Indoor Localization in Multi-Floor Environments with Reduced Effort

Hua-Yan Wang*, Vincent W. Zheng†, Junhui Zhao‡ and Qiang Yang†
* *Stanford University, Email: huayanw@cs.stanford.edu*
† *Hong Kong University of Science and Technology, Email: {vincentz,qyang}@cse.ust.hk*
‡ *NEC Labs China, Email: zhaojunhui@research.nec.com.cn*

***Abstract*—In pervasive computing, localizing a user in wireless indoor environments is an important yet challenging task. Among the state-of-art localization methods, fingerprinting is shown to be quite successful by statistically learning the signal to location relations. However, a major drawback for fingerprinting is that, it usually requires a lot of labeled data to train an accurate localization model. To establish a fingerprinting-based localization model in a building with many floors, we have to collect sufficient labeled data on each floor. This effort can be very burdensome. In this paper, we study how to reduce this calibration effort by only collecting the labeled data on one floor, while collecting unlabeled data on other floors. Our idea is inspired by the observation that, although the wireless signals can be quite different, the floor-plans in a building are similar. Therefore, if we co-embed these different floors' data in some common low-dimensional manifold, we are able to align the unlabeled data with the labeled data well so that we can then propagate the labels to the unlabeled data. We conduct empirical evaluations on real-world multi-floor data sets to validate our proposed method.**

***Keywords*-Reduced Calibration Effort, Wireless Localization, Multi-Floor Environment**

I. Introduction

With the proliferation of wireless technologies, indoor localization using wireless signal strength has attracted increasing interests from both research and industrial communities [1], [2]. Based on accurate location information, many useful services can be provided, such as object tracking, security control, *etc*. A number of localization methods have been proposed. Among them, fingerprinting is shown to be quite successful. In general, given sufficient labeled data in some environment, fingerprinting methods statistically learn the relations between the received signal strengths and the locations [3], [4]. However, getting a lot of labeled data can be very expensive. This drawback poses a major difficulty for fingerprinting methods in real-world applications. For example, to facilitate the indoor localization for a building with many floors, we have to collect sufficient labeled data and train a localization model at each floor. When there are many floors in the building, it will be very burdensome to collect labeled data for all the floors. In this paper, we are interested in studying how to reduce such effort. We show that, by using our proposed method, we only have to collect the labeled data on one floor. For the other floors, we only need to carry a wireless device and walk in an arbitrary way to collect some unlabeled data. After that, a sufficiently accurate localization model for the floors without labeled data can be obtained for the other floors, saving a significant amount of labeling effort.

Our idea is inspired by the observation that, although the wireless signals can be quite different, the floor-plans in a building are usually similar. As an example, a floor plan for our building is shown in Figure 1. Generally, the

Figure 1. An academic building at HKUST (partial)

wireless signal data are composed of the received signal strengths (RSS) from various access points (APs) in the environments. On different floors, we may detect different APs. Moreover, even for the same APs, we may receive different signal strength values at different floors. So the collected signal data from different floors can constitute different high-dimensional spaces, which can be described as high-dimensional manifolds [20]. Fortunately, although these high-dimensional data spaces vary across the floors, they are all constrained to some common low-dimensional representation, such as 2-D locations in a floor-plan. Therefore, if we can embed these different floors' data in some common low-dimensional manifold by using some correspondence-learning method, we are able to align the unlabeled data with the labeled data. We will show in Section IV-A that, by using a maximal-RSS criterion, we can identify such correspondences well. After that, we can propagate the labels from the labeled data to the unlabeled data and train a localization model for the floors with only unlabeled data. We can save a significant amount of effort in this way in multi-floor localization.

Our contributions are summarized as follows:

- We describe a new multi-floor indoor localization problem, and analyze the traditional fingerprinting solutions on this problem.

- We provide a reduced-effort solution by using the unlabeled data. Compared to other multi-floor localization work [11], [12], [13], our method does not require data labeling at every other floor, thus greatly reducing the calibration effort. Compared to other localization methods also using unlabeled data [5], our model is capable of handling a more difficult case when we have multiple signal datasets with the received signal strengths (*i.e.* data distributions) and the detected access points (*i.e.* data feature dimensions) being different.
- We validate our proposed method in real-world multi-floor environments.

II. Related Work

In recent years, there have been many studies on how to utilize the radio frequency values of wireless signals for indoor localization. In general, the proposed wireless localization methods fall into two main categories: *Propagation Models* and *Fingerprinting Models*. Propagation Models can benefit from the knowledge of radio propagation and also the access points' locations. By using trilateration or triangulation techniques, propagation models can compute the position for the mobile target [6], [7]. A drawback for such models is that, they cannot well handle the signal uncertainties. Fingerprinting models, also known as learning-based models, aim to use some data mining techniques, to discover the signal-location patterns [8]. Typical pattern descriptions include histogram [9], mixture of gaussian[10], or simply the mean value of signal strength at different locations [1]. As fingerprinting models can take the signal uncertainties into consideration, they are usually shown to have rather satisfying performance in localization. However, there are still some drawbacks for such models; and one main drawback of them is that they usually require much human calibration effort to build a localization model for a given environment. When the environment changes, the learned signal-location patterns may change as well, thus the already-built localization model may not work well anymore. Our work belongs to the category of fingerprinting models, but it goes beyond the standard fingerprinting methods by handling the environmental changes caused by the effects of multiple floors with reduced effort.

Some previous works also considered multiple floors in wireless localization. For example, Otsason *et al.* studied the GSM-based indoor localization by using wide signal-strength fingerprints [11]. They tested their model on three multi-floor buildings, and showed that it can effectively differentiate between floors and achieve a satisfying accuracy for within-floor localization. Letchner et al. proposed to use hierarchical Bayesian network for large-scale wireless localization [12]. For an indoor environment, they tested their model in a 7-floor building. Varshavsky et al. designed a system based on a GSM fingerprinting-based localization system know as SkyLoc, to identify which floor a mobile phone user is on in multi-floor buildings [13]. They studied several feature-selection methods for their system, and tested them in three multi-floor buildings. However, the above works required users to collect labeled data (*i.e.* fingerprints) at all the floors, thus are quite expensive for human calibration. In contrast, we only need to collect labeled data on *a single floor*, and collect unlabeled data on all other floors. It will greatly reduce the labeling cost. There is also some other work on studying how to track a user in a multi-floor building such as [14], but they typically require some other specific sensors instead of using only the wireless adapters.

