

Article

Lightweight Workload Fingerprinting Localization Using Affinity Propagation Clustering and Gaussian Process Regression

Santosh Subedi  and Jae-Young Pyun * 

Department of Information and Communication Engineering, Chosun University, Kwangju 501-759, Korea; santoshmsubedi@chosun.kr

* Correspondence: jypyun@chosun.ac.kr; Tel.: +82-62-230-6199

Received: 12 November 2018; Accepted: 3 December 2018; Published: 4 December 2018



Abstract: Fingerprinting localization approach is widely used in indoor positioning applications owing to its high reliability. However, the learning procedure of radio signals in fingerprinting is time-consuming and labor-intensive. In this paper, an affinity propagation clustering (APC)-based fingerprinting localization system with Gaussian process regression (GPR) is presented for a practical positioning system with the reduced offline workload and low online computation cost. The proposed system collects sparse received signal strength (RSS) data from the deployed Bluetooth low energy beacons and trains them with the Gaussian process model. As the signal estimation component, GPR predicts not only the mean RSS but also the variance, which indicates the uncertainty of the estimation. The predicted RSS and variance can be employed for probabilistic-based fingerprinting localization. As the clustering component, the APC minimizes the searching space of reference points on the testbed. Consequently, it also helps to reduce the localization estimation error and the computational cost of the positioning system. The proposed method is evaluated through real field deployments. Experimental results show that the proposed method can reduce the offline workload and increase localization accuracy with less computational cost. This method outperforms the existing methods owing to RSS prediction using GPR and RSS clustering using APC.

Keywords: affinity propagation clustering; bluetooth low energy; fingerprinting localization; gaussian process regression

1. Introduction

Indoor location-based service (LBS) has been attracting significant interest in recent times owing to an increase in the number of smart devices and technologies being used. Pervasive applications such as behavior recognition, smart medication, and smart building require accurate position information of the users to yield accurate and timely services [1]. Although the global navigation satellite system is widely used for LBS, its signal cannot penetrate buildings. In addition, indoor environments are often complex owing to the presence of obstacles and environment changes, resulting in signal fluctuation or noise. Nevertheless, high localization accuracy (meter level) is often expected for adequate LBS. Some of the consumer technologies adopted in the development of indoor positioning system (IPS) include Bluetooth low energy (BLE), Wi-Fi, ultra-wideband, visible light wave, and geomagnetism [2–6]. The wireless signal measuring principles in IPS are time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS) [7].

Trilateration and fingerprinting localization are used in most IPS studies. Trilateration estimates the position of a tag device by observing its distance from multiple access points (APs). The distance is estimated by employing time-of-arrival between devices and APs or obtaining the RSS from the

neighborhood devices. Here, a few nonlinear equations are solved to estimate the 2-D position of the tag device, where three non-collinear and non-collocated APs are located.

Fingerprinting is based on scene analysis that consists of two phases: offline and online. A radio map consisting of both RSS information from the deployed APs and reference point (RP) locations is built in the offline phase and is preserved in a database. Subsequently, it is compared with online-observed RSS data in the online phase to deduce the tag location. Apart from RSS, channel state information [8] and geo-magnetic field strength [9] are used as the location fingerprint. Fingerprinting is a pervasive method governing IPS owing to its high reliability. However, the offline phase of fingerprinting is time-consuming and labor-intensive. Moreover, the radio map needs frequent updates owing to dynamic and unpredictable changes in the radio environment [10]. Hence, the time and effort required to build the radio map act as a tradeoff for the localization reliability of fingerprinting localization. Furthermore, the fingerprinting localization can be realized with probabilistic and deterministic approaches. The probabilistic approach yields localization information by estimating a probability distribution over the reference points, whereas the deterministic approach implements fingerprinting data comparison algorithms to find the estimated position. The probabilistic method of fingerprinting has better localization accuracy than the deterministic method. However, the computational complexity of the probabilistic approach is higher compared to the deterministic approach [11]. Clustering helps to reduce the region of interest from the whole fingerprint database to a subset of it in the online phase, and thus reduces the computational complexity [12].

Many studies have endeavored to reduce the offline workload of fingerprinting localization. A self-guided robot armed with inertial measurement unit sensors is employed in reference [13] to explore the testbed and obtain the RSS data. Although this approach may reduce the human workload, it is not practically feasible. The study in reference [14] combines the traditional fingerprinting with weighted centroid localization (WCL) to yield acceptable location estimation by employing a fewer number of RPs across the testbed. Here, the WCL estimation obtained with nearby beacons is used as coarse localization to determine the distances to the selected k -nearest RPs; further, the estimated distances and the respective coordinate of k -nearest RPs are utilized to run the next WCL operation. Subsequently, the fine localization result is produced. The WK-NN interpolation algorithm is combined with affinity propagation clustering (APC) to reduce the offline workload in [15]. This method employs APC to find trusted neighbors for Wk-NN interpolation. First, the cluster representative is selected and the physical distance between the unknown RPs and the cluster representative RP is calculated. Second, the unknown RPs are divided into corresponding clusters based on minimum distance to cluster representative where signal strength at each unknown RPs is calculated using a Wk-NN interpolation algorithm. Despite the requirement of manually collecting the RSS to begin the algorithm (RSS clustering with APC), this method can be helpful while updating the fingerprint radio map in a timely manner. Another approach to reduce human effort in acquiring the radio map is to use crowdsourcing, machine learning, and a fusion of similarity-based sequence and dead reckoning [16–18].

In this paper, we present a probabilistic framework for handling sparse training data through APC-based fingerprinting localization. Additionally, the offline workload is decreased by employing the Gaussian process regression (GPR) in the localization to predict the RSS at locations with no prior measurements. The radio map constructed by using limited training data is further divided into clusters or sub-areas by applying APC. The RSS clustering with APC helps to reduce the localization estimation error and computational complexity. BLE beacon, which consumes less energy, is used in this study as the indoor positioning technology.

The remainder of this paper is organized as follows: Section 2 briefly describes some published literature on fingerprinting localization using GPR and APC. Section 3 introduces a general model of GPR and APC. The proposed positioning method is elaborated in Section 4. Experimental results and discussion are presented in Section 5, and the conclusion is presented in Section 6.

