


Article

Time-Series Laplacian Semi-Supervised Learning for Indoor Localization [†]

Jaehyun Yoo 

Department of Electrical, Electronic and Control Engineering, Hankyong National University, Anseoung 17579, Korea; jhyoo@hknu.ac.kr

[†] This paper is an extended version of our paper published in Yoo, J.; Johansson, K.H. Semi-supervised learning for mobile robot localization using wireless signal strengths. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017.

Received: 10 July 2019; Accepted: 5 September 2019; Published: 7 September 2019



Abstract: Machine learning-based indoor localization used to suffer from the collection, construction, and maintenance of labeled training databases for practical implementation. Semi-supervised learning methods have been developed as efficient indoor localization methods to reduce use of labeled training data. To boost the efficiency and the accuracy of indoor localization, this paper proposes a new time-series semi-supervised learning algorithm. The key aspect of the developed method, which distinguishes it from conventional semi-supervised algorithms, is the use of unlabeled data. The learning algorithm finds spatio-temporal relationships in the unlabeled data, and pseudolabels are generated to compensate for the lack of labeled training data. In the next step, another balancing-optimization learning algorithm learns a positioning model. The proposed method is evaluated for estimating the location of a smartphone user by using a Wi-Fi received signal strength indicator (RSSI) measurement. The experimental results show that the developed learning algorithm outperforms some existing semi-supervised algorithms according to the variation of the number of training data and access points. Also, the proposed method is discussed in terms of why it gives better performance, by the analysis of the impact of the learning parameters. Moreover, the extended localization scheme in conjunction with a particle filter is executed to include additional information, such as a floor plan.

Keywords: Wi-Fi RSSI-based indoor localization; semi-supervised learning; time-series learning

1. Introduction

Wi-Fi received signal strength indicator (RSSI) is one of the basic sensory observations widely used for indoor localization. Due to its nonlinear and random properties, many machine learning approaches have been applied to Wi-Fi RSSI localization [1–5]. In particular, semi-supervised learning algorithms have been suggested to improve the efficiency, which reduces the human effort necessary for calibrating training data. For example, a large amount of unlabeled data can be easily collected by recording only Wi-Fi RSSI measurements, without assigning position labels, which can save resources for collection and calibration. By contrast, labeled training data must be created manually. Adding a large amount of unlabeled data in the semi-supervised learning framework can prevent the decrement in localization accuracy when using a small amount of labeled data.

The majority of the existing semi-supervised learning algorithms use the unlabeled data for the manifold regularization that captures the intrinsic geometric structure of the whole training data [6–10]. More progressed usage of unlabeled data is the pseudolabeling where unlabeled data are artificially labeled, and it is used for learning the estimation model. In [11–13], pseudolabels are iteratively updated, and the localization model is learned by the final pseudolabels. In [14], usage

of unlabeled data avoids the biased parameter estimation against a small number of labeled data points, to obtain the probabilistic location-RSSI model. In [15,16], unlabeled data are used to find an embedding function from RSSI signal space to 2D location space. Semi-supervised deep learning approaches [17–20] improve positioning accuracy by using the unlabeled data for extracting features from the RSSI measurements in deep neural network framework. Compared to discriminative model such as support vector machine (SVM); however, the deep learning methods normally take much time to finish the learning.

This paper proposes a new semi-supervised learning algorithm by interpreting the unlabeled data as spatio-temporal data. In the previous paper [21], the older version of the semi-supervised learning algorithm was applied to a mobile robot in a room size toy experiment. In this paper, the scalable algorithm has been developed for public indoor localization. The algorithm has two separate pseudolabeling and learning processes. First, in the pseudolabeling process, the time-series graph Laplacian SVM optimization is constructed, which estimates labels of the unlabeled data. The existing Laplacian SVM algorithms [6–10] consider only spatial relationship for the training data. To add the temporal meaning to the unlabeled data, this paper attaches the time-series regularization into the Laplacian SVM framework. Time-series learning is reasonable for the indoor localization targeting a smoothly moving human, and the corresponding data comes in a chronological order. As a result, the accurate pseudolabels are made by the proposed time-series semi-supervised learning.

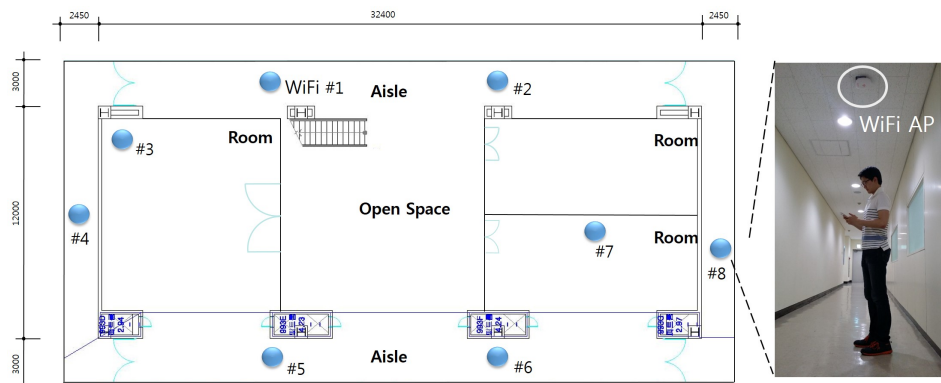
Second, in the learning process, another learning framework to estimate position is constructed. Because the pseudolabels are artificial, the pseudolabeled data cannot be reliable as much as the labeled data. Therefore, it is desirable to limit the reliance on the pseudolabeled data. This can be dealt with a balancing optimization by weighting different balancing parameters between the labeled and pseudolabeled data. The existing semi-supervised methods [22], Ref. [11] might address this balance issue. In [22], the optimization framework consists of a singular value decomposition, a manifold regularization, and a loss function. In [11], the intermediate variable is introduced as a substitute of the original labeled data based on the graph Laplacian optimization framework. However, because many balancing parameters in [22], Ref. [11] are coupled to both the labeled and the unlabeled (or pseudolabeled) terms, it is difficult to adjust the balance. Also, as fewer labeled data are used, the pseudolabels become inaccurate. The proposed method solves this imbalance problem by adding Laplacian Least Square (LapLS) that is produced in this paper by combining the manifold regularization into the transductive SVM (TSVM [23]) structure. Because the pseudolabels are used as the learning input of LapLS, the proposed optimization becomes a linear problem that can be solved fast. In this learning process, two decoupled balancing parameters are individually weighed to the labeled term and the pseudolabeled term, separately, which makes it simple to balance the labeled and the pseudolabeled data. This balancing optimization adjusts the reliance on the pseudolabeled data relative to the labeled data. Outstanding performance of the proposed method is found even when a small amount of labeled training data are used.

The proposed algorithm is evaluated to estimate location of a smartphone carried by a user, and it is compared with some SVM-oriented semi-supervised algorithms. The accurate performance is validated by the analysis of the impact of the learning parameters. Also, according to the variation of the number of the labeled training data and the Wi-Fi access points, the proposed algorithm gives the best localization performance without sacrificing the computation time. In addition, the combination of the proposed learning-based estimation and the particle filter is implemented.

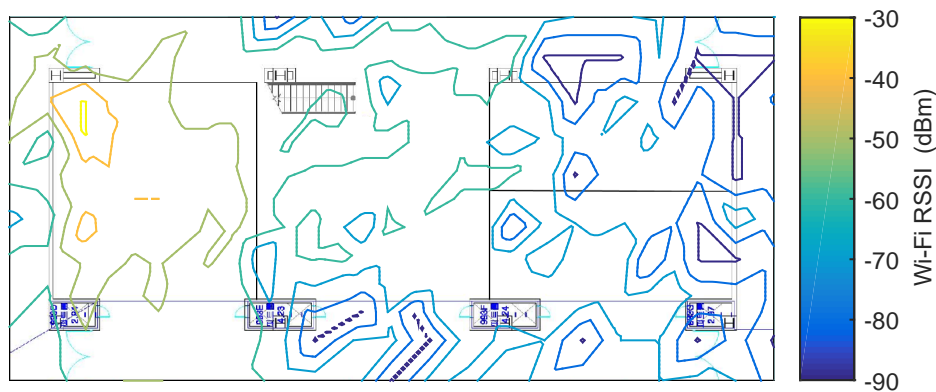
This paper is organized as follows. Section 2 overviews the learning-based indoor localization problem with description of experimental setup for Wi-Fi RSSI use. Section 3 presents the semi-supervised learning algorithms based on the graph Laplacian manifold learning. Section 4 devotes to introduce the proposed algorithm. Section 5 reports empirical results with the parameter setting, the localization with the variance of the number of the training data and access points, the computational time, and the filter-combined localization. Finally, concluding remarks are given in Section 6.

