An Efficient Indoor Positioning Method Based on Wi-Fi RSS Fingerprint and Classification Algorithm

Balaji Ezhumalai 1, Moonbae Song 2 and Kwangjin Park 1,*

1 Department of Information and Communication Engineering, Wonkwang University, Iksan 570-749, Korea; balajiezhumalai91@gmail.com
2 Samsung Electronics, Suwon 497-001, Korea; moonbae.song@samsung.com
* Correspondence: kjpark@wku.ac.kr; Tel.: +82-63-850-6347

Abstract: Wi-Fi received signal strength (RSS) fingerprint-based indoor positioning has been widely used because of its low cost and universality advantages. However, the Wi-Fi RSS is greatly affected by multipath interference in indoor environments, which can cause significant errors in RSS observations. Many methods have been proposed to overcome this issue, including the average method and the error handling method, but these existing methods do not consider the ever-changing dynamics of RSS in indoor environments. In addition, traditional RSS-based clustering algorithms have been proposed in the literature, but they make clusters without considering the nonlinear similarity between reference points (RPs) and the signal distribution in ever-changing indoor environments. Therefore, to improve the positioning accuracy, this paper presents an improved RSS measurement technique (IRSSMT) to minimize the error of RSS observation by using the number of selected RSS and its median values, and the strongest access point (SAP) information-based clustering technique, which groups the RPs using their SAP similarity. The performance of this proposed method is tested by experiments conducted in two different experimental environments. The results reveal that our proposed method can greatly outperform the existing algorithms and improve the positioning accuracy by 89.06% and 67.48%, respectively.

Keywords: indoor positioning; received signal strength (RSS); Wi-Fi fingerprint; SAP similarity; fingerprint clustering; RSS extraction

1. Introduction

The rapid development of wireless technology leads people to pay more attention to location-based services (LBS). In recent years, indoor LBS has been widely used in airports, subways, shopping malls, and many other indoor environments for location identification, indoor navigation for the users, and the positioning demands of the different groups of people. Therefore, it is important to build an accurate, reliable, and real-time indoor positioning system to effectively satisfy the public demand for indoor positioning. With the rapid development of smartphones, more sensors and intelligent terminals have become excellent tools for indoor positioning [1–3].

Nowadays, the global navigation satellite system (GNSS) is well matured and is commonly used in outdoor positioning technology. LBS requirements in outdoor environments were well addressed and positioned in GNSS [4]. However, the GNSS is not applicable for indoor positioning systems due to its signal becoming weaker after penetrating through indoor environments; hence, it fails to produce reliable position information [5]. Different wireless technologies such as Bluetooth (BLE beacon) [6], radio frequency identification (RFID) [7], ultra-wideband (UWB) [8], geomagnetism [9], visible light [10], and Wi-Fi [11] were used in indoor positioning systems. Among the aforementioned technologies, Wi-Fi is the most effective technology used in indoor positioning research areas due to its low power consumption, high precision, and low cost [3]. The universality and common imple-
Time of arrival (TOA) [12], time difference of arrival (TDOA) [13], angle of arrival (AOA) [12], and the fingerprinting method [14] are the different methods used in Wi-Fi-based indoor positioning systems. TOA, TDOA, and AOA require an angle of information or distance between two points. Even though their calculations are simple, they mainly rely on line-of-sight propagation (LOSP) between the user’s position and the Wi-Fi access points (APs) [15]. Therefore, the fingerprinting method is widely used in indoor positioning due to its low power consumption, high accuracy, low cost, and the fact that it does not require the location of the Wi-Fi APs and the LOSP [16]. In the Wi-Fi fingerprinting method, the signal strength received from the surrounding Wi-Fi APs is used to estimate the user’s position. The features (fingerprints) of a scene are collected and further estimate the user’s location by matching the online measurement of the collected data [17]. It is also known as the RSS-based fingerprinting method, and it can be implemented in any indoor environment where Wi-Fi devices are deployed. The RSS-based fingerprint method has two phases: offline phase (creating radio map) and online phase (estimating user’s location). In the offline phase, the fingerprint database was created by performing a site survey in the indoor area. In the online phase, the RSS is observed from the user’s unknown location and compared with the RSS stored in the fingerprint database through a localization algorithm unit, which estimates the user’s current location.

The fingerprint positioning methods used to estimate the user location in the online phase can be divided into probabilistic and deterministic methods. In the probabilistic method, it is necessary to know the probability distribution of all deducted APs at each RP stored in the fingerprint database. In contrast, the deterministic method does not need to know the statistical probability model of the Wi-Fi RSS. The advantage of this method is generally simple to implement and useful for real-time applications. In the deterministic method, the nearest neighbor-based localization algorithms are most frequently used; there are nearest neighbor (NN), $k$-nearest neighbor ($k$NN), and weighted $k$ nearest neighbor ($W_k$NN) algorithms. The $W_k$NN is the improved classification algorithm of NN and $k$NN, and it is the most popular deterministic localization algorithm used in Wi-Fi RSS fingerprint-based indoor positioning [1,3].

At present, the main research directions for Wi-Fi RSS fingerprint-based indoor positioning are improving the positioning accuracy and reducing the computational complexity. The accuracy of the RSS fingerprint method is far from adequate due to obstacles in the indoor environment, such as fading and shadowing. The online positioning accuracy of Wi-Fi RSS fingerprint-based indoor positioning is fully dependent on the RSS vectors of each RP stored in the offline fingerprint database, so the RSS extraction is really important for constructing a strong and error-free offline fingerprint database. The average value of the collected RSS samples is generally taken as its RP fingerprint value, and it will be stored in the fingerprint database. In [18], an improved location fingerprinting method was proposed using a mean smoothing algorithm to eliminate the gross error in the offline fingerprint database construction. In [19], the average value of a number of selected maximum RSS values is taken as its RP fingerprint value. To avoid the presence of noisy measurements, the data selection method with a closed-form lease square (LS) approach is proposed in [20] to disregard the bad measurements. The localization accuracy can be increased to some extent in RSS-based positioning using particle filtering [21]. However, the average value of RSS used in the above RSS extraction methods cannot accurately reflect the dynamic behavior of Wi-Fi signals. An error handling method for fingerprint collection is proposed in [22], but the RSS extraction process is very laborious and time-consuming compared to other methods.