Besides, there are some works discussed how to reduce the human labeling effort in wireless localization. For example, [15] and [16] only asked for unlabeled data to do localization; however, both of them required the access points' locations to be known in advance, which is not always available in practice due to security control. In contrast, we don't need to know the APs' locations. [17] and [18] addressed the environmental changes with time and devices in WiFi localization with minimal effort; however, both of them required at least some labeled data in the new environments (*i.e.* either time or devices), while in our current model, we don't need to collect any labeled data in the new environments (*i.e.* floors).

III. Localization for Multi-Floor Environments

In our proposed localization method, we only require the users to collect the labeled data (*i.e.* fingerprints) on one floor, and collect unlabeled data on other floors. Then, we will build a localization model for the floors with unlabeled data. For clarity, we will use two floors as an example to demonstrate our idea in the following sections[1]. In particular, we will take as input a source floor having sufficient labeled data and a target floor having only unlabeled data. After training, we assign labels for the unlabeled data in the target floor, and thus build a localization model (such as nearest neighbor [1], *etc.*). Finally, we will test the model on some held-out test data (labeled) on the target floor. In applying the localization model in real time, for a multi-floor building environment, when a query signal vector is sent to our localization system, we will compare its detected APs with each floor's AP list (as each floor's AP list is different) and thus determine which floor the real-time signal vector may belong to. After that, we will use that floor's localization model to output the predicted location.

Assume that, in some floor $\mathbf{A}$, we carry a wireless device and collect a set of fingerprints from D_A access points. Therefore, we will have the labeled data ($\mathbf{X}^A_{trn}$, $\mathbf{Y}^A_{trn}$) in floor $\mathbf{A}$ for training. We call $\mathbf{A}$ the source floor. Here, $\mathbf{X}^A_{trn}$ is a $N_1 \times D_A$ matrix with each row as a received signal strength (RSS) vector and each column corresponding to an

[1]Our model can also naturally extend to cope with more than two floors at a time, but in our real-world experiments, we didn't find that it necessarily brings any performance improvement.

access point. $\mathbf{Y}^A_{trn}$ is a $N_1 \times 2$ matrix, with each row as the 2-D location where the corresponding RSS vector in $\mathbf{X}^A_{trn}$ is collected. Similarly, in a target floor **B**, we will collect some unlabeled signal data from D_B access points for training. Denote the data as $\mathbf{X}^B_{trn}$, which is a $N_2 \times D_B$ matrix. We also collect another set of test data ($\mathbf{X}^B_{tst}$, $\mathbf{Y}^B_{tst}$) for the target floor. Here, $\mathbf{X}^B_{tst}$ is a $N_3 \times D_B$ matrix and $\mathbf{Y}^B_{tst}$ is a $N_3 \times 2$ matrix. These test data are held out for evaluation.

In practice, the signal data for two different floors are generally different in both received signal strengths and detected access points. In other words, the columns in $\mathbf{X}^A_{trn}$ and $\mathbf{X}^B_{trn}$ are generally different as the floor is changed, such that the labeled data $\mathbf{X}^A_{trn}$ and their labels $\mathbf{Y}^A_{trn}$ cannot be directly used by putting them together. This is the difficulty for the multi-floor localization problem. Fortunately, we find that, although the signal data can be different, the underlying floor-plans are usually similar. Furthermore, in a building, different floors usually share *some* access points in the practical scenarios. The correspondences between these shared access points can be established by examining the MAC addresses of the access points recorded by the receiving devices. We will manage to make use of both observations to facilitate our reduced-effort localization in the multi-floor environment.

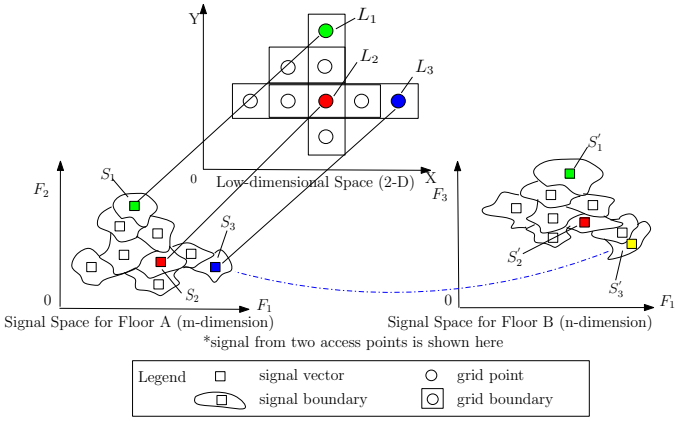

Figure 2. Multi-manifold to some latent space (e.g. 2-D location space)

As shown in Figure 2, the two sets of signal vectors $\mathbf{X}^A_{trn}$ and $\mathbf{X}^B_{trn}$ can constitute two manifolds in two high-dimensional spaces; at the same time, they are constrained to some low-dimensional space such as a 2-D location floor-plan in our case. For example, the signal vector S_1 for floor **A** and the signal vector S'_1 for floor **B**, although in different signal spaces ($\{F1, F2\}$ vs. $\{F1, F3\}$), they both correspond to a 2-D location L_1. Because we do not have any labels on floor **B**, we cannot align the signal S'_1 to location L_1. Similarly, we cannot align S'_2 to L_2. Fortunately, we find that signal vector S_3 and S'_3 have some inherent correspondences, *i.e.* they both have the maximal signal strengths from the access point F_1 in their own

