Mapless Indoor Localization by Trajectory Learning from a Crowd

Jaehyun Yoo, H. Jin Kim, and Karl H. Johansson

***Abstract*—This paper suggests a mapless indoor localization using wifi received signal strength (RSS) of a smartphone, collected by multiple people. A new trajectory learning algorithm by combining a dynamic time warping and a machine learning technique is proposed in order to generate an alternative map. Moreover, we combine particle filter and Gaussian process (GP) for the position estimation, because it can use the alternative map as the probabilistic function (the prior), and can use probabilistic relationship (the likelihood) between wifi RSSs and location. Field experimental results confirm the usefulness of our algorithm when the map is not available and robustness against outliers, in that the accuracy of the proposed localization is similar to that using the true map information.**

I. INTRODUCTION

Indoor localization becomes of increasing interests due to the need for location information where GPS is not available. Fortunately, prevalence of wireless access points built in many buildings and public spaces helps developing a wifi-based indoor localization without additional installation of positioning devices such as lidar, camera, and ultra wide band.

Wifi-based localization consists of training and test phases, which is called fingerprinting. In the training phase, a service provider collects the wifi received signal strengths (RSSs) at known locations, and then builds a relationship between RSS and location. In the test phase, a user asks for position by sending currently measured RSS to the server. On the other hand, the server runs a localization algorithm and returns an estimate with context such as a map. Here, it is important to address the assumption that the server is always able to provide map information. The problem is that, during communicating map information, the privacy of both a user and a service provider can be invaded by disclosing the user's location and identity, and the service provider's data. The mapless localization proposed in this paper eliminates the need to disclose such information.

The mapless localization and navigation have been studied in many robotic and cybernetic applications. The works [1], [2] exploit IMU sensors to track the trajectory of a smartphone user. However, in general indoor environments, the accurate estimation for the velocity and the direction is difficult. For example, for obtaining the accurate velocity, the attitude of the smartphone needs to be kept fixed without rotating. More specifically, using a step detection for estimating the moving distance of a user involves significant error when the user swings the smartphone by walking. Also, the estimation of the heading direction is much biased due to the ferrous material and the accumulated gyro error. In [3], [4], a RSS propagation model is assumed to be so accurate that precise localization is possible without map information. However, the indoor signal propagation model significantly varies over area due to the multi-path fading problem caused by walls. In [5], [6], they rely on visual sensors that need to always face forward, which is impractical. This paper eliminates to need the particular behavior restriction about smartphone and the accurate signal propagation model.

Our indoor localization is characterized by relieving those restrictions and not using private data. Theses can facilitate a crowdsourcing localization in that the massive amount of the qualified training samples can be collected without costly efforts [7], [8]. This paper proposes a mapless localization algorithm by learning hidden intended trajectories from the trajectories sampled from crowds. The intended trajectories become the alternative map information to support the absence of the true map information.

The algorithm supporting the absence of map consists of two parts. First, it detects the start and the end points (landmarks) of any trajectory samples obtained from crowds, by recognizing the pattern of wifi signals. This work is necessary to obtain the similar trajectories that have the same start and end points. We combine linear discriminant analysis (LDA) and principal component analysis (PCA) [9], which play a role of both dimensionality reduction and clustering datapoints obtained from different landmarks. The landmark involves the featured sites such as room, toilet, front of elevator or stair.

Second, we learn scattered trajectory samples that have the same start and end points, into one intended trajectory. In order to match different length and time synchronization of the samples, we apply dynamic time warping with kalman smoothing [10], [11]. Also, robust trajectory learning is achieved, which allows to classify the outlier trajectory.

As the final estimator for a positioning, a combination of Gaussian process (GP) and a particle filter [12]–[14] is applied. The prior distribution of the particle filter is defined as a function of the learned trajectories. The likelihood is defined as GP output that models wifi RSS with respect to the corresponding positions [15].

Jaehyun Yoo and Karl H. Johansson are with the ACCESS Linnaeus Center and the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. H. Jin Kim is with the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea. {yjh5455@kth.se, hjinkim@snu.ac.kr, kallej@kth.se}. This work has been supported in part by the Knut and Alice Wallenberg Foundation, the Swedish Research Council, and the Swedish Foundation for Strategic Research, including the SSF-NRF Sweden-Korea research program.

978-1-5090-2425-4/16/$31.00 © 2016 IEEE

Our final contribution involves field experiments in an office building at Seoul National University. We obtain training datapoints from different people carrying a smartphone, and test our indoor localization algorithm for another participant. The participants are not given any guideline about restricted attitude to carry the smartphone, for example, not to swing the smartphone or not to put it in pocket. From the experimental results, we find out that the learned trajectories are very close to the true map. Therefore, the result of our indoor localization without map information is accurate almost as the result using the true map information.

