

WiFi-SLAM Using Gaussian Process Latent Variable Models

Brian Ferris Dieter Fox Neil Lawrence[†]

University of Washington, Department of Computer Science & Engineering

[†]University of Sheffield, Department of Computer Science

Abstract

WiFi localization, the task of determining the physical location of a mobile device from wireless signal strengths, has been shown to be an accurate method of indoor and outdoor localization and a powerful building block for location-aware applications. However, most localization techniques require a training set of signal strength readings labeled against a ground truth location map, which is prohibitive to collect and maintain as maps grow large. In this paper we propose a novel technique for solving the WiFi SLAM problem using the Gaussian Process Latent Variable Model (GP-LVM) to determine the latent-space locations of unlabeled signal strength data. We show how GP-LVM, in combination with an appropriate motion dynamics model, can be used to reconstruct a topological connectivity graph from a signal strength sequence which, in combination with the learned Gaussian Process signal strength model, can be used to perform efficient localization.

1 Introduction

The use of wireless signal strength information to localize mobile devices 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.

The unpredictability of signal propagation through indoor environments is the key challenge in location estimation from wireless signal strength. 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 good likelihood models from calibration data collected in an environment. This data typically consists of signal strength measurements annotated with the ground truth locations of the device. The calibration data is then often used to estimate the parameters of locally Gaussian models, which have proven to be very effective for localization [Haeberlen *et al.*, 2004; Letchner *et al.*, 2005; Ferris *et al.*, 2006].

The reliance on ground truth calibration data clearly limits the applicability of existing techniques. In order to enable location-aware mobile devices, these devices need to be able to build their own spatial representations based on sequences of raw, unlabeled signal strength data. While the robotics and computer vision communities have developed techniques for jointly estimating the locations of a device and a map of an environment, the nature of wireless signal strength prohibits the use of standard SLAM (simultaneous localization and mapping) techniques [Thrun *et al.*, 2005].

In this paper we present WiFi-SLAM, a novel technique for building wireless signal strength maps without requiring any location labels in the training data. To do so, our approach builds on Gaussian process latent variable models (GP-LVM), a recently developed technique for mapping high-dimensional data to a low-dimensional latent space [Lawrence, 2004]. In our context, the high-dimensional data corresponds to the signal strength information for all WiFi access points in an environment. GP-LVMs map these signal strength measurements to a two-dimensional latent space, which can be interpreted as the xy -coordinates of the device. Our embedding technique considers the following constraints to solve this challenging SLAM problem.

1. *Locations* \rightarrow *signal strength*: Locations that are near each other should observe similar signal strength measurements.
2. *Signal strengths* \rightarrow *locations*: Similar signal strength measurements indicate that they were observed at locations near each other. This constraint is extremely important to close loops, *i.e.* to detect when the person returns to a previously visited location during the mapping process. While this constraint might be violated for individual access points, it is a good approximation for typical office environments, where multiple access points are visible at each point in time.
3. *Locations* \rightarrow *locations*: Locations that follow each other in the data stream should be near each other. This constraint models the fact that mapping data is sequential, collected by a person walking through the building.

Our approach incorporates these constraints in a single, statistically sound way. As a result, the technique is able to build topologically correct maps from raw signal strength measurements. Furthermore, the resulting map provides a signal

strength sensor model that can be used for Bayesian filtering for location estimation.

This paper is organized as follows. After discussing related work in the next section, we provide background on GPLVMs in Section 3. Their application to WiFi-SLAM is described in Section 4, followed by experiments and conclusions.

2 Related Work

WiFi localization techniques fall into a number of broad categories. Several location estimation techniques attempt to model directly signal propagation through space [Bahl and Padmanabhan, 2000], assuming known locations of access points and an exponential signal attenuation model. However, even when considering the location and material of walls and furniture inside buildings, the accuracy of signal propagation models is very limited. Other techniques attempt to model reading likelihoods on a location-specific basis [Haeberlen *et al.*, 2004; Letchner *et al.*, 2005], representing signal strength at locations of interest with probability distributions learned from training data. While more accurate than signal propagation models, these methods are inherently discrete and have only limited capabilities for interpolation between locations.

