An iterative scheme for feature-based positioning using weighted dissimilarity measure

Caifa Zhou (caifa.zhou@geod.baug.ethz.ch), Andreas Wieser

May 21, 2019

Abstract

We propose an iterative scheme for feature-based positioning using the weighted dissimilarity measure with the goal of reducing large errors. The computation of the weights is derived from the robustly estimated variability of the reference fingerprint map (RFM). The location-wise standard deviation estimation for each individual feature, which is treated as an additional layer of the RFM, is obtained by analyzing the kinematically collected RFM using spatial filtering and kernel smoothing. During the positioning stage, the weighting scheme iteratively adapts the contribution of each feature to the dissimilarity measure, which quantifies the difference between the online measured features and the ones stored in the RFM, according to its variability when searching the candidate estimation of the user’s position. These searched locations are subsequently used for refining the estimation of the user’s location. The experimental results show that the percentage of locations whose positioning error is larger than 5 m is reduced from 11% to 4% (i.e. reduced by almost two-fold) comparing to the performance of kNN without weighted dissimilarity measure.

Index terms: weighted dissimilarity measure; iterative scheme; signals of opportunity; feature-based indoor positioning

1 Introduction

Feature-based (i.e. fingerprinting-based) indoor positioning systems (FIPSs), one of the promising indoor positioning solutions, have been proposed using varies technologies (e.g. WLAN/BLE signals (Padmanabhan et al., 2000; Youssef and Agrawala, 2008), geomagnetic field (He and Shin, 2018) or machine visions (Guan et al., 2016)) for providing indoor location-based services (LBSs) to pedestrians (Brena et al., 2017; He and Chan, 2016; Pei et al., 2016). The positioning accuracy of the state-of-the-art FIPSs using the received signal strength (RSS) of WLAN APs is in the range of a few meters (Mautz, 2012), which is in most cases adequate for pedestrian indoor positioning and navigation. However, unexpected and unacceptable large errors (e.g. ≥ 20 m in horizontal error (Torres-Sospedra et al., 2017)) observed in the real environments jeopardize the practical usability of FIPSs (Wu et al., 2017; Torres-Sospedra and Moreira, 2017). The large error in positioning is highly likely caused by the large deviations of the feature values when performing the location estimation (Kaemarungsi and Krishnamurthy, 2012).

In order to leverage the attractive characteristics of FIPS while being able to circumvent the limitation of large errors, the trend is to combine the feature-based positioning
with other techniques. Hybrid positioning approaches combine the feature-based information with e.g. pedestrian dead reckoning (PDR) (Li et al., 2016), indoor map matching (Wang et al., 2015), the structure of buildings (Wang et al., 2012), infrared (Bitew et al., 2015), or BLE signals (Zhuang et al., 2016). In addition, Bayes filtering methods, such as Kalman and particle filters are introduced to continuously track the trajectory of pedestrians (Li et al., 2016; Robesaat et al., 2017). Though the merge of different positioning solutions mitigates the impact of large errors on the quality of LBSs, it requires either to deploy additional infrastructure or to provide extra information (e.g. the indoor map). On the other hand, the detection of large errors in FIPS using only intrinsically available data and their reduction using robust estimation schemes, has attracted little research attention in the past, see e.g. (Wu et al., 2017; Torres-Sospedra and Moreira, 2017; Lemic et al., 2019).

The proposed approach for reducing large errors in FIPS is started with analyzing the variability of feature values stored in a reference fingerprint map (RFM), which is generated during the offline phase for representing the relationship between locations and their associated features. The variability estimation is carried out by empirically analyzing the spatial distribution of the raw data (e.g. RSS values) included in the RFM and yields the extended representation of the RFM, which contains not only the spatially smoothed feature values, but also the location-wise robustly estimated standard deviation (STD) of each individual feature (see Section 4). The estimated location-wise STD values of the individual features can be widely employed into potential use cases, e.g. identifying changes of features for keeping the RFM up-to-date (He and Chan, 2016; Tao and Zhao, 2018; He et al., 2016). We then use the estimated variability of features to enhance the robustness the location estimation for mitigating large errors occurred in positioning. A weighted dissimilarity measure, which quantifies the difference between the online measured features and the ones stored in reference fingerprint map (RFM), is introduced for adapting the contribution of individual features to the difference relative to their estimated STD values (see Section 5.1). The positioning process is carried out in an iterative way because we need to assume the user’s location, which is required for retrieving the STD of the online measured features (see Section 5.2).

Though we have introduced additional steps into the two phases of FIPS, these require no extra acquisition of data and have minor impact on the positioning service during the online phase. On the one hand, the estimation of the variability of the RFM is carried out during the offline phase using only the raw RFM. On the other hand, our way of the iterative positioning is still able to provide an initial estimation of the user’s location by using a typical feature-based positioning approach (e.g. kNN), while being capable of refining the estimation of the location using the iteratively searched locations. This search process can also be implemented in the background (e.g. on a server) in case of limited computation resource on the mobile device.

The remaining of the paper is organized as follows: Section 2 summarizes the work related to reducing large errors in an FIPS. The fundamentals of the feature-based positioning are briefly described in Section 3. The robust estimation of the variability of the RFM and its application to positioning are presented in Section 4 and 5 respectively. Finally, the evaluation of the variability estimation as well as the positioning performance using the iterative scheme are presented in Section 6.
2 Related work

Herein we focus on the works that address the detection and reduction of large errors in an FIPS. We refer the interested readers for more information about the indoor positioning to (Mautz, 2012; He and Chan, 2016; Brenæ et al., 2017). Furthermore, a comprehensive comparison of different feature-based indoor positioning algorithms using various similarity/dissimilarity metric is available in (Retscher and Joksch, 2016; Torres-Sospedra et al., 2015; Minaev et al., 2017). Finally, a short review of the methods used for generation/creation of the RFM can be found in e.g. (He and Chan, 2016; Zhou and Wieser, 2019).