The rest of this paper is organized as follows. Section II formalizes the wifi-based indoor localization using a combination of Gaussian process and the particle filter. Section III describes the contributed trajectory learning algorithm. In section IV, experiments and analysis are presented. Finally, section V is devoted to concluding remark.

II. Problem formulation

This section formalizes the indoor localization using smartphone and its measurement, i.e. wifi received signal strengths (RSSs) obtained between wireless access points (APs) and the smartphone. We use Bayesian filtering to estimate the location of a smartphone user from RSS measurements. In section II-A, Gaussian process (GP) is applied to find a relationship between measurements and positions. In section II-B, particle filter is used for combining the GP model and smoothing filters for indoor localization. The main motivation can be found while establishing the particle filter when we cast the question how we improve the localization without a true map information.

A. Gaussian process for modelling wifi RSS likelihood

First, we are interested in building a relationship between the 2-D position $\mathbf{x}$ and the corresponding RSS measurement y obtained from one wifi access point. Let $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be a set of l number of training datapoints drawn from

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

where the noise ε is drawn from normal distribution $N(0, \sigma_{gp}^2)$ with the known variance σ_{gp}^2. In the GP model, the joint distribution over the noisy training outputs, $Y = \{y_1, \ldots, y_l\}^{\mathrm{T}}$, is a function of the training inputs, $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l\}$, with the form

$$Y \sim N(0, K(\mathbf{X}, \mathbf{X}) + \sigma_{gp}^2 I), \tag{2}$$

where $K(\mathbf{X}, \mathbf{X})$ is $l \times l$ kernel matrix whose (i, j)-th element is $k(\mathbf{x}_i, \mathbf{x}_j)$. The squared exponential is a commonly used kernel function, given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \mathrm{exp}\left(\frac{- \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2}{2\theta_2}\right), \tag{3}$$

where θ_1 and θ_2 are hyperparameters of the kernel, which are learned by the conjugate gradient descent method [16].

After the training, the GP estimates the output value of the Gaussian process for the input value that is the location of interest $\mathbf{x}_*$ in our setting. The output takes Gaussian distribution with the mean $\mu_{\mathbf{x}_*}$ and variance $\sigma_{\mathbf{x}_*}^2$:

$$p\left(g(\mathbf{x}_*) | \mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) = N\left(g(\mathbf{x}_*); \mu_{\mathbf{x}_*}, \sigma_{\mathbf{x}_*}^2\right), \tag{4}$$

and

$$\begin{aligned} \mu_{\mathbf{x}_*} &= k_*^{\mathrm{T}} (K(\mathbf{X}, \mathbf{X}) + \sigma_{GP}^2 I)^{-1} Y \\ \sigma_{\mathbf{x}_*}^2 &= k(\mathbf{x}_*, \mathbf{x}_*) - k_*^{\mathrm{T}} (K(\mathbf{X}, \mathbf{X}) + \sigma_{GP}^2 I)^{-1} k_*, \end{aligned}$$

where k_* is the $l \times 1$ vector of covariances between $\mathbf{x}_*$ and $\mathbf{X}$.

The equation (4) describes the likelihood of RSS obtained from 'one' AP at particular position. Assuming that the RSSs from different wifi APs are independent, we can obtain the joint likelihood model as follows:

$$p(\mathbf{y}_t | \mathbf{x}_t) = \prod_{i=1}^{d} \left(p(y_i | \mathbf{x}_t)\right), \tag{5}$$

where d is the number of wifi APs, $p(y_i | \mathbf{x}_t)$ is the simplified notation of (4), t is time index, and $\mathbf{y}_t = [y_{1t}, \cdots, y_{dt}]^{\mathrm{T}}$ is a set of RSSs obtained from d different wifi APs at time t.

B. Particle filter for estimating location

Particle filter estimates the posterior probabilistic density function (pdf), $p\left(\mathbf{x}_t | \mathbf{y}_{1:t}\right)$, over a user's location $\mathbf{x}_t$ with given wifi RSSs $\mathbf{y}_{1:t}$. By Bayes's rule, the pdf can be given by:

$$p\left(\mathbf{x}_t | \mathbf{y}_{1:t}\right) = \frac{p\left(\mathbf{y}_t | \mathbf{x}_t\right) p\left(\mathbf{x}_t | \mathbf{y}_{1:t-1}\right)}{p\left(\mathbf{y}_t | \mathbf{y}_{1:t-1}\right)}, \tag{6}$$

where

$$p\left(\mathbf{x}_t | \mathbf{y}_{1:t-1}\right) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p\left(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}\right) d\mathbf{x}_{t-1}.$$

