Efficient, Generalized Indoor WiFi GraphSLAM

Joseph Huang, David Millman, Morgan Quigley, David Stavens, Sebastian Thrun and Alok Aggarwal

Abstract— **The widespread deployment of wireless networks presents an opportunity for localization and mapping using only signal-strength measurements. The current state of the art is to use Gaussian process latent variable models (GP-LVM). This method works well, but relies on a signature uniqueness assumption which limits its applicability to only signal-rich environments. Moreover, it does not scale computationally to large sets of data, requiring $O\left(N^3\right)$ operations per iteration. We present a GraphSLAM-like algorithm for signal strength SLAM. Our algorithm shares many of the benefits of Gaussian processes, yet is viable for a broader range of environments since it makes no signature uniqueness assumptions. It is also more tractable to larger map sizes, requiring $O\left(N^2\right)$ operations per iteration. We compare our algorithm to a laser-SLAM ground truth, showing it produces excellent results in practice.**

I. Introduction

The widespread deployment of wireless networks presents an opportunity for localization and mapping using signal-strength measurements. Wireless networks are ubiquitous, whether in the home, office, shopping malls, or airports.

Recent work in signal-strength-based simultaneous localization and mapping (SLAM) uses Gaussian process latent variable models (GP-LVM). However, this work requires that maps are limited to very specific predefined shapes (e.g. narrow and straight hallways) and WiFi fingerprints are assumed unique at distinct locations. As acknowledged by [1], in the absence of any odometry information, arbitrary assumptions must be made about human walking patterns and data association.

GraphSLAM is a commonly used technique in the robotics community for simultaneously estimating a trajectory and building a map offline. It shares many benefits of Gaussian processes, but can be applied to a broader range of environments. We show how wireless signal strength SLAM can be formulated as a GraphSLAM problem. By using GraphSLAM, we address limitations of previous work and improve runtime complexity from $O\left(N^3\right)$ to $O\left(N^2\right)$, where N is the dimensionality of the state space i.e. the number of poses being estimated.

In both GraphSLAM and Gaussian processes, measurement likelihoods are modeled as Gaussian random variables. Gaussian processes can always improve their model fit by simply moving all points away from each other. To prevent these trivial solutions, GP-LVM methods require special constraints. In the case of signal strength SLAM, the special constraints force similar signal strengths to similar locations. GraphSLAM requires no special constraints. This makes GraphSLAM suitable to a wider range of real-world environments.

An appeal of GraphSLAM is that it reduces to a standard non-linear least squares problem. This gives GraphSLAM access to widely used and well-studied techniques for its optimization. We present a parameterization of the state space for typical mobile phone applications.

Our results compare the GraphSLAM approach for WiFi SLAM against a LIDAR-based GraphSLAM implementation. Using real-world datasets, we are able to demonstrate a localization accuracy of between 1.75 m and 2.18 m over an area of 600 square meters. We explain how the resultant maps are directly applicable to online Monte Carlo Localization.

II. Background

A. Related Work

If wireless signal strength maps are determined ahead of time, Monte Carlo localization methods can achieve high accuracy indoor localization. [2] discretizes the signal strength map into a spatial grid and, combined with contact sensing, obtains 0.25 m accuracy using standard Monte Carlo methods while improving convergence time over contact sensing alone. [3] also performs spatial discretization of the signal strength map and combines WiFi with a low-cost image sensor to localize within 3 m. [4] expresses signal-strength maps as a hybrid connectivity-graph/free-space representation and achieves 1.69 m localization with signal-strength sensors alone. However until now, the process for obtaining signal strength maps remains expensive and time consuming.

This paper focuses on techniques to improve the state of the art in signal-strength-only SLAM in indoor environments. Outdoor applications are likely better handled by GPS and/or attenuation model [5] or range-based SLAM [6] methods. Other indoor signal-strength-based localization research relies on extensive training phases [7] or incorporates other features of the signal such as time-of-arrival or angle-of-arrival measurements [8], [9]. However, in most pedestrian applications, such data is inaccessible to the general public without additional infrastructure costs. The implications of low-cost signal-strength SLAM are especially meaningful for large (indoor) GPS deprived environments such as shopping malls, airports, etc. where wireless internet infrastructure is readily accessible.