The large space of indoor environments such as airports and subways has larger fingerprint databases; hence, finding the few nearest neighbors over a large fingerprint database causes high computational overhead that will affect the algorithm’s efficiency. To overcome this issue, clustering-based fingerprinting methods are proposed; instead of
searching all the RPs, a particular cluster (which has a group of RPs) will be searched to find the user’s location using the clustering technique. The researchers have proposed different clustering techniques, such as $k$-means [23,24], Bisecting $k$-means [25], smallest enclosing circle [26], and affinity propagation [27,28]. However, the clustering criteria used in the above-mentioned clustering techniques do not consider the nonlinear similarity between RPs, which means the clustering result cannot perfectly reflect the position distribution of RPs. Since the $k$-means method is an unsupervised machine learning algorithm, the data are unlabeled, and the algorithm learns by itself using the dataset. The disadvantage of this method is that it randomly places the centroids to make clusters, and we never know how this algorithm sorted the data. The output obtained may not be what the user was expecting due to a data interpretation mismatch. When $k$ is too large, the relatively relevant points might be divided into different clusters, and when $k$ is too small, the less relevant points are much more likely to be divided into the same cluster. Cluster matching in an online phase will become inaccurate due to the inconsistency of the clustering criteria. A location-based clustering is proposed in [29]. Since the signal is not distributed equally due to the obstacles in an indoor environment, location-based clustering will lead to positioning errors. The combination of position label assisted clustering and signal weighted Euclidean distance is proposed in [30], which outperforms the traditional $Wk$ NN with Euclidean distance and a $k$-means clustering algorithm; however, it is only effective for positioning with the line-of-sight APs, and it is not suitable for positioning with non-line of sight APs.

On the other hand, researchers have focused on automatic fingerprint database construction methods such as interpolation and crowdsourcing [31]. A network fingerprint approach is proposed in [32] to auto construct the fingerprint database through Wi-Fi APs. The crowdsource-based adaptive area localization to enable area classification for continuously generated data is proposed in [33], which dynamically subdivides the floor plan into areas based on the quality and features of available training data. The authors of [34] built the Wi-Fi-based indoor navigation application that uses an interpolation method to generate artificial Wi-Fi fingerprints, but the interpolation technique hardly outperformed the accuracy achieved using manual site surveys. The automatic site survey may cause inaccuracy in the fingerprint database, so in this research, our fingerprint collection work is done manually for better demonstration.

The ever-changing dynamics of RSS in indoor environments impact the average value of missing APs, and the nonlinear similarity between RPs of cluster construction was not considered in the above methods. This paper proposes an IRSSMT RSS extraction method and SAP information-based clustering algorithm to tackle the above-mentioned problems of Wi-Fi fingerprint-based indoor positioning. The proposed IRSSMT helps make the RSS smoother and construct an effective fingerprint database, and the proposed SAP information-based clustering algorithm groups the RPs accurately according to its SAP similarity criterion. First, the Wi-Fi signals are collected and stored in RPs using IRSSMT, then the RPs are grouped as clusters according to their SAP similarity in the offline phase. Finally, the $Wk$ NN algorithm is employed in the online phase to estimate the user’s position.

The rest of this paper is organized as follows: In Section 2, an overview of the proposed method is presented, and the proposed algorithms are described in detail. Section 3 consists of the experimental results and the performance evaluation of the proposed method. In Section 4, we present a discussion of the results. Finally, the conclusion and future works are discussed in Section 5.

2. The Proposed Wi-Fi Fingerprint Method

In the following section, we introduce our approach to the efficient indoor positioning method. We describe the overview of the proposed Wi-Fi RSS fingerprint method. Subsequently, we introduce the proposed concept of RSS preprocessing using IRSSMT and SAP information-based clustering. Finally, we describe the online localization algorithm used to select delegate clusters and find user locations.
2.1. Overview of the Proposed Wi-Fi RSS Fingerprint Positioning Method

The overall architecture of the proposed Wi-Fi RSS fingerprint method is shown in Figure 1. Since the proposed method is based on the traditional fingerprint algorithm, it also consists of two phases: the offline phase and the online phase. The traditional preprocessing methods only use the mean value of the RSS samples, and it does not reflect the ever-changing dynamics of RSS and the impact of the average value of its missing APs stored in the fingerprint database; to deal with this problem, we propose an IRSSMT. In the offline phase, we first collect the \( N \) number of original RSS observation samples at each RP. Then, the collected RSS observations are preprocessed using the median value of the RSS samples to make the RSS smoother and reduce the influence of shadowing and fading. The preprocessed RSS values are then stored in the fingerprint database. The traditional clustering methods cannot position the distribution of RPs; to overcome this problem, we propose an SAP information-based clustering algorithm. We first find the SAP information of RPs, and then that SAP information is assigned as the label of that RPs. Then, using the RP labels as the primary information, the RPs are partitioned into several clusters. This process ensures that the SAP information-based clustering results are consistent with the position distribution of RPs. In the online phase, one particular cluster is chosen as the delegate using SAP cluster matching. The user location is then calculated using the \( k \) NN algorithm. The experiment results indicate that the proposed method can greatly improve the positioning accuracy.