Because eqn. (6) cannot be calculated analytically except for some special cases such as the linear Gaussian state-space model (e.g., Kalman filter), the particle filter approximates the posterior pdf using a finite set of weighted particles. Let $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^{N_p}$ be a set of N_p number of particles and weights, where the weights are normalized, i.e. $\sum_{i=1}^{N_p} w_t^i = 1$. Each $\mathbf{x}_t^i$ represents the hypothetical state of the true state with the corresponding probabilistic value w_t^i. Thus, eqn. (6) is approximated as follows:

$$p\left(\mathbf{x}_t | \mathbf{y}_{1:t}\right) \approx \sum_{i=1}^{N_p} w_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i), \tag{7}$$

where $\delta(\cdot)$ is Dirac delta function. Then, the estimate of the state $\mathbf{x}_t$ can be described by:

$$\begin{aligned} \hat{\mathbf{x}}_t &= E[\mathbf{x}_t | \mathbf{y}_{1:t}] \\ &\approx \sum_{i=1}^{N_p} w_t^i \mathbf{x}_t^i. \end{aligned} \tag{8}$$

The weights can be obtained in the following:

$$\begin{aligned} w_t^i &= w_{t-1}^i \frac{p\left(\mathbf{y}_t | \mathbf{x}_t^i\right) p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)}{q\left(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}\right)} \\ &= w_{t-1}^i p\left(\mathbf{y}_t | \mathbf{x}_t^i\right). \end{aligned} \tag{9}$$

Thus, the weights are updated by likelihood $p(\mathbf{y}_t|\mathbf{x}_t^i)$ only. Note that the likelihood is defined as the result of GP in equation (5).

An important remaining part of the particle filter is sampling the particles using the prior probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The role of the prior is to smooth target trajectory and to generate particles in areas irrelevant to the true location. In many researches, the prior is modeled by:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = P_{dis} \cdot P_{map}, \tag{10}$$

where

$$P_{dis} = N\left(\parallel \mathbf{x}_t - \mathbf{x}_{t-1} \parallel; v\Delta t, \sigma_v \Delta t\right),$$

and

$$P_{map} = \begin{cases} 0 & \text{if a particle crossed a wall} \\ 1 & \text{if a particle did not cross a wall.} \end{cases}$$

Here, we tackle problems of using P_{dis} and P_{map} in the conventional indoor localization. The probability P_{dis} needs the velocity v and the variance σ_v of the user. However, obtaining accurate velocity requires the condition that the attitude of a smartphone has to be kept fixed without rotating, e.g. foot-mounted pedestrian tracking [17]. This is not practical because most people swing a smartphone when walking. Therefore, instead of P_{dis}, we apply Hodrick-prescott filter [18], P_{HF}, which obtains a smoothed-curve representation of a time-series trajectory, given by:

$$P_{HF} = N\left(\parallel \mathbf{x}_t - \mathbf{x}_{t-2} - 2\mathbf{x}_{t-1} \parallel^2; 0, \sigma_v\right), \tag{11}$$

where it does not require the velocity estimation.

The other issue is that the accuracy of the indoor localization is much affected by an informative map, i.e. P_{map}. If the map is unknown, the accuracy is dramatically reduced. Our major motivation is to tackle situations when the map is not available and the contribution is to generate an alternative function P_{TL} that imitates map information. The prior we will use is modified from (10) is as follows:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = P_{HF} \cdot P_{TL}, \tag{12}$$

and the specific description for P_{TL} will be shown in the next section.

III. Learning indoor trajectory

The purpose of the trajectory learning is to help localization accurate when map information is not available. In section III-A, we show the concept of the trajectory learning problem. Section III-B describes the specific trajectory learning algorithm and section III-C addresses the underlying issues to support the algorithm given in III-B.

A. Concept of learning indoor trajectory

The basic idea for learning the indoor trajectories comes from that people tend to walk similar trajectories when they have the same departure point and destination. In statistical view, when the trajectories $\mathbf{x}_{1:t}^{1:M}$ of M different people (they have the same departure and destination) are given, there exists a hidden intended trajectory $z_{1:t}$ that is representative of all $\mathbf{x}_{1:t}^{1:M}$. For example, $z_{1:t}$ can be average trajectory of $\mathbf{x}_{1:t}^{1:M}$.

For considering a realistic indoor localization, first we need to consider that each of $\mathbf{x}_{1:t}^{1:M}$ can have different length because people move at different speed. Second, we allow some people to have different trajectories although they have the same departure and destination. Therefore, the algorithm is required to detect those different trajectories (or outliers). Satisfying these conditions, dynamic time warping with kalman smoothing [10] is greatly suitable for learning indoor trajectory, which is described in the following section III-B.

On the other hand, the assumption that we need to know the departure and the destination, can be solved by recognizing patterns of wifi RSSs. We apply a machine learning technique for detecting landmarks such as the departure and the destination, which will be shown in section III-C.