2. Learning-Based Indoor Localization

Figure 1a shows the localization setup where 8 Wi-Fi access points (APs) are deployed in the $37 \times 18 \text{ m}^2$ floor. The APs are pre-installed by Korean telecommunication companies such as SKT and LGT. By distinguishing APs' unique MAC (media access control) addresses, it can define a concentrated set by assigning each RSSI measurement sent from the different APs to the specific location in the set. The experimental floor consists of 3 rooms, open space, and aisle. Each room is filled with desks and chairs, and there are some obstacles such as a vase and a rubbish bin in places on the floor. 5 different people join to collect the training data. In case of collecting the labeled data, the trainers are guided to record the labeled data points on every $1.5 \times 1.5 \text{ m}^2$ grid. During the calibration, the repeated measurement sets whose labels are the same location are averaged. As a result, 283 labeled training data are obtained. Similarly, another user is employed to collect 283 labeled *test* data, which are not used for any training algorithm. The smartphone device is Samsung Galaxy S7 operated by Android OS. The smartphone application to measure the Wi-Fi RSSIs and the locations is developed by Java Eclipse. The Wi-Fi scan function in the Android platform provides the information of MAC address and name of AP, and the RSSI value in dBm. Also, in this experiment, only 2.4 GHz RSSI signal can be collected.

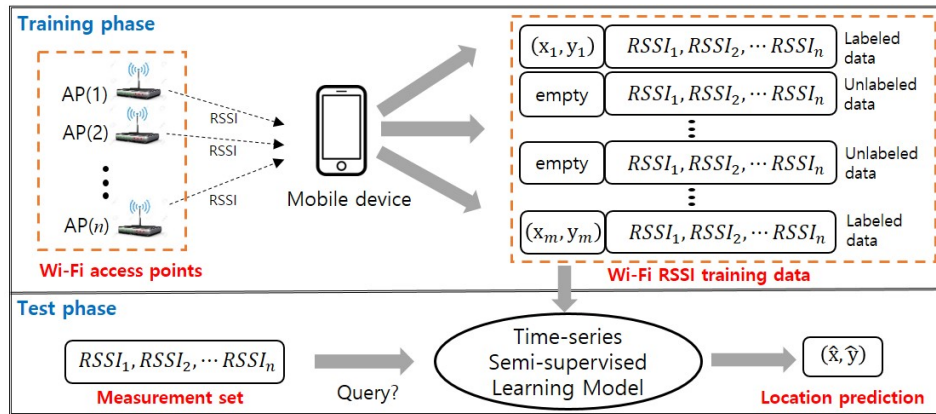


(a) Testbed in $37 \text{ m} \times 18 \text{ m}$ with 8 Wi-Fi access points.



(b) RSSI distribution sent from #3 Wi-Fi access point across the floor

Figure 1. Cont.



(c) Learning-based localization using Wi-Fi RSSI.

Figure 1. (a) Experimental floor with the 8 visible Wi-Fi access points, (b) Wi-Fi RSSI distribution of the #3 access point across the floor, and (c) learning-based localization architecture using Wi-Fi RSSI.

Labeled training data are obtained by placing the receiver at different locations. Let us define the Wi-Fi observation set as $x_i = \{z_{i1}, \dots, z_{in}\} \in \mathcal{R}^n$ received from n different APs ($n = 8$ in this paper), where z_{ij} ($1 \leq j \leq n$) is a scalar RSSI measurement sent from the j -th access point corresponding to the user's location (y_{Xi}, y_{Yi}) . Total of the l labeled training data are given by $\{x_i\}_{i=1}^l$ with $x_i \in X \subseteq \mathcal{R}^n$, and $\{y_{Xi}\}_{i=1}^l, \{y_{Yi}\}_{i=1}^l$. The unlabeled data set $\{x_i\}_{i=l+1}^{l+u}$ consists of only the RSSI measurements, without position labels. The training phase builds separate mappings $f_X : X \rightarrow \mathcal{R}$ and $f_Y : X \rightarrow \mathcal{R}$ which refer to relationships between Wi-Fi signal strength and location, using the labeled training data $\{(x_i, y_{Xi})\}_{i=1}^l$ and $\{(x_i, y_{Yi})\}_{i=1}^l$, respectively, and the unlabeled data $\{x_i\}_{i=l+1}^{l+u}$. Because the models f_X and f_Y are learned independently, we omit the subscripts of f_X, f_Y , and y_X, y_Y , for simplification.

Figure 1c illustrates overview of the proposed indoor localization architecture using Wi-Fi RSSI. The main purpose of the semi-supervised learning-based localization is to learn the accurate positioning model even when a small amount of labeled data is used. For example, Figure 1b shows the Wi-Fi RSSI distribution of the access point #3 that is in a room. Contribution of the semi-supervised learning is to prevent the distribution from being distorted when using much less labeled training data. This will be restated in Example 1 of Section 4.1.

3. Semi-Supervised Learning

This section describes basic semi-supervised learning framework in Section 3.1 and reviews Laplacian least square SVR (LapLS) in Section 3.2 and Laplacian embedded regression least square (LapERLS) in Section 3.3. Key ideas from these algorithms will be applied for the proposed algorithm in the next Section 4.

3.1. Basic Semi-Supervised SVM Framework

Given a set of l labeled samples $\{(x_i, y_i)\}_{i=1}^l$ and a set of u unlabeled samples $\{x_i\}_{i=l+1}^{l+u}$, Laplacian semi-supervised learning aims to establish a mapping f by the following regularized minimization functional [24]:

$$f^* = \arg \min_{f \in \mathcal{H}_k} C \sum_i V(x_i, y_i, f) + \gamma_A \|f\|_A^2 + \gamma_I \|f\|_I^2, \quad (1)$$

where V is a loss function, $\|f\|_A^2$ is the norm of the function in the Reproducing Kernel Hilbert Space (RKHS) \mathcal{H}_k , $\|f\|_I^2$ is the norm of the function in the low-dimensional manifold, and C, γ_A, γ_I are the regularization weight parameters.

The solution of (1) is defined as an expansion of kernel function over the labeled and the unlabeled data, given by

$$f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) + b, \quad (2)$$

with the bias term b and the kernel function $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, where $\phi(\cdot)$ is a nonlinear mapping to RKHS.

The regularization term $\|f\|_A^2$ associated with RKHS is defined as

$$\|f\|_A^2 = (\Phi\alpha)^T(\Phi\alpha) = \alpha^T K\alpha, \quad (3)$$

where $\Phi = [\phi(x_1), \dots, \phi(x_{l+u})]$, $\alpha = [\alpha_1, \dots, \alpha_{l+u}]^T$, and K is the $(l+u) \times (l+u)$ kernel matrix whose element is K_{ij} . We adopt Gaussian kernel given by

$$K_{ij} = K(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / \sigma_k^2\right), \quad (4)$$

where σ_k^2 is the kernel width parameter.

According to the manifold regularization, data points are samples obtained from a low-dimensional manifold embedded in a high-dimensional space. This is represented by the graph Laplacian:

$$\begin{aligned} \|f\|_l^2 &= \frac{1}{(l+u)^2} \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} W_{ij} (f(x_i) - f(x_j))^2, \\ &= \frac{1}{(l+u)^2} \mathbf{f}^T L \mathbf{f}, \end{aligned} \quad (5)$$

where L is the normalized graph Laplacian given by $L = D^{-1/2}(D - W)D^{-1/2}$, $\mathbf{f} = [f(x_1), \dots, f(x_{l+u})]^T$, W is the adjacency matrix of the data graph, and D is the diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. In general, the edge weights W_{ij} are defined as Gaussian function of Euclidean distance, given by

$$W_{ij} = \exp\left(-\|x_i - x_j\|^2 / \sigma_w^2\right), \quad (6)$$

where σ_w^2 is the kernel width parameter.