2. Related Work

An up-to-date radio map database is required for pragmatic fingerprinting localization applications. Conventionally, it is performed by manually acquiring RSS data at RPs across the testbed. Hence, the offline workload of traditional fingerprinting localization makes the system tedious, time-consuming, and labor-intensive. Whereas in the online phase, the probabilistic approach of fingerprinting positioning has good localization accuracy, but it also has high computational complexity.

The use of regression to predict RSS data with training datasets helps reduce the offline workload of fingerprinting localization. Some representative approaches based on regression or interpolation include polynomial fitting, exponential fitting, and the log model [19]. However, they can predict the mean RSS but not the variance of the estimate. GPR has an advantage of predicting the mean RSS along with the variance, which indicates the uncertainty of the estimation. Hence, GPR is suitable for a probabilistic-based localization method. A GPR-based fingerprinting IPS using indoor Wi-Fi APs is presented in reference [10]. The hyperparameters of the GPR are obtained by using the firefly algorithm to achieve a median localization error of 3 m. It has also been shown that the probabilistic-based localization performs better than the deterministic-based localization. Another example of GPR-based fingerprinting IPS using Wi-Fi is presented in reference [20]. Here, the GPR parameters or the hyperparameters are estimated by using a subspace trust-region method. This work reports that the localization accuracy with a radio map constructed using GPR is higher than that of the Horus fingerprinting method [21]. A GPR plus method using Bluetooth transmitters is proposed in reference [22]. The authors compared their IPS architecture with reference [10] by using the naïve Bayes algorithm to reduce the computation complexity of the IPS.

Clustering helps to minimize the searching space of RPs on the online phase of fingerprinting localization. Some well-known clustering algorithms are K-means, fuzzy c-means, and APC [23–25]. An artificial neural network (ANN)-based fingerprinting employing K-means and fuzzy c-means clustering in wireless sensor networks is presented in reference [26]. Here, after dividing the fingerprinting database into clusters, a separate ANN is trained for each cluster by using only those fingerprints that belong to the cluster. Moreover, a prototype signal strength vector is determined for each cluster during clustering and it is compared with online signal strength measurements for final localization estimation. APC has also been widely used in deterministic (Wk-NN) and probabilistic approaches of fingerprinting positioning [27–30]. All of this literature compares the fingerprint with the cluster heads RSS value in their class-matching methods. In addition to similarity criterion to the RSS of cluster heads, Ref. [27] proposes to use mean of the fingerprints of members of each cluster and reference [28] proposes to use Mahalanobis norm for coarse localization.

Although the GPR and APC are used in many kinds of literature, we find that there is no related literature that combines them to address the real issues of conventional fingerprinting localization like offline workload (labor intensive and time-consuming) and computational complexity. In our study, we use the APC algorithm along with a machine learning approach of fingerprinting localization. The mean RSS and variance are predicted using GPR while using little training data to construct a radio map. RSS clustering is performed on the radio map using APC, i.e., the testbed can be divided into clusters of RPs generated by APC. In the online phase, any exemplar RP is estimated as coarse localization. Subsequently, the RPs grouped under the exemplar RP are used for fine localization. Here, we evaluate two exemplar decision rules depending on the posterior probability and RSS distance as a metric to determine the exemplar RP.

3. Background

3.1. Gaussian Process Regression

A Gaussian process (GP) is a stochastic process and is an extension of multivariate Gaussian to an infinite-sized collection of real-valued variables. The GP defines a distribution over functions from the view of function space [10,31]. Let us consider an observation model as follows:

$$y = f(\mathbf{x}) + v, \quad (1)$$

where v denotes the independent and identically distributed (i.i.d) Gaussian noise with zero mean and variance (σ_v^2), which can be summarized as: $v \sim N(0, \sigma_v^2)$. Here, y and \mathbf{x} represent the observed RSS data and RPs location coordinate (input features), respectively.

The mean function, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x}_m, \mathbf{x}_n)$, for the latent function, f , can be expressed as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \quad (2)$$

$$k(\mathbf{x}_m, \mathbf{x}_n) = \mathbb{E}[(f(\mathbf{x}_m) - m(\mathbf{x}_m))(f(\mathbf{x}_n) - m(\mathbf{x}_n))], \quad (3)$$

where $\mathbb{E}(\cdot)$ represents the expectation operator. For simplification, the mean function can be considered to be zero without loss of generality. The following kernel function is adopted in this study:

$$k(\mathbf{x}_m, \mathbf{x}_n) = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x}_m - \mathbf{x}_n\|}{2l^2}\right), \quad (4)$$

where σ_f^2 is the signal variance and l is the length scale parameter. Here, $\theta = [\sigma_f, l]$ are the GPR parameter or the hyperparameter that needs to be optimized [32]. Note that l characterizes the smoothness of the predicted mean RSS. Using (4), the Gramian matrix, $\tilde{\mathbf{K}}$, is defined as

$$\tilde{\mathbf{K}} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}, \quad (5)$$

where N denotes the total number of RPs across the testbed. Hence, the GP equation is

$$f(\mathbf{x}) \sim gp(m(\mathbf{x}), k(\mathbf{x}_m, \mathbf{x}_n)). \quad (6)$$

As it is assumed in GP modeling that the data can be represented as a sample from a multivariate Gaussian distribution, it can be inferred that

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})_{N \times N} & \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}^*)_{N \times N^*} \\ \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X})_{N^* \times N} & \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X}^*)_{N^* \times N^*} \end{bmatrix}\right), \quad (7)$$

where \mathbf{y} and \mathbf{y}^* are N training and N^* test RSS data, respectively. The block covariance matrix represents the size of that particular matrix for N and N^* . The conditional probability of $\mathbf{y}^*|\mathbf{y}$ can be computed as

$$P(\mathbf{y}^*|\mathbf{y}) \sim N\left(\tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X})\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}, \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X}^*) - \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X})\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})^{-1}\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}^*)\right). \quad (8)$$

Here, the posterior distribution, $P(\mathbf{y}^*|\mathbf{y})$ indicates how likely a prediction \mathbf{y}^* is, given the training data \mathbf{y} . The prediction \mathbf{y}^* is the mean of this distribution $\bar{\mathbf{y}}^* = \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X})\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}$, and the uncertainty of the estimation is $var(f(\mathbf{X})) = \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X}^*) - \tilde{\mathbf{K}}(\mathbf{X}^*, \mathbf{X})\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X})^{-1}\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}^*)$. GPR can be used to build a radio map with $\bar{\mathbf{y}}^*$ and $var(f(\mathbf{X}))$ for locations without prior measurements.