B. Trajectory learning algorithm

We are given M trajectories $\mathbf{x}_j^k$ of length $N^{(k)}$, where $k = 0, \ldots, M-1$ and $j = 0, \ldots, N^{(k)} - 1$. Difference from the notation used in the previous section, i.e., $\mathbf{x}_{1:t}^{1:M}$, is because we allow the trajectories to have different length. Our goal is to find one hidden intended trajectory z_t of length T at $t = 0, \ldots, T-1$. The length T is set to the twice the average length of the trajectories, i.e., $T = 2/M \sum_{k=1}^{M} N^{(k)}$.

The trajectory learning algorithm considers the trajectories $\mathbf{x}_j^k$ (having the same departing point and destination), as the observations of the one intended trajectory, z_t. It is expressed as

$$\begin{aligned} z_{t+1} &= f(z_t) + w_t^z, \quad w^z \sim N(0, \Sigma^z) & (13) \\ \mathbf{x}_j^k &= z_{\tau_j^k} + w_j^{\mathbf{x}}, \quad w^{\mathbf{x}} \sim N(0, \Sigma^{\mathbf{x}}), & (14) \end{aligned}$$

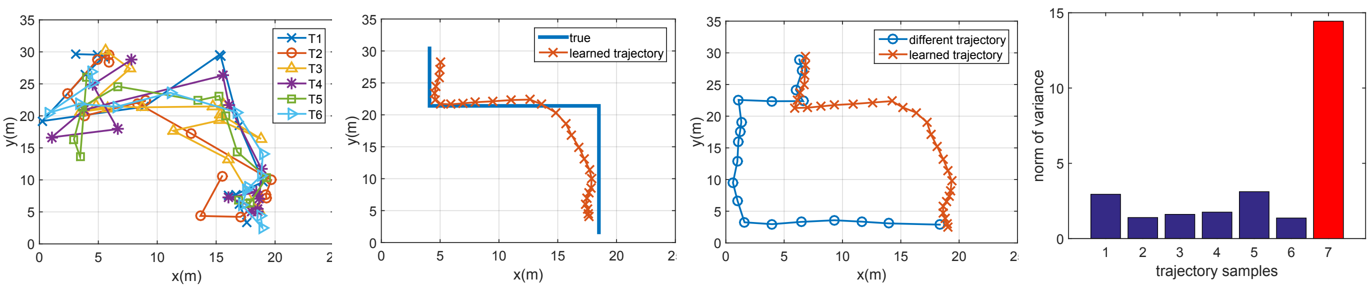
where w^z and $w^{\mathbf{x}}$ are the Gaussian noises whose covariance matrices Σ^z and $\Sigma^{\mathbf{x}}$ are to be estimated. The subscript τ_j^k is the time index $\mathbf{z}$ to which the observation $\mathbf{x}_j^k$ is mapped. The time indices τ_j^k are assumed to follow a multinomial distribution, i.e. $\tau_j^k \sim p(\tau_{j+1}^k|\tau_j^k)$.

Estimation of the hidden trajectory z_t and the time indices τ_j^k can be done by maximizing the following log-likelihood:

$$\max_{\tau, \Sigma^{(\cdot)}} \log p(z, \tau; \Sigma^{(\cdot)}), \tag{15}$$

where $\Sigma^{(\cdot)}$ denotes both Σ^z and $\Sigma^{\mathbf{x}}$. However, it is difficult to optimize the likelihood over $\Sigma^{(\cdot)}$ and τ simultaneously. We maximize (15) through two iterative algorithms. First, we update the covariance matrix $\Sigma^{(\cdot)}$ with the fixed τ. In E-step, the pairwise marginals over the latent variables $z_{1:t}$ are evaluated with the current $\Sigma^{(\cdot)}$, by using a Kalman smoother. Then, M-step uses theses marginals to update the covariance matrix $\Sigma^{(\cdot)}$.

Second, to optimize over the time-indexing variables τ, dynamic time warping (DTW) is used, where DTW is a sequence alignment algorithm by measuring similarity between two different signals.



(a) (b) (c) (d)

Fig. 1: Illustration of trajectory learning algorithm: the trajectories sampled from 6 different people (shown in (a)) are learned. The corresponding result is very close to the true trajectory as shown in (b). In (c), one outlier trajectory is added to the trajectories used in (a) and the result is still close to the true as shown in (c). We can detect the outlier trajectory by examining the norm of the covariance in (d), where the seventh outlier trajectory has an outstanding value.