Minimizing $\|f\|_l^2$ is equivalent to penalizing the rapid changes of the regression function evaluated between two data points. Therefore, $\gamma_l \|f\|_l^2$ in (1) controls the smoothness of the data geometric structure.

3.2. Laplacian Least Square (LapLS)

This section describes developing of the LapLS algorithm algorithm by combining manifold regularization defined in (5) and transductive SVM (TSVM) [23]. In LapLS, the loss function V in (1) is defined by

$$V(x_i, y_i, f) = e_i = y_i - \left(\sum_{i=1}^{l+u} \alpha_i K(x_i, x) + b \right). \quad (7)$$

LapLS finds optimal parameters α, b , and the labels y_1^*, \dots, y_u^* of the unlabeled data when regularization parameters C and C^* are given:

$$\begin{aligned} \min_{\alpha, e, e^*, b, y_1^*, \dots, y_u^*} & \frac{C}{2} \sum_{i=1}^l e_i^2 + \frac{C^*}{2} \sum_{j=1}^u (e_j^*)^2 + \gamma_A \alpha^T K \alpha + \gamma_l \alpha^T K L K \alpha, \\ \text{subject to : } & y_i - \sum_{k=1}^{l+u} \alpha_k K_{ik} - b - e_i = 0, \quad i = 1, \dots, l, \\ & y_j^* - \sum_{k=1}^{l+u} \alpha_k K_{jk} - b - e_j^* = 0, \quad j = 1, \dots, u. \end{aligned} \quad (8)$$

Optimization in (8) with respect to all y_1^*, \dots, y_u^* is a combinatorial problem [23]. To find the solution, we must search over all possible 2^u labels of the unlabeled data. Therefore, this method is not useful when a large amount of the unlabeled data is applied.

3.3. Laplacian Embedded Regularized Least Square (LapERLS)

LapERLS introduces an intermediate decision variable $g \in \mathcal{R}^{(l+u)}$ and additional regularization parameter γ_C into the Laplacian semi-supervised framework (1), as follows [11]:

$$\min_{f \in \mathcal{H}_k, g \in \mathcal{R}^{(l+u)}} C \sum_{i=1}^{l+u} V(x_i, g_i, f) + \gamma_C \sum_{i=1}^l (g_i - y_i)^2 + \gamma_A \|f\|_A^2 + \gamma_I \|g\|_I^2.$$

The optimization problem of (9) enforces the intermediate decision variable g to be close to the labeled data and to be smooth with respect to the graph manifold.

Loss function is given by:

$$V(x_i, g_i, f) = \xi_i = g_i - \left(\sum_{j=1}^{l+u} \alpha_j K(x_i, x_j) \right). \quad (9)$$

After reorganizing the terms in (9) with respect to manifold regularization and decision function and corresponding parameter, the primal optimization problem is as follows:

$$\begin{aligned} \min_{\alpha, g, \xi \in \mathcal{R}^{(l+u)}} & \frac{C}{2} \sum_{i=1}^{l+u} \xi_i^2 + \alpha^T K \alpha + \frac{1}{2} (g - y)^T \Lambda (g - y) + \frac{1}{2} \mu g^T L g, \\ \text{subject to: } & \xi_i = g_i - \sum_{k=1}^{l+u} \alpha_k K_{ik}, \quad i = 1, \dots, l+u, \end{aligned} \quad (10)$$

where Λ is a diagonal matrix of trade-off parameters with $\Lambda_{ii} = \lambda$ if x_i is a labeled data point, and $\Lambda_{ii} = 0$ if x_i is unlabeled, $y = [y_1, \dots, y_l, 0, \dots, 0]^T \in \mathcal{R}^{(l+u)}$, and C, λ, μ are tuning parameters.

Also, dual formulation of (10) is given by:

$$\min_{\beta \in \mathcal{R}^{(l+u)}} \frac{1}{2} \beta^T \tilde{Q} \beta + \beta^T \tilde{y}, \quad (11)$$

where

$$\begin{aligned} \tilde{Q} &= K + (\Lambda + \mu L)^{-1}, \\ \tilde{y} &= (\Lambda + \mu L)^{-1} \Lambda y, \\ \beta &= -\alpha. \end{aligned} \quad (12)$$

The main characteristics of this method lies in using \tilde{y} as the input to learning, unlike standard semi-supervised learning that uses $y = [y_1, \dots, y_l, 0, \dots, 0]^T$. In other words, zero values in y are modified to some values denoting pseudolabels of unlabeled data.

It is noted that when the original labeled set y is replaced with the intermediate decision variable g , the accuracy of the learning may decrease. Moreover, when amount of available labeled data is small, the accuracy of the pseudolabels, which is difference between true labels of unlabeled data points and pseudolabels of the unlabeled data points, decreases significantly.

4. Time-Series Semi-Supervised Learning

This section describes a new algorithm by extracting key ideas from LapLS and LapERLS introduced in the previous section. In Section 4.1, a time-series representation is added to the unlabeled data to obtain pseudolabels. In Section 4.2, the pseudolabels are used in LapLS structure to derive an optimal solution by balancing the pseudolabeled and the labeled data. Notations are equivalent to the previous section.

4.1. Time-Series LapERLS

In [25], a time-series learning optimization problem is suggested by applying Hodric-Prescott (H-P) filter [26], which can capture a smoothed-curve representation of a time-series from the training data, given by

$$\min_f \sum_{i=1}^t (f(x_i) - y_i)^2 + \gamma_T \sum_{i=3}^t (f(x_i) + f(x_{i-2}) - 2f(x_{i-1}))^2, \quad (13)$$

where $\{(x_i, y_i)\}_{i=1}^t$ is the time-series labeled training data. The second term is to make the sequential points $f(x_i), f(x_{i-1}), f(x_{i-2})$ on a line. The solution of (13) in the matrix form is,

$$f = (I + \gamma_T DD^T)^{-1} y,$$

where

$$D = \begin{bmatrix} 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}_{t \times t}. \quad (14)$$

The main idea for a new semi-supervised learning is to assign additional temporal meaning to the unlabeled data. In the proposed algorithm, the H-P filter is added into the LapERLS formulation (9) in the following optimization:

$$\min_{f \in \mathcal{H}_k, g \in \mathcal{R}^{(l+u)}} C \sum_{i=1}^{l+u} V(x_i, g_i, f) + \gamma_C \sum_{i=1}^l (g_i - y_i)^2 + \gamma_A \|f\|_A^2 + \gamma_I \|g\|_I^2 + \gamma_T \sum_{i=3}^{l+u} (g(x_i) + g(x_{i-2}) - 2g(x_{i-1}))^2. \quad (15)$$

After rearranging (15) using the process similar to Equations (9)–(11), the optimization form of the proposed time-series LapERLS is given by:

$$\min_{\beta \in \mathcal{R}^{(l+u)}} \frac{1}{2} \beta^T \tilde{Q} \beta + \beta^T \tilde{y}, \quad (16)$$

where

$$\tilde{Q} = K + (\Lambda + \mu_1 L + \mu_2 DD^T)^{-1}, \quad (17)$$

$$\tilde{y} = (\Lambda + \mu_1 L + \mu_2 DD^T)^{-1} \Lambda y, \quad (18)$$

$$\beta = -\alpha.$$

In comparison with (12) of the standard LapERLS, $\mu_2 DD^T$ is added to (18). In the following, the experimental examples for describing the difference of the standard LapERLS and the time-series LapERLS are introduced.

Example 1. Figure 2 examines the accuracy comparison between the standard LapERLS and the time-series LapERLS. In this example, a time-series training data set is collected as a user moves along the path illustrated in Figure 2a. This simulation uses 20% of the labeled training data and 80% of the unlabeled data among total 283 training data. Figure 2 illustrates estimations of the pseudolabels using each time-series LapERLS and standard LapERLS. As shown in Figure 2b,c, the pseudolabels produced by the time-series LapERLS are accurate while the standard LapERLS does not show meaningful pseudolabels. Therefore, the trajectory of pseudolabels from

the time-series LapERLS can recover the true trajectory while the standard LapERLS cannot. Obviously, many incorrect pseudolabels such as Figure 2c will derive inaccurate localization performance.