Existing wireless mapping techniques model the signal data in different ways. Some assume a model of the signal propagation [10], [5]. Others use a connectivity graph of predetermined cells to localize coarsely [11], [12]. Since these techniques rely on pre-existing information about the environment, they do not handle the problem of mapping in unknown locations. We avoid the requirement that wall

locations are known [13] or even that small amounts of data have been pre-labeled [14].

The current state of the art uses Gaussian processes to determine a map of signal strength without modeling the propagation from transmitting nodes explicitly [1]. Gaussian processes are applied to WiFi-SLAM under a specific set of assumptions.

B. Motivation

We wish to lift the restriction that similar signal strength fingerprints/signatures all correspond to a similar location on the map. If the geographic distribution of access points is sparse, there are more spatial configurations where the fingerprint uniqueness assumption breaks down. As Figure 1 illustrates, real-world hypotheses are often multi-modal. Especially at lower signal strengths, due to the log relationship between signal strength and distance, signal strength can be almost completely invariant over very large sections of space.
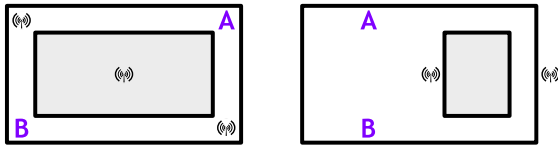

Fig. 1. Two examples of wireless node deployment. In each example, location A and location B share the same signal strength signature/fingerprint. More generally, at indoor scales, signal strength can often be relatively invariant over long regions of open areas in line-of-sight directions.

Relaxing this requirement brings modern signal-strength-based SLAM to sparse signal environments. Furthermore, explicitly mapping similar signal strengths to similar locations hurts scalability: as the dataset grows, the risk increases of erroneous measurements being incorrectly mapped by this constraint. Since such mappings are hard constraints, these errors are completely unrecoverable. The current state of the art cannot achieve signal-strength SLAM without relying upon explicit fingerprint uniqueness back-constraints. As claimed by [1], the GP-LVM method is only reasonable in dense environments. Our method does not make any assumptions of fingerprint uniqueness. Therefore, signal density or sparsity, which influences fingerprint uniqueness, is no longer a concern.

In order to provide a SLAM solution suitable for both recti-linear corridor-type environments as well as open atrium-type environments, we incorporate low-cost IMU data. Introducing motion measurements makes the sensor model general enough to apply to a wide range of crowdsourcing applications. Subjects need not explicitly cooperate with predetermined walking patterns, consistent walking speeds, etc. Admittedly, motion sensors also make the problem easier. In section VI we demonstrate the viability of low-cost WiFi SLAM and compare it directly to a laser-SLAM implementation.

We improve the scalability of modern signal strength SLAM to larger datasets. We show that the proposed GraphSLAM based method has better runtime complexity than Gaussian process latent variable models. We demonstrate

experimentally that it produces useful results on real-world data.

III. TRADITIONAL GRAPHSLAM

The GraphSLAM family of techniques are commonly used in the robotics community for simultaneously estimating a trajectory and building a map offline. Such techniques have been used successfully in many applications in computer vision and robotics [15], [16], [17], [18]. We will reduce the signal strength SLAM problem to an instance of GraphSLAM in section IV.

GraphSLAM is traditionally formulated as a network of Gaussian constraints between robot poses and landmarks [15]. Measurements are assumed to contain only additive Gaussian noise and to be conditionally independent given the world states. For each measurement Z_i from any sensor, a state-to-measurement mapping function $h_i(\mathbf{X})$ describes the *true* measurement that would have taken place if the value of the state variables were known:

$$Z_i = h_i(\mathbf{X}) + \varepsilon_i$$

$\mathbf{X} = \{\vec{x}_{t_1}, \vec{x}_{t_2}, \ldots, \vec{m}_1, \vec{m}_2, \ldots\}$ represents the collection of all state variables (e.g. robot locations $\{\vec{x}_t\}$ and landmark locations $\{\vec{m}_i\}$) The measurement likelihood as a function of z_i is Gaussian with mean $h_i(\mathbf{X})$ and variance $\text{Var}\{\varepsilon_i\}$. We refer the reader to [19] and [15] for more details.