We estimate the time index $\hat{\tau}$ that maximizes the log-likelihood at fixed $\Sigma^{(\cdot)}$ and z, given by:

$$\begin{aligned}\hat{\tau} &= \text{argmax}_{\tau} \text{log}p(z, \tau; \Sigma^{(\cdot)}) \\ &= \text{argmax}_{\tau} \sum_{k=0}^{M-1} \sum_{j=0}^{N^k-1} [\text{log}p(\mathbf{x}_j^k | z_{\tau_j^k | T}, \tau_j^k) \\ &\quad + \text{log}p(\tau_j^k | \tau_{j-1}^k)]. \end{aligned} \tag{16}$$

DTW solves the optimization problem (16) using dynamic programming. More detailed description for this algorithm can be found in [10].

Figs. 1(a) and 1(b) show the result of trajectory learning when 6 different people move the same intended trajectory. In Fig. 1(b), individual samples are made from the combination of the GP and the particle filter in section II. After the trajectory learning algorithm, the learned trajectory is very close to the true as shown in Fig. 1(a).

Until now, we assume that the trajectories, $\mathbf{x}_j^k$'s, are similar. We now suppose that some of $\mathbf{x}_j^k$'s are different (e.g., detour) although they have the same start and end points of the trajectories. It is possible to detect the outliers by examining the estimated covariance $\Sigma^{\mathbf{x}}$ because it implies the difference between the learned trajectory z and a sample trajectory $\mathbf{x}$ from the equations (13). Figs. 1(c) and 1(d) show this situation when the additional seventh person moves along another trajectory described in Fig. 1(c). Despite this outlier, the learning algorithm gives the accurate trajectory shown in Fig. 1(c), which is similar to the learned trajectory in Fig. 1(a). Also, when we examine the norm of covariance $\Sigma^{\mathbf{x}}$ of each trajectory, the different trajectory has a distinctive value in comparison with the others, as shown in in Fig. 1(d). From this, we can recognize the outlier trajectory.

C. Landmark detection

This section provides the detection of the start (departing point) and the end points (destination) in order to gather the similar trajectories. This is a necessary part to obtain the sample trajectories $\mathbf{x}_j^k$ that were used for making the hidden trajectory z in the previous section.

First of all, we define the specific site (or landmarks) which can be the start and the end points of any trajectory. In this paper, we define landmark locations as the front of elevator or stair, and the middle of room and toilet.

Let $W = \{(\mathbf{y}_i, w_i)\}_{i=1}^n$ be a set of n number of training datapoints, where $\mathbf{y} = [y_1, \cdots, y_d]^{\mathrm{T}}$ is a set of the RSSs obtained from d wifi APs, $w \in \{1, 2, \ldots, C\}$ denotes the label of the designated landmarks, and C is the number of landmarks. In many indoor areas, d is large. For example, 193 (d=193) APs are found in an office building of Seoul National University and 531 APs are used in the open UJIIndoorLoc dataset [19]. Therefore, we need a dimensionality reduction to deal with high dimensional data in a fast computation.

Also, $\mathbf{y}$ has that many empty elements because signal of one wifi AP does not reach the large indoor area. These empty elements are set to the minimum RSS value. The dimensionality reduction method plays a role of eliminating those meaningless elements.

Linear discriminant analysis (LDA) and principal component analysis (PCA) are fundamental dimensionality reduction techniques. LDA is a supervised dimensionality reduction in that only labeled training data are used, and PCA is an unsupervised method in that only unlabeled training data are used. Semi-supervised approaches can outperform the individual supervised or unsupervised methods by using both labeled and unlabeled data [20]. The following is a description for the semi-supervised dimensionality reduction algorithm based on LDA and PCA.

The between-class covariance matrix S_b, the within-class covariance matrix S_w, and the scatter matrix S are defined as:

$$S_b = \sum_{c=1}^{C} n_c (\mu_c - \mu)(\mu_c - \mu)^{\mathrm{T}}, \tag{17}$$

$$S_w = \sum_{c=1}^{C} \sum_{i:w_i=c} (\mathbf{y}_i - \mu_c)(\mathbf{y}_i - \mu_c)^{\mathrm{T}}, \tag{18}$$

$$S = \sum_{i=1}^{l} (\mathbf{y}_i - \mu)(\mathbf{y}_i - \mu)^{\mathrm{T}}, \tag{19}$$

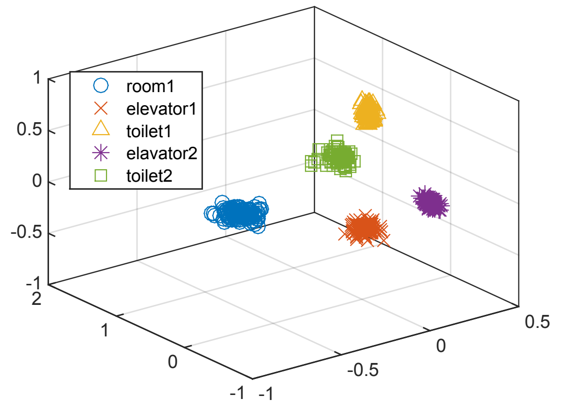


Fig. 2: Clustered landmark datapoints in the reduced RSS signal space using the semi-supervised PCA-LDA of section III-C, where the training datapoints of the landmarks are obtained on the 14-th floor of Bldg 301 of Seoul National University shown in Fig. 4.

