Enhanced Weighted K-Nearest Neighbor Algorithm for Indoor Wi-Fi Positioning Systems

Beomju Shin
Department of Information & Communication Engineering
Sejong University
Seoul, Republic of Korea
zxzx7@hanmail.net

Jung Ho Lee
Sensor System Research Center
Korea Institute of Science & Technology
Seoul, Republic of Korea
salbbaenda@kist.re.kr

Taikjin Lee
Sensor System Research Center
Korea Institute of Science & Technology
Seoul, Republic of Korea
taikjin@kist.re.kr

Hyung Seok Kim
Department of Information & Communication Engineering
Sejong University
Seoul, Republic of Korea
hyungkim@sejong.ac.kr

***Abstract*— Location-based systems for indoor positioning have been studied widely owing to their application in various fields. The fingerprinting approach is often used in Wi-Fi positioning systems. The K-nearest-neighbor fingerprinting algorithm uses a fixed number of neighbors, which reduces positioning accuracy. Here, we propose a novel fingerprinting algorithm, the enhanced weighted K-nearest neighbor (EWKNN) algorithm, which improves accuracy by changing the number of considered neighbors. Experimental results show that the proposed algorithm gives higher accuracy.**

Keywords— Location based system, Wi-Fi positioning system, Indoor navigation system, Fingerprinting.

I. INTRODUCTION

Location based systems (LBSs) have been applied in many domains and are required in various fields, e.g., mobile commerce, parcel or vehicle tracking, discovering the nearest shops or restaurants, social networking, etc. Positioning is a significant technique in LBS and is defined as the determination of the device's location, either relative to other devices in a system or on a global level [1]. Positioning is classified into two main categories: outdoor and indoor positioning. In outdoor positioning, the global positioning system (GPS) is usually used to locate the position of the target. Because positioning using a GPS usually requires line of sight (LOS) to ensure reasonable accuracy, it is unsuitable for indoor positioning. Therefore, a Wi-Fi positioning system (WPS) is used for indoor positioning in most cases, because Wi-Fi access points (APs) are installed at many indoor and outdoor locations and they are spreading rapidly. The Wi-Fi approach is also suitable in terms of both cost and accessibility.

Wi-Fi indoor positioning approaches are mainly of two types: a triangulation approach using the radio propagation model (RPM) and a fingerprinting approach. In the triangulation approach, an RPM is constructed and distance is estimated through the propagation model and the received signal strength (RSS) between the AP and mobile stations. Though this approach facilitates distance estimation, it cannot provide accurate positioning because the RSS of Wi-Fi is easily influenced by surroundings [2]. In contrast, fingerprinting gives better results than the triangulation approach because fingerprinting uses a database in which information on surrounding APs is stored. The database contains all reference points (RPs) and each RP has information on APs that can be received by mobile stations. The triangulation approach typically uses three APs to locate a position, whereas fingerprinting uses more AP information. Study of [3] introduced the K-nearest neighbor (KNN) algorithm, based on the fingerprinting algorithm, to locate mobile stations indoors. The KNN algorithm is a tool that entails selecting the nearest K neighbors around a device to determine its own position.

Using a fixed number (K) of nearest neighbors may decrease positioning accuracy in the KNN algorithm. If K is not changed during the positioning process, sometimes, neighbors far from the mobile station might be included in the K nearest neighbors. Therefore, eliminating some neighbors far from the mobile station is necessary. This process would increase computational cost; however, smart phones or tablet PCs that can be used for indoor positioning have high-performance processors and enough resources to handle the complexity of fingerprinting. In this paper, we propose the enhanced weighted K-nearest neighbor (EWKNN) algorithm to achieve improved accuracy over existing fingerprinting algorithms.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 proposes the EWKNN algorithm as a new positioning algorithm that improves accuracy. Section 4 describes the test bed and experimental results. Finally, Section 5 concludes the paper.

II. OVERVIEW OF PREVIOUS WORK

LBSs using a Wi-Fi signal have the advantage of not requiring additional implementation or cost because they utilize the Wi-Fi network infrastructure. The strength of a Wi-Fi signal received by a mobile station is inversely proportional to the distance over which it is transmitted. Using this relationship, a mobile station can estimate the distance between itself and an AP using the RSS.

A. Triangulation Approach

This technique is described in [3] in detail. It entails constructing an RPM, representing the surrounding environment, and selecting proper parameters, such as the maximum number of walls and the number of obstructions, in order to calculate the distance. The mobile station receives signals from each AP and estimates the distance between itself and an AP, with a priori knowledge of AP's positions, by the RPM. The location is usually calculated through a triangulation method. This approach does not need much preliminary effort relative to the fingerprinting technique. However, its positioning accuracy is poorer than that of fingerprinting.