(Torres-Sospedra and Moreira, 2017) provides a detailed analysis of the sources of large errors when employing deterministic feature-based positioning approaches (e.g. kNN). In their analysis, which is based on the simulations under different indoor scenarios, they consider the influence of several factors, such as the quantization error of signal acquisition, the density of the reference measurement, and the selected dissimilarity metrics on the positioning error. One relevant finding in their analysis is that the large errors mostly occur at the locations where both the mean and the maximum value of RSS are low. However they do not validate their analysis in the deployed FIPSs. On a related note, (Kaemarungsi and Krishnamurthy, 2012) proposes to simply disregard features with a large standard deviation for the estimation of the user’s position.

There are only few works that focus on the reducing or estimating the positioning errors based on the analysis of the RFM. (Wu et al., 2017) introduces a weighted dissimilarity measure by computing the discriminative indicator for each feature according to the Log-distance path loss model. However, they do not take the variability of the online measured features, which has impact on the estimation of the discriminative factor, into account. In (Lemic et al., 2019) and (Li et al., 2019), the authors propose different regression models (e.g. neural networks, random forest, or Gaussian processes) for estimating the positioning errors and uncertainties that can be used to improve the performance of tracking pedestrian’s trajectories. Even if the focus of their work is not on the reduction of large errors in the FIPSs, their results suggest that these kind of regression-based error prediction models cannot help to reduce the large errors because the predicted positioning errors have a large uncertainty.

Compared to previous publications, we carry out the variability analysis of the RFM using a kinematically collected dataset, which includes not only the noise originating from the short term fluctuations of the features measured by a mobile device, but also the noise introduced by the motion status (e.g. moving speed and headings) of the mobile device. This setup is closer to the realistic situation of positioning and tracking pedestrians. The estimation of the variability is based solely on the raw RFM and is later used for reducing large errors by introducing an iterative scheme with the weighted dissimilarity measure into the online positioning phase.

3 Feature-based positioning

We start this section by introducing the fundamental concepts of feature-based positioning and then continued by briefly describing the process of kinematically collecting the RFM.

---

1 Torres-Sospedra and Moreira, 2017 provides a complete discussion of the works focusing on reducing large errors occurred in positioning with the support of other technologies (e.g. PDR, or Bayes filtering).
3.1 Fundamental concepts

Each measured feature is uniquely identifiable and has a measured value. E.g., the signal from an AP, can be identified by its media access control (MAC) address and is associated with an RSS. Features are thus formulated as pairs of attribute $a$ and value $v$, i.e. $(a, v)$. A measurement (i.e. fingerprint) $O_u^i$ taken by the user $u$ at the location/time $i$ consists of a set of measured features, i.e. $O_u^i := \{(a_{uk}^i, v_{uk}^i) | a_{uk}^i \in A; v_{uk}^i \in R; k \in \{1, 2, \ldots, N_u^i\}\}$, where $A$ is the complete set of the identifiers of all available features and $N_u^i$ ($N_u^i = |O_u^i|$) is the number of features observed by the user $u$ at $i$. The set of attributes of $O_u^i$ is defined as $A_u^i := \{a_{uk}^i | \exists (a_{uk}^i, v_{uk}^i) \in O_u^i\}$ ($A_u^i \subseteq A$). The positioning process consists of inferring the estimated user location $\hat{l}_u^i = f(O_u^i)$ ($\hat{l}_u^i \in R^d$) as a function of the fingerprint and the RFM $\mathcal{F}$, where $f$ is a suitable mapping algorithm from fingerprint to location.

3.2 Kinematically acquired RFM

The kinematically obtained dataset used as the RFM herein has been employed in (Zhou and Wieser, 2019). It was acquired using a mobile device (Nexus 6P) whose ground truth location was continuously tracked through the high-precision total station measurements to a mini prism mounted on top of the mobile device. This procedure enables to simultaneously obtain high precision reference coordinates (cm-level) and realistic fingerprinting data from pedestrians. The measurements were obtained at arbitrary locations lying on the trajectory of a pedestrian because the data acquisition scenario on the mobile phone is passively triggered by the status of measurable features (e.g., the arrival of new features or the change of feature values) (Schulz et al., 2018). By carrying out a thorough site-survey, all the collected measurements and their tracked trajectories were merged and used to generate the raw RFM. Herein we use this dataset as the basis of our analysis and more details of its acquisition and processing can be found in (Zhou and Wieser, 2019).

Fig. 1a and 1b show two examples of the raw data of the collected RFM with the RSS values from WLAN APs as the features. These signals are dedicated for providing the Internet access and opportunistically measurable for the purpose of position estimation. The raw measurements are acquired at arbitrary locations throughout the RoI, which consists of several rooms and corridors within an office building.