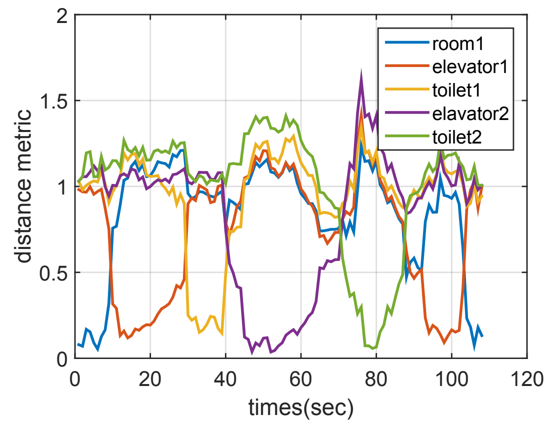


Fig. 3: Distance in the signal space between center of each cluster in Fig. 2 and the path of the user. The testing user moves along with "room1 $\rightarrow$ elevator1 $\rightarrow$ toilet1 $\rightarrow$ elevator2 $\rightarrow$ toilet2 $\rightarrow$ elevator1 $\rightarrow$ room1".

where n_c is the number of the labeled samples in class $c \in \{1, 2, \ldots, C\}$. Also, μ_c and μ are the mean vectors of the datapoints of class c and all datapoints, respectively, given by:

$$\mu_c = \frac{1}{n} \sum_{i:w_i=c} \mathbf{y}_i, \ \ \mu = \frac{1}{n_c} \sum_{i=1}^{n} \mathbf{y}_i.$$

LDA is to find the transformation matrix such that the between-class covariance matrix S_b in the embedding space is maximized and the within-class covariance matrix S_w in the embedding space is minimized. The transformation matrix T_{LDA} is given by the following generalized eigenvalue problem:

$$S_b T_{LDA} = S_w T_{LDA} \Lambda,$$

where Λ and T_{LDA} are matrices whose diagonal elements correspond to the generalized eigenvalues $\{\lambda_i\}_{i=1}^{d}$ and a matrix whose column vectors $\{\varphi_i\}_{i=1}^{d}$ correspond to the generalized eigenvectors, respectively.

Similarly, PCA finds the transformation matrix T_{PCA} in the following:

$$S T_{PCA} = T_{PCA} \Lambda.$$

To use the concept of semi-supervised approach, we modify the generalized eigenvalue problem as follows:

$$\begin{aligned} S_{lb} T &= S_{lw} T \Lambda, && (20)\\ S_{lb} = \alpha S_b &+ (1-\alpha) S, && (21)\\ S_{lw} = \alpha S_w &+ (1-\alpha) I, && (22) \end{aligned}$$

where I is the identity matrix, and α is the weight parameter on LDA and PCA.

Assuming $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, the transformation matrix Q is obtained as follows:

$$Q = \left(\sqrt{\lambda_1} \varphi_1 | \sqrt{\lambda_2} \varphi_2 | \cdots | \sqrt{\lambda_r} \varphi_r \right), \qquad (23)$$

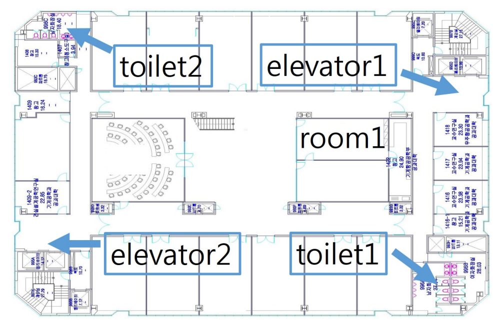


Fig. 4: Floor plan of the 14-th floor of Bldg 301 at Seoul National University.

where r is the dimension of the reduced space. Therefore, a low-dimensional representation $\bar{\mathbf{y}}_i \in \mathcal{R}^r$ of the original data $\mathbf{y}_i \in \mathcal{R}^d$ $(r \ll d)$ is obtained by

