $a_{n,l}$ is focused on a (mostly) one-dimensional manifold.

Finally, we note that in our implementation, measurements that sporadically become $\theta_{n,k} = 0$ are discarded from the calculation. This is because in practice, wireless receivers occasionally fail to detect a nearby access points, in which cases they return a zero measurement. This filter effectively catches such outliers; without it, our Gaussian noise model would be inappropriate.

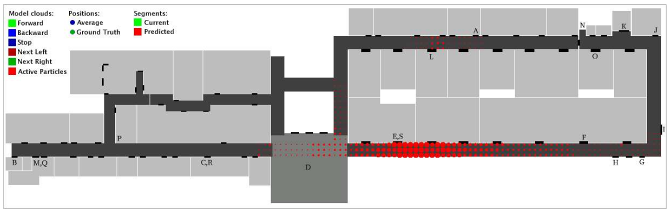

Figure 4: Access point strength as a function of measured location: the size of th circles indicate the strength of an individual access point.

2 Monte Carlo People Localization

People are localized using the MCL algorithm [19]. MCL is a version of Markov localization based on particle filters [7], specifically adept to tracking moving entities in continuous space.

In essence, MCL estimates a posterior density over a person's state. The estimate at time t is described by the following posterior distribution

$$p(x_t \mid \theta_1, \ldots, \theta_t) \tag{7}$$

Here θ_t is the measurement vector received at time t, and x_t is the state of the person at time t. The state is comprised of a person's location y_t (including orientation) and her velocity $\dot{y}_t$:

$$x_t \quad = \quad (y_t, \dot{y}_t)^T \tag{8}$$

MCL represents this posterior by a set of particles, which are samples of states:

$$p(x_t \mid \theta_1, \ldots, \theta_t) \quad \approx \quad \{x_t^{[m]}\} \tag{9}$$

Here m is the index of the sample, and each $x_t^{[m]}$ is a concrete position and velocity. The set of samples at time t is calculated recursively from that at time $t-1$ using the standard particle filter algorithm [7]:

1. For each $x_{t-1}^{[m]}$, draw a

   $$x_t^{[m]} \quad \sim \quad p(x_t \mid x_{t-1}^{[m]}) \tag{10}$$

   The function $p(x_t \mid x_{t-1})$ is the *motion model* of a person, In the simplest case, it is a Brownian motion model, meaning that a person's velocities are assumed to change in small, random increments.

2. Calculate the importance weight

   $$\begin{aligned} w_t^{[m]} &= p(\theta \mid x_t^{[m]}) \qquad (11) \\ &= \prod_n \frac{1}{2\pi\sigma^2} \, \exp\left\{ -\frac{1}{2} \frac{(\theta_{n,t} - \mu_n(y_t^{[m]}))^2}{\sigma^2} \right\} \end{aligned}$$

   where $y_t^{[m]}$ is the location estimate of the m-th particle, and $\theta_{n,t}$ is the signal strength of the n-th access point at time t.

3. Resample the resulting set of particles. Each state $x_t^{[m]}$ is drawn (with replacement) with a probability proportional to its importance weight $w_t^{[m]}$.

In the asymptotic limit, this algorithms recovers the exact posterior. MCL has previously been popular for mobile robot localization. It application to people localization, albeit new, it straightforward.

3 Learning Motion Models from Unlabeled Data

The primary contribution of this paper is a methodology to continuously improve the performance of the localizer, particularly in the absence of labeled data. Unlabeled data is of the form

$$(\theta_{1,t}, \ldots, \theta_{N,t})^T \tag{12}$$

It consists of signal measurements *without* a pose estimate.

Our approach utilizes unlabeled data to learn an informed motion model $p(x_t \mid x_{t-1})$. The intuition behind our approach is that for most people, uninformed Brownian motion is a poor model of navigation. Instead, people tend to walk straight through corridors, take specific turns, and enter specific offices. As will be demonstrated in the experimental results section below, a motion model specific to an individual person can yield much better prediction as to where a person is to go next, which in turn improves the ability of our algorithm to localize people.

In our approach, the motion model is realized by a Gaussian mixture model. In particular, a person at location y with velocity $\dot{y}$ may, with a certain probability, continue at approximately the same velocity, stop, or reverse her direction. If the person faces an intersection she might, with a certain probability, turn left or right (depending on the type of the intersection). We will refer to those discrete decisions as *actions*. Our model distinguished up to five different such actions, all of which are modeled by a different Gaussian posterior in state space. The exact number of possibilities in the motion model is a function of the topology of the environment (between three in corridors to up to five at intersections).

