Online Co-Localization in Indoor Wireless Networks by Dimension Reduction

Jeffrey Junfeng Pan, Qiang Yang and Sinno Jialin Pan *

Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong

{panjf, qyang, sinnopan}@cse.ust.hk

Abstract

This paper addresses the problem of recovering the locations of both mobile devices and access points from radio signals that come in a stream manner, a problem which we call *online co-localization*, by exploiting both labeled and unlabeled data from mobile devices and access points. Many tracking systems function in two phases: an *offline training phase* and an *online localization phase*. In the training phase, models are built from a batch of data that are collected offline. Many of them can not cope with a dynamic environment in which calibration data may come sequentially. In such case, these systems may gradually become inaccurate without a manually costly re-calibration. To solve this problem, we proposed an *online co-localization* method that can deal with labeled and unlabeled data stream based on semi-supervised manifold-learning techniques. Experiments conducted in wireless local area networks show that we can achieve high accuracy with less calibration effort as compared to several previous systems. Furthermore, our method can deal with online stream data relatively faster than its two-phase counterpart.

Introduction

With the recent advance in pervasive computing and mobile technology, tracking wireless devices using received-signal-strength (RSS) has attracted intense interest in many research communities. It is a useful task in robotics and activity recognition. It is also a difficult task since radio signals usually attenuate in a highly nonlinear and uncertain way in a complex environment where client devices may be moving. Existing approaches to RSS localization fall into two main categories (Ferris, Hahnel, & Fox 2006): (1) radio propagation models (Maligan *et al.* 2005; Savvides, Han, & Strivastava 2001), which rely on the knowledge of access point locations; (2) statistical machine learning models (Nguyen, Jordan, & Sinopoli 2005; Letchner, Fox, & LaMarca 2005; Bahl & Padmanabhan 2000), which require a large amount of costly calibration.

In general, machine-learning-based systems using RSS values function in two phases (Pan *et al.* 2006a): an *offline training phase* and an *online, localization phase*. In the offline phase, a *probabilistic model* is trained by considering the signal strength values received from the access points at selected locations in the area of interest. These values comprise the training data gathered from a physical region, which are used to calibrate a probabilistic location-estimation system. In the online localization phase, the real-time signal strength samples received from the access points are used to estimate the current location based on the learned model.

However, in many applications, access points are not deployed in a static environment in which calibration and uncalibrated data come in a stream manner. Access points may be removed, relocated and added for better coverage and link quality. In either case, a localization system may gradually become inaccurate without a manually costly re-calibration and re-run the whole training process. It is also wasteful to discard previous computation results. A better idea is to construct an online model where calibration data come in stream. Previous works have been done in online graph learning (Herbster, Pontil, & Wainer 2005; Law & Jain 2006; Kouropteva, Okun, & Pietikainen 2005; Jenkins & Mataric 2004). More specifically in wireless and sensor networks, (Funiak *et al.* 2006) describes a similar problem when beacon nodes are cameras. (Taylor *et al.* 2006) presents a framework for simultaneously localization and mapping with ultrasonic sensors based on Baysian Filter (Fox *et al.* 2003). (Ferris, Fox, & Lawrence 2007) shows WiFi SLAM using Gaussian Process model.

In this paper, we address the problem of recovering the locations of both mobile devices and access points from radio signals that come in a stream manner, a problem which we call *online co-localization*, by exploiting both labeled and unlabeled data from mobile devices and access points. The proposed method is based on online and incremental manifold-learning techniques (Law & Jain 2006; Kouropteva, Okun, & Pietikainen 2005; Jenkins & Mataric 2004), semi-supervised techniques that can cope with labeled and unlabeled data that come sequentially.

We test our *online co-localization* in a Wireless Local Area Network (WLAN). Experiments show that we can achieve high accuracy with less calibration effort as compared to several previous systems. Furthermore, our method can incrementally deal with data stream *online* relatively faster than its two-phase counterpart.

Related Works

Propagation-model-based approaches are widely used for location estimation due to their simplicity and efficiency (Letchner, Fox, & LaMarca 2005). These methods usually assume that access points are *labeled*, e.g., their locations are known. They estimate the distance of the mobile devices relative to some fixed access points based on signal strengths through models that predicts the signal propagation patterns (Savvides, Han, & Strivastava 2001). Researchers have also used Bayesian models to encode the sig-

*Research supported by NEC Lab China (NECLC05/06.EG01).