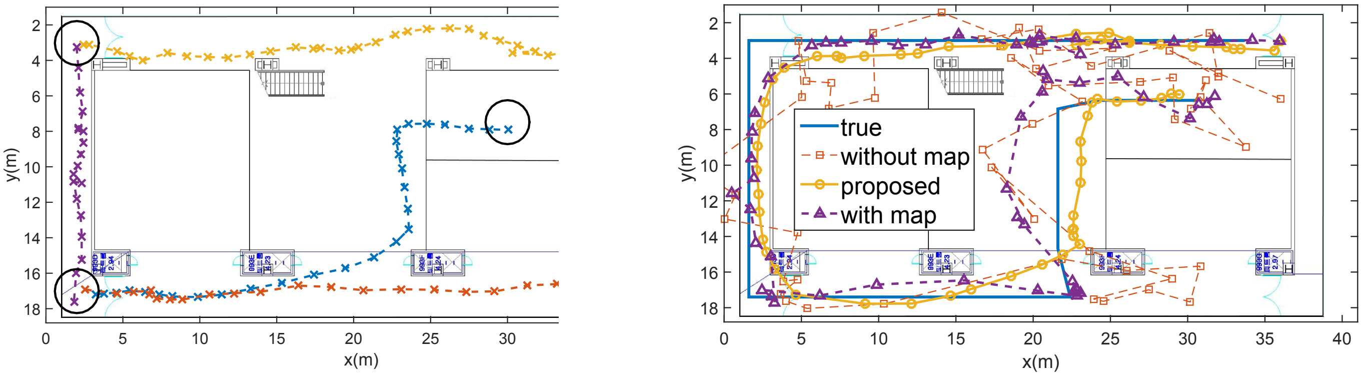
$$\bar{\mathbf{y}}_i = T^{\mathrm{T}} \mathbf{y}_i. \qquad (24)$$

Fig. 2 illustrates that different landmark datapoints have been clearly clustered in the reduced space. With the well-classified data, we detect the moment at which the user is located at the designated landmark points, based on the distance-metric threshold. We define the distance metric as:

$$D_c = \|\bar{\mu}_c - \bar{\mathbf{y}}_d\| \qquad (25)$$

which denotes the distance in the reduced space between center of each cluster, i.e. $\bar{\mu}_c = T^{\mathrm{T}} \mu_c$, and the test point $\bar{\mathbf{y}}_d$. Fig. 3 shows the distance metric D_c with respect to the movement of a user for 2 minute. Fig. 3 shows the accurate history of the varying distance between the designated landmarks and the user position, in the signal space.

After detecting landmarks of trajectory samples, we collect the refined trajectories. Let $\mathbf{z} = \{z^{(1)}, z^{(2)}, \cdots, z^{(m)}\}$ be a set of m learned trajectories, where each $z^{(\cdot)}$ has different start and end points. For example, departure and destination of $z^{(1)}$



(a) crossed line: learned trajectories, circles: the designated landmarks

(b) Comparison of different localizations

Fig. 5: Experimental results of the localization: (circle-line) with the proposed algorithm, (triangle-line) given map, and (square-line) without map. The proposed algorithm, which does not use the map information, gives similar performance to the case using the true map information.

are toilet and room, and those of $z^{(2)}$ are elevator and toilet, respectively. Keep in mind that the learned trajectories $\mathbf{z}$ are used for building the probability P_{TL}, in (12). We model P_{TL} as the Gaussian distribution:

$$P_{TL} \sim N\left(\mathbf{x}_t; \mu_{TL}, \Sigma_{TL}\right) \tag{26}$$

and

$$\mu_{TL} = \text{arg}_{\mathbf{z}}\text{min}\|\mathbf{z} - \mathbf{x}_t\|,$$

where μ_{TL} is the closest position among $\mathbf{z}$ to the particle sample $\mathbf{x}_t$. Also, the variance Σ_{TL} is defined as the estimated covariance, Σ^z in eqn. (13), which represents the degree at which the particles are allowed to be apart from the learned trajectory $\mathbf{z}$. Closing this section, we summarize the flow of our algorithm into **Algorithm 1**.

Algorithm 1 Pseudo code of the proposed indoor localization without map

Given:
· Gaussian process training data set, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$ in II-A.
· Landmark training data set, $\{(\mathbf{y}_i, w_i)\}_{i=1}^{n}$ in III-C.
Initial estimation phase:
· The prior $p(\mathbf{x}_{t+1}|\mathbf{x}_t) = P_{HF}$ in (11).
· The likelihood of GP $p(\mathbf{y}_t|\mathbf{x}_t)$ in (5).
· Run the particle filter and estimate trajectories in II. Theses trajectories are stored to update the prior in the update estimation phase.
Update estimation phase:
· Detect landmarks using the algorithm in III-C and collect the trajectories that have the same start and end landmarks.
· Do trajectory learning algorithm in III-B.
· Build the probability P_{TL} in (26).
· Update the prior $p(\mathbf{x}_{t+1}|\mathbf{x}_t) = P_{HF} \cdot P_{TL}$ in (12).
· Run the particle filter for localization in II.