B. Fingerprinting Approach

Fingerprinting involves two stages: offline and online stages. The objective of the offline stage is to make a database and that of the online stage is to determine the position using the database and the measured RSS. Making the database is significant in the fingerprinting process. All RP information, such as the RSS and the RP position, is saved in the database. An RP located on the map is a standard point used to find the position of a user, based on the RSS. The distance between RPs is an important factor influencing the accuracy of the estimated position. If the distance between RPs increases, the positioning accuracy decreases, and vice versa. However, installing a large number of RPs increases offline effort significantly, so it is essential to set a proper distance between RPs. At each RP, the RSSs of signals from surrounding APs are measured and recorded in the database, and this procedure is repeated for all the RPs. This requires much effort for manual measurements and, as such, a method for reducing this offline effort has previously been proposed in [4]. In the online stage, the mobile station receives RSS indications (RSSIs) from surrounding APs. The measured data are compared with the data stored at the database, and the mobile station chooses an RP that has the lowest distance with RSSs.

To improve the positioning accuracy, it is better to consider multiple RPs because the location of a mobile station is not precisely the same as the location of a RP and there may be several RPs whose distance are similar. If a low RSS is indicated, multiple neighbors can alleviate the problem caused by it. Furthermore, by computing proper weightings, the WKNN algorithm can provide improved accuracy. However, WKNN with a fixed number of RPs cannot always achieve the required accuracy, because some RPs that do not have RSSs equivalent to others can be included in the fixed number of neighbors, and this reduces the positioning accuracy. Therefore, it is better to change the number of RPs during the positioning process of the fingerprinting approach.

III. Proposed Algorithm

The KNN algorithm uses a fixed number of neighbors to calculate the position of a mobile station. Although its accuracy is better than the nearest neighbor (NN) algorithm, its results are not always accurate. For example, if the number of nearest neighbors, K, is set to 5 when finding the present position, the mobile station calculates its own position based on exactly five neighbors. Because of the fixed value of K, sometimes, several of the K neighbors have large distance. If only three out of five have small distance and the remainder has large distance, this case does not provide better accuracy than results from considering just three neighbors that have low distance.
Furthermore, in a corridor that is a one-dimensional space, a few nearest neighbors may be present around a device. In a space such as a hall, there are usually more nearest neighbors available than are present in a corridor, which means that the hall-type space has more RPs with low distance than does a corridor-type space. For this reason, the number of nearest neighbors the device chooses should vary, based on the type of space in which it is located. During the online phase, it is necessary to select the appropriate value of K dynamically, to improve positioning accuracy.

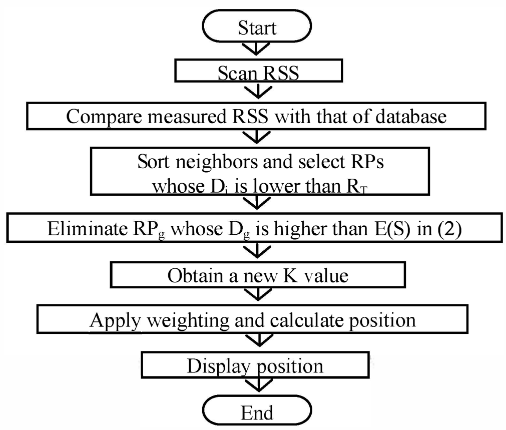

Figure 1. Flow diagram of the proposed algorithm.

Now we explain the proposed algorithm. The mobile station receives RSSIs from surrounding APs, compares their RSSs with the RSSs of RPs in the database, and then calculates D_i for each RP_i as

$$D_i = \sum_{j=1}^{N} |A_j - R_{i,j}|, \qquad i = 1, 2, 3, \ldots, L \quad (1)$$

where A_j is the RSS from the jth AP, $R_{i,j}$ is the RSS of the jth AP at the ith RP stored in the database, N is the number of APs, and L is the number of candidate RPs. The mobile station receives RSSIs from surrounding APs and selects candidate RPs available from the list of surrounding APs. A set of D_i is obtained and arranged in ascending order, i.e., the smallest one among D_i becomes D_1 and the largest becomes D_L. Let R_T denote the threshold for filtering RPs. RPs whose D_i is larger than the threshold R_T are removed in the list.

Let G denotes the number of remaining RPs and let S_g denote the difference between D_1 and D_g ($g = 2, \ldots, G$). The average of differences is obtained as follows:

$$E(S) = \frac{(S_2+S_3+\cdots+S_G)}{G-1}. \tag{2}$$

The RPs that have a larger S_g than E(S) are neglected and the RPs whose difference S_g is less than or equal to E(S) are retained in the location calculation. K, in our algorithm, is the number of remaining RPs. The aforementioned process is repeated periodically, and a new set of K RPs is selected every period.