A. Motion Model

The typical GraphSLAM motion model can be expressed as state-to-measurement mappings. For example, a pedometry measurement on the interval between time t_i and t_{i+1} is related to the state space by the function

$$h_i^{\text{pedometry}}(\mathbf{X}) = \left\| \vec{x}_{t_{i+1}} - \vec{x}_{t_i} \right\|_2$$

Similarly, angular velocity measurements[1] correspond to

$$h_i^{\text{gyro}}(\mathbf{X}) = \frac{\text{atan2}\left(\vec{x}_{t_{i+1}} - \vec{x}_{t_i}\right) - \text{atan2}\left(\vec{x}_{t_i} - \vec{x}_{t_{i-1}}\right)}{\Delta t}$$

where $\Delta t = (t_{i+1} - t_{i-1})/2$.

The definition of a state-to-measurement mapping h together with the variance of the corresponding sensors noise, $\sigma_\varepsilon = \text{Var}(\varepsilon)$, completely describes the GraphSLAM measurement likelihood function for each sensor. The methods presented in section IV will work with any class of sensors. Pedometry and gyroscopes are merely used as examples here because they are the sensors of the sample dataset in section VI-A

IV. WIFI GRAPHSLAM

Let us define the state-to-measurement mapping for the ith signal strength measurement to be

$$h_i^{\text{WiFi}}(\mathbf{X}) = \vec{\beta}_i^{\text{T}} \vec{z}^{\text{WiFi}}$$

$$\vec{\beta}_i = \vec{w}_i - [\vec{w}_i]_i \hat{e}_i \quad , \qquad \left\| \vec{\beta}_i \right\|_1 = 1$$

---

[1] Implementations must be careful to account for headings that cross over from $-\pi$ to $+\pi$ and vice versa.

For notational convenience, $\vec{z}^{\text{WiFi}}$ is the vector of all signal strength measurements. $\vec{\beta}_i$ excludes the ith element from $\vec{w}_i$. $\vec{w}_i$ is the vector of interpolation weights for h_i^{WiFi}. The notation $[\vec{a}]_i$ is shorthand for "the ith element" of the vector $\vec{a}$, and $\hat{e}_i$ is the unit vector with all elements zero except the ith element. Note that h_i does not interpolate the ith measurement. This allows the measurement model to operate in sparse WiFi environments (see section IV-A).

At most distances from a transmission node, propagating radio waves are expected to have nearly the same power within any small region of free space. However, over larger regions across non-free space the relationship between "nearby" signal strengths is highly dependent on building structure/architecture. Without a model for building structure, we simply interpolate over small regions likely to be free space.

Intuitively, the quantity $[\vec{w}_i]_j$ can be imagined as related to the probability that location i and location j are "interpolatable", e.g. nearby and separated by only free space. In practice, at reasonable scales and in lieu of additional knowledge, Gaussian interpolation weights are a popular kernel choice and have been used with success to interpolate WiFi signal strengths in [20] as well as in many other machine learning applications [21], [22].

$$[\vec{w}_i]_j \propto \exp\left(-\frac{1}{2\tau^2}\left\|\vec{x}_{t_i} - \vec{x}_{t_j}\right\|_2^2\right)$$

τ is a scale parameter, related roughly to the distance between walls (e.g. 95% of walls can be considered at least 2τ away from measurement locations). τ can be learned from training data (the experimental value of τ for our dataset was approximately 2.2 meters). Figure 2 illustrates typical behavior of this signal interpolation method.
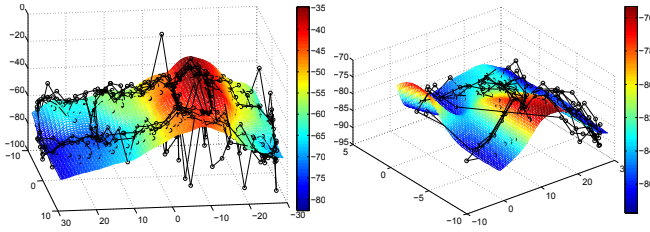

Fig. 2. Sample plots of h^{WiFi}, Gaussian weighted interpolation of WiFi points, evaluated over a grid with scale parameter $\tau = 2.2$ m. Black circles denote measured values. The vertical axis represents signal strength (dBm) and the horizontal axes represent spatial location.