Formally, the motion model $p(x_t \mid x_{t-1})$ is a piecewise-linear Gaussian mixture over the state space. For each state x_{t-1}, the mixture is given by the sum of the following Gaussians:

$$\begin{aligned} p(x_t \mid x_{t-1}) \quad &= \quad \sum_j \nu_j(x_{t-1}) |2\pi\Sigma|^{-1} \qquad (13) \\ &\exp\left\{ (y_t - [y_{t-1} + \Delta_j])^T \Sigma^{-1} (y_t - [y_{t-1} + \Delta_j]) \right\} \end{aligned}$$

Here j corresponds to a particular action (e.g., going straight, turning), and $\nu_j(x_{t-1})$ is the probability that the person choses this action at state x_{t-1}. If action j is chosen, the person's location will change by Δ_j in expectation, with a variance of Σ; our exact implementation adapts the velocities accordingly.

The function $\nu_j(x)$ is realized by a piecewise linear function over the locations y and the person's heading direction (discretized into N,S,W,E). The location along with the heading direction will be denoted $\bar{y}$:

$$\nu_j(x) \quad = \quad \frac{1}{\sum_l d(\bar{y}_l, \bar{y})} \sum_l d(\bar{y}_l, \bar{y}) \; b_{l,j} \tag{14}$$

The coefficients $b_{l,j}$ are the variables defining the motion model. Each $b_{l,j}$ defines the probability that a person, placed at location l with a specific heading direction, will engage in action j. The parameters $b_{l,j}$ are learned from data.

In principle, learning $b_{l,j}$ is relatively easy if we know what action a person chose. In our setting, however, we are only provided with sequences of measurements. The key insight is that this problem can be solved via the EM algorithm, along

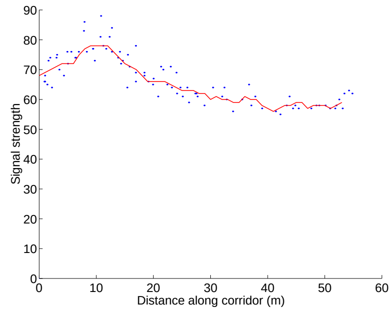

Figure 5: The wireless signal strength for one access point over three separate traversals of a straight corridor.

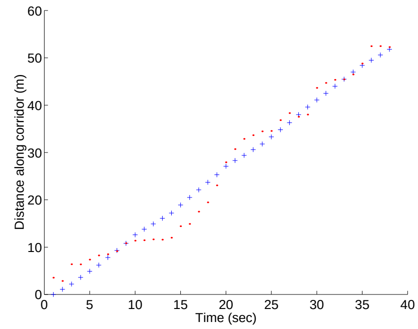

Figure 6: The crosses plot the location of a person over time, walking down a straight corridor. The dots are the corresponding locations obtained as the arithmetic mean of the particle filter. The maximum error in the run is 5.6 meters with a standard deviation of 1.5 m

the lines of the literature of learning models from unlabeled data [3; 14; 16] and particularly [2]. Since we use particle filters for tracking, generating training signals for the parameters $b_{l,j}$ can be done elegantly by counting.

In particular, assume that we are given a specific set of parameters, which define the motion model in the particular filter described in the previous section. Then, after the particle filter prediction step (Step 1 of the algorithm above), the fraction of particles generated from action j will (in expectation) match the value $\nu_j(x_{t-1})\ p(x_{t-1} \mid \theta_1, \ldots, \theta_{t-1})$. This is because the particle set at time $t-1$ represents the posterior $p(x_{t-1} \mid \theta_1, \ldots, \theta_{t-1})$, and (in expectation) $\nu_j(x_{t-1})$ of all particles are drawn according to the action j.

The posterior for action j is simply obtained by the (normalized) sum of all importance weights $w_t^{[m]}$ that correspond to action j:

$$\hat{j} \quad = \quad \frac{\sum_m w_t^{[m]} : x_t^{[m]} \text{ sampled from } j}{\sum_m w_t^{[m]}} \tag{15}$$

where "sampled from" means that action j was chosen in the generation of particle $x_t^{[m]}$. Thus, the resulting values of $\hat{j}$ are now a "training example" for the function $\nu_j(x_{t-1})$ at x_{t-1}.