The other physical interpretation about the pseudolabels can be seen from Figure 3. Figure 3a shows the RSSI distribution as the ground truth by using the entire labeled training data, Figure 3b shows the RSSI distribution recovered by the pseudolabels obtained by the time-series LapERLS, and Figure 3c is the RSSI distribution estimated by the pseudolabels of the standard LapERLS. In case of the standard LapERLS as shown in Figure 3c, the distribution is severely distorted due to the incorrectly estimated pseudolabels. On the other hand, the time-series LapERLS gives the very similar distribution to the original distribution.

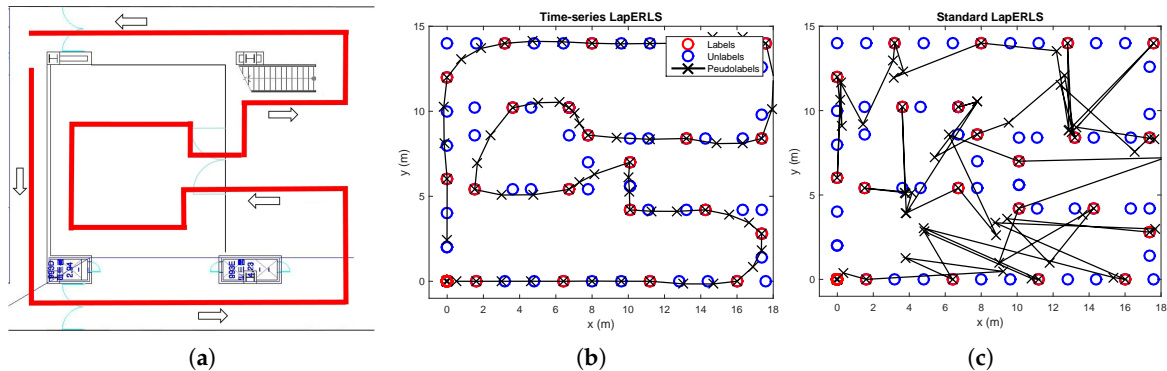


Figure 2. Comparison of accuracy of pseudolabels between time-series LapERLS (18) and standard LapERLS (12). The labeled, unlabeled, and pseudolabeled data points are marked by red circles, blue circles, and black crosses, respectively. (a) A user follows the red path and collects the labeled and unlabeled data. (b) The resultant pseudolabels from time-series LapERLS are so accurate that the trajectory made by the pseudolabels is close to the user's true path. (c) Due to inaccurate pseudolabels obtained from the standard LapERLS, this trajectory is severely incorrect.

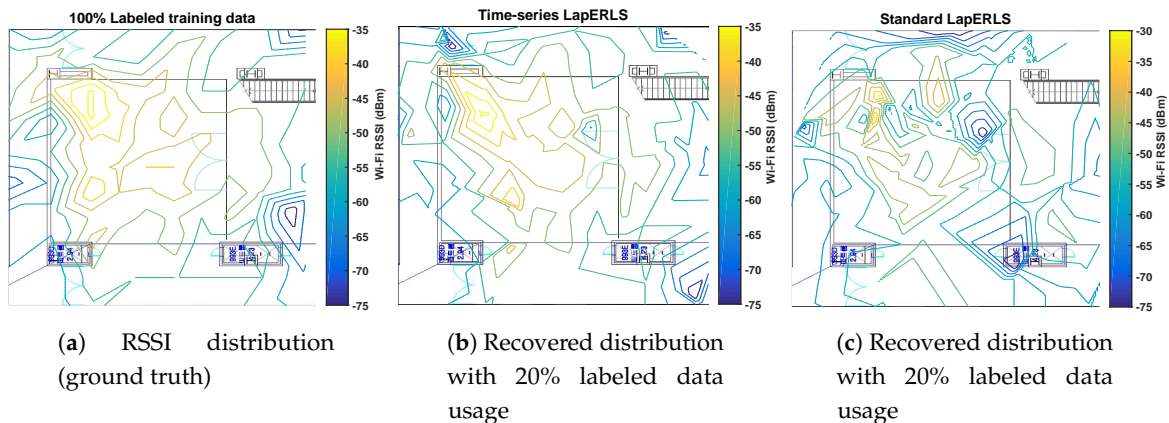


Figure 3. Comparison of the Wi-Fi RSSI distributions made by the pseudolabels between the time-series LapERLS (proposed) and the standard LapERLS (compared). In (a), the original RSSI distribution sent from #3 access point marked in Figure 1 is given as the ground truth and is made by using the entire labeled training data. In (b), the time-series LapERLS algorithm with usage of 20% of the labeled training data produces the similar distribution to the original distribution due to the accurately estimated pseudolabels. In (c), the standard LapERLS with usage of 20% labeled training data produces the severely distorted distribution.

Example 2. Accuracy of the pseudolabels is examined by the three cases with comparison to a linear interpolation.

- Sinusoidal trajectory:

A situation that a user moves a sinusoidal trajectory as described in Figure 4a is considered. The linear interpolation produces the pseudolabels laying on the straight line between two consecutive labeled points. On the other hand, because the proposed algorithm considers both spatial and temporal relation by using manifold and time-series learning, the result of the suggested algorithm can generate accurate pseudolabels as shown in Figure 4b.

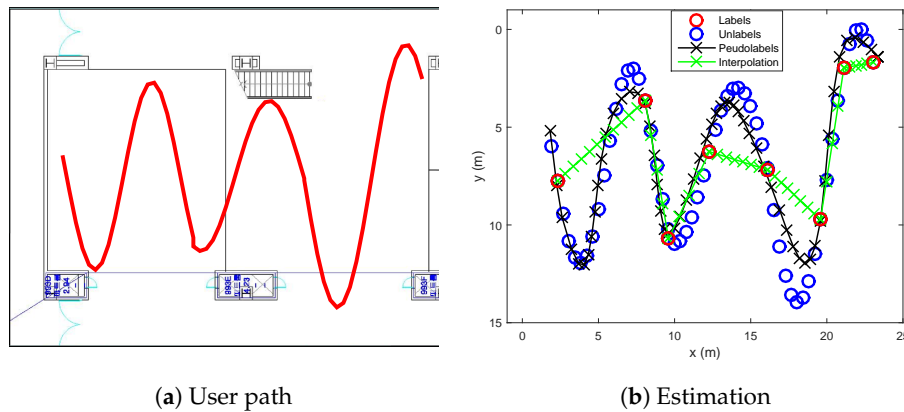


Figure 4. Estimated pseudolabels on the sinusoidal trajectory in comparison with the linear interpolation.

- Wandered trajectory:

The other case in which the linear interpolation is not useful is when the user does not walk straight forward to a destination. For example, in Figure 5, the user wanders around middle of the path. Because there are only two labeled data (one is at bottom, and the other is at top), the linear interpolation could not represent the wandered trajectory. In Figure 5b, it is shown that the developed algorithm generates accurate pseudolabels with respect to the wandered motion of the user.

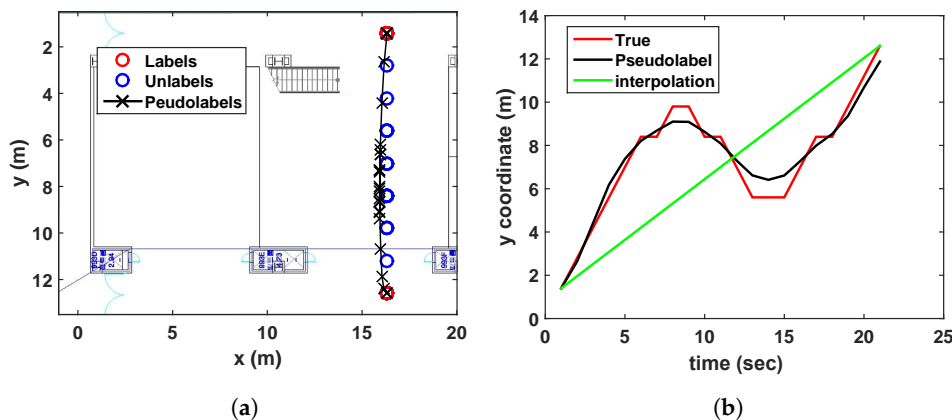


Figure 5. Estimated pseudolabels on the wandered trajectory in (a), in comparison with the linear interpolation in (b).