Before estimating the mean RSS and the variance, the GP model needs to be trained by optimizing the unknown hyperparameter vector θ with the training dataset S . The GP hyperparameters can be inferred using procedures like marginal likelihood (ML), cross-validation (CV), and Bayesian optimization [33–35]. Use of the ML approach is optimal and computationally efficient when the data truly follows the GP model. Furthermore, ML can be realized with approaches such as Maximum a Posteriori (MAP) estimator and Minimum Mean Square Error (MMSE) estimator. The marginal log-likelihood function for the Gaussian distributed noise is given by [17,31]:

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T \left[\tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}) + \sigma_v^2 \mathbf{I} \right]^{-1} \mathbf{y} - \frac{1}{2} \log \left| \tilde{\mathbf{K}}(\mathbf{X}, \mathbf{X}) + \sigma_v^2 \mathbf{I} \right| - \frac{\nu}{2} \log 2\pi. \quad (9)$$

The above function is intrinsic to GPR and balances between model complexity and data fit to avoid overly complex models. In this work, we use the limited memory BFGS-B algorithm to solve the optimization problem [36,37].

3.2. Affinity Propagation Clustering

APC divides a set of elements into clusters and selects an exemplar to represent each cluster [23]. In IPS, APC algorithm divides the entire testbed into clusters (or radio map into clusters) based on RSS values at the RPs from the deployed APs. The cluster head is searched first on the online phase (coarse localization) based on a specified similarity metric. It helps to reduce the RP space that reduces the computational complexity of the subsequent fine localization and improve the positioning accuracy. APC begins by assigning each element the same chance to become an exemplar, whereas traditional K-means clustering begins by choosing both the number of output clusters and the corresponding random set of initial exemplars [38]. APC outperforms the K-means clustering owing to the initialization-independent property and better selection of cluster heads. Let the radio map (ψ_{RSS}) consisting of the predicted mean RSS be:

$$\psi_{RSS} = \begin{pmatrix} \psi_{1,1} & \psi_{1,2} & \dots & \psi_{1,N} \\ \psi_{2,1} & \psi_{2,2} & \dots & \psi_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{B,1} & \psi_{B,2} & \dots & \psi_{B,N} \end{pmatrix}, \quad (10)$$

where $\psi_{b,j} = (\bar{y}^*)_{b,j}$ ($b = 1, 2, \dots, B; j = 1, 2, \dots, N$) is the predicted mean RSS of the b^{th} beacon (AP_b) signal at the j^{th} RP (RP_j).

APC uses a pairwise similarity $sim^{CL}(RP_i, RP_j)$ (for $i \neq j$) to describe the fitness of RP_j to be selected as the exemplar with respect to RP_i . Here, the subscript CL indicates that the similarity is evaluated during the clustering step of operation. The pairwise similarity is defined by squared Euclidean distance as follows:

$$sim^{CL}(RP_i, RP_j) = -\|\vec{\psi}_i - \vec{\psi}_j\|^2, \forall i, j \{1, 2, \dots, N\} \quad (11)$$

where $\vec{\psi}_j = [\psi_{1,j}, \psi_{2,j}, \dots, \psi_{B,j}]^T$.

The self-similarity or preference ($Pref$) is defined as

$$Pref = median \left\{ sim^{CL}(RP_i, RP_j) \right\}, \forall i, j \{1, 2, \dots, N\}, i \neq j. \quad (12)$$

Two types of messages, namely responsibility, and availability, are transmitted among the RPs for APC. The responsibility message ($r(RP_i, RP_j)$) contains the information about the clustering head

whereas the availability message ($a(RP_i, RP_j)$) provides the attachment relations between the RPs and clusters. Both the messages are iteratively updated according to the following relationships:

$$r(RP_i, RP_j) = sim^{CL}(RP_i, RP_j) - \left(\max_{j' \neq j} \right) \left\{ a(RP_i, RP_{j'}) + sim^{CL}(RP_i, RP_{j'}) \right\} \quad (13)$$

$$a(RP_i, RP_j) = \min \left\{ 0, r(RP_j, RP_i) + \sum_{i' \neq i, j} \max \{ 0, r(RP_{i'}, RP_j) \} \right\}. \quad (14)$$

A damping factor (γ) [0.5, 1) is introduced to avoid the possible ringing oscillations while updating the messages (13) and (14). The resulting values of responsibility and availability are:

$$\begin{aligned} r_t(RP_i, RP_j) &= \gamma * r_{t-1}(RP_i, RP_j) + (1 - \gamma) * r_t(RP_i, RP_j), \\ a_t(RP_i, RP_j) &= \gamma * a_{t-1}(RP_i, RP_j) + (1 - \gamma) * a_t(RP_i, RP_j) \end{aligned} \quad (15)$$

In (15), $r_t(RP_i, RP_j)$ and $a_t(RP_i, RP_j)$ corresponds to the value of responsibility and availability of the current iteration, respectively. Similarly, $r_{t-1}(RP_i, RP_j)$ and $a_{t-1}(RP_i, RP_j)$ is the value of responsibility and value of availability of the last iteration. Here, greater the sum of $r_t(RP_i, RP_j)$ and $a_t(RP_i, RP_j)$, the greater the probability of RP_j be the cluster head for the RP_i . The APC connects all the points in the large space and makes each node (RP in our case) a potential exemplar. APC updates the responsibility and availability message iteratively until it converges and yields clusters and its members finally.