With this formulation, for any specific measured value z_i, we can evaluate:

$$z_i - h_i(\mathbf{X}) = \varepsilon_i$$

$$\Rightarrow \begin{cases} \mathrm{E}\left[\vec{z}_i^{\text{WiFi}} - \sum_j \left[\vec{\beta}_i\right]_j z_j^{\text{WiFi}}\right] = 0 \\ \mathrm{Var}\left(\varepsilon_i^{\text{WiFi}} + h_i^{\text{WiFi}}(\mathbf{X})\right) = \left(1 + \left\|\vec{\beta}_i\right\|_2^2\right)\mathrm{Var}\left(\varepsilon^{\text{WiFi}}\right) \end{cases}$$

and thus the WiFi measurement likelihood, as a function of z_i, is Gaussian with mean $h_i(\mathbf{X})$ and variance $\left(1 + \left\|\vec{\beta}_i\right\|_2^2\right)\sigma_\varepsilon^2$.

$\sigma_\varepsilon^2 = \mathrm{Var}\left(\varepsilon^{\text{WiFi}}\right)$ denotes the measurement noise variance associated with the WiFi sensor. In practice, $\left\|\vec{\beta}_i\right\|_2^2 \ll 1$ for any sufficiently dense dataset (recall that $\left\|\vec{\beta}_i\right\|_1 = 1$).

Observe that this formulation is free of any restrictions on fingerprint uniqueness, and is therefore equally applicable to both sparse and dense signal environments.

A. Relationship to Gaussian Processes

Here we develop a few intuitions about key differences between our measurement model and Gaussian processes. In Gaussian processes [1], [23] the model fit to the WiFi measurements, as a function of z_i, has mean:

$$h_i^{\text{GP}}(\mathbf{X}) = \vec{k}_i^{\text{T}}\left(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}\right)^{-1}\vec{z}^{\text{WiFi}}$$

$$h_i^{\text{WiFi}}(\mathbf{X}) = \vec{\beta}_i^{\text{T}}\vec{z}^{\text{WiFi}}$$

σ_ε^2 is measurement noise variance associated with the WiFi sensor. $\vec{k}_i$ and $\mathbf{K}$ come from the choice of kernel weighting function.

The comparison to GraphSLAM is most clear when $[\vec{w}_1, \vec{w}_2, \ldots] \propto \mathbf{K}$, a square matrix whose j^{th} column is $\vec{k}_j \propto \vec{w}_j$. The two key differences in the measurement model presented here are the omission of the $\left(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}\right)^{-1}$ term and exclusion of z_i^{WiFi} from $\vec{z}^{\text{WiFi}}$ in the weighted average, i.e. $\left[\vec{\beta}_i\right]_i = 0$.

Omission of $\left(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}\right)^{-1}$

The $\left(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}\right)^{-1}$ term can be thought of as a whitening transform on the weighted observations and their weights. If, for example, ten observations appear at the same location $\vec{x}$ and the same value z, they would collectively only be given one "vote" in the weighted average, rather than ten. This makes sense when attempting to make statistically consistent function value estimates over large distances. Only at small scales can signal strengths be averaged meaningfully without physical modeling of the surrounding materials. At these scales, giving (nearby) past measurements each an "equal vote" provides a larger sample size with which to predict future measurements, which is the methodology employed within the GraphSLAM formulation. Formally, treating past measurements in this way is equivalent to the approximation of $e^{-\frac{1}{2\tau^2}} \ll \sigma_\varepsilon^2$ for a Gaussian Process, i.e. the measurement noise dominates innate signal variance. At wireless frequencies this is a reasonable assumption[2].

Furthermore, the size of $\mathbf{K}$ grows quadratically in the size of the dataset (number of measurement locations, N). The need to invert $\mathbf{K}$ when computing h and its derivatives makes each iteration a $\mathrm{O}\left(N^3\right)$ operation [24]. This is the main reason that it is difficult to scale Gaussian process techniques to larger datasets and omitting this term in the interpolation allows GraphSLAM to achieve a $\mathrm{O}\left(N^2\right)$ asymptotic runtime

[2]Error in measured signal strength is tightly coupled to innate signal variance by the dynamics of the environment. Distinguishing measurement noise from signal variance would require extensive prior information of building materials, population distributions, etc.

and in turn makes the GraphSLAM technique easier to scale to larger datasets.