- Revisiting the learned trajectory:

The user used to revisit the same site during collecting training data. Suppose that the locations corresponding to the Wi-Fi RSSI measurements are not recorded during walking a path, except the

start and end points as shown in Figure 6. It is assumed that we have already learned those area. The result under this situation is shown in Figure 6, where the developed algorithm generates accurate pseudolabels, while the linear interpolation cannot reflect the reality.

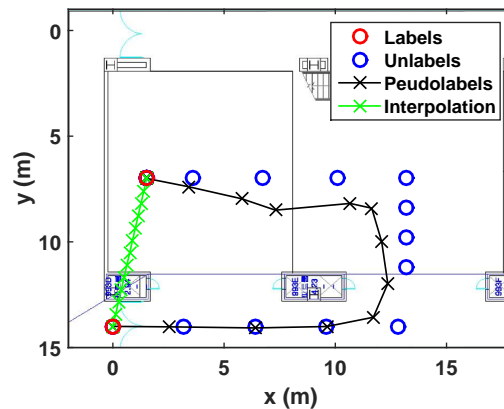


Figure 6. Estimated pseudolabels when revisiting the learned trajectory in comparison with the linear interpolation.

4.2. Balancing Labeled and Pseudolabeled Data

Regardless of how accurate the pseudolabels are, it is impossible to regard the pseudolabels as the labeled data, because true labels of the unlabeled data are unknown. One desirable approach is to properly balance the labeled and the pseudolabeled data in the learning process. This is feasible by applying LapLS structure in Section 3.2, which can control the balance of training data by the decoupled parameters C and C^* introduced in (8).

Example 3. Figure 7 illustrates an estimation of the sine function using LapLS in Section 3.2, where we divide the labeled training set in half and use the different values of the parameters, i.e., $C = 0.5$ and $C^* = 0.1$. In the latter part, the estimation with $C^* = 0.1$ is not accurate. This result can be validated from (8). As the parameter $C^{(*)}$ becomes smaller, the related term $C^{(*)} \sum_i (e_i^{(*)})^2$ becomes also smaller. In other words, the optimization focuses less on the training data points with the smaller parameter value of $C^{(*)}$.

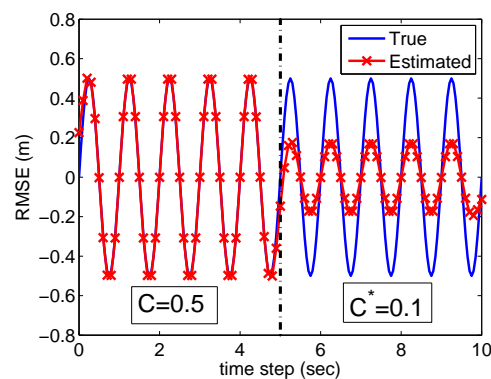


Figure 7. Sine function estimation using LapLS in Section 3.2 with different values of balancing parameters C and C^* .

For the final step in the suggested algorithm, the idea is to treat pseudolabels \tilde{y} in (18) as the labels of the unlabeled data in the LapLS (8) framework, which forms the following optimization:

$$\begin{aligned} \min_{\alpha \in \mathcal{R}^{(l+u)}, \rho \in \mathcal{R}^l, e^* \in \mathcal{R}^u, b \in \mathcal{R}} & \frac{C}{2} \sum_{i=1}^l e_i^2 + \frac{C^*}{2} \sum_{j=1}^u (\tilde{e}_j^*)^2 + \gamma_A \alpha^T K \alpha + \gamma_I \alpha^T K L K \alpha, \\ \text{subject to : } & y_i - \sum_{k=1}^{l+u} \alpha_k K_{ik} - b - e_i = 0, \quad i = 1, \dots, l, \\ & \tilde{y}_j^* - \sum_{k=1}^{l+u} \alpha_k K_{jk} - b - \tilde{e}_j^* = 0, \quad j = 1, \dots, u, \end{aligned} \quad (19)$$

where \tilde{y}_j^* are pseudolabels of the unlabeled data from \tilde{y} (18). Therefore, the non-convex problem of LapLS (8) is modified to a convex problem due to insertion of the pseudolabels. After KKT conditions, we obtain the following linear system:

$$A \mathbf{X} = \mathbf{Y}, \quad (20)$$

with

$$\begin{aligned} A &= \begin{bmatrix} K + \Gamma & \mathbf{1}_{(l+u) \times 1} \\ \mathbf{1}_{1 \times (l+u)} & 0 \end{bmatrix}, \\ \mathbf{X} &= \begin{bmatrix} \alpha \\ b \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} \tilde{Y} \\ 0 \end{bmatrix}, \end{aligned}$$

where K is the kernel matrix in (4), $\alpha = [\alpha_1, \dots, \alpha_{l+u}] \in \mathcal{R}^{(l+u)}$, $b \in \mathcal{R}$ in (2) is the bias value, $\mathbf{1}_{(l+u) \times 1} = [1, \dots, 1]^T \in \mathcal{R}^{(l+u)}$ is the one vector, and Γ is the diagonal matrix with $\Gamma_{ii} = 1/C$ for $i = 1, \dots, l$ and $\Gamma_{ii} = 1/C^*$ for $i = l+1, \dots, l+u$. The pseudolabel vector $\tilde{Y} \in \mathcal{R}^{(l+u)}$ is a time-series set of the labeled and pseudolabeled data. The optimal solution α^* and b^* obtained after solving (20) becomes the final parameters of the localization model, which is defined in (2). In the test phase, when a user queries location by sending a RSSI measurement set, the location is estimated based on the learned localization model.

The proposed algorithm in (20) improves the performance of LapERLS by combining the structure of LapLS. First, the pseudolabels are accurately estimated by assigning temporal-spatio representation into the unlabeled data. Second, it is easy to adjust the balance between the pseudolabels and the labeled data. Moreover, by incorporating the pseudolabels into the LapLS structure, the non-convex problem is transformed to a convex problem that can be computed far faster. The proposed algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed semi-supervised learning for localization

- Step 1: Collect labeled training data set $\{(x_i, y_i)\}_{i=1}^l$ and unlabeled data set $\{x_j\}_{j=1}^u$ in a time-series.
 - Step 2: Obtain kernel matrix K in (4), normalized Laplacian matrix L in (5) and matrix Λ in (10).
 - Step 3: Choose values of μ_1 and μ_2 in (18), and then calculate the pseudolabels in (18).
 - Step 4: Choose values of C and C^* , and then solve the linear equation in (20).
 - Step 5: Based on the optimal solution α^* and b^* from Step 4, builds the localization model in (2).
-

5. Experiments

To evaluate the proposed algorithm in comparison with other semi-supervised learning algorithms, two kinds of error are defined. First is the pseudolabel error defined as the average of the distance errors between unlabeled data and pseudolabeled data. Second is the test error is defined as the average of the distance errors between true locations and estimates.

This section consists of the parameter setting in Section 5.1 and the compared localization results according to the variation of the training data in Section 5.2. In Section 5.3, the result according to

the variation of the number of Wi-Fi access points and the computation analysis are given. Finally, Section 5.4 describes a combination of particle filter and the suggested learning algorithm.

5.1. Parameter Setting

The hyper-parameters in machine learning algorithms are generally tuned by cross validation that selects parameters values to minimize training error of some split training data sets, e.g., 10-fold cross validation [27]. However, most semi-supervised learning applications use a small number of the labeled data, so it is not suitable to employ the cross validation. This section provides a guideline for selecting all the parameters used in the developed algorithm, by describing the physical meaning of each parameter.

First, λ , i.e., the diagonal elements of Λ in Step 2 of Algorithm 1, can be interpreted as the importance of the labeled data relative to the unlabeled data by (12) and (18). If λ is relatively smaller than the particular value defined as $\lambda_{critical}$, resultant pseudolabels of the labeled data are different from the true labels. Therefore, it is decided to select value of λ larger than such critical value. Figure 8a shows the pseudolabel error according to the variation of λ , where $\lambda_{critical} = 10$ becomes the proper parameter selection.