![Figure 1. The overall structure of the proposed algorithm.](image)

2.2. Received Signal Strength Preprocessing and Fingerprint Database Construction

As mentioned earlier, due to the obstacles in indoor environments, Wi-Fi signals can be affected by reflection, diffraction, scattering, and absorption. Therefore, preprocessing the RSS is necessary in order to reduce these influences and make the RSS smoother. The flow chart of the IRSSMT is shown in Figure 2.
In this new and improved method, \( N \) RSS observations were collected at a 3 s sampling rate at each RP. Any portable APs such as personal Wi-Fi hotspot (WiBro) and temporary Wi-Fi devices placed inside the room for personal use will affect the fingerprint database; to overcome this problem, the portable APs are filtered. Then the RSS samples are sorted in descending order. A weak Wi-Fi signal received from distant APs may affect the positioning accuracy by presenting less stability and reliability, this should be considered during data collection, so the empirical RSS threshold is set to \( t \) for rejection of APs with ineffective RSS values. In this research, \(-90\) dBm (\( t = -90 \) dBm) is set as an empirical threshold value; it is suitable for our scenario, and it may change in other scenarios with different hardware and AP positions. The existing methods, such as the average method, error handling method, and particle filters, may not be able to handle the complex Wi-Fi RSS measurement noise in dynamic and ever-changing indoor environments [19]. For example, according to the distance between RP and AP in the environment, some APs may not be heard in some RPs due to fading and shadowing. In such cases, the missing APs were filled with \(-99\) dBm, and it is considered as the maximum RSS received at the missing APs. These missing AP’s maximum RSS values make a huge difference in the mean value of the collected samples. Therefore, the mean value of the collected RSS and existing methods may not reflect the dynamic behavior of the RSS and the impact of the missing APs. In this case, the median value of the processed RSS samples is a more appropriate representative of RP (see Section 3.1 for a thorough explanation); so, the median value of the original RSS samples is calculated.

The processed RSS vector is stored in the RP and is expressed as:

\[
RP_i = (RSS_{i1}, RSS_{i2}, \ldots, RSS_{im})
\]  

where \( RP_i \) is the processed RSS values stored at the \( i \)th RP, and \( RSS_{ij} \) is the processed RSS of the \( j \)th AP at \( i \)th RP, \( i = 1, 2, \ldots, n \), where \( n \) is the number of RPs and \( j = 1, 2, \ldots, m \), where \( m \) is the number of existing APs.

The RPs are then stored in the fingerprint database along with their RP position (X, Y coordinates), and the format of the fingerprint database is shown in Table 1.
The first column represents the number of RPs in the database, X and Y are the coordinates (RP position) of the ith RP measured in the two-dimensional coordinate system of the floor plan. As mentioned in Equation (1), RSS\textsubscript{ij} represents the processed RSS value of the jth AP at the ith RP.

### 2.3. Strongest Access Point Information-Based Clustering Algorithm

We analyzed the similarities between RPs in the offline fingerprint database based on real-time data and found that the set of RPs obtained the strongest RSS value from the particular AP. We continued to analyze the similarities and collected 12 RPs in different locations in a two-dimensional coordinate system and analyzed their RSS values as shown in Table 2.

| RPs | RSS of AP1 | RSS of AP2 | RSS of AP3 | RSS of AP4 | RSS of AP5 | RSS of AP6 | SAP Label |
|-----|------------|------------|------------|------------|------------|------------|------------|
| 1   | -48        | -62        | -71        | -55        | -72        | -81        | 1          |
| 2   | -48        | -65        | -68        | -60        | -76        | -78        | 1          |
| 3   | -48        | -59        | -71        | -56        | -69        | -74        | 1          |
| 4   | -49        | -60        | -70        | -60        | -78        | -78        | 1          |
| 5   | -62        | -38        | -59        | -78        | -57        | -63        | 2          |
| 6   | -64        | -35        | -61        | -75        | -61        | -66        | 2          |
| 7   | -72        | -48        | -56        | -84        | -70        | -60        | 2          |
| 8   | -67        | -43        | -59        | -77        | -67        | -61        | 2          |
| 9   | -76        | -70        | -57        | -99        | -80        | -76        | 3          |
| 10  | -70        | -66        | -48        | -99        | -86        | -69        | 3          |
| 11  | -77        | -66        | -54        | -90        | -84        | -77        | 3          |
| 12  | -74        | -70        | -54        | -90        | -84        | -78        | 3          |

The strongest RSS value obtained in RP1, RP2, RP3, and RP4 is from AP1, and RP5 to RP8 is from AP2, and RP9 to RP12 is from AP3. According to this real-time result of preprocessed data, we can see that the set of RPs belongs to the particular AP coverage area. Likewise, we can assume that the user is also present somewhere near that same AP coverage area, and those APs are called SAPs.

Inspired by this distribution characteristic of the Wi-Fi signals, we propose an SAP information-based clustering algorithm. The proposed SAP cluster strategy takes more consideration of the signal propagation of the APs and their environment. The AP that obtained the strongest RSS value in RP\textsubscript{i} will be set as its SAP label and is denoted by:

\[
SAP_i = \arg\max_{j=1}^{m} RSS_{ij}
\]

where SAP\textsubscript{i} is the SAP label of the ith RP, \(\arg\max_{j=1}^{m} RSS_{ij}\) is the maximum value of RSS obtained from the jth AP at the ith RP, m is the number of APs for RP\textsubscript{i}. The SAP number will be updated at the SAP label column as shown in Table 2.

In the fingerprint clustering algorithm, the similarity criteria can be calculated using the signal distance (SD) between RPs using Euclidean distance [35], whereas, in this
proposed method, the similarity of the SAP labels between RPs is calculated using the following expression:

$$S_d(RP_i, RP_j) = |SAP_i - SAP_j|$$ (3)

where $S_d(RP_i, RP_j)$ represents the similarity distance between the SAP label of the $i$th RP and $j$th RP. To obtain the clusters, the SAP information-based clustering is implemented based on the SAP label distance between RPs, as indicated in Equation (3).

The set of clusters are denoted as:

$$C_l = \{RP_{l1}, RP_{l2}, \ldots, RP_{ln}\}$$ (4)

where $C_l$ is the group of RPs stored at the $l$th cluster, $l = 1, 2, \ldots, p$ where $p$ is the number of clusters. In this method, the number of clusters is fixed, and it depends on the number of deducted SAPs in the indoor environment. Then, the mean value of the RPs in $C_l$ will be calculated and assigned as the cluster centroid.

The set of cluster centroids are denoted as:

$$CC_r = \{CC_1, CC_2, \ldots, CC_q\}$$ (5)

where $CC_r$ is the number of cluster centroids stored in the $r$th CC and $r = 1, 2, \ldots, q$ where $q$ is the number of CC.