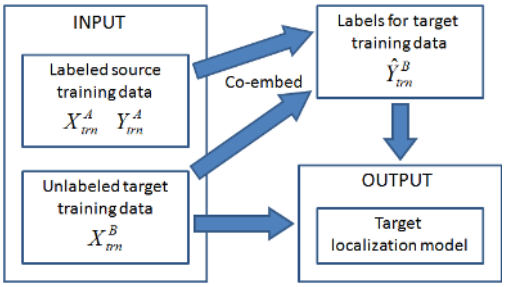

Figure 3. A flowchart of our proposed method.

datasets. As the signal strengths will attenuate as the distance from the device to the access point increases, we may infer that their 2-D locations are very close, or sometimes even the same. By using such a correspondence, we can propagate the label (*i.e.* L_3) from S_3 to S'_3 (in a sophisticated way as we will shown in the following sections). Therefore, by using the neighborhood information, we can further propagate the needed label information to S'_1 and S'_2, finally aligning them to L_1 and L_2. The above procedure constitutes our idea to *co-embed* both the labeled and unlabeled signal data from the two floors into a common latent space (such as the 2-D floor-plan) for label propagation. Our algorithm flowchart is shown in Figure 3. First, we embed $\mathbf{X}^A_{trn}$ and $\mathbf{X}^B_{trn}$ into a common latent space, so as to infer the labels $\widehat{\mathbf{Y}}^B_{trn}$ for $\mathbf{X}^B_{trn}$. Then, we learn a localization model from $\mathbf{X}^B_{trn}$ and $\widehat{\mathbf{Y}}^B_{trn}$, which is readily used for localization in floor **B** (evaluated on test data $\mathbf{X}^B_{tst}$ in our experiments).

IV. Low-cost Localization by Co-embedding

To make the co-embedding possible, we will introduce two intra-manifold graphs and one inter-manifold graph, given the data on two floors[2]. Recall that each floor's signal data naturally constitute one manifold in some high-dimensional space. To describe such a manifold, we build an intra-manifold graph encoding the connections (or distances) among the RSS vectors. Hence, for two floors, we can formally describe them as two intra-manifold graphs. As our aim is to co-embed these two manifolds in some common low-dimensional space for label propagation, we will need to build some connections between these two manifolds. We achieve this by modeling an inter-manifold graph, which can be initialized by some correspondences between the RSS vectors on both manifolds. As discussed above, we will assign such a correspondence for two RSS vectors having the maximal signal strengths from some shared AP respectively on both floors. After that, we will further refine the inter-manifold graph by making used of the intra-manifold graphs.

Having the inter-manifold graph and two intra-manifold graphs, we are now ready to co-embed both floors' data.

[2]Hence, if we want to deal with m floors ($m > 2$) at a time, we will have m intra-manifold graphs and $\binom{m}{2}$ inter-manifold graphs.

There can be two alternatives to do this: (1) *Co-embedding with location constraints*, in which we will use the known locations $\mathbf{Y}^A_{trn}$ as constraints and co-embed the two floors' data into a 2-D space. In this case, the label information $\mathbf{Y}^A_{trn}$ will be used in both co-embedding and label propagation. (2) *Co-embedding without location constraints*, in which we will ignore $\mathbf{Y}^A_{trn}$ and co-embed the two floors' data into some arbitrary high dimensional space (thus can be higher than 2-D). In this case, the label information $\mathbf{Y}^A_{trn}$ will only be used in label propagation.

A. Build Graphs for Co-embedding

In order to constrain the embedding of the RSS vectors, we build graphs within each of the datasets $\mathbf{X}^A_{trn}$ and $\mathbf{X}^B_{trn}$ as well as between them. For the intra-dataset graphs, the distances between the RSS vectors in $\mathbf{X}^A_{trn}$ (as well as $\mathbf{X}^B_{trn}$) can be readily computed (*e.g.* by the Euclidean distance). Thus we can simply connect each RSS vector with its K nearest neighbors (which results in a binary graph matrices $\mathbf{W}^A_{N_1\times N_1}$ and $\mathbf{W}^B_{N_2\times N_2}$. For the inter-dataset graph, distances between the RSS vectors across $\mathbf{X}^A_{trn}$ and $\mathbf{X}^B_{trn}$ cannot be directly computed due to different feature spaces. To build the connections between the datasets, we propose to use a maximal-RSS criterion to get the correspondences for initializing the inter-dataset graph.

- ***Maximal-RSS Criterion***: For each shared access point, we connect the two RSS vectors (from $\mathbf{X}^A_{trn}$ and $\mathbf{X}^B_{trn}$ respectively) with maximum received signal strength from that access point.

The resulted graph (with binary graph matrix $\mathbf{W}^{AB}_{N_1\times N_2}$) is generally too sparse to effectively relate the embedding of two datasets, because the number of shared access points is usually much less than number of RSS vectors in a densely sampled dataset. We therefore use the following iterative rule to enhance the inter-dataset graph by incorporating the intra-dataset graphs:

$$\mathbf{W}^{AB} \leftarrow \mathbf{W}^A\mathbf{W}^{AB}\mathbf{W}^B. \tag{1}$$

Each such update essentially propagates and enhances the inter-dataset connections according to the intra-dataset connections. In practice, (1) is executed a sufficient number of times to allow $\mathbf{W}^{AB}$ effectively relate the two datasets.