4. GPR/APC-Based Fingerprinting Localization

The proposed APC-based fingerprinting localization method adopting GPR has two phases of operations, namely offline and online. In the offline phase, we collect the training data from a few sparse RPs that are a subset of the total RPs on the testbed. The testbed is divided into N uniform grids and each grid center represents an RP. Hence, the testbed has N RPs and the RSS data of B beacons (B APs) are acquired at r^{th} RP ($r \in \{1, 2, \dots, N\}$) q times, that is

$$AP_r = \{ap_{r,1}, ap_{r,2}, \dots, ap_{r,B}\},$$

where

$$ap_{r,i} = Avg(ap_{r,i}^1, ap_{r,i}^2, \dots, ap_{r,i}^q).$$

The observed online RSS data at the unknown location inside the testbed can be represented as follows:

$$AP_* = \{ap_{*,1}, ap_{*,2}, \dots, ap_{*,B}\}.$$

The framework of the proposed localization system is illustrated in Figure 1. First, training data are collected from sparse RPs (S) that are a subset to N ($S < N$). Here, for the stability of the randomly fluctuating RSS data, a time average of RSS data is considered. Subsequently, GPR is employed to build the posterior mean RSS and variance at each of the RPs on the testbed. Later on, RSS clustering of the predicted RSS is performed using APC. The predicted RSS mean and variance from the GPR and the RSS clustering with APC are stored in the database for future reference.

Each RP is assigned a probability computed as the probability of RP_m , given the online RSS observation vector AP_* . The posterior probability is given as

$$Pr(RP_m | AP_*) = \frac{Pr(AP_* | RP_m) Pr(RP_m)}{\sum_{n=1}^N Pr(AP_* | RP_n) Pr(RP_n)}, \quad (16)$$

where $Pr(AP_*|RP_m)$ is the likelihood defined by the following relationship:

$$Pr(AP_*|RP_m) = \prod_{b=1}^B \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{|AP_b - \mu_b|^2}{2\sigma_b^2}\right). \quad (17)$$

Here, μ_b and σ_b^2 are the predicted mean RSS and its variance of beacon b signal at the RP location, respectively. We define \mathcal{C} and N' as the set of cluster heads and the number of RPs grouped under a cluster head, respectively. Note that N' may vary with the cluster, where $N' < N$. First, coarse localization is performed to determine the exemplar RP. In our proposed APC localization system, the exemplar RP (RP_e) can be determined in either of the following ways:

$$RP_e = \operatorname{argmax}_m [Pr(RP_m|AP_*)], m \in \mathcal{C} \quad (18)$$

$$RP_e = \operatorname{argmin}_{m \in \mathcal{C}} \|\vec{\psi}_r - \vec{\psi}_m\|^2, \vec{\psi}_r = [\psi_{1,r}, \psi_{2,r}, \dots, \psi_{B,r}]^T \quad (19)$$

In reference (18), given the online RSS, any RP in \mathcal{C} that has the largest posterior probability is determined as the cluster head, whereas in (19), the RP in \mathcal{C} with the least RSS distance to the online-acquired RSS data is selected as the cluster head.

Second, the final localization estimation is obtained as a weighted sum of N' RPs with their respective probabilities.

$$\hat{E} = \sum_{i=1}^{N'} Pr(RP_i|AP_*) \times RP_i \quad (20)$$

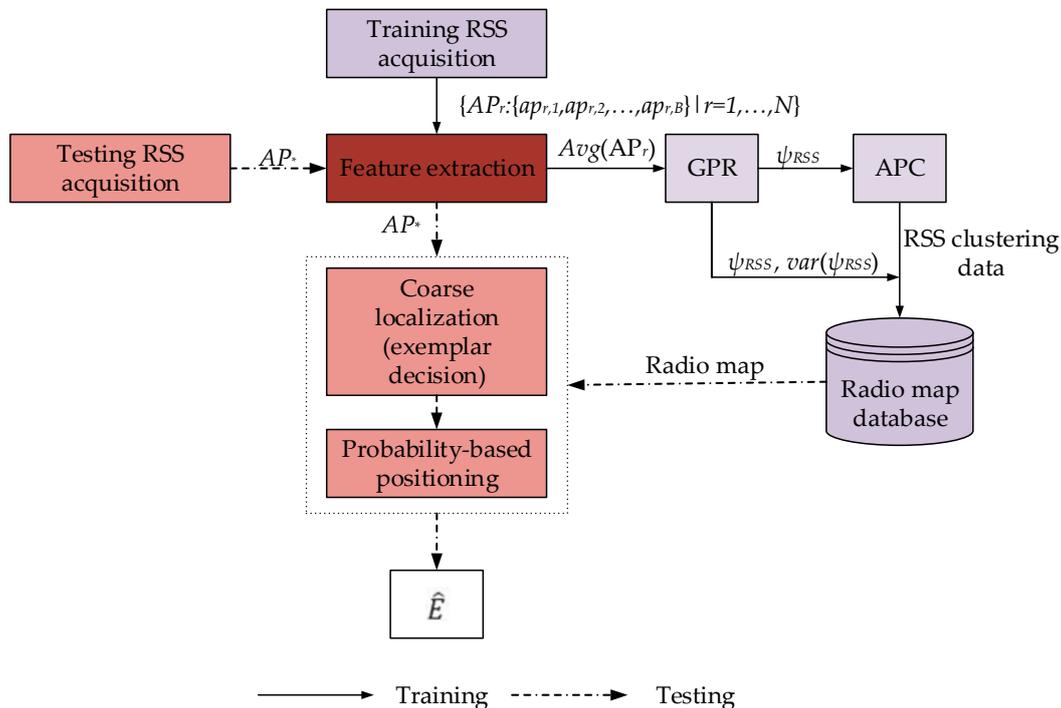


Figure 1. Framework of the proposed APC-based fingerprinting localization adopting GPR.