It should be noted that both GraphSLAM and Gaussian process latent variable models can improve their runtime complexities by means of sparsification or other approximation methods [25], [26].

Exclusion of z_i^{WiFi}

The second key difference in interpolation methods is that our proposed model fit always excludes z_i^{WiFi} from $\vec{z}^{\text{WiFi}}$ when computing the weighted average for h_i^{WiFi}. Intuitively, we are always attempting to determine the model fit, or self-consistency, of observing certain measurements. In both GraphSLAM and Gaussian processes, model fit/measurement likelihood is modeled as Gaussian random variables/vectors and as such are defined by their mean and (co)variance. That is, we wish to determine the fit of z_i^{WiFi}. The fit of z_i^{WiFi} will be determined by a certain distribution p. In this setting, it wouldn't make sense that we would use z_i^{WiFi} itself to compute the parameters of p.

As a consequence of including z_i^{WiFi} in its own model fit definition, for any fixed τ, the latent space optimization of GP-LVM can always improve model fit by simply moving all points away from each other. To circumvent this behavior, GP-LVM methods almost always require explicit "signal strength $\rightarrow$ location" back constraints or carefully selected priors [1], [27], [28]. The GraphSLAM approach, on the other hand, will require no hard constraints. Excluding z_i^{WiFi} naturally causes measurements to be "attracted" to similar neighboring measurements. This makes GraphSLAM suitable to a wider range of real-world environments, including those where wireless signatures are not rich enough to guarantee uniqueness but still provide enough information to augment an existing SLAM implementation.

V. NON-LINEAR LEAST SQUARES

One of the primary appeals of GraphSLAM methods in general is that minimizing the negative log posterior reduces to standard non-linear least squares [19], giving GraphSLAM access to a vast set of widely used and well-studied techniques for its optimization.

We need only $h^{\text{pedometry}}$, h^{gyro}, h^{WiFi}, together with $\text{Var}\left(\varepsilon^{\text{pedometry}}\right)$, $\text{Var}\left(\varepsilon^{\text{gyro}}\right)$, $\text{Var}\left(\varepsilon^{\text{WiFi}}\right)$ to formulate the least squares problem:

$$-\log \prod_i \mathrm{P}_{Z_i}\left(z_i|\mathbf{X}\right)$$

$$= \frac{1}{2}\sum_i \left[z_i - h_i\left(\mathbf{X}\right)\right]^{\mathrm{T}}\left[\text{Var}\left(\varepsilon_i\right)\right]^{-1}\left[z_i - h_i\left(\mathbf{X}\right)\right]$$

Depending on the application/environment/domain, we can assume a uniform prior and simply maximize the likelihood, or we can add any number of Gaussian priors (e.g. inertial priors or smoothness constraints) in a straightforward way. For the experiments of section VI we have assumed uniform priors and maximize the data likelihood directly.

A. Solvers

In general, non-linear least squares is a well-studied problem in numerical optimization communities [29] and any number of solvers can be used instead. Methods such as gradient/steepest descent, Levenberg-Marquardt, BFGS [30] and many Conjugate gradient based methods [31] are readily available and can be applied directly.

Typical solvers depend on local linearization to iterate toward an optimum [29], [15]. Let $\mathbf{J}_{h_i}\left(\mathbf{X}\right)$ denote the derivatives of h with respect to each state variable in $\mathbf{X}$ (each row of $\mathbf{J}$ is a gradient of h). For example, if this Jacobian is known, any initial guess $\mathbf{X}_0$ can be iteratively refined by solving

$$\mathbf{X}_{\text{new}} := \mathbf{X}_0 + \left[\sum_{h_i}\mathbf{J}_{h_i}^{\mathrm{T}}\mathbf{\Omega}_{h_i}\mathbf{J}_{h_i}\right]^{-1}\left[\sum_{h_i}\mathbf{J}_{h_i}^{\mathrm{T}}\mathbf{\Omega}_{h_i}\left(z_i - h_i\left(\mathbf{X}_0\right)\right)\right]$$

$$\mathbf{\Omega}_{h_i} = \left[\text{Var}\left(\varepsilon_i\right)\right]^{-1}$$

with $\mathbf{J}_{h_i}$ evaluated at $\mathbf{X}_0$ on each iteration. This is known as Gauss-Newton iteration.