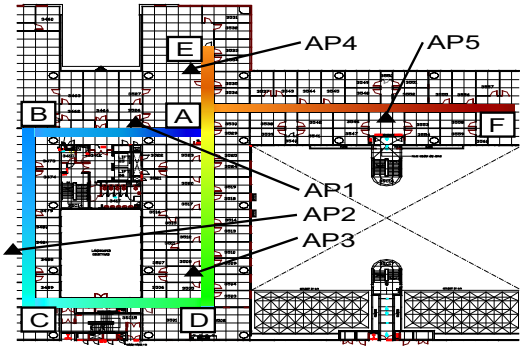


Figure 1: WLAN Test-bed

nal propagation pattern (Letchner, Fox, & LaMarca 2005; Maligan *et al.* 2005) and infer the locations using Monte Carlo methods (Thrun *et al.* 2001). A drawback of propagation-model-based methods is that these models may become inaccurate in a complex domain.

An alternative is to apply machine-learning-based algorithms. With these algorithms the *labels* of access points need not be known. Instead, they usually rely on models that are trained with RSS data collected on a mobile device and are *labeled* with physical locations (Letchner, Fox, & LaMarca 2005; Nguyen, Jordan, & Sinopoli 2005; Ni *et al.* 2003; Bahl & Padmanabhan 2000). The training data are usually collected offline. These signal values may be noisy and nonlinear due to environmental dynamics. Therefore, sufficient data shall be collected to power algorithms for approximating the signal to location mapping functions using K-Nearest-Neighbors (Bahl & Padmanabhan 2000), kernels (Pan *et al.* 2005), Bayesian filters (Letchner, Fox, & LaMarca 2005) and Gaussian processes (Ferris, Hahnel, & Fox 2006). A drawback of these models is that they may require much calibration effort.

A viable approach is to use both *labeled* and *unlabeled* data. For example, Bayesian frameworks can be applied to use both *labeled* and *unlabeled* access points (Letchner, Fox, & LaMarca 2005). Our work differs from the above in that we treat mobile devices and access points in a completely symmetric manner: we use both the *labeled* and *unlabeled* data from mobile devices and access points to recover the locations of both of them rather than locating the mobile devices only. Our work is related to (Taylor *et al.* 2006; Ferris, Fox, & Lawrence 2007) in the sense that we combine the training an localization phases together into an online and incremental model that can dynamically adopt new calibration data sequentially.

Methodology

Problem Statement

Co-localization addresses the problem of recovering the locations of both mobile devices and access points, by exploiting both *labeled* and *unlabeled* data from both mobile devices and access points (Pan & Yang 2007). *Co-localization* can be done in a traditional Two-Phase manner: an *Offline Training Phase* and an *Online Localization Phase*. However, in a dynamic environment where calibration data come sequentially, it will be inefficient to build the model repeatedly. A better idea is to adjust the current model *online*.

Table 1: Signal Strength (unit:dBm)

	AP_1	AP_2	AP_3	AP_4	AP_5
t_A	-40		-60	-40	-70
t_B	-50	-60		-80	
t_C		-40	-70		
t_D	-80		**-40**	-70	
$t_{A'}$	-40		-70	-40	-60
t_E	-40		-70	-40	-80
t_F	-80			-80	-50

(All values are rounded for illustration)

Consider a 2-dimensional *online co-localization* problem: Assume that a user holds a mobile device and navigates in an indoor wireless environment $\mathcal{C} \subseteq \mathbb{R}^2$ of n access points, which can periodically send out beacon signals. At some time t_i, the RSS values from all the n access points are measured by the mobile device to form a row vector $\mathbf{s}_i = [s_{i1}\ s_{i2}\ \ldots\ s_{in}] \in \mathbb{R}^n$. As time elapses, these row vectors come in a stream manner. After m time ticks, we get a sequence of m signal strength vectors form an $m \times n$ matrix $S = [\mathbf{s}'_1\ \mathbf{s}'_2\ \ldots\ \mathbf{s}'_m]'$, where "prime" is used to denote matrix transposition. Here, the locations of some access points and the mobile devices at some time t are known or *labeled*, while the rest are *unlabeled*.

