Gaussian Processes for Signal Strength-Based Location Estimation

Brian Ferris **Dirk Hähnel**[†] **Dieter Fox**
University of Washington, Department of Computer Science & Engineering, Seattle, WA
[†]Intel Research Seattle, Seattle, WA

***Abstract*— Estimating the location of a mobile device or a robot from wireless signal strength has become an area of highly active research. The key problem in this context stems from the complexity of how signals propagate through space, especially in the presence of obstacles such as buildings, walls or people. In this paper we show how Gaussian processes can be used to generate a likelihood model for signal strength measurements. We also show how parameters of the model, such as signal noise and spatial correlation between measurements, can be learned from data via hyperparameter estimation. Experiments using WiFi indoor data and GSM cellphone connectivity demonstrate the superior performance of our approach.**

I. Introduction

Over the last years, the use of wireless signal strength information to localize mobile devices or robots has gained significant interest in several research communities. This is mainly due to the increasing availability of 802.11 WiFi networks and the importance of location information for applications such as activity recognition, surveillance, and context-aware computing.

What distinguishes location estimation using wireless signal strength from many robotics localization problems is the unpredictability of signal propagation through indoor environments. This unpredictability makes it difficult to generate an adequate likelihood model of signal strength measurements. Thus, the main focus of research in this area has been on the development of techniques that can generate such models from small amounts of calibration data collected in an environment. Existing approaches to signal strength localization fall into two main categories. The first class of techniques assume knowledge about the locations of access points and then model the propagation of signals through space to determine the *expected* signal strength at any location based on the distance from an access point [15], [1], [8]. Unfortunately, these parametric models have only limited accuracy, even when taking information about the locations of walls and furniture into account. The second class of techniques compute measurement likelihoods using location-specific statistics extracted from calibration data. These local statistics include histograms [7], Gaussians [5], [4], [10], or even raw measurements [1]. Local approaches typically result in higher localization accuracy if enough calibration data is available. A main problem for these models, however, is to generate likelihoods at locations for which no calibration data is available. While several groups have addressed this problem via spatial smoothing [10], [5], [8], existing techniques have important limitations with respect to considering all available information in a statistically sound way.

In this paper we show how Gaussian processes (GP) [13] can be used to overcome these limitations. GPs are non-parametric models that estimate Gaussian distributions over *functions* based on training data. GP regression has been used with great success in a variety of applications, including sensor networks [3], data visualization [9], and computer animation [12]. GPs have several properties that make them ideally suited for modeling signal strength measurements:

Continuous locations: GPs do not require a discretized representation of an environment, or the collection of calibration data at pre-specified locations. They are able to predict signal strength measurements at arbitrary locations.

Arbitrary likelihood models: GPs are non-parametric regression models and thereby able to approximate an extremely wide range of non-linear signal propagation models.

Correct uncertainty handling: In contrast to other regression models, GPs provide uncertainty estimates for predictions at any set of locations. This uncertainty takes into account the local density of calibration data and the noise of the data points.

Consistent parameter estimation: The parameters of GPs can be learned from the calibration data via hyperparameter estimation. These parameters include the spatial correlation between measurements and the measurement noise.

The use of GPs for signal strength based location estimation has been proposed by Schwaighofer and colleagues [14]. In this paper we extend their work in several directions. More specifically, we introduce a Bayesian filter for location estimation that builds on a mixed graph / free space representation of indoor environments. While hallways, stair cases, and elevators are represented by edges in a graph, areas such as rooms are represented by bounded polygons. Using this representation we can model both constrained motion such as moving down a hallway, or going upstairs, and less constrained motion through rooms and open spaces. The likelihood of signal strength measurements is extracted from a GP that is learned from calibration data. In contrast to existing approaches, our technique explicitly models the probability of *not* detecting an access point, which can greatly increase the quality of the global localization process.

In our experiments we demonstrate various features of GPs for signal strength localization. We also show that the same technique can applied to model GSM cellphone connectivity, which results in significant improvements over existing outdoor localization techniques.

This paper is organized as follows. In the next section, we will give an overview of GPs and show how they can be used to model signal strength measurements. Then, in Section III, we will introduce our mixed graph / free space representation of indoor environments and describe a particle filter for location estimation in such models. Related work will be discussed in Section IV, followed by experimental results. We conclude in Section VI.

II. Gaussian Processes for Modeling Signal Strength Measurements

We perform Bayesian filtering to estimate the location of a person from signal strength measurements. A key component of a Bayes filter is the observation model, which describes the likelihood of making an observation at the different locations in an environment [16]. Before we discuss the specifics of our approach to localization, we will show how Gaussian processes can be used to generate an observation model for signal strength measurements from calibration data.

A. Preliminaries

GPs can be derived in different ways. Here, we follow closely the function-space view described in [13]. Let $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ be a set of training samples drawn from a noisy process

$$y_i \;=\; f(\mathbf{x}_i) + \varepsilon, \tag{1}$$

where each $\mathbf{x}_i$ is an input sample in $\mathbb{R}^d$ and each y_i is a target value, or observation, in $\mathbb{R}$. ε is zero mean, additive Gaussian noise with known variance σ_n^2. For notational convenience, we aggregate the n input vectors $\mathbf{x}_i$ into a $d \times n$ matrix $\mathbf{X}$, and the target values y_i into the vector denoted $\mathbf{y}$.