The function ν is a linear interpolation of the points $b_{l,j}$. To accommodate this training example, our approach adapts the parameters $b_{l,j}$ online, and in proportion to their contribution to $\nu_j(x_{t-1})$:

$$b_{l,j} \quad \longleftarrow \quad \alpha \cdot \hat{j} + (1-\alpha) \cdot b_{l,j} \tag{16}$$

with the learning rate

$$\alpha \quad = \quad \text{lrate} \cdot \frac{d(\bar{y}_l, \bar{y}_{t-1}^{[m]})}{\sum_{l'} d(\bar{y}_{l'}, \bar{y}_{t-1}^{[m]})} \tag{17}$$

Here "lrate" is the standard learning rate (e.g., 0.01).

It is important to notice that this update rule implements an online version of EM [13], where the latent variables are the discrete action parameters j. In particular, the values $\hat{j}$ are Monte Carlo expectations of a particular action at a particular location. They correspond to the α-expectations for the hidden state in an HMM. The learning rule converges to the maximum likelihood values for the parameters b; in standard HMM learning, those maximum likelihood values are set explicitly in an offline step (the M-step of EM); here they are changed gradually using an exponential decay (learning rate) that accounts for the natural stochastic variation in the data. The convergence of the b-values is only guaranteed to a local maximum, and under other assumptions such as infinite size particle sets and appropriate cooling of the learning rate. As demonstrated in the next section, our approach yields excellent results when continually learning from unlabeled data.

4 Experimental Results

This section illustrates the major components of our approach in practice, and characterizes the accuracy of people localization as a function of the labeled and unlabeled training data. In particular, we show successful tracking for a map learned from labeled data, and subsequent improvement of the accuracy when unlabeled data is used to learn a motion model. The data used in this experiment has been acquired in an indoor environment, by a person walking with a hand-held laser range finder. Labels are only available for the labeled training data and an independent testing set; the training of the model is done using purely unlabeled data.

Our first set of results pertains to the raw data, justifying our choice of piecewise linear Gaussian model. Fig. 3 plots the signal strength of a wireless access point relative to a fixed location, accumulated over a ten minute interval. This example justifies our Gaussian noise assumption for measurements of value larger than zero, in that it indicates a Gaussian spread of the signal strength values over time. It also illustrates the occasional detection failures, which result in a measurement of zero value. Fig. 4 plots the average strength μ of an access point measured at different locations. The function is reasonably close to locally linear in the distance to the access point, until it levels off at zero. This is illustrated in Fig. 5, which plots the signal strength to an access point located at meter ten in the horizontal direction. This finding justifies our choice of a locally linear method.

Figure 7 shows an example track by the MCL localizer with a uniformly initialized motion model (that is, all parameters $b_{l,j}$ are uniform). The accuracy of this path tracker is apparent from Figure 6, which compares the ground truth (hand-labeled position) with the mean of all particles. As this diagram shows, even with an uninformed motion model the tracking along a corridor yields very good results.

Figure 8 illustrates our motion model (with uniform

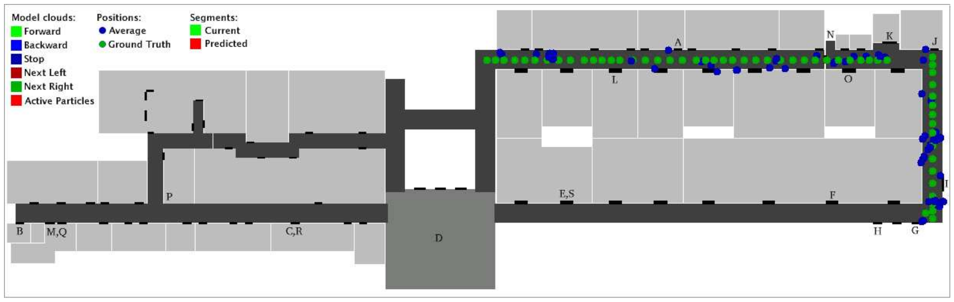

Figure 7: Trace of localized path with ground truth statistics in gray and predicted position in black.

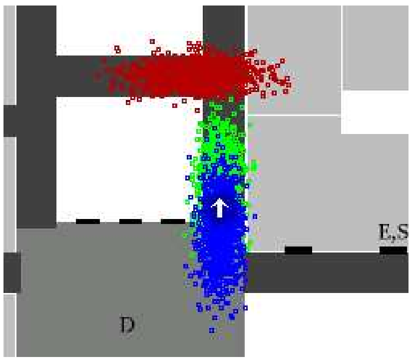

Figure 8: A sample form the motion model with uniform parameters. The posterior corresponds to nearby turns motion model distribution. Clearly visible are some of the modes induces by the different actions a person may take.