Our objectives are stated as follows: In an *online and incremental* manner, we wish to estimate the $m \times 2$ location matrix $P = [\mathbf{p}'_1, \mathbf{p}'_2, \ldots, \mathbf{p}'_m]'$ where $\mathbf{p}_i = [p_{i1}\ p_{i2}] \in \mathcal{C}$ is the location of the mobile device at time t_i and the $n \times 2$ location matrix $Q = [\mathbf{q}'_1, \mathbf{q}'_2, \ldots, \mathbf{q}'_n]'$ where $\mathbf{q}_j = [q_{j1}\ q_{j2}] \in \mathcal{C}$ is the location of the j access points.

Our objectives are to determine and update the locations of all of the remaining access points and the trajectory of the mobile device in real-time as partially calibrated data come sequentially. Note that m is not a constant value. As time elapses, m may increase from 1, 2, ..., to any number. We want to dynamically adjust the model when observing new data without (or with) an *offline training phase*. We call this problem *online co-localization*.

Example 1 As an example, Figure 1 shows an indoor 802.11 wireless LAN environment of size about $60m \times 50m$. It is equipped with $n = 5$ access points. A user with an IBM T42 notebook that is equipped with an Intel Pro/2200BG internal wireless card walks from A through B, C, D, A, E to F at time $t_A, t_B, t_C, t_D, t_{A'}, t_E, t_F$. Correspondingly, a total number of $m = 1, 2, \ldots, 7$ signal strength vectors are incrementally extracted. The final 7×5 matrix S is shown in Table 1. By walking from A to F in the hallways, we collected 500 signal strength vectors from 5 access points. Note that the blank cells denote the missing values, which we can fill in a small default value, e.g., $-100dBm$.

Our task is to dynamically update the trajectory matrix P of the mobile device at each time when new data come and to determine the location matrix Q of the access points $AP_1, AP_2, \ldots, AP_5$ in an *online* manner.

Domain Characteristics

There are four main characteristics about received-signal-strengths by observing the data in Table 1:

1. Considering two *rows* of the data, the mobile device at two different time may be spatially close if their pairwise signal strengths are similar from most access points, e.g., the time t_A and $t_{A'}$.
2. Considering two *columns* of the data, two access points may be spatially close if their pairwise signal strengths are similar most of the time, e.g., AP_1 and AP_4.
3. Considering a *single cell* s_{ij} of the data, the mobile device and the j access point may be spatially close to each other at time t_i if the signal is strong, e.g., the mobile device is close to AP_3 at time t_D.
4. Considering two *neighbored rows* of the data, the mobile device at two consecutive time may be spatially close if their time interval is small by assuming that a user may not move too fast or too irregularly. For example, the locations of the mobile device at time $t_{A'}$ and t_E are close since $|t_{A'} - t_E| < \Delta T$.

It is not surprising that the above observations are related to the assumption of manifold-learning techniques: When the locations of some access points and the mobile device at some time are known, we can ground the unknown coordinates by exploiting the geometry of the signal distribution. Manifold-based methods generally assume that if two points are close in the intrinsic geometry of the marginal distribution, their conditional distributions are similar (Belkin, Niyogi, & Sindhwani 2005; Ham, Lee, & Saul 2005), which approximately holds in our above observations and several previous works (Pan *et al.* 2006b; Pan & Yang 2007)

Solution I: *Two-Phase Co-Localization*

Offline Training Phase When the manifold assumption holds, the optimal solution is give by $\mathbf{f}^* = \arg\min_{\mathbf{f}} \Sigma_{i=1}^{l} |f_i - y_i|^2 + \gamma \mathbf{f}^T L \mathbf{f}$ (Ham, Lee, & Saul 2005) where the first term measures the fitting error and the second term poses the smoothness along the manifold and L is the graph Laplacian (Chung 1997).