A Gaussian process estimates posterior distributions over functions f from training data D. These distributions are represented non-parametrically, in terms of the training points. A key idea underlying GPs is the requirement that the function values at different points are correlated, where the covariance between two function values, $f(\mathbf{x}_p)$ and $f(\mathbf{x}_q)$, depends on the input values, $\mathbf{x}_p$ and $\mathbf{x}_q$. This dependency can be specified via an arbitrary covariance function, or kernel $k(\mathbf{x}_p, \mathbf{x}_q)$. The choice of the kernel function is typically left to the user, the most widely used being the squared exponential, or Gaussian, kernel:

$$k(\mathbf{x}_p, \mathbf{x}_q) \;=\; \sigma_f^2 \exp\left(-\frac{1}{2l^2}|\mathbf{x}_p - \mathbf{x}_q|^2\right) \tag{2}$$

Here, σ_f^2 is the signal variance and l is a length scale that determines how strongly the correlation between points drops off. Both parameters control the smoothness of the functions estimated by a GP. We will show in Section II-C how these values can be learned from training data. As can be seen in (2), the covariance between function values decreases with the distance between their corresponding input values.

Since we do not have direct access to the function values, but only noisy observations thereof, it is necessary to represent the corresponding covariance function for noisy observations:

$$\operatorname{cov}(y_p, y_q) \;=\; k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \tag{3}$$

Here σ_n^2 is the Gaussian observation noise and δ_{pq} is one if $p = q$ and zero otherwise. For an entire set of input values $\mathbf{X}$, the covariance over the corresponding observations $\mathbf{y}$ becomes

$$\operatorname{cov}(\mathbf{y}) \;=\; K + \sigma_n^2 I, \tag{4}$$

where K is the $n \times n$ covariance matrix of the input values, that is, $K[p, q] = k(\mathbf{x}_p, \mathbf{x}_q)$.

Note that (4) represents a prior over functions: For any set of values $\mathbf{X}$, one can generate the matrix K and then sample a set of corresponding targets $\mathbf{y}$ that have the desired covariance [13]. The sampled values are jointly Gaussian with $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, K + \sigma_n^2 I)$. More relevant, however, is the posterior distribution over functions given training data $\mathbf{X}, \mathbf{y}$. Here, we are interested in predicting the function value at an arbitrary point $\mathbf{x}^*$, conditioned on training data $\mathbf{X}, \mathbf{y}$. From (2) follows that the posterior over function values is Gaussian with mean $\mu_{\mathbf{x}_*}$ and variance $\sigma^2_{\mathbf{x}_*}$:

$$p\left(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) \;=\; \mathcal{N}\left(f(\mathbf{x}_*); \mu_{\mathbf{x}_*}, \sigma^2_{\mathbf{x}_*}\right), \text{where}$$

$$\mu_{\mathbf{x}_*} \;=\; {\mathbf{k}_*}^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{y} \tag{5}$$

$$\sigma^2_{\mathbf{x}_*} \;=\; k(\mathbf{x}_*, \mathbf{x}_*) - {\mathbf{k}_*}^T \left(K + \sigma_n^2 I\right)^{-1} \mathbf{k}_* \tag{6}$$

Here $\mathbf{k}_*$ is the $n \times 1$ vector of covariances between $\mathbf{x}_*$ and the n training inputs $\mathbf{X}$, and K is the covariance matrix of the inputs $\mathbf{X}$. As can be seen from (5), the mean function is a linear combination of the training observations $\mathbf{y}$, where the weight of each observation is directly related to $\mathbf{k}_*$, the correlation between the test point $\mathbf{x}_*$ and the corresponding training input. The middle term is the inverse of the covariance function (4). The covariance of the function estimate, $\sigma^2_{\mathbf{x}_*}$, is given by the prior covariance, $k(\mathbf{x}_*, \mathbf{x}_*)$, minus the information provided by the training data (via the inverse of the covariance matrix K). Note that the covariance is independent of the observed values $\mathbf{y}$.

The predictive distribution in (5) and (6) summarizes the key advantages of GPs for signal strength likelihood models. In addition to providing a regression model based on training data, the GP also represents the uncertainty at any location.

B. Application to Signal Strength Modeling

In the context of signal strength localization, the input values $\mathbf{X}$ correspond to locations, and the observations $\mathbf{y}$ correspond to signal strength measurements obtained at these locations. The GP posterior is estimated from a calibration trace of signal strength measurements annotated with their locations. Assuming independence between different access points, we estimate a GP for each access point separately. During localization, the likelihood of observing a measurement can then be computed at any location using (5) and (6).