2.4. Online Localization Algorithm

In the online phase, we first collect the RSS vectors at the test point ($TP$), and it is expressed as:

$$TP = (RSS_1, RSS_2, \ldots, RSS_m)$$ (6)

where $TP$ is the set of RSS vectors collected at the $TP$ (user’s position), and $RSS_j$ is the collected RSS of $j$th AP at the $TP$.

2.4.1. Cluster Matching

The main goal of the delegate cluster selection step is to reduce the RP space, through the selection of a delegate cluster centroid that matches the online reading, according to the SAP similarity metric. Instead of comparing the $TP$ with all the RPs in the fingerprint database, we first compare the $TP$ with the SAP cluster centroids $CC_r$, then select the particular cluster with the minimum distance as the position set the $TP$ belongs to. The delegate cluster is selected using the following equation:

$$D(TP, CC_r) = \sqrt{\sum_{j=1}^{m} (RSS_j - RSS_{rj})^2}$$ (7)

where $D(TP, CC_r)$ is the distance between the $TP$ and cluster centroids, $RSS_{rj}$ is the mean RSS value of the $j$th AP at $r$th CC. After selecting the delegate cluster using Equation (7), the search space for the algorithm is reduced.

2.4.2. Weighted k-Nearest Neighbor Algorithm

Once the delegate cluster is chosen, the $TP$ is compared with all the RPs belonging to that cluster to determine the position of the user by applying the $W \ k \ NN$ algorithm. The $W \ k \ NN$ algorithm is an improved classification algorithm of $k \ NN$, and the working mechanism is to find $k$ nearest RPs, which have a minimum signal distance from the RSS observed at the user location, and weighted average the positions of the selected $k$ nearest RPs according to their signal distance. The $W \ k \ NN$ global method will apply weights to all the data points in the database, which also increases the computational overhead. To overcome this issue, the $W \ k \ NN$ local method is applied in this research. The weights are applied only to the $k$ found neighbors. It chooses the RP that has the highest weight and minimum distance to $TP$ in the signal space [36].
The $k$-nearest neighbor is calculated by:

$$D(TP, C_l) = \sqrt{\sum_{i=1}^{n} (RSS_i - RSS_{ij})^2}$$  \hspace{1cm} (8)

where $D(TP, C_l)$ is the distance between the TP and the RPs in the $l$th delegate cluster $C$. The distance values are sorted in ascending order, and the set of $k$ nearest RPs can be selected according to their minimum distance. In the $k$ NN algorithm, the $k$ value can be chosen as:

$$k = \{k_1, k_2, \ldots, k_u\}$$  \hspace{1cm} (9)

The coordinate of the $k$ nearest RPs with a minimum distance in all $D(TP, C_l)$ is calculated by:

$$(x, y) = \sum_{i=1}^{k} \tilde{\omega}_i (x_i, y_i)$$  \hspace{1cm} (10)

where $(x, y)$ is the user position in the Cartesian coordinates, $(x_i, y_i)$ is the position of the $i$th RP, and $\tilde{\omega}_i$ is the weighting factor of the $i$th $k$ selected RP. The weight $\tilde{\omega}_i$ can be calculated by:

$$\tilde{\omega}_i = \frac{1}{D(TP, C_l)^2} \frac{1}{\sum_{j=1}^{k} D(TP, C_l)^2_j}$$  \hspace{1cm} (11)

The closest RP with the highest weight and the lowest distance among the $k$ selected RP is considered as the user position.

3. Performance Evaluation

To validate the performance of our proposed method, two experiments were conducted. Experiment 1 was conducted in the information and communication engineering (ICE) department corridor, 4th floor of the engineering building in Wonkwang University, Republic of Korea. Figure 3 shows the schematic diagram of the corridor on the fourth floor. There were more than 20 APs deployed, and a total of 60 RPs were marked in the experimental field. The accuracy of the position depends on the density of RPs, so the distance between adjacent RPs was set to 1.8 m, and the length of the corridor was 60 m. The position of the RP was measured in the two-dimensional coordinate system and was marked in the X, Y coordinate of the floor plan. A Toshiba Satellite C50-A with an Intel Core i5-3230M CPU 2.60 GHz, 4 GB RAM, and a 64-bit OS was used as the server and the client machine. To deduct APs and collect Wi-Fi RSS data, we used a Wi-Fi network scanner application called “inSSIDer” version 4.4.0 from MetaGeek, LLC. Experiment 2 is explained in Section 3.4.

3.1. Results of the Improved RSS Measurement Technique

There are more than 15 APs deducted at each RP, but they were filtered as mentioned in Section 2.2. The mean and median value was taken from $N$ ($N = 5$) samples of the RP and is compared as shown in Figure 4. The RSS from APs 6, 7, 8, 11, 12, 14, 15 was not heard in some samples due to obstacles in the indoor environment, and these are called missing APs. In this case, $-99$ dBm was set as its maximum RSS value for those APs as mentioned in Section 2.2, and it makes a huge difference between the mean and median values. Generally, the RSS characteristics are highly based on the mean or probability of the Wi-Fi signal. However, due to fading and shadowing in a complex and ever-changing indoor environment, the characteristics of the RSS are spatially and temporally different. The basic general situations are: (1) The Wi-Fi signal probability distribution is negative-skewed in the presence of weak multipath interference; (2) the Wi-Fi signal probability distribution is positive-skewed if the multipath is stronger than the signal. The mean value of the selected RSS is used in [19]; due to the above-mentioned reasons, the mean value of
RSS may not be accurate, and it will affect the positioning accuracy, so the median value of RSS is generally considered to be the most suitable representative of the center of the RSS value.

**Figure 3.** Experimental environment 1. The blue dots represent the RPs, and the total size of the experimental area is 20 m × 60 m.

To verify that the improved RSS measurement will reduce Wi-Fi signal fluctuations, we compared the waveforms of the original N RSS samples and the processed RSS of the Wi-Fi signal, as shown in Figure 5. There are N (N = 5) RSS samples collected at the same place from 17 APs, and each sample has a different RSS value from the same AP with a different time interval. This shows the importance of signal preprocessing.