First of all, we can express the similarity within all the collected signal vectors $\mathbf{s}_i$ $(i = 1, 2, \ldots, m)$ by constructing the neighborhood graph and its graph Laplacian matrix L_P (Chung 1997). The objective is to optimize:

$$P^* = \arg\min_{P \in \mathbb{R}^{m \times 2}} (P - Y_P)' J_P (P - Y_P) + \gamma_P P' L_P P \quad (1)$$

where P is the coordinate matrix of the mobile device to be determined; $J_P = diag(\delta_1, \delta_2, \ldots, \delta_m)$ is an indication matrix where $\delta_i = 1$ if the coordinate of the mobile device at time t_i is given and otherwise $\delta_i = 0$; $Y_P = [\mathbf{y}'_1, \mathbf{y}'_2, \ldots, \mathbf{y}'_m]'$ is an $m \times 2$ matrix supplying the calibration data where $\mathbf{y}_i$ is the given coordinate of the mobile device at time t_i if $\delta_i = 1$ and otherwise the value of $\mathbf{y}_i$ can be any, e.g., $\mathbf{y}_i = [0\ 0]$; γ_P controls the smoothness of the coordinates along the manifold; $L_P = D_P - W_P$ is the graph Laplacian; $W_P = [w_{ij}]_{m \times m}$ is the weight matrix and $w_{ij} = exp(-\|\mathbf{s}_i - \mathbf{s}_j\|^2 / 2\sigma_P^2)$ if $\mathbf{s}_i$ and $\mathbf{s}_j$ are neighbors along the manifold and otherwise $w_{ij} = 0$; $D_P = diag(d_1, d_2, \ldots, d_m)$ and $d_i = \sum_{j=1}^{m} w_{ij}$.

Setting the derivative of Equation (1) to zero, the optimal solution is given by (Ham, Lee, & Saul 2005)

$$P^* = (J_P + \gamma_P L_P)^{-1} J_P Y_P \quad (2)$$

We also construct the neighborhood graph for access points in a similar way, the optimal solution is given by

$$Q^* = \arg\min_{Q \in \mathbb{R}^{n \times 2}} (Q - Y_Q)' J_Q (Q - Y_Q) + \gamma_Q Q' L_Q Q \quad (3)$$

where $L_Q = D_Q - W_Q$ is the graph Laplacian, W_Q is the weight matrix and D_Q is constructed from W_Q.

Furthermore, we encode the similarity between access points and mobile devices by transforming the signal matrix $S = [s_{ij}]_{m \times n}$ to a non-negative weight matrix $A = [a_{ij}]_{m \times n}$ by a Gaussian kernel $a_{ij} = exp(-|s_{ij} - s^{max}|^2 / 2\sigma_A^2)$ where s^{max} is the maximal signal strength detected, e.g., the signal strength around an access point or $-30dBm$. From the weight matrix A, we construct the graph Laplacian $L_A = D_A - W_A$ where $W_A = \begin{bmatrix} 0 & A \\ A' & 0 \end{bmatrix}$ and $D_A = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$. We also reform L_P and L_Q to larger matrices with $L_B = \begin{bmatrix} L_P & 0 \\ 0 & 0 \end{bmatrix}$ and $L_C = \begin{bmatrix} 0 & 0 \\ 0 & L_Q \end{bmatrix}$. Now L_A, L_B and L_C are graph Laplacians of the same size. L_A describes the similarity between mobile devices and access points. L_B and L_C express the similarity within them respectively. Putting these together, our objective is to optimize:

$$R^* = \arg\min_{R \in \mathbb{R}^{(m+n) \times 2}} (R - Y)' J (R - Y) + \gamma R' L R \quad (4)$$

where $R = [\mathbf{r}'_1, \mathbf{r}_2, \ldots, \mathbf{r}'_{m+n}]' = [P'\ Q']'$ is the coordinate matrix of the mobile device and the access points; $Y = [Y'_P\ Y'_Q]'$ supports the partial labels; $J = \begin{bmatrix} J_P & 0 \\ 0 & J_Q \end{bmatrix}$ is the indication matrix; $L = \gamma_A L_A + \gamma_B L_B + \gamma_C L_C = D - W$ is the graph Laplacian. The optimal solution is given by:

$$R^* = (J + \gamma L)^{-1} J Y \quad (5)$$

We can export the estimated coordinates of the mobile device trajectory P^* and access point locations Q^* from $R^* = [P^{*\prime}\ Q^{*\prime}]'$.

Online Localization Phase In the localization phase, the location of a new signal strength vector $\mathbf{s}_i$ is predicted as follows:

1. Find the k neighbors closest to $\mathbf{s}_i$ in the training data $S = [\mathbf{s}'_1\ \mathbf{s}'_2\ \ldots\ \mathbf{s}'_m]'$. Let $\mathcal{C}_i$ be the index set of the k nearest neighbors. Besides, we link $\mathbf{s}_i$ to those access points from which we can detect the radio signal. We also link $\mathbf{s}_i$ to $\mathbf{s}_{i-1}$ in order to pose the temporal constraint by assuming that a user may not move too fast ($t_i - t_{i-1} < \Delta T$). Denote the index set for these additional links as $\mathcal{B}_i$.