4 Robust estimation of the feature variability

To estimate the noise of the measurable features at each location throughout the RoI, the features have to be measured (ideally consecutively) multiple times at each location. However, such an RFM is difficult to obtain because measuring the features at fixed locations multiple times, while preserving the characteristics of the motion of pedestrians is very time-consuming and labor-intensive. Though the fingerprints of the

---

5 The RFM $\mathcal{F}$ is omitted from the positioning algorithm $f$ for simplicity. $d$ (e.g., $d = 2$) is the dimension of the coordinates.
Figure 1: Examples of raw and spatially filtered RFM of two APs. Third row shows residuals between the spatially filtered RFM and the raw RFM. The density of the reference locations over the RoI varies due to different accessibility (e.g. areas blocked by furniture or other facilities) and by different visiting frequency of the users.

kinematically collected RFM are not acquired multiple times exactly at the same location, enough number of measurements are obtained in the close of each reference location (see Fig[1]). In addition, the measurements associated to the neighboring reference locations are collected within a short time interval (e.g. less than half an hour) as the data collector passes by the close locations multiple times. These two characteristics of the kinematically collected RFM make it applicable to analyze the variability of the RFM.

The primary idea is to approximate the location-wise STD of each feature according to the measurements associated with the neighborhood of a given reference location by assuming that the expected value of each feature is locally obtainable. More formally, at the reference location $l_j$ in the RFM $F$, the STD $\sigma_k$ of $k$-th ($k = \{1, 2, \cdots, |\tilde{O}_j|\}$) feature is computed. These estimated values of the STD are collected as an extended representation of the RFM, i.e. $F := \{l_j, \tilde{S}_j\}$ with $\tilde{S}_j = \{(a_{jk}, v_{jk}, \sigma_{jk}) | a_{jk} \in \tilde{A}_j\}$. We start by applying the spatial filtering to the raw measurements in order to reduce the large variations (e.g. noise from the mobile device or from the short term fluctuations of the signal sources), which are not originated from the spatial variability. We then proceed with the kernel smoothing (KS) that enables us to obtain the continuous rep-
representation of the RFM and to approximate the expected value of the measurements at any location throughout the RoI, before estimating the location-wise STD for each measurable feature. We perform spatial filtering and KS in two sequential steps for demonstrating the contribution of each step, though, ideally the spatial filtering can be built into the KS. In the following, individual steps of the algorithm are explained in detail.

As can be seen in the raw measurements of the features (Fig.1a and 1b), the measured feature values in the neighborhood of a given location have large variations, especially in the case of the measured value is low and thus close to the sensitivity of the mobile devices. In order to mitigate the impact of these variations on the representation of the RFM, we apply the spatial filtering for replacing the measured feature value \( v_a \) of feature \( a \) at the given location \( l \) with the median value of the ones measured in the neighboring region of \( l \). The support set for the spatial filtering contains the measurements collected at \( m \) closest locations to \( l \) that at the same time lie within the given radius \( r \) centered at \( l \) (see the schematic in Fig.2).

In the second step, we estimate a continuous RFM using KS in order to be able to retrieve the expected measurement at any location within the RoI (Berlinit and Thomas-Agnan, 2011). Albeit KS can partially filter out the noisy measurements contained in the raw RFM, it cannot smoothen out the measurements which contain large variations in the measured features (Fig.3a and 3b). Therefore, we apply KS to the spatially filtered RFM. Because the structure of the indoor region is not taken into account when computing the distance between the neighboring locations, the KS tends to smoothen the RFM over the inaccessible regions (e.g. the structure of the building) or discontinuities. This over-smoothing problem degrades the spatial discontinuity of the features resulting from the indoor environments, which is relevant for positioning (Bong and Kim, 2012), especially when using radio frequency signals such as WLAN, whose propagation is highly influenced by the obstacles. Herein we employ a modified version of KS, which uses only a subset of the RFM in the neighborhood of a given location for approximating the expected feature values. This alleviates the impact of over-smoothing, while at the same time reduces the computational complexity (Berlinit and Thomas-Agnan, 2011; Cormen et al., 2009).

The distribution of the measured noise shown in Fig.4 clearly suggests that the variance of a given feature is location-dependent and is different for different features, i.e. we have to model the STD location-wise and for each individual feature. To this end, we compute the absolute residuals by comparing the spatially smoothed RFM (i.e. KS using the spatially filtered RFM) to the raw RFM in order to derive the robust estimation of the STD. At the reference location \( l_j \) in the RFM \( F \), the STD \( \sigma_{jk} \) of

---

The detailed analysis of the over-smoothing problem and computational complexity of KS is beyond the scope of this paper.
The $k$-th ($k = \{1, 2, \cdots, |\tilde{O}_j|\}$) feature contained in $\tilde{O}_j$ is computed by the median absolute deviation (MAD) of the measured feature values associated to locations defined as the support set for spatial filtering. The extended representation of the RFM with the estimated STD at location $l_j$ is denoted as $\tilde{S}_j = \{(a_{jk}, v_{jk}, \sigma_{jk}) | a_{jk} \in \tilde{A}_j\}$ and is continuously represented using KS, i.e. $\tilde{S}_j := \mathcal{F}(1)$.

In addition, it is possible that the estimated STD of a measurable feature is close to zero as can be seen in Fig.4. This close-zero STD value neither agrees with the empirical observation of the variability of feature values nor benefits for properly estimating the contribution of the individual features to the dissimilarity measure when weighting them according their estimated STD value as described in Section 5. Therefore, a lower bound $\sigma_{\text{min}}$ of the STD is introduced to threshold the estimated STD values. I.e. any of their estimated STD values that is smaller than $\sigma_{\text{min}}$ is replaced by the lower bound. The threshold $\sigma_{\text{min}}$ has been selected experimentally as a reasonable minimum possible values for the STD while additionally establishing a maximum on the amount of features whose estimated STD values are modified. In this way we can potentially discard most of the underestimated values while guaranteeing that this thresholding has only a marginal impact on the iterative scheme for positioning.