Let $\{\mathbf{p}^{(k)}\}_{k=1}^K \subset \mathbb{R}^{N_1+N_2}$ denote the embedding coordinates of the source and target training set (in K-dimensional latent space). For $i = 1, \cdots, N_1$, $\mathbf{p}^{(k)}(i)$ is the k-th coordinates of the i-th RSS vector in $\mathbf{X}^A_{trn}$, and for $j = 1, \cdots, N_2$, $\mathbf{p}^{(k)}(N_1 + j)$ is the k-th coordinates of the j-th RSS vector in $\mathbf{X}^B_{trn}$. The co-embedding should be constrained by the following regularization term in order to be smooth with respect to the inter- and intra-dataset graphs:

$$\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}(\mathbf{p}^{(k)}(i)-\mathbf{p}^{(k)}(N_1+j))^2\ \mathbf{W}^{AB}(i,j) = {\mathbf{p}^{(k)}}^\top \mathbf{L}^{AB}\mathbf{p}^{(k)}, \tag{2}$$

where $\mathbf{L}^{AB}$ is a graph Laplacian matrix, defined as

$$\mathbf{L}^{AB}_{(N_1+N_2)\times(N_1+N_2)} = \begin{pmatrix} \mathbf{D}^{AB}_{N_1\times N_1} & -\mathbf{W}^{AB}_{N_1\times N_2} \\ -(\mathbf{W}^{AB})^\top_{N_2\times N_1} & \mathbf{D}^{BA}_{N_2\times N_2} \end{pmatrix}. \tag{3}$$

Here, $\mathbf{D}^{AB}_{N_1\times N_1}$ is a diagonal matrix with diagonal elements $\{\sum_j \mathbf{W}^{AB}(i,j)\}_{i=1}^{N_1}$, and $\mathbf{D}^{BA}_{N_2\times N_2}$ is a diagonal matrix with diagonal elements $\{\sum_i \mathbf{W}^{AB}(i,j)\}_{j=1}^{N_2}$.

In co-embedding, the inter-dataset graph is combined with the intra-dataset graphs with some relative weight μ. Finally, the composite graph Laplacian is

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}^A & 0 \\ 0 & \mathbf{L}^B \end{pmatrix} + \mu\, \mathbf{L}^{AB}_{(N_1+N_2)\times(N_1+N_2)} \tag{4}$$

where $\mathbf{L}^A$ and $\mathbf{L}^B$ are the graph Laplacian of the intra-dataset graphs, defined similarly with (3). So the constraint imposed by the graphs on the co-embedding can be summarized in one term ${\mathbf{p}^{(k)}}^\top\mathbf{L}\mathbf{p}^{(k)}$ for each dimension $k \in \{1, \cdots, K\}$.

B. Co-embedding with Location Constraints

Given the 2-D location labels $\mathbf{Y}^A_{trn}$, we can use the 2-D floor-plan as the common low-dimensional embedding space, and the image of each RSS vector in $\mathbf{X}^A_{trn}$ is constrained by its 2-D location [19] (as given in $\mathbf{Y}^A_{trn}$). Specifically, the optimal embedding should satisfy:

$$\begin{aligned}\widehat{\mathbf{p}^{(k)}} = \arg\min_{\mathbf{p}^{(k)}}[&(\mathbf{p}^{(k)} - \mathbf{y}^{(k)}_P)^\top\mathbf{J}_P(\mathbf{p}^{(k)} - \mathbf{y}^{(k)}_P) \\ &+ \gamma\ {\mathbf{p}^{(k)}}^\top\mathbf{L}\mathbf{p}^{(k)}], \quad k = 1, 2\end{aligned} \tag{5}$$

where

$$\mathbf{J}_P = \begin{pmatrix} \mathbf{I}_{N_1\times N_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}_{(N_1+N_2)\times(N_1+N_2)}, \tag{6}$$

and for $k = 1, 2$, we define $\mathbf{y}^{(k)}_P \in \mathbb{R}^{N_1+N_2}$ whose first N_1 elements are the known coordinates given in $\mathbf{Y}^A_{trn}$, and the next N_2 elements can be arbitrary values. The user-specified parameter γ controls the relative strength of the graph regularization and the known location constraints. We will study its impact in the experiments.

It turns out that (5) can be solved in closed form:

$$\widehat{\mathbf{p}^{(k)}} = (\mathbf{J}_P + \gamma\mathbf{L})^{-1}\mathbf{J}_P\mathbf{y}^{(k)}_P, \quad k = 1, 2 \tag{7}$$

Hence, the signal data from floors **A** and **B** are embedded in a common low-dimensional space. Then we can propagate the labels by assigning each RSS vector in $\mathbf{X}^B_{trn}$ the location label of its nearest neighbor from $\mathbf{X}^A_{trn}$ in the low-dimensional space.

C. Co-embedding without Location Constraints

One possible concern for co-embedding with location constraints can be that, it may unnecessarily restrict the embedding in a 2-D space, and thus unclear whether other embeddings in higher-dimensional spaces could be better to recover the relations between the RSS vectors. So we also

study the case of co-embedding without location constraints. Similar to the co-embedding method introduced in the last section, we now leave alone the locations $\mathbf{Y}^A_{trn}$ and the objective function becomes:

$$\widehat{\mathbf{p}^{(k)}} = \arg\min_{\mathbf{p}^{(k)}} {\mathbf{p}^{(k)}}^\top \mathbf{L}\mathbf{p}^{(k)}. \quad (8)$$

In order the prevent this objective function value from going arbitrarily small, we will impose some scale constraint on the embedding

$${\mathbf{p}^{(k)}}^\top \mathbf{p}^{(k)} = 1. \quad (9)$$

Fortunately, this optimization problem is well known to have the solution as smallest eigenvalues (excluding the zero eigenvalue) [20]. In other words, if we want to find a K dimensional embedding, we simply take the second to the $(K+1)$-th eigenvectors of $\mathbf{L}$. As in the last section, we can assign each RSS vector in $\mathbf{X}^B_{trn}$ the location label of its nearest neighbor from $\mathbf{X}^A_{trn}$ in the low-dimensional space.

D. Algorithm and Complexity Analysis

Here we summarize our multi-floor localization algorithm:

Offline Phase: Densely sampled labeled data is collected in one floor (source floor).

Online Calibration and Learning: When we move to another floor (target floor) in the same building with almost identical layout, we carry a wireless device and walk around in an arbitrary manner to collect unlabeled data. Then we co-embed the labeled data collected in the offline phase and the unlabeled data collected in online calibration phase. There can be two ways for co-embedding: co-embedding with location constraints in Equation (5) or without location constraints in Equation (8). After co-embedding, we can propagate the labels to the unlabeled data by finding the nearest neighbors in the common low-dimensional space. Finally, we have labeled data for the target floor.