2. Approximately, we can predict the location using a property of harmonic functions (Zhu, Ghahramani, & Lafferty 2003; Ham, Lee, & Saul 2005), which are smooth functions on the graph such that $\mathbf{r}_i$ is determined by the weighted average of its neighbors. This property holds if there is no uncertainty in the labeled locations of matrix P during training ($\gamma \to 0$).

$$\tilde{\mathbf{r}}_i \approx \frac{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij} \mathbf{r}_j}{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij}} \quad (6)$$

Note that the above $\tilde{\mathbf{r}}_i$ is an approximation because adding $\mathbf{s}_i$ to the existing neighborhood graph from the train data may slightly change the graph structure: We have linked the i^{th} node to the node set $\mathcal{C}_i$; However, we have not yet eliminated any existing edge in the graph to maintain the k-neighbor relationship among all nodes.

Solution II: *Online Co-Localization*

We will extend the above *Two-Phase Co-Localization* model to an *online* version. We wish that it can dynamically adjust itself when new data come sequentially in real-time. The key point is how to add the new data into the learned graph by updating the k-neighbor relationship and the corresponding weight matrix W. This can be done repeatedly in two online steps: Predict and Update.

Predict Given a new signal vector $\mathbf{s}_i$ at time t_i, we find its k nearest neighbors and use Equation (6) in the above *online localization phase* for predicting the location $\tilde{\mathbf{r}}_i$.

Update The addition and deletion of nodes can modify the neighborhood graph and the corresponding graph Laplacian. We use the method described in (Law & Jain 2006) for updating the neighborhood graph structure locally.

•**Node Addition** Let $\mathcal{A}_i^+$ and $\mathcal{D}_i^+$ be the set of edges to be added and deleted after inserting v_i to the neighborhood graph, respectively. Let τ_i be the index of the k^{th} nearest neighbor of v_i. So, v_j is in the k nearest neighborhood of v_i if $\Delta_{ij} \leq \Delta_{i,\tau_i}$. When $\Delta_{i,\tau_i} > \Delta_{i,\tau_i}$, v_{n+1} replaces v_{τ_i} in the knn neighborhood of v_i. It is easy to see that:

$$\begin{aligned} \mathcal{A}_i^+ = &\quad \{e(j, n+1) \,:\, j \in \mathcal{C}_i \; or \; \Delta_{j,\tau_j} > \Delta_{ji}\} \\ \mathcal{D}_i^+ = &\quad \{e(j, \tau_j) \,:\, \Delta_{j,\tau_j} > \Delta_{j,i} \; and \; \Delta_{\tau_j,j} > \Delta_{\tau_j,l_j}\} \\ &\text{where } l_i \text{ is the index of the } k^th \text{ nearest neighbor} \\ &\text{of } v_{\tau_j} \text{ after inserting } v_i \text{ in the graph.} \end{aligned}$$

•**Node Deletion** Similarly, let $\mathcal{A}_i^-$ and $\mathcal{D}_i^-$ denote the set of edges to be added and deleted after removing v_i from the neighborhood graph, respectively. The graph update can be done as follows:

$$\begin{aligned} \mathcal{A}_i^- = &\quad \{e(i, h_i)\} \text{ where } h_i \text{ is the } (k+1)^{th} \text{ nearest} \\ &\text{neighbor before removing } v_i \text{ in the graph.} \\ \mathcal{D}_i^- = &\quad \{e(i, j) \,:\, j \in \mathcal{C}_i\} \end{aligned}$$

After updating the neighborhood graph, it is straightforward to modify the corresponding weight matrix W. For an added edge $e(i, j)$, we set both the values of w_{ij} and w_{ji} because the neighborhood graph is symmetric. If it is a deleted edge, we clear the values of w_{ij} and w_{ji}. The graph Laplacian $L = D - W$ can be updated in a similar way.