Our results in section VI-B are obtained using standard Gauss-Newton iteration. In our case z and h are always scalar valued so $\mathbf{J}_{h_i}\left(\mathbf{X}\right) = \nabla h_i$ is a row vector and $\mathbf{\Omega}_{h_i} = \sigma_{h_i}^{-2}$ is simply a scalar. Each iteration, then, is equivalent to solving the overconstrained matrix system of the form $\mathbf{A}\vec{\Delta} = \vec{b}$:

$$\begin{bmatrix} - & \sigma_{h_1}^{-1}\nabla h_1 & - \\ - & \sigma_{h_2}^{-1}\nabla h_2 & - \\ & \vdots & \\ - & \sigma_{h_N}^{-1}\nabla h_N & - \end{bmatrix}\vec{\Delta} = \begin{bmatrix} \sigma_{h_1}^{-1}\left(z_1 - h_1\right) \\ \sigma_{h_2}^{-1}\left(z_2 - h_2\right) \\ \vdots \\ \sigma_{h_N}^{-1}\left(z_N - h_N\right) \end{bmatrix}$$

and updating $\mathbf{X}_{\text{new}} := \mathbf{X}_0 + \vec{\Delta}$.

B. State Space

The convergence characteristics of GraphSLAM depend on the linearizability of h. We make some effort to transform our state space to improve linearization. In certain settings (e.g. [32]) reformulation of a GraphSLAM state space has lead to dramatically improved performance and result quality. So far, all three of the state-to-measurement mappings $h^{\text{pedometry}}$, h^{gyro} and h^{WiFi} are non-linear functions if the world state is represented as $\mathbf{X} = \{\vec{x}_{t_1}, \vec{x}_{t_2}, \ldots\}$, describing "robot location" at each point in time.

Due to the exponential terms created by the interpolation weight kernel, h_{WiFi} will be non-linear regardless of the state space parametrization. We choose to solve for our state space in terms of the headings $\{\phi_1, \phi_2, \ldots, \phi_{N-1}\}$ and distances $\{d_1, d_2, \ldots, d_{N-1}\}$ between each WiFi scan, e.g. $\vec{x} = \begin{bmatrix} \sum d\cos\left(\phi\right) \\ \sum d\sin\left(\phi\right) \end{bmatrix}$. Then,

$$h_{\text{pedometry}}\left(\mathbf{X}\right) = d_i$$

$$h_{\text{gyro}}\left(\mathbf{X}\right) = \phi_{i+1} - \phi_i$$

which linearize trivially with infinite radius of convergence. This allows us to eliminate linearization error in all but one of the sensors.

To compute derivatives $\nabla_{\{\vec{d},\vec{\phi}\}} h_{\text{WiFi}}(\mathbf{X})$:

$$\nabla_{\{\vec{d},\vec{\phi}\}} h_{\text{WiFi}}(\mathbf{X}) = \left[\nabla_{\{\vec{x}\}} h_{\text{WiFi}}(\mathbf{X})\right] \mathbf{J}_{\{\vec{x}\}}\left(\left\{\vec{d},\vec{\phi}\right\}\right)$$

$$\nabla_{\{\vec{x}\}} h_{\text{WiFi}}(\mathbf{X}) = \sum_j \left[\vec{\beta}_i\right]_j (z_j - h_i) \frac{-1}{2\tau^2}\left(\nabla_{\{\vec{x}\}} \|\vec{x}_j - \vec{x}_i\|^2\right)$$

The gradient of $h_{\text{WiFi}}(\mathbf{X})$ under this 'heading and distance' parametrization is fast to compute in practice. This is because the columns of $\mathbf{J}_{\{\vec{x}\}}\left(\left\{\vec{d},\vec{\phi}\right\}\right)$ are always constant valued with leading zeros, and elements of $\nabla_{\{\vec{x}\}} \|\vec{x}_j - \vec{x}_i\|^2$ are all zero except for those corresponding to the $\frac{\partial}{\partial \vec{x}_i}$ or $\frac{\partial}{\partial \vec{x}_j}$ elements.