5. Experimental Results

The experimental results are presented in this section. The offline measurements were obtained on the third floor of the IT building, Chosun University, South Korea. The testbed considered is an academic environment with frequent movement of people along the hallways. Six BLE beacons were deployed to cover the whole range of the testbed area. All the beacons were set to have an

advertisement interval of 300 ms and beacon transmission power of +4 dBm. The testbed area with the deployed beacons is shown in Figure 2. The testbed area was divided into uniform grids with side measurements of 0.9 m to form 203 measurement locations. An iOS application on iPhone-6S was used to record the RSS data from six beacons. The training data were measured at 11 randomly selected sparse locations ($S = 11$) where 10 time samples of RSS in four different directions (total of 40 samples) were measured at each measurement location. For the locations with undetected beacons or null reading from the beacons, a default RSS value considered -95 dBm in our work was used as invalid data [10,27].

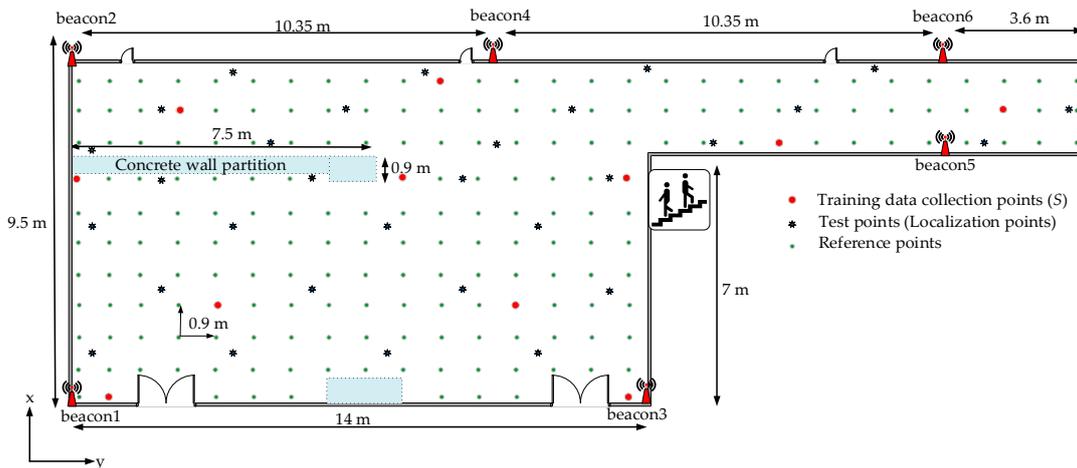


Figure 2. Graphical representation of the testbed (A hallway at the IT building in Chosun University, Korea).

Figure 3a presents the predicted mean RSS as well as the manually measured RSS at the testbed. It is observed that the predicted RSS is almost similar to the real measured data; in particular, the average difference between the measured and predicted RSS data is 6.50 dBm. Figure 3b presents the corresponding standard deviation of the predicted mean RSS. Figure 4a shows the surface plot of the predicted RSS of *beacon₄* on the testbed. The surface in red depicts the region on the testbed close to the *beacon₄* location in Figure 2. Similarly, Figure 4b shows the surface plot of its standard deviation. Here, the surface in blue indicates that measurement data are available in the surroundings, whereas the red/yellow surface indicates the lack of measurement data around it.

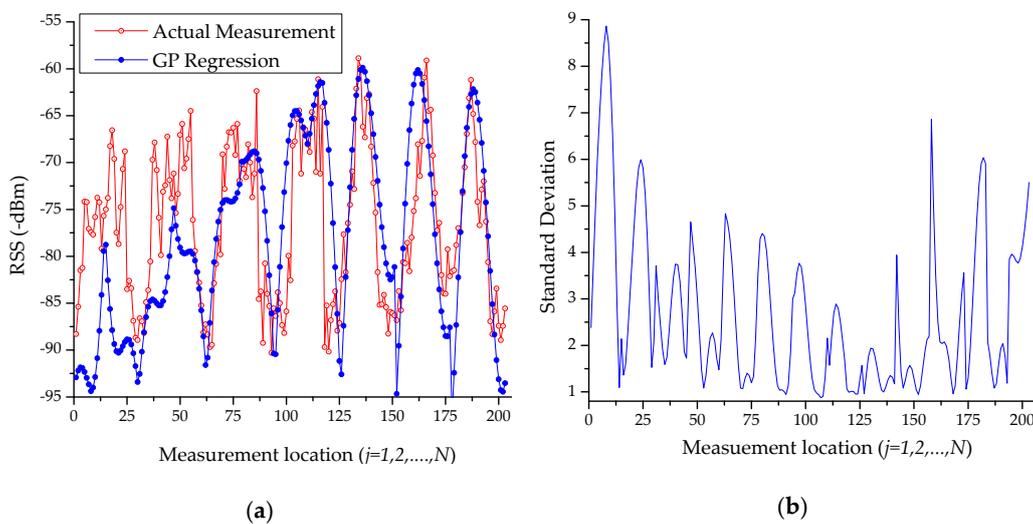


Figure 3. Estimation of *beacon₄* signal with GPR (a) predicted RSS, (b) the corresponding standard deviation at the measurement locations.

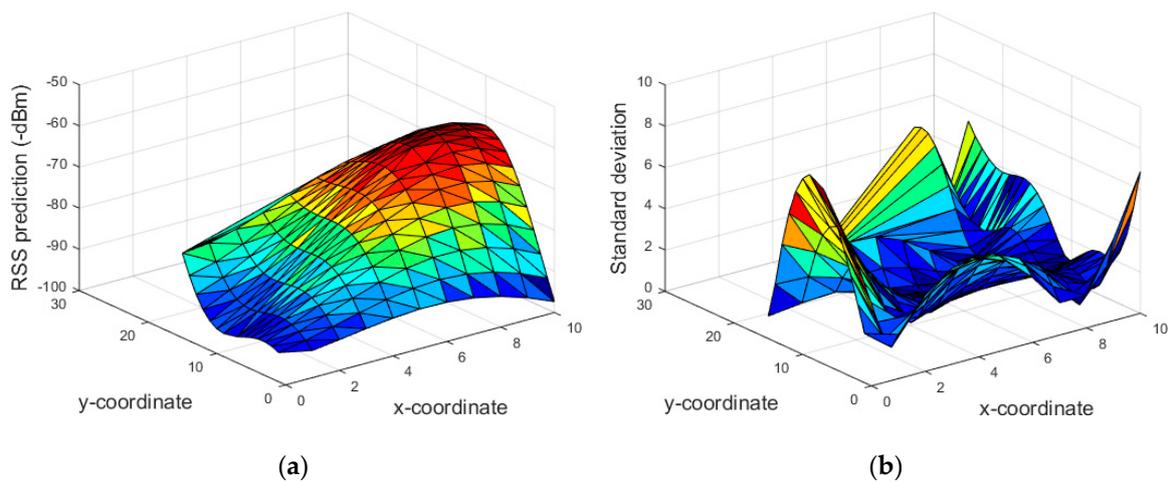


Figure 4. Surface plot of $beacon_4$ signal (a) predicted RSS, (b) corresponding standard deviation.