Online Localization: We learn a localization model with the newly obtained labeled data for the target floor. Various techniques can be used to learn such a localization model. For example, we can use a nearest neighbor method: given a RSS vector from the target floor for testing, we find its nearest neighbor in the obtained labeled dataset for the target floor, and assign the corresponding location label to it.

Most computation cost of the procedure lies in computing the eigen-decomposition and inversion of the matrices in co-embedding with/without location constraints respectively, which are generally in the order of $O(n^3)$, where n is the number of RSS vectors in our situation.

V. Experiments

A. Datasets

To validate our approaches, we collected three sets of wireless signal data in an academic building at our university[3], where each dataset was collected in one floor with a size of around $60m \times 40m$ as shown in Figure 1. Each floor is discretized into 118 $1.5m \times 1.5m$ grids on the hallways. The dataset descriptions can be found in the following table. We note that, dataset 2 shares 91 APs with dataset 1; dataset 3 shares 51 APs with dataset 1 and 55 APs with dataset 2.

	# of different APs	# Samples
Dataset 1	118	2080
Dataset 2	120	1864
Dataset 3	114	2513

Among these datasets, we can take any dataset as a source floor and another one as a target floor, thus simulating 6 calibration tasks. The target dataset is (randomly) divided into several halves, where one half is used as *unlabeled* training data and the other half is used as held-out test data for evaluating the localization accuracy. The following results are all based on an average of 10-trial experiments.

B. Impact of the Maximal-RSS criterion

In order to relate the source and target datasets, we exploit the assumption (*i.e.* the Maximal-RSS criterion) that, for a given access points, maximal RSS are received at close locations in different datasets. In this way, we can connect the data vectors with maximal RSS values from two datasets for each shared AP. To see how such a criterion performs on real-world datasets, we plot the learned inter-dataset graphs (denoted as $\mathbf{W}^{AB}$) by using Eq.(1). As shown in Figure 4, the underlying (nearly) rectangle shapes are the experimental areas where we collected data (in different floors though). Each blue-line circle denotes a connection between two RSS vectors from both datasets in some location. The smaller a circle is, the closer the corresponding (2-D) locations where we collected the two connected RSS vectors are. As we can see, the graph between dataset 1 and dataset 2 has more connections/circles, which shows that the learned inter-manifold graph is better. This coincides with the fact that these two datasets share more APs and thus may have higher performance in co-embedding.

C. Impact of the model parameters

There are some parameters in our approach that we need to set manually, including the weight of inter-dataset graph versus intra-manifold graph (denoted as μ), the number of nearest neighbors in building the intra-dataset graphs (denoted as # NN), and the weight of graph constraints versus location constraints (denoted as γ) (only in location-constrained embedding). In all our experiments, we measure the performance by the percentage of correct predictions (*i.e.* accuracy) in the test set within some error distance; here, we set it as 5 meters, and we will study different error distances in Section V-D). Here, we use dataset 1 as source and dataset 2 as target for illustration for Figures 6, 7 and 8; we also

[3]We could not access the other multi-floor datasets used in the previous works [11], [12], [13] after several requests.

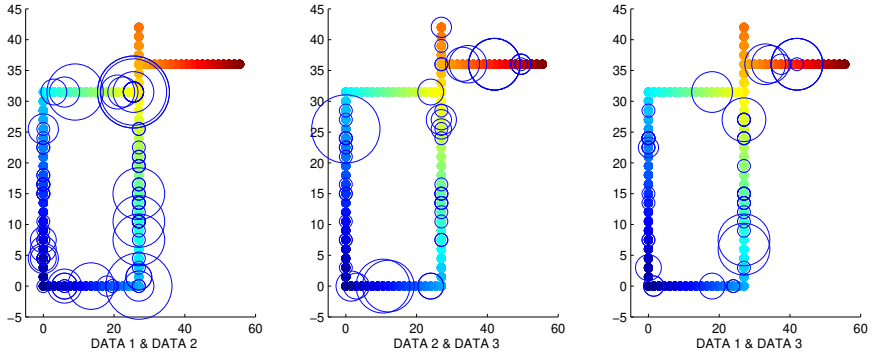

Figure 4. The qualities of the resulting inter-manifold graphs. The x-axis and y-axis denote the 2-D location coordinates (unit: meter). A circle is placed at some location where there is an inter-dataset connection for two RSS vectors, and the size of the circle indicates the connection quality.

Table I
RESULTS FOR USING DIFFERENT PAIRS OF FLOORS. WE REPORT THE (ACCURACY $\pm$ STANDARD DEVIATION).

# SHARED APS	S TO T	WITH CONS.	WITHOUT CONS.	BASELINE-1 [1]	BASELINE-2 [5]
91	1 TO 2	$65 \pm 3\%$	$69 \pm 2\%$	$11 \pm 1\%$	$7 \pm 1\%$
91	2 TO 1	$64 \pm 3\%$	$66 \pm 3\%$	$6 \pm 2\%$	$7 \pm 1\%$
51	1 TO 3	$55 \pm 5\%$	$57 \pm 4\%$	$8 \pm 4\%$	$8 \pm 1\%$
51	3 TO 1	$51 \pm 3\%$	$55 \pm 5\%$	$9 \pm 3\%$	$8 \pm 1\%$
55	2 TO 3	$53 \pm 4\%$	$52 \pm 4\%$	$3 \pm 1\%$	$9 \pm 1\%$
55	3 TO 2	$56 \pm 5\%$	$54 \pm 3\%$	$9 \pm 1\%$	$8 \pm 1\%$

get similar results by using other pairs of datasets, but due to space limit, we will not report them.

In Figure 6, generally we observe simple unimodal behaviors of the parameter μ, # NN and γ except that if we set # NN too small or too large (beyond the scales in the figure), the method sometimes "crashes" and we get degenerated embeddings. For clearer illustration, we further plot the behavior of the co-embeddings with increasing μ in Figure 7 and increasing γ in Figure 8, respectively. As we can see from both the figures, as μ and γ approach their own unimodal points (e.g. $\mu = 0.1$ and $\gamma = 10$), the co-embeddings for two floors' data can align better.