In order to improve accuracy, each RP's location is weighted on the basis of its D_i [5]. The algorithm using a dynamically varying K and weighting is called the EWKNN. The position of the device is estimated as the average of the coordinates of K RPs. Using the EWKNN algorithm, the position of a device is obtained as follows:

$$P = \frac{\frac{1}{D_1}L(RP_1)+\frac{1}{D_2}L(RP_2)+\ldots+\frac{1}{D_K}L(RP_K)}{\frac{1}{D_1}+\frac{1}{D_2}+\frac{1}{D_3}+\ldots+\frac{1}{D_K}}, \tag{3}$$

where $L(RP_i)$ denotes the location of RP_i and P denotes the estimated position of the device. The flow diagram of the proposed algorithm is shown in Fig. 1.

IV. Implementation and Experiment

We compare the proposed EWKNN algorithm with the NN, KNN, and WKNN algorithms, through an experiment. The experiment was conducted in a building at the Korea Institute of Science and Technology (KIST) in Seoul, Republic of Korea. The size of the test space is 48 m × 22 m; pre-installed APs were used for this experiment. The smart phone we used is a Nexus-one manufactured by HTC Corporation, which has Android 2.3 O/S installed.

Fig. 2 shows each RP position and user movements. There are 46 RPs and the distance between a RP and it's the nearest neighbor RP is 3 m. A person moved counterclockwise. In our experiment, the database is located in the smart phone and the RSSs of APs measured at each RP are stored in the smart phone. In order to construct a database before the online stage, we chose RPs and collected RSSs from surrounding APs ten times using the smart phone. The mean of the RSSs was stored on a database in the smart phone.

Fig. 3 shows the impact of distance between RPs on positioning error in each algorithm. When the distance between RPs is short due to many RPs, the positioning error is small. However, too many RPs may incur many manual efforts for making database.

Fig. 4 shows positioning error with various K. Positioning error decreases until K increases up to 5, and after that, positioning error increases. Thus, we set K to 5 in KNN and WKNN algorithm.

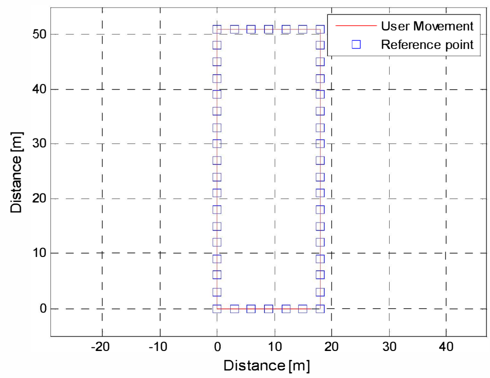

Figure 2. Position of reference points and user movement.

We obtained the location error between the estimated position and the actual position under the same conditions (3m distance between RPs) in each algorithm. Table 1 lists the location error result for each algorithm. The NN algorithm uses just one neighbor, so its location error is larger than the other algorithms. The result of the KNN algorithm shows better positioning accuracy than the NN algorithm does. Because of the weighting on each neighbor, the WKNN algorithm provides better positioning accuracy than the KNN algorithm. The proposed EWKNN algorithm shows the smallest location error of all of them because the number of neighbors, *K*, is changed appropriately during the course of operation.

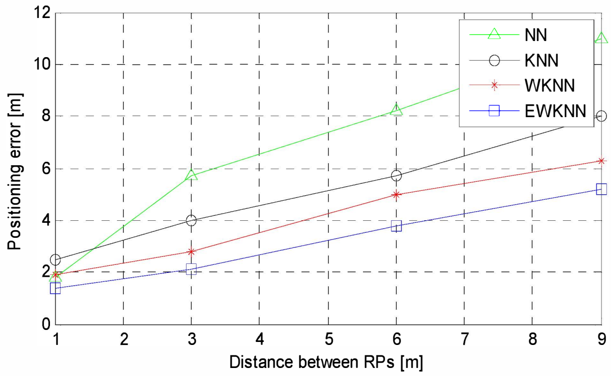

Figure 3. Impact of distance between RPs on positioning error.