Finally, we have to re-estimate the location matrix $R = [P'Q']'$ of the mobile devices and the access points so that it can reflect the change of the neighborhood graph and the new graph Laplacian L. Instead of using Equation (5) for solving R, we update R by iteration. In each iteration cycle, we apply:

$$\mathbf{r}_i^{new} \leftarrow \frac{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij} \mathbf{r}_j^{old}}{\sum_{j \in \mathcal{C}_i \cup \mathcal{B}_i} w_{ij}} \; (i = 1, 2, \ldots, m+n) \quad (7)$$

We use the predicted $\tilde{\mathbf{r}}_i$ $(i = 1, \ldots, m+n)$ as the initial values for iteration. Furthermore, the weight matrix W does vary too much after addition or deletion. We can obtain very good estimation after a few iterations.

Example 2 A user with a mobile device walks in the office area shown in Figure 1. The mobile device periodically collects signal vectors. The user can mark down his location when he walks by some landmark points such as corners and dead-ends of the hallways ($A, B, \ldots, F$). Thus, the data that come in stream are partially *labeled*. By applying the *online co-localization* method, we continuously update the recovered locations of the mobile devices and the access points. Figure 2 shows the *online co-localizaiton* results at six key frames when the user walks by $A, B, \ldots, F$. As can be seen, the locations of the user trajectory and the access points are dynamically calibrated when obtaining new data. For example, AP_3 gradually converges to its true location.

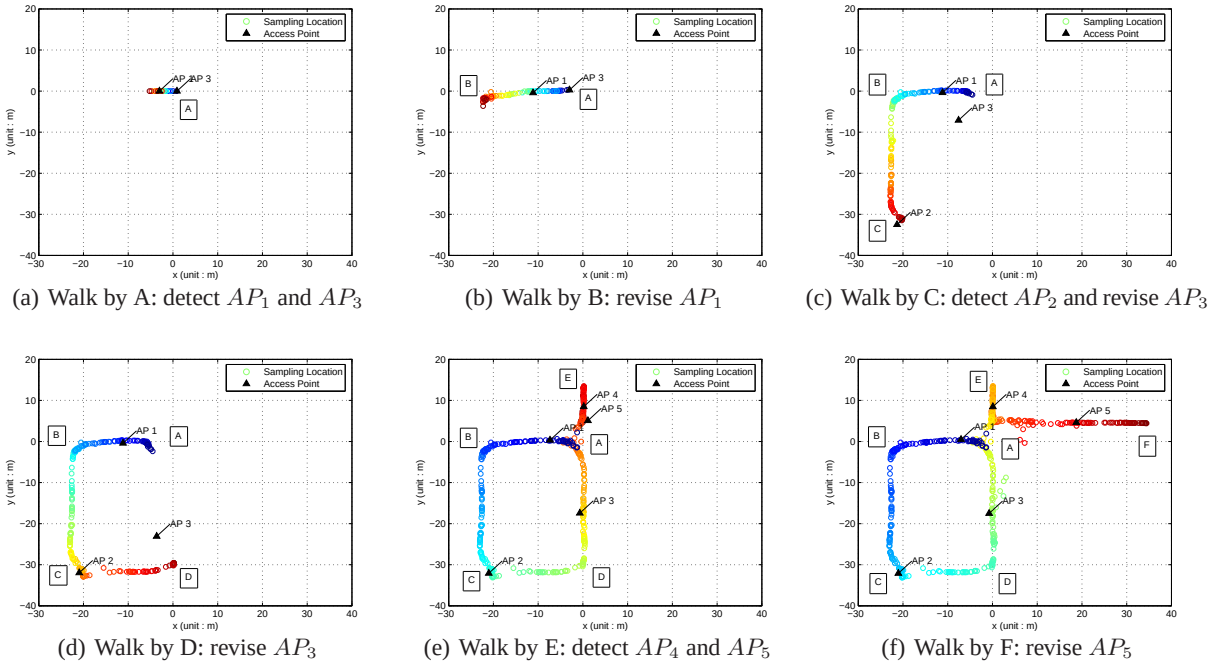
Experiments

Accuracy and Calibration Effort

We evaluated the performance of the *co-localization* algorithm in an 802.11 WLAN as shown in Figure 1. A person carried an IBM T42 laptop which is equipped with an Intel Pro/2200GB internal wireless card and walked in the environment. A total of 2000 examples are collected sequentially with sample rate $2Hz$. The ground-truth location labels are obtained by referring to landmark points such as doors, corners and dead-ends.