D. Overall results

We tested our two methods, "co-embedding with location constraints" and "co-embedding without location constraints", on the 6 localization tasks generated from the 3 datasets (each task uses 2 datasets). The localization accuracies are shown in Table I, all under a 5-meter error distance. We also compare our methods with 2 baselines. "baseline-1" refers to the RADAR algorithm [1], which is a k-nearest-neighbor method using only labeled data for localization. As it cannot be used for handling the data with different feature dimensions (*e.g.* two floors' data are composed of signals from different sets of APs), we only used the shared APs as the features in RADAR. In particular, we use the labeled data from the source floor ($\mathbf{X}^A_{trn}$, $\mathbf{Y}^A_{trn}$) as training data, and train a K-NN classifier (where K is set as 1 to get the highest accuracies) for the unlabeled test data from the target floor $\mathbf{X}^B_{tst}$. Besides, "baseline-2" refers to the LeMan algorithm [5], which is a manifold learning method and also able to use the unlabeled data. However, LeMan requires the labeled data and the unlabeled data to follow the same data distribution and have same feature dimensions, so that those data can be smoothly distributed in one manifold for label propagation. Because in multi-floor environment, different floors' data can have different feature dimensions and distributions, LeMan may not work as well as our method. To use LeMan for comparison, we also only use the shared APs as the data features. Specifically, we take the source floor's labeled training data ($\mathbf{X}^A_{trn}$, $\mathbf{Y}^A_{trn}$) and the target floor's unlabeled training data $\mathbf{X}^B_{trn}$ as inputs for training a localization model. We then use the model to predict the labels for the target floor's test data $\mathbf{X}^B_{tst}$.

As we can see from Table I, our methods can greatly outperform both baselines, because we use co-embedding to carefully align two floor's data and meanwhile the baseline methods suffer a lot from signal variations over different floors. Moreover, we also find that, for the localization task between datasets 1 and 2, which share more access points, the performance is better, validating our motivation of co-embedding with shared APs.

We also compared the performances of our co-embedding methods using 2 datasets and using 3 datasets. The co-embeddings with 2 datasets follow the same settings with Table I, while the co-embeddings with 3 datasets works as

follows: for the task $i \rightarrow j$ (*e.g.* $i = 1$, $j = 2$), we also use the unlabeled data from another dataset k (*e.g.* $k = 3$) for co-embedding. As shown in Table II, co-embedding with 2 datasets consistently outperforms co-embedding with 3 datasets. This can be because the floors' data are quite different and adding one more extra unlabeled dataset for co-embedding may distract the attention of label propagation between the labeled dataset and the target unlabeled dataset. Moreover, according to our experience, the model using 3 datasets tends to be more sensitive to the parameters and easier to crash. We will study how to better incorporate extra unlabeled datasets in the future. Note that the results in Table II are based on co-embedding without location constraints, and we obtained similar results for co-embedding with location constraints.

Table II
COMPARISON FOR CO-EMBEDDING WITH 2 DATASETS VS. 3 DATASETS, USING CO-EMBEDDING WITHOUT LOCATION CONSTRAINTS.

S TO T	CO-EMBED W/ 2 SETS	CO-EMBED W/ 3 SETS
1 TO 2	$69 \pm 2\%$	$59 \pm 5\%$
2 TO 1	$66 \pm 3\%$	$58 \pm 3\%$
1 TO 3	$57 \pm 4\%$	$50 \pm 5\%$
3 TO 1	$55 \pm 5\%$	$49 \pm 7\%$
2 TO 3	$52 \pm 4\%$	$44 \pm 7\%$
3 TO 2	$54 \pm 3\%$	$44 \pm 4\%$

We further studied our model's performances (in form of cumulative probabilities) under varying error distances in Figure 5. Due to the space limit, we do not plot all 12 sets of results for the combinations of different dataset pairs and co-embedding strategies (with or without constraints). Instead, we only plot a set of results for a typical run of the "co-embedding without constraint" method using dataset 1 as source and dataset 2 as target. We observe similar patterns for the remaining 11 sets of results under varying error distances. Note that to get the results in Table I and Figure 5, we simply use the nearest neighbor method after using co-embedding to propagate labels to the target floor's unlabeled training data. We may expect that, using more sophisticated methods such as [3], [4], [8], we can further improve the localization accuracy. We also plot the corresponding co-embeddings in Figures 9, 10 and 11. The color in these figures denotes the location of each RSS vector, so it is more desirable that points with similar color are embedded closely. As we can see from these figures, in general, when the different data manifolds are better aligned by co-embedding, the localization results will be better (w.r.t. Table I).

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we put forward a multi-floor indoor localization problem and a solution that can effectively estimate a user's locations from his/her received signal strengths with much reduced effort. Our method is motivated by the daily life observations that, the floor-plans in a building are usually almost identical, and although the signals can be quite different in different floors, there can be some correspondences between them. To make use of such observations, we proposed to co-embed different floors' data in some common low-dimensional space, where we can align the unlabeled data with the labeled data and further propagate the labels. In this way, we only need to collect the labeled data on one floor, and collect unlabeled data on other floors.

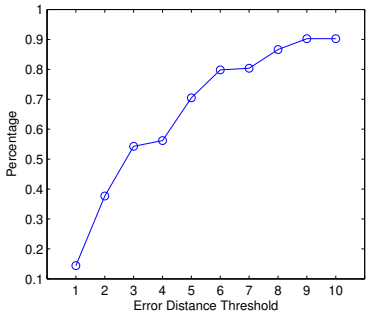

Figure 5. Accuracies under varying error distances, where the x-axis denotes error distance (unit: meter) and y-axis denotes localization accuracy.

We believe that, our general methodology deserves further investigation, whereby the graph embedding methods can be used to help reduce calibration effort for more general types of sensor network data. In the future, we will study to use more sophisticated learning methods coupled with our co-embedding method to further improve the localization performance. Also, although different floors in the same building have similar floor plans, some floors follow more complicated structures. We are interested in extending our method to cope with such more complex situations. We also expect to test and improve our methods on more datasets in larger-scale environments.