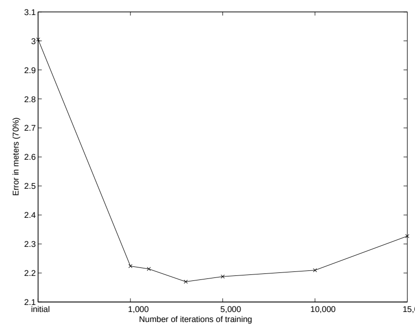

Figure 9: Average error in meters, as a function of training epochs with unlabeled data. See [10;11] for the origin of this specifi c statistic.

weights), in a situation with multiple choices (including a left turn). The particles sampled from this model correspond to multiple modes that naturally arise if people pursue different actions. Different action choices are colored differently in this plot.

In a longer term experiment aimed at evaluating our learning algoithm for the motion model, a single person walked through the environment with a handheld computing device equipped with a wireless card. In a small number of these experiments the person also carried a bulky handheld localization device using a laser range scanner. This data was used for testing. While it is difficult to guarantee that the testing data was actually drawn from the same distribution—after all, people's walking habits are somewhat subject to change—we believe that the testing data is a representative sample of the person's normal behavior.

Figure 9 illustrates the result of continual learning from the unlabeled data set, involving in total several hours worth of unlabeled training data. The error rate on the far left is obtained after training from labeled data, and constitutes the starting point of this experiment. The labeled data provides no information on the motion model, since its collection involves a sensor assembly that is relatively difficult to move. Thus, the motion model is initialized uniformly. As the motion model is being trained, the localization error gradually decreases through an improved ability to predict. This specific motion model is trained specific to a single person. Figure 9 shows the effect of learning on the individual sampling process. Shown there is the average error for 70% of all cases, a statistics prevously reported in [10; 11] (see this reference for a motivation of a 70% statistics). As this graph shows, the initial error of 3.0 meters is gradually reduced to approximately 2.1 meters. As training in unlabeled data progresses, the error increases somewhat, which is probably an effect of overfitting or of a different motion pattern that occurred when collecting the testing data. Nevertheless, we believe this is strong evidence that the unlabeled data indeed improves the performance of the localizer.

We conducted further experiments to evaluate how good the predictions are of the trained motion models. In general, the goodness of the prediction can be assessed by the probability $p(\theta_t \mid \theta_1, \ldots, \theta_{t-1})$, which describes the likelihood of the t-th measurement under the state prediction of the model (and convolved with the previous belief). The more likely a measurement under the prediction generated by the motion model, hence, the better the prediction. In particle filtering, this probability is measured in a Monte Carlo fashion by the importance weights. Fig. 10 shows the logarithm of the sum of the weights; These histograms illustrate the goodness of the learned motion model for predicting state. As is easily seen from this diagram, the average value of this histogram drifts closer and closer to zero as learning on the unlabeled data progresses, indicating the the predictions are improving in quality.

Figure 11 shows successful particle filtering tracking using the learned motion model. It shows a sequence of tracking from global uncertainty to local uncertainty.

5 Discussion

We have presented a technique for learning the parameters of a people localizer based on the signal strength of IEEE 802.11b

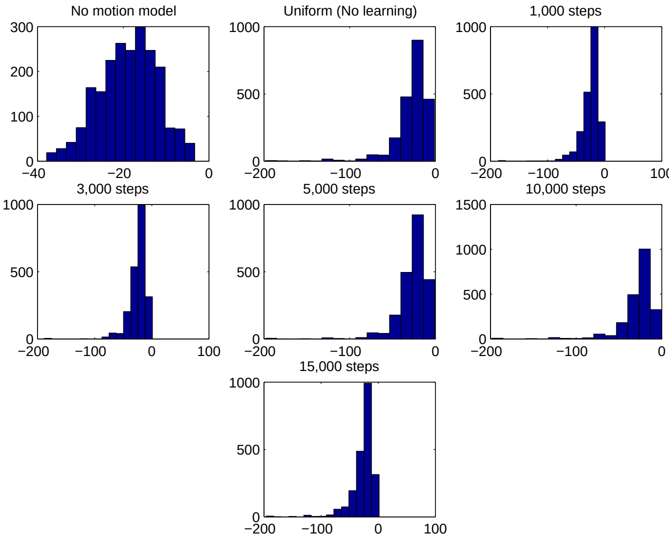

Figure 10: Histograms of the log of the sum of the weights for 5,000 particles, at various points in time of the continual learning process from unlabeled data.