**Figure 4.** Comparison of mean and median RSS at the RP.
3.2. Results of the SAP Cluster Experiment

The RPs are grouped into three clusters using their SAP information, and their cluster centroid was calculated according to the description in Section 2.3, and as shown in Figure 6. The three different colors represent three different clusters.

To compare the performance of the SAP cluster with the traditional $k$-means, we set the $k$-means as a comparison group because it is one of the most widely used robust clustering algorithms for Wi-Fi RSS fingerprint-based indoor positioning [37]. We can see from Figure 6 that the similarity between RPs in cluster 0 shows that the strongest RSS is received from SAP 1, and in clusters 1 and 2, the strongest RSS is received from SAPs 2 and 3, respectively. The SAP cluster is plotted as shown in Figure 7.

Usually, in $k$-means cluster analysis, the Elbow method is used in determining the optimal $k$ value [22]. In this research, we used two different $k$ values for the $k$-means algorithm, and their result was compared with the proposed SAP algorithm to show the similarity correlation between RPs.

The cluster results of $k$-means and the comparison between $k$-means and SAP are shown in Figure 8. In the $k$-means cluster, $k = 2$, and $k = 5$ are applied and are plotted as shown in (a, c), and (b, d) shows the comparison between the $k$-means and the SAP cluster results. In (b, d), three different groups of RPs on the Y-axis 0, 1, 2 represent three different SAP clusters (cluster 0, 1, 2). The RPs with green and red colors in (b) represent $k$-means clusters where $k = 2$. Likewise, five different colors in (d) represent $k$-means clusters where
$k = 5$. The simulation result of the $k$-means algorithm shows that the RPs that are less relevant are divided into the same cluster when the $k$ value is small, and the RPs that are relatively relevant are divided into different clusters when the $k$ value is large. Whereas, in this proposed SAP cluster, the grouped RPs are more relevant to each other, and the number of clusters is fixed.

![Figure 7. SAP clustered RPs. A group of RPs in y-axis 0, 1, 2 represents cluster 1, 2, 3, respectively.](image1)

![Figure 8. Comparison of $k$-means and SAP clusters: (a) $k$-means cluster ($k = 2$); (b) $k$-means ($k = 2$) compared with SAP; (c) $k$-means cluster ($k = 5$); (d) $k$-means ($k = 5$) compared with SAP.](image2)
3.3. Results of the Positioning Experiment

The TPs were taken at the corridor of the experiment field, and the location of the user is estimated using the traditional methods and proposed method. The results of the positioning experiment of the proposed method are shown in Figure 9. The black arrow marks and circles represent the distance between the TP (original user location) and the estimated user location.

![Figure 9. Determined user location of the proposed method. The blue dots represent the RPs, and the orange dots represent the TPs. The RPs pointed with an arrow and circle is the determined user location.](image)

The error distance between the real user location (TP) and estimated user location is calculated using the below equation:

\[
DEE = \left( \frac{1}{T} \sum_{i=1}^{T} \left( x_{test} - x_{est} \right)^2 + \left( y_{test} - y_{est} \right)^2 \right)
\]  

(12)

where \(DEE\) is the distance estimation error between the TP and the estimated user location, \(x_{test}, y_{test}\) is the X Y coordinate of the TP, and \(x_{est}, y_{est}\) is the X Y coordinate of the estimated user location.

To evaluate the performance of our proposed algorithm, the positioning accuracy was compared with the traditional \(W_k\) NN algorithm, \(W_k\) NN + k-means algorithm, and decision tree algorithm, [25,30] shows that these algorithms are reasonable benchmarks. The comparison of DEE between the proposed algorithm and the other algorithms is shown in Figure 10.

As shown in Figure 10, 35% of TPs (6 out of 17 TPs) have achieved the minimum error (0 m) in the proposed method, whereas the \(W_k\) NN, \(W_k\) NN + k-means, and decision tree algorithms achieved 0% (0 out of 17 TPs), 5% (1 out of 17 TPs), and 11% (2 out of 17 TPs) respectively. The maximum error of the proposed method is 2.54 m, whereas the maximum error of \(W_k\) NN, \(W_k\) NN + k-means, and decision tree algorithm are 10.86, 6.41, and 26.1 m, respectively. The physical distance between the TP and the estimated user location is more consistent with the proposed method.

3.4. Experimental Environment 2

To validate the applicability of our proposed method in a different scenario with different hardware and AP positions, we experimented with our method in a larger indoor environment. Experiment 2 was conducted on the electronics convergence engineering (ECE) department, which is located on the 4th floor of the engineering building. The schematic diagram and the clustering results of experimental environment 2 are shown in Figure 11. A total of 179 RPs were marked in the experimental field, and the distance between adjacent RP was 1.8 m. A total of 895 RSS samples were collected and stored in 179 RPs after preprocessing according to Section 2.2. Overall, 25 APs were deducted during the site survey, and 10 APs were selected as SAPs according to their signal distribution in the environment. The RPs were grouped into 10 different clusters according to their SAP information. Different colors represent different clusters in Figure 11. The TPs were collected at different locations, and the user position is finally estimated.
Figure 10. Distance estimation error comparison in meters. Estimated user location of traditional methods and the proposed method.

Figure 11. Experimental environment 2. Blue dots represent the RPs, and the size of the experimental field is 60 m × 40 m.
4. Discussion

In this section, we examine the results of the proposed method and various algorithms used to acquire the user location. We collected 17 TPs in experimental environment 1 and 35 TPs in experimental environment 2. We analyzed the distribution of error distance using the cumulative distribution function (CDF), which is a method used to describe the distribution of error distance, and it is predominantly useful to observe the DEE as a cumulative error distribution function. For convenience, the CDF was adopted for the theoretical analysis in this paper. As indicated in [38], it is probable that a normal function variant is between 0 and 1, or it can be explained as the probability at which specific error occurred as in the following equation:

\[ CDF(d) = \frac{2}{\sqrt{\pi}} \int_{0}^{d} \text{error}^{-1} dt \] (13)

Using the above equation, we could identify the probability of positioning error at some definite distance. The probability of DEE is calculated, and the CDF is shown in Figure 12.