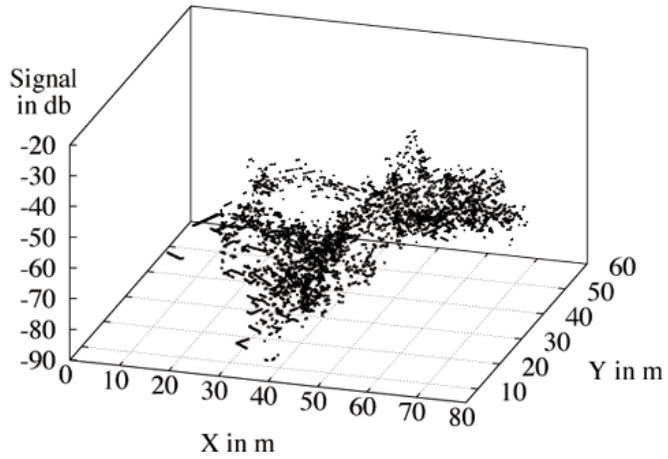

Fig. 1. Raw signal strength measurements for one access point

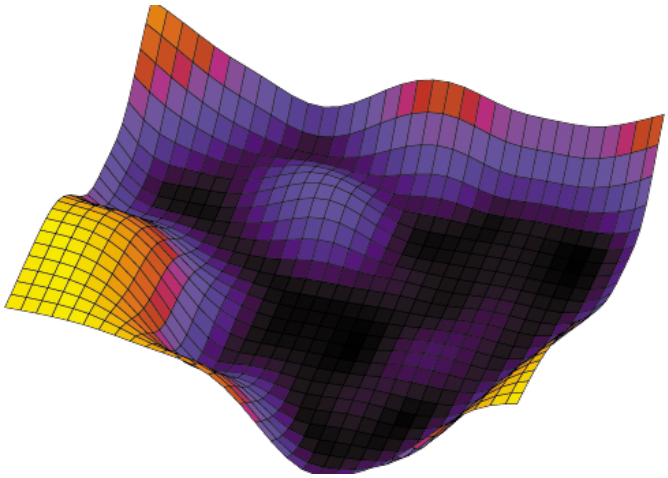

Fig. 3. Variance of GP prediction for one access point

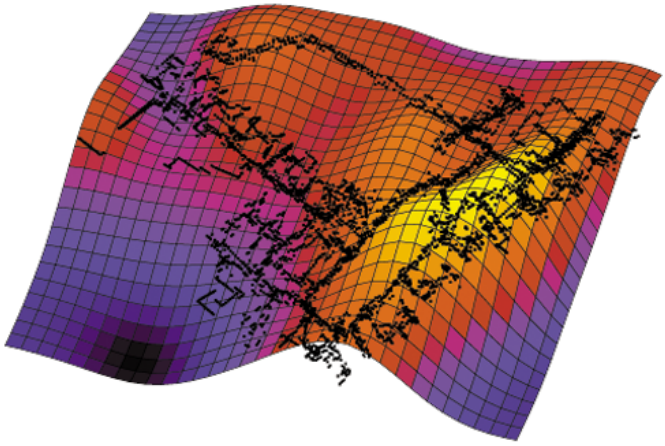

Fig. 2. Mean of GP prediction for one access point

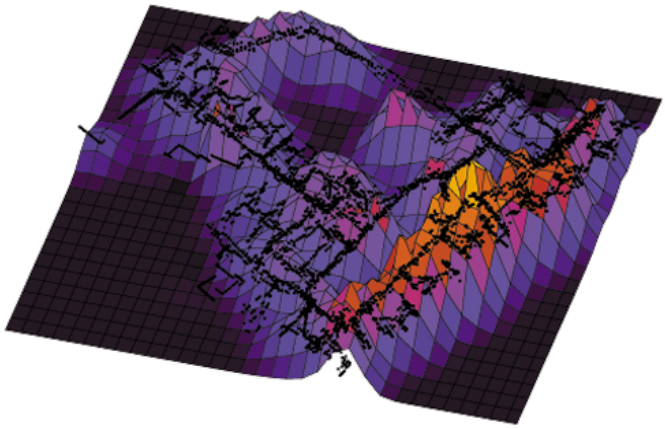

Fig. 4. Mean of GP prediction with different covariance parameters

Fig. 1 illustrates the GP signal strength model for one access point on one floor of our test environment. The raw signal strength measurements are shown in the upper left panel. The size of the area covered by these measurements is 60×50 meters. Obviously, these measurements can not be represented adequately by a radial signal propagation model. The mean and variance of the GP posterior for these data points are shown in Fig. 2 and Fig. 3, respectively. As can be seen, the GP smoothly approximates the data points. The variance of the prediction increases in areas that are not covered by the data. The gap in the middle of the data, generating the "bump" in the variance function, corresponds to a large, inaccessible atrium.

This interpolation was achieved with values 17.8 and 8.2 for the parameters l and σ_f^2 of the covariance function k defined in (2), and a signal noise σ_n^2 of 4.0. The impact of these parameters is illustrated by Fig. 4, which shows the GP mean values for the same data when using 17.8, 2.0 and 2.0 as parameter values. Obviously, it is extremely important to determine adequate parameter values,

C. Hyperparameter Estimation

Fortunately, it is possible to learn these parameters based on the training data $\mathbf{X}, \mathbf{y}$ using hyperparameter estimation. More specifically, we estimate the values of these parameters by maximizing the log likelihood of the observations $\mathbf{y}$. Let $\theta = \langle \sigma_n^2, l, \sigma_f^2 \rangle$ denote the hyperparameters we wish to estimate. The log likelihood of the observations is given by [13]