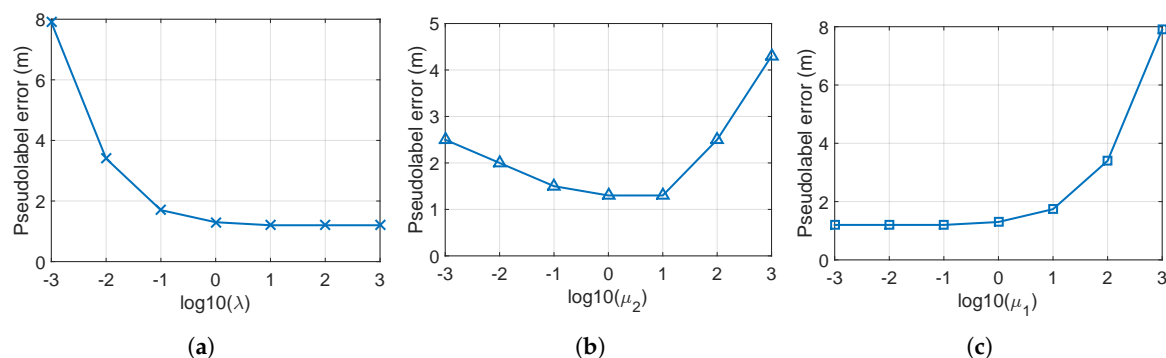


Figure 8. Impact of the parameters λ in Step 2, μ_2 , and μ_1 in Step 3 of Algorithm 1. (a) Impact of the value λ when $\mu_1 = 1$, $\mu_2 = 1$. (b) Impact of the ratio μ_2/μ_1 ; we set $\mu_1 = 1$ and vary μ_2 . (c) Impact of the value of μ_1 and μ_2 if the ratio μ_2/μ_1 is fixed, we vary μ_1 and μ_2 with constraint $\mu_2/\mu_1 = 1$.

Second, selection of μ_1 and μ_2 in Step 3 of Algorithm 1 represents a trade-off relationship between spatial and temporal correlation. If it is desirable to weigh the temporal meaning more than the spatial meaning, μ_2 is selected larger than μ_1 . Figure 8b shows the error of the pseudolabels according to the variation of μ_2 at fixed μ_1 , which suggests that $1 < \mu_2 < 10$ is proper for this data set. Also, Figure 8c gives the impact of the ratio μ_2/μ_1 , where $\mu_2/\mu_1 < 1$ is proper to this data set.

Third, selection of C and C^* in Step 4 of Algorithm 1 represents a trade-off relationship about the relative importance between the labeled data and the pseudolabeled data, as described in V-B. It is highlighted again that C^* should be smaller than C if we want to reduce the reliance of the pseudolabeled data relative to the labeled data. We used 10%, 25%, and 50% labeled data points among total of 283 labeled data points to test an effect of the parameters C and C^* whose results are shown in Figure 9a,b. From Figure 9a, it is observed that C^* smaller than C can reduce the test error. In particular, the same test error is found at $C^* = 1$, which implies that the algorithm with only 10% labeled data can have the same performance when using large amount of 50% used labeled data. This result highlights the importance of balancing the labeled and the pseudolabeled data and proves why the proposed algorithm can give good results when using a small number of the labeled data points. Also, Figure 9b shows the impact of values of C and C^* when the ratio C/C^* is fixed. It is shown that $10 < C/C^* < 50$ is proper for this data set. Lastly, Figure 9c shows the test error with respect to the variance of σ_k used for the kernel matrix. Table 1. summarizes the parameter values used for the rest of the experiments.

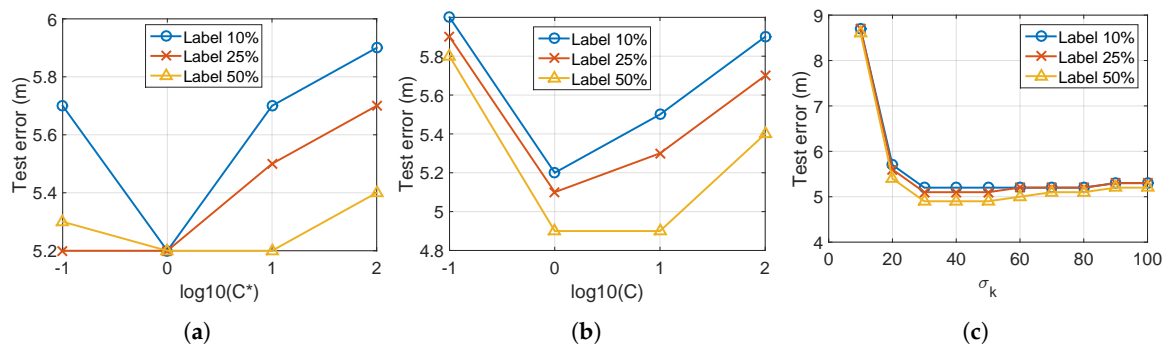


Figure 9. Impact of the parameters C , C^* , and σ_k in Step 4 of Algorithm 1. Other parameters used in the prior Step 3 are set to $\mu_1 = 1$, $\mu_2 = 3$, and $\lambda = 5$. (a) Impact of the ratio C^*/C when $C = 40$. (b) Impact of the values of C and C^* if the ratio C/C^* is fixed, varying C and C^* with constraint $C/C^* = 1$. (c) Impact of the value of σ_k when $C = 40$ and $C^* = 40$.

Table 1. Selected parameter values.

Value of λ	Value of μ_1	Value of μ_2	Value of C	Value of C^*	Value of σ_k
10	5	2	10	1	30

5.2. Variation of Number of Training Data

This experiment compares the semi-supervised algorithms, i.e., SSL [28], SSC [22], LapERLS [11] to the proposed algorithm with respect to the variation of the number of labeled training data at the fixed total of the training data, and the results are shown in Figure 10. For example, in case of 75% labeled data, 25% unlabeled data is used. The total of 283 and 93 training data points are used in Figure 10a,b, respectively. From the both results, we can observe that our algorithm outperforms the compared methods. In cases of 100% and 75% labeled data in Figure 10b, SSC gives slightly smaller error than the others. However, considering the advantage given to only SSC, i.e., the information of locations of the Wi-Fi access points, this error reduction is not noticeable.

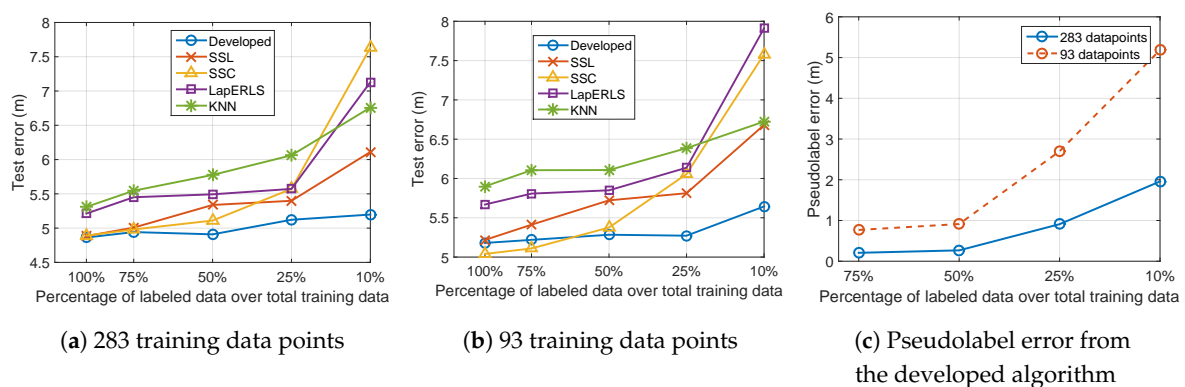


Figure 10. Localization results of the compared algorithms with respect to the variation of the ratio of the labeled training data over (a) total of 283 training data points and (b) total of 93 training data points. In (c), accuracy of the pseudolabels obtained from the developed algorithm is examined according to variation of the number of labeled data.