As shown in Figure 12, in experiment environment 1, the probability of getting less than 2 m of DEE in the proposed method is 0.941, whereas in the WkNN, WkNN + k-means, and decision tree are 0.058, 0.411, and 0.352, respectively. In experiment environment 2, the probability of getting less than 2 m of DEE in the proposed method is 0.571, whereas in the WkNN, WkNN + k-means, and decision tree are 0.371, 0.4, and 0.371, respectively. The CDF of the distance estimation error shows that the proposed method obtained a higher probability of better accuracy in both experiment environments 1 and 2 compared to the other methods, which proves that the proposed method is not limited to specific scenarios. The main reason for the improved accuracy is that the SAP cluster divides the RPs more accurately according to its SAP similarity between RPs. In this case, the TPs collected at the boundary of the clusters are well classified into its delegate...
cluster during simulation, which helps to avoid the error. The better clustering results have produced better positioning accuracy.

4.1. Computational Complexity

The main purpose of using clustering algorithms is to reduce the computational complexity of the online localization phase. Furthermore, it will lead to an improvement in positioning accuracy by eliminating possible outliers [35].

The $W_k$NN algorithm simply compares the $TP$ with the fingerprint database and calculates the distance between $TP$ and RPs. Then, the $k$ nearest RPs ($k = 4$) is found from the fingerprint database. Finally, the nearest RP with minimum distance and highest weight among the $k$ found neighbors is considered to be the user location. In this algorithm, to find the user location, the $TP$ is compared with all the RPs that are stored in the fingerprint database, which causes high computational complexity.

The $W_k$NN + $k$-means algorithm first groups the RPs and makes clusters. Instead of comparing the $TP$ with all the RPs in the database, only the particular cluster with minimum distance to the $TP$ is taken for the next process. Finally, the $W_k$NN algorithm is performed in that particular cluster to find the user location. For example, if there are 1000 RPs in the fingerprint database, it will be divided into several clusters using the $k$-means algorithm (if $k = 10$, each cluster will have 100 RPs), instead of comparing the $TP$ with all the 1000 RPs in the fingerprint database, it selects the single suitable cluster and calculates the user position by comparing only 100 RPs in that selected cluster. The remaining 900 RPs will be omitted, which helps to reduce the computational complexity. Meanwhile, as mentioned in Section 1, the $k$-means clustering algorithm randomly places its centroids to the fingerprint database to make clusters, and it does not consider the nonlinear similarity between RPs, which seriously reduced the positioning accuracy during the simulation.

Whereas, in this proposed method, the clusters are made perfectly by grouping the RPs using its SAP similarity. Therefore, the accuracy of this method is better than the other two compared algorithms. Since the proposed method has a clustering technique, so it can also reduce the computational complexity.

The Big O notation is used in this paper for complexity analysis. It is the most common metric for calculating runtime complexity (execution time). It describes the execution time of a task concerning the number of operations required to complete it. All these traditional and proposed methods are based on the $W_k$NN algorithm and the prediction time complexity of $W_k$NN is expressed by:

$$O(k \times n \times d)$$

(14)

where $k$ is the number of neighbors that we considered in the algorithm ($k = 4$), $n$ is the number of RPs in the training dataset, and $d$ is the data dimensionality (RSS vectors received from different APs). Equation (14) loop through all points $k$ times. First, it computes the distance between $TP$ and RPs in the training dataset, then it remembers the index of the element with the smallest distance and adds the class at found RP to the counter. Finally, it returns the class with the most votes as a determined user location. When the number of searching points (RP) in the dataset is high, the prediction time complexity will be higher. The dataset and the execution time are shown in Table 3.
Table 3. Structure of dataset and execution time based on simulation data.

| Method                | Experimental Environment 1 |       | Experimental Environment 2 |       |
|-----------------------|----------------------------|-------|----------------------------|-------|
|                       | Number of Clusters | Number of Rps in Each Cluster | Execution Time | Number of Clusters | Number of Rps in Each Cluster | Execution Time |
| WkNN                  | 0                          | 60                | 0.059000                 | 0      | 179 RPs                        | 0.300371       |
| WkNN + k-means        | 3                          | Cluster1–24 RPs | 0.028958                 | 3      | Cluster1–74 RPs                | 0.049465       |
|                       |                            | Cluster2–20 RPs | 0.022051                 |        | Cluster2–65 RPs                | 0.061348       |
|                       |                            | Cluster3–16 RPs | 0.026892                 |        | Cluster3–40 RPs                | 0.042885       |
|                       |                            |                   |                           |        | Cluster1–9 RPs                 | 0.014109       |
|                       |                            |                   |                           |        | Cluster2–27 RPs                | 0.015463       |
|                       |                            |                   |                           |        | Cluster3–16 RPs                | 0.016708       |
|                       |                            |                   |                           |        | Cluster4–14 RPs                | 0.011992       |
|                       |                            |                   |                           |        | Cluster5–21 RPs                | 0.015547       |
|                       |                            |                   |                           |        | Cluster6–15 RPs                | 0.019880       |
|                       |                            |                   |                           |        | Cluster7–24 RPs                | 0.013052       |
|                       |                            |                   |                           |        | Cluster8–12 RPs                | 0.015146       |
|                       |                            |                   |                           |        | Cluster9–22 RPs                | 0.014271       |
|                       |                            |                   |                           |        | Cluster10–19 RPs               | 0.015090       |
| Proposed method       | 3                          | Cluster1–23 RPs  | 0.014309                 | 10     | Cluster1–23 RPs                | 0.016508       |
|                       |                            | Cluster2–14 RPs  | 0.013577                 |        | Cluster2–14 RPs                | 0.016508       |
|                       |                            | Cluster3–23 RPs  | 0.016508                 |        | Cluster3–23 RPs                | 0.016508       |

As shown in the above table, there are no clusters in the WkNN method, and the TP is directly compared with all the RPs in the fingerprint database. In contrast, the WkNN + k-means algorithm has three clusters in both experimental environments 1 and 2 (the number of clusters in k-means comes from the Elbow method where \( k = 3 \)). On the other hand, the proposed method has three clusters in experimental environment 1, and 10 clusters in experimental environment 2 (the number of clusters in the proposed method comes from the SAP cluster technique). It can be seen from the results that the execution time highly depends on the number of RPs stored in that cluster, and the user location calculation is significantly reduced in the proposed method.