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = \\ -\frac{1}{2}\mathbf{y}^T(K + \sigma_n^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi, \tag{7}$$

which follows directly from the fact that the observations are jointly Gaussian. (7) can be maximized using conjugate gradient descent (LBFGS). To do so, we need to compute the partial derivatives of the log likelihood.

$$\frac{\partial}{\partial \theta_j}\log p(\mathbf{y} \mid \mathbf{X}, \theta) = \frac{1}{2}\mathrm{tr}\left((K^{-1}\mathbf{y})(K^{-1}\mathbf{y})^T\frac{\partial K}{\partial \theta_j}\right). \tag{8}$$

We now consider the subsequent partial derivatives of the kernel function with respect to the kernel parameters. Con-

sider, as an example, the Gaussian kernel function. The partial derivatives of each element $K[p,q]$ follow as

$$\frac{\partial K}{\partial \sigma_f^2} = 2\sigma_f \exp\left(-\frac{1}{2}\left(\frac{d}{l}\right)^2\right) \quad (9)$$

$$\frac{\partial K}{\partial l} = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{d}{l}\right)^2\right)\frac{d^2}{l^3} \quad (10)$$

$$\frac{\partial K}{\partial \sigma_n^2} = 2\sigma_n \delta_{pq}, \quad (11)$$

where $d = x_p - x_q$.

The most computationally complex step in the hyperparameter estimation is the inversion of the covariance matrix K in (8), which takes time $O(n^3)$, where n is the number of training points. This inversion must be performed with each new value θ, so an efficient gradient descent algorithm is key for tractable optimization.

D. Zero Mean Offset

A Gaussian process is, by default, a zero mean process. In absence of training data, the process tends to zero. For simple data relations, the mean of the data can be subtracted before training such that the process is centered around the mean. However for complex data relations, a more nuanced approach is required.

Modeling WiFi signal strength propagation is such a case where the zero-mean is an issue. When far enough from the access point, all readings should tend to zero. However, if we have a large region near the access point without training data, we would like the model not to tend to zero completely.

For WiFi, we assume a very simple offset model where signal strength decreases linearly with distance from the access point. Such a model takes the following form:

$$ss = m||x - x_{AP}|| + b \quad (12)$$

where x is the input point, x_{AP} is the location of the access point, $||x - x_{AP}||$ is the distance between the input and access point, m is the propagation slope, b is the signal strength recorded at the access point, and ss is the resulting signal strength prediction. We estimate the value of the parameters m, b, and x_{AP} by minimizing the difference between ss and actual training values with resepect to the parameters using conjugate gradient descent. Clearly, real world data will deviate from this simple model, but in practice the simple model offers an improvement when confronted with sparse training data.

III. Bayesian Filtering on Mixed Graph / Free Space Representations

The goal of Bayesian localization is to estimate the posterior over a person's location, x_t, conditioned on all sensor measurements, $z_{1:t}$, obtained through time t. At the core of each Bayes filter is the following recursive equation, which is updated whenever new sensor information becomes available [16]:

$$p(x_t|z_{1:t}) \propto p(z_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})\,dx_{t-1} \quad (13)$$

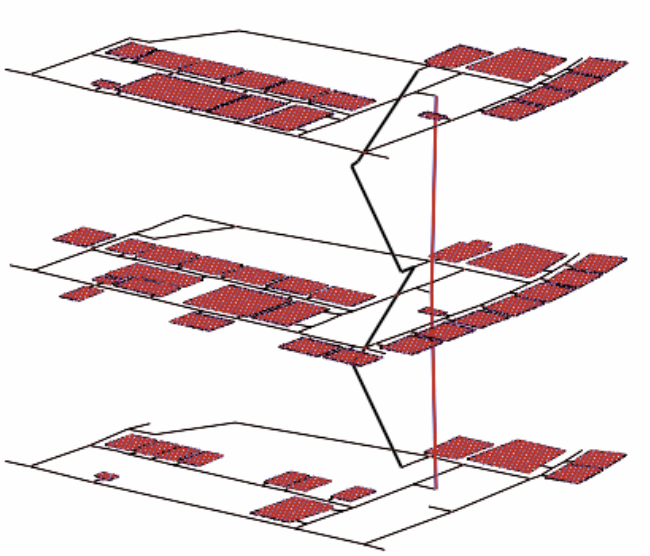

Fig. 5. Mixed representation of part of an indoor environment. Hallways, stair cases, and elevators are modeled as edges on a connectivity graph. Rooms and break-out areas are modeled as bounded free space areas.

Here, we have the special case that no control information u is available. The term $p(x_t|x_{t-1})$ represents the motion model, which we will describe in more detail after discussing our spatial representation. The term $p(z_t|x_t)$ is the measurement likelihood model, which in our case describes the likelihood of observing a set of signal strength measurements z_t at a location x_t. As described in Section II-B, we use a Gaussian process to generate this likelihood. As is done typically for such types of sensors, we compute the likelihood of a complete set of readings by multiplying the individual reading likelihoods [16]. However, since the GP models were learned independently of each other, the resulting likelihood can become highly peaked, which results in overconfident estimates. We take this approximation into account by "smoothing" the likelihood model:

$$p(z_{t[1:n]}|x_t) = \left(\prod_{i=1}^{n} p(z_{t[i]} \mid x_t)\right)^{\gamma} \quad (14)$$

Here, n is the number of detected access points and $\gamma \in [0:1]$ plays the role of a smoothing coefficient [6]. In our experiments we set γ to $1/n$, resulting in the geometric mean of the individual likelihoods.