5 Iterative scheme for online positioning

Inspired by the finding that the variability of features has the most influential effect on the positioning error in (Kaemarungsi and Krishnamurthy, 2012), we employ the robustly estimated STD of features for the purpose of reducing large errors occurred in feature-based positioning. We construct a weighting scheme that reduces the weight of a feature with high STD relative to features with low STD. Therefore, certain discrepancy between online measured and expected value of a feature with low STD has more impact on dissimilarity measure. This dissimilarity measure is used to identify which subset of reference locations is taken into account when inferring the user’s location using the deterministic feature-based positioning algorithms such as kNN. Because the STD of features contained in an online measurement depends on the ground truth location and it is initially unknown during the online positioning phase, we devise an iterative way of estimating the user’s location. This enables us to apply the location-dependent STD in each iterative step and search for the candidate location estimation but on the other hand it requires us to initialize the user’s coarse location in the first iteration (see Section 5.2).

5.1 Weighted dissimilarity measure

Given the online measured features $\tilde{O}_i^u$ at the location $l_i^u$, the weighted dissimilarity measure $d^w$ between $\tilde{O}_i^u$ and the $j$-th reference fingerprint $\tilde{O}_j$, stored in the RFM is computed as:

$$d^w(\tilde{O}_i^u, \tilde{O}_j) = \sum_{a \in \tilde{A}_i^u \cap \tilde{A}_j} w^u_{ik} g(v_{ik}^u, v_{jk}) + \sum_{a \in \tilde{A}_i^u \setminus \tilde{A}_j} w^u_{ik} g(v_{ik}^u, \gamma) + \sum_{a \in \tilde{A}_j \setminus \tilde{A}_i^u} w^u_{ik} g(\gamma, v_{jk})$$

(1)

where $g$ is the selected dissimilarity measure (e.g. Minkowski distance) and $\gamma$ is the missing value indicator (e.g. -110 dBm). $w^u_{ik}$ is the weight of the $k$-th feature at the location/time $i$ and is computed by employing the variability derived from the estimated expected measurement $\tilde{S}_i^u$ obtained at $l_i^u$. In case that the $k$-th feature in $\tilde{A}_{ij}$
\((\tilde{A}_{ij} := A^u_i \cup \tilde{A}_j)\) is not measurable at location \(l^u_i\), the weight of the corresponding feature is set to the minimum value of weights of the measurable features to minimize their impact on location estimation. (1) computes the dissimilarity measure by dividing it into three parts according to the measurability of the features, which is similar to the compound dissimilarity measure (CDM) [Zhou and Wieser, 2018]. However, the weighting scheme used in CDM takes only the measurability into account and is derived from the pairwise measurability difference. Such a way of calculating the weights makes it difficult to benefit from the matrix operations when implementing the CDM.

In maximum likelihood estimation (MLE) the weights are traditionally calculated inversely proportional to their variance (i.e. \(w_{ik} \propto \sigma^{-2}_{ik}\)). However, the initial experimental analysis shows that the iterative search for positioning is less stable and requires a large number of iterations for terminating the search process when the weights are calculated in this way. One plausible explanation is that the error made in estimating the STD strongly influences the contribution of each feature to the dissimilarity measure, especially when the STD is underestimated. The underestimation of the STD makes the contribution of that feature dominant in the dissimilarity measure. In order to stabilize the iterative search, we approximate and normalize the weight of each feature using the estimated STD in conjunction with the softmax function (Murphy, 2012; Gal and Ghahramani, 2016):

\[
w_{ik}^n = \frac{e^{-\beta \sigma^{-2}_{ik}}}{\sum_{l=1}^{\tilde{S}^u_i} e^{-\beta \sigma^{-2}_{lk}}}
\]

where \(\tilde{S}^u_i = \mathcal{F}(l^u_i)\) and \(\beta > 0\) is the scale factor for adapting the concentration of the softmax function. The normalized weights make the solution be invariant to scaling the weighting. The weighted dissimilarity measure is used to identify the candidate locations, whose dissimilarity values are smallest among all reference fingerprints stored in the RFM. The user’s location is estimated by averaging or weighted averaging (e.g. inversely proportional to the value of dissimilarity measure) of the candidate locations. More details about the \(k\)NN or weighted \(k\)NN can be found in e.g. [Padmanabhan et al., 2000; Zhou and Wieser, 2019].

5.2 Iterative scheme

The ground truth location of the user used for deriving the weights is unknown at online stage. We thus carry out the positioning in an iterative way by assuming the ground truth location of the user and have to initialize the coarse location of the user to obtain the first set of weights. The proposed iterative scheme consists of three relevant modules: i) the initialization; ii) the update step; and iii) the termination condition of the searching processing, which are explained in detail in the following sections.

5.2.1 Determination of the initial location

The initial location \(l^{(0)}_i\) of the user is used to derive the weights for the first iteration. One straightforward way of initializing it is to choose the location estimated by the \(k\)NN without the weighted dissimilarity measure (i.e. the traditional \(k\)NN). In addition, we found out that the termination of the searching process is quite stable when initializing the location randomly (see Section 6), which suggests that the positioning performance is not strongly dependent on the choice of the initial location.
5.2.2 Update step
At the $t$-th iteration ($t \in \mathbb{N}^+$), the weights as well as the dissimilarity are computed according to the variability obtained at the location searched at the $(t-1)$-th iteration. The weight $w_{ik}^{(t)}$ of the $k$-th feature at location/time $i$ and the $t$-th iteration is defined as:

$$w_{ik}^{(t)} = \frac{e^{-\beta \left(\frac{\sigma_{ik}^{(t-1)}}{s_i^{(t-1)}}\right)^2}}{\sum_{l=1}^{\tilde{s}_{i}^{(t-1)}} e^{-\beta \left(\frac{\sigma_{il}^{(t-1)}}{s_l^{(t-1)}}\right)^2}}$$

(3)

where $\tilde{s}_{i}^{(t-1)}$ is the estimated expected value of features with their STD at location $\hat{l}_{i}^{(t-1)}$. This updated weights are used to compute the dissimilarity measure as defined in (1) and consequently to infer the estimated location $\hat{l}_{i}^{(t)}$ at the $t$-th iteration using e.g. kNN algorithm.