After the GPR operation, the predicted mean RSS was fed to the APC for RSS clustering. While updating the responsibility and availability messages in APC, we evaluated the APC operation by varying the damping factor (γ) [0.5, 0.9] introduced in (15). The obtained clustering result is presented in Figure 5.

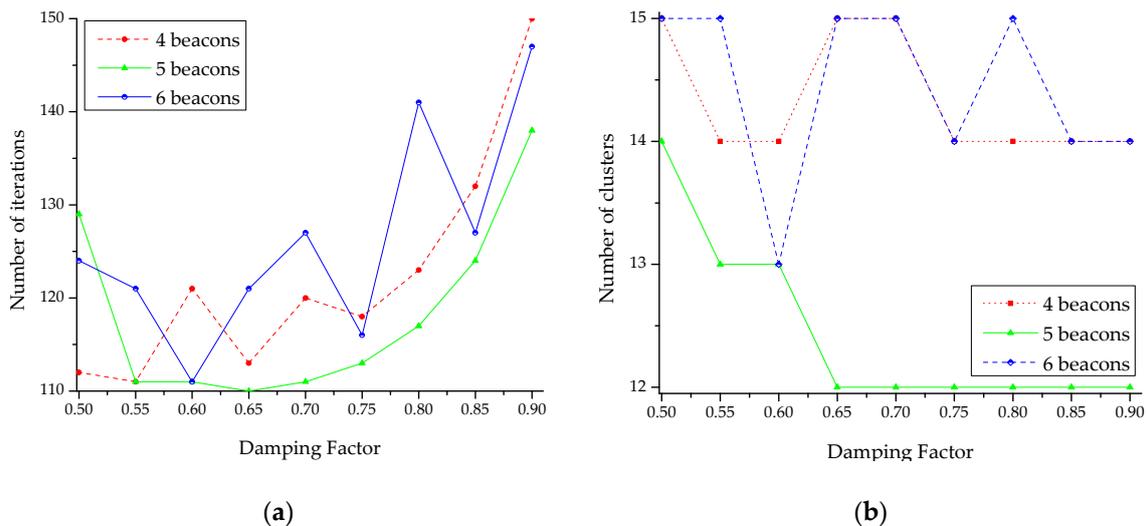


Figure 5. Effect of damping factor (γ) in APC (a) the number of clustering operation iterations, (b) the number of clusters with respect to the damping factor.

We evaluated the localization methods under the deployment of different numbers of beacons on the testbed, i.e., 4, 5, and 6 beacons. For the deployment scenarios of 4 and 5 beacons in the testbed, $\{beacon_4, beacon_5\}$ and $\{beacon_5\}$ were removed from the testbed, respectively (see Figure 2). As shown in Figure 5, the damping factor with $\gamma = 0.6$ yields the APC result with the least number of iterations or the least computational cost. We used this value for the localization estimation in our work. The cumulative probability of the localization estimation error in the testbed area is shown in Figure 6. Both the decision rules for exemplar estimation, (18) and (19), presented in Section 4 are considered. At each point of 30 uniformly distributed test locations on the testbed, 15 samples of localization estimation were recorded. We compared our method with a typical GPR-based probabilistic method [20] and Horus method [39]. The Horus method estimates the position of the tag device at the location on the testbed with the largest posterior probability through the Bayesian interference [40]. Moreover, the Horus method employs a clustering module where any cluster is a set of RPs sharing a

common set of access points (Wi-Fi). In this study, we applied APC to the manually measured RSS data in the Horus method, where (19) determined the cluster head. In Figure 6, only the Horus method uses the manually constructed radio map data whereas the other methods use the predicted RSS data. Similarly, all the methods under study except the typical GPR employ APC for localization estimation.

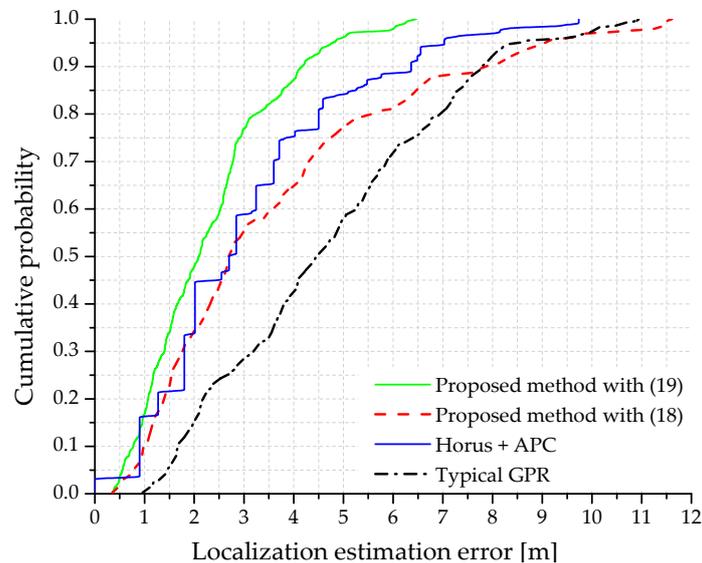


Figure 6. Cumulative probability of the localization estimation error.