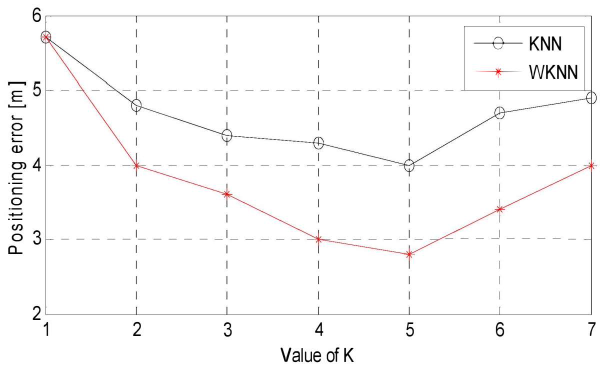

Figure 4. Impact of value of K on positioning error.

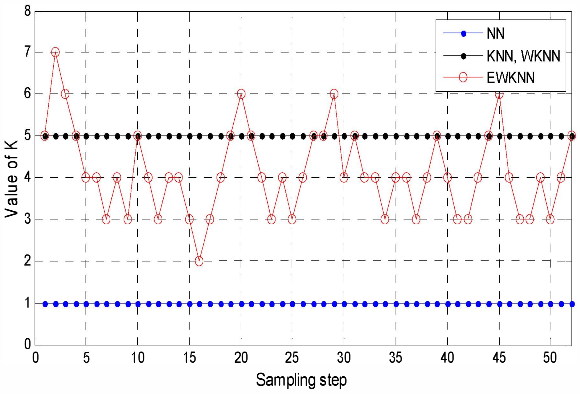

Figure 5. Value of K at sampling step in each algorithm.

TABLE 1. Positioning error of each algorithm.

Algorithm	Positioning error
NN	5.7126 m
KNN	4.0473 m
WKNN	2.8661 m
EWKNN	2.1077 m

Fig. 5 shows the value of K at sampling steps when using each algorithm. Algorithms except EWKNN have their own fixed K values, but EWKNN changes its K to consider only RPs having short distances from present position of device.

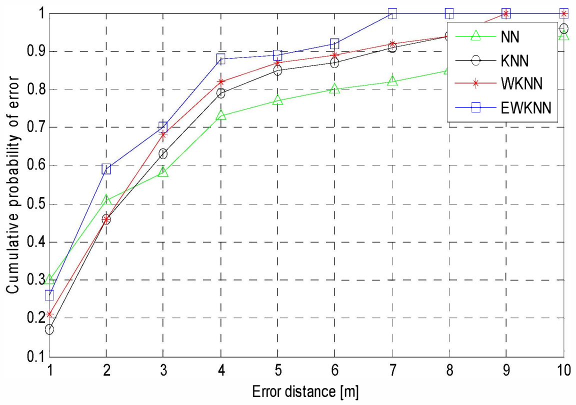

Figure 6. Cumulative probability of error distance in each algorithm.

Fig. 6 illustrates the probability of localizing the device within a given error distance. EWKNN shows the highest cumulative location error distribution among all algorithms. When the cumulative location error distribution is 90%, the error distance is 6.5m in WKNN while in EWKNN it is 5.3m.

EWKNN has higher complexity than the compared others. However, because EWKNN has only an additional part dynamically changing K when compared with the others, its complexity's gap from others is small.

V. Conclusion

We developed an indoor WPS for Android smart phones. Positioning using Wi-Fi signals is easy to implement and requires lower cost than other localization systems. We installed APs dedicated for localization at specific locations to improve positioning accuracy. We proposed a new algorithm to filter error signals and find the location of the smart phone. It acquires a proper scan time and threshold thereby yielding a low error rate. We expect the indoor WPS for smart phones to be used at various places.

Acknowledgment

This research was supported by MKE (The Ministry of Knowledge Economy), Korea, under the Convergence-ITRC (Convergence Information Technology Research Center) Support Program (NIPA-2012-C6150-1201-0003) supervised by NIPA (National IT Industry Promotion Agency) and was also supported by Seoul R&BD Program (SS110012C0214831).

References

[1] D. Tancharoen and K. Aizawa, "Wearable Video Retrieval and Navigation System using GPS Data," *IEEE Int. Conf. Computer and Information Technology*, 2010.

[2] K. Kaemarungsi and P. Krishnamurthy, "Properties of Indoor Received Signal Strength for WLAN Location," *Int. Conf. on IEEE Mobile and Ubiquitous Systems: Networking and Services*, 2004.

[3] P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," Microsoft Research, 2000.

[4] Xiaoyong chai and qiang Yang, "Reducing the Calibration Effort for Probabilistic Indoor Location Estimation," *IEEE Transaction on Mobile Computing*, vol. 6, no. 6, June 2007.

[5] Binghao Li, James Salter, Andrew G. Dempster and Chris Rizos, "Indoor Positioning Techniques Based on Wireless LAN", IEEE International Coference on Wireless Broadband and Ultra Wideband Communications, 2007.