A. Mixed Graph / Free Space Representation

Our representation of a person's locations is motivated by the Voronoi motion graphs introduced by Liao and colleagues [11]. The key idea of their approach is to represent indoor environments by graphs whose edges correspond to the Voronoi graph of an environment. Liao *et al.* showed that by constraining a person's location and motion to edges on such a graph, their system is able to adequately represent typical motion patterns through indoor environments; resulting in improved tracking and learning performance.

While such constraints are adequate for hallway environments, they are not well-suited to model a person's motion through open spaces such as rooms or laboratories. We overcome this limitation by introducing a mixed graph / free space representation of indoor environments. While our novel representation can be applied to both indoor and outdoor environments, the focus of this paper is on indoor localization. In outdoor environments, edges would correspond to streets and walkways, and open spaces would correspond to parks or parking lots.

Our representation is an enhanced graph structure $G = (E, R, V)$, where E is a set of undirected edges e_i that correspond to hallways, stair cases, and elevators; the set R contains polygonal regions r_i that represent open spaces such as rooms and break-out areas; and V are vertices v_i that connect edges and regions. The vertices play an important role in the motion model of our tracking algorithm since they correspond to *choice points*, which are locations where a person has a discrete number of choices as to where to move next. A representation of three floors of our test environment is shown in Fig. 5. While the lines indicate hallways, elevators, and a stair case, the shaded regions represent rooms and break-out areas.

B. Particle Filter-Based Tracking

We implement Bayesian filtering in our representation using particle filters, which represent and propagate posteriors using sets $S_t = \{\langle x_t^{(i)}, w^{(i)_t}\rangle | i = 1, \ldots, n\}$ of weighted samples [2]. Each sample $x_t^{(i)}$ is a potential location of the person, and each has an associated importance weight $w_t^{(i)}$. Standard particle filters realize Bayes filter updates by propagating samples through time according to the following sampling procedure: *Re-sampling:* Draw with replacement a random sample $x_{t-1}^{(i)}$ from the previous sample set according to the importance weights $w_{t-1}^{(i)}$. *Sampling:* Generate a new particle $x_t^{(j)}$ by sampling from the motion model $p(x_t^{(j)} \mid x_{t-1}^{(i)})$. *Importance sampling:* Weight the sample by the measurement likelihood $p(z_t \mid x_t^{(j)})$.

When using a particle filter for signal strength localization, the state x_t represents a person's location inside a building. The incorporation of the Gaussian process likelihood model is straightforward; it only requires the evaluation of (5) and (6) at the corresponding sample location. In addition to the location in the global reference frame of a building, each particle contains information that enables us to relate the person's location to the enhanced graph structure of our mixed representation. More specifically, each state is represented as

$$x_t = \begin{array}{ll} \langle e_t, d_t, m_t \rangle & \text{if location is on edge} \\ \langle r_t, x_t, y_t, \alpha_t, m_t \rangle & \text{if location is in region,} \end{array} \quad (15)$$

where e_t is an edge identifier, d_t indicates the distance from the start of the edge, and $m_t \in \{\texttt{stopped}, \texttt{moving}\}$ indicates the current motion state. Furthermore, r_t denotes a region and x_t, y_t, α_t represent the person's location and heading within the region. The motion update of the particle filter requires sampling from the motion model $p(x_t^{(j)} \mid x_{t-1}^{(i)})$. To define the motion model for our enhanced graph structure, we need to incorporate the following:

Motion state transitions $p(m_t \mid m_{t-1})$ represent the probability of motion state m_t being `moving` or `stopped` given the previous motion state. This 2×2 matrix models a preference of staying in the previous state, thereby avoiding too rapid switching between motion states. Furthermore, our system uses two different motion state transition matrices, one for particles on edges and one for particles in regions. This enables the system to model the fact that a person is far more likely to stop when being in a room versus a hallway.

Edge transitions $p(e_t \mid e_{t-1})$ are stored at each vertex of the graph. They represent preferences when moving through the graph structure. For instance, when reaching a vertex in a hallway, the probability of choosing the next edge along the hallway is higher than the probability of entering an edge that leads to a room. The graph also contains special vertices that connect an edge to a region. Whenever such a vertex is reached from an edge, then the particle enters the region with probability one, and vice versa.

Free space motion is applied to particles in regions. We use a rather simplistic motion model that prefers straight motion when the person is in the `moving` mode and allows arbitrary rotations when the person is in the `stopped` motion mode. Whenever a particle reaches the boundary of a region, the particle is forced to stay in the region by reversing its heading direction. The only way to exit a region is via one of the vertices that connect the region to an edge. In our model, the probability of "hopping" onto such a vertex is inverse proportional to the distance from the vertex.