4.2. Error Statistics

To show the efficiency of the proposed method more intuitively, Table 4 lists the comparison of error statistics between the proposed method and the other traditional methods.

Table 4. Comparison of error statistics of the traditional algorithms and proposed method.

| Method                | Experimental Environment 1 |       | Experimental Environment 2 |       |
|-----------------------|----------------------------|-------|----------------------------|-------|
|                       | 50% Error | 75% Error | Mean Error | 50% Error | 75% Error | Mean Error |
| WkNN                  | 4.02      | 4.58      | 4.46        | 2.6      | 5.4       | 3.14       |
| WkNN + k-means        | 2.3       | 2.54      | 2.86        | 2.5      | 5         | 3.13       |
| Decision tree         | 3.6       | 8.99      | 7.77        | 2.54     | 4.02      | 6.49       |
| Proposed method       | 0.69      | 1.2       | 0.85        | 1.25     | 2.6       | 2.11       |

The error metrics used in Table 4 are followed from [30]. As we can see from the above table, the 50% and 75% mean error of DEE of the proposed method is less than the other compared methods in both experimental environments. The mean error of DEE is used in this paper as a metric to judge the positioning accuracy. The mean error of the proposed method in experimental environment 1 is 80.94%, 69.69%, and 89.06%, and in experimental environment 2 is 32.80%, 32.58%, and 67.48% lower than the mean error of the other methods used for comparison. We can see from Figures 10 and 12 and Tables 3 and 4 that the proposed indoor positioning algorithm outperforms traditional WkNN, WkNN + k-means, and decision tree indoor positioning algorithms. The results indicate that the proposed method can significantly improve positioning accuracy.
The positioning accuracy of this Wi-Fi RSS fingerprint-based indoor positioning highly relies on the number of acquired RPs and RSS observed in the indoor environment. Meanwhile, the ever-changing APs in the indoor environment (for example, the addition of new APs and removal of damaged existing APs) are one of the nagging issues of this method. Moreover, the changes in the indoor infrastructure (modifying the structure of the environment, for example, adding new rooms and removing existing partitions) will also reflect the fingerprint database. Due to the above-mentioned reasons, the positioning accuracy of this method is limited.

5. Conclusions

In this paper, we presented an efficient Wi-Fi indoor positioning method based on the IRSSMT Wi-Fi signal extraction method and the SAP information-based clustering algorithm. The proposed method is mainly intended to cope with the issue of multipath interference in indoor environments, impacts of the average value of missing APs, and randomly constructed clusters with randomly placed cluster centroids. The proposed method is based on the analysis of the characteristics of the Wi-Fi signals. The selected RSS median value is employed for better offline fingerprint database construction, and the SAP information reflects the better RP similarity and helps to create the most appropriate cluster. The performance of this proposed method is evaluated by experiments in two different indoor environments. The results show that the proposed IRSSMT and SAP information-based clustering algorithm outperform the traditional indoor positioning algorithms. Besides, the SAP information-based clustering algorithm produces significantly better clustering solutions quite consistently according to the SAP similarity measures of cluster quality, and the proposed algorithm has better-improved accuracy by 89.06% and 67.48% when compared to the other two algorithms. In addition, the run time of the proposed method is incredibly attractive, which greatly reduced the computational complexity. For future work, we will analyze and modify various filters, clustering algorithms, and localization algorithms to enhance the environmental adaptability of fingerprint positioning.

Author Contributions: Conceptualization, B.E., M.S., and K.P. (Kwangjin Park); methodology, B.E., M.S., and K.P.; software, B.E.; writing—original draft preparation, B.E.; writing—review and editing, M.S. and K.P.; supervision, M.S. and K.P.; funding acquisition, K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This paper was supported by Wonkwang university, Korea in 2020.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this manuscript.

References
1. Liu, F. Survey on WiFi-based indoor positioning techniques. IET Commun. 2020, 14, 1372–1383. [CrossRef]
2. Huang, H.; Gartner, G.; Krisp, J.M.; Raubal, M.; Van De Weghe, N. Location based services: Ongoing evolution and research agenda. J. Locat. Based Serv. 2018, 12, 63–93. [CrossRef]
3. Huang, H.; Gartner, G. Current Trends and Challenges in Location-Based Services. ISPRS Int. J. Geo-Inf. 2018, 7, 199. [CrossRef]
4. Enge, P.; Misra, P. Special Issue on Global Positioning System. Proc. IEEE 1999, 87, 3–15. [CrossRef]
5. Yassin, M.; Elias, R. A survey of positioning techniques and location based services in wireless net-works. In Proceedings of the 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Kozhikode, India, 19–21 February 2015; pp. 1–5.
6. Memon, S.; Memon, M.M.; Shaikh, F.K.; Laghari, S. Smart indoor positioning using BLE technology. In Proceedings of the 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), Salmabad, Bahrain, 29 November–1 December 2017; pp. 1–5.
7. Liu, C.; Cheng, Z.; Zhang, Y.; Wang, G. An indoor positioning system based on RFID with rotating antenna and passive tags. In Proceedings of the 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, China, 29–31 December 2017; pp. 455–459.

8. Cheng, Y.; Zhou, T. UWB Indoor Positioning Algorithm Based on TDOA Technology. In Proceedings of the 2019 10th International Conference on Information Technology in Medicine and Education (ITME), Qingdao, China, 23–25 August 2019; pp. 777–782.

9. Shahidi, S.; Valae, S. GIPSy: Geomagnetic indoor positioning system for smartphones. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–7.