5.2.3 Termination condition
In order to ensure that the iteratively searching process is accomplished in a reasonable time, we have to determine when to stop the iteration. Ideally the searching process should converge either to one fixed location or to the case that distance between two consecutively searched locations is under a given small threshold (called converging state). The iterative process is thus terminated when

$$|\hat{l}_{i}^{(t)} - \hat{l}_{i}^{(t-1)}|_2 < d_{\text{min}},$$

where $d_{\text{min}}$ is the threshold, which is set to $10^{-3}$ m in the experimental analysis. In addition, the empirical observation shows that the iterative process proposed herein typically enters a loop, in which a subset of locations are repeatedly visited in the same sequence (called looping state). This means that the iterative search process runs continuously if it was not terminated intentionally. We therefore introduce another termination condition, which is the threshold on the minimum distance between the location obtained at the iteration $t$ and the ones estimated at previous iterations except the estimated location at the $(t-1)$-th iteration. More formally, the second condition is satisfied and the iteration is terminated when

$$\min_{m=1,\ldots,T'} \{|\hat{l}_{i}^{(t)} - \hat{l}_{i}^{(m)}|_2 \} < d_{\text{min}}.$$

Furthermore, the maximum number iterations $T$ is limited (e.g. $T = 100$) in order to prevent too long or endless iterative search (called max. state). We then determine the final estimation of the user’s location $\hat{l}_{i}^{u}$ according to the searched locations $\hat{l}_{i}^{u} := \{\hat{l}_{1}^{(1)}, \hat{l}_{2}^{(2)}, \ldots, \hat{l}_{i}^{(T')}\}$, where $T'$ is the number of iterations until the termination condition is fulfilled. The results obtained if the iteration has terminated because of looping state or of reaching the maximum number of iterations conditions is not of the same quality as compared to the converged cases. Each termination condition is thus denoted by a termination flag (TF), i.e. the termination condition of estimating the user u’s location at location/time $i$ is indicated by $\varepsilon_{iu}^{u} \in \{0, 1, 2\}$. This TF value can be potentially used to flag the outliers. The final estimated user’s location $\hat{l}_{i}^{u}$ as well as its TF $\varepsilon_{iu}^{u}$ are determined with respect to each individual termination condition:
• **Converging state**: The location estimated at $T'-\text{th}$ iteration is treated as the final estimated user’s location $\hat{l}_u$ and $\varepsilon_u$ is set to 0.

• **Looping state**: In this case the searched locations do not converge to a single point and the locations consisting of the looping state are diversely distributed (see the black polygon in Fig.6). We thus use the minimum covariance determinant (MCD) estimator for computing the estimated location $\hat{F}_u$ of the user according to the locations contained in the convex hull (denoted as the black polygon in Fig.6 and 8). The convex hull consists of the locations that are repeatedly searched in the same sequence. In addition, MCD fails to robustly estimate the mean of the searched locations within polygon when the number of them is less than the minimum required number. The failure of MCD is handled the same as the max. state. In addition, the TF $\varepsilon_u$ is set to 1 in this case.

• **Max. state/failure of MCD**: In these two cases, we have to determine $\hat{l}_u$ from all searched locations $\hat{L}$. Because the searched locations are quite diversely distributed (see Fig.6 and 8), we cannot simply take the estimation from the last iteration as the final estimated location. Therefore, we analyze the similarities between the user measured fingerprint $O_{u}$ and the expected ones at these searched locations for determining the final estimation. Specifically, we employ the modified Jaccard index (MJI), which has been used for identifying subregions according to the measurability of features (Zhou and Wieser, 2019), as the similarity metric. The MJI value $s_{\text{MJI}}$ between $O_{u}$ and the expected one $\tilde{O}_t := \mathcal{H}(\hat{F}_t)$ at the searched location $\hat{F}_t$ of the $t$-th iteration is computed by:

$$s_{\text{MJI}}(O_{u}, \tilde{O}_t) = \frac{1}{2} \left( \frac{|A_u \cap \tilde{A}_t|}{|A_u \cup \tilde{A}_t|} + \frac{|\tilde{A}_t \cap \tilde{A}_t|}{|\tilde{A}_t|} \right)$$

(4)

where $A_u$ and $\tilde{A}_t$ are the sets of the measured features contained in $O_{u}$ and $\tilde{O}_t$, respectively. The estimated user’s location $\hat{F}_u$ is then the one that has the biggest MJI value among all searched locations $\hat{L}$, i.e. the one has the maximum number of overlapped measurable features is selected as final estimation of the user’s location.

6 Analysis of the variability estimation and positioning performance

We firstly present the results of location-wise variability of each individual feature estimated according to the kinematically collected RFM, which has been detailedly discussed in (Zhou and Wieser, 2019). A brief description for how to experimentally determine the lower bound $\sigma_{\text{min}}$ of the STD, which is used to mitigate the impact of underestimation of STD value, is also included. A detailed analysis concerning the characteristics of iterative searched locations as well as the positioning performance of the proposed iterative scheme is presented in the subsequent section.