Sampling from the resulting motion model is done as follows. If $x_{t-1}^{(i)} = \langle e_{t-1}, d_{t-1}, m_{t-1} \rangle$ is on an edge in the graph, then we proceed similar to Liao *et al.* [11]: We first sample the discrete motion state m_t with probability proportional to $p(m_t \mid m_{t-1})$. If $m_t = \texttt{stopped}$, then x_t is set to be x_{t-1}. Otherwise, if $m_t = \texttt{moving}$, then we randomly draw a motion distance d according to a Gaussian velocity distribution. For this distance d, we determine whether the motion along the edge results in a transition over the end vertex of e_{t-1}. If not, then $d_t = d_{t-1} + d$ and $e_t = e_{t-1}$. Otherwise, if the end vertex is connected to other edges, then we set $d_t = d_{t-1} + d - |e_{t-1}|$ and the next edge e_t is sampled with probability $p(e_t \mid e_{t-1})$. If the end vertex is connected to a region r_t, then the next state is initialized with random heading α_t and with location x_t, y_t within this region, drawn from a Gaussian with mean at the entry vertex.

If $x_{t-1}^{(i)} = \langle r_{t-1}, x_{t-1}, y_{t-1}, \alpha_{t-1}, m_{t-1} \rangle$ was already in a region, then we first sample whether or not the particle exits the region. This sampling is done inverse proportional to the distance between $\langle x_{t-1}, y_{t-1} \rangle$ and the closest vertex connected to the region. If the particle exits the region, then its location is initialized at the start of the edge connected

to the corresponding vertex. Otherwise, we first sample the motion state m_t and corresponding motion distance d. The new position $\langle x_t, y_t \rangle$ is then determined based on a straight motion starting at $\langle x_{t-1}, y_{t-1} \rangle$ in direction α_{t-1}. If the motion state is `moving`, then α_t is sampled from a Gaussian with mean at α_{t-1}, otherwise, α_t is sampled uniformly from $[0 : 2\pi]$.

IV. Related Work

Several location estimation techniques model signal strength measurements by their propagation through space [15], [1], [8]. They assume an exponential attenuation model for wireless signals, and use this path loss to determine likelihoods based upon distance from an access point, whose location is assumed known. [15], [1] showed how information about the location and material of walls and furniture inside buildings can be used to better estimate path loss. Even with such information, however, the accuracy of signal propagation models is limited due to the inherent unpredictability of how signals propagate through indoor environments.

Alternative techniques ignore signal attenuation and instead compute likelihoods from location-specific statistics compiled from training data. While such techniques require more training data, they are able to represent arbitrary likelihood models, which typically results in better localization performance. In order to generate a probabilistic likelihood model, Ladd and colleagues [7] used histograms over measurements collected at a fixed set of locations in an office environment. They later showed that replacing the histograms by Gaussians requires smaller training sets and results in better localization performance [4]. Howard and colleagues [5] show how spatial smoothing on a discrete grid of points can significantly improve the quality of a sensor model, especially when the training data contains gaps. However, they do not show how to estimate model parameters, and their technique does not estimate the uncertainty in the measurement prediction, which is crucial for adequate likelihood models. Recently, Letchner *et al.* [10] introduced a hierarchical Bayesian technique that incorporates a signal propagation model via hyperparameters in order to estimate Gaussian likelihoods on a grid. An important aspect of this method is that the prediction certainty takes number of training points into account. However, the spatial smoothing of this technique does not correlate the signal strengths measured at neighboring locations.

In contrast to our approach, all these existing techniques rely on a pre-specified set of discrete locations; they are not able to adequately incorporate training data collected at arbitrary, continuous locations. Furthermore, none of these approaches is able to interpolate between data points while correctly estimating the uncertainty resulting from the interpolation. Our approach, on the other hand, is able to naturally interpolate between continuous data points even in 3D environments, while still being able to estimate the resulting uncertainties in predictions.

In [14], Schwaighofer and colleagues showed how to apply Gaussian processes to modeling signal strength measurements. They achieved 10m location accuracy based on DECT wireless phone connectivity, without performing any temporal integration of sensor information. Our work goes beyond their technique in several aspects: We show the applicability of GPs for GSM connectivity and WiFi based localization in large scale, structured environments. To do so, we introduce a novel Bayesian filter for location estimation that builds on a mixed graph / free space representation of indoor environments. This representation combines the advantages of graph-based tracking [11] with the flexibility of modeling arbitrary paths through free space.

V. Experimental Results

In our experiments we evaluate Gaussian processes for signal strength localization using WiFi indoor data and GSM connectivity data.

A. Setup of Indoor Experiments

Our test environment consists of the three floors shown in Fig. 5. To collect calibration data, we used an iPAQ hx4705 PDA with a built in wireless device polling WiFi signal strength every 0.5 seconds. The ground truth locations were estimated based on manual annotation of waypoints using the iPAQ during data collection. The path was then estimated based on linear interpolation between these waypoints, thus assuming constant velocity.

The calibration data was collected during one hour of walking through the environment, covering all rooms, hallways, elevators, and stair cases shown in Fig. 5. All told, the data referenced 75 unique access points, visiting 54 rooms. The test data consisted of one hour of trace data, covering about 3 km of travel distance and spread across ten distinct traces. This data was collected during different times within two days. During test data collection, the person used the elevators and stair cases, moving through 30 different rooms, resulting in a total of 47 room visits. The ordering of rooms visited was generated by a random ordering of available rooms.