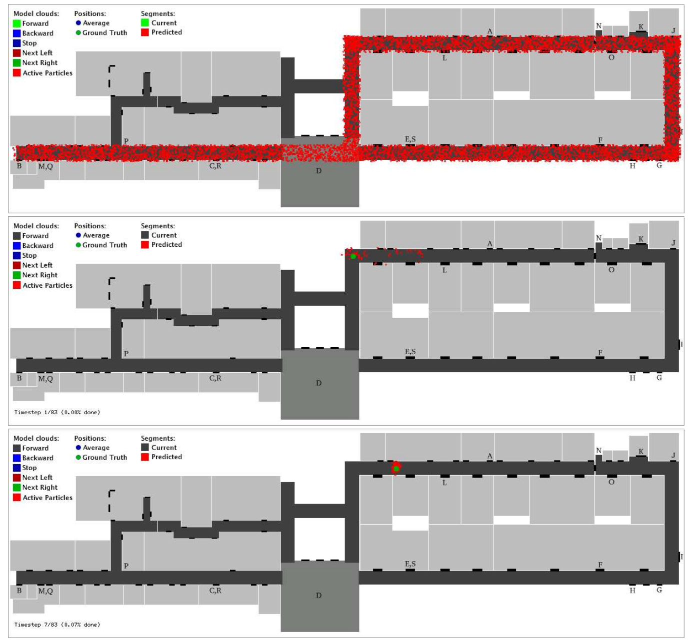

Figure 11: Tracking using the learned motion model: From global to local uncertainty. Global localization is trivial since wireless access point carry a unique ID.

wireless networks. Our approach employs a version of the Monte Carlo localization algorithm for tracking people. The algorithm requires two sets of parameters: those pertaining to the sensor model (in particular a signal strength map), and those pertaining to the motion model (a distribution over different modes in a Gaussian mixture). The former are learned offline from data labeled by the correct position, whereas the latter are learned online from unlabeled data, using an online version of EM to continually adapt to individual motion patterns.

The research raises several open questions that warrant further research. For example, unlabeled data is excursively used to tune the motion model; however, it could equally be used to further refine the signal strength map. Such an approach, however, would require a parameter trading off the strength of labeled vs unlabeled data, to avoid that the information of the labeled data vanishes over time. Furthermore, the approach does not model hidden state other than people's position. Experience suggests that factors such as the height at which a receiver is carried may influence signal strength.

Regardless of these limitations, we have demonstrated that learning individual motion models can significantly improve the localization accuracy for people localization. We believe that our approach is unique in its ability to learn models that improve continually, without a necessity to provide labeled data. As a secondary result, this paper has shown an effective way to learn models for people localization, and devised an extension of MCL suitable for tracking people using wireless data.

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[2] M. Bennewitz, W. Burgard, and S. Thrun. Using EM to learn motion behaviors of persons with mobile robots. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.

[3] V. Castelli and T. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16:105–111, January 1995.

[4] H. Choset and J.W. Burdick. Sensor Based Planning: The Hierarchical Generalized Voronoi Graph. In *Proc. Workshop on Algorithmic Foundations of Robotics*, Toulouse, France, 1996.

[5] P.K. Chrysanthis. On indoor position location with wireless lans. In *Proceedings of the IEEE International Symposium on Personal, Indoor, and Radio Communications*, Lisbon, Portugal, 2002.

[6] A.P. Dempster, A.N. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[7] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer Verlag, New York, 2001.

[8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[9] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: Towards distraction-free pervasive computing. *IEEE Pervasive Computing*, 21(2):22–31, 2002.

[10] A. M. Ladd, K.E. Bekris, G. Marceau, A. Rudys, L. Kavraki, and D. Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, 2002.

[11] A. M. Ladd, K.E. Bekris, G. Marceau, A. Rudys, D. Wallach, and L. Kavraki. Using wireless ethernet for localization. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.

[12] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, New York, 1997.

[13] R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifi es incremental, sparse, and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic Press, 1998.

[14] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. *Machine Learning*, 39(2/3):1–32, 1998.

[15] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[16] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing*, 32(5):1087–1095, Sept 1994.

[17] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of IJCAI-95*, pages 1080–1087, Montreal, Canada, August 1995. IJCAI, Inc.

[18] J. Small, A. Smailagic, and D. Siewiorek. Determining user location for context aware computing through the use of a wireless LAN infrastructure. Submitted for publicaitons, see http://www.cs.cmu.edu/∼aura/docdir/small00.pdf., 2003.

[19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[20] M. Wallbaum. Wheremops: An indoor geolocation system. In *Proceedings of the IEEE International Symposium on Personal, Indoor, and Radio Communications*, Lisbon, Portugal, 2002.

[21] Kamin Whitehouse and David Culler. Calibration as parameter estimation in sensor networks. In *Proceedings of ACM First workshop on Wireless Sensor Networks and Applications*, 2002.