---

4 MCD is a highly robust estimation of multivariate location and scatter. We use the implementation of MCD from scikit-learn (Pedregosa et al., 2011).
6.1 Results of the variability estimation

Herein we empirically set $m = 20$ and $r = 2$ m to obtain the spatially filtered RFM, which has adequate spatial consistency in the neighborhood of each location. Fig. 1c and 1d show clear that the spatial filtering can reduce the large variations contained in the raw RFM to a large extent. For further analysis, we compute the residuals between the raw and the spatially filtered one. The obtained residuals are close to zero-mean distributed and have a location-dependent magnitude over the space as illustrated in Fig. 1c and 1f. The large residual occur either in the region close to the boundaries of the RoI (e.g. close to the walls or corners of rooms and corridors) or at the locations where the RSS values are hardly measurable by the mobile phone. In both cases the features are very likely effected by the obstacles and thus yield large variations.

![Examples of kernel smoothed RFM of two APs](image)

Figure 3: Examples of kernel smoothed RFM of two APs. The results of the first two rows are yielded by taking the raw RFM and the spatially filtered one (i.e. the ones depicted in the first two rows of Fig. 1) as the input to KS, respectively. Third row shows the residual between the kernel smoothed RFM using the spatially filtered one and the spatially filtered RFM. Though the KS provides the continuous representation of the RFM, we only visualize the smoothed features at the locations contained in the raw RFM for easy comparison.

![Residual](image)

Fig. 4 shows the results of the estimated STD value using the MAD of the measured
feature values associated to the neighborhood of a given location. As can be seen in the figure, each feature has different variability throughout the RoI, i.e. the STD value is dependent both on feature as well as on the location. This is the primary motivation that the variability is modeled location-wise for each individual feature instead of simply relating the variability as a function of the measured feature value or as a constant value. The region where the feature values have a high STD is obvious correlated to the local variations of the measured feature value as well as the geometric of the building as can be seen when combining the estimated STD values in Fig.4 and their corresponding continuous representation in Fig.3 (second row). Especially, the high variances are more likely to occur in the case either the RSS values large residuals or little number of measurements has been collected in the neighborhood region. The latter leads to the failure of the assumption that the expected feature values are locally obtainable.

![Absolute residual](image1)

![Estimated STD](image2)

Figure 4: Examples of the absolute residuals and estimated STD for two APs

Furthermore, it can happen that the STD values are underestimated to a very small value (e.g. close to zero) even if the feature is measurable as shown in Fig.4. As aforementioned in Section 4, we have experimentally determined a lower bound $\sigma_{\text{min}}$ of the STD value for mitigating the impact of underestimating the STD values on deteriorating the positioning performance of the proposed approach. The determined $\sigma_{\text{min}}$ value of the kinematically collected RFM is 0.74, which is the low-tail (e.g. 2.5-percentile) of the distribution (depicted in Fig.5) of the estimated STD values (only non-zero ones) for all measurable features stored in the RFM. We have made a compromise between two factors when determining $\sigma_{\text{min}}$ in this way. On the one hand, it has low risk of messing up the weighting scheme when using 2.5-percentile as the lower bound even in case of this value is not equal to the true one though the ground truth STD values of each feature is unknown. On the other hand, the experimental analysis has verified that thresholding the estimated STD values using the determined lower bound can mitigate the diversity level of the iteratively searched locations at the looping state as can be seen in Fig.6.
The proposed iterative scheme for feature-based positioning is implemented using the application programming interface of scikit-learn package, a widely used machine learning package in Python (Pedregosa et al., 2011). Herein we present the results of the iterative positioning using \( k \)NN with the weighted Euclidean distance as the dissimilarity metric for measuring the distance in the feature space. The number of nearest neighbors \( k \) in \( k \)NN algorithm and the scale factor \( \beta \) of the softmax function are empirically set to 1 and 2.0, respectively for the sake of preserving high ratio of converging state (Fig.7) as well as obtaining fairly good results. Optimization of the parameters (e.g. using grid/random search or Bayesian optimization (Wang and de Freitas, 2014)) for achieving the best positioning performance is left for future work.

Figure 6 illustrates the influence of thresholding the estimated STD values according to the selected lower bound \( \sigma_{\min} \) on the positioning of iterative scheme when setting \( k = 1 \) and 3 for \( k \)NN. The results presented in Fig.6 are obtained by initializing the iterative search at the same location. Each individual subplot depicts the results of the iterative positioning at a fixed test location. The subplot illustrates the searched locations (red squares), the final estimation (blue triangle or coral diamond), the estimation of the traditional \( k \)NN, the ground truth (black square) as well as the convex hull consisting of the repeatedly searched locations. From the results shown in Fig.6, we can conclude that: i) the iteratively searched locations as well as those consisting of the looping state are diversely distributed. This characteristic makes it difficult to accurately retrieve the final estimation of the user’s location from these searched locations; and ii) both the thresholding scheme and number of nearest neighbors used for \( k \)NN have impacts on the termination state. One the one hand, the thresholding scheme makes the iterative search become converging state from looping state without thresholding when \( k \) is equal to 1. On the other hand, it takes more iterations to terminate the searching process when \( k = 3 \) as compared to that of \( k = 1 \). In addition, we present the results of the TF at different locations in Fig.7. The ratio of locations terminated with different condition is calculated. Fig.7 shows that about two times more positioning cases that the searching processes converge to a fixed location when \( k = 1 \) than that of \( k = 3 \). In case of \( k = 1 \), all the searched processes do not reach the maximum number of iterations. The smaller number of iterations to perform and higher chance to converge are the two convincing

\footnote{In order to improve the readability, the initial location has not been visualized. This also holds for Fig.8}
points to set $k = 1$ with the thresholding scheme in the later analysis of the positioning performance using the iterative scheme.