To learn the hyperparameters of the GP, we randomly sampled 300 data points for each access point. We then used the gradient descent technique described in Section II-C to find the global parameter settings that minimized the negative log-likelihood of the training data of *all* access points. To avoid local minima, we used randomly selected start values over multiple iterations. This learning process took typically less than one hour on a standard desktop PC. A typical sensor model generated with the trained hyperparameters is shown in Fig. 2.

Once learning converged, we used the trained hyperparameters to generate the GP model. For this purpose, we randomly drew 700 samples from the training data of each access point and computed the $\left(K + \sigma_n^2 I\right)^{-1} \mathbf{y}$ term used for the mean and variance of the likelihood model given in (5) and (6). This step, which was dominated by the inversion of the 700×700 covariance matrix, took typically 30 minutes for the entire set of access points.

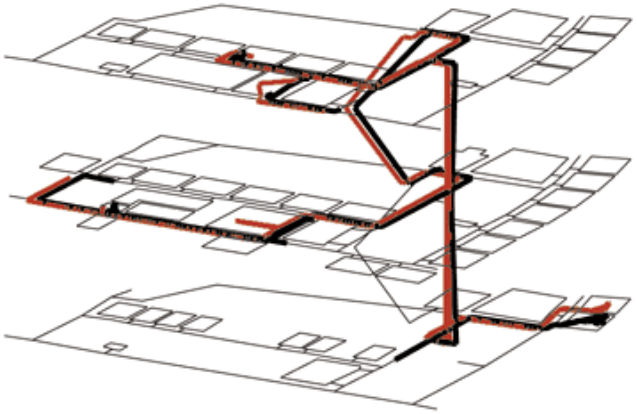

Fig. 6. Ground truth path (red / grey) and most likely particle path estimate (black) for one of the test traces.

For localization, we used a particle filter with 200 particles. At every update of the filter, the likelihood of each signal strength measurement was computed for each particle by evaluating the GP for that particle's location. The complexity of this update is $O(nm)$, where m is the number of particles and n is the number of calibration points. The particle filter ran in real time on a standard PC.

B. Indoor WiFi Localization Accuracy

To evaluate the accuracy of our localization algorithm, we compared at each iteration the particle with the highest weight to the ground truth position. The average error over the 3 km of test data was 2.12 meters. We additionally compared the complete trajectory of the most likely particle at the end of each run to the ground truth locations. One of these paths is shown in Fig. 6. The error of the most likely trajectories was only 1.69 meters on average. We believe that these error values are among the best reported in the literature. They were achieved under extremely challenging localization conditions: the person moved constantly through the building; entering rooms, taking stairs and elevators.

In order to assess the quality of the localization process on a more qualitative scale, we also evaluated the topological correctness of the path estimated by the most likely particle. These results are summarized in the following table:

	% correct room	% wrong room	% hallway
Ground truth in room	81	17	2
Estimate in room	83	14	3

TABLE I

The first row evaluates the accuracy when the person was actually in a room against the path prediction of the most likely particle. As can be seen, the system confuses a room with its neighboring rooms and hallways in less than 20% of the time spent in rooms. The second row evaluates the accuracy when the most likely particle path is in a room against the ground truth location. Again, the error rate for the particle is less than 20%. Note that further smoothing could be applied as appropriate to regularlize discontinuities in the location trace.

We additionally evaluated the sequence of rooms visited by the most likely particle path during the test traces. We compared this sequence against the ground truth room sequence with a string edit distance. Specifically, we consider the number of additions or deletions of rooms to match ground truth. Over our ten evaluation traces, we had a total edit distance of only 10, suggesting that our path misclassifies approximately one room per trace; either visiting a room that was not in the ground truth sequence, or missing a room that was actually visited.

C. Dealing with Sparse Data

To evaluate the ability of GPs to deal with sparse data, we removed the training data collected in 25 out of the 54 rooms (the test traces visited 10 of these rooms). We then performed the same localization experiments as done for the complete training data. In all but one of the 10 test traces, the accuracy was virtually indistinguishable from the results achieved with the complete data. In only one of the 10 traces did the filter lose track, resulting in a path error of 16m.

These results show that the GP is able to accurately extrapolate the signal strength model into rooms for which no training data is available at all, especially in combination with a simple zero mean offset model. We have not seen any reports of such a capability in the literature.

D. GSM Localization

The second experiment is a wide-area localization using GSM signal information. For the experiment we collected data using a standard GPS unit (Sirf III) and an Audiovox SMT5600 mobile phone. The mobile phone is able to collect information of the connected cell tower as well as the up to eight neighboring cells. The information includes a unique identifier for each cell and the corresponding signal strength. The training data was collected over an area of 465 square kilometers, driving for 208 hours (see Table II for details).

	Training Data	Test Data		
		Downtown	Residential	Suburban
Duration	208hr	70min	80min	89min
Distance	4350km	24km	38km	51km
Dimension	25.0x18.6km	2.2x2.1km	2.4x4.4km	4.5x5.5km

TABLE II

In addition to the GPS unit, which was mounted on top of the car roof, we put three phones on the dashboard. Each of the phones was connected to a different cell phone provider (ATT, Cingular, and T-mobile). In addition, we collected three test traces in different areas of town, chosen to cover different cell tower densities. The density of cell towers is important as a cell tower can theoretically be seen at distances of up to 35km. Therefore, GSM signal strength is by far not as discriminative as WiFi signal strength.