ACKNOWLEDGMENT

We would like to thank NEC Labs China for their support, and the anonymous reviewers for their insightful comments.

REFERENCES

[1] P. Bahl and V.N. Padmanabhan, *RADAR: An In-Building RF-Based User Location and Tracking System*, In Proc. of the Conference on Computer Communications, pp.775-784, 2000.

[2] L.M. Ni and Y. Liu and Y.C. Lau and A.P. Patil, *LANDMARC: Indoor Location Sensing Using Active RFID*, In Proc. of the 1st IEEE International Conference on Pervasive Computing and Communications, 2003.

[3] A. Haeberlen and E. Flannery and A.M. Ladd and A. Rudys and D.S. Wallach and L.E. Kavraki, *Practical Robust Localization over Large-Scale 802.11 Wireless Networks*, In the 8th Annual International Conference on Mobile Computing and Networking, 2002.

[4] M. Youssef and A. Agrawala, *The Horus WLAN location determination system*, In Proc. of the 3rd international conference on Mobile systems, applications, and services, pp.205-218, 2005.

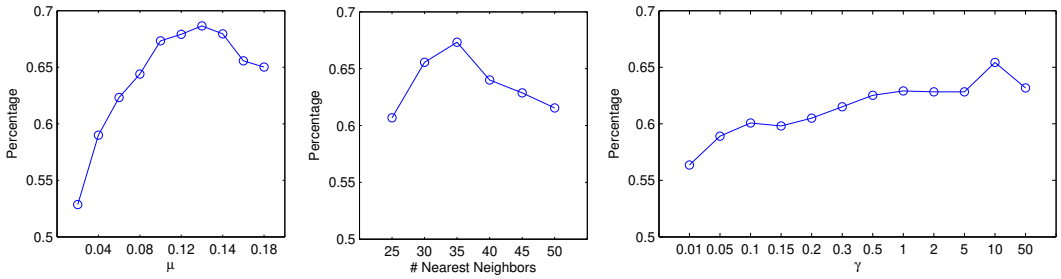

Figure 6. The impact of the parameters. In the left and middle figure we use the "embedding without location constraints" method (similar results are obtained for with constraints setting, although not reported here due to space limit). In the right figure we use the "embedding with location constraints" method, and have μ fixed at 0.1 and # NN fixed at 35. Note that the γ axis is not in uniform scale.

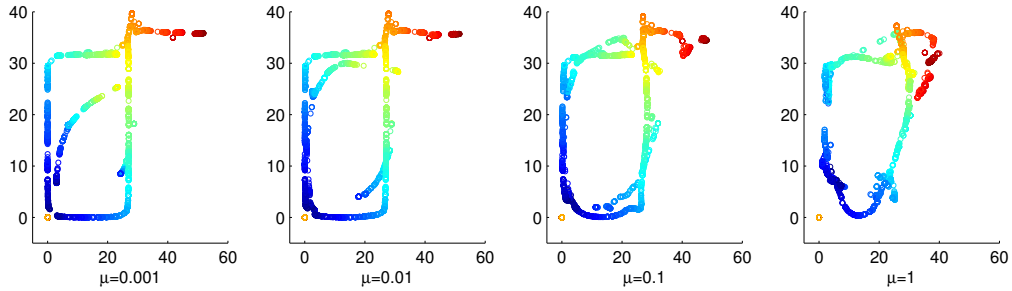

Figure 7. The impact of μ on co-embedding. We use the "co-embedding with location constraints" method (similar results are obtained for without constraints setting, although not reported here due to space limit). We can see that with a larger μ, the attraction between the source and target training data are getting stronger, as the inter-manifold graph is becoming stronger.

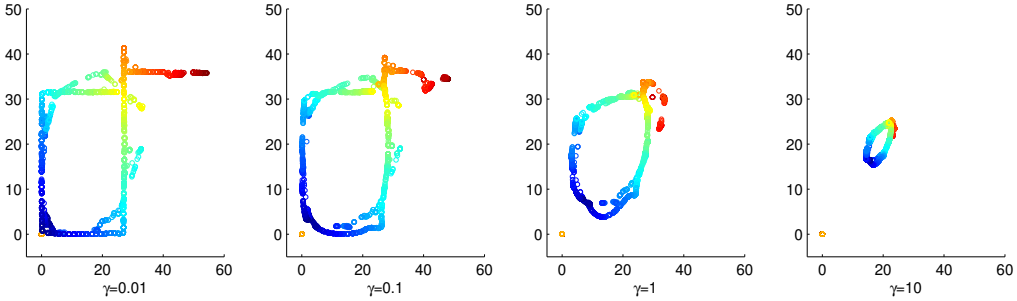

Figure 8. The impact of γ on co-embedding. We use the "co-embedding with location constraints" method. We can see that with a small γ (left plot), the relative weight of the location constraints is large, hence the source training data are embedded almost "right at" their ground truth locations. With increasing γ, the location constraints gradually fade out. Note that in all these 4 figures, there are some (overlapping, in warm color) data points located at location (0,0). They are actually the target training data from the right-wing hallway (warm color), and are embedded in a wrong location because there is no Maximal-RSS correspondence found in the right wing hallway between dataset 1 and 2, as shown in Figure 4's left plot.

[5] J.J. Pan and Q. Yang and H. Chang and Dit-Yan Yeung, *A Manifold Regularization Approach to Calibration Reduction for Sensor-Network Based Tracking*, In Proc. of the 21st National Conference on Artificial Intelligence, 2006.

[6] N. Priyantha and A. Chakraborty and H. Balakrishnan, *The Cricket Location-Support system*, In Proc. of the 6th Annual ACM International Conference on Mobile Computing and Networking, 2000.

[7] A. Savvides and C. Han and M.B. Strivastava, *Dynamic fine-grained localization in Ad-Hoc networks of sensors*, In Proc. of the 7th International Conference on Mobile Computing and Networking, 2001.