To overcome these limitations, Schwaighofer and colleagues [2003] showed how to apply Gaussian processes to signal strength localization, resulting in a model that provided interpolation over continuous locations with direct modeling of uncertainty from the training data. [Ferris *et al.*, 2006] extended this technique to WiFi localization by combining the GP signal strength model with graph-based tracking, allowing for accurate localization in large scale spaces. Our WiFi-SLAM technique builds on these recent models and extends them to the case of mapping with unknown locations.

The robotics community has developed various approaches for SLAM, see [Thrun *et al.*, 2005]. However, these techniques typically rely on the ability to sense and match discrete entities such as visual landmarks or obstacles detected by sonar or laser range-finders. Compared to these approaches, our SLAM problem is particularly challenging, since we have only very limited information about the person’s motion, and since signal strength measurements vary extremely smoothly over the location space. In principle we could extract distance to access point estimates from signal strength measurements and then apply range-only SLAM [Newman and Leonard, 2003] to map the locations of access points. However, such techniques typically rely on rather accurate range measurements, which are clearly not available in our context.

The WiFi-SLAM problem can be seen as an application-specific case of the general dimensionality reduction problem. Specifically, we wish to reduce high-dimensional signal strength to some two-dimensional location space. Unfortunately, dimensionality reduction algorithms such as PCA do not typically perform well with highly non-linear manifolds such as those found in WiFi signal propagation. A very promising class of techniques extend non-linear dimensionality reduction methods to allow the specification of additional constraints that reflect the generative model of the underlying data process. One such technique is action respecting embeddings (ARE) [Bowling *et al.*, 2005], which extends Semidefinite

Embedding (SDE) by placing additional constraints on the SDE similarity matrix. Specifically, when the same control action (*e.g.*, robot motion) is applied to two different points in the latent space, their successive latent points should share a similar relative transformation. While ARE could be used in our context to represent time-series constraints between successive signal strength readings, we wish to more generally constrain higher level properties of the underlying generative process.

A powerful technique for representing such constraints is proposed in Gaussian Process Dynamical Models [Wang *et al.*, 2006]. The technique extends GP-LVM by including an additional likelihood model for the latent-space variables. Specifically, the technique models dynamic constraints via a Gaussian process mapping between consecutive data points. Though their technique was specifically designed for vision-based motion tracking, we have extended it for our WiFi-SLAM approach.

3 Gaussian Process Latent Variable Models

In this section, we will first describe Gaussian processes for regression. We will then illustrate their application to estimating a likelihood model of wireless signal strength measurements, assuming that the ground truth locations of the training data are known. Finally, we will show how Gaussian process latent variable models can be used to handle the case of unknown locations.

3.1 Gaussian Processes

GPs can be derived in different ways. Here, we follow closely the function-space view described in [Rasmussen and Williams, 2005]. Let our data, $D = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, be a set of training samples drawn from a noisy process

$$\mathbf{y}_i = f(\mathbf{x}_i) + \varepsilon, \quad (1)$$

where each \mathbf{x}_i is an input sample in \mathbb{R}^q and each \mathbf{y}_i is a target value, or observation, in \mathbb{R} . The noise, ε , is assumed to come from a zero mean, additive Gaussian distribution with known variance σ_n^2 . For notational convenience, we aggregate the n input vectors \mathbf{x}_i into a $n \times q$ design matrix \mathbf{X} , and the target values \mathbf{y}_i into the $n \times 1$ vector denoted \mathbf{Y} . Thus, \mathbf{x}_i and \mathbf{y}_i are vectors that correspond to the i -th row of \mathbf{X} and \mathbf{Y} , respectively. In the context of signal strength maps, \mathbf{X} represents locations in the 2D plane, and \mathbf{Y} represents signal strength measurements of a single WiFi access point.

A Gaussian process estimates posterior distributions over functions f from the 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, $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$, depends on the covariance between the two input values, some \mathbf{x}_i and \mathbf{x}_j . This dependency can be specified via an arbitrary covariance function, or kernel $k(\mathbf{x}_i, \mathbf{x}_j)$. The choice of the kernel function is typically left to the user, the most widely used being the squared exponential, or Gaussian, kernel:

$$\text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}|\mathbf{x}_i - \mathbf{x}_j|^2\right). \quad (2)$$

Here, σ_f^2 is the signal variance and l is a length scale that determines the strength of correlation between points. Both parameters control the smoothness of the functions estimated by a GP. 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 access to the function values, but only noisy observations thereof, it is necessary to represent the corresponding covariance function for noisy observations:

$$\text{cov}(\mathbf{y}_r, \mathbf{y}_s) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{rs}. \quad (3)$$

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

$$\text{cov}(\mathbf{Y}) = K + \sigma_n^2 I, \quad (4)$$

where K is the $n \times n$ covariance matrix of the input values, that is, $K[i, j] = k(\mathbf{x}_i, \mathbf{x}_j)$.