For comparison, we also run the following baseline algorithms (1) LANDMARC, a nearest-neighbor weighting based method (Ni *et al.* 2003); (2) Support Vector Regression (SVR), a simplified variant of a kernel-based method (Nguyen, Jordan, & Sinopoli 2005); (3) RADAR, a K-Nearest-Neighbor method (Bahl & Padmanabhan 2000).

In each experiment, we randomly pick up a sequence of 500 examples for training and the rest for testing. The training data are further split into *labeled* and *unlabeled* parts. The results shown in Figure 3 are averaged over 10 repetitions for reducing statistical variability. LANDMARC, RADAR and SVR use the *labeled* part of training data only. In contrary, the *online co-localization* method uses both *labeled* and *unlabeled* data. Figure 3(a) plots the cumulative probability with respect to error distance. As can be seen, the proposed *online co-localization* benefits from the



(a) Walk by A: detect AP_1 and AP_3 (b) Walk by B: revise AP_1 (c) Walk by C: detect AP_2 and revise AP_3

(d) Walk by D: revise AP_3 (e) Walk by E: detect AP_4 and AP_5 (f) Walk by F: revise AP_5

Figure 2: Illustration of the Online Co-Localization when a user walks from A through $B, \ldots, D$ to F

additional *unlabeled* data and increases the accuracy. Figure 3(b) and 3(c) show the localization error of the mobile device and access points by varying the number of labeled examples in a training subset which size is fixed to 500. Again, the proposed method performs relatively better than the baselines. By employing the *unlabeled* data, we save the calibration effort.

Speed Test

We compare the speed of Two-Phase *co-localization* method and the *online* version. Suppose that data come one by one sequentially. Once we get a new signal vector, the Two-Phase method adds it as a training example and rebuilds the whole model. In comparison, *Online co-localization* updates the estimation incrementally. Figure 4 shows the average running time for adding a new vector. The test is done in Matlab on a computer that has a 2.0GHz CPU. Experimental results show that we can greatly reduce the time for the model adaption in an online manner. For example, when the training dataset size is incrementally enlarged to about 500, the Two-Phase method needs $1.2s$ to re-estimate everything while the online method spends no more than $0.1s$. The *online* method is more than ten times faster.

Conclusion and Future Works

We have developed a manifold-based online and incremental approach to solve the problem of recovering the locations of both mobile devices and access points from radio signals that come in a stream manner. In our *online co-localization* framework, we exploit both labeled and unlabeled data from mobile devices and access points *online* when data come in stream. Experiments conducted in an 802.11 WLAN show

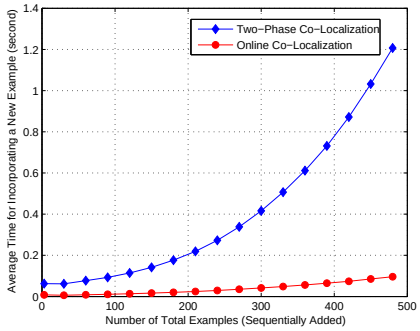


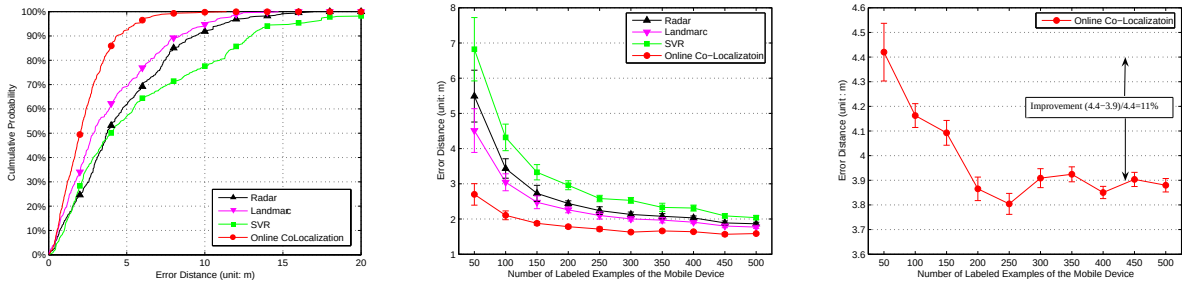
Figure 4: Average Running Time Comparison

that we can achieve high accuracy with less calibration effort as compared to several previous systems. Furthermore, our method can deal with data stream *online* relatively faster while compared to its Two-Phase counterpart. In the future, we would continue to study the environment in which access points may be removed, relocated and added for better coverage and link quality and how the algorithm can adapt the change dynamically.