Figure 6: Impact of thresholding the STD values on the positioning results of iterative scheme. The convex hull (i.e. the black polygon) is formed by the locations, which are repeatedly searched in the same sequence (i.e. the looping state) when using the iterative scheme.

Figure 7: Comparison of the ratio locations terminated with different conditions for $k = 1$ and 3

Fig. 8 shows several examples of the searched locations of the iterative positioning with the random as well as $k$NN initialization. Each individual subplot depicts the results of the iterative positioning at a fixed test location. The initialization of the initial location only has a minor impact on the iterative searching process. Indeed, the looping states denoted by the black colored convex hull are similar when comparing different initializations. It also can happen that the searched locations are densely distributed together (e.g. Fig. 8a). Our schemes for determining the final estimation of the user’s location from these searched locations do not achieve the best potential position-
ing performance using the iterative scheme. Because there are locations in \( \hat{L} \) that are closer to the ground truth but they are not taken as the final estimation (e.g. second row in Fig.8). This suggests that the iterative scheme for positioning has the potential to further improve the positioning performance if the proper technique is applied to retrieve the final estimation from these searched locations. The optimal positioning accuracy (denoted as Ours (opt.) in Fig.9 and in TABLE 1) achieved by assuming that the technique for the final estimation is capable of retrieving the searched location that is the closest to the ground truth is depicted in Fig.9. It shows that the best achievable positioning accuracy within 3 m is over 99%, which is about 10% locations higher than that of using the current way for the final estimation.

![Random initialization 1](image1)
![Random initialization 2](image2)
![kNN initialization](image3)

![Loc. 1](image4)
![Loc. 2](image5)
![Loc. 3](image6)

Figure 8: Examples of iteratively searched location with different initializations. The convex hull is formed by the locations, which are repeatedly searched in the same sequence (i.e. the looping state) when using the iterative scheme.

The empirical cumulative distribution function (ECDF) of the radial positioning errors is presented in Fig.9. Compare to the performance of the traditional \( k \)NN, our approach can significantly improve the positioning performance. Using the algorithm proposed herein, the percentage of locations whose positioning error is less than 2 m and 4 m are about 80% and 94%, and it represents an improvement of up to 20 and 10 percent locations when compared to the traditional \( k \)NN, respectively. The percentage of locations whose error distance is larger than 5 m is reduced from about 11% to 4% as compared to the traditional \( k \)NN. We also report the circular error (CE) defined as the minimum radius for including a given percentage of positioning errors (e.g. CE 50 for the 50th percentile) in TABLE 1. The maximum positioning error is reduced by more than one-fold, from 29.4 m to 13.4 m. The CE 50, CE 75, and CE 90 are...
reduced by about one-third when compared to the $k$NN without iterative positioning. Furthermore, in Fig.10 we illustrate and compare the distribution of the locations, at which the positioning error is large than 8 m using the original $k$NN. Fig.10a shows that these locations yielding large errors are mostly located close to the accessible boundaries of the indoor regions, i.e. close to corners of corridors and rooms, or to the walls. This pattern is similar to the spatial distribution of high variance of the feature values contained in the raw RFM as shown in Fig.4. Our approach can significantly reduce the positioning errors at these locations.

Figure 9: Comparison of ECDF with respect to the radial positioning errors.

|                  | CE 30 | CE 75 | CE 90 | Max. error |
|------------------|-------|-------|-------|------------|
| $k$NN            | 1.5   | 2.9   | 5.3   | 29.4       |
| Ours             | 1.1   | 1.8   | 3.0   | 13.7       |
| Ours (opt.)      | 0.8   | 1.3   | 1.9   | 8.6        |

Table 1: Statistics of positioning errors (meters)

Figure 10: Distribution of the locations yielding large errors in positioning

7 Conclusion

We have proposed an iterative scheme for feature-based positioning, which is based on the weighted dissimilarity measure, for reducing large errors occurred in the FIPSs. In order to derive the proper weighting scheme for each feature, we have analyzed the variability of the RFM, which is collected by kinematically acquired in an office.
building. By applying the spatial filtering to the raw measurement, we found that the measuring noise has a diverse spatial distribution for each feature. In addition, we apply KS in order to obtain a continuous RFM, which enables retrieving feature values at any location within the region of interest. The location-wise standard deviation of each feature is robustly computed as the MAD between the raw RFM and the spatially smoothed one. This variability information is stored as an additional layer of the RFM and used for weighting the contribution of each feature to dissimilarity measure during the online positioning phase.

We have validated the performance of the proposed iterative scheme with an experimental analysis. The maximum positioning error is reduced by more than one-fold and the iterative scheme can improve the overall positioning performance. The positioning accuracy defined as the percentage of the locations whose radial positioning error is less than 2 m is improved from 61% to 80% when compared to traditional kNN. In future work, we will further investigate the detection of large error cases according to different termination conditions. Furthermore, we will investigate the possibility to improve the determination of the final estimated location and hence improving the positioning accuracy.

Acknowledgment

The Chinese Scholarship Council has supported C. Zhou during his doctoral studies at ETH Zürich.

References

Berlinet, A. and Thomas-Agnan, C. (2011). Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer Science & Business Media, Berlin Heidelberg.

Bitew, M. A., Hsiao, R.-S., Lin, H.-P., and Lin, D.-B. (2015). Hybrid indoor human localization system for addressing the issue of rss variation in fingerprinting. International Journal of Distributed Sensor Networks, 11(3):831423.