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 [Rasmussen and Williams, 2005]. 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 $\sigma_{\mathbf{x}_*}^2$:

$p(f(\mathbf{x}_*) | \mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \mathcal{N}(f(\mathbf{x}_*); \mu_{\mathbf{x}_*}, \sigma_{\mathbf{x}_*}^2)$, where

$$\mu_{\mathbf{x}_*} = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{Y} \quad (5)$$

$$\sigma_{\mathbf{x}_*}^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \quad (6)$$

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, $\sigma_{\mathbf{x}_*}^2$, 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} ¹.

Finally, in order to get the predictive distribution for a noisy observation, y_* , one simply has to add the observation noise:

$$\begin{aligned} p(y_* | \mathbf{x}_*, \mathbf{X}, \mathbf{Y}) &= \int p(y_* | f(\mathbf{x}_*)) p(f(\mathbf{x}_*) | \mathbf{x}_*, \mathbf{X}, \mathbf{Y}) df(\mathbf{x}_*) \\ &= \mathcal{N}(y_*; \mu_{\mathbf{x}_*}, \sigma_{\mathbf{x}_*}^2 + \sigma_n^2) \end{aligned} \quad (7)$$

This predictive distribution summarizes the key advantages of GPs in the context of wireless signal strength models. In addition to providing a regression model based on training data, the GP also represents the uncertainty of the prediction, taking both sensor noise and model uncertainty into account.

¹Observed values \mathbf{Y} may impact the *parameters* of the covariance function via hyperparameter estimation [Ferris *et al.*, 2006].

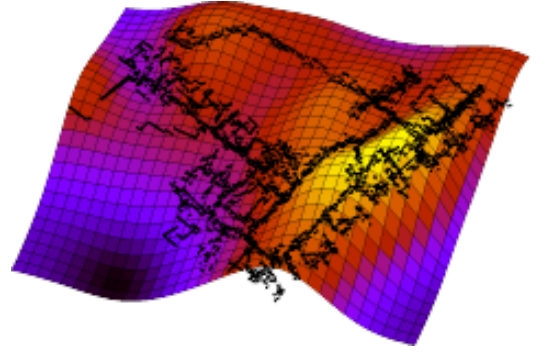


Figure 1: GP model for signal strength of one access point. Black dots indicate training data along the ground truth path.

3.2 Application to Signal Strength Modeling

GPs have been applied very successfully to signal strength modeling, where the input values \mathbf{X} correspond to locations, and the observations \mathbf{Y} correspond to signal strength measurements obtained at these locations [Schwaighofer *et al.*, 2003; Ferris *et al.*, 2006]. Here, the GP posterior is estimated from a trace of signal strength measurements annotated with their locations. Assuming independence between different access points, these techniques estimate a GP for each access point separately.

Fig. 1 illustrates the GP signal strength model for one access point in an environment of size 60×50 meters, including hallways, offices, and open areas. Fig. 1 clearly demonstrates that signal strength propagation cannot be represented adequately by a parametric sensor model such as one that estimates the location of an access point and then assumes a radial signal propagation model. The GP on the other hand, smoothly approximates the data points (see [Ferris *et al.*, 2006] for details).

3.3 Latent Variable Models

So far we have assumed that the locations \mathbf{X} of the training data are observed. The goal of our research is to build GP signal strength maps *without* relying on location data. To build such maps, we treat the locations \mathbf{X} as hidden, or latent, variables. The resulting, much more challenging problem can be addressed by Gaussian process latent variable models (GP-LVM), which were introduced by [Lawrence, 2004; 2005] in the context of data visualization.