[8] B. Ferris and D. Haehnel and D. Fox, *Gaussian Processes for Signal Strength-Based Location Estimation*, In Proc. of Robotics: Science and Systems, 2006.

[9] M. Youssef and A. Agrawala and U. Shankar, *WLAN Location Determination via Clustering and Probability Distributions*, In Proc. of the 1st IEEE International Conference on Pervasive Computing and Communications, pp.143-150, 2003.

[10] T. Roos and P. Myllymaki and H. Tirri and P. Misikangas and J. Sievanen, *A Probabilistic Approach to WLAN User Location Estimation*, International Journal of Wireless Information Networks, vol.9, pp.155-164, num.3, 2002.

[11] V. Otsason and A. Varshavsky and A. LaMarca and Eyal de Lara, *Accurate GSM Indoor Localization*, In Proc. of the 7th International Conference on Ubiquitous Computing, pp.141-158, 2005.

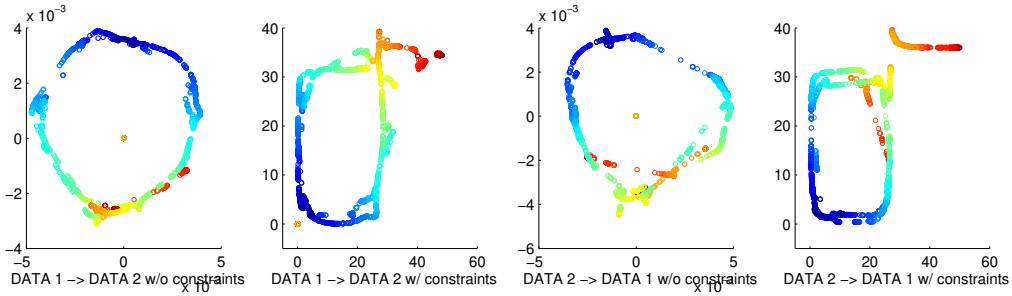

Figure 9. The co-embedding results for localization task between datasets 1 and 2.

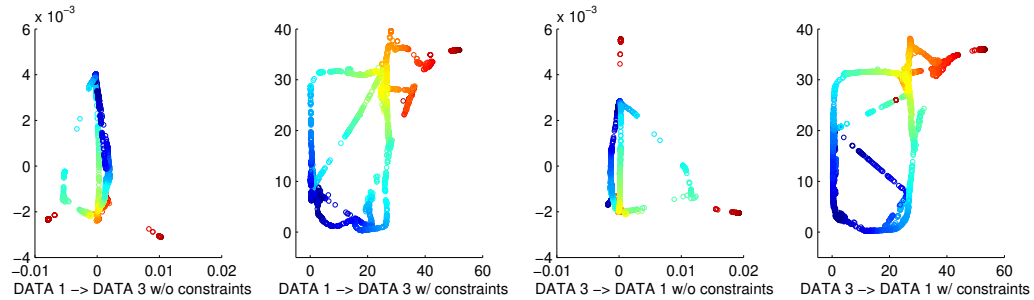

Figure 10. The co-embedding results for localization task between datasets 1 and 3.

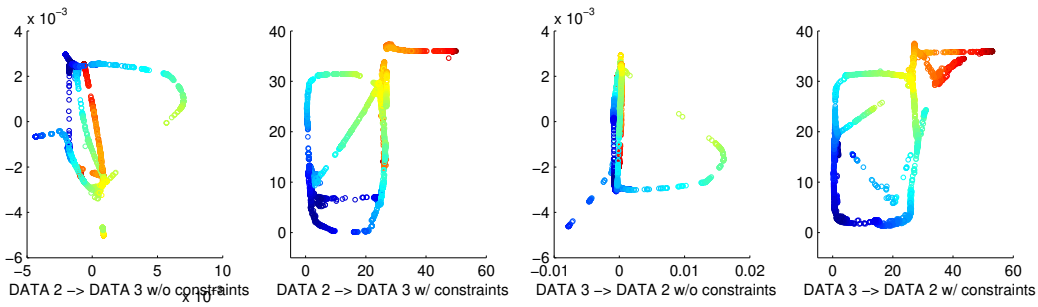

Figure 11. The co-embedding results for localization task between datasets 2 and 3.

[12] J. Letchner and D. Fox and A. LaMarca, *Large-Scale Localization from Wireless Signal Strength*, In Proc. the 20th National Conference on Artificial Intelligence, pp.15-20, 2005.

[13] A. Varshavsky and A. LaMarca and J. Hightower and Eyal de Lara, *The SkyLoc Floor Localization System*, In Proc. of the 5th IEEE International Conference on Pervasive Computing and Communications, pp.125-134, 2007.

[14] O. Woodman and R. Harle, *Pedestrian localisation for indoor environments*, In Proc. of the 10th international conference on Ubiquitous computing, pp.114-123, 2008.

[15] D. Madigan, E. Einahrawy, R.P. Martin, W.-H. Ju, P. Krishnan, A.S. Krishnakumar, *Bayesian indoor positioning systems*. In Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pp.1217-1227, 2005.

[16] H. Lim, L.-C. Kung, J.C. Hou, H. Luo, *Zero-Configuration, Robust Indoor Localization: Theory and Experimentation*. In Proc. of the 25th IEEE International Conference on Computer Communications, pp.1-12, 2006.

[17] V.W. Zheng, S.J. Pan, Q. Yang, J.J. Pan, *Transferring Multi-device Localization Models using Latent Multi-task Learning*. In Proc. of the 23rd AAAI Conference on Artificial Intelligence, pp.1427-1432, 2008.

[18] V.W. Zheng, E.W. Xiang, Q. Yang, D. Shen, *Transferring Localization Models Over Time*. In Proc. of the 23rd AAAI Conference on Artificial Intelligence, pp.1421-1426, 2008.

[19] J. Hamm and D. Lee and L.K. Saul, *Semisupervised alignment of manifolds*. In Proc. of the 10th International Workshop on Artificial Intelligence and Statistics, 120-127, 2005.

[20] M. Belkin and P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*. Neural Computation, 15, 2003.