Bong, W. and Kim, Y. C. (2012). Fingerprint wi-fi radio map interpolated by discontinuity preserving smoothing. In Convergence and Hybrid Information Technology, pages 138–145, Berlin, Heidelberg. Springer Berlin Heidelberg.

Brena, R. F., García-Vázquez, J. P., Galván-Tejada, C. E., Muñoz-Rodriguez, D., Vargas-Rosales, C., and Fangmeyer, J. (2017). Evolution of indoor positioning technologies: A survey. Journal of Sensors, 2017.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). Introduction to Algorithms, Third Edition. The MIT Press, 3rd edition.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059.

Guan, K., Ma, L., Tan, X., and Guo, S. (2016). Vision-based indoor localization approach based on surf and landmark. In 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), pages 655–659. IEEE.
He, S. and Chan, S.-H. G. (2016). Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490.

He, S., Ji, B., and Chan, S. H. G. (2016). Chameleon: Survey-free updating of a fingerprint database for indoor localization. *IEEE Pervasive Computing*, 15(4):66–75.

He, S. and Shin, K. G. (2018). Geomagnetism for smartphone-based indoor localization: Challenges, advances, and comparisons. *ACM Computing Surveys (CSUR)*, 50(6):97.

Kaemarungsi, K. and Krishnamurthy, P. (2012). Analysis of wlan’s received signal strength indication for indoor location fingerprinting. *Pervasive and Mobile Computing*, 8(2):292 – 316. Special Issue: Wide-Scale Vehicular Sensor Networks and Mobile Sensing.

Lemic, F., Handziski, V., Aernouts, M., Janssen, T., Berkvens, R., Wolisz, A., and Famaey, J. (2019). Regression-based estimation of individual errors in fingerprinting localization. *IEEE Access*, 7:33652–33664.

Li, X., Wang, J., Liu, C., Zhang, L., and Li, Z. (2016). Integrated wifi/pdr/smartphone using an adaptive system noise extended kalman filter algorithm for indoor localization. *ISPRS International Journal of Geo-Information*, 5(2):8.

Li, Y., Gao, Z., He, Z., Zhuang, Y., Radi, A., Chen, R., and El-Sheimy, N. (2019). Wireless fingerprinting uncertainty prediction based on machine learning. *Sensors*, 19(2):324.

Mautz, R. (2012). Indoor positioning technologies. *Habilitation Thesis, Department of Civil, Environmental and Geomatic Engineering, ETH Zurich, Switzerland*.

Minaev, G., Visa, A., and Pich, R. (2017). Comprehensive survey of similarity measures for ranked based location fingerprinting algorithm. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–4.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Padmanabhan, P. B., N., V., and N., V. (2000). RADAR: An in-building RF based user location and tracking system. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, 2(c):775–784.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pei, L., Zhang, M., Zou, D., Chen, R., and Chen, Y. (2016). A survey of crowd sensing opportunistic signals for indoor localization. *Mobile Information Systems*, 2016.

Retscher, G. and Joksch, J. (2016). Comparison of different vector distance measure calculation variants for indoor location fingerprinting. In *Location-Based Services (LBSs), 2016 International Conference on*, pages 53–76. ICA Commission on LBSs.
Röbesaat, J., Zhang, P., Abdelaal, M., and Theel, O. (2017). An improved ble indoor localization with kalman-based fusion: An experimental study. *Sensors*, 17(5):951.

Schulz, M., Link, J., Gringoli, F., and Hollick, M. (2018). Shadow wi-fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over wi-fi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys ’18, pages 256–268, New York, NY, USA. ACM.

Tao, Y. and Zhao, L. (2018). A novel system for wifi radio map automatic adaptation and indoor positioning. *IEEE Transactions on Vehicular Technology*, 67(11):10683–10692.

Torres-Sospedra, J., Jiménez, A., Knauth, S., Moreira, A., Beer, Y., Fetzer, T., Ta, V.-C., Montoliu, R., Seco, F., Mendoza-Silva, G., et al. (2017). The smartphone-based offline indoor location competition at ipin 2016: Analysis and future work. *Sensors*, 17(3):557.

Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, A., and Huerta, J. (2015). Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, 42(23):9263–9278.

Torres-Sospedra, J. and Moreira, A. (2017). Analysis of sources of large positioning errors in deterministic fingerprinting. *Sensors*, 17(12):2736.

Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., and Choudhury, R. R. (2012). No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210. ACM.

Wang, J., Hu, A., Liu, C., and Li, X. (2015). A floor-map-aided wifi/pseudo-odometry integration algorithm for an indoor positioning system. *Sensors*, 15(4):7096–7124.

Wang, Z. and de Freitas, N. (2014). Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters. *CoRR*, abs/1406.7758.

Wu, C., Yang, Z., Zhou, Z., Liu, Y., and Liu, M. (2017). Mitigating large errors in wifi-based indoor localization for smartphones. *IEEE Transactions on Vehicular Technology*, 66(7):6246–6257.

Youssef, M. and Agrawala, A. (2008). The Horus location determination system. *Wireless Networks*, 14(3):357–374.

Zhou, C. and Wieser, A. (2018). Cdm: Compound dissimilarity measure and an application to fingerprinting-based positioning. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7.

Zhou, C. and Wieser, A. (2019). Modified jaccard index analysis and adaptive feature selection for location fingerprinting with limited computational complexity. *Journal of Location Based Services*, 13(2):128–157.

Zhuang, Y., Yang, J., Li, Y., Qi, L., and El-Sheimy, N. (2016). Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors*, 16(5):596.