In the previous section we treated different access points individually, resulting in a 1-dimensional observation space. Here, it will be necessary to jointly consider the signal strength of multiple access points, which results in d -dimensional observations \mathbf{y}_i , reporting the signal strength s of d access points. The traditional probabilistic approach to latent variable techniques models the relationship between latent variables, $\mathbf{X} \in \mathbb{R}^{n \times q}$, and a set of observed data, $\mathbf{Y} \in \mathbb{R}^{n \times d}$, through the parameterized function

$$y_{ij} = f(\mathbf{x}_i; \mathbf{w}_j) + \epsilon, \quad (8)$$

where y_{ij} is the element from the i -th row and j -th column of \mathbf{Y} , \mathbf{x}_i is the vector formed from the i th row of \mathbf{X} , \mathbf{w}_j are the parameters of the (possibly non-linear) function $f(\cdot)$, and ϵ is

a zero mean noise term. This function, which is the same as (1) with explicit parameters, implies a probabilistic relationship between the latent variables and the data, $p(y_{ij}|\mathbf{x}_i, \mathbf{w}_j)$. If independence is assumed across the dimensions and instantiations of the data this relationship factorizes as

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{ij} p(y_{ij}|\mathbf{x}_i, \mathbf{w}_j). \quad (9)$$

Since the latent variables are unknown, they must now be dealt with. Traditionally this is done through marginalization, necessitating the definition of a prior distribution. A common choice of prior is the normalized zero mean Gaussian, $p(x_{ij}) = \mathcal{N}(x_{ij}; 0, 1)$, $p(\mathbf{X}) = \prod_{ij} p(x_{ij})$. Since the noise is assumed to be drawn from a zero mean Gaussian with a given variance σ_n^2 , the parametric relationship between the latent variables and the data can then be recovered through optimization of the marginal likelihood. If the functional relationship between the latent variables and the data is linear,

$$f(\mathbf{x}_i; \mathbf{w}_j) = \mathbf{w}_j^T \mathbf{x}_i, \quad (10)$$

then the marginal likelihood is of the form [Tipping and Bishop, 1999]

$$p(\mathbf{Y}|\mathbf{W}) = \mathcal{N}(\mathbf{Y}; \mathbf{0}, \mathbf{W}\mathbf{W}^T + \sigma_n^2 I). \quad (11)$$

The likelihood of this model can then be maximized with respect to \mathbf{W} with the result that \mathbf{W} spans the principal subspace of the data [Tipping and Bishop, 1999]. In the context of WiFi mapping, however, $f(\mathbf{x}_i; \mathbf{w}_j)$ is clearly non-linear and thus the marginalization to obtain (11) cannot be performed without reverting to approximations.

In the GP-LVM an alternative approach is proposed. The GP-LVM is inspired from the Bayesian point of view. In the Bayesian approach, rather than maximizing the likelihood of the parameters, they are treated as latent variables and marginalized. Unfortunately, even for the linear case we have just described, it is not possible to marginalize both the latent variables and the parameters. If we do not wish to turn to approximate methods we are therefore faced with a choice of what to marginalize: the latent variables or the parameters. In [Lawrence, 2004; 2005] it was shown that if the prior distribution for the parameters is Gaussian, $p(w_{ij}) = \mathcal{N}(w_{ij}; 0, 1)$, $p(\mathbf{W}) = \prod_{ij} p(w_{ij})$, then maximizing the likelihood with respect to \mathbf{X} still leads to principal component analysis. For this case the marginal likelihood takes the form

$$p(\mathbf{Y}|\mathbf{X}) = \prod_j \mathcal{N}(\mathbf{y}_j; \mathbf{0}, \mathbf{X}\mathbf{X}^T + \sigma_n^2 I). \quad (12)$$

By comparison with (7) this is recognized as a product of Gaussian process models with a linear covariance function. We can extend this model to arbitrary covariance relationships by considering the covariance defined via a kernel function, as in (4). Thus, maximizing (12) with respect to \mathbf{X} determines the x_{ij} values that maximize the data likelihood assuming an underlying Gaussian process model.

4 GP-LVMs for WiFi SLAM

We utilize the GP-LVM framework to probabilistically model the relationship between WiFi signal strength readings \mathbf{Y} and

their underlying positions \mathbf{X} in the two-dimensional coordinate space. The likelihood of our model takes the form:

$$p(\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}) \quad (13)$$

In the GP-LVM framework, $p(\mathbf{Y}|\mathbf{X})$ is modeled by a GP and $p(\mathbf{X})$ is the expression of our latent-space dynamics model. The set of constraints we discussed in the introduction of the WiFi-SLAM problem are nicely captured within these terms, as we shall describe below.