VI. EXPERIMENTAL RESULTS

A. Data

To evaluate the algorithms proposed in this paper, we used a trace of 536 WiFi scans captured over 17 minutes across a 60m × 10m area of one floor of a university building. The trajectory covers about 1.2 km of travel distance. This dataset contains corresponding pedometry data, readings from a MEMS gyroscope, and an accompanying LIDAR-derived ground truth. The ground truth has been derived by processing the LIDAR, pedometry, and gyroscope measurements through off-the-shelf LIDAR SLAM.

Maximizing the likelihood of the dataset over parameter τ yields an optimal value of 2.2 m (e.g. the average distance to a wall in all directions is roughly 4.4 m for 95% of measurement locations), however in our experiments any values from $\tau \approx 1.5$ m through $\tau \approx 3.5$ m all produce good results.

B. Results

The ground truth is accurate to about 10 cm. The ground truth and LIDAR are used only to evaluate the results. GraphSLAM requires no labeled data.

We use the state space parametrization of section V-B. Figure 3 is a qualitative visualization of the performance of the GraphSLAM method. By nature of the sensors, GraphSLAM output is displayed in units of steps; ground truth is plotted in meters. Notice that the results show straight halls, despite the GraphSLAM method containing absolutely no explicit shape prior. This adds further confidence to the notion that signal-strength SLAM can be achieved in a completely unsupervised manner, without relying on trajectory priors.

The sensor used in our experiment measures pedometry in units of steps. The ground truth is collected using a laser-based technique that produces coordinates measured in meters. The location of the nodes in GraphSLAM are in a different reference frame than those in the ground-truth trajectory. To report error in standard units, as in [1], we compute localization accuracy with the subjective-objective technique of [33]: We characterize "how accurately a person detects returning to a previously visited location". For each time t_i during our trajectory, we have an inferred 'subjective' location $\vec{\tilde{x}}_{t_i}$ from GraphSLAM and a true 'objective' location $\vec{x}_{t_i}^*$ for the same timestamp in the ground truth. For every objective
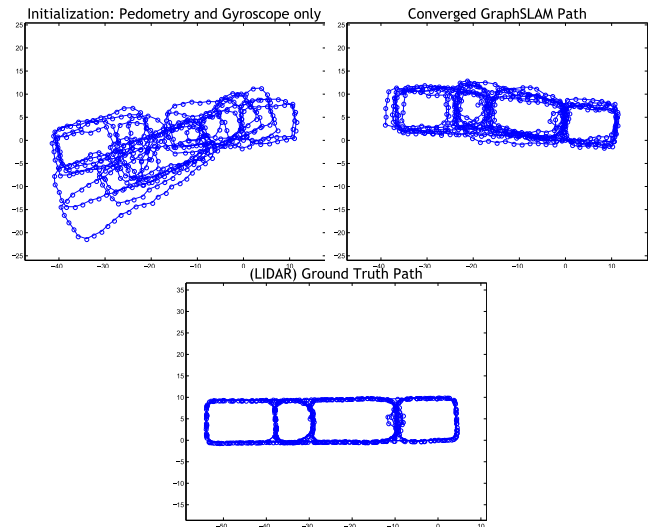

Fig. 3. Comparison of GraphSLAM initialization (top-left), and resulting optimized GraphSLAM posterior (top-right). The unaligned ground truth trajectory is provided for reference (bottom).

location $\vec{x}_{t_i}^*$ we denote its 'objective neighbors' to be those points within r meters of $\vec{x}_{t_i}^*$. For each objective neighbor of $\vec{x}_{t_i^*}^*$, the 'subjective' location $\vec{\tilde{x}}_{t_i}$ has a corresponding 'subjective neighbor' $\vec{\tilde{x}}_{t_i^*}$. We scale the GraphSLAM output so that the total distance traveled matches the ground truth's total distance traveled, and define localization error to be the mean of the distance $\left\|\vec{\tilde{x}}_{t_i} - \vec{\tilde{x}}_{t_i^*}\right\|_2$ across all pairs (t_i, t_i^*). r has been chosen to be the mean distance traveled between consecutive WiFi scans.

With this metric, we achieve a mean localization error of 2.18 meters. For comparison, using the same metric, the mean localization error from only pedometry and gyroscope without WiFi is 7.10 m.