The major contribution of our algorithm is found from the results when a small amount of labeled data is used. From Figure 10a,b, our algorithm shows the slightly increasing errors as less labeled data are used, while the others give the highly increasing error from 25% percentage. To analyze

this result, we check the error of the pseudolabeled data points made in our algorithm, in Figure 10c. When using many labeled training data points such as 50%~75%, accurate pseudolabels are made, so good learning result is obvious. Even though the pseudolabeled data is inaccurate such that only 10% labeled data is used as in Figure 10c, the test error is still small as in Figure 10a,b. This is because we balanced the pseudolabeled data by reducing the reliance on the pseudolabeled data relative to the labeled data.

5.3. Variation of Number of Wi-Fi Access Points and Computational Time

Figure 11 compares the results when the number of Wi-Fi access points (AP) is changed from 2 to 7, where the APs depicted in Figure 1 are removed by this order #1 → #6 → #3 → #7 → #8 → #4.

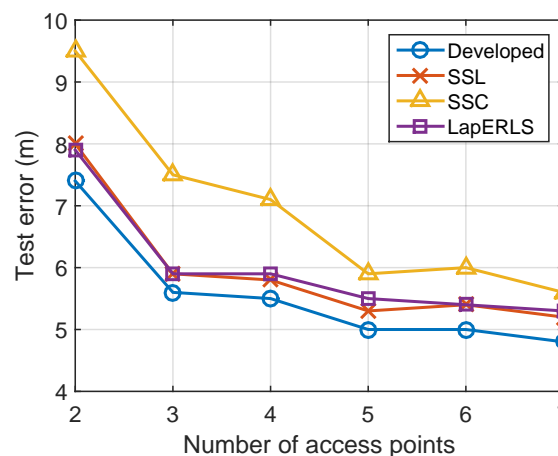


Figure 11. Impact of the number of Wi-Fi access points for localization performance.

Figure 12 summarizes the computational training times of the compared algorithms. Computational complexity of the optimization of SSL, LapERLS, and the proposed algorithm is $\mathcal{O}(dn^2 + n^3)$ [29] by the matrix computation and its inversion in (18), where d is the dimension of training data and n is the number of training data. In case of SSC, it additionally needs the computation of harmonic function using k -NN search algorithm which is $\mathcal{O}(nk(n + d)\log(n + d))$ [30]. The computation time of SSC increases significantly according to the increasing amount of the labeled data while the others remain lower-bounded in 0.5 s. The proposed algorithm needs little more time than LapERLS and SSL (at most 0.2 s).

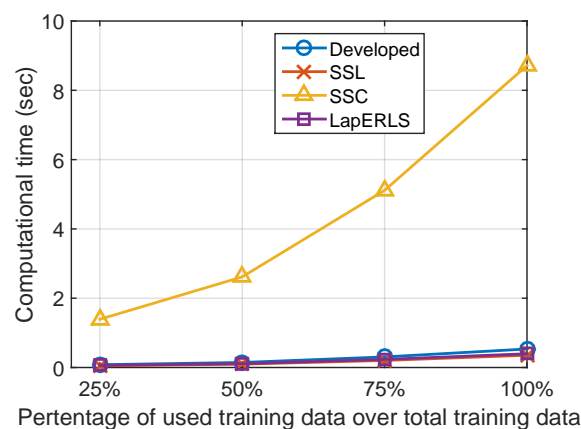


Figure 12. Computational time of the compared algorithms.

5.4. Combination with Particle Filter

While the previous sections have shown the results of localization using the Wi-Fi RSSIs based on the learning algorithms, this section provides a combination with a filtering algorithm. An application of the particle filter can use other kinds of positioning information such as an IMU measurement and map information, to improve indoor localization. Because a detailed description about the particle filter can be found in many works [31–33], this paper simply describes how that information with the learning-based localization can be combined in the particle filter framework.

First, a dynamic model given an IMU sensor data can be defined by:

$$X_{t+1} = X_t + \hat{V}_t \Delta t \begin{bmatrix} \cos(\hat{\theta}_t) \\ \sin(\hat{\theta}_t) \end{bmatrix}, \quad (21)$$

where X_t is a 2-D location at time step t , and \hat{V}_t and $\hat{\theta}_t$ are the velocity and direction of an user, respectively. However, in (21), accurate estimation for the velocity and the direction is difficult to be obtained for the indoor positioning. For example, for obtaining accurate velocity, the attitude of the smartphone needs to be kept fixed without rotating. More specifically, the estimates of a step detection and velocity of the user have the significant error when the user swings the smartphone by walking. Also, the estimation of the direction is much biased in the indoor environment due to the ferrous material and the integral gyro error.

For such reasons, an alternative dynamic model without employing IMU measurement is the random walk defined by the Gaussian distribution as follows:

$$X_{t+1} \sim N(X_t, \Sigma_d). \quad (22)$$

Next, the particle filter uses the estimated location obtained from the learning algorithm as the observation model in the following:

$$Z_t = HX_t + \Gamma,$$

where $Z_t = [f_X, f_Y]^T$ is the estimated location from the learning algorithm f_X , f_Y , and H is the identity matrix, and Γ is a Gaussian noise whose mean is zero and variance is Σ_o .

The key point of the particle filter is to update the weights w_t^i of each particle X_t^i for $i = 1, \dots, m$, where m is the number of the particles. It decides the estimation \hat{X}_t as the weighted summation $\hat{X}_t = \sum_{i=1}^m w_t^i X_t^i$. The weights are updated in the following:

$$w_t^i = w_{t-1}^i \cdot P(Z_t | X_t^i) \cdot P(X_t^i | X_{t-1}^i),$$

where $P(Z_t | X_t^i) \sim N(Z_t - HX_t^i, \Sigma_o)$. The map information can be used in the probability $P(X_t^i | X_{t-1}^i)$ in the following:

$$P(X_t^i | X_{t-1}^i) = \begin{cases} P_m & \text{if a particle crossed a wall} \\ 1 - P_m & \text{if a particle did not cross a wall,} \end{cases}$$

where P_m simply set to zero because human cannot pass the wall.

Evaluation of the particle filter based on the learning algorithm is shown in Figure 13. It is assumed that the user walks constant speed along the designated path. By the records of timestamp using the smartphone, we calculate the average 5.2 m localization error when using only learning method, and 2.9 m localization error when combining the particle filtering.

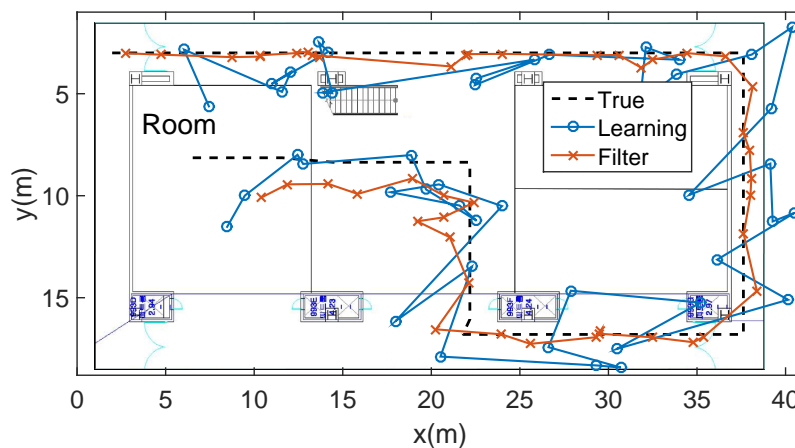


Figure 13. Combination of the particle filter and the learning algorithm, given map information. The test data is obtained through the dotted line. Blue circle line is the result when using only learning localization. Red cross line is the result when we combine the learning with the filter, illustrating a smoother and more accurate trajectory.

6. Conclusions

This paper proposes a new semi-supervised learning algorithm by combining core concepts of LapERLS's pseudolabeling, time-series learning, and LapLS-based balancing optimization. From the indoor localization experiment, the suggested algorithm achieves high accuracy by using only a small amount of the labeled training data in comparison with the other semi-supervised algorithms. Also, this paper provides a guidance and an analysis for selecting all the parameters used in the proposed algorithm. In addition to the Wi-Fi RSSI-based localization, there are various types of indoor localization methods using IMU, Bluetooth, UWB (Ultrawide band) and camera. This paper has shown the combination with the particle filter from which expandability to involve other types of information is validated. Thus, it is expected that a fusion algorithm with the other positioning approaches might improve localization accuracy for future work.