We compared the results achieved with our GP with other techniques for localization. The simplest one is the centroid technique where the estimated position is the average of the locations of the seen cell towers. The weighted centroid approach uses additional weights for the position of the seen cell tower corresponding to their current signal strength. In dense areas these techniques can estimate the location within a comparable accuracy, but fail in less dense areas (see

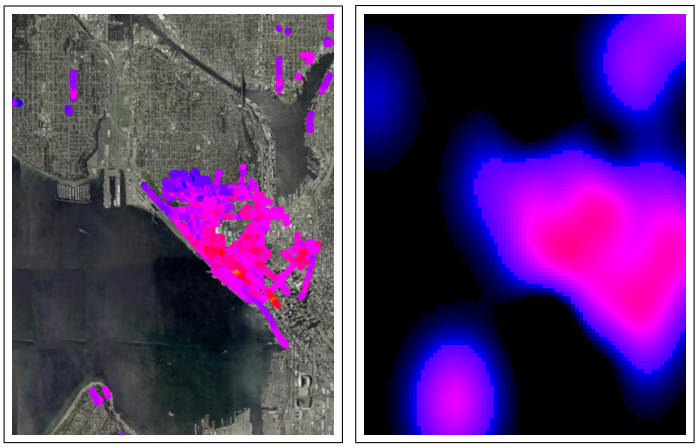

Fig. 7. The left image shows the measurements for one cell tower. The color of the measurements corresponds to the signal strength. The right image shows the GP mean estimate of the signal strength.

Technique	Median Error in m		
	Downtown	Residential	Suburban
Centroid	232	1209	612
Weighted Centroid	184	765	561
Fingerprinting	94	255	293
Gaussian Processes	128	208	236

TABLE III

Table III). The third technique is fingerprinting [1]. The basic idea here is to mark every location with a unique set of cell tower identifications and signal strengths. The current measurement is compared with the database of all fingerprints and the location of the fingerprint that corresponds at most to the measurements is then chosen. Fingerprinting needs dense training coverage as it is not able to localize in areas that are not included in the training data. Table III shows the accuracy of this technique which is comparable to the GP based technique and slightly better in the downtown area. This area has the highest density of cell towers. This advantage will be mitigated when evaluated in sparse training environments, where GPs outperform fingerprinting techniques.

VI. Conclusions

We presented Gaussian processes for localization based on signal strength measurements. GPs are ideally suited for representing the complex likelihood models of such measurements. They overcome various limitations of previous techniques: they do not rely on a discrete representation of space, they are non-parametric and can thus represent arbitrary likelihood models, they correctly represent uncertainty due to sparse training data, and they enable the consistent estimation of hyperparameters.

We show how to incorporate a GP likelihood model into a Bayesian filter operating in a novel representation of indoor environments. This representation combines a graph structure with free space regions. Our representation allows the Bayes filter to constrain a person's path when moving through hallways or elevators while allowing for free movement in open areas. Our experiments show that the resulting system can accurately track a person moving through a large indoor environment. Furthermore, the GP is able to accurately predict WiFi measurements in rooms that were not visited in the training phase. We also present results in large scale outdoor environments using GSM signal strength. We believe that the results achieved with our approach are superior to those presented in the literature so far.

One of the main problems of GPs is the complexity of model learning when using large data sets (≥ 800 data points). Fortunately, there exist various sparse approximations for GPs and we are currently investigating their use. We strongly believe that GP regression can be applied successfully to various robotics problems, including robot localization [5] and mobile sensor networks [3]. We are additionally investigating the use of GPs for WiFi-SLAM, where a signal strength map is generated by moving through an unknown environment.

VII. Acknowledgments

The authors thank Aaron Hertzmann for useful discussions. This work has partly been supported by DARPA's ASSIST and CALO Programmes (contract numbers: NBCH-C-05-0137, SRI subcontract 27-000968).

References

[1] P. Bahl and V.N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of IEEE Infocom*, 2000.

[2] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.

[3] C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements using Gaussian processes. In *Proc. of the International Conference on Machine Learning (ICML)*, 2005.

[4] A. Haeberlen, E. Flannery, A.M. Ladd, A. Rudys, D.S. Wallach, and L.E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proc. of the Tenth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2004.

[5] A. Howard, S. Siddiqi, and G. Sukhatme. An experimental study of localization using wireless ethernet. In *Proc. of the International Conference on Field and Service Robotics*, 2003.

[6] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001.

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

[8] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Self-mapping in 802.11 location systems. In *International Conference on Ubiquitous Computing (UbiComp)*, 2005.

[9] N. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[10] J. Letchner, D. Fox, and A. LaMarca. Large-scale localization from wireless signal strength. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[11] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. Voronoi tracking: Location estimation using sparse and noisy sensor data. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[12] K. Liu, A. Hertzmann, and Z. Popovic. Learning physics-based motion style with nonlinear inverse optimization. In *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 2005.

[13] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2005.

[14] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann. GPPS: A Gaussian process positioning system for cellular networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[15] S. Seidel and T. Rappaport. 914 MHz path loss prediction models for indoor wireless communictions in multifloored buildings. *IEEE Transactions on Antennas and Propagation*, 40(2), 1992.

[16] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, September 2005. ISBN 0-262-20162-3.