Location \rightarrow signal strength: We wish to constrain that similar locations should have similar signal strengths. This property is captured by most dimensionality reduction techniques, including GP-LVM. Specifically, the GP used to model $p(\mathbf{Y}|\mathbf{X})$ provides this property, where the similarity of points in the latent space is controlled by the underlying kernel function. For the WiFi case, we use the squared exponential kernel (2) with parameter values trained from previously labeled WiFi training data.

Location \rightarrow location: We wish to model the motion of the person collecting the mapping data. These constraints encompass the dynamics component of the GP-LVM. For the WiFi case, we do not have strong odometry data, so we additionally constrain \mathbf{X} against a generative model of a generic office building. Remember that \mathbf{X} is an $n \times q$ design matrix, where each row \mathbf{x}_i is the latent position of the WiFi readings y_i . We model three constraints in our dynamics models:

- *Distance between successive positions*
- *Change in orientation between successive positions*
- *Alignment of parallel line segments*

We represent the constraints as a conjunction of probabilities:

$$p(\mathbf{X}) = p(\text{distance})p(\text{orientation})p(\text{alignment}). \quad (14)$$

To constrain the distance between successive points, we apply a simple time-series constraint based on a Gaussian prior on walking velocity $\mathcal{N}(\mu_v, \sigma_v)$, which results in

$$p(\text{distance}) = \prod_{i=1}^{n-1} \mathcal{N}(\|\mathbf{x}_{i+1} - \mathbf{x}_i\|; \Delta_i \mu_v, \Delta_i \sigma_v). \quad (15)$$

Here, Δ_i is the time difference between consecutive measurements. We also consider the change in orientation between successive points, where θ_i is the change in orientation between $\mathbf{u}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$ and $\mathbf{u}_{i+1} = \mathbf{x}_{i+1} - \mathbf{x}_i$. We assume that for a general office environment, most time will be spent walking in a straight line along hallways, with periodic intersections. We place a prior on change in orientation as a zero-mean Gaussian with sigma σ_o . Thus, we can model the likelihood of latent locations against our orientation model as

$$p(\text{orientation}) = \prod_{i=2}^{n-1} \mathcal{N}(\theta_i; 0, \sigma_o). \quad (16)$$

We finally consider the alignment constraint. The intuition behind this constraint is that when a trace visits the same hallway twice, there will be two roughly parallel paths that should be aligned. To model this constraint, we consider the probability that two segments $\mathbf{u} = \mathbf{x}_i \rightarrow \mathbf{x}_{i+1}$ and

$\mathbf{v} = \mathbf{x}_j \rightarrow \mathbf{x}_{j+1}$ are parallel as $p(\text{parallel}) = \mathcal{N}(\theta; 0, \sigma)$ where θ is the relative orientation between the two segments. We only align parallel segments which are already close to each other, as captured by $p(d_p < 5)$ where d_p is the perpendicular distance between the two segments, measured in meters. The inequality is modeled as the likelihood that $\mathcal{N}(d_p, \sigma_1)$ is less than $\mathcal{N}(5, \sigma_2)$, defined as $1/2 \left(1 + \text{erf} \left((d_p - 5) / \sqrt{2(\sigma_1^2 + \sigma_2^2)} \right) \right)$. If both segments are parallel and close by, we constrain their separation distance to zero as $p(\text{separation}) = \mathcal{N}(d_p; 0, \sigma)$. Thus, our full alignment constraint is

$$p(\text{alignment}) = p(\text{parallel})p(d_p < 5) \Rightarrow p(\text{separation}). \quad (17)$$

All told, we have defined the conjunction of three probabilistic models constraining distance, orientation, and alignment of latent space coordinates to provide $p(\mathbf{X})$ as the dynamics model. When combined with the GP model of $p(\mathbf{Y}|\mathbf{X})$, we have defined our full probability model. However, we have one constraint left.

Signal Strength \rightarrow location: We wish to constrain that similar signal strength readings should be mapped to similar locations. Note that this constraint is only reasonable in dense WiFi environments such as offices and urban areas. This constraint is not provided by default in GP-LVM. However, Lawrence and Quiñero-Candela [2006] demonstrate the use of back constraints to constrain local distance in \mathbf{Y} to local distances in \mathbf{X} . Back constraints are implemented as a smooth internal mapping of $\mathbf{X} = g(\mathbf{Y}, \phi)$ where ϕ is a set of mapping parameters. The smoothness of the mapping ensures that nearby points in \mathbf{Y} are also nearby in \mathbf{X} .