IV. Experiments

A. Setup

Our experimental field is the 14-th floor of building 301 at Seoul National University, as shown in Fig 4. The used device is Samsung galaxy S4 whose cpu is 1.6 GHz with android OS and wifi communication uses IEEE 802.11 protocol. We program the smartphone application using Java eclipse for obtaining the wifi fingerprints, sensor measurements, and communicating with the laptop. The function of scanning wifi in android provides us the information of mac address, name of AP, and the decibel level of RSS. Ten different people participated in collecting training data and they are not given any guideline about their attitude, for example, not to swing the smartphone and not to put it in pocket. We record videos of the movement where Gopro camera was built on head, which can be used to extract the ground truth trajectories. The number of used wifi APs is 193, but we do not know their locations. We use 10-dimensional data from the 193 raw data through the dimensionality reduction described in section III-C.

B. Results

First of all, we show the learned trajectories that represent the alternative map information. The training participants in our experiments moved freely in the experimental area. After obtaining those localized results using **Initial estimation phase** in **Algorithm 1**, the landmark detection algorithm collects 44 trajectories and divides them into 5 kinds of the trajectories whose start and end points are the designated 5 landmarks respectively, as shown in Fig. 5(a). Then, the trajectory learning algorithm finds only 5 learned trajectories as shown in Fig. 5(a). The learned trajectories are very helpful for the localization in our algorithm, in Fig. 5(b), in that the result of the proposed algorithm is almost accurate as the result using the true map information. The average distance errors between the estimated positions and the true are 3.15 m (the proposed), 2.95 m (given map), and 4.15 m (without map), respectively. A few among the estimated positions, in our algorithm, stray from the map because it cannot perfectly restrict the estimates in the reachable area as much as the algorithm using the true map. This makes a slight difference between our algorithm and the localization using the map. The result of the proposed algorithm much outperforms the result without the map. From this result, we confirm that our algorithm is useful for the situation when a map is not available.

V. Conclusion

This paper addressed the indoor localization that uses no map information and preserves the privacy of the participants. The proposed trajectory learning algorithm provided the similar information map to the true map so that achieved accurate localization results. Because this trajectory learning algorithm does not require any particular behavior restriction to the participants, it is well-suited for crowdsourcing. Our crowdsourcing localization can continuously update the pseudo-map more accurately, which is useful for assisting humans effort and robotic navigating in indoor area.

References

[1] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 6, pp. 889–898, 2012.

[2] T. Yairi, "Map building without localization by dimensionality reduction techniques," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1071–1078.

[3] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 173–184.

[4] L. Bruno and P. Robertson, "Wislam: Improving footslam with wifi," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*. IEEE, 2011, pp. 1–10.

[5] Z. Mathews, M. Lechón, J. B. Calvo, A. Dhir, A. Duff, P. F. Verschure, *et al.*, "Insect-like mapless navigation based on head direction cells and contextual learning using chemo-visual sensors," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 2243–2250.

[6] C. Giovannangeli, P. Gaussier, and G. Désilles, "Robust mapless outdoor vision-based navigation," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 3293–3300.

[7] V. Huang, H. Huang, S. Thatipamala, and C. J. Tomlin, "Contrails: Crowd-sourced learning of human models in an aircraft landing game," in *Proceedings of the AIAA GNC Conference*, 2013, pp. 591–594.

[8] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *Internet Computing, IEEE*, vol. 16, no. 5, pp. 36–44, 2012.

[9] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local fisher discriminant analysis for dimensionality reduction," *Machine learning*, vol. 78, no. 1-2, pp. 35–61, 2010.

[10] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[11] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, 2010.

[12] J. Ko and D. Fox, "Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[13] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models." in *IJCAI*, vol. 7, 2007, pp. 2480–2485.

[14] F. Duvallet and A. D. Tews, "Wifi position estimation in industrial environments using gaussian processes," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 2216–2221.

[15] J. H. Yoo, W. Kim, and H. J. Kim, "Event-driven gaussian process for object localization in wireless sensor networks," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2790–2795.

[16] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[17] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications, IEEE*, vol. 25, no. 6, pp. 38–46, 2005.

[18] M. O. Ravn and H. Uhlig, "On adjusting the hodrick-prescott filter for the frequency of observations," *Review of economics and statistics*, vol. 84, no. 2, pp. 371–376, 2002.

[19] S. Joaquin, M. Raul, M.-U. Adolfo, J. A. Tomar, P. A. Joan, B.-B. Mauri, and H. Joaquin, "Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems," in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*. IEEE, 2014.

[20] O. Chapelle, B. Schölkopf, A. Zien, *et al.*, *Semi-supervised learning*. MIT press Cambridge, 2006.