As shown in Figure 6, in the proposed method utilizing the RSS distance as the exemplar decision rule, there is a probability of 48% that the localization error is below 2 m. Similarly, in the typical GPR and Horus methods, the corresponding probabilities are 15% and 34%, respectively. The proposed method utilizing the RSS distance shown in (19) as the exemplar decision rule performed better than the method using the largest posterior probability in (18). We recommend the use of (19) for the proposed method, whereas (18) can be used when (19) is not available. Note that approximately 7.5% of the localization estimation error is above 4.5 m in the proposed method owing to the following reasons:

- Faulty exemplar estimation in the online phase: Owing to the fluctuations in the received RSS, the system may lead to the estimation of an RP as an exemplar that is physically far from the real position of the tag device. As the RPs grouped under the cluster head or exemplar are responsible for the final localization estimation, the wrong estimation of exemplar eventually adds some localization estimation error.
- Outliers in the offline phase: The APC algorithm might lead to an RP that belongs to a cluster, but is physically far from the cluster head [30]. This problem can be solved by considering the benefit of the known position of each RP. Here, each outlier RP can be forced to join the cluster characterized by the exemplar at the minimum distance from the outlier itself. This process helps reduce the localization error during the localization operation.

The localization estimation error of the typical GPR method is larger than that of the Horus (with APC) method. However, recording the RSS distribution at RPs in the Horus method is time-consuming. The localization accuracy of the typical GPR method can be improved by increasing the number of training data ($S > 11$ in this work) [10]. However, increasing the size of training data would also increase the offline workload. The average localization estimation error of various positioning methods with different numbers of beacons deployed on the testbed is presented in Figure 7.

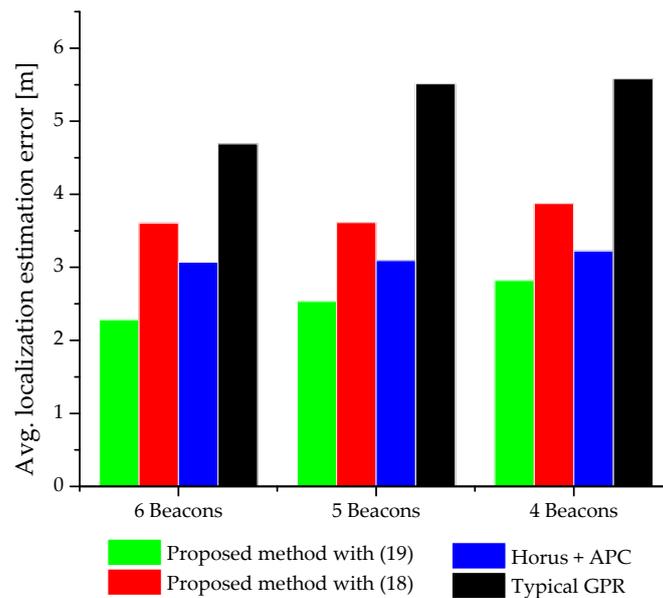


Figure 7. Average localization error from different positioning methods at different beacon density scenarios.

Figure 7 shows the impact of different beacon densities on the positioning result. As expected, decreasing the beacon density on the testbed has an inverse effect on the positioning result. The proposed method addresses the real issues of conventional fingerprinting localization to realize a practical IPS. The time consuming and labor-intensive problems of offline phase in conventional fingerprinting are minimized by using GPR to predict the mean RSS and standard deviation at RPs with no prior measurements. For example, we train our GP model using training data from 11 measurement locations to populate the radio map for the entire testbed (203 RPs). Similarly, various published literature has shown that the computational complexity of the probabilistic approaches of fingerprinting localization is high and the localization accuracy of the deterministic approaches is low. The proposed method solves the computational complexity problem of the probabilistic method using APC.

Computational Complexity

Finally, we discuss the complexity of our proposed method and the conventional probabilistic fingerprinting method. For the conventional probabilistic fingerprinting method, its complexity is $O(NB)$, where N is the total number of fingerprints to be compared and B is the total number of APs (beacons) in a given indoor environment, respectively. Here, the computational complexity grows with N . The coarse localization stage (cluster head determination stage of online phase) of the proposed method reduces the area of interest from N RPs into N' RPs (N' is number of RPs grouped under a cluster head by APC and $N' \ll N$). Hence, the total number of fingerprints to be compared for fine localization is N' , the complexity of the proposed method becomes $O(N'B)$. Therefore, it can be seen that APC minimizes the searching space of RPs and reduces the computational complexity.

6. Conclusions

Despite the reliability of fingerprinting localization in terms of localization accuracy, the main issue with this method is the offline workload that is both tedious and time-consuming. We have employed GPR to overcome the labor-intensive task of the traditional fingerprinting method. In addition, we have performed clustering on the predicted RSS data to divide the whole testbed into many subsections or clusters, which helps to minimize the computational cost of online positioning. The choice of GPR for regression is due to its property of yielding both the mean RSS and the variance, which can be used

for probabilistic fingerprinting. APC is selected for clustering owing to its initialization-independent property and better selection of cluster heads compared with K-means clustering.

We used BLE beacons to implement the proposed method considering that they are embedded in many current smartphones and consume less energy than other technologies. We evaluated two different cluster head estimation approaches in the online phase. The experimental result on our testbed shows that the RSS-distance-based exemplar decision method performs better for estimating the cluster head. The positioning result of the proposed method is better than that of the existing methods.