Finally, we optimize the marginalized latent positions \mathbf{X} by minimizing the negative log-likelihood of our full GP-LVM likelihood model $P(\mathbf{X}, \mathbf{Y})$. We can do this efficiently with conjugate gradient-descent, where the gradient of the log-likelihood can be evaluated with respect to each element of \mathbf{X} . Performance is improved and local minima avoided if we initialize the starting point of the descent with a reasonable value. In our case, we apply Isomap to generate an acceptable initialization. We will contrast the results of Isomap with the full GP-LVM in the results section.

5 Experimental Results

Our experimental evaluation consists of collecting a series of WiFi signal-strength traces, generating a sensor model and map using the WiFi-SLAM technique, and then cross-validating the model by localizing the remaining traces and considering the error.

We collected three traces over one floor of a university building. Traces ranged in length from 250 meters to half a kilometer. Traces were collected with an iPAQ hand-held, a standard WiFi card, and a click-to-map-based annotation program for establishing ground truth location. As seen in the ground truth path in Fig. 3, traces included three loop closures and multiple trips through each hallway. For each trace, we initialized GP-LVM with an Isomap embedding of \mathbf{X} , which, as also seen in Fig. 3, does a reasonable job of recovering the overall structure of the WiFi trace. However, the embedding lacks finer resolution that would make it possible

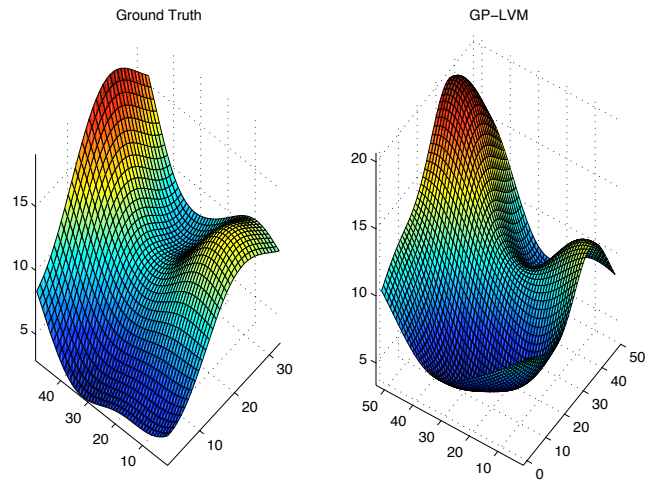


Figure 2: Comparison of ground truth Gaussian process versus that constructed by GP-LVM.

to easily establish the topological connectivity graph for later localization. We will demonstrate these finer features in our GP-LVM results.

We see in the final plot in Fig. 3 the result of the latent space optimization of GP-LVM. The plot shows clear definition of intersections, correct alignment of paths, and a strong topographical similarity to the ground truth trace. We had similar results for our other test traces, though they are not displayed here for brevity. The plot also demonstrates the current limitations of our approach. The ground truth trace includes a non-right-angle intersection, which is simplified as a right angle in our GP-LVM result. While we have no explicit angular constraint in our motion model, the constraint emphasizing straight and aligned paths tends to prefer a grid like structure in the resulting map.

Given the resulting latent-space optimization, it is trivial to extract an intersection connectivity graph from this representation. The resulting graph and the GP sensor model produced as by-product of GP-LVM are immediately applicable to the task of Bayesian localization, as described in [Ferris *et al.*, 2006]. In Fig. 2, we see a comparison of a GP sensor model resulting from the ground truth locations and a GP model resulting from the optimized GP-LVM latent space. We present these results as a demonstration that the two models are quite similar, suggesting our optimized latent space is reasonable.

To evaluate localization accuracy, we cross-validated each GP-LVM result by performing localization using the remaining two test traces. Since the mapping from the latent coordinate system \mathbf{X} to the underlying ground truth coordinate system is not well-defined, we evaluated localization accuracy using the metric described in [Bowling *et al.*, 2005]. Specifically, to determine the location in the ground truth coordinate system, we find the closest training point in the latent space to our current location and use its ground truth location as our own location. Put differently, whenever a person returns to a previously visited location, we estimate how accurately our system can detect this event.