References

Bahl, P., and Padmanabhan, V. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the Conference on Computer Communications*, volume 2, 775–784.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2005. On manifold regularization. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 17–24. Society for Artificial Intelligence and Statistics.



(a) Comparison of Accuracy (b) WLAN Mobile Devices (c) WLAN Access Points

Figure 3: Experimental Results over 10 Repetitions (Mean and Std.)

Chung, F. 1997. *Spectral Graph Theory*. American Mathematical Society.

Ferris, B.; Fox, D.; and Lawrence, N. 2007. Wifi-slam using gaussian process latent variable models. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2480–2485.

Ferris, B.; Hahnel, D.; and Fox, D. 2006. Gaussian processes for signal strength-based location estimation. In *Proceedings of Robotics: Science and Systems*.

Fox, D.; Hightower, J.; Liao, L.; and Schulz, D. 2003. Bayesian filtering for location estimation. *IEEE Pervasive Computing* 2(3):24–33.

Funiak, S.; Guestrin, C.; Paskin, M.; and Sukthankar, R. 2006. Distributed localization of networked cameras. In *Proceedings of the fifth international conference on Information processing in sensor networks*, 34–42. New York, NY, USA: ACM Press.

Ham, J.; Lee, D.; and Saul, L. 2005. Semisupervised alignment of manifolds. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 120–127. Society for Artificial Intelligence and Statistics.

Herbster, M.; Pontil, M.; and Wainer, L. 2005. Online learning over graphs. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, 305–312. New York, NY, USA: ACM Press.

Jenkins, O. C., and Mataric, M. J. 2004. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the twenty-first international conference on Machine learning*, 56. New York, NY, USA: ACM Press.

Kouropteva, O.; Okun, O.; and Pietikainen, M. 2005. Incremental locally linear embedding algorithm. *Pattern Recognition* 38(10):1764–1767.

Law, M. H. C., and Jain, A. K. 2006. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 28(3):377–391.

Letchner, J.; Fox, D.; and LaMarca, A. 2005. Large-scale localization from wireless signal strength. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 15–20.

Maligan, D.; Elnahrawy, E.; Martin, R.; Ju, W.; Krishnan, P.; and Krishnakumar, A. 2005. Bayesian indoor positioning systems. In *Proceedings of the Conference on Computer Communications*, volume 2, 1217–1227.

Nguyen, X.; Jordan, M. I.; and Sinopoli, B. 2005. A kernel-based learning approach to ad hoc sensor network localization. *ACM Transactions on Sensor Networks* 1(1):134–152.

Ni, L.; Liu, Y.; Lau, Y.; and Patil, A. 2003. LANDMARC: Indoor location sensing using active RFID. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, 407–416.

Pan, J. J., and Yang, Q. 2007. Co-localization from labeled and unlabeled data using graph laplacian. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 2166–2171.

Pan, J. J.; Kwok, J. T.; Yang, Q.; and Chen, Y. 2005. Accurate and low-cost location estimation using kernels. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, 1366–1371.

Pan, J. J.; Kwok, J. T.; Yang, Q.; and Chen, Y. 2006a. Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. In *IEEE Transactions on Knowledge and Data Engineering*, volume 18, 1181– 1193.

Pan, J. J.; Yang, Q.; Chang, H.; and Yeung, D. Y. 2006b. A manifold regularization approach to calibration reduction for sensor-network based tracking. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 988–993.

Savvides, A.; Han, C.; and Strivastava, M. B. 2001. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, 166–179.

Taylor, C.; Rahimi, A.; Bachrach, J.; Shrobe, H.; and Grue, A. 2006. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. In *Proceedings of the fifth international conference on Information processing in sensor networks*, 27–33. New York, NY, USA: ACM Press.

Thrun, S.; Fox, D.; Burgard, W.; and Dallaert, F. 2001. Robust monte carlo localization for mobile robots. *Artificial Intelligence* 128(1-2):99–141.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on Machine Learning*, 912–919.