We presume these experiments to represent a lower bound on result quality for this algorithm. As a non-linear least squares problem in general, we expect to benefit from more robust solvers, e.g. Levenberg-Marquardt, simulated annealing, etc. Given that the GraphSLAM framework has been well-studied in the SLAM community, any number of solvers are likely to improve performance or result quality further.

VII. CONCLUSIONS

We have reformulated signal strength SLAM into an instance of GraphSLAM. In doing so, we have improved scalability, reduced runtime complexity, relaxed limitations on WiFi density/richness, and removed all shape priors. Experimental results on real-world data demonstrate the effectiveness of using GraphSLAM approach to solve this problem. Future work is likely to explore 3D variants of the WiFi SLAM problem, multi-agent extensions, time-of-arrival/round-trip-time sensor models, improved initialization techniques as well as more specialized solvers.

REFERENCES

[1] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," *Proceedings of IJCAI 2007*, pp. 2480–2485, 2007.

[2] A. Howard, S. Siddiqi, and Sukhatme, "An experimental study of localization using wireless ethernet," in *Field and Service Robotics*. Springer, 2006, pp. 145–153.

[3] M. Quigley, D. Stavens, A. Coates, and S. Thrun, "Sub-Meter Indoor Localization in Unmodified Environments with Inexpensive Sensors."

[4] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proc. of Robotics Science and Systems*, vol. 442, 2006.

[5] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[6] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only slam for robots operating cooperatively with sensor networks," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 2078–2084.

[7] H. Lim, L. Kung, J. Hou, and H. Luo, "Zero-configuration, robust indoor localization: theory and experimentation," in *Proceedings of IEEE Infocom*, 2006, pp. 123–125.

[8] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[9] Y. Chan, W. Tsui, H. So, and P. Ching, "Time-of-arrival based localization under NLOS conditions," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 17–24, 2006.

[10] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEE infocom*, vol. 2, 2000, pp. 775–784.

[11] A. Haeberlen, E. Flannery, A. Ladd, A. Rudys, D. Wallach, and L. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 70–84.

[12] J. Letchner, D. Fox, and A. LaMarca, "Large-scale localization from wireless signal strength," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 15.

[13] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "ARIADNE: a dynamic indoor signal map construction and localization system," in *Proceedings of the 4th international conference on Mobile systems, applications and services, June*, 2006, pp. 19–22.

[14] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-mapping in 802.11 location systems," *UbiComp 2005: Ubiquitous Computing*, pp. 87–104.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. [Online]. Available: http://www.probabilistic-robotics.org

[16] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Proceedings of the Robotics: Science and Systems Conference, Atlanta, USA*, 2007.

[17] E. Eade and T. Drummond, "Unified loop closing and recovery for real time monocular slam," in *Proc. European Conference on Computer Vision*, 2008.

[18] M. Milford and G. Wyeth, "Mapping a suburb with a single camera using a biologically inspired SLAM system," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, 2008.

[19] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, p. 403, 2006.

[20] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.

[21] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[22] J. Park and I. Sandberg, "Universal approximation using radial-basis-function networks," *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.

[23] C. Rasmussen, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71, 2006.

[24] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *The Journal of Machine Learning Research*, vol. 6, p. 1816, 2005.

[25] ——, "Learning for larger datasets with the Gaussian process latent variable model," in *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*, 2007.

[26] M. Walter, R. Eustice, and J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *The International Journal of Robotics Research*, vol. 26, no. 4, p. 335, 2007.

[27] S. Pan, J. Kwok, Q. Yang, and J. Pan, "Adaptive localization in a dynamic wifi environment through multi-view learning," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 1108.

[28] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283–298, 2008.

[29] G. Golub and C. Van Loan, *Matrix computations*. Johns Hopkins Univ Pr, 1996.

[30] C. Broyden, J. Dennis Jr, and J. More, "On the local and superlinear convergence of quasi-Newton methods," *IMA Journal of Applied Mathematics*, vol. 12, no. 3, p. 223, 1973.

[31] J. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[32] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 2262–2269.

[33] M. Bowling, D. Wilkinson, A. Ghodsi, and A. Milstein, "Subjective localization with action respecting embedding," *Robotics Research*, pp. 190–202, 2007.