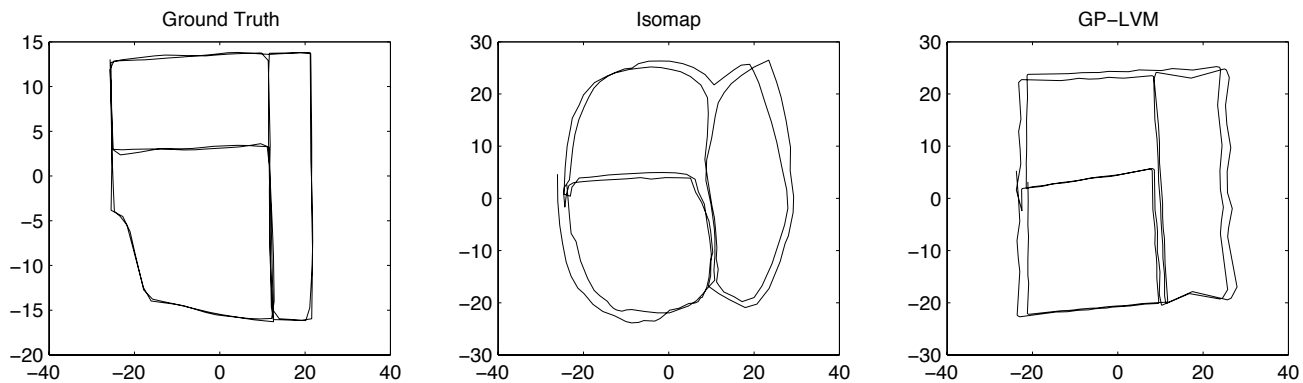


Figure 3: The results of an GP-LVM run for one of our traces. While the Isomap result (center) shows general structure, the GP-LVM result (right) has clearer correspondence with the ground truth trace (left).

We calculated the mean localization error over six localization runs to be 3.97 ± 0.59 meters. While the localization accuracy is not as high as those achieved with techniques based on labeled calibration data, the fact that reasonably accurate localization can be performed without any labeled training data indicates the success of our technique.

6 Conclusion

We have demonstrated a method for addressing the WiFi-SLAM problem. The key idea of our approach is to apply Gaussian process latent variable models (GP-LVM) in order to estimate the locations of WiFi signal strength readings collected by a person walking through a building. GP-LVMs provide a powerful framework for jointly modeling constraints on WiFi signal strength measurements and on a person’s motion. We have shown that GP-LVMs in combination with a simple dynamics model can reconstruct topographically-correct connectivity graphs and GP sensor models that can be used directly for WiFi localization.

WiFi-SLAM serves as a promising solution to the problem of collection and maintenance of WiFi sensor models for large-scale localization. In our future work, we wish to integrate additional sensors, such as accelerometer, digital compass, and barometer, to increase the accuracy of our technique. We additionally wish to extend mapping into more complex office environments, enhancing our ongoing research in desktop and mobile activity recognition.

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

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

[Bowling *et al.*, 2005] M. Bowling, D. Wilkinson, A. Ghodsi, and A. Milstein. Subjective localization with action respecting em-

bedding. *The International Symposium of Robotics Research*, 143, 2005.

[Ferris *et al.*, 2006] B. Ferris, D. Hähnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proc. of Robotics Science and Systems*, 2006.

[Haerberlen *et al.*, 2004] A. Haerberlen, 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.

[Lawrence and Quiñero Candela, 2006] Neil D. Lawrence and Joaquin Quiñero Candela. Local distance preservation in the GP-LVM through back constraints. volume 23. Omnipress, 2006.

[Lawrence, 2004] Neil D. Lawrence. Gaussian process models for visualisation of high dimensional data. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, Cambridge, MA, 2004.

[Lawrence, 2005] Neil D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6, Nov 2005.

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

[Newman and Leonard, 2003] P. Newman and J. Leonard. Pure range-only subsea slam. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2003.

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

[Schwaighofer *et al.*, 2003] 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.

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

[Tipping and Bishop, 1999] Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999.

[Wang *et al.*, 2006] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18 [Proceedings of NIPS 2005]*, pages 1441–1448. The MIT Press, 2006.