Funding: This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by ICT R&D program of MSIP/IITP, 2017-0-00543, Development of Precise Positioning Technology for the Enhancement of Pedestrian's Position/Spatial Cognition and Sports Competition Analysis.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Hernández, N.; Ocaña, M.; Alonso, J.; Kim, E. Continuous space estimation: Increasing WiFi-based indoor localization resolution without increasing the site-survey effort. *Sensors* **2017**, *17*, 147. [[CrossRef](#)] [[PubMed](#)]
2. Zheng, L.; Hu, B.; Chen, H. A high accuracy time-reversal based WiFi indoor localization approach with a single antenna. *Sensors* **2018**, *18*, 3437. [[CrossRef](#)]
3. Wang, Y.; Xiu, C.; Zhang, X.; Yang, D. WiFi indoor localization with CSI fingerprinting-based random forest. *Sensors* **2018**, *18*, 2869. [[CrossRef](#)] [[PubMed](#)]
4. Nuño-Maganda, M.; Herrera-Rivas, H.; Torres-Huitzil, C.; Marisol Marin-Castro, H.; Coronado-Pérez, Y. On-Device learning of indoor location for WiFi fingerprint approach. *Sensors* **2018**, *18*, 2202. [[CrossRef](#)] [[PubMed](#)]
5. Botta, M.; Simek, M. Adaptive distance estimation based on RSSI in 802.15. 4 network. *Radioengineering* **2013**, *22*, 1162–1168.
6. Zhou, M.; Tang, Y.; Nie, W.; Xie, L.; Yang, X. GrassMA: Graph-based semi-supervised manifold alignment for indoor WLAN localization. *IEEE Sens. J.* **2017**, *17*, 7086–7095. [[CrossRef](#)]
7. Zhang, L.; Valaee, S.; Xu, Y.; Ma, L.; Vedadi, F. Graph-based semi-supervised learning for indoor localization using crowdsourced data. *Appl. Sci.* **2017**, *7*, 467. [[CrossRef](#)]

8. Du, B.; Xinyao, T.; Wang, Z.; Zhang, L.; Tao, D. Robust graph-based semisupervised learning for noisy labeled data via maximum correntropy criterion. *IEEE Trans. Cybern.* **2018**, *49*, 1440–1453. [[CrossRef](#)]
9. Wang, M.; Fu, W.; Hao, S.; Tao, D.; Wu, X. Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1864–1877. [[CrossRef](#)]
10. Yoo, J.; Kim, H. Target localization in wireless sensor networks using online semi-supervised support vector regression. *Sensors* **2015**, *15*, 12539–12559. [[CrossRef](#)]
11. Chen, L.; Tsang, I.W.; Xu, D. Laplacian embedded regression for scalable manifold regularization. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 902–915. [[CrossRef](#)] [[PubMed](#)]
12. Nie, F.; Xu, D.; Li, X.; Xiang, S. Semisupervised dimensionality reduction and classification through virtual label regression. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2011**, *41*, 675–685.
13. Kumar Mallapragada, P.; Jin, R.; Jain, A.K.; Liu, Y. Semiboost: Boosting for semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 2000–2014. [[CrossRef](#)] [[PubMed](#)]
14. Ouyang, R.W.; Wong, A.K.S.; Lea, C.T.; Chiang, M. Indoor location estimation with reduced calibration exploiting unlabeled data via hybrid generative/discriminative learning. *IEEE Trans. Mobile Comput.* **2012**, *11*, 1613–1626. [[CrossRef](#)]
15. Jain, V.K.; Tapaswi, S.; Shukla, A. RSS Fingerprints Based Distributed Semi-Supervised Locally Linear Embedding (DSSLLE) Location Estimation System for Indoor WLAN. *Wirel. Pers. Commun.* **2013**, *71*, 1175–1192. [[CrossRef](#)]
16. Xia, Y.; Ma, L.; Zhang, Z.; Wang, Y. Semi-Supervised Positioning Algorithm in Indoor WLAN Environment. In Proceedings of the IEEE Vehicular Technology Conference, Glasgow, UK, 11–14 May 2015; pp. 1–5.
17. Mohammadi, M.; Al-Fuqaha, A.; Guizani, M.; Oh, J.S. Semisupervised deep reinforcement learning in support of IoT and smart city services. *IEEE Internet Things J.* **2017**, *5*, 624–635. [[CrossRef](#)]
18. Gu, Y.; Chen, Y.; Liu, J.; Jiang, X. Semi-supervised deep extreme learning machine for Wi-Fi based localization. *Neurocomputing* **2015**, *166*, 282–293. [[CrossRef](#)]
19. Khatab, Z.E.; Hajihoseini, A.; Ghorashi, S.A. A fingerprint method for indoor localization using autoencoder based deep extreme learning machine. *IEEE Sens. Lett.* **2017**, *2*, 1–4. [[CrossRef](#)]
20. Jiang, X.; Chen, Y.; Liu, J.; Gu, Y.; Hu, L. FSELM: Fusion semi-supervised extreme learning machine for indoor localization with Wi-Fi and Bluetooth fingerprints. *Soft Comput.* **2018**, *22*, 3621–3635. [[CrossRef](#)]
21. Yoo, J.; Johansson, K.H. Semi-supervised learning for mobile robot localization using wireless signal strengths. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation, Sapporo, Japan, 18–21 September 2017; pp. 1–8.
22. Pan, J.J.; Pan, S.J.; Yin, J.; Ni, L.M.; Yang, Q. Tracking mobile users in wireless networks via semi-supervised colocalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 587–600. [[CrossRef](#)]
23. Chapelle, O.; Vapnik, V.; Weston, J. Transductive Inference for Estimating Values of Functions. In Proceedings of the Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 1999; Volume 12, pp. 421–427.
24. Belkin, M.; Niyogi, P. Semi-supervised learning on Riemannian manifolds. *Mach. Learn.* **2004**, *56*, 209–239. [[CrossRef](#)]
25. Tran, D.A.; Zhang, T. Fingerprint-based location tracking with Hodrick-Prescott filtering. In Proceedings of the IFIP Wireless and Mobile Networking Conference, Vilamoura, Portugal, 20–22 May 2014; pp. 1–8.
26. Ravn, M.O.; Uhlig, H. On adjusting the Hodrick-Prescott filter for the frequency of observations. *Rev. Econ. Stat.* **2002**, *84*, 371–376. [[CrossRef](#)]
27. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the IJCAI, Montreal, QC, Canada, 20–25 August 1995; Volume 14, pp. 1137–1145.
28. Yoo, J.; Kim, H.J. Online estimation using semi-supervised least square svr. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, USA, 5–8 October 2014; pp. 1624–1629.
29. Chapelle, O. Training a support vector machine in the primal. *Neural Comput.* **2007**, *19*, 1155–1178. [[CrossRef](#)] [[PubMed](#)]
30. Chapelle, O.; Zien, A. Semi-Supervised Classification by Low Density Separation. In Proceedings of the AISTATS, Bridgetown, Barbados, 6–8 January 2005; pp. 57–64.
31. Gao, Z.; Mu, D.; Zhong, Y.; Gu, C. Constrained Unscented Particle Filter for SINS/GNSS/ADS Integrated Airship Navigation in the Presence of Wind Field Disturbance. *Sensors* **2019**, *19*, 471. [[CrossRef](#)] [[PubMed](#)]

32. Dampf, J.; Frankl, K.; Pany, T. Optimal particle filter weight for bayesian direct position estimation in a gnss receiver. *Sensors* **2018**, *18*, 2736. [[CrossRef](#)] [[PubMed](#)]
33. Gao, W.; Wang, W.; Zhu, H.; Huang, G.; Wu, D.; Du, Z. Robust Radiation Sources Localization Based on the Peak Suppressed Particle Filter for Mixed Multi-Modal Environments. *Sensors* **2018**, *18*, 3784. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).