10. Naz, A.; Asif, H.M.; Umer, T.; Kim, B.-S. PDOA Based Indoor Positioning Using Visible Light Communication. IEEE Access 2018, 6, 7557–7564. [CrossRef]

11. Bai, Y.B.; Wu, S.; Retscher, G.; Kealy, A.; Holden, L.; Tomko, M.; Borriak, A.; Hu, B.; Wu, H.R.; Zhang, K. A new method for improving Wi-Fi-based indoor positioning accuracy. J. Locat. Based Serv. 2014, 8, 135–147. [CrossRef]

12. Yang, C.; Huai-Rong, S. WiFi-based indoor positioning. IEEE Commun. Mag. 2015, 53, 150–157. [CrossRef]

13. Wu, C.; Hou, H.; Wang, W.; Huang, Q.; Gao, X. TDOA Based Indoor Positioning with NLOS Identification by Machine Learning. In Proceedings of the 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 18–20 October 2018; pp. 1–6.

14. Chen, L.; Li, B.; Zhao, K.; Rizos, C.; Zheng, Z. An Improved Algorithm to Generate a Wi-Fi Fingerprint Database for Indoor Positioning. Sensors 2013, 13, 11085–11096. [CrossRef] [PubMed]

15. Zhang, S.; Guo, J.; Luo, N.; Zhang, D.; Wang, W.; Wang, L. A Calibration-Free Method Based on Grey Relational Analysis for Heterogeneous Smartphones-Based Indoor Positioning. Sensors 2019, 18, 3885.

16. Xia, S.; Liu, Y.; Yuan, G.; Zhu, M.; Wang, Z. Indoor Fingerprint Positioning Based on Wi-Fi: An Overview. ISPRS Int. J. Geo-Inf. 2017, 6, 135. [CrossRef]

17. Li, C.; Zhengyang, Q.; Changtong, L. An improved weighted k-nearest neighbor algorithm for indoor positioning. Wirel. Pers. Commun. 2017, 96, 2239–2251. [CrossRef]

18. Xia, M.; Chen, J.; Song, C.; Li, N.; Chen, K. The indoor positioning algorithm research based on improved location fingerprinting. In Proceedings of the 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 23–25 May 2015; pp. 5736–5739.

19. Xue, W.; Qiu, W.; Hua, X.; Yu, K. Improved Wi-Fi RSSI Measurement for Indoor Localization. IEEE Sens. J. 2017, 17, 2224–2230. [CrossRef]

20. Apolinario, J.A.; Yazdanpanah, H.; Nascimento, A.S.; De Campos, M. A Data-selective LS Solution to TDOA-based Source Localization. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 4400–4404.

21. Kilani, M.B.; Raymond, A.J.; Gagnon, F.; Gagnon, G.; Lavoie, P. RSSI-based indoor tracking using the extended Kalman filter and circularly polarized antennas. In Proceedings of the 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), Dresden, Germany, 11–13 March 2014; pp. 1–6.

22. Ma, R.; Guo, Q.; Hu, C.; Xue, J. An Improved WiFi Indoor Positioning Algorithm by Weighted Fusion. Sensors 2015, 15, 21824–21843. [CrossRef] [PubMed]

23. Park, C.; Rhee, S.H. Indoor positioning using Wi-Fi fingerprint with signal clustering. In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 18–20 October 2017; pp. 820–822.

24. Razavi, A.; Mikko, V.; Elena-Simona, L. K-means fingerprint clustering for low-complexity floor esti-mation in indoor mobile localization. In Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015; pp. 1–7.

25. Chen, Y.; Liu, W.; Zhao, H.; Cao, S.; Fu, S.; Jiang, D. Bisecting k-means based fingerprint indoor localization. Wirel. Netw. 2020, 1–10. [CrossRef]

26. Liu, W.; Fu, X.; Deng, Z.; Xu, L.; Jiao, J. Smallest enclosing circle-based fingerprint clustering and modified-WKNN matching algorithm for indoor positioning. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain, 3–6 October 2016; pp. 1–6.

27. Kargar, P.A. Wireless fingerprinting indoor positioning using affinity propagation clustering meth-ods. Wirel. Netw. 2018, 24, 2825–2833. [CrossRef]

28. Subedi, S.; Gang, H.-S.; Ko, N.K.; Hwang, S.-S.; Pyun, J.Y. Improving indoor fingerprinting positioning with affinity propagation clustering and weighted cen-troid fingerprint. IEEE Access 2019, 7, 31738–31750. [CrossRef]

29. Abusara, A.; Hassam, M.S.; Ismail, M.H. Reduced-complexity fingerprinting in WLAN-based indoor positioning. Telecommun. Syst. 2017, 65, 407–417. [CrossRef]

30. Wang, B.; Liu, X.; Yu, B.; Jia, R.; Gan, X. An Improved WiFi Positioning Method Based on Fingerprint Clustering and Signal Weighted Euclidean Distance. Sensors 2019, 19, 2300. [CrossRef]

31. Zhou, B.; Li, Q.; Mao, Q.; Tu, W. A Robust Crowdsourcing-Based Indoor Localization System. Sensors 2017, 17, 864. [CrossRef]

32. Kim, J.; Han, D. Passive WiFi Fingerprinting Method. In Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nantes, France, 24–27 September 2018; pp. 1–8.
33. Laska, M.; Blankenbach, J.; Klamma, R. Adaptive indoor area localization for perpetual crowdsourced data collection. *Sensors* **2020**, *20*, 1443. [CrossRef]

34. Han, D.; Jung, S.; Lee, M.; Yoon, G. Building a Practical Wi-Fi-Based Indoor Navigation System. *IEEE Pervasive Comput.* **2014**, *13*, 72–79. [CrossRef]

35. 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]

36. Khodayari, S.; Maleki, M.; Hamedi, E. A RSS-based fingerprinting method for positioning based on historical data. In Proceedings of the 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS’10), Ottawa, ON, Canada, 11–14 July 2010; pp. 306–310.

37. Hu, X.; Shang, J.; Gu, F.; Han, Q. Improving Wi-Fi indoor positioning via AP sets similarity and semi-supervised affinity propagation clustering. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 109642. [CrossRef]

38. Ismail, A.; Kitagawa, H.; Tasaki, R.; Terashima, K. WiFi RSS fingerprint database construction for mobile robot indoor positioning system. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 001561–001566.