Author Contributions: S.S. and J.-Y.P. conceived the idea and research metrology. S.S. designed the proposed localization algorithm and performed the experiments. J.-Y.P. contributed to the conception of the study and analysis by directing and supervising the research.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Shi, K.; Ma, Z.; Zhang, R.; Hu, W.; Chen, H. Support vector regression based indoor location in IEEE 802.11 environments. *Mob. Inf. Syst.* **2015**, *2015*, 295652. [[CrossRef](#)]
- He, S.; Chan, S.-H.G. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 466–490. [[CrossRef](#)]
- Subedi, S.; Kwon, G.-R.; Shin, S.; Hwang, S.; Pyun, J.-Y. Beacon based indoor positioning system using weighted centroid localization approach. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 1016–1019.
- Alarifi, A.; Al-Salman, A.; Alsaleh, M.; Alnafessah, A.; Al-Hadhrami, S.; Al-Ammar, M.A.; Al-Khalifa, H.S. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors* **2016**, *16*, 707. [[CrossRef](#)] [[PubMed](#)]
- Jiao, Z.; Zhang, B.; Liu, M.; Li, C. Visible light communication based indoor positioning techniques. *IEEE Netw.* **2017**, *31*. [[CrossRef](#)]
- Haverinen, J.; Kemppainen, A. Global indoor self-localization based on the ambient magnetic field. *Rob. Auton. Syst.* **2009**, *57*, 1028–1035. [[CrossRef](#)]
- Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Liu, R.; Yuen, C.; Raulefs, R.; Aboutanios, E. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1327–1346. [[CrossRef](#)]
- Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 763–776. [[CrossRef](#)]
- Carrillo, D.; Moreno, V.; Úbeda, B.; Skarmeta, A.F. Magicfinger: 3d magnetic fingerprints for indoor location. *Sensors* **2015**, *15*, 17168–17194. [[CrossRef](#)]
- Yiu, S.; Yang, K. Gaussian process assisted fingerprinting localization. *IEEE Internet Things J.* **2016**, *3*, 683–690. [[CrossRef](#)]
- Khalajmehrabadi, A.; Gatsis, N.; Akopian, D. Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges. *arXiv*, 2016; arXiv:1610.05424.
- Feng, C.; Au, W.S.A.; Valaee, S.; Tan, Z. Received-signal-strength-based indoor positioning using compressive sensing. *IEEE Trans. Mob. Comput.* **2012**, *11*, 1983–1993. [[CrossRef](#)]
- Yeh, L.-W.; Hsu, M.-H.; Huang, H.-Y.; Tseng, Y.-C. Design and implementation of a self-guided indoor robot based on a two-tier localization architecture. *Pervasive Mob. Comput.* **2012**, *8*, 271–281. [[CrossRef](#)]
- Subedi, S.; Pyun, J.-Y. Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons. *J. Sens.* **2017**, *2017*. [[CrossRef](#)]
- Hu, J.; Liu, H.; Liu, D.; Yan, Z.; Xu, K. Reducing Wi-Fi Fingerprint Collection Based on Affinity Propagation Clustering and WKNN Interpolation Algorithm. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 11–13 October 2018; pp. 2463–2468.

16. Rai, A.; Chintalapudi, K.K.; Padmanabhan, V.N.; Sen, R. Zee: Zero-effort crowdsourcing for indoor localization. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 293–304.
17. Richter, P.; Toledano-Ayala, M. Revisiting Gaussian process regression modeling for localization in wireless sensor networks. *Sensors* **2015**, *15*, 22587–22615. [[CrossRef](#)] [[PubMed](#)]
18. Liu, R.; Yuen, C.; Do, T.; Tan, U. Fusing Similarity-Based Sequence and Dead Reckoning for Indoor Positioning Without Training. *IEEE Sens. J.* **2017**, *17*, 4197–4207. [[CrossRef](#)]
19. Yang, J.; Chen, Y. Indoor localization using improved rss-based lateration methods. In Proceedings of the Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
20. Kumar, S.; Hegde, R.M.; Trigoni, N. Gaussian process regression for fingerprinting based localization. *Ad Hoc Netw.* **2016**, *51*, 1–10. [[CrossRef](#)]
21. Youssef, M.; Agrawala, A. The Horus location determination system. *Wireless Netw.* **2008**, *14*, 357–374. [[CrossRef](#)]
22. Liu, K.; Meng, Z.; Own, C.-M. Gaussian Process Regression Plus Method for Localization Reliability Improvement. *Sensors* **2016**, *16*, 1193. [[CrossRef](#)]
23. Mézard, M. Passing messages between disciplines. *Science* **2003**, *301*, 1685–1686. [[CrossRef](#)] [[PubMed](#)]
24. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recogn. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
25. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [[CrossRef](#)]
26. Bernas, M.; Płaczek, B. Fully Connected Neural Networks Ensemble with Signal Strength Clustering for Indoor Localization in Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 403242. [[CrossRef](#)]
27. Luo, J.; Fu, L. A Smartphone Indoor Localization Algorithm Based on WLAN Location Fingerprinting with Feature Extraction and Clustering. *Sensors* **2017**, *17*, 1339.
28. Karegar, P.A. Wireless fingerprinting indoor positioning using affinity propagation clustering methods. *Wireless Netw.* **2018**, *24*, 2825–2833. [[CrossRef](#)]
29. Tian, Z.; Tang, X.; Zhou, M.; Tan, Z. Fingerprint indoor positioning algorithm based on affinity propagation clustering. *EURASIP J. Wireless Commun. Netw.* **2013**, *2013*, 272. [[CrossRef](#)]
30. Caso, G.; De Nardis, L.; Di Benedetto, M.-G. A mixed approach to similarity metric selection in affinity propagation-based WiFi fingerprinting indoor positioning. *Sensors* **2015**, *15*, 27692–27720. [[CrossRef](#)] [[PubMed](#)]
31. Rasmussen, C.E. *CKI Williams Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
32. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [[CrossRef](#)]
33. Martino, L.; Laparra, V.; Camps-Valls, G. Probabilistic cross-validation estimators for Gaussian process regression. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017; pp. 823–827.
34. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mac. Learn. Res.* **2012**, *13*, 281–305.
35. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 2951–2959.
36. Byrd, R.H.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [[CrossRef](#)]
37. Zhu, C.; Byrd, R.H.; Lu, P.; Nocedal, J. *L_BFGS_B Fortran Subroutines for Large Scale Bound Constrained Optimization, L_BFGS_B_FORTRAN Sub Routines for Large Scale Bound Constrained Optimization*; Tech. Report; Department of Electrical Engineering and Computer Science, Northwestern University: Evanston, IL, USA, 1995.
38. Gokcay, E.; Principe, J.C. Information theoretic clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 158–171. [[CrossRef](#)]

39. Youssef, M.; Agrawala, A. The Horus WLAN location determination system. In Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, WA, USA, 6–8 June 2005; pp. 205–218.
40. Zang, H.; Baccelli, F.; Bolot, J. Bayesian inference for localization in cellular networks. In Proceedings of the 2010 IEEE INFOCOM Conference, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